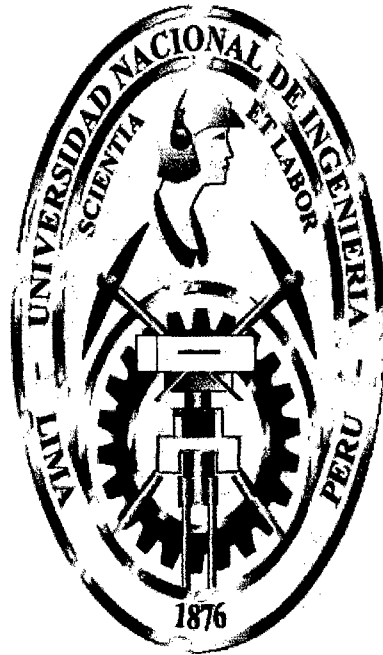


**UNIVERSIDAD NACIONAL DE INGENIERÍA**

**FACULTAD DE INGENIERÍA MECÁNICA**



**“DISEÑO DE UN SISTEMA AUTÓNOMO INTELIGENTE  
SELECTOR DE FRUTAS USANDO dsPIC”**

**TESIS**

**PARA OPTAR EL TÍTULO PROFESIONAL DE:**

**INGENIERO MECATRÓNICO**

**RUDECINDO ALFONSO CÓRDOVA FLORES**

**PROMOCIÓN 2008 – I**

**LIMA – PERÚ**

**2011**

**Digitalizado por:**

**Consortio Digital del  
Conocimiento MebLatam,  
Hemisferio y Dalse**

El presente trabajo se lo dedico a mi familia en especial a mi padre, a mi madre y a mis hermanos por haberme apoyado en todo momento, por enseñarme que todo esfuerzo no es en vano, por entrenarme para ser un hombre de bien, por su constante motivación para seguir adelante, pero más que nada, por el inmenso amor incondicional que me ofrecen.

## ÍNDICE

|   |           |
|---|-----------|
| Prólogo .....   | 1         |
| <b>CAPÍTULO 1: INTRODUCCIÓN .....</b>   | <b>4</b>  |
| 1.1 Antecedentes .....  | 5         |
| 1.2 Planteamiento del Problema .....  | 7         |
| 1.3 Justificación .....   | 7         |
| 1.4 Objetivos .....   | 8         |
| 1.4.1 Objetivo General.....   | 8         |
| 1.4.2 Objetivos Específicos .....   | 8         |
| 1.5 Alcances y Limitaciones .....   | 9         |
| 1.5.1 Alcances del Sistema.....   | 9         |
| 1.5.2 Limitaciones del Sistema.....   | 9         |
| <b>CAPÍTULO 2: DISEÑO MECÁNICO DEL SISTEMA .....</b>  | <b>11</b> |
| 2.1 Descripción del Sistema Mecánico .....  | 11        |
| 2.2 Cálculos para el Diseño del Sistema Mecánico .....  | 16        |
| 2.2.1 Pruebas Experimentales del Sistema Mecánico .....   | 16        |
| 2.2.2 Selección del Motor Paso a Paso para el Mecanismo de Corte de las Frutas (Mecanismo Biela – Manivela) ..... | 23        |
| 2.2.3 Selección del Motor Paso a Paso para el Soporte de Giro de la Licuadora .....                               | 49        |
| 2.3 Componentes del Sistema Mecánico .....  | 62        |
| 2.3.1 Estructura del Sistema Mecánico .....   | 62        |
| 2.3.2 Sistema Selector de Frutas .....  | 69        |
| 2.3.3 Licuadora con Soporte de Giro .....   | 70        |
| 2.3.4 Mecanismo de Separación y Posicionado de Vasos.....   | 72        |
| <b>CAPÍTULO 3: DISEÑO ELECTRÓNICO DEL SISTEMA .....</b>   | <b>73</b> |
| 3.1 Microcontroladores y DSP .....  | 73        |
| 3.2 Fuente de Poder .....   | 78        |
| 3.3 Circuito de Control del Sistema .....   | 82        |
| 3.4 Módulo MPLAB ICD3.....  | 84        |

|  |            |
|--|------------|
| 3.5 Entradas y Salidas Digitales .....                                     | 86         |
| 3.5.1 Entradas Digitales.....  | 87         |
| 3.5.2 Salidas Digitales .....  | 97         |
| 3.6 Motores Paso a Paso .....  | 103        |
| 3.6.1 Tipos y Principio de Funcionamiento de los Motores Paso a Paso ..... | 103        |
| 3.6.2 Controlador de Motores Paso a Paso .....                             | 111        |
| <b>CAPÍTULO 4: DISEÑO DEL SISTEMA DE CONTROL .....</b>                     | <b>118</b> |
| 4.1 Entradas y Salidas del dsPIC para el Control del Sistema .....         | 118        |
| 4.2 Diagrama de Flujo del Sistema .....                                    | 120        |
| 4.3 Programación del dsPIC en Lenguaje C MPLAB IDE .....                   | 123        |
| 4.3.1 Componentes del MPLAB IDE .....                                      | 123        |
| 4.3.2 El Lenguaje de Programación C30.....                                 | 125        |
| 4.3.3 Creando un Proyecto en MPLAB IDE .....                               | 128        |
| 4.4 Funcionamiento del Sistema para el Preparado de un Jugo.....           | 138        |
| 4.5 Diagrama de Tiempos para el Sistema Selector de Frutas.....            | 147        |
| <b>CAPÍTULO 5: ESTUDIO ECONÓMICO - FINANCIERO .....</b>                    | <b>148</b> |
| 5.1 Introducción .....   | 148        |
| 5.2 Costo del Sistema Implementado.....                                    | 149        |
| 5.3 La Empresa y el Sistema Selector de Frutas .....                       | 154        |
| 5.3.1 Oferta del Producto .....  | 154        |
| 5.3.2 Factores de Producción .....   | 154        |
| 5.3.3 Inversión .....  | 157        |
| 5.3.4 Cálculo de la Estructura de Costos y Precio de Referencia .....      | 157        |
| 5.3.5 Estimación de la Producción de Equilibrio ( $Q_e$ ).....             | 159        |
| 5.3.6 Meta de Utilidades o Ganancias .....                                 | 160        |
| 5.3.7 Evaluación Económica .....   | 162        |
| <b>CONCLUSIONES .....</b>  | <b>164</b> |
| <b>RECOMENDACIONES .....</b>   | <b>167</b> |
| <b>BIBLIOGRAFÍA .....</b>  | <b>168</b> |
| <b>ANEXOS .....</b>  | <b>170</b> |

## PRÓLOGO

El propósito de esta tesis es diseñar un sistema autónomo inteligente, basado en una máquina expendedora de jugos de fruta de modo que el consumo sea al instante, es decir, se desea diseñar un sistema que va a trabajar con 5 tipos de frutas y un líquido tratado para la combinación del jugo, de tal forma, que se pueden hacer jugos hasta de un solo tipo de fruta, ello dependerá de la selección deseada; nuestro sistema es netamente autónomo, porque el uso comienza con el ingreso de unas monedas, precio del jugo deseado, luego la selección del jugo de frutas que se quiere, para que el sistema comience a trabajar seleccionando las frutas deseadas y luego verterlos en la licuadora para la preparación del jugo, posteriormente servirse en un vaso y todo esto automáticamente sin la intervención humana. Cabe mencionar que nuestro sistema trabaja con las frutas peladas y picadas, estos se colocarán en distintos recipientes similares y acomodados de tal manera que puedan trabajar en conjunto.

Este sistema será controlado con un microcontrolador, un dsPIC de microchip, el control será netamente digital. Como entradas digitales tendremos pulsadores, interruptores, sensores capacitivos, bits o líneas de un selector de monedas para dar inicio al trabajo del sistema. En cuanto a las salidas digitales tendremos bits y pulsos para dar dirección y giro a los motores paso a paso, bit para activar la válvula solenoide, bit para activar la licuadora, bit para activar la alarma en caso de fallas, etc.

En el primer capítulo tenemos la parte introductoria donde se mostrará los antecedentes del sistema, el planteamiento de problema a resolver, la justificación, los objetivos a alcanzar con el desarrollo de esta tesis, el alcance y las restricciones del sistema.

En el segundo capítulo tenemos el diseño de la parte mecánica, donde se muestra el diseño completo del sistema mecánico con todas sus partes como: los recipientes para las frutas y el líquido, el soporte para los recipientes, el soporte para el mecanismo de corte, los mecanismos de acción y corte de frutas para el llenado de estas a la licuadora, el soporte de giro de la licuadora para verter el jugo y el mecanismo de separación y posicionado de vasos para el servido del jugo. También se muestra algunos cálculos para determinar los volúmenes de los recipientes, cálculos para seleccionar los motores paso a paso tanto para el mecanismo de acción y corte de las frutas como para el soporte de giro de la licuadora, además seleccionaremos el material a utilizar en los diseños, debido a que se trabaja con un producto alimenticio.

En el tercer capítulo tenemos el diseño electrónico, donde se muestra el diseño del hardware del controlador del sistema cuyo cerebro es un dsPIC de microchip, que va desde el circuito del oscilador, la fuente de poder, las entradas digitales, las salidas digitales, los acondicionadores de señal tanto para los sensores como para los actuadores, hasta el circuito de potencia para el control electrónico de los motores paso a paso. La mayoría los circuitos se han dibujado y simulado en el programa Proteus y algunos en el Eagle.

En el cuarto capítulo tenemos el diseño del sistema de control, donde se muestra todas las señales que tenemos para realizar nuestra lógica de control (entradas y salidas digitales), el diagrama de flujo de la lógica de control que regirá el sistema y luego pasarlo a desarrollar en el software MPLAB IDE de Microchip en el lenguaje C con la explicación de cada línea de programación. También mostramos lo que resulta de nuestro diseño de control del sistema con el funcionamiento normal para el preparado de un jugo y finalmente mostramos un diagrama de tiempos del sistema para resumir nuestro diseño de control del sistema.

En el quinto capítulo tenemos el estudio económico - financiero, donde se muestra los costos de los componentes, el costo de la manufactura y el costo de la implementación, para mostrar un costo total del sistema implementado, luego se muestra un proyecto para crear una empresa con este sistema donde analizamos los costos fijos, los costos variables, le asignamos un precio de mercado para sacar un punto de equilibrio de las ventas de los jugos que realiza el sistema, por último realizamos un análisis del VAN (Valor Actual Neto), el TIR (Tasa Interna de Retorno) y el  $t_r$  (Tiempo de recuperación) para evaluar si es rentable realizar una empresa con el sistema.

Finalmente se indican las conclusiones presentadas a lo largo del proyecto y algunas recomendaciones acerca de una futura implementación.

## **CAPÍTULO 1**

### **INTRODUCCIÓN**

El presente trabajo pretende la automatización de una máquina para hacer jugos de frutas y que el despacho del mismo sea al instante, el mercado de la venta de jugos de frutas es muy rentable debido a la necesidad de las personas a ingerir productos naturales, saludables y deliciosos; la mayoría de personas que consume jugos naturales tiene que ir a un mercado o a un local de venta de jugos donde tiene que lidiar con el que hace el jugo, si se lavó o no las manos, las moscas, la mala higiene de los recipientes donde se sirve el jugo, etc.

Analizando el caso desde el punto de vista del vendedor, este también tiene que lidiar con el cobro del jugo, si se cobro o no, si la moneda es falso, lidiar con el preparado del jugo, si se olvidó alguna fruta, o lleno en poca proporción, limpieza, etc.

Es por eso que se pretende implementar un sistema inteligente, autónomo, útil, novedoso e higiénico para el preparado de jugos con distintos tipos de frutas, una máquina expendedora para jugos de fruta fresca.



### 1.1. Antecedentes

Hoy en día el avance tecnológico los vemos más común en cualquier centro comercial, hemos visto máquinas “automáticas” como las máquinas expendedoras, estas proporcionan aperitivos, bebidas, golosinas y otros productos a los consumidores. Se trata de vender sin la presencia de un dependiente para cobrar los artículos. Periódicamente un empleado repone el producto y recoge el dinero en forma de monedas o, menos habitualmente, billetes; a veces también se puede pagar con tarjeta monedero, tarjeta de crédito o teléfono móvil.

Las posibilidades de las máquinas expendedoras son amplias. Normalmente suelen vender refrescos, café, comida, etc. Pero también existen modelos diseñados para vender periódicos, libros, sellos de correos, billetes del transporte público, bebidas alcohólicas, cigarrillos de tabaco, también son frecuentes, en las oficinas que atienden al público, las máquinas expendedoras de un impreso pequeño con el número de turno del solicitante. Una máquina expendedora en inglés se conoce como “vending machine”. Por eso, la palabra “vending” se utiliza con frecuencia para referirse al sector de las máquinas expendedoras.

El vending podría tener su origen en Egipto, pues la primera máquina expendedora de que se tiene constancia escrita fue diseñada por Herón de

Aleandría para dispensar agua bendita en los templos de Tebas y el alto Egipto.

A pesar de este inicio, las máquinas expendedoras se desarrollan con la Revolución industrial. En Londres, (Inglaterra), al principio de la década de 1880, se instalaron las primeras máquinas modernas que vendían tarjetas postales. En Estados Unidos, en 1888, la compañía Thomas Adams Gum Company instala máquinas dispensadoras de chicle o goma de mascar en los andenes del metro de Nueva York. En 1897, se añaden unas figurillas animadas a las máquinas para llamar la atención y favorecer la compra, este es el precedente de las máquinas tragamonedas que existen en la actualidad. En 1902 abre en Philadelphia un restaurante con funcionamiento exclusivamente a través de máquinas expendedoras; se mantuvo abierto hasta 1962. En 1907, se introdujeron los chicles en forma de bolas de colores recubiertos de una capa de caramelo. Poco después las máquinas expendedoras ofrecían todo tipo de artículos. En 1920, aparecen las primeras máquinas automáticas que venden bebidas gaseosas servidas en vasos desechables. En 1926, se inventa la primera máquina de venta de cigarrillos. La venta de bebidas gaseosas embotelladas, enfriadas con hielo, comienza en 1930; poco después los refrigeradores sustituyen al hielo. En 1946, las máquinas dispensadoras de café caliente marcan un hito en la historia del “vending” se extienden por todo el mundo. En la década siguiente hace su aparición la primera máquina refrigerada de venta de sándwiches.

## 1.2. Planteamiento del Problema

Debido a la gran demanda de productos alimenticios naturales y saludables se desea implementar un sistema autónomo inteligente para la elaboración de jugos y que el despacho y consumo de ellos sea al instante. El sistema que se pretende diseñar es un sistema que va a trabajar con 5 tipos de frutas (papaya, piña, manzana, plátano y melón) y un líquido tratado para la combinación del jugo, de tal forma, que se pueden hacer jugos hasta de un solo tipo de fruta, ello dependerá de la selección deseada, para tal fin el sistema constará de un selector de monedas, un sistema selector de frutas, una licuadora potente con soporte de giro para el vaciado del jugo y un mecanismo de 2 grados de libertad que saca y sujeta el vaso donde se servirá el jugo.

¿Será el sistema capaz de superar el tiempo, cantidad, calidad y precisión de la mezcla de frutas en la preparación de jugos por un humano que lo realiza en 3 minutos?

## 1.3. Justificación

La gran demanda de productos naturales y la necesidad de un sistema moderno, autónomo, novedoso, útil e higiénico para el preparado de jugos y se utilice como las demás máquinas expendedoras que existen en el mercado y para todo público.

## **1.4. Objetivos**

### **1.4.1. Objetivo General**

Diseñar un sistema autónomo inteligente de preparado de jugos de hasta 5 frutas frescas (papaya, piña, manzana, plátano y melón) y se venda sin la presencia de un dependiente para cobrar por el jugo, este sistema constará de un selector y clasificador de monedas, un sistema selector de frutas con mecanismos de acción y corte para la separación de las frutas, una licuadora con giro controlado por un motor paso a paso para el vaciado del jugo y un mecanismo de separación y posicionado del vaso donde se servirá el jugo.

### **1.4.2. Objetivos Específicos**

- Diseñar un sistema mecatrónico de selección y corte de frutas para llevarlos a la licuadora para el preparado del jugo.
- Diseñar un sistema mecatrónico de giro de la licuadora para el vaciado del jugo al vaso donde se consumirá el jugo.
- Diseñar un sistema mecatrónico para sacar un vaso y llevarlo al punto para la recepción del jugo.
- Reducir los costos de implementación del sistema, seleccionando sensores y actuadores de bajo costo, pero eficientes
- Investigar sobre la tecnología dsPIC, su arquitectura, sus aplicaciones, su entorno de desarrollo y sus compiladores.
- Diseñar todo los circuitos electrónicos que necesita el sistema para su eficiente interacción.

## **1.5. Alcances y Limitaciones**

Esta tesis busca crear una herramienta útil y novedosa para la creación de una pequeña empresa en el rubro de la venta de jugos naturales, dicha herramienta es una innovación en este rubro por tal motivo tiene un valor agregado que puede ser muy bien aprovechado.

### **1.5.1. Alcances del Sistema**

Para el establecimiento de los alcances se tomó como base el cumplimiento del objetivo general y los objetivos específicos.

### **1.5.2. Limitaciones del Sistema**

- En los recipientes de las frutas se ingresarán las frutas peladas y picadas de un tamaño relativamente pequeño para que el mecanismo de corte pueda separar volúmenes iguales de fruta.
- El sistema está diseñado para que trabaje por un máximo de 10 horas por día (ejemplo de 7 am a 5 pm), quiere decir que la cantidad de fruta que se debe de pelar y picar, debe ser solo para que trabaje por ese tiempo.
- El sistema está diseñado para que trabaje en la ciudad de Huaraz donde la temperatura varia de 7 °C a 19°C, con una temperatura promedio de 13°C (dentro de un local) y una humedad relativa promedio de 85%. No solo Huaraz tiene estas características climatológicas en todo el Perú.

En la Tabla 1.1 se muestra algunas temperaturas y humedades recomendadas para el almacenamiento y conservación de las frutas,

para nuestro caso solamente nos interesa las temperaturas de conservación y las vidas de almacenamiento de la papaya, la piña, la manzana, el plátano y el melón, no hay una información exacta de duración de conservación cuando las frutas están peladas y picadas a la temperatura de 13 °C, pero cabe resaltar que por pruebas experimentales en la ciudad de Huaraz a esta temperatura las frutas con las que vamos a trabajar peladas y picadas pero bien protegidas se conservan hasta por 12 horas, siendo el plátano el único que se conserva menos, en promedio 10 horas, se ennegrece pero no pierde su calidad nutritiva.

Tabla 1.1 Guía de temperaturas y humedades recomendadas para el almacenamiento de algunas frutas.

| Producto                   | Temperatura (°C) | Humedad Relativa (%) | Vida Aproximada de almacenamiento |
|----------------------------|------------------|----------------------|-----------------------------------|
| Guayaba                    | 8 a 10           | 90                   | 2 a 3 meses                       |
| Lima                       | 8,5 a 10         | 85 - 90              | 1 a 4 meses                       |
| Limón verde en general     | 10 a 14          | 85 - 90              | 2 a 3 semanas                     |
| Limón coloreado en general | 0 a 4,5          | 85 - 90              | 2 a 6 meses                       |
| Limón verde Europeo        | 11 a 14          | 85 - 90              | 1 a 4 meses                       |
| Limón Europeo amarillo     | 0 a 10           | 85 - 90              | 3 a 6 semanas                     |
| Limón Mexicano             | 8 a 10           | 85 - 90              | 3 a 8 semanas                     |
| Mango                      | 7 a 12           | 90                   | 3 a 6 semanas                     |
| Mandarina                  | 4                | 90 - 95              | 2 a 4 semanas                     |
| Melón                      | 7 a 10           | 85 - 90              | 3 a 7 semanas                     |
| Naranja                    | 3 a 9            | 85 - 90              | 3 a 12 semanas                    |
| Aguacate                   | 7 a 12           | 85 - 90              | 1 a 2 semanas                     |
| Papaya                     | 7 a 13           | 85 - 90              | 1 a 3 semanas                     |
| Piña verde                 | 10 a 13          | 85 - 90              | 2 a 4 semanas                     |
| Piña madura                | 7 a 8            | 85 - 90              | 2 a 4 semanas                     |
| Plátano coloreado          | 13 a 16          | 85 - 90              | 20 días                           |
| Plátano verde              | 12 a 13          | 85 - 90              | 1 a 4 semanas                     |
| Sandía                     | 5 a 10           | 85 - 90              | 2 a 3 semanas                     |
| Toronja                    | 10 a 15          | 85 - 90              | 6 a 8 semanas                     |
| Uva                        | -1 a 0           | 90 - 95              | 1 a 4 meses                       |

- El sistema está realizado para que trabaje en un local donde no solo se venda el jugo sino también algunas otras cosas como pasteles, kekes, sándwiches, etc. Esto también quiere decir que podrá tener una supervisión de un personal cerca para poder realizar la realimentación de las frutas en caso se acaben.

## **CAPÍTULO 2**

### **DISEÑO MECÁNICO DEL SISTEMA**

#### **2.1. Descripción del Sistema Mecánico**

El sistema mecánico diseñado se muestra en la Figura 2.1, consta de un selector de monedas para inicio de operación del sistema, un sistema selector de frutas con un mecanismo de corte de frutas tipo biela-manivela, una licuadora con giro de 90 grados para el vaciado de jugo y un mecanismo de separación y posicionado de vasos de 32 onzas para su respectivo servido del jugo.

Para realizar todo el diseño mecánico partimos desde el tipo y la cantidad de jugo que el sistema tiene que hacer, según estudio de mercado el jugo que se consume más en el mercado es el surtido, es un jugo preparado con distintos tipos de frutas, en nuestro caso las frutas a colocar son papaya, piña, manzana, plátano y melón. Vamos a tomar como referencia para nuestro diseño el jugo surtido en una cantidad máxima de 32 onzas (946ml).

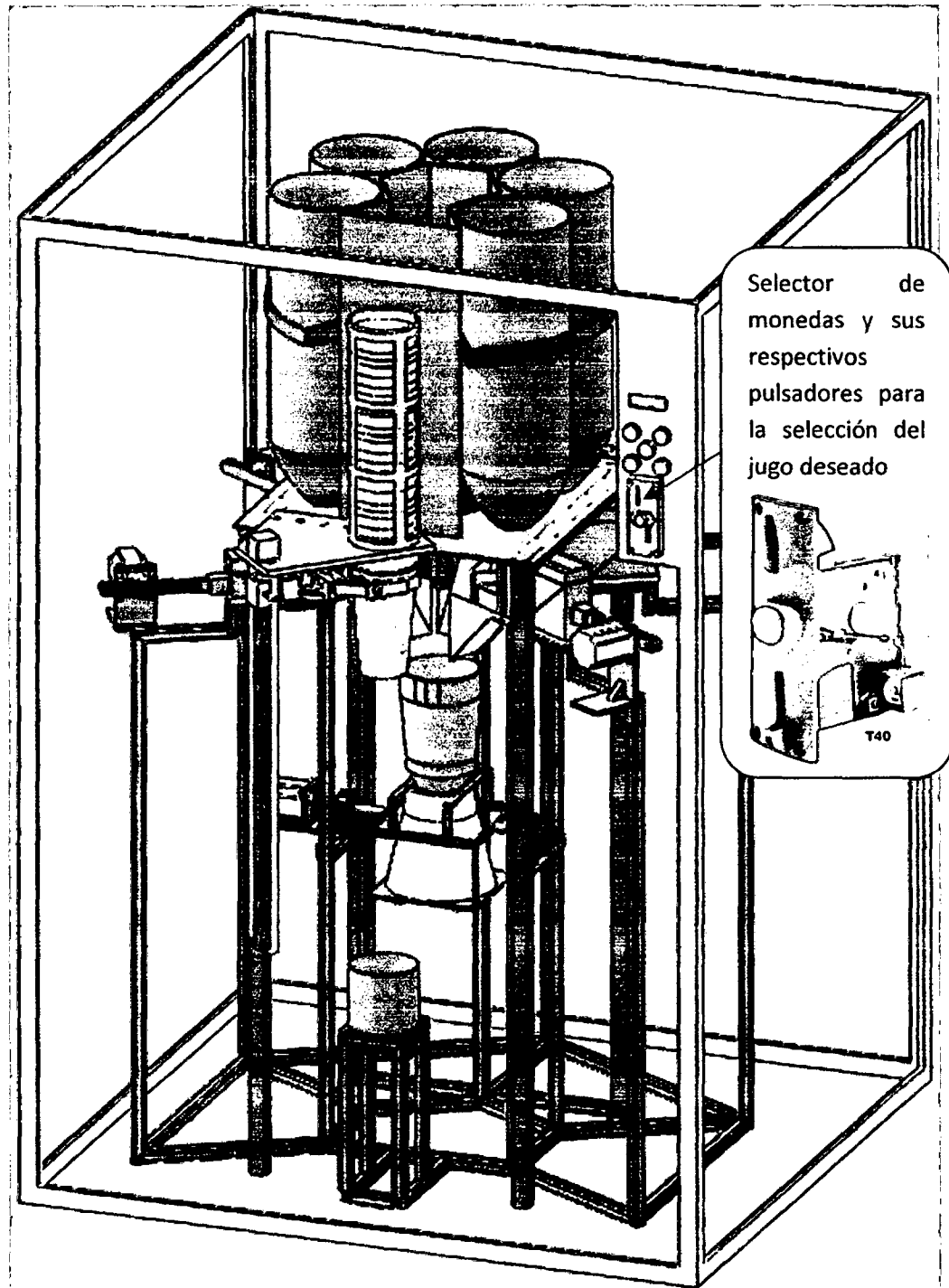


Figura 2.1 Sistema mecánico completo



Ahora para preparar este jugo se tiene una receta para hacer un exquisito jugo surtido, según esto diseñamos un sistema que va a separar la cantidad de cada fruta que indica poner en la receta, para hacer más sencillo el diseño se eligió separar las frutas por volumen, para esto se diseño un **sistema selector de frutas**, que se muestra en la Figura 2.2, el cálculo experimental para el volumen de los recipientes de las frutas y el volumen de separación de la fruta por cada preparado de jugo se verá en la sección 2.2.1. También en este sistema selector de frutas tenemos un mecanismo tipo biela – manivela para hacer el corte del volumen de fruta en cada preparado de jugo para esto también se hizo un cálculo estático - dinámico para la selección de motores paso a paso y esto se verá en la sección 2.2.2

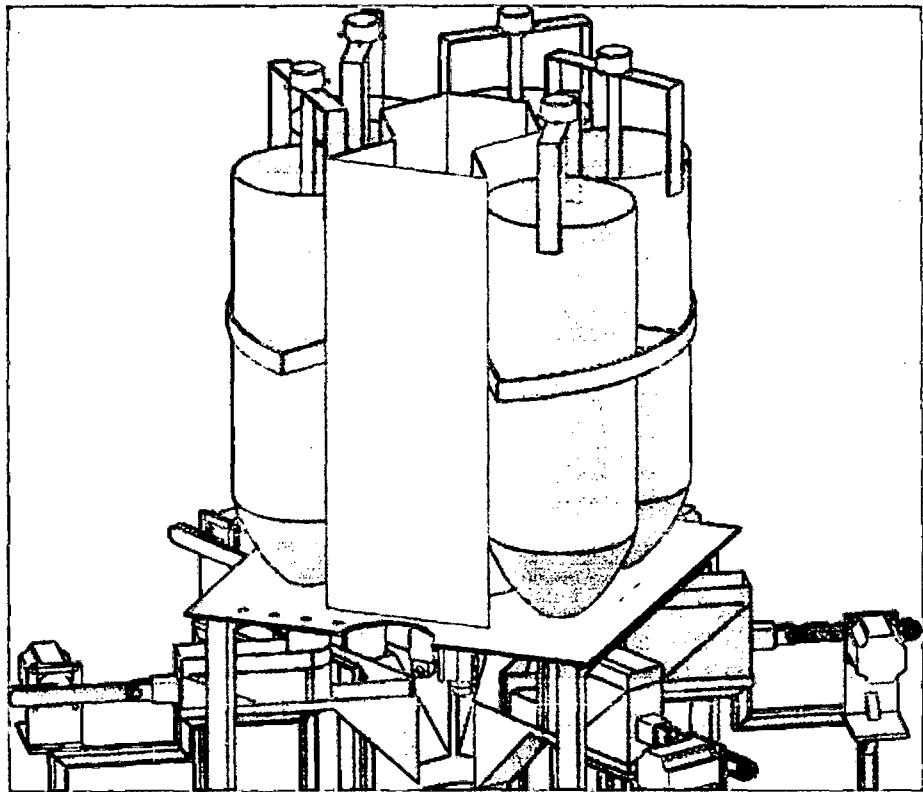


Figura 2.2 Sistema selector de frutas

Después de diseñar el sistema selector de frutas seleccionamos la licuadora y diseñamos el soporte de giro para la licuadora, esto para el preparado y vaciado del jugo licuado a un vaso de 32 onzas, como se muestra en la Figura 2.3, para diseñar el soporte de giro de la licuadora se hizo un estudio estático - dinámico para la selección del motor paso a paso y esto se verá en la sección 2.2.3.

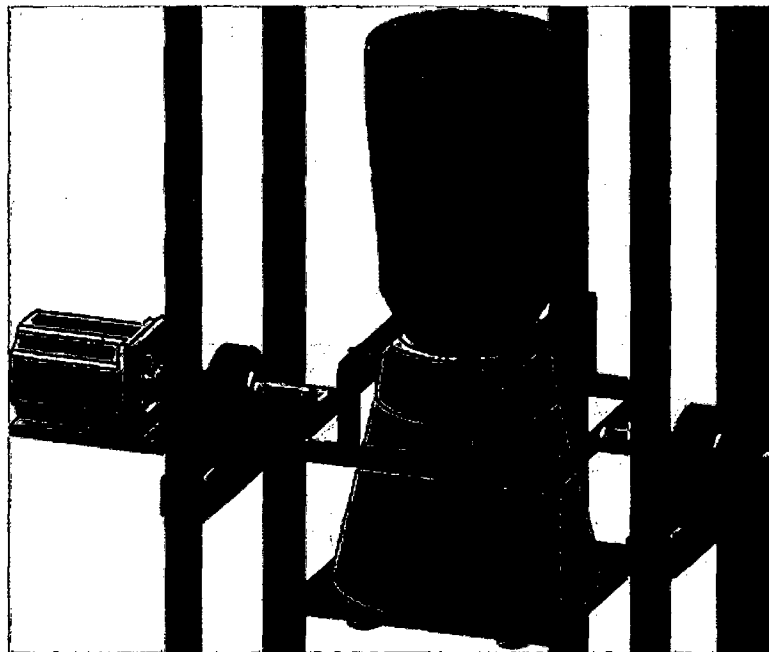


Figura 2.3 Licuadora con soporte de giro

Ahora para la recepción del jugo se diseñó un mecanismo de separación y posicionado de vasos, este mecanismo es un brazo de 2 grados de libertad para separar los vasos de un conjunto de vasos unidos concéntricamente a presión y a la vez lleve el vaso a la posición donde caerá el jugo, este brazo se muestra en la Figura 2.4.

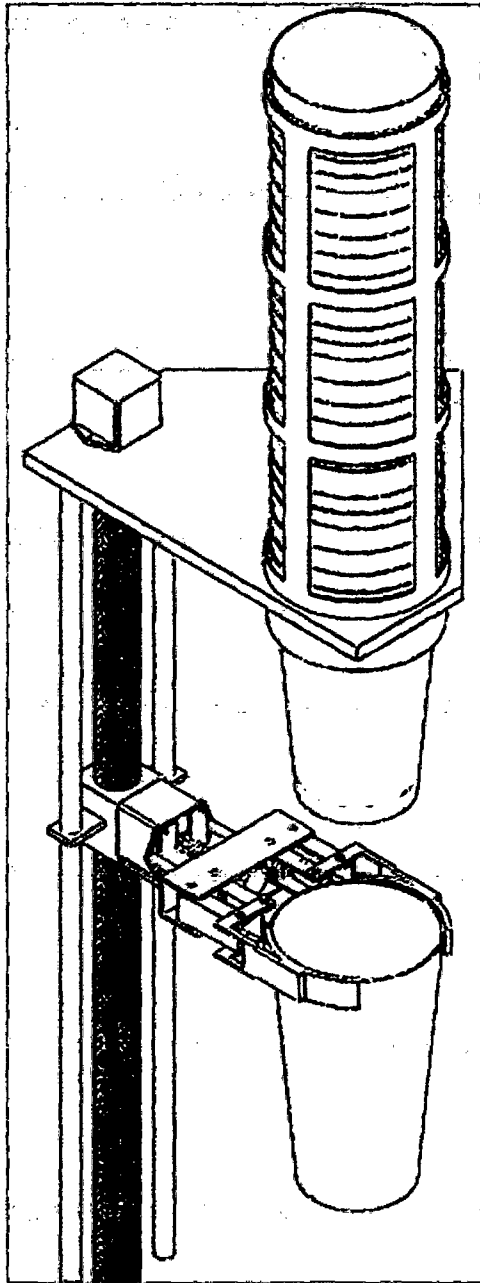


Figura 2.4 Mecanismo de separación y posicionado de vasos

## **2.2 Cálculos para el Diseño del Sistema Mecánico**

### **2.2.1 Pruebas Experimentales del Sistema Mecánico**

Cabe acotar que para hacer los cálculos para el diseño nos ayudamos de las herramientas que tiene el SolidWorks como las propiedades físicas de cada elemento: el volumen, la masa, los centros de gravedad, los momentos de inercia, etc.

Como se había mencionado en la sección 2.1 para diseñar la parte mecánica empezamos desde la cantidad de cada fruta a llenar a la licuadora, para preparar 32 onzas de jugo surtido, por conocimiento previo del preparado del jugo surtido (receta) y pruebas experimentales llegamos a encontrar un recipiente experimental que es un cilindro hueco en ambos lados superior e inferior con diámetro de la base de 0.05m y una altura de 0.1m, según este recipiente experimental una porción de papaya mas una porción de piña mas una porción de manzana mas una porción de plátano mas media porción de melón y 0.5 litros de líquido hacen un exquisito jugo surtido, solo media porción de melón porque esta fruta es, en el jugo surtido, como el ajos en las comidas, este recipiente se muestra en la Figura 2.5.

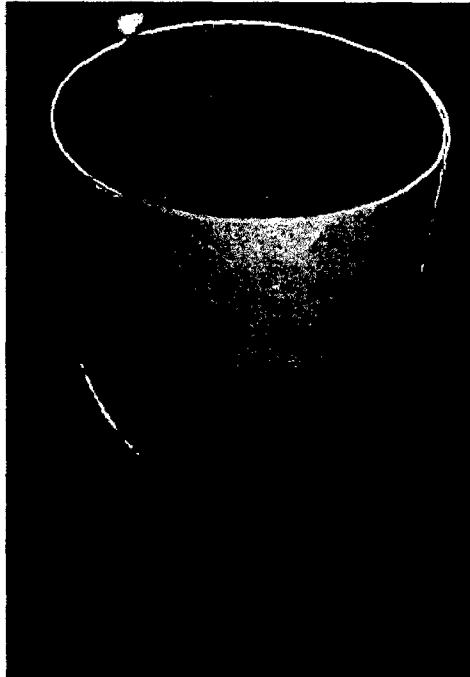


Figura 2.5 Recipiente experimental

El volumen del recipiente experimental es:

$$V_r = \frac{\pi * (0.05)^2 * 0.1}{4} m^3 = 0.00019635 m^3$$

Entonces se requiere diseñar un sistema para que prepare mínimo 50 vasos de 32 onzas de jugo surtido, por tal motivo cada recipiente que contenga las frutas debe tener un volumen mayor a 50 veces  $V_r$ , entonces el volumen de cada recipiente de fruta debe de ser mayor a  $0.0098175m^3$ , esto para tener en cuenta en el diseño de los recipientes de las frutas y la estructura del sistema, ahora veamos las siguientes pruebas experimentales que se realizaron con este recipiente experimental.

- Un volumen de manzana pelado y picado en el recipiente experimental pesa  $0.06Kg$ , según esto la densidad de la manzana pelado y picado es:

$$\rho_{manzana} = 305,5768 \frac{Kg}{m^3}$$

También se hicieron los siguientes cálculos:

5 porciones de este volumen de manzana pelado y picado mas 0.5 litros de líquido en 40 segundos de licuado hacen un delicioso jugo de manzana de 32 onzas, esto para nuestro cálculo en el preparado de los jugos de una sola fruta.

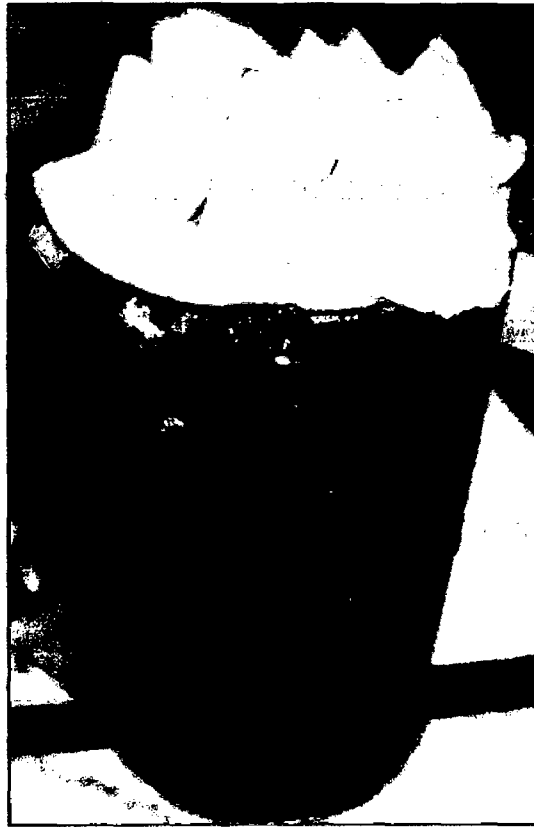


Figura 2.6 Porción de manzana pelado y picado

- Un volumen de papaya pelado y picado en el recipiente experimental pesa  $0.11Kg$ , según esto la densidad de la papaya pelado y picado es:

$$\rho_{papaya} = 560.2241 \frac{Kg}{m^3}$$

También se hicieron los siguientes cálculos:

5 porciones de este volumen de papaya pelado y picado mas 0.4 litros de líquido en 40 segundos de licuado hacen un delicioso jugo de papaya de 32 onzas, esto para nuestro cálculo en el preparado de los jugos de una sola fruta.

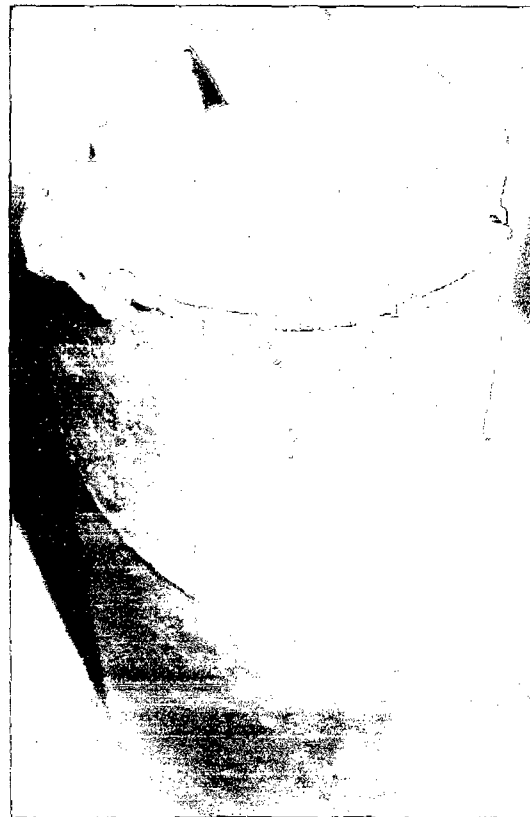


Figura 2.7 Porción de papaya pelado y picado

- Un volumen de piña pelado y picado en el recipiente experimental pesa  $0.1Kg$ , según esto la densidad de la piña pelado y picado es:

$$\rho_{\text{piña}} = 509,2946 \frac{Kg}{m^3}$$

También se hicieron los siguientes cálculos:

5 porciones de este volumen de piña pelado y picado mas 0.4 litros de líquido en 1 minuto de licuado hacen un delicioso jugo de piña de 32 onzas, esto para nuestro cálculo en el preparado de los jugos de una sola fruta.

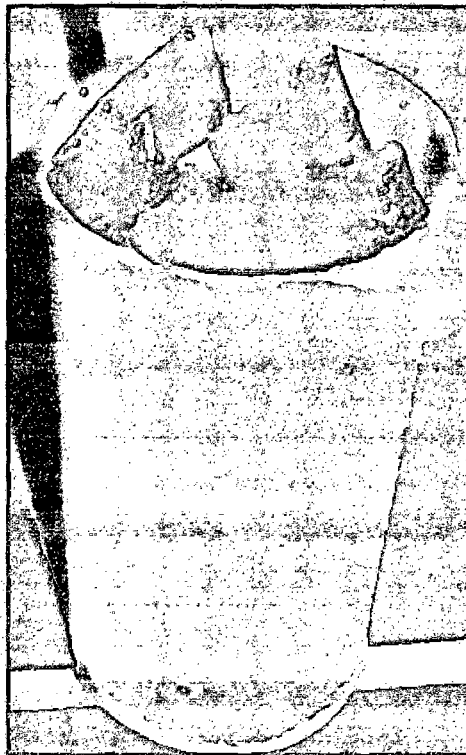


Figura 2.8 Porción de piña pelado y picado



- Un volumen de plátano pelado y picado en el recipiente experimental pesa 0.1Kg, según esto la densidad del plátano pelado y picado es:

$$\rho_{\text{plátano}} = 509.2946 \frac{\text{Kg}}{\text{m}^3}$$

También se hicieron los siguientes cálculos:

4 porciones de este volumen de plátano pelado y picado mas 0.5 litros de líquido en 40 segundos de licuado hacen un delicioso jugo de plátano de 32 onzas, esto para nuestro cálculo en el preparado de los jugos de una sola fruta.



Figura 2.9 Porción de plátano pelado y picado

- Medio volumen de melón pelado y picado en el recipiente experimental pesa  $0.06Kg$ , según esto la densidad del melón pelado y picado es:

$$\rho_{melon} = 611.1535 \frac{Kg}{m^3}$$

No se pretende hacer un jugo de melón solo, esta fruta sólo entra en el jugo surtido como un condimento para el jugo.



Figura 2.10 Media porción de melón pelado y picado

- Ahora con 1 porción de papaya mas una porción de piña mas una porción de manzana mas una porción de plátano mas media porción de melón, mas 500 ml de líquido en 40seg de licuado hacen un delicioso jugo de surtido de 32 onzas.



Figura 2.11 Jugo surtido en pleno licuado

### 2.2.2 Selección del Motor Paso a Paso para el Mecanismo de Corte de las Frutas (Mecanismo Biela – Manivela)

Se tiene el siguiente mecanismo para hacer el corte de las frutas el cual se muestra en la Figura 2.12, este mecanismo es similar a un mecanismo de un motor (biela – manivela) y la acción de corte es realizado por un motor paso a paso, para hacer una buena selección del motor paso a paso se tiene que hacer un estudio de los esfuerzos estáticos y dinámicos en el mecanismo para hallar el par motor efectivo del motor a seleccionar.

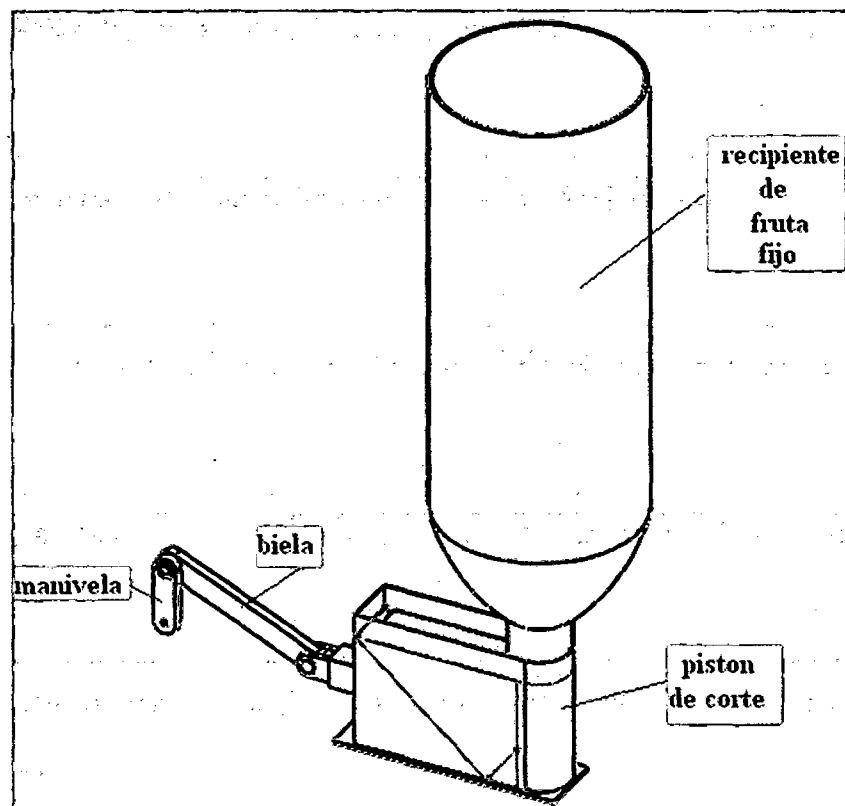


Figura 2.12 Mecanismo de corte de las frutas

Primero tenemos que buscar las fuerzas que actúan en el mecanismo de corte, los cuales se muestran en la Figura 2.13.

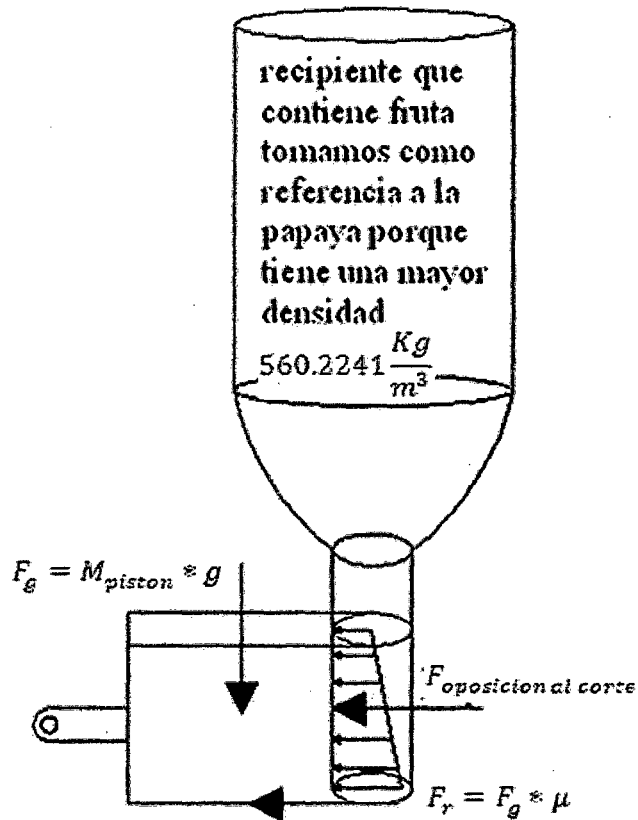


Figura 2.13 Fuerzas que actúan en el pistón de corte

Tenemos las dos fuerzas de oposición al corte de las frutas y ellas son: la fuerza de rozamiento del pistón de corte,  $F_r = F_g * \mu$  y  $F_{oposicion\ al\ corte}$ , según el diseño del pistón de corte en el SolidWorks es de acero inoxidable AISI 316 y tiene una masa de  $M_{piston} = 0.784Kg$ ,  $g = 9.81\ m/s^2$  y  $\mu_{acero/acero} = 0.6$ , según estos datos:

$$F_r = 4.62N.$$

Para calcular la  $F_{\text{oposicion al corte}}$  tomamos en consideración a la fruta pelado y picado dentro del recipiente como si fuera un fluido con alta viscosidad, esto debido a que la densidad de la fruta pelado y picado es menor que la del agua, además tomamos como referencia a la papaya pelado y picado porque es la fruta de mayor densidad según los cálculos experimentales de la sección 2.2.1.

Según esta consideración  $F_{\text{oposicion al corte}}$  es una resultante de fuerzas horizontales causada por la presión sobre las superficies que encierran al fluido, la resultante de ese sistema de fuerzas paralelas es en general una fuerza paralela aplicada en un punto arbitrario, más el torque de todas esas fuerzas distribuidas respecto a ese mismo punto. Es conveniente calcular la resultante de esas fuerzas en un cierto punto, llamado centro de presión, respecto al cual el torque de las fuerzas distribuidas es nulo, la fuerza debido a la presión atmosférica actúa por ambos lados, entonces la omitiremos del análisis por no contribuir en forma neta a la fuerza horizontal actuando sobre la superficie, tenemos la siguiente situación a calcular en la Figura 2.14 donde las medidas están en milímetros.

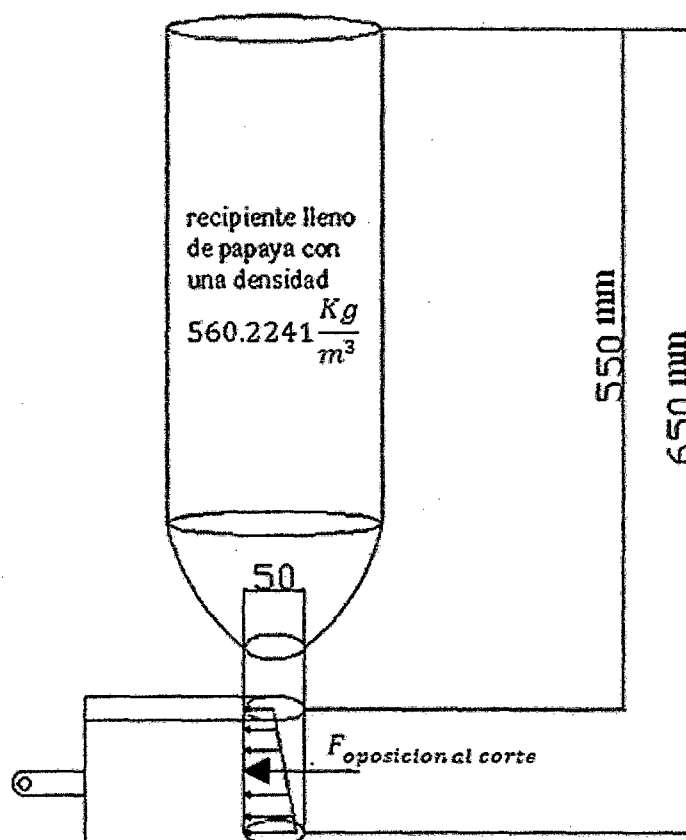


Figura 2.14 Situación a resolver para hallar la  $F_{oposicion\ al\ corte}$

De la figura anterior tenemos los siguientes datos:

$$h_1 = 0.55m$$

$$h_2 = 0.65m$$

$$\rho_{papaya\ pelado\ y\ picado} = 560.22Kg/m^3$$

$$Radio = 0.025m$$

$$g = 9.81\ m/s^2$$

Tomamos como si fueran placas rectangulares diferenciales que van de 0 grados a 180 grados porque solo requiero las fuerzas que se oponen al corte, como se muestra en la Figura 2.15.

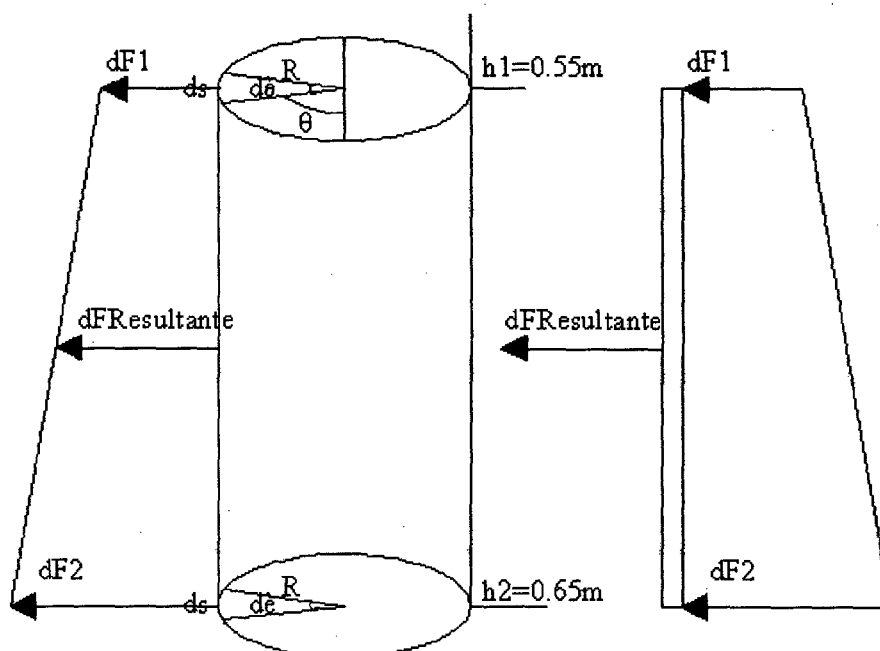


Figura 2.15 Evaluación de la derivada de la fuerza resultante

Sabemos por hidrostática que la fuerza resultante y el punto de aplicación corresponden al área y al centroide de la figura correspondiente a la fuerza distribuida entre  $h_1$  y  $h_2$ , como sabemos el área de nuestra aplicación es:

$$\frac{1}{2}(dF_1 + dF_2)(h_2 - h_1)$$

Y el centroide es:

$$\frac{1}{3} \frac{(dF_1 + 2dF_2)}{dF_1 + dF_2} (h_2 - h_1)$$

Donde  $dF_1$  y  $dF_2$  son las diferenciales de fuerza por unidad de diferencial de longitud en  $h_1$  y  $h_2$

$$dF_1 = \rho * g * h_1 * ds$$

$$dF_2 = \rho * g * h_2 * ds$$

Así entonces el diferencial de fuerza resultante es:

$$dF_{Resultante} = \frac{1}{2} \rho g (h_1 + h_2) (h_2 - h_1) ds$$

Además  $ds = R d\theta$ , según esto:

$$dF_{Resultante} = \frac{1}{2} \rho g (h_1 + h_2) (h_2 - h_1) R d\theta$$

Y su punto de aplicación será:

$$h_p = h_1 + \frac{1}{3} \frac{(h_1 + 2h_2)}{h_1 + h_2} (h_2 - h_1) = 0.6014m$$

Como se muestra en la Figura 2.16 la  $dF_{Resultante}$  va en dirección de  $\theta=0^\circ$  hasta  $\theta=180^\circ$ , pero a nosotros solo nos interesa la fuerza de oposición al corte que es:

$$dF_{oposicion\ al\ corte} = dF_{Resultante} \sin\theta \Big|_{\theta=0}^{\theta=180}$$



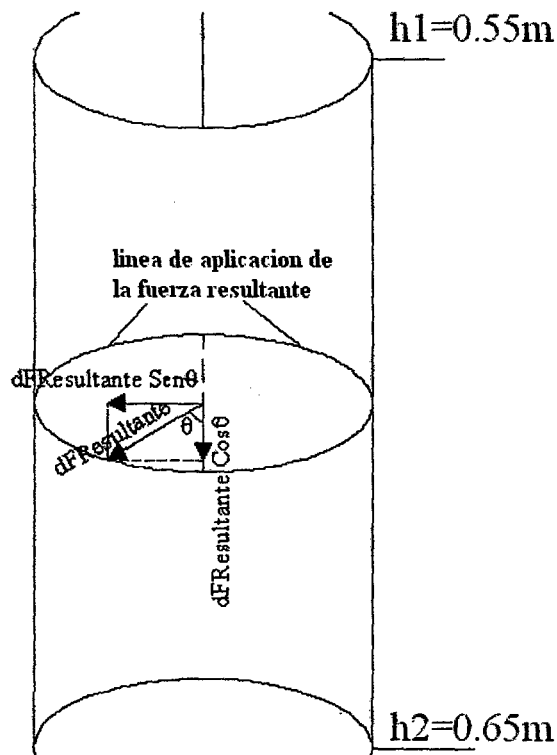


Figura 2.16 Descomposición de la derivada de la fuerza resultante

$$dF_{\text{oposicion al corte}} = \frac{1}{2} \rho g (h_1 + h_2) (h_2 - h_1) R \sin \theta d\theta$$

Integrando este diferencial de  $\theta=0^\circ$  hasta  $\theta=180^\circ$

$$\int dF_{\text{oposicion}} = \int_0^{180} \frac{1}{2} \rho g (h_1 + h_2) (h_2 - h_1) R \sin \theta d\theta$$

Nos da el siguiente resultado

$$F_{\text{oposicion al corte}} = \rho g (h_1 + h_2) (h_2 - h_1) R$$

Reemplazando los datos mostrados anteriormente

$$F_{\text{oposicion al corte}} \approx 16.5N$$

Ahora ya tenemos las dos fuerzas que se oponen al corte, pero no son fuerzas en un mismo punto de aplicación para poder sumarlos, entonces aplicamos la descomposición de una fuerza dada en una fuerza y un par para trasladar las fuerzas a un mismo punto de aplicación, claro que aparecerán pares de tales fuerzas que trasladamos, pero como el pistón de corte es forzado a que se traslade linealmente y además las distancias de traslación de las fuerzas son pequeñas, entonces despreciamos estos pares generados por la traslación de las fuerzas y nuestra fuerza total de oposición al corte serán la suma de la fuerza de rozamiento del pistón de corte mas  $F_{\text{oposición al corte}}$  de la fruta esto es:

$$F_{TOC} = F_{\text{total de oposición al corte}} = 4.62N + 16.5N$$

$$F_{TOC} \approx 22N$$

Ahora ya podemos hacer el análisis de los esfuerzos totales.

En el mecanismo de la Figura 2.17 se tiene:

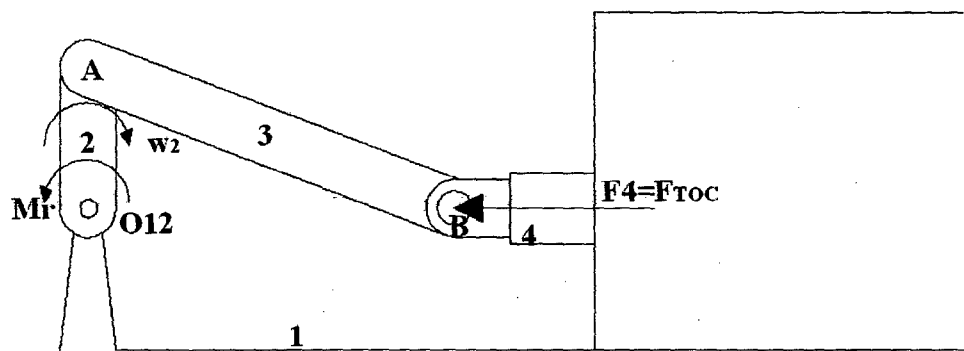


Figura 2.17 Mecanismo de corte de las frutas

Datos:

- $F_4 = F_{TOC} = 22N$ , debido a la oposición al corte de las frutas.
- $w_2$ , velocidad de giro de la manivela, como queremos que el motor haga 1 vuelta en 2 segundos, entonces  $w_2 = 30\text{rev}/\text{min}$ .
- masas y momentos de inercia de los elementos que interactúan en este mecanismo y para determinarlos nos ayudamos de una de las herramientas que tiene el SolidWorks que son las propiedades físicas de cada pieza diseñada, según esto:

$$m_2 = 0.0488Kg, I_{G2} = 0.00001865Kg * m^2$$

$$m_3 = 0.123Kg, I_{G3} = 0.000254Kg * m^2$$

$$m_4 = 0.7836Kg, I_{G4} = 0.00431Kg * m^2$$

Incógnitas:

- $Mr$  o par resistente (esfuerzo total de giro).
- Esfuerzos sobre todos los miembros.

Ahora efectuamos por separado el análisis de los esfuerzos estáticos y dinámicos, aplicando luego el principio de superposición.

## 1. Estudio de los esfuerzos estáticos

Aplicando las ecuaciones generales a cada uno de los miembros del mecanismo se tiene:

### 1.1 Miembro 4 (pistón)

Sobre el actúan las siguientes fuerzas:

- $F_4$  = oposición al corte.
- $F_{14}$  = reacción del bastidor 1 sobre el pistón 4.
- $F_{34}$  = reacción de la biela 3 sobre el pistón 4.

Para que estén en equilibrio se ha de cumplir:

$$\vec{F}_4 + \vec{F}_{14} + \vec{F}_{34} = 0 \dots (1)$$

$$M(\vec{F}_4) + M(\vec{F}_{14}) + M(\vec{F}_{34}) = 0 \dots (2)$$

La ecuación (2) se cumple evidentemente, por ser fuerzas concurrentes. De la ecuación (1) se conoce  $\vec{F}_4$  y la dirección de  $\vec{F}_{14}$  y  $\vec{F}_{34}$ , por lo que se pueden calcular gráficamente, como se ve en la Figura 2.18, ojo que tomamos el mecanismo en esa posición por el mismo hecho de que en esa posición el esfuerzo total de giro es mayor, para una mejor selección del motor paso a paso.

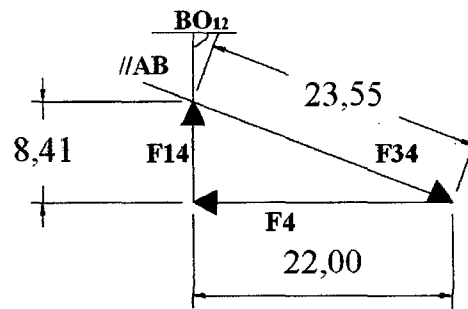


Figura 2.18 Fuerzas en el pistón

## 1.2 Miembro 3 (biela)

Sobre el actúan las fuerzas:

- $F_{43}$  = reacción del pistón.
- $F_{23}$  = reacción de la manivela.

Para que estén en equilibrio se ha de cumplir:

$$\vec{F}_{43} + \vec{F}_{23} = 0 \dots\dots(3)$$

$$M(\vec{F}_{43}) + M(\vec{F}_{23}) = 0 \dots\dots(4)$$

La ecuación (4) se cumple evidentemente, por estar ambas fuerzas sobre la misma línea de acción. De la ecuación (3) se obtiene que  $\vec{F}_{43}$  (igual y contraria  $\vec{F}_{34}$ ) es igual y contraria a  $\vec{F}_{23}$ , como se ve en la Figura 2.19.

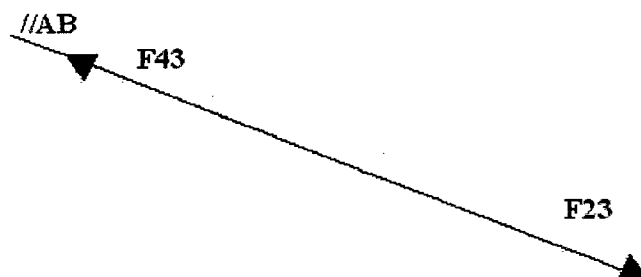


Figura 2.19 Fuerzas en el miembro 3 (biela)

### 1.3 Miembro 2 (manivela)

Sobre el actúan las fuerzas:

- $F_{32}$  = reacción de la biela.
- $F_{12}$  = reacción del bastidor.
- $Mr$  = par resistente aplicado.

Por estar en equilibrio se tendrá:

$$\vec{F}_{32} + \vec{F}_{12} = 0 \dots(5)$$

$$M(\vec{F}_{32}) + M(\vec{F}_{12}) + Mr = 0 \dots(6)$$

De la ecuación (5) se deduce que  $\vec{F}_{32}$  (igual y contraria a  $\vec{F}_{23}$ ) debe ser igual y contraria a  $\vec{F}_{12}$ . Con esto queda perfectamente determinada como se ve en la Figura 2.20

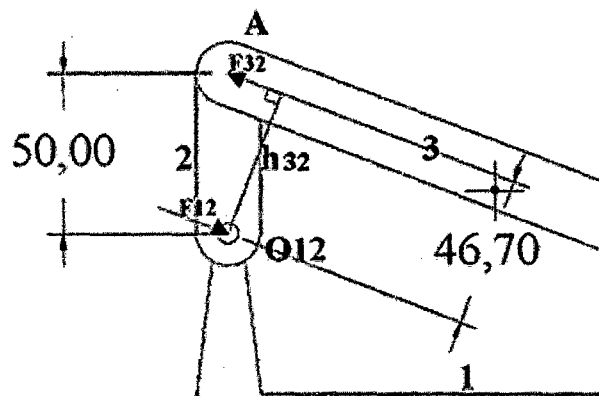


Figura 2.20 Fuerzas en el miembro 2 (manivela)

De la ecuación (6) se deduce que el par resistente que se puede vencer con la fuerza  $\vec{F}_4$  de oposición al corte, vale:

$$Mr = F_{32} * h_{32} = 23.55N * 0.0467m = 1.1N * m$$

#### 1.4 Miembro 1 (bastidor)

Las fuerzas actuantes son:

- $F_4$  = oposición al corte.
- $F_{21}$  = reacción de la manivela 2 sobre el bastidor 1.
- $F_{41}$  = reacción del pistón 4 sobre el bastidor 1.

Para que estuviera en equilibrio se habría de cumplir

$$\vec{F}_4 + \vec{F}_{21} + \vec{F}_{41} = 0 \dots (7)$$

$$M(\vec{F}_4) + M(\vec{F}_{21}) + M(\vec{F}_{41}) = 0 \dots (8)$$

La ecuación (7) se cumple evidentemente, como se ve en la Figura 2.21.

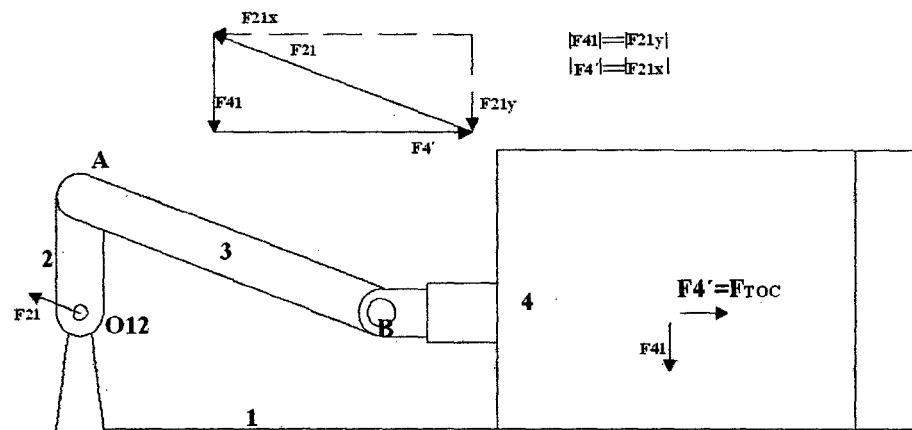


Figura 2.21 Fuerzas en el bastidor o corredera del pistón

Sin embargo, la ecuación (8) no se cumple. En efecto, de la descomposición efectuada en la figura 2.18 se observa que  $\sum \vec{F} = 0$  y que las fuerzas  $\vec{F}_{41}$  y  $\vec{F}_{21}^y$  forman un par que tiende a hacer girar todo el bastidor alrededor del eje del cigüeñal O12. (Su valor es exactamente igual al par resistente  $Mr$ , puesto que en definitiva no es más que su consecuencia.) Por consiguiente, el bastidor de un motor no se encuentra en equilibrio: está sometido a un par, denominado par de vuelco.

## 2. Estudio de los esfuerzos dinámicos

En el mecanismo de la Figura 2.22 se tiene:

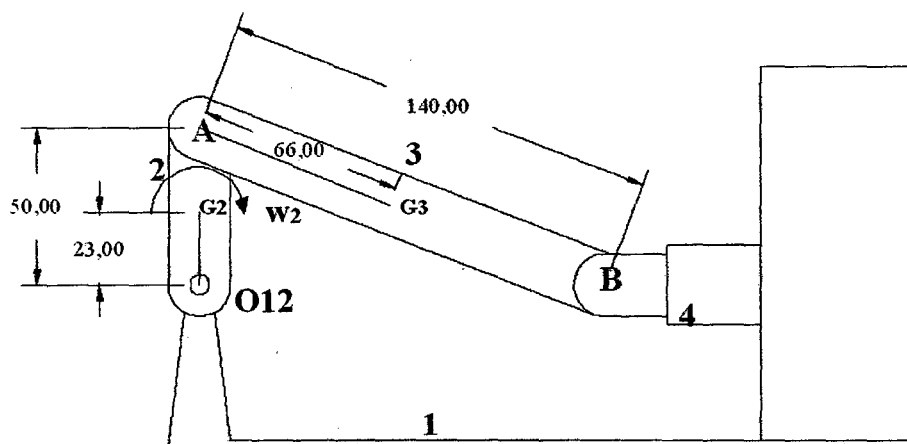


Figura 2.22 Ubicación centros de masa de los miembros del mecanismo

2.1 conocido  $w_2$  se hallan los polígonos de velocidades y aceleraciones, calculando  $\vec{a}_{G2}$ ,  $\vec{a}_{G3}$  y  $\vec{a}_{G4} = \vec{a}_4$ , como se ve en la Figura 2.23.



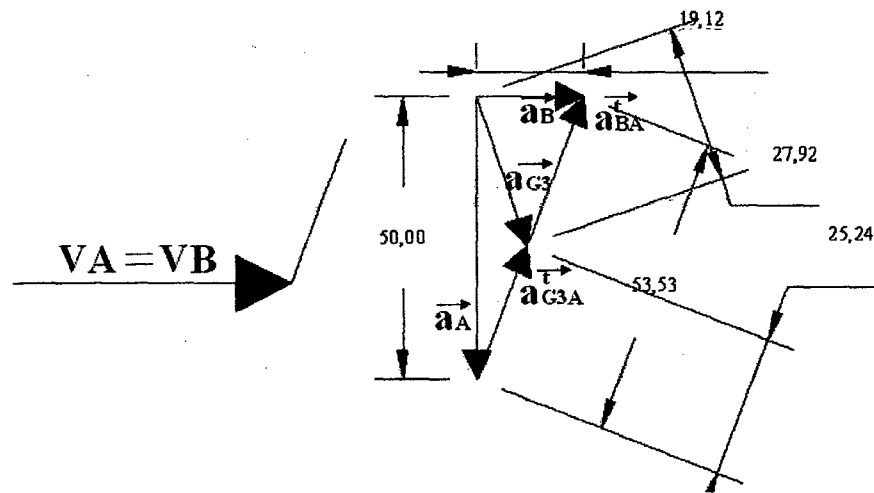


Figura 2.23 Polígonos de velocidades y aceleraciones

$$V_A = w_2 * \overline{O_{12}A} = 30RPM * \frac{2\pi}{60s} * 0.05m = 0.157 \frac{m}{s}$$

$$V_{G2} = w_2 * \overline{O_{12}G_2} = 30RPM * \frac{2\pi}{60s} * 0.023m = 0.072 \frac{m}{s}$$

Se conoce la dirección y sentido de las velocidades de los puntos A y B en ese instante y tienen la misma dirección y sentido y como  $\vec{V}_B = \vec{V}_A + \vec{V}_{BA}$ , entonces:

$$\underbrace{\vec{V}_B = \vec{V}_A}_{\text{misma dirección}} + \underbrace{\vec{V}_{BA}}_{\text{perpendicular a AB}} \rightarrow \vec{V}_{BA} = 0$$

$$\therefore \vec{V}_B = \vec{V}_A \dots\dots\dots (9)$$

$$\text{Si } \vec{V}_{BA} = 0 \rightarrow w_{AB} = w_3 = 0 \text{ y } \overline{a_{AB}^n} = 0$$

Además  $w_2 = cte$ , entonces  $\alpha_2 = 0$ , según esto:

$$\overline{a_A} = \overline{a_A^n} = \frac{|\overline{v_A}|^2}{AO_{12}} = 0.493 \frac{m}{s^2} \dots\dots\dots (10)$$

$$\vec{a}_{G2} = \vec{a}_{G2}^n = \frac{|v_{G2}|^2}{G2O12} = 0.225 \frac{m}{s^2} \dots\dots\dots (11)$$

Sabemos que  $\vec{a}_B = \vec{a}_A + \underbrace{\vec{a}_{BA}^n}_0 + \vec{a}_{BA}^t$ , conocemos el modulo, la direcci3n y sentido de  $\vec{a}_A$ , conocemos la direcci3n y sentido de  $\vec{a}_B$ ,  $\vec{a}_{BA}^t$  es perpendicular a la l3nea  $\overline{AB}$ , seleccionamos una escala  $k_a$  para dibujar los vectores y con los dem3s datos podemos hallar el modulo de  $\vec{a}_B$  y  $\vec{a}_{BA}^t$  en el pol3gono de aceleraciones.

Vemos que:

$$k_a = \frac{0.493 \frac{m}{s^2}}{50 \text{ mm}} \dots\dots\dots (12)$$

Seg3n la Figura 2.20  $|\vec{a}_{BA}^t| = 53.53mm$ , lo multiplicamos por  $k_a$  y nos sale  $|\vec{a}_{BA}^t| = 0.528m/s^2$ , con esto ya podemos hallar  $\alpha_{BA} = \alpha_3$  con la siguiente ecuaci3n:  $|\vec{a}_{BA}^t| = \alpha_{BA} * \overline{AB}$ , entonces:

$$\alpha_{BA} = \alpha_3 = 3.77rad/s^2 \dots\dots\dots (13)$$

Ahora hallamos  $\vec{a}_{G3}$ , con la ecuaci3n:

$$\vec{a}_{G3} = \vec{a}_A + \underbrace{\vec{a}_{G3A}^n}_0 + \vec{a}_{G3A}^t \dots\dots\dots (14)$$

Tenemos modulo, direcci3n y sentido de  $\vec{a}_A$ , direcci3n de  $\vec{a}_{G3A}^t$  y el modulo los hallamos con la siguiente ecuaci3n  $|\vec{a}_{G3A}^t| = \alpha_{BA} * \overline{G3A} = 0.249m/s^2$ , en el Figura 2.20  $|\vec{a}_{G3A}^t| = 25.235mm$  y con estos datos podemos hallar  $\vec{a}_{G3}$  y  $\vec{a}_{G4} = \vec{a}_4$  en el pol3gono de aceleraciones  $|\vec{a}_{G3}| = 0.275m/s^2$  y  $|\vec{a}_4| = 0.188m/s^2$

2.2 Conocido las aceleraciones se calculan las fuerzas y los momentos de inercia que obran sobre cada miembro:

$$\vec{J}_2 = -m_2 * \vec{a}_{G2}, \vec{J}_3 = -m_3 * \vec{a}_{G3}, \vec{J}_4 = -m_4 * \vec{a}_4 \dots\dots\dots (15)$$

$$M_{G2} = -\alpha_2 * I_{G2} = 0, M_{G3} = -\alpha_3 * I_{G3},$$

$$M_{G4} = -\alpha_4 * I_{G4} = 0 \dots\dots\dots (16)$$

$$|\vec{J}_2| = 0.011N, |\vec{J}_3| = 0.0338N, |\vec{J}_4| = 0.1473N$$

$$\text{sea } k_j = 0.0338N/50mm, |\vec{J}_3| = 50mm$$

2.3 Así, se ha obtenido el sistema de fuerzas que obra sobre el mecanismo Figura 2.24.

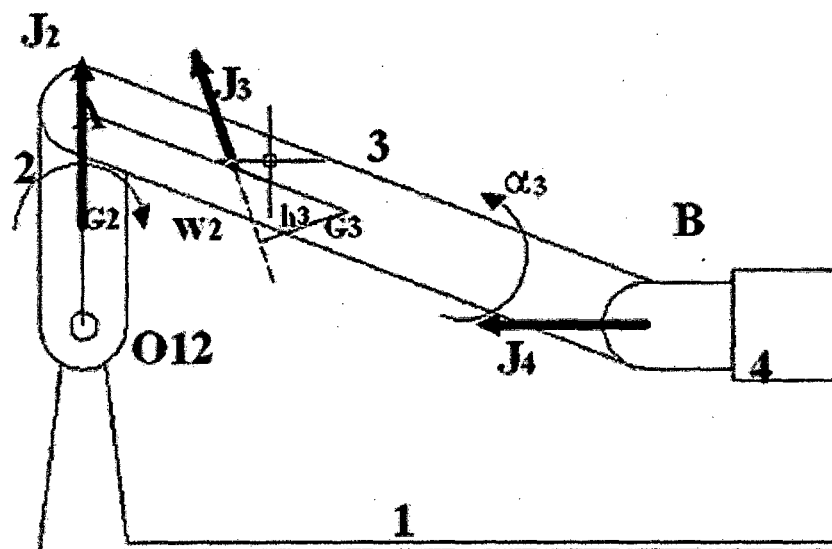


Figura 2.24 Sistema de fuerzas sobre el mecanismo

Puede observarse que  $J_3$  se ha desplazado " $h_3$ " de forma que  $M_{G3}$  se opone a  $\alpha_3$ .

$$h_3 = \frac{\alpha_3 I_{G3}}{|J_3|} = \frac{3.77 \times 0.00254}{0.0338} = 0.0283m \dots\dots\dots (17)$$

Para calcular las acciones sobre diferentes miembros se aplicaran las ecuaciones generales de la estática.

#### 2.4 Miembro 4 (pistón)

Las fuerzas que sobre él actúan son:  $\vec{J}_4, \vec{J}_{14}, \vec{J}_{34}$ . Por estar en equilibrio:

$$\vec{J}_4 + \vec{J}_{14} + \vec{J}_{34} = 0 \dots\dots\dots (18)$$

$$M(\vec{J}_4) + M(\vec{J}_{14}) + M(\vec{J}_{34}) = 0 \dots\dots\dots (19)$$

La ecuación (19) se cumple evidentemente, por ser fuerzas concurrentes.

La ecuación (18) solo se conoce  $\vec{J}_4$  la dirección de  $\vec{J}_{14}$ .

#### 2.5 Miembro 3 (biela)

Las fuerzas que sobre él actúan son:  $\vec{J}_3, \vec{J}_{23}, \vec{J}_{43}$ . Por estar en equilibrio:

$$\vec{J}_3 + \vec{J}_{23} + \vec{J}_{43} = 0 \dots\dots\dots (20)$$

$$M(\vec{J}_3) + M(\vec{J}_{23}) + M(\vec{J}_{43}) = 0 \dots\dots\dots (21)$$

De la ecuación (20) solo se conoce  $\vec{J}_3$ , de la ecuación (21) se puede hallar  $\vec{J}_{23}^n$ , como se muestra en la Figura 2.25, tomando momentos respecto del punto B. con ello, también se conoce la dirección de  $\vec{J}_{23}^t$ .

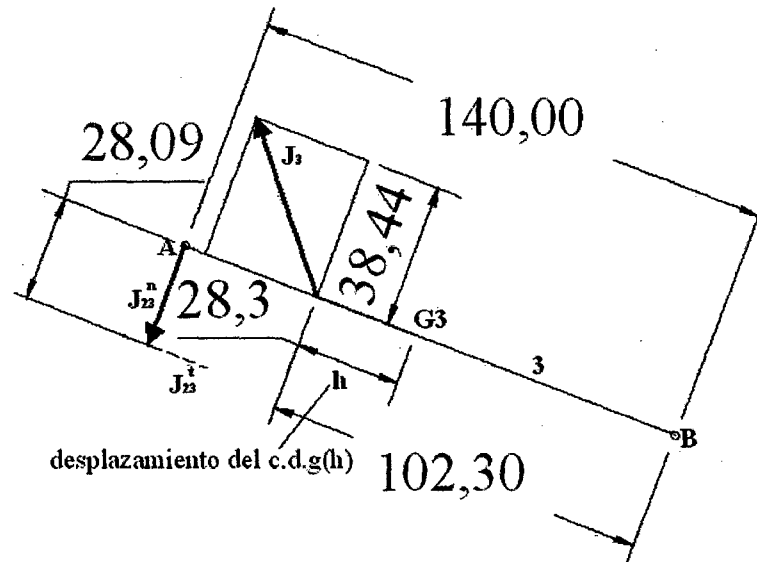


Figura 2.25 Determinación de las fuerzas que actúan en el miembro 3 (biela)

Como  $\vec{J}_{43} = -\vec{J}_{34}$ , se puede eliminar este término entre las ecuaciones (18) y (20).

$$\vec{J}_4 + \vec{J}_{14} = -\vec{J}_{34} \dots \dots (22)$$

$$-\vec{J}_{43} = \vec{J}_{34} \dots \dots (23)$$

Sumando (22) + (23):

$$\vec{J}_4 + \vec{J}_{14} + \vec{J}_3 + \vec{J}_{23}^n + \vec{J}_{23}^t = 0$$

Ecuación vectorial con dos incógnitas (los módulos de  $J_{14}$  y  $J_{23}^t$ ) que puede resolverse gráficamente, como se ve en la Figura 2.26, obteniéndose los valores de  $J_{34}$ ,  $J_{14}$  y  $J_{23}$ .

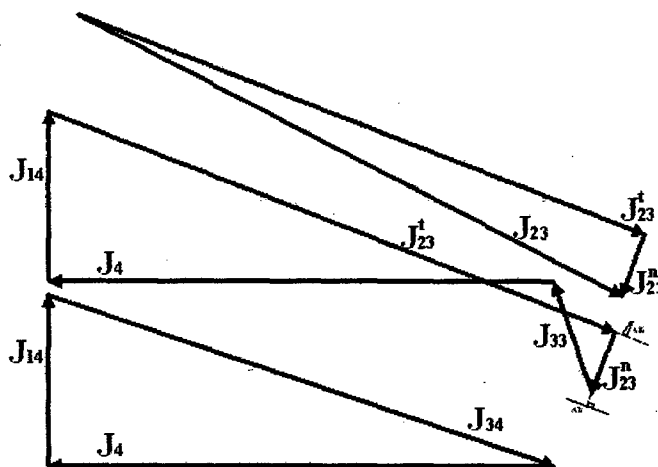


Figura 2.26 Fuerzas que actúan sobre el miembro 3 (biela)

## 2.6 Miembro 2 (manivela)

Las fuerzas que sobre él actúan son:  $\vec{J}_2, \vec{J}_{12}, \vec{J}_{32}$ . Por estar en equilibrio:

$$\vec{J}_2 + \vec{J}_{12} + \vec{J}_{32} = 0 \dots \dots \dots (24)$$

$$M(\vec{J}_2) + M(\vec{J}_{12}) + M(\vec{J}_{32}) = 0 \dots \dots \dots (25)$$

Como  $\vec{J}_2$  y  $\vec{J}_{32} = -\vec{J}_{23}$  son conocidas, la ecuación (24) permite hallar  $\vec{J}_{12}$ , como se ve en la Figura 2.27. Al examinar la ecuación (25) se observa que no se cumple, por no ser las tres fuerzas concurrentes.

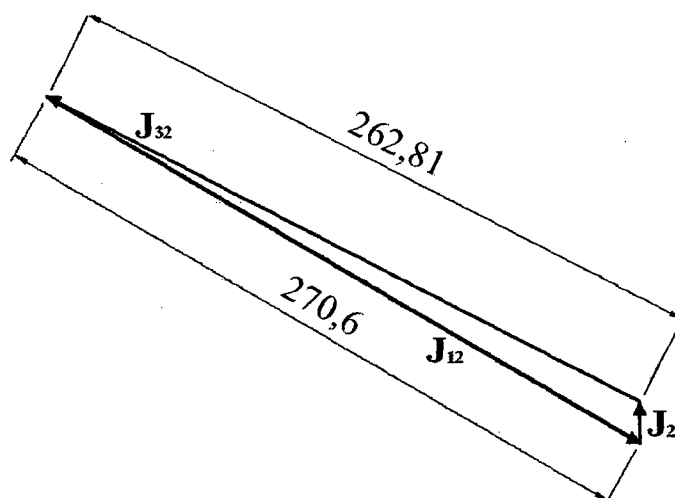


Figura 2.27 Fuerzas miembro 2 (manivela)

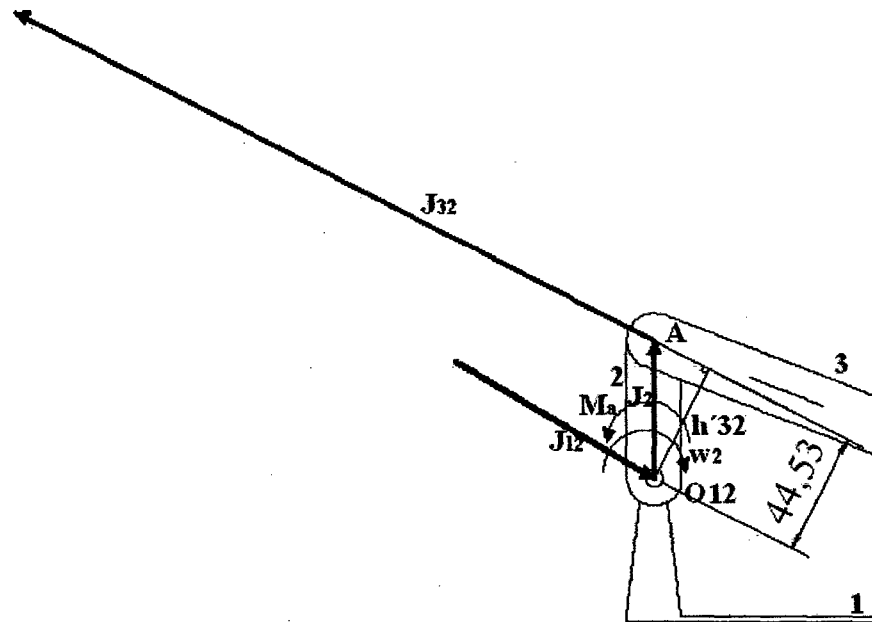


Figura 2.28 Par acelerador, par no equilibrado

Como se muestra en la Figura 2.28. Aparece por tanto un par no equilibrado cuyo valor puede hallarse fácilmente tomando momentos respecto de O12.

$$Ma = J_{32} * h'_{32} = 262.81mm * \frac{0.0338N}{50mm} * 0.04453m$$

$$Ma = 0.008N * m$$

Este par actúa sobre el miembro conducido hacia la izquierda en este caso (en otro instante podría hacerlo hacia la derecha), frenando el movimiento del motor. En consecuencia, el miembro 2 no está equilibrado, llamando a este par (par acelerador).

### 3. Estudio de los esfuerzos combinados

Aplicando el principio de superposición, el esfuerzo giratorio total será la suma del par estático y el par acelerador, se le denomina “par motor efectivo”. En el instante considerado el par acelerador tiene el mismo sentido al par motor y el par total o par motor efectivo será:

$$Me = F_{32} * h_{32} + J_{32} * h'_{32} = 1.108N * m$$

Se ve claramente que el efecto del “par acelerador” es el de modificar el valor del par motor (estático) haciendo que el “par motor total o efectivo” sea mayor o menor que él.

#### **Procedimiento de selección del motor paso a paso**

Ahora que ya conocemos el par motor efectivo que requiere cada motor del mecanismo de corte de las frutas para vencer las fuerzas de oposición y el torque dinámico, podemos seleccionar un motor paso a paso con un aceptable factor de seguridad, se recomienda aplicar al motor un factor de seguridad entre 20 a 100%, dependiendo del sistema en cuestión, para evitar que el motor deje de ejecutar pulsos o se pare por cambios de carga, ya que hay varios efectos difíciles de cuantificar, tal como la velocidad del lubricante, desgaste del actuador, etc.

Tenemos que nuestro par motor efectivo es 1.108N-m que es lo mismo a 157 oz-in, encontramos en el mercado motores paso a paso con torques cercanos y mayores a este valor, como uno de 175 oz-in y otro de 280 oz-in,



ahora veamos los factores de seguridad con los dos torques cercanos al torque requerido por nuestro motor, con el motor de torque 175 oz-in tenemos un factor de seguridad de 11% y con el motor de torque 280 oz-in tenemos un factor de seguridad de 78%, elegimos este motor de 280 oz-in por tener un factor de seguridad alto y pueda compensar en algunos efectos difíciles de cuantificar para nuestro torque requerido, además la velocidad con la que van a trabajar estos motores son de 30RPM, a esta velocidad no hay problema en la elección de motor, porque el torque a esta velocidad es alto como se muestra en la gráfica del torque VS velocidad en la Figura 2.29.

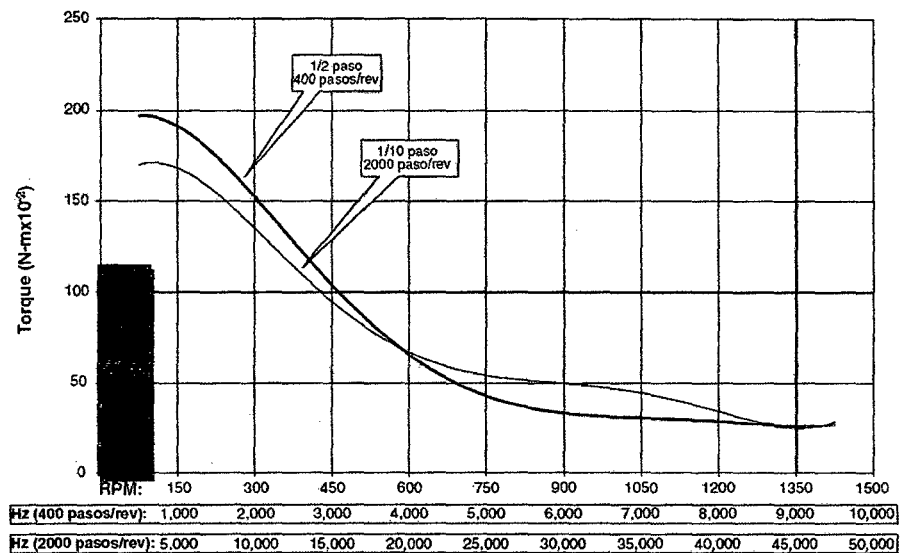


Figura 2.29 Grafica torque VS velocidad motor pap 280 oz-in

También podemos seleccionar el ángulo de paso del motor paso a paso en nuestro caso es suficiente que tenga un ángulo de paso de  $90^\circ$  pero no es muy comercial, encuentras con ese ángulo de paso, pero no con el torque requerido. El que sí es muy comercial y lo encuentras en toda cantidad de

torque es el de  $1.8^\circ$  de ángulo de paso, según este ángulo de paso, necesitaría 200 pasos para hacer un giro completo.

Se tiene seleccionado el motor paso a paso para el mecanismo de corte y este es un motor paso a paso de **280 oz-in (1.85 N-m)** de torque y  $1.8^\circ$  de ángulo de paso. En el ANEXO E se muestra el catálogo de este motor paso a paso seleccionado.

Ahora veamos algunos requerimientos para hacer funcionar a este motor paso a paso seleccionado.

¿Cuántos pulsos debe generar el controlador para hacer el movimiento hasta la posición deseada?

El número total de pulsos para hacer el movimiento es expresado por la siguiente fórmula:

$$P_{total} = \text{pulsos totales} = \left( \frac{D_{total}}{\frac{d_{carga}}{i}} \right) * \theta_{paso} \dots\dots (26)$$

Donde:

$D_{total}$ : Distancia total de movimiento

$d_{carga}$ : Distancia que se mueve la carga por rotación del eje del actuador

$\theta_{paso}$ : Resolución del paso del accionamiento (pasos/rev<sub>motor</sub>)

$i$  = Razón de la reducción (rev<sub>motor</sub>/rev<sub>ejereducator</sub>)

En nuestro caso:

$D_{total} = 1rev$ , porque el mecanismo requiere que solo de una vuelta, en esa vuelta hará el corte y regresará a su posición inicial

$$d_{carga} = 1rev_{motor}/rev_{ejereductor},$$

$\theta_{paso} = 200pasos/rev$ , por la selección de un motor de  $1.8^\circ$  de ángulo de paso

$i = 1rev_{motor}/rev_{ejereductor}$ , porque no tiene un caja de reducción de velocidad.

Según esto:

$$P_{total} = 200 \text{ pulsos}$$

¿Cuál es la frecuencia de pulsos para obtener el tiempo de movimiento?

El tipo más básico de perfil de movimiento es el perfil “partir-parar” donde no hay un periodo de aceleración o uno de desalación. Este tipo de perfil de movimiento se usa solamente para aplicaciones de baja velocidad porque la carga “se mueve de un tirón” a partir de una velocidad a otra y el motor que camina se atascara o faltaran pulsos si se trata de hacer cambios excesivos de velocidad. Esta tipo de perfil de movimiento se muestra en la Figura 2.30

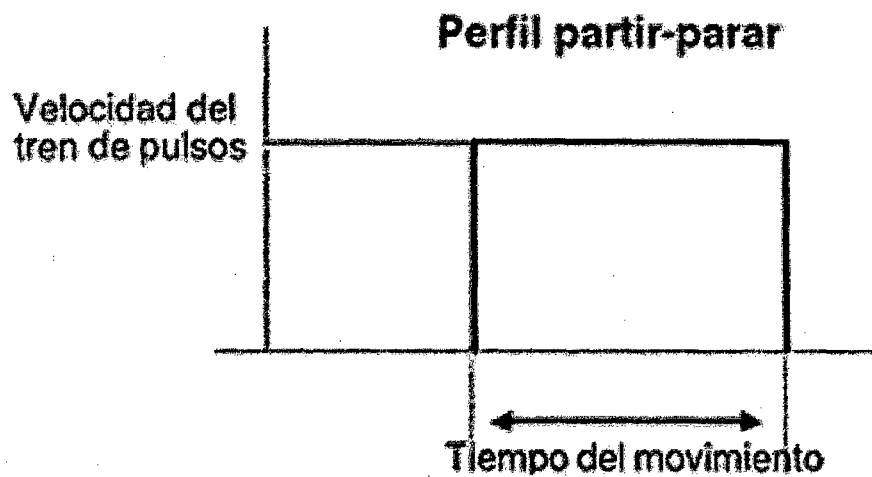


Figura 2.30 Perfil de movimiento partir-parar

La fórmula para encontrar la velocidad del tren de pulsos para el movimiento “partir-parar” es:

$$f_{ss} = \text{velocidad del tren de pulsos} = \frac{P_{total}}{t_{total}} \dots\dots (27)$$

Donde:

$$P_{total} = \text{pulsos totales} = 200\text{pulsos}$$

$$t_{total} = \text{tiempo del movimiento} = 2\text{segundos}$$

Según esto:

$$f_{ss} = 100\text{Hz}$$

### 2.2.3 Selección del Motor Paso a Paso para el Soporte de Giro de la Licuadora

Tenemos el siguiente mecanismo para el giro de la licuadora contenida de jugo para su respectivo vaciado, el cual se muestra en la Figura 2.31.

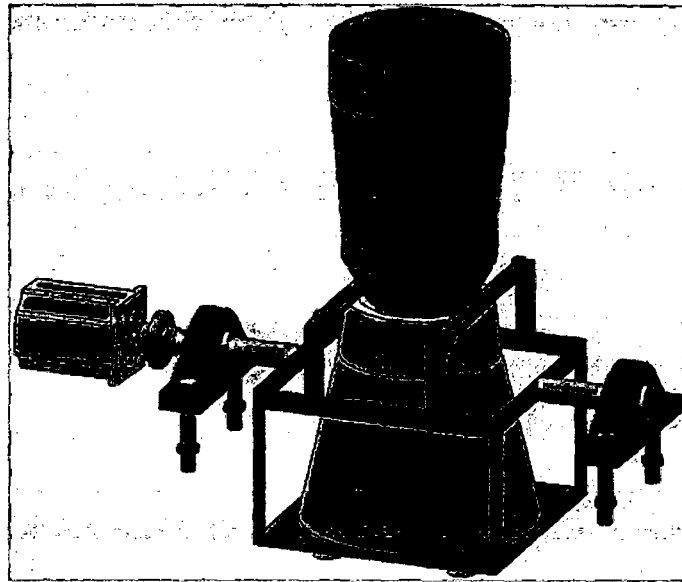


Figura 2.31 Licuadora con soporte de giro

Consta de una licuadora con un soporte que le permitirá girar sin separarse del vaso de la licuadora, se tiene un motor paso a paso para el giro de todo el sistema licuadora – soporte de giro, es para este sistema que se requiere seleccionar un óptimo motor paso a paso y para esto tenemos un procedimiento ya establecido por algunos fabricantes de motores paso a paso.

Veamos el procedimiento para seleccionar el motor paso a paso para realizar el giro de la licuadora con su soporte de giro.

### 1. Determinar el modelo de operación

Lo único que se requiere de este sistema licuadora – soporte de giro que se muestra en la figura 2.30 es que gire 90 grados en un sentido para vaciar el jugo y luego girar 90 grados en sentido opuesto para regresar a su posición inicial como se muestra en la Figura 2.32.

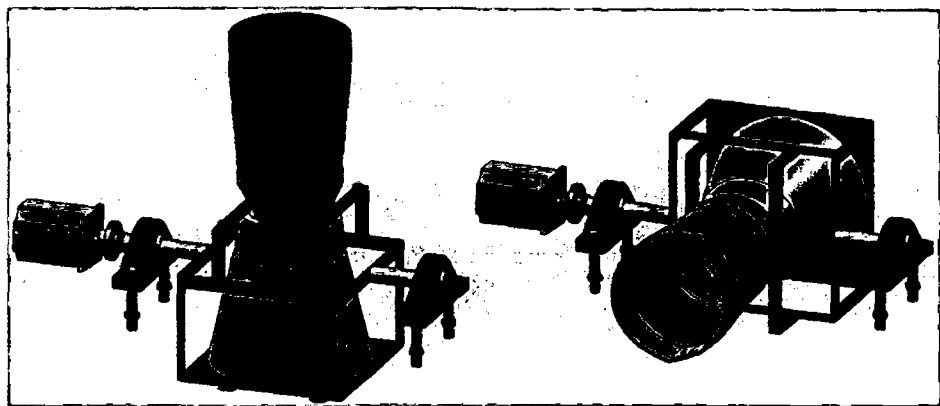


Figura 2.32 Operación a realizar de la licuadora con su soporte de giro

Los datos más importantes que se requiere para realizar los cálculos para una óptima selección del motor son la masa y el momento de inercia del sistema licuadora – soporte de giro y para esto nos ayudamos de las propiedades físicas del SolidWorks.

$$W_{\text{sistema licuadora - soporte de giro}} = 6.3 \text{ Kg}$$

$$J_{\text{sistema licuadora - soporte de giro}} = 0.0583 \text{ Kg} \cdot \text{m}^2$$

Despreciamos las fuerzas de rozamiento en las chumaceras y nuestro eje del motor paso a paso va directo al eje de nuestro sistema licuadora – soporte de giro.

## 2. Cálculo de la resolución requerida

Deseamos saber cuánto se moverá la carga para un pulso o paso del eje del motor, lo que se requiere es que su resolución sea pequeña para que el vaciado del jugo sea lento y el jugo se deposite en el vaso sin ningún problema, en el mercado encontramos motores paso a paso de  $1.8^\circ$  de ángulo de paso y con variaciones de torque, elegimos este ángulo de paso y debido a que el eje del motor con el eje del sistema licuadora – soporte de giro se unen directamente con un acople, no hay un reductor de velocidad, la resolución será igual al ángulo de paso, por lo tanto la resolución requerida es  $L_\theta = 1.8^\circ$ .

## 3. Determinar los pulsos de funcionamiento

El número de pulsos de funcionamiento se expresa como el número de señales de pulso que se suma al ángulo que el motor debe moverse para obtener el trabajo del punto A al punto B. Se requiere que gire solo  $90^\circ$  para vaciar todo el jugo, por lo tanto en número de pulsos que se requiere para realizar un giro de  $90^\circ$  es:

$$P_{totales} = \text{pulsos totales} = \frac{90^\circ}{1.8^\circ} = 50 \text{ pulsos}$$

## 4. Determinar la frecuencia de pulsos para obtener el tiempo de movimiento

El tipo más básico de perfil de movimiento es un perfil "partir-parar" donde no hay un período de aceleración o uno de desaceleración. Este tipo de perfil de movimiento se usa solamente para aplicaciones de baja velocidad

porque la carga "se mueve de un tirón" a partir de una velocidad a otra y el motor que camina se atascará o faltarán pulsos si se trata de hacer cambios excesivos de velocidad y su gráfica se muestra en la Figura 2.30.

La fórmula para encontrar la velocidad del tren de pulsos para el movimiento "partir-parar" es:

$$f_s = \text{velocidad del tren de pulsos} = \frac{P_{total}}{t_{total}}$$

Para una operación de velocidad más alta, el perfil "trapezoidal" de movimiento incluye aceleración y desaceleración controlada y una velocidad inicial diferente de cero. Con tiempos de aceleración y de desaceleración iguales, cuya grafica se muestra en la Figura 2.33.

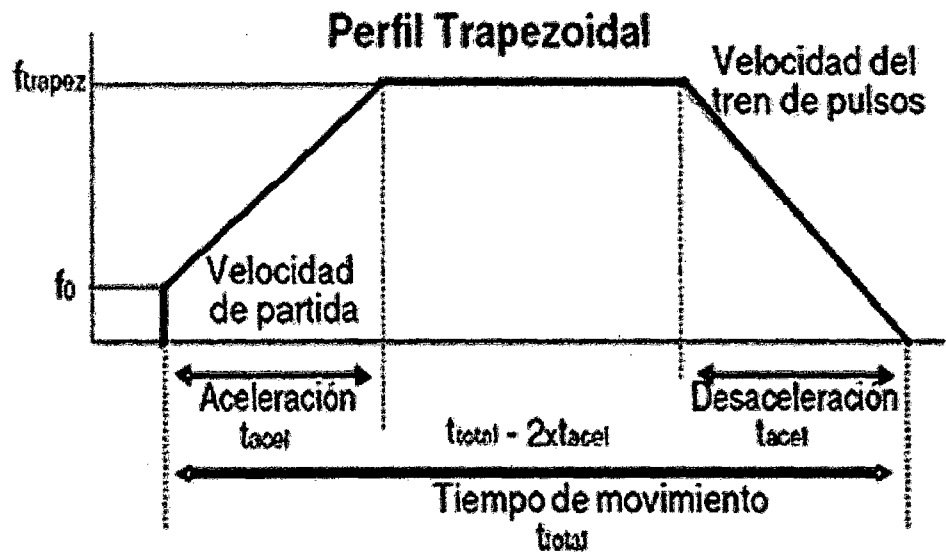


Figura 2.33 Perfil de movimiento trapezoidal



La velocidad máxima del tren de pulsos se puede encontrar usando la fórmula (28):

$$f_{\text{trapez}} = \frac{(P_{\text{total}} - (f_0 \cdot t_{\text{acel}}))}{t_{\text{total}} - t_{\text{acel}}} \dots\dots (28)$$

Donde:  $f_0 = \text{velocidad de partida}$  y  
 $t_{\text{acel}} = \text{tiempo de aceleracion o desaceleracion.}$

Para nuestro caso necesariamente tiene que ser un perfil trapezoidal para que no gire de un tirón sino acelere desde una velocidad cero hasta la velocidad requerida, por lo tanto tenemos los siguientes datos:  $f_0 = 0$ ,  $t_{\text{acel}} = 2 \text{ seg}$ ,  $t_{\text{total}} = 10 \text{ seg}$ , según esto:

$$f_{\text{trapez}} = 6.25 \text{ Hz}$$

Además como el ángulo de paso es  $1.8^\circ$  entonces en una revolución da 200 pasos, con esto la velocidad en RPM es:

$$6.25 \text{ Hz} * (60 \text{ sec}/1 \text{ min}) / 200 \text{ pasos/rev} = 1.875 \text{ RPM.}$$

## 5. Calcular el torque necesario

El torque que requiere ser suministrado por el sistema de accionamiento al actuador debe ser mayor que el torque resistivo y se puede determinar como la suma del torque de aceleración y del torque resistivo a velocidad constante y se recomienda aplicar al motor un factor de seguridad entre 20 a 100%, dependiendo del sistema en cuestión, para evitar que el motor deje de ejecutar

pulsos o se pare por cambios de carga, ya que hay varios efectos difíciles de cuantificar, tal como la velocidad del lubricante, desgaste del actuador, etc.

La fórmula para el torque mínimo que el motor debe suministrar es:

$$T_{motor} = T_{acel} + T_{resist} \dots \dots (a)$$

$T_{acel}$  = Torque que requiere la carga para acelerar y desacelerar la inercia total del sistema incluyendo la del motor.

$T_{resist}$  = torque de carga a velocidad constante para hacer funcionar el mecanismo, para vencer la fricción, a fuerzas externas, etc.

El cálculo del torque a velocidad constante para actuadores mecánicos comunes y de ese cálculo resulta una curva típica resistiva como se muestra en la Figura 2.34.

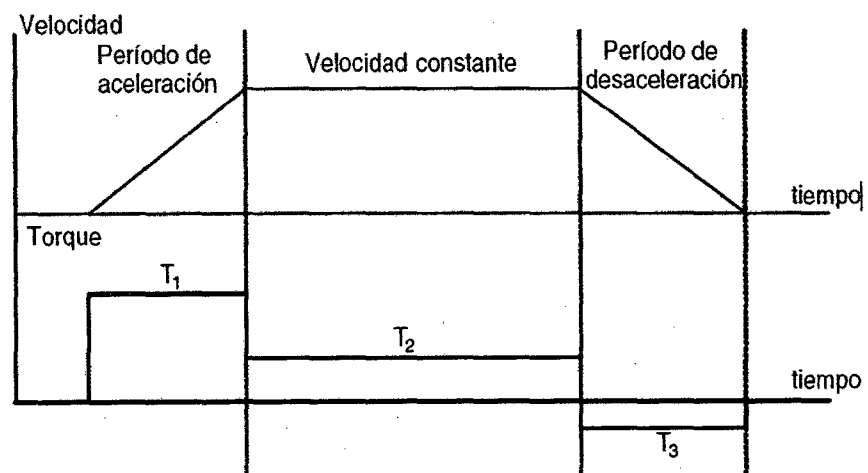


Figura 2.34 Curva típica resistiva en función de la velocidad

Note que el torque dinámico de aceleración aumenta al aumentar la aceleración.

Por eso es posible, escoger un motor de menor torque si se disminuye la aceleración, esto es, el tiempo que demora para alcanzar la velocidad constante. En relación al cálculo de inercias, se acostumbra considerar la eficiencia del actuador en este cálculo (aunque no es correcto) ya que la eficiencia debe entrar más bien en el cálculo del torque resistivo. Sin embargo, esto permite considerar formulas más simples en la determinación del torque resistivo.

El motor paso a paso no suministra el mayor torque cuando está parado sino cuando está en el medio de un paso, pero debe estar activado con la corriente del accionamiento (el accionamiento debe estar activado). Note de las curvas que se muestran en la Figura 2.35, donde el torque del motor disminuye al aumentar la velocidad (debido a la influencia de la inductancia del motor) y al aumentar la cantidad de pulsos por revolución del eje.

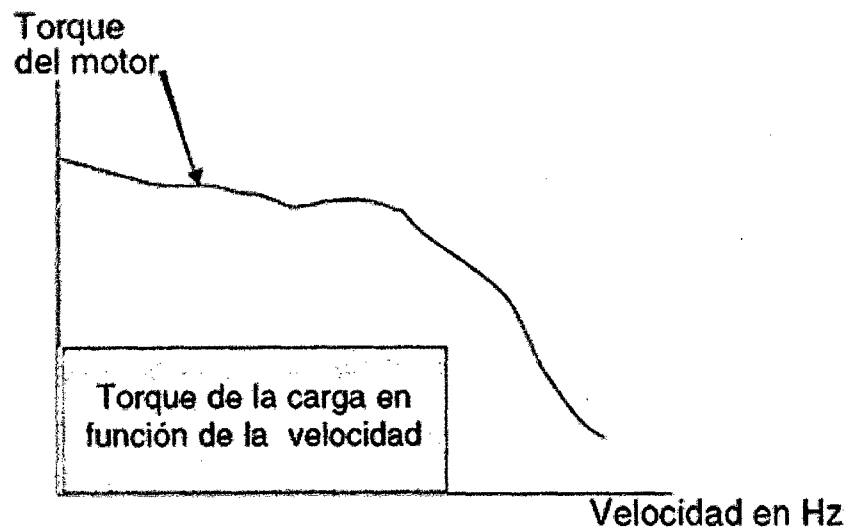


Figura 2.35 Curva de torque vs velocidad

La curva de torque versus pulsos que se entrega en los datos del motor es el torque máximo que el motor puede suministrar a cada velocidad.

El factor de seguridad típico para calcular el motor es tal que se mantenga el torque de la carga en general abajo del 50% del torque del motor disponible para evitar que el motor salte pasos o se detenga.

El torque requerido para acelerar o desacelerar una carga con inercia con un cambio lineal en velocidad es:

$$T_{acel}[N.m] = J_{total}[Kg.m^2] * \left( \frac{\Delta velocidad[RPM]}{\Delta tiempo_{acel}[s]} \right) * \frac{2\pi}{60}$$

$J_{total}$  es la inercia del motor más la inercia de la carga, el factor  $2\pi/60$  es usado para convertir el cambio en velocidad expresada en RPM a una velocidad angular (radianes/segundo).

$$J_{total} = J_{motor} + J_{reductor} + (J_{acople} + J_{carga})/i^2$$

Como nuestro sistema no tiene reductor, también despreciamos la inercia del acople, entonces  $J_{reductor} = 0$  e  $i=1$ , por tanto nuestra ecuación queda así:

$$J_{total} = J_{motor} + J_{carga}$$

Tenemos el momento de inercia del sistema licuadora – soporte de giro y para esto nos ayudamos de las propiedades físicas del SolidWorks, donde  $J_{carga} = 0.0583 Kg * m^2$ . Como todavía no sabemos qué motor elegir por el momento no sabemos la inercia del motor ( $J_{motor}$ ), no lo utilizamos pero

luego para confirmar la selección lo incluimos, con esto  $J_{total} = J_{carga} = 0.0583 \text{ Kg} \cdot \text{m}^2$ , por lo tanto el torque requerido para acelerar la inercia es:

$$T_{acel}[\text{N} \cdot \text{m}] \approx J_{total}[\text{Kg} \cdot \text{m}^2] * \left( \frac{\Delta \text{velocidad}[\text{RPM}]}{\Delta \text{tiempo}_{acel}[\text{s}]} \right) * \frac{2\pi}{60}$$

$$\approx 0.0583[\text{Kg} \cdot \text{m}^2] * \left( \frac{1.875[\text{RPM}]}{2[\text{s}]} \right) * \frac{2\pi}{60} \approx 0.0058[\text{N} \cdot \text{m}]$$

Después, necesitamos determinar el torque resistivo en la operación de movimiento. Si existe la máquina es a veces posible medir realmente el torque resistivo. De otra forma, es necesario estimar este valor por experiencias similares, el diseño del sistema licuadora – soporte de giro se hizo de tal manera que el centro de masa esté alineado con el eje de giro del motor, para que solo se pueda vencer el torque de aceleración, ojo que ese alineamiento del centro de masa es con el jugo incluido en el vaso de la licuadora, pero cuando se empieza a vaciar el jugo el centro de masa empieza a variar y por este efecto empieza a aparecer un torque de resistencia al giro, es este torque que tenemos que calcular.

Ahora veamos en cuanto varía el centro de masa y el torque que tenemos que vencer. En la Figura 2.36 vemos que cuando el centro de masa está alineado con el eje de giro las  $F_g = F_a + F_b$ , debido a que despreciamos rozamientos en las chumaceras y no hay torque que se oponga al giro.

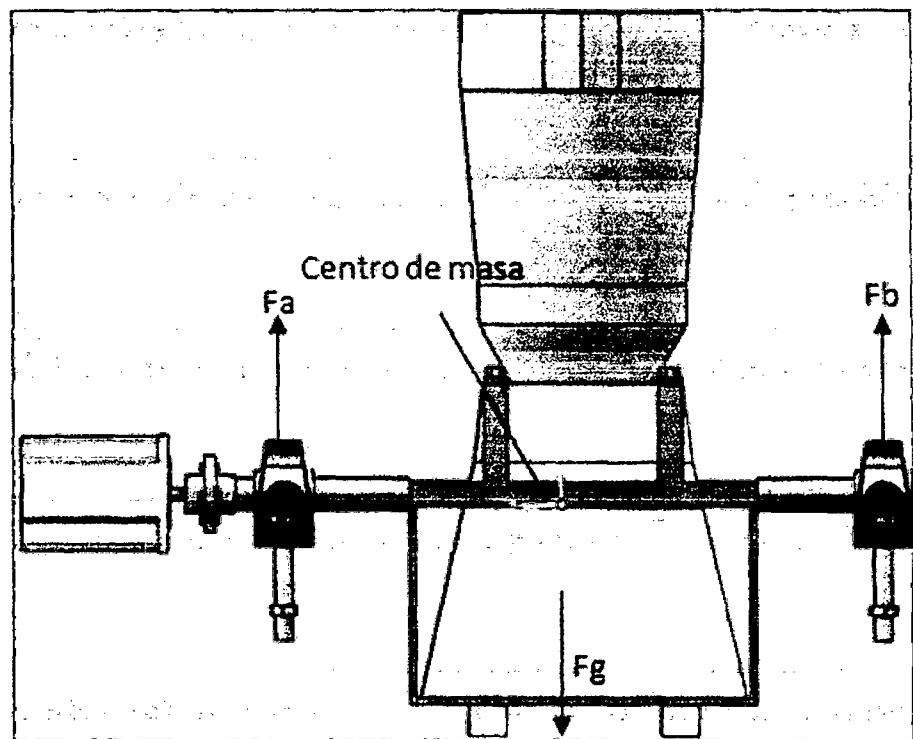


Figura 2.36 Centro de masa del sistema licuadora-soporte alineado con el eje

Pero cuando empieza a vaciar el jugo el centro de masa varia. Como el vaso solo tiene un peso despreciable, entonces para simular cuanto varia el centro de masa eliminamos el vaso que inicialmente esta con las frutas y el líquido, es por esto que tiene un peso, pero eliminado el vaso se elimina todo ese peso que resulta de ser vaciado el jugo y el centro de masa varia como se muestra Figura 2.37 y varia 45 mm.

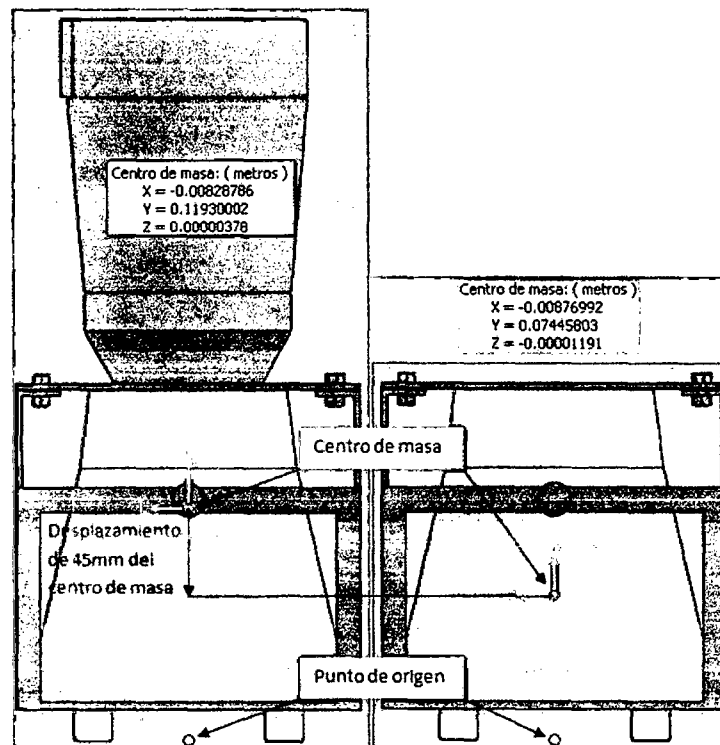


Figura 2.37 Variación del centro del masa al verter el jugo

Con esto ya podemos estimar el torque de resistencia, debido a que hay experiencias similares como el que se muestra en la Figura 2.38 donde  $F_B$  es similar a la fuerza de gravedad del sistema licuadora – soporte de giro sin el peso del vaso con contenido y  $D/2$  similar a la variación de del centro de masa.

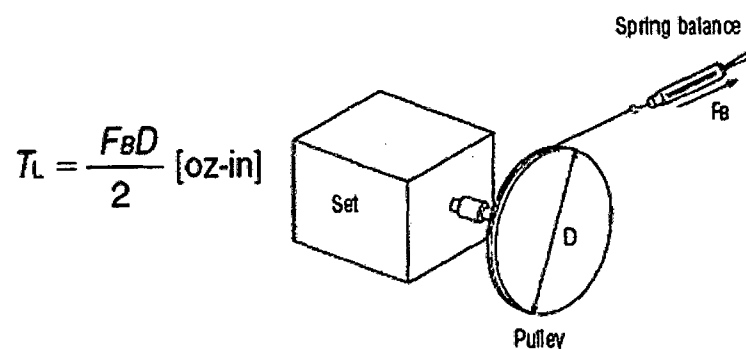


Figura 2.38 Modelo a copiar para estimar el torque de resistencia

Según esto tenemos la siguiente fórmula

$$\begin{aligned}
 T_{resist} &= m_{sistema\ licuad-soporte} [Kg] * g \left[ \frac{m}{s^2} \right] * 0.045[m] \\
 &= 4.3[Kg] * 9.81 \left[ \frac{m}{s^2} \right] * 0.045[m] \approx 1.9[N * m]
 \end{aligned}$$

De la ecuación (a) tenemos:

$$\begin{aligned}
 T_{motor} &= T_{acel} + T_{resist} = 0.0058[N * m] + 1.9[N * m] = 1.9058[N * m] \\
 &= 270[oz - in]
 \end{aligned}$$

## 6. Seleccionar y confirmar el motor paso a paso

En relación a motores paso a paso tenemos ya asignado un proveedor y este proveedor tiene un motor paso a paso de torque bipolar de 400[oz-in] que es superior a nuestro torque requerido y la inercia del rotor es 0.00006[Kg.m<sup>2</sup>] por lo tanto el torque real del motor seria modificado:

$$\begin{aligned}
 T_{acel}[N.m] &\approx J_{total}[Kg.m^2] * \left( \frac{\Delta velocidad[RPM]}{\Delta tiempo_{acel}[s]} \right) * \frac{2\pi}{60} \\
 &\approx (0.00006 + 0.0583)[Kg.m^2] \left( \frac{1.875[RPM]}{2[s]} \right) \frac{2\pi}{60} \approx 0.0058[N * m]
 \end{aligned}$$

De modo que no ha variado en nada:

$$\begin{aligned}
 T_{motor} &= T_{acel} + T_{resist} = 0.0058[N * m] + 1.9[N * m] = 1.9058[N * m] \\
 &= 270[oz - in]
 \end{aligned}$$



El motor seleccionado puede satisfacer las condiciones dadas. Observe las curvas del torque en función de la velocidad, de la Figura 2.39, donde se puede ver que el motor siempre tiene más torque disponible que el requerimiento del torque resistivo. El factor de seguridad en este caso es:

$$400[\text{oz} * \text{in}]/270[\text{oz} * \text{in}] = 1.48$$

Un 48% más de factor de seguridad que es admisible.

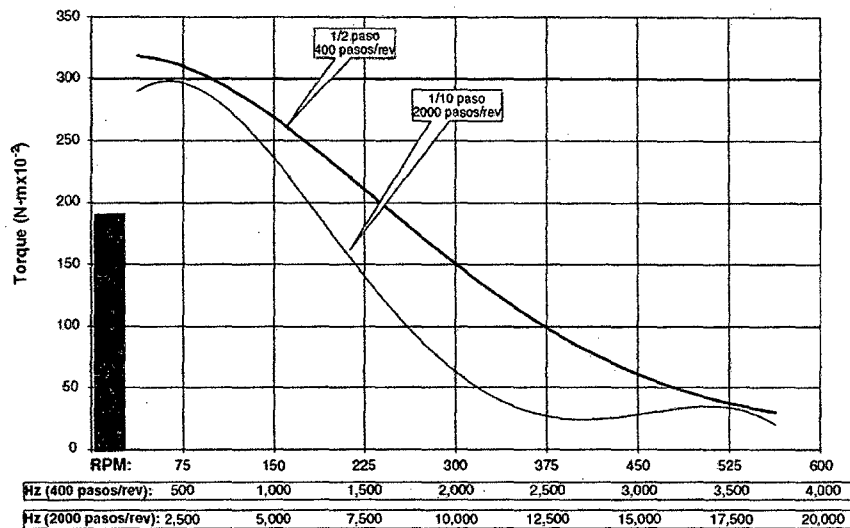


Figura 2.39 Curva torque vs velocidad motor pap 400 oz-in

Como se puede observar a bajas velocidades este motor tiene un gran torque y no tendríamos ningún problema en utilizarlo para nuestro sistema. En el ANEXO E se muestra el catálogo de este motor paso a paso seleccionado.

### 2.3. Componentes del Sistema Mecánico

En esta parte vamos a describir los componentes del sistema mecánico por partes, sus funcionalidades, el material utilizado, además haremos algunos análisis de diseño de algunos componentes que requieren dicho análisis, para esto utilizaremos el SimulationXpress del SolidWorks. Solo mostraremos los componentes sin medidas, El plano del sistema completo se muestra en el Anexo A.

#### 2.3.1. Estructura del Sistema Mecánico

##### Soporte fruteros

Este componente es el soporte de los recipientes de las 5 frutas y el recipiente del líquido tratado para preparar el jugo, esta estructura se muestra en la Figura 2.40. El material es de acero inoxidable AISI 304, soportará una carga de 785N entre los pesos de las frutas, el líquido y los recipientes.

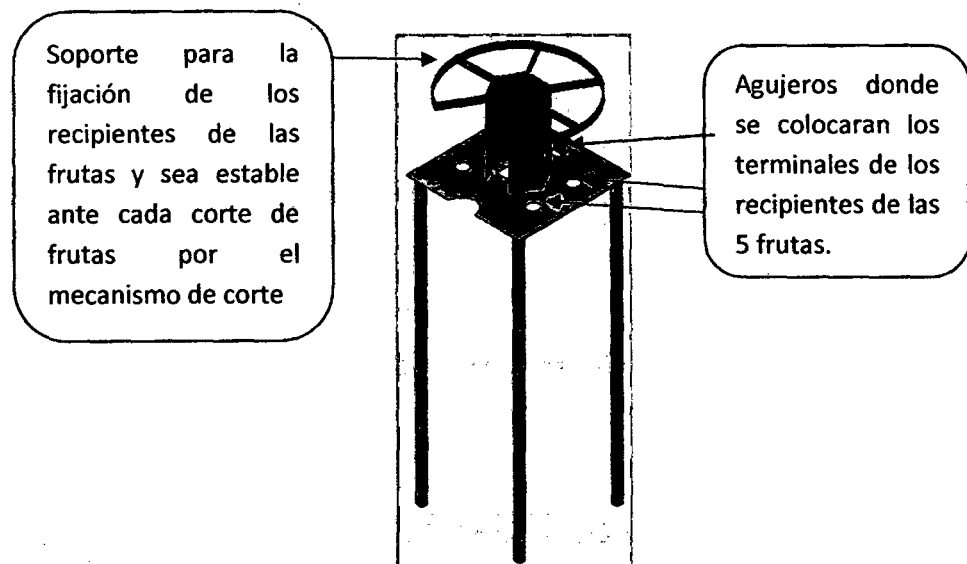


Figura 2.40 Soporte recipientes de las frutas y el líquido

En la Figura 2.41 se muestra el análisis de diseño donde se aplicó una fuerza de 800N a la cara que soportara los recipientes de frutas y del líquido, se muestra la distribución de desplazamiento en el modelo, donde podemos apreciar que el máximo desplazamiento es de 0.229mm, prácticamente no se desplaza ninguna parte del modelo, además tiene un factor de seguridad de 13, quiere decir que nuestro modelo está bien y podemos seguir con ese diseño.

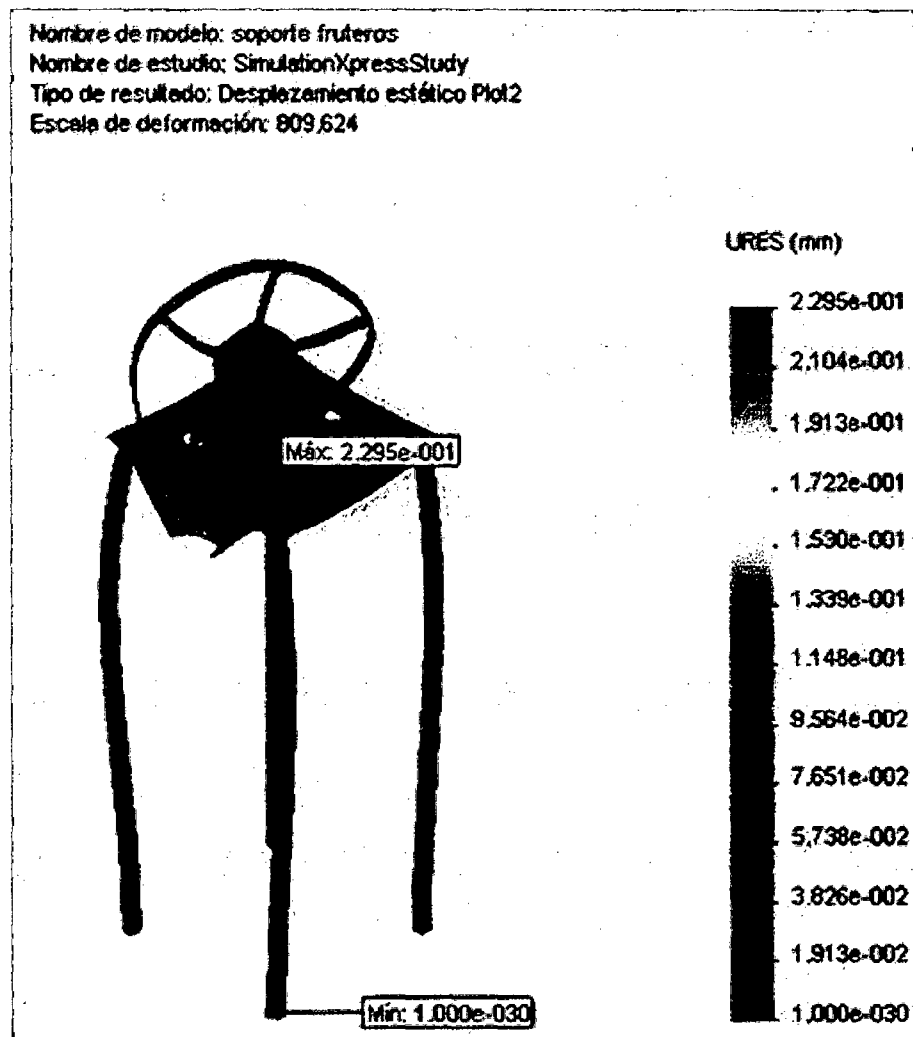


Figura 2.41 Distribución de desplazamiento del modelo

### Soporte mecanismo de corte

Este componente es soporte de los mecanismos biela-manivela, las correderas de los pistones del mecanismo de corte, los soportes de los motores paso a paso, el soporte para las chumaceras donde se colocará la licuadora con su soporte de giro, soporte para el motor paso a paso para el giro de la licuadora y soporte para la recepción del vaso de jugo, este soporte se muestra en la figura 2.42. El material es de acero inoxidable AISI 304 y no soporta mucha carga.

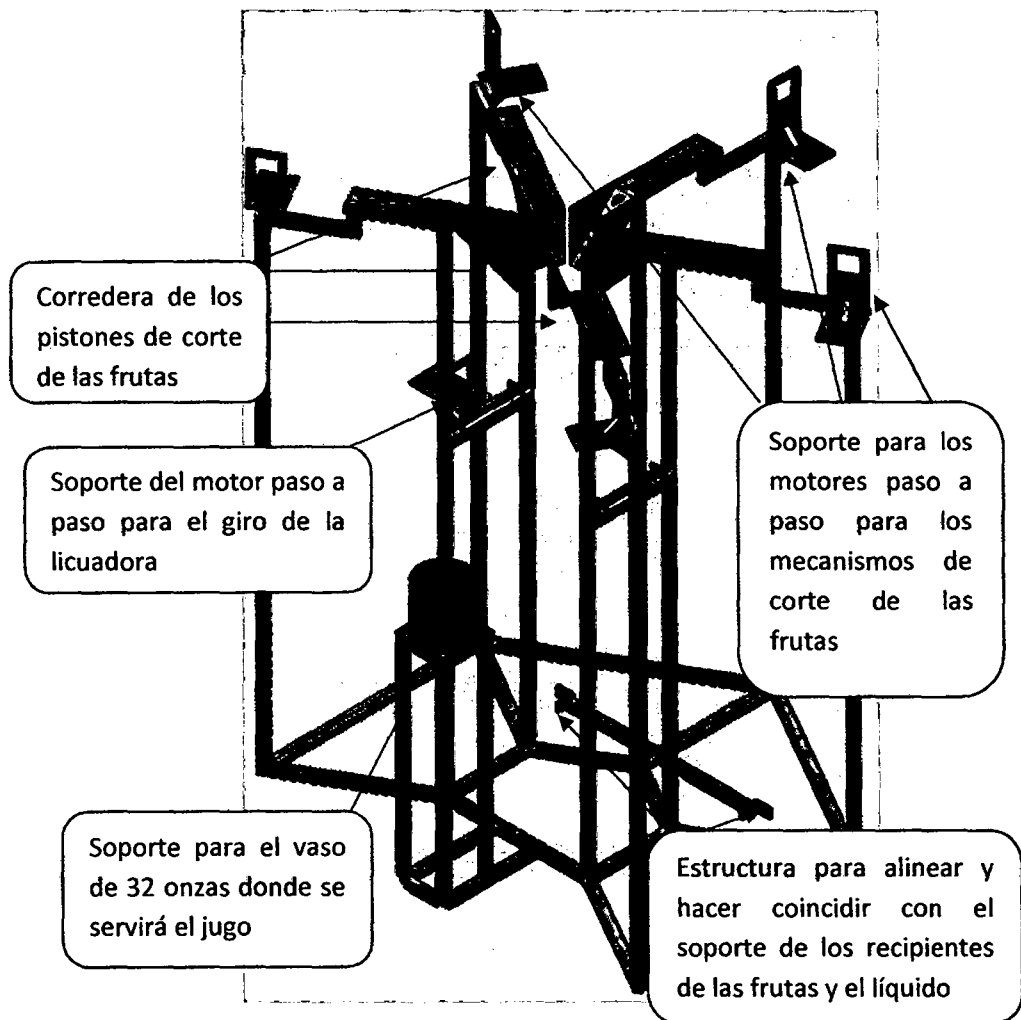


Figura 2.42 Soporte para los mecanismos de corte de las frutas

### Recipientes de las frutas

En la Figura 2.43 se muestran estos recipientes y están diseñados de acuerdo a la cantidad de jugo a preparar por este sistema, como mencionamos en la sección 2.2.1, que este sistema tiene que preparar 50 jugos surtidos, por lo tanto estos recipientes tienen que tener un volumen mayor a 50 veces el volumen del recipiente experimental (volumen del pistón de corte de las frutas), que es  $V_r = 0.00019635 \text{ m}^3$ , según este análisis el volumen de los recipientes de las frutas es de  $0,010658\text{m}^3$ ,  $0,010658\text{m}^3/50 > 0.00019635 \text{ m}^3$ , además el material es de acero inoxidable AISI 316.

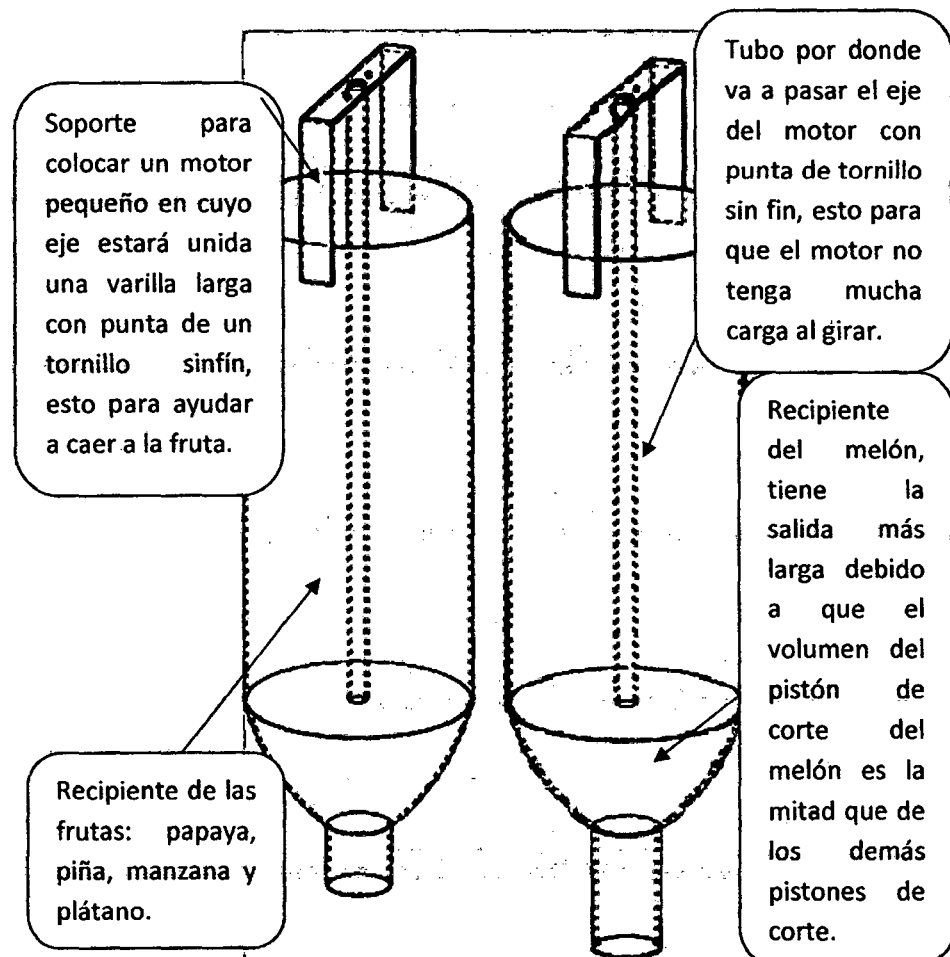


Figura 2.43 Recipientes de las frutas

Ahora este diseño de los recipientes de las frutas podría generar un atascamiento de estas, justo en la parte final del embudo, como se muestra en la Figura 2.44

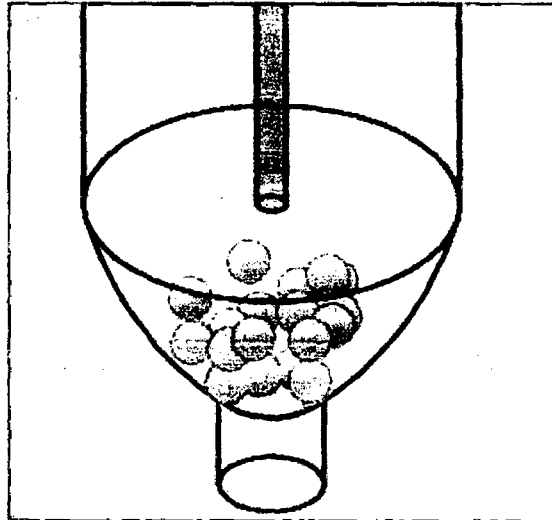


Figura 2.44 Punto probable de atascamiento de las frutas

Para garantizar que esto no suceda, nosotros vamos a poner un motor AC, pequeño pero de gran torque, que estará unido a un eje con una punta similar a un tornillo sinfin, tal como se muestra en la Figura 2.45

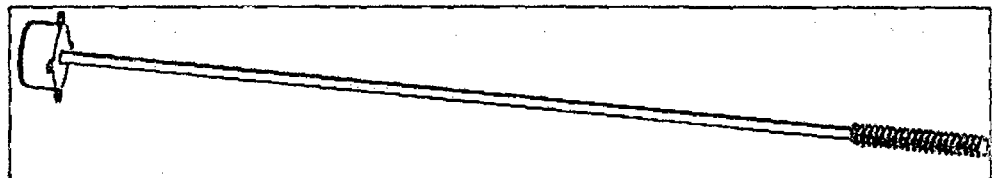


Figura 2.45 Motor unido a un eje largo con punta de tornillo sinfin

Este motor acoplado con dicho eje será colocado sobre un soporte de tal manera que la punta de este eje pueda llegar al punto probable de atascamiento de la fruta y así ayudarlas a caer, tal como se muestra en la Figura 2.46.

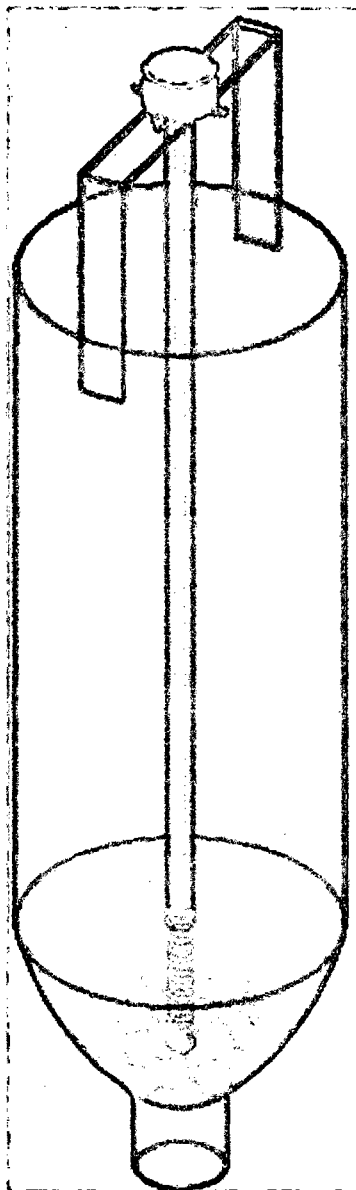


Figura 2.46 Punta del eje del motor que llega al punto de atascamiento

Cuando el motor gira en la dirección en el que el tornillo sinfín empuja hacia abajo, este hará que las frutas picadas resbalen y caigan hacia el pistón de corte. Cabe mencionar que este motor se activará por 5 segundos antes que el pistón salga a separar el volumen de fruta y así asegurar la porción de fruta para el preparado del jugo.

### Recipiente del líquido

Este recipiente también está diseñado de acuerdo a la cantidad de jugos a preparar por este sistema, por cada jugo surtido preparado se llena 0.4 litros de líquido, por lo tanto se requiere que este recipiente tenga un volumen mayor a 20 litros, según este análisis nuestro recipiente de líquido tiene 21.54 litros suficiente para los 50 jugos surtidos a preparar, este recipiente se muestra en la Figura 2.47 y como se puede apreciar tiene una geometría no convencional, esto para que quepa en nuestro diseño, el material es de acero inoxidable AISI 316, también se le acopla una válvula solenoide para que el vaciado del líquido a la licuadora sea por tiempo.

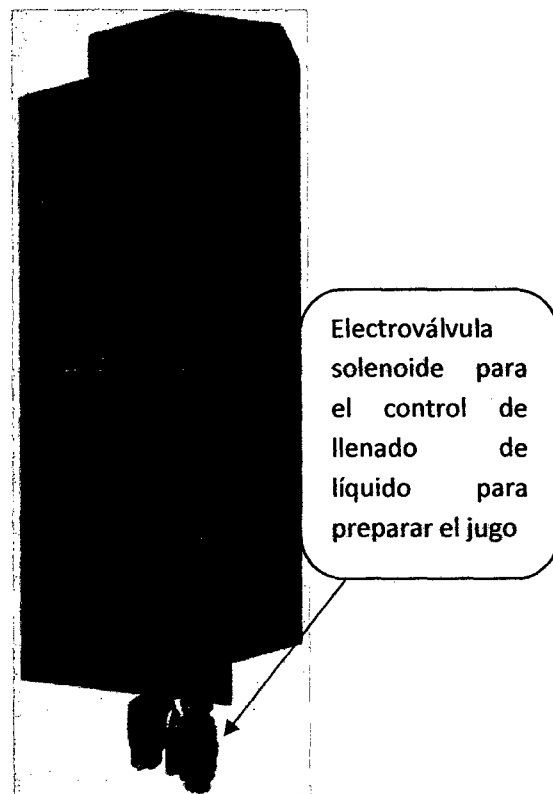


Figura 2.47 Recipiente del líquido acoplado con una electroválvula



### 2.3.2. Sistema Selector de Frutas

#### Mecanismo de corte de las frutas

Este mecanismo es un mecanismo tipo biela- manivela accionado por un motor paso a paso para controlar el giro, el pistón que tiene este mecanismo tiene un cilindro hueco por ambos lados superior e inferior cuyo volumen es el mismo volumen del recipiente experimental:

$$V_r = \frac{\pi * (0.05)^2 * 0.1}{4} m^3 = 0.00019635 m^3$$

Este volumen es el que tiene que separar este mecanismo para el preparado del jugo, también tenemos una corredera del pistón para que toda la fruta separada por este mecanismo caiga al vaso de la licuadora, pistón y corredera son de material AISI 316 y se muestran en la Figura 2.48.

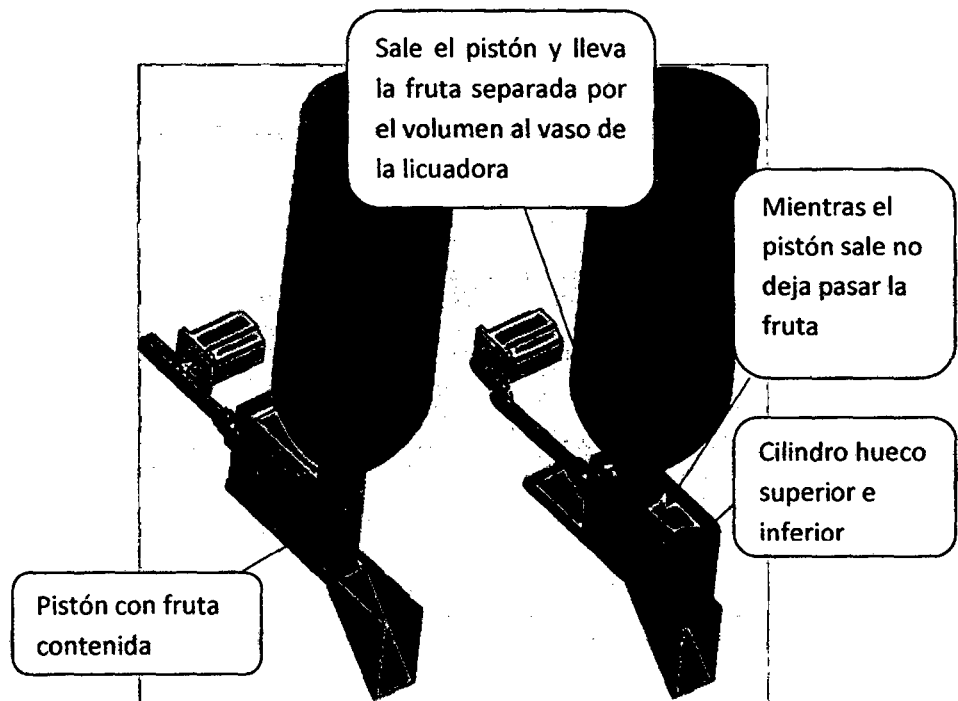


Figura 2.48 Mecanismo de corte de las frutas

### 2.3.3. Licuadora con Soporte de Giro

Este sistema consta de una licuadora sujeta con un soporte de giro para el vaciado del jugo a un vaso de 32 onzas, está accionado por un motor paso a paso para realizar el giro de 90 grados para vaciar el jugo y luego girar otros 90 grados en sentido inverso para regresar a su posición inicial a esperar las frutas para un nuevo proceso, en la Figura 2.49 se muestra una licuadora acoplada con un soporte, este soporte es de acero inoxidable AISI 304 y tiene su eje de giro justo en el centro de masa de la licuadora contenida con el jugo, para una menor carga en el giro.

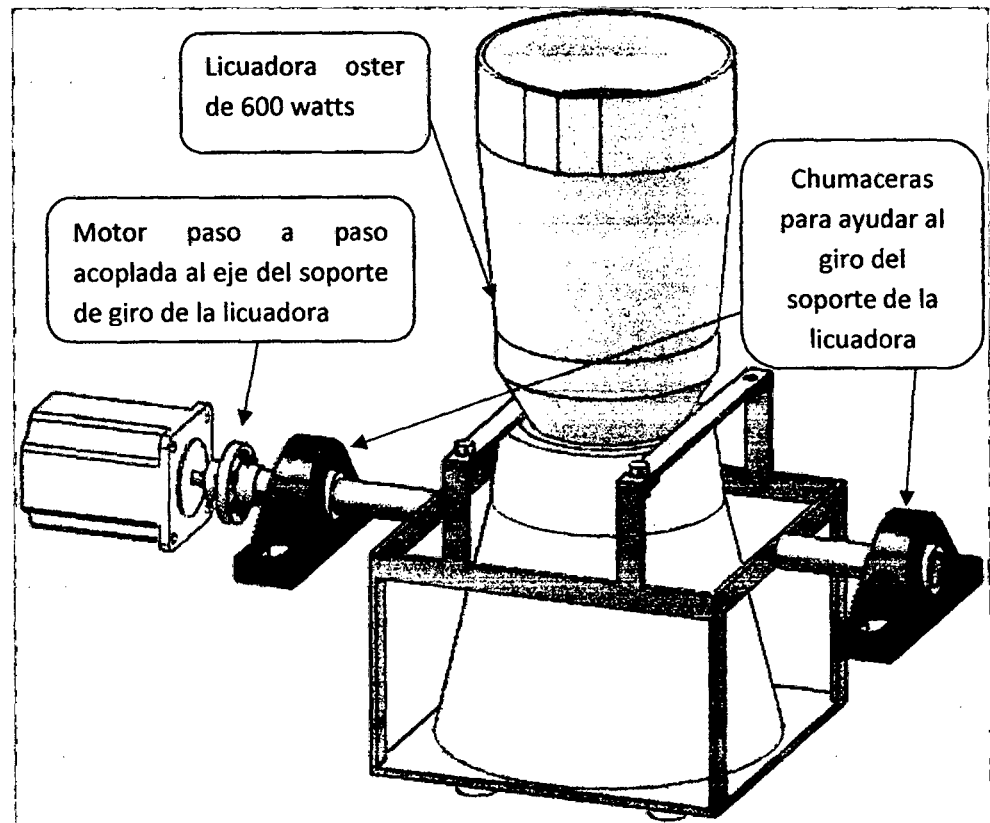


Figura 2.49 Licuadora con soporte de giro

En la Figura 2.50 se muestra el análisis de diseño del soporte de giro de la licuadora donde se aplicó una fuerza de 50N a la cara que soportara el peso de la licuadora contenida con el jugo, se muestra la distribución de desplazamiento en el modelo, donde podemos apreciar que el máximo desplazamiento es de 0.03162 mm, prácticamente no se desplaza ninguna parte del modelo, además tiene un factor de seguridad de 30, quiere decir que nuestro modelo está bien y podemos seguir con ese diseño.

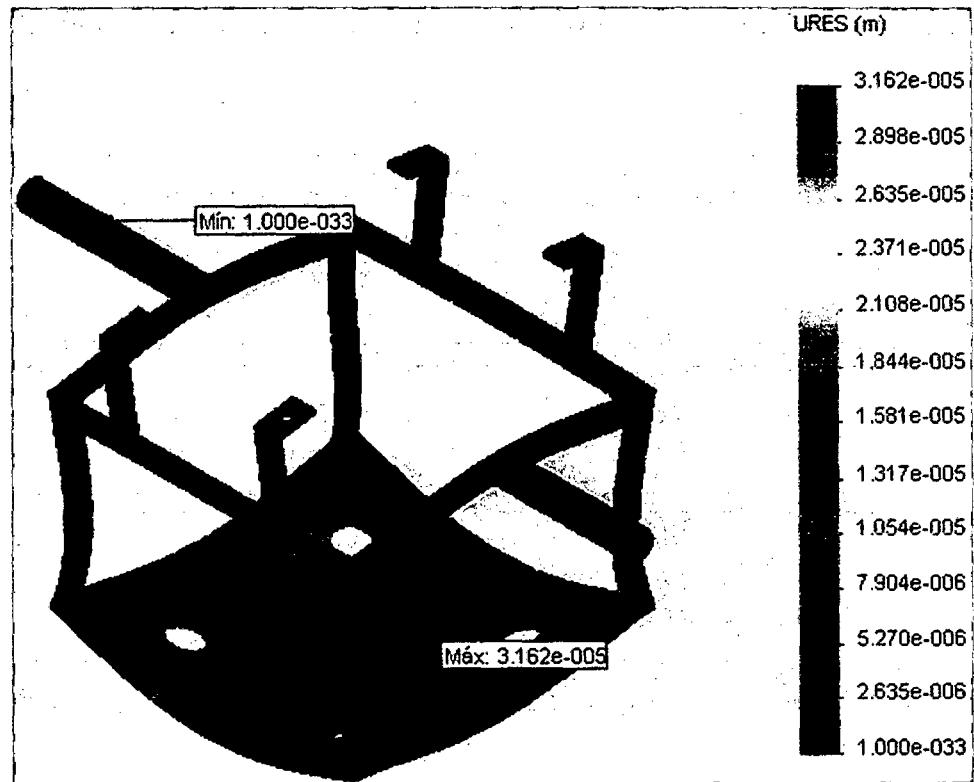


Figura 2.50 Distribución de desplazamiento del modelo

### 2.3.4. Mecanismo de Separación y Posicionado de Vasos

Este mecanismo sirve para separar un vaso del conjunto de vasos unidos concéntricamente a presión, para luego bajar el vaso hasta el punto donde recibirá el jugo de la licuadora. Este mecanismo se muestra la Figura 2.51, donde podemos apreciar el mecanismo completo con la explicación de las funciones de sus elementos más importantes.

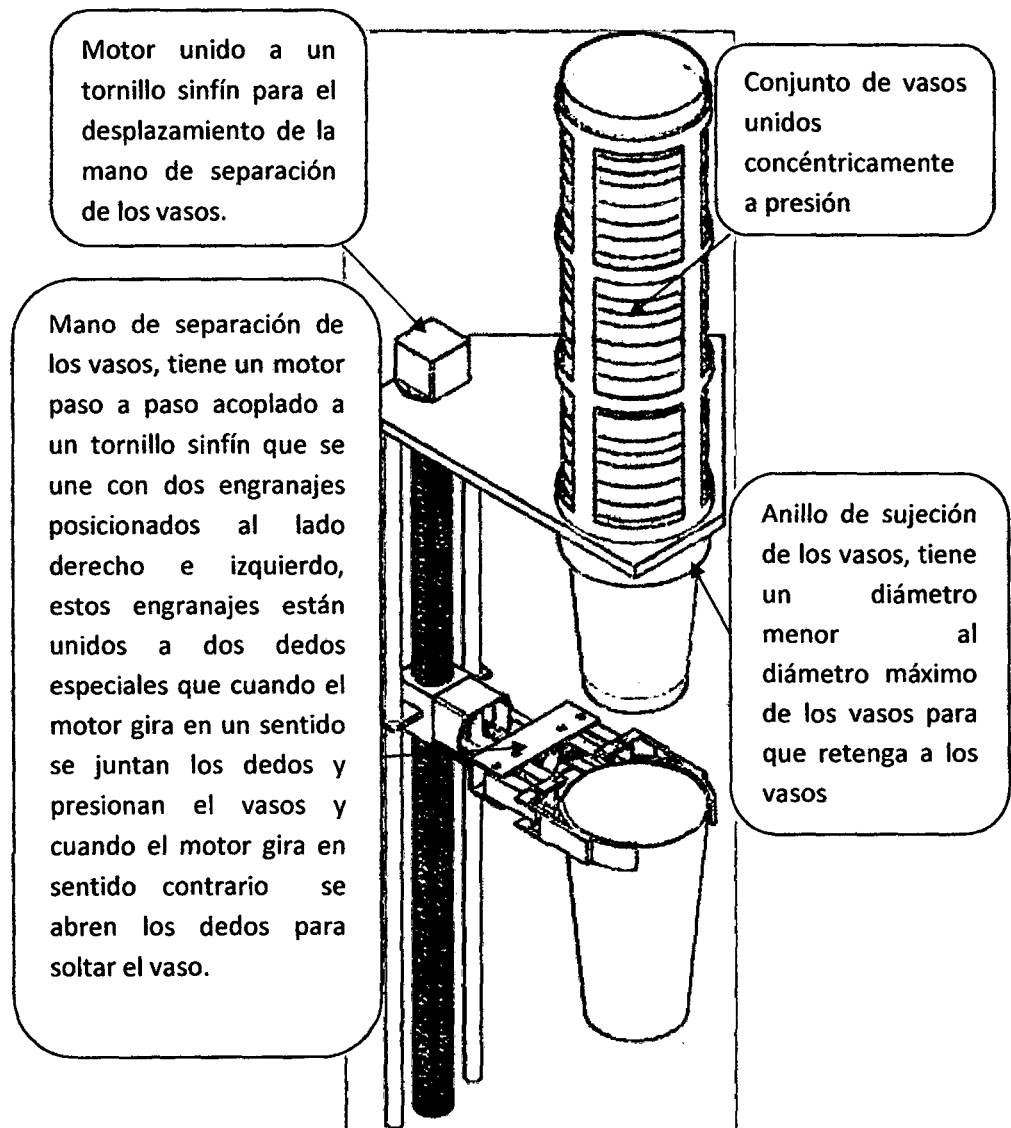


Figura 2.51 Mecanismo de separación y posicionado de vasos

## CAPÍTULO 3

### DISEÑO ELECTRÓNICO DEL SISTEMA

#### 3.1. Microcontroladores y DSP

Los microcontroladores clásicos, denominados de forma resumida MCU, son circuitos integrados que contienen un procesador digital completo junto a diversos periféricos auxiliares que facilitan el desarrollo a las aplicaciones a las que se dedican. Su parecido con los DSP es muy grande, pero las diferencias que los distinguen hacen que sus campos de aplicación sean diferentes.

De forma resumida puede decirse que los DSP son microcontroladores dotados de los recursos físicos y lógicos necesarios para poder soportar las aplicaciones específicas del procesamiento digital de señales.

Las instrucciones aritméticas complejas de los MCU se ejecutan en varios ciclos, mientras que las de los DSP sólo precisan uno. En los DSP siempre se dispone de conversores AD rápidos y precisos. Dado el carácter marcadamente matemático de los programas para DSP, éstos están preparados para ser programados con lenguajes de alto nivel, como el C. La velocidad y el rendimiento de los DSP son muy superiores a los habituales en los MCU.

En el momento que la expansión de las aplicaciones de los MCU ha alcanzado cotas extraordinarias y se fabrica un sinfín de productos y sistemas que llevan embebidos uno o varios, el nivel científico y los avances tecnológicos han logrado fabricar DSP que añaden al MCU los recursos necesarios para soportar el procesamiento digital de las señales y cubrir así todo el campo de posibilidades y aplicaciones que envuelve.

Al analizar el comportamiento del MCU se espera que ejecute la secuencia de instrucciones tal como la especifica el programa, siendo las interrupciones la única causa que puede apartarle de la misma. Su funcionamiento es claramente previsible y su actuación determinística. Por el contrario, en muchas aplicaciones con DSP los resultados son consecuencia inmediata del flujo de datos que suministra la señal muestreada y la ejecución es dependiente de los datos obtenidos en tiempo real

### **DSP y DSC**

La empresa **Microchip Technology Inc.** Ocupa el primer puesto en ranking mundial de microcontroladores de 8 bits desde el año 2003; sus modelos son conocidos popularmente con el nombre genérico PIC. Tras el exitoso lanzamiento de las familias de microcontroladores de 16 bits PIC24FXXX y PIC24HXXX, los usuarios necesitan nuevos dispositivos que soporten funciones de procesamiento digital de señales para atender las nuevas tendencias del mercado orientado al aumento de la conectividad por internet, las mejoras relacionadas con la imagen y el sonido, el control de motores, etc.

Las aplicaciones modernas mezclan las funciones típicas MCU con las de procesamiento digital de señales (DSP). Esta situación ha impulsado a microchip a fabricar un circuito híbrido MCU/DSP, cuyo manejo es similar a los clásicos microcontroladores pero que incluye las principales prestaciones de los DSP. Así ha nacido el **Controlador Digital de Señales**, abreviadamente **DSC (Digital Signal Controller)**, que reúne las características de un microcontrolador PIC de 16 bits y las de un DSP de gama baja. En base a este controlador digital de señales, Microchip consigue reunir lo mejor de ambos mundos, marcando el comienzo de una nueva era en el mercado de controladores.

La primera generación de DSC, denominada dsPIC30F, constituyó un intento de parte de microchip de facilitar el acercamiento al mundo del procesamiento digital de señales a sus usuarios de MCU de 8 y 16 bits. Con esta intención se ha mantenido el entorno de la arquitectura y juego de instrucciones de los MCU tradicionales, pero añadiendo la funcionalidad y los principales requisitos de los DSP, para facilitar la transición a sus clientes habituales. Posteriormente, la segunda generación de los DSP, llamada dsPIC33F, ha potenciado las capacidades, el número de periféricos y el rendimiento, permitiendo acceder a campos de aplicación más complejos, además el voltaje de alimentación admite un rango menor a los demás MCU comprendido entre 2 y 3.6 VDC, el rendimiento máximo alcanza 40MIPS cuando el voltaje de alimentación tiene un valor de 3,3 VDC, tiene una memoria FLASH que alcanza 256KB.

### **Microcontrolador dsPIC**

Para elegir el controlador que pueda gobernar nuestro sistema, debemos basarnos por ciertas características técnicas, como por ejemplo la cantidad de entradas y salidas digitales de nuestro sistema, la cantidad de memoria flash de programa que pueda utilizar nuestro controlador, la velocidad de respuesta del controlador y los periféricos externos a utilizar. En cuanto a las entradas y salidas digitales nuestro sistema tiene 13 entradas digitales y 40 salidas digitales, por lo tanto se requiere un controlador con un mínimo de 53 entradas o salidas configurables, en cuanto a la memoria flash de programa se necesita un mínimo de 16Kbytes, en cuanto al tiempo de respuesta de las salidas digitales se requiere de un periodo mínimo de 10ms entre cada pulso y en cuanto a periféricos podríamos utilizar, los interruptores externos, los temporizadores, en un futuro los módulos de comunicación como el UART o SPI.

Entonces a partir de estas características técnicas elegimos un controlador, en el mercado encontramos una infinidad de controladores, van apareciendo nuevas tecnologías con mejoras sustanciales y empaques más pequeños, es por eso que nosotros elegimos un dsPIC por la cantidad de pines que maneja (hasta 100 pines), porque tiene una capacidad de memoria flash de programa de hasta 256Kbytes, porque trabaja hasta 40MIPS (mega instrucciones por segundo), porque tienen una inmensa cantidad de periféricos para distintas aplicaciones embebidos en un solo chip y además porque es una tecnología nueva, porque a futuro podría ayudarnos en el monitoreo de nuestro sistema



vía internet y además estos dsPIC's no solo se programan con el assembler sino que ahora ya existe un compilador C para que se pueda programar estos dispositivos en código C de mayor entendimiento y facilidad de programación y luego compilarlo a código assembler para su grabado al dsPIC.

El dsPIC que vamos a utilizar para controlar nuestro sistema va a ser un dsPIC33FJ64GS606 que tiene 64 pines, de estos 55 son entradas o salidas digitales configurables, que es lo mínimo que requerimos para nuestro sistema, además tienen una memoria flash de programa de 64Kbytes y algunas características técnicas más que mostramos en el ANEXO B. En la Figura 3.1 mostramos nuestra tarjeta de desarrollo con el controlador (hecho por nosotros) para el control de nuestro sistema, el cual tiene toda la conexión necesaria que muestra en la Figura 3.2, además de su conector para el grabado del programa.

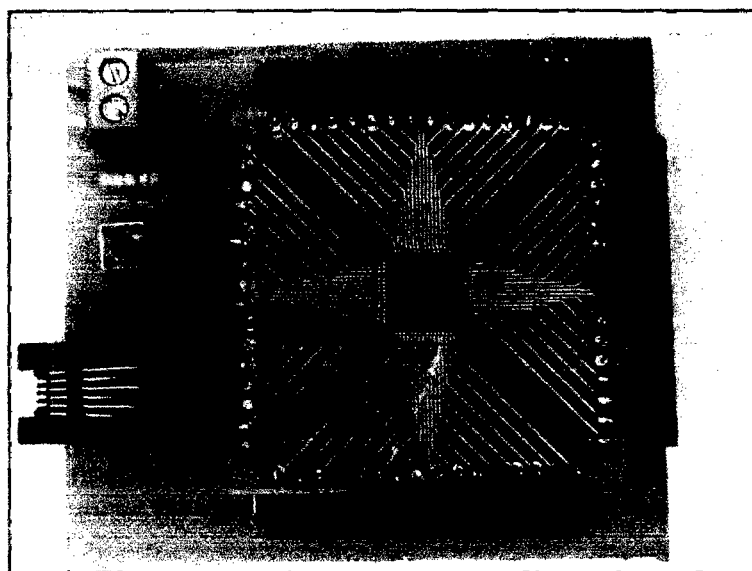


Figura 3.1 Tarjeta de desarrollo con un microcontrolador dsPIC33FJ64GS606

En la Figura 3.2 vemos la conexión necesaria que requiere un dsPIC33 para una óptima operación.

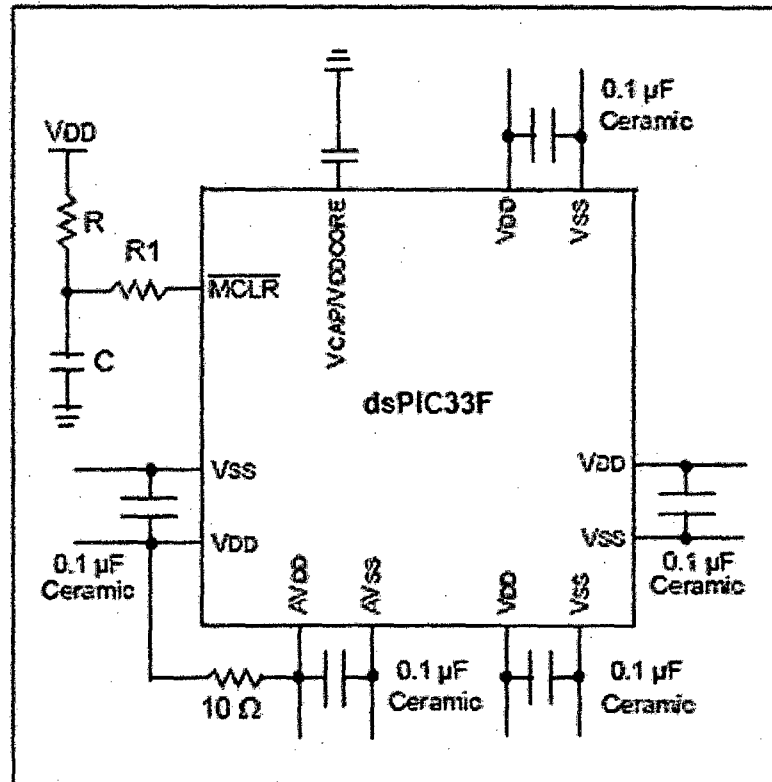


Figura 3.2 Conexión necesaria para un óptimo funcionamiento del dsPIC

### 3.2. Fuente de Poder

Nuestro sistema necesita de una fuente de voltaje con varias salidas de voltaje, para nuestro dsPIC necesitamos 3.3VDC, para nuestros controladores de los motores paso a paso necesitamos 5VDC, para los selectores de monedas 12VDC y para los motores paso a paso 24VDC, como podemos notar es necesario hacer una fuente con estas salidas de voltaje y que además sea muy estable y robusta para que no haya problemas con todo el sistema.

Debemos tener en cuenta que se van a utilizar 8 motores paso a paso de corriente de fase entre 1.5 a 2 amperios y 24VDC, pero para no utilizar un transformador de 15 amperios, se ha visto la necesidad de hacer funcionar el sistema utilizando un motor a la vez, cosa que cuando cada motor necesite de la corriente para funcionar, solo se irá al motor que estará funcionando.

En la Figura 3.3 se muestra el circuito de la fuente de poder donde se muestra desde el transformador 220V/24V 5A hasta los voltajes de 24VDC, 12VDC, 5VDC y 3.3V.

Empecemos a analizar este circuito, primeramente partimos de una red alterna 220voltios, es necesario que esta red también pueda entregarnos unos 10 amperios, para el funcionamiento de la licuadora, la electroválvula, etc. Primeramente como es lógico tenemos que reducir la tensión alterna y para esto utilizamos un transformador de 220V/24V de 5 amperios, luego esta tensión de 24V es rectificadora por un puente de diodos (BR1) de 50V y 8A para que soporte los 5A que pueden circular por ahí, obteniéndose así una onda completa. Después la señal se filtra por medio de C1 un condensador electrolítico de 35V y 4700uF, consiguiendo de esta forma una tensión continua no estabilizada pero continua. A partir de esta tensión no estabilizada de 24 voltios DC se inyectan a los distintos circuitos de los reguladores de voltaje, también esta tensión de 24 voltios continua pero no estabilizada servirá para el manejo de los motores paso a paso que trabajarán a esta tensión, debido a que los motores paso a paso no necesitan de una fuente regulada, solamente es necesaria que sea continua, por las mismas características de los motores.

Para conseguir los 5V y 3.3V que son esenciales para el control de nuestro sistema debemos de ser muy cuidadosos, por el mismo hecho de reducir de 24 voltios a 5 voltios, entre la entrada del regulador y la salida del regulador, definitivamente que los tradicionales 7805 no nos servirían por la gran diferencia de reducción de voltaje, estos reguladores funcionan para un voltaje de entrada de 9VDC, es por eso que la mejor opción para estos casos es utilizar un regulador step-down (switching), que se les puede introducir directamente tensiones de hasta 40VDC sin que afecte su rendimiento, hablamos del LM2575, cuyo datasheet nos recomienda usarlo en la configuración que se muestra Figura 3.3 como podrán ver son un diodo schottky y una bobina mas respecto al regulador lineal del 7805. A partir de este tensión de 5 V regulado se puede sin ningún problema reducir a un voltaje regulado de 3.3V mediante el regulador LM 1117-3.3.

Para los 12V regulados, lo hacemos directo desde los 24 VDC no estabilizado, otra vez el mismo problema de la diferencia de tensiones de entrada y salida del regulador, se podría hacer lo mismo que se hizo para la tensión de 5 voltios, porque el LM2575 viene para 3.3V, 5V, 12V, 15V y una versión para voltajes variables, pero en este caso ya no es tan crítico la diferencia, además solo lo utilizaremos para los selectores de monedas, que no necesitan de mucha corriente y no trabajaran mucho en nuestro sistema, es por eso que utilizamos el regulador lineal 7812, cuya configuración utilizada es como se muestra en su datasheet.

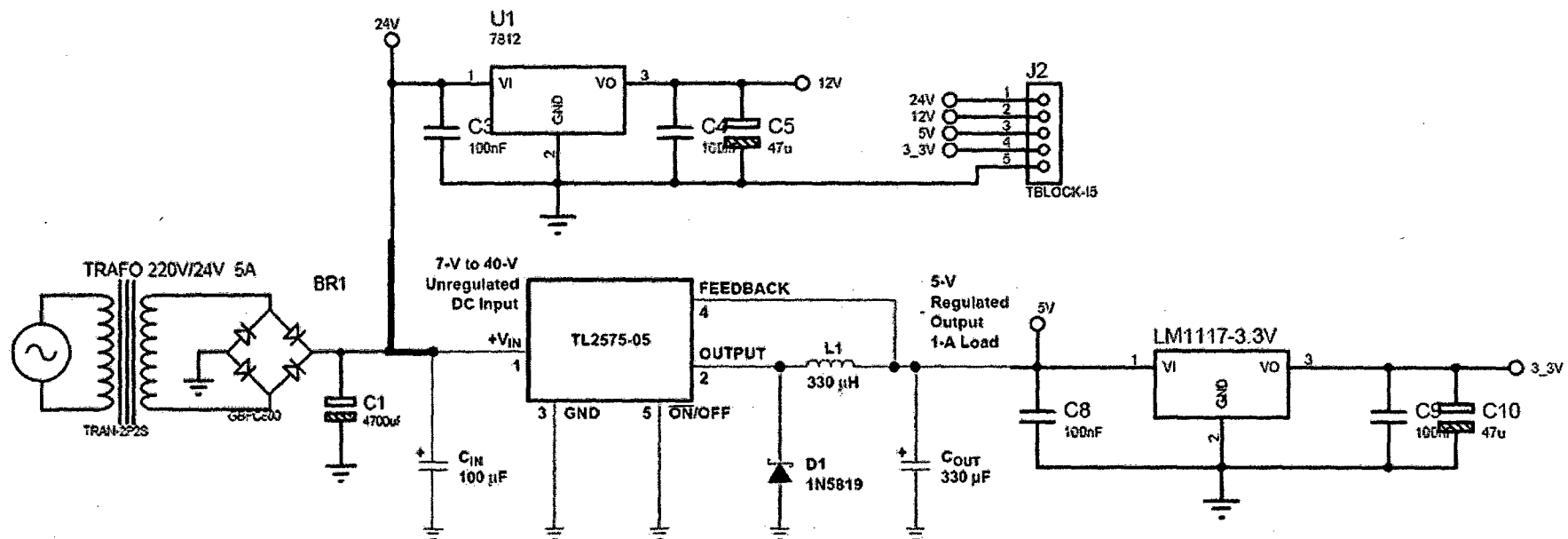


Figura 3.3 Circuito Fuente de Poder alimentación de todo el sistema.

### 3.3. Circuito de Control del Sistema

Esta parte del circuito del control que gobernará nuestro sistema lo vamos a realizar en dos partes, un circuito para agrandar los pines de nuestro dsPIC y en el otro donde irán sus conexiones necesarias para su funcionamiento, como su fuente de voltaje, el oscilador externo, y su conector al hardware de programación. Debido a que el dsPIC que vamos a utilizar es un chip muy pequeño de 64 pines y que requiere de cierta tecnología para poder crear su circuito impreso, lo hacemos por separado, lo único que hacemos es sacar los pines del microcontrolador a unos conectores más grandes y así poder trabajar sin complicaciones, tal como se muestra en la Figura 3.4.

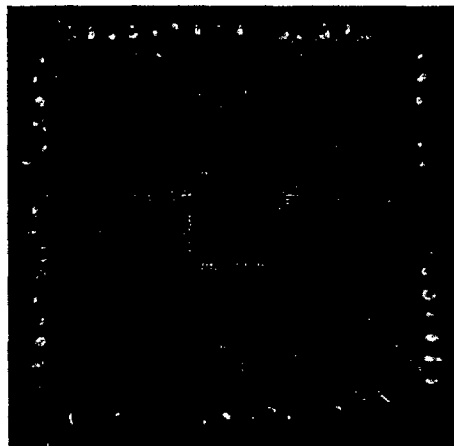


Figura 3.4 Circuito impreso del dsPIC, expansión a conectores más amplios

El circuito complementario donde irán sus conexiones necesarias para su correcto funcionamiento se observa en la Figura 3.5, donde muestra su conexión a la fuente de 3.3V, al cristal oscilador, al Vddcore y al conector para conectar al hardware de programación del dsPIC.

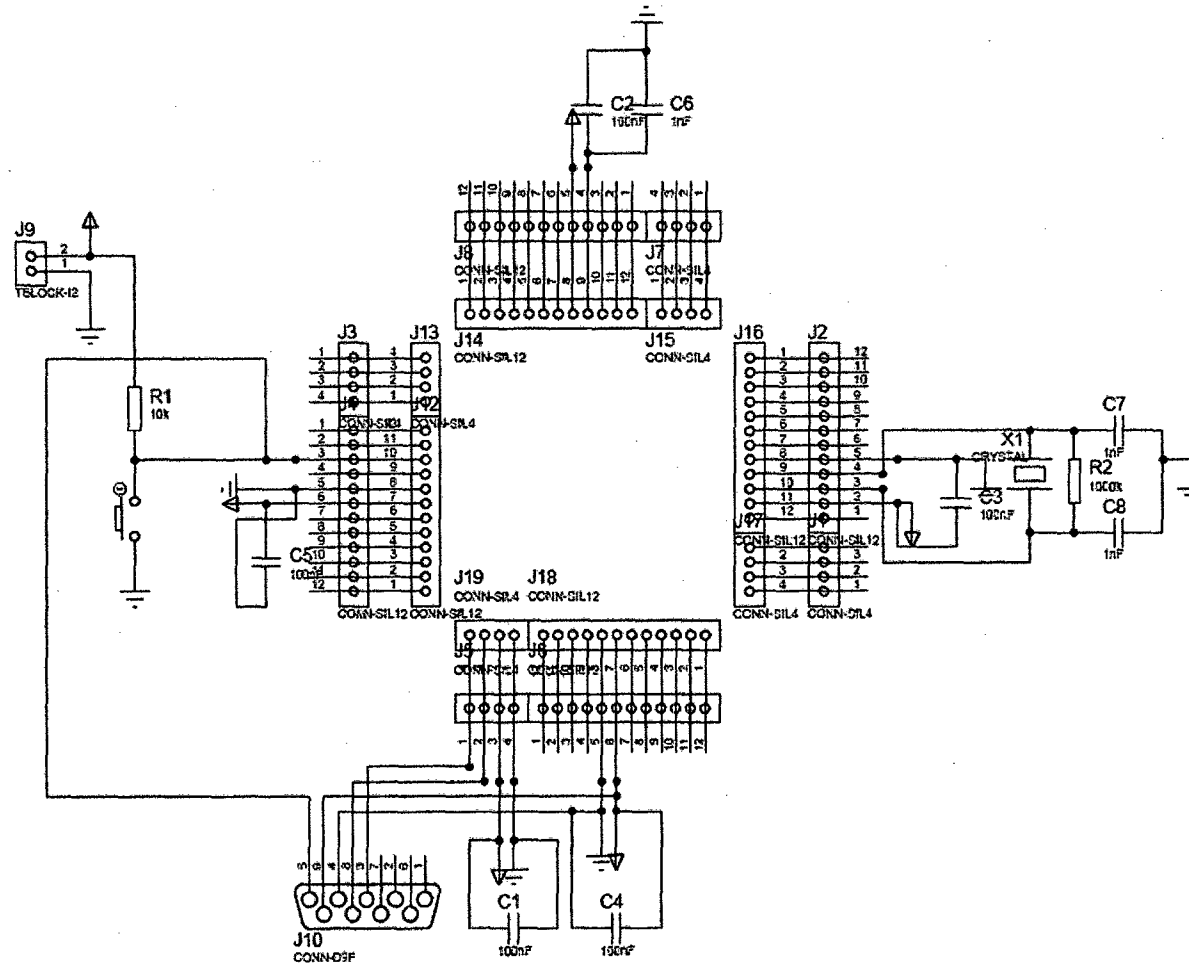


Figura 3.5 Circuito complementario para el correcto funcionamiento del dsPIC

### 3.4. Módulo MPLAB ICD3

Se necesita de un módulo grabador (quemador) del programa que controlará el sistema, para esto hay muchas alternativas, pero microchip tiene uno que es el más sofisticado, el módulo MPLAB ICD3, que puede programar desde el controlador más pequeño (gama baja) hasta lo nuevo que está sacando Microchip, el entorno de desarrollo de los MCU de Microchip es el MPLAB IDE.

El MPLAB ICD3 trabaja en dos modos: como Depurador o como programador. En modo Depurador, ICD3 monitorea las variables dentro del dsPIC y puede iniciar, detener, reiniciar y reconfigurar el microcontrolador. Lo más importante es cuando se detiene en modo Depurador, el ICD3 puede presentarnos los valores existentes de los registros del sistema, esta información es relevante en caso de que sea difícil encontrar errores en programación. En modo Programador el ICD3 graba (quema) el programa en la memoria FLASH del dsPIC, luego desconectas el ICD3 y presionas el botón RESET para que empiece a trabajar tu programa grabado en el dsPIC.

La conexión del ICD3 es muy simple, este grabador tiene un conector RJ11 macho de 6 pines para poder grabar un programa en cualquier microcontrolador de Microchip, entonces para poder grabar un dsPIC es necesario poner un conector RJ11 hembra en la tarjeta de control donde irá el dsPIC y conectar sus pines de este conector a los pines del dsPIC tal como se observa en la Figura 3.6.



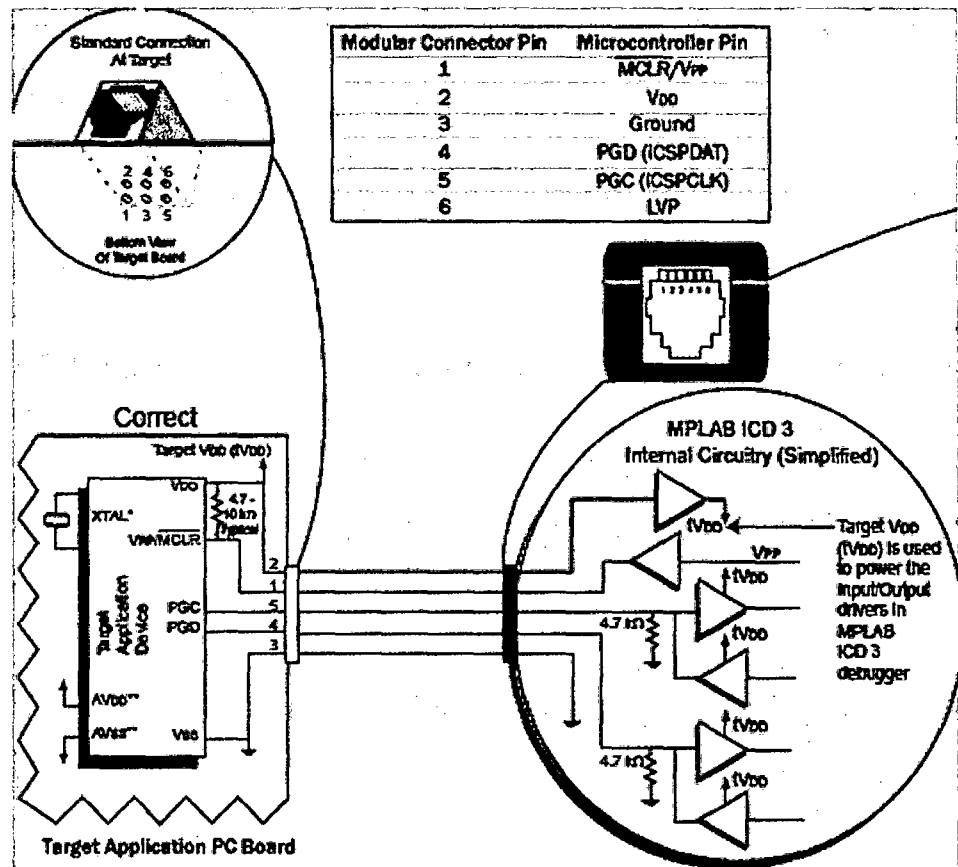


Figura 3.6 Conexión del ICD3 a los pines del dsPIC

En la Figura 3.7 se observa el conexionado de los pines del conector RJ11 hembra a los pines de los dsPIC, para que se inserte el conector RJ11 macho del ICD3, pero en realidad lo que se muestra no es un terminal RJ-11 sino es el terminal DB9 hembra, lo que pasa es que en el Proteus no hay una librería para el RJ-11, pero los pines del DB9 son iguales en espaciado de pines del RJ-11, quiere decir que podemos utilizarlo en caso de hacer su PCB en el Ares del Proteus.

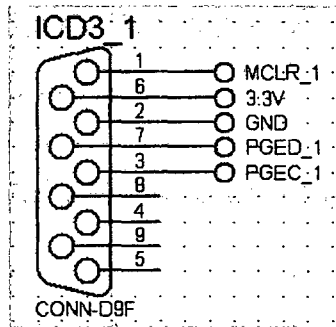


Figura 3.7 Conectores hembra para insertar el terminal del ICD3

El primer pin se conecta al MCLR del dsPIC, pin2 va a los 3.3V de la fuente de poder, el pin 3 va a la tierra de la fuente de poder, los pines 4 y 5 que son PGED y PGEC pueden conectarse a los pines PGED1 y PGEC1; PGED2 y PGEC2; PGED3 y PGEC3 del dsPIC33FJ64GS606, esto depende de la configuración en el software de programación y la necesidad de utilizar los distintos pines para distintas funciones, en nuestro caso vamos a utilizar los pines PGED1 y PGEC1 del dsPIC, como se muestra en la Figura 3.5.

### 3.5. Entradas y Salidas Digitales

Este sistema solamente será controlado digitalmente, tendremos entradas digitales, como pulsadores, switches, sensores capacitivos, etc. Como salidas tendremos señales de control de motores paso a paso, LCD, encendido de licuadora, encendido de electroválvula, alarmas, etc. Pero en el sistema absolutamente todo el control es digital.

### 3.5.1 Entradas Digitales

Vamos a mostrar todos los circuitos de las entradas digitales que tenemos, sus funciones y a que pin del dsPIC lo vamos a asignar, esto lo mostramos en la Tabla 3.1.

Tabla 3.1 Entradas digitales del sistema

| Dispositivo       | Función                           | Abreviación | pin  |
|-------------------|-----------------------------------|-------------|------|
| pulsador          | selección jugo surtido            | J1          | RB0  |
| pulsador          | selección jugo papaya             | J2          | RB1  |
| pulsador          | selección jugo piña               | J3          | RB2  |
| pulsador          | selección jugo manzana            | J4          | RB3  |
| pulsador          | selección jugo plátano            | J5          | RB4  |
| switch            | selección manual/automático       | M/A         | RB5  |
| switch            | encender electroválvula manual    | EV_M        | RG9  |
| monedero          | monedero de 2 soles               | C           | RE7  |
| monedero          | monedero de 1 sol                 | B           | RE6  |
| monedero          | monedero de 0.5 soles             | A           | RE5  |
| sensor capacitivo | reconocer si falta vaso           | FV          | RG8  |
| termistor         | mide temperatura de licuadora     | TL          | RD11 |
| sensor capacitivo | reconocer vaso listo para servido | LV          | RG6  |

### Conexión de los Pulsadores y los Interruptores

Para conectar un pulsador o un interruptor, estos se conectan con una resistencia en serie de un valor grande para no hacer circular una corriente grande ya que esto no hace falta. Se conectará al pin RB0 del dsPIC, que previamente debe haber sido programado como entrada, desde la unión del pulsador J1 y la resistencia R2 como se muestra en la Figura 3.8, previamente también hay una resistencia R1 de valor pequeño, pero no es necesario porque

los pines de entrada tienen una alta impedancia y por la tanto un buen aislamiento, pero siempre es prudente limitar la corriente para prevenir algún daño accidental. El valor de la resistencia R2 no es crítico podría variar de 1K hasta 100Kohms, se prefiere un valor grande para no hacer circular una corriente grande cuando el pulsador está cerrado.

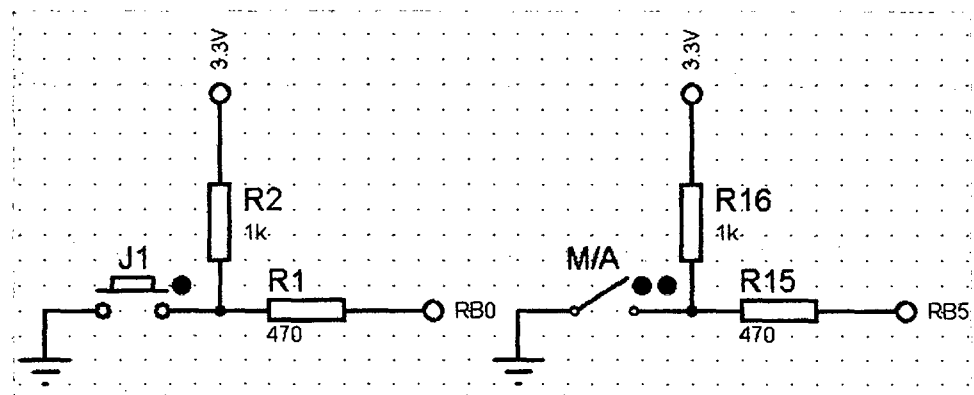


Figura 3.8 Conexionado de pulsadores e interruptores, entradas digitales

En el caso de la figura cuando se apriete el pulsador aparecerá en la entrada del microcontrolador un voltaje de cero voltios, quiere decir que el dsPIC verá interiormente en el registro del puerto un 0 lógico y un 1 cuando se suelte y la entrada esté a 3.3 voltios (1 lógico para los dsPIC).

En el caso de los interruptores es lo mismo en cuanto a conexionado pero tiene distinta aplicación, mientras este presionado presenta un 0 lógico hasta que los cambias de estado y presentan un 1 lógico.

### Conexión salida del selector de monedas al dsPIC

En el mercado se encuentran varios tipos de selectores de monedas, algunos que reconocen varias monedas, otros que además dan vuelto, y otros que solo reconocen una sola moneda de acuerdo a una moneda modelo que se les inserta para que haga de muestra (selector de monedas comparadores), en nuestro caso vamos a utilizar estos últimos selectores de monedas comparadores, los que reconoce una sola moneda, esto debido al bajo costo de estos selectores, pero utilizaremos tres de estos selectores de moneda, uno para 2 nuevos soles, otro para un nuevo sol y otro para 50 céntimos de sol.

Veamos paso a paso primeramente como es este selector de monedas y como funciona, para luego mostrar como lo vamos a utilizar en nuestro sistema.

#### PASO 1

Se coloca la moneda modelo, tal como se muestra en la Figura 3.9, que es la moneda que solo va a reconocer el selector.

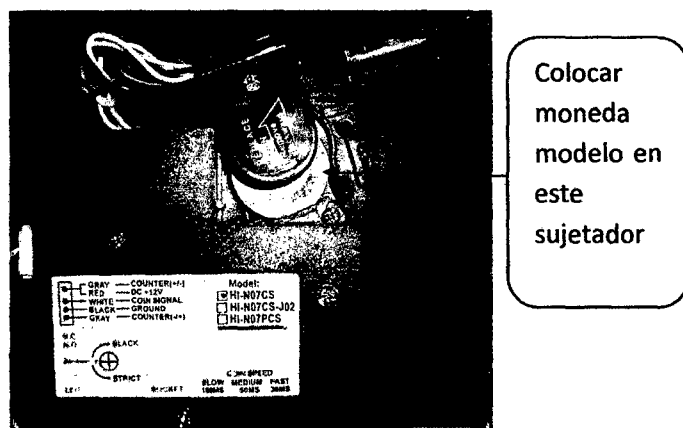


Figura 3.9 moneda modelo colocado en si respectivo sujetador

Los siguientes pasos se muestran en la siguiente Figura 3.10.

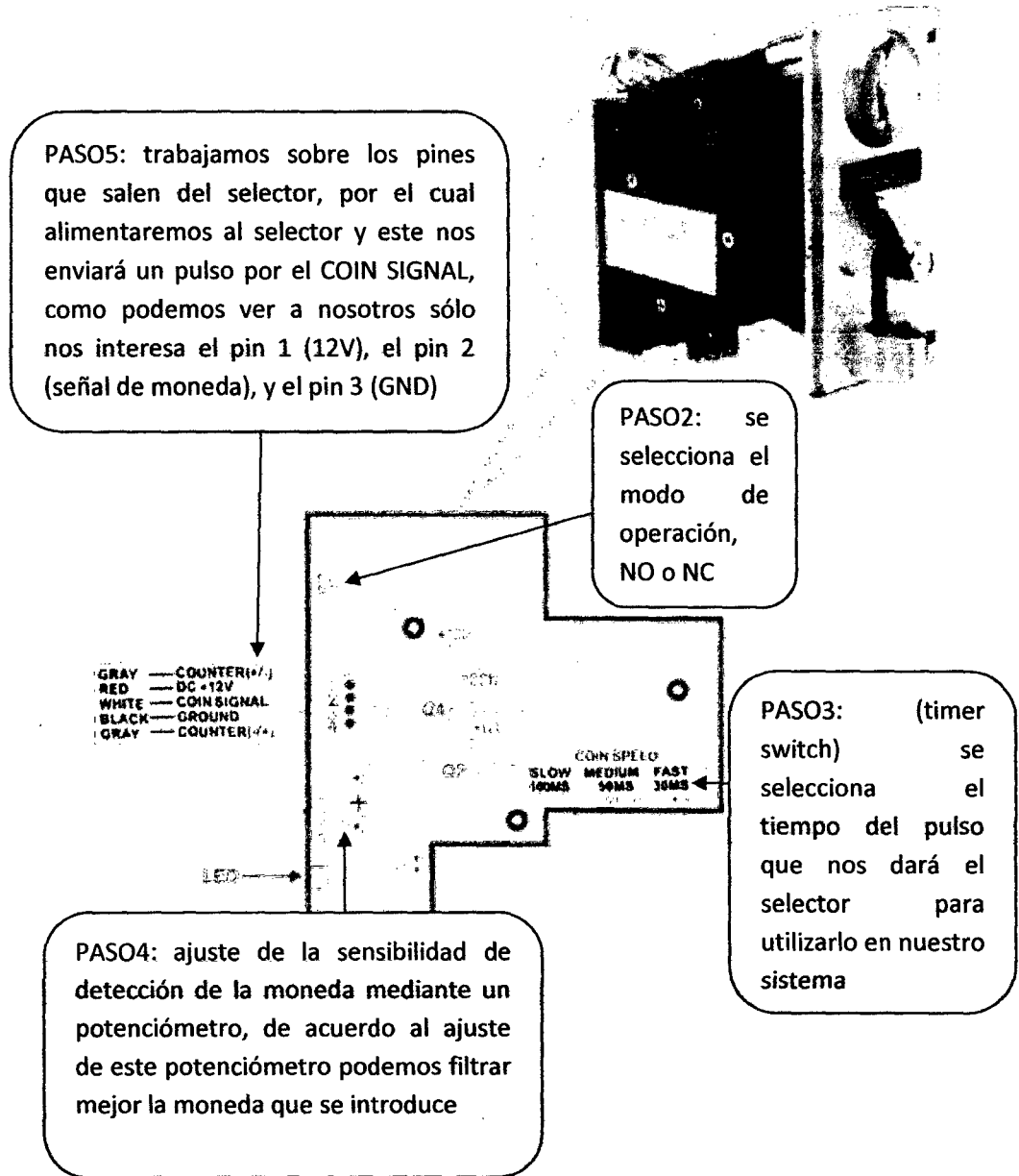


Figura 3.10 Modo de operación del selector de moneda comparador

Ahora nosotros vamos a trabajar sobre el paso 5, donde lo único que nos interesa es la alimentación de 12V al selector, la señal de moneda ingresada y el GND, el cual lo vamos a trabajar tal como se observa en la Figura 3.11,

donde podemos apreciar el circuito que utilizamos para adquirir esa señal de moneda y utilizarlo en nuestro sistema, como podrán ver en la figura 3.10, la señal de moneda es el colector del transistor Q4, entonces cuando el selector reconoce una moneda, en el colector de Q4 obtenemos 0V (SEÑAL MONEDA), es en este instante que se activa el relay RL1, para enviar un cero lógico a nuestra entrada del dsPIC, como podrán observar es similar a nuestros circuitos de pulsadores e interruptores, en este caso el relay hace de pulsador, D1 es un diodo que se encarga de eliminar las corrientes inversas que aparecen al activarse las bobinas de relay.

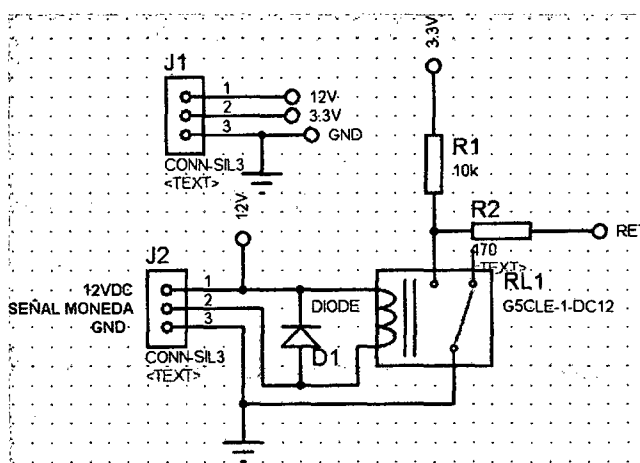


Figura 3.11 Circuito de adaptación de la salida del selector a nuestro dsPIC

### Conexión de los sensores capacitivos

Para nuestro sistema se va a requerir de dos sensores capacitivos, uno para detectar la presencia de los vasos en el punto de servicio del jugo y el otro para detectar si hay vasos o no en el sistema para su respectivo traslado al punto de servicio de los vasos. Por lo tanto son dos entradas para nuestros dsPIC, estos sensores capacitivos se alimentan con un voltaje de 24 VDC y

tienen una salida NPN NA (normalmente abierta), como se muestra en la Figura 3.12.

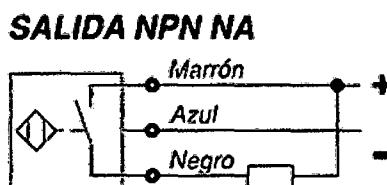


Figura 3.12 conexión sensores capacitivos

Pero como nuestro dsPIC trabaja con 3.3V no podemos utilizarlo directo como si fuera un pulsador, entonces nos tenemos que ayudar de un dispositivo para aislarlo y hacerlo operar como un pulsador, para esto podemos utilizar un relay o un optoacoplador, como trabaja a 24 voltios es suficiente trabajar con un relay, conectamos el relay en serie con el NA del sensor y la salida del relay lo utilizamos como un circuito de un pulsador, como se observa en la Figura 3.13. Simulamos nuestro sensor capacitivo como si fuera un switch (FV, LV), porque eso es lo que hace al detectar cualquier material, automáticamente se cierra el contacto que estaba normalmente abierto. En el ANEXO E se muestra un catálogo de estos sensores capacitivos.

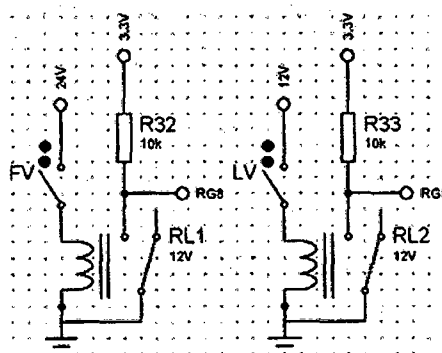


Figura 3.13 Conexión de sensores capacitivos a los dsPIC



### **Conexión de sensor de temperatura**

Es necesaria la medición de la temperatura en la licuadora debido a que la licuadora por mucho uso llegue a malograrse, desgastarse el carbón, desgastarse la armadura del rotor, etc. Una manera de detectar que la licuadora está a punto de malograrse es que se empieza a calentar toda la armadura del estator y su disipador hasta una temperatura alta promedio de 70°C, según recomendación de los técnicos electricistas. Esta temperatura nosotros la podemos medir y utilizarla para detener nuestro sistema y poder hacer un mantenimiento de la licuadora o en todo caso cambiarla.

Hay muchas maneras de medir la temperatura pero para pequeños rangos es suficiente un termistor, claro que para un termistor se tendría que, filtrar la señal, amplificarla y luego calibrarla, pero hay maneras de utilizarla, una de ellas es utilizarla en un divisor de tensión y medir tensión analógica, otra es utilizarlo con un comparador para que a una determinada temperatura active un relay y nos dé una utilidad de control de temperatura, para nuestro sistema vamos a utilizar este tipo de circuito, donde al final a determinada temperatura se active un relay y la salida de este relay lo utilizamos como si fuera un interruptor para activar nuestra entrada del dsPIC.

Un termistor es un sensor resistivo de temperatura su funcionamiento se basa en la variación de la resistividad que presenta un semiconductor con la temperatura. Existen dos tipos de termistores: NTC (coeficiente de temperatura negativo) y PTC (coeficiente de temperatura positivo).

Para los termistores NTC, al aumentar la temperatura, aumentará también la concentración de portadores, por lo que la resistencia será menor, de ahí que el coeficiente sea negativo. Para los termistores PTC, en el caso de un semiconductor con un dopado muy intenso, éste adquirirá propiedades metálicas, tomando un coeficiente positivo en un margen de temperatura limitado. En nuestro circuito vamos a utilizar un termistor NTC. Con la previa descripción de un termistor, podemos analizar el circuito que se muestra en la Figura 3.14.

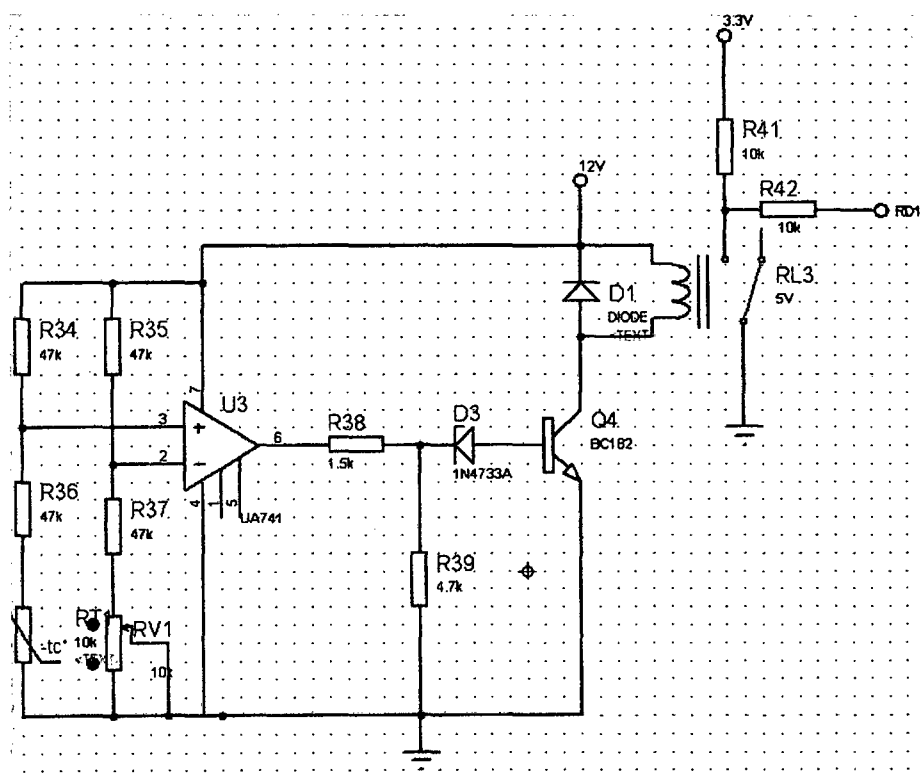


Figura 3.14 conexionado termistor modo comparador salida relay

Este circuito requiere de una alimentación de 12VDC, el cual nos proporciona nuestra fuente de poder. La primera etapa del circuito,

constituida por los divisores de voltaje formados por las resistencias R34 a R37; potenciómetro RV1 y el termistor, componen el detector de temperatura.

Como podemos apreciar, las resistencias R34 a R37 son del mismo valor, dejando la posible variación en el valor de salida (hacia los comparadores) al potenciómetro y al termistor. El potenciómetro RV1 sirve para ajustar el valor de referencia para que el termistor funcione a su temperatura en frío. Este valor del potenciómetro es susceptible de cambio, ya que el termistor puede tener un valor en frío tal, que el potenciómetro no alcance a compensar este valor de resistencia.

Con algunos cálculos sencillos podemos determinar, por ejemplo, si aumenta la resistencia de potenciómetro, el voltaje de referencia aumenta; por el contrario, si la resistencia disminuye, también lo hace el voltaje de referencia. El termistor tiene un coeficiente negativo de resistencia, lo cual indica que a mayor temperatura, menor resistencia. Con esto podemos concluir que si aumenta la temperatura, disminuye la resistencia y el voltaje de referencia y si, por el contrario, disminuye la temperatura, la resistencia aumentará provocando también un incremento en el voltaje de referencia (terminal negativa del comparador).

Inicialmente este circuito se encuentra a temperatura normal; la salida del comparador es cero, esto es, tiene un nivel de salida BAJO. Al momento en que se incrementa la temperatura, el valor en la entrada del comparador varía, sobre todo en la entrada de referencia positiva que es el que está conectado al termistor. Esta variación de temperatura provoca una conmutación en la

salida del comparador (el intervalo para que se dé la conmutación se ajusta con el potenciómetro RV1).

La siguiente etapa utiliza la salida del comparador para trasladar esta conmutación hacia un relay para utilizarlos como switch.

Cuando la salida del comparador es BAJA, el diodo zener no se polariza, por lo que el transistor Q4 está apagado manteniendo la salida de la etapa en alto (1 lógico).

Por el contrario, si la salida del comparador es ALTA, el diodo zener se polariza, con lo que provoca que el transistor llegue a su estado de saturación y por lo tanto de conducción.

Esta configuración del transistor es un inversor; debido a esto, la salida es virtualmente conectada a tierra, así que ésta tiene un estado BAJO (0 lógico).

El diodo zener se utiliza para garantizar que el transistor se mantenga en los estados de saturación y corte, es decir, para estar en función de encendido y apagado.

Esta salida como ya analizamos, es la salida inversa del comparador, es la que activa el relay para que trabaje como un switch y sea una entrada a nuestro dsPIC, que al activarse este sensor, parara el sistema por si pueda ocurrir alguna falla con la licuadora.

Hasta ahora hemos visto todas las entradas que tendrá nuestro sistema, ahora empezaremos a analizar todas las salidas del sistema.

### 3.5.2 Salidas Digitales

Vamos a mostrar el conexionado de todas las salidas digitales que actuaran para hacer trabajar al sistema, pero antes mostraremos qué salidas digitales tenemos, la función de cada salida y a que pin de nuestro dsPIC le asignamos, esto se muestra en la Tabla 3.2.

Tabla 3.2 salidas digitales del sistema

| Dispositivo | Función                             | Abreviación | pin  |
|-------------|-------------------------------------|-------------|------|
| LCD         | transferencia de información        | D0          | RD0  |
| LCD         | transferencia de información        | D1          | RD1  |
| LCD         | transferencia de información        | D2          | RD2  |
| LCD         | transferencia de información        | D3          | RD3  |
| LCD         | transferencia de información        | D4          | RD4  |
| LCD         | transferencia de información        | D5          | RD5  |
| LCD         | transferencia de información        | D6          | RD6  |
| LCD         | transferencia de información        | D7          | RD7  |
| LCD         | selección de registro control/datos | RS          | RD9  |
| LCD         | señal de lectura/escritura          | R/W         | RD8  |
| LCD         | señal activación modulo LCD         | E           | RD10 |
| motor 1     | señal giro motor 1                  | C-M1        | RF0  |
| motor 1     | señal ENABLE motor 1                | E-M1        | RF1  |
| motor 2     | señal giro motor 2                  | C-M2        | RE0  |
| motor 2     | señal ENABLE motor 2                | E-M2        | RE1  |
| motor 3     | señal giro motor 3                  | C-M3        | RC13 |
| motor 3     | señal ENABLE motor 3                | E-M3        | RC14 |
| motor 4     | señal giro motor 4                  | C-M4        | RC12 |
| motor 4     | señal ENABLE motor 4                | E-M4        | RC15 |
| motor 5     | señal giro motor 5                  | C-M5        | RG2  |
| motor 5     | señal ENABLE motor 5                | E-M5        | RG3  |
| motor 6     | señal giro motor 6                  | C-M6        | RE2  |
| motor 6     | señal ENABLE motor 6                | E-M6        | RE3  |
| motor 6     | señal dirección motor 6             | D-M6        | RE4  |
| motor 7     | señal giro motor 7                  | C-M7        | RF3  |
| motor 7     | señal ENABLE motor 7                | E-M7        | RF2  |
| motor 7     | señal dirección motor 7             | D-M7        | RF6  |
| motor 8     | señal giro motor 8                  | C-M8        | RF4  |

|                |                              |      |      |
|----------------|------------------------------|------|------|
| motor 8        | señal ENABLE motor 8         | E-M8 | RF5  |
| motor 8        | señal dirección motor 8      | D-M8 | RG7  |
| licuadora      | encendido licuadora          | EL   | RB8  |
| electroválvula | encendido electroválvula     | EEV  | RB9  |
| alarma         | encendido alarma por fallas  | A    | RB10 |
| Motor fruta 1  | para asegurar caída de fruta | MF1  | RB11 |
| Motor fruta 2  | para asegurar caída de fruta | MF2  | RB12 |
| Motor fruta 3  | para asegurar caída de fruta | MF3  | RB13 |
| Motor fruta 4  | para asegurar caída de fruta | MF4  | RB14 |
| Motor fruta 5  | para asegurar caída de fruta | MF5  | RB15 |

### Conexión Visualizador LCD

Las pantallas de cristal líquido LCD o display LCD para mensajes tienen la capacidad de mostrar cualquier carácter alfanumérico, permitiendo representar la información que genera cualquier equipo electrónico de una forma fácil y económica.

La Figura 3.15 muestra esta pantalla que consta de una matriz de caracteres distribuidos en dos líneas de 16 caracteres cada línea. El proceso de visualización es gobernado por un microcontrolador incorporado a la pantalla, siendo el Hitachi 44780 el modelo de controlador más utilizado.

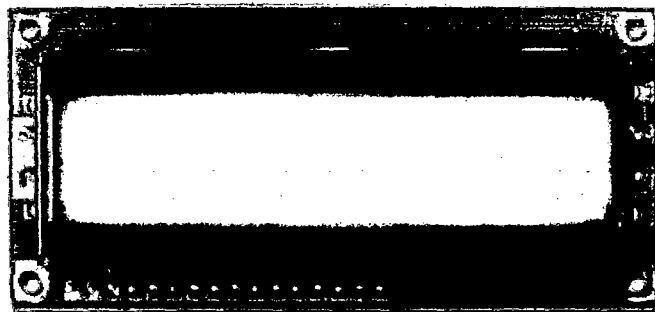


Figura 3.15 LCD 2x16: está compuesto por dos líneas de 16 caracteres

Este LCD será para la interacción con el cliente, donde se mostrará el precio del jugo, el jugo que desea elegir, las fallas del sistema, etc.

El conexionado con el dsPIC es directo de las salidas del MCU, excepto la alimentación del LCD. A continuación mostraremos las señales que emplea el LCD, así como el número de pin a la que corresponde.

| PIN N° | SÍMBOLO         | DESCRIPCIÓN  |
|--------|-----------------|--|
| 1      | V <sub>SS</sub> | Patilla de tierra de alimentación  |
| 2      | V <sub>DD</sub> | Patilla de alimentación de 5 V   |
| 3      | V <sub>O</sub>  | Patilla de contraste del cristal líquido. Normalmente se conecta a un potenciómetro a través del cual se aplica una tensión variable entre 0 y +5V que permite regular el contraste del cristal líquido. |
| 4      | RS              | Selección del registro de control/registro de datos:<br>RS=0 Selección del registro de control<br>RS=1 Selección del registro de datos   |
| 5      | R/W             | Señal de lectura/escritura<br>R/W=0 El módulo LCD es escrito<br>R/W=1 El módulo LCD es leído   |
| 6      | E               | Señal de activación del módulo LCD:<br>E=0 Módulo desconectado<br>E=1 Módulo conectado   |
| 7-14   | D0-D7           | Bus de datos bi-direccional. A través de estas líneas se realiza la transferencia de información entre el módulo LCD y el sistema informático que lo gestiona  |

En la Figura 3.16 se muestra en conexionado del LCD al dsPIC.

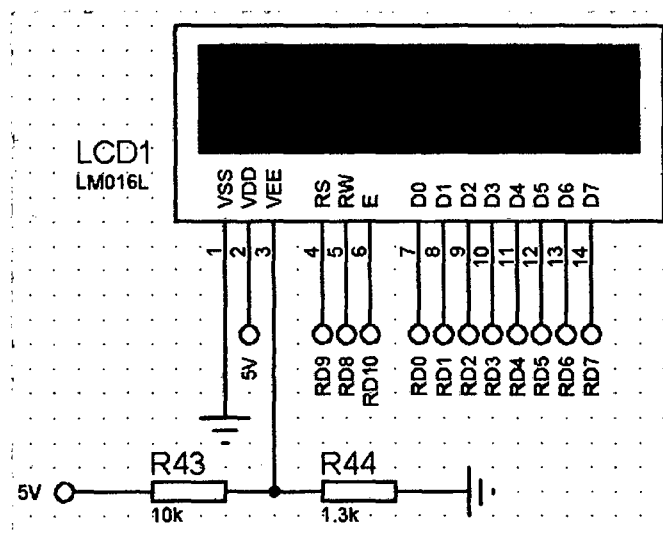


Figura 3.16 conexionado visualizador LCD al dsPIC

### **Conexión de las señales de control de los motores paso a paso**

Hay muchas formas de controlar motores paso a paso, existen algunos drivers como el driver de doble puente H el L298, que con el controlador de motores L297 hacen que el control de motores sea sencillo, además pueden trabajar con motores de hasta 2 amperios por fase, es por eso que elegimos usar estos dos dispositivos para controlar los motores, mas adelante en la sección 3.6 les mostraremos el circuito que controla los motores, por ahora debemos saber que para controlar el giro de los motores paso a paso, solo necesitamos tres señales: un pulso de reloj, una señal de 0 o 5 voltios para el sentido de giro y una señal de habilitación del controlador del motor, solo con esas tres señales vamos a trabajar para controlar los motores paso a paso.

En los 5 motores que controlan los pistones de corte de las frutas, solo se requiere la habilitación del controlador y los pulsos de reloj para girar los motores paso a paso, porque siempre da una vuelta, mientras que en los motores de giro de licuadora, del mecanismo de separación y posicionado de vasos si necesitan aparte de la habilitación del motor y los pulsos de reloj una señal para el sentido de giro.

Los pulsos de reloj, la señal de sentido de giro y la habilitación del controlador nos van a dar nuestras salidas digitales del dsPIC como se muestra en la Tabla 3.2, pero estas salidas tienen un voltaje de 3.3V, tenemos que acondicionarle un circuito para trabajar con 5 voltios, porque con este voltaje trabaja el circuito que controla los motores.



En este circuito acondicionado, para cada salida lleva un transistor, para que trabaje en colector abierto en modo de saturación, el cual significa que va a trabajar como un interruptor controlado o un relay eléctrico. Además se sabe que al polarizar positivamente la unión base emisor en saturación, la salida en el colector es aproximadamente cero, por tal motivo nosotros para utilizar esta señal con nuestro controlador de motores paso a paso lo invertimos mediante el 74HC14. En la Figura 3.17 se muestra el circuito que servirá para acondicionar nuestras salidas del dsPIC hacia nuestros controladores de motores paso a paso.

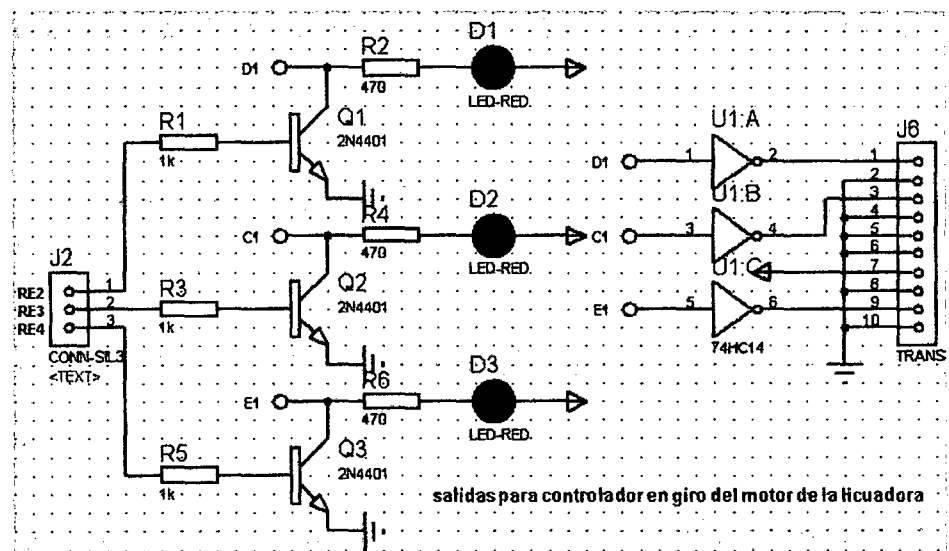


Figura 3.17 Acondicionador salidas digitales dsPIC, para controlar motor pap.

Se debe evitar manejar demasiada corriente para cualquier pin de salida a pesar de que el dsPIC es capaz de suministrar hasta 25mA por cada uno de sus pines. Siempre se debe limitar la corriente de los pines de salida al mínimo necesario, a fin de evitar problemas en la disipación de calor del

microcontrolador. Por eso se usa un transistor de colector abierto y una resistencia limitador de corriente, como Q1 y R1, para así lograr un buen aislamiento entre la fuente de poder y el circuito de control.

Este tipo de acondicionador va a ser igual para todas las salidas digitales que controlan los motores paso a paso (señal de reloj, dirección y habilitación controlador motor paso a paso).

### **Conexión encendido de licuadora, electroválvula, motores para asegurar caída de frutas y alarma**

Tenemos varios dispositivos que funcionan a 220VAC, pero que tienen que ser controlados desde el dsPIC, que trabaja 3,3VDC, entonces necesitamos un dispositivo electrónico que nos aisle eléctricamente las dos conexiones, y que mejor que un relay en paralelo con un diodo para hacer esta función, pero para activar el relay es necesario mínimo un voltaje de 5 voltios o 12 voltios, entonces para esto utilizamos un transistor en colector abierto y una resistencia limitador de corriente después del pin que controla el encendido de estas máquinas, como se muestra en la Figura 3.18.

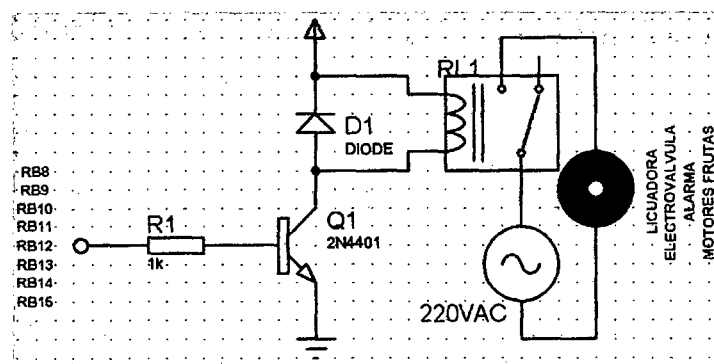


Figura 3.18 Conexión licuadora, electroválvula, alarma, motores AC

### **3.6. Motores Paso a Paso**

Los motores paso a paso son ideales para la construcción de mecanismos en donde se requieren movimientos muy precisos, como son los que requerimos para nuestro sistema. La característica principal de estos motores es el hecho de poder moverlos un paso a la vez por cada pulso que se le aplique. Este paso puede variar desde  $90^\circ$  hasta pequeños movimientos de tan solo  $1.8^\circ$ , es decir, que se necesitarán 4 pasos en el primer caso ( $90^\circ$ ) y 200 para el segundo caso ( $1.8^\circ$ ), para completar un giro completo de  $360^\circ$ .

Estos motores poseen la habilidad de poder quedar enclavados en una posición o bien totalmente libres. Si una o más de sus bobinas están energizadas, el motor estará enclavado en la posición correspondiente y por el contrario quedará completamente libre si no circula corriente por ninguna de sus bobinas.

#### **3.6.1. Tipos y Principio de Funcionamiento de los Motores Paso a Paso**

Los motores paso a paso pueden ser de tipo bipolar o unipolar.

Los motores bipolares son llamados así porque para hacer girar el eje, deben invertir la polaridad de la alimentación de sus bobinas, según una secuencia bien precisa. Estos motores se reconocen fácilmente por 4 hilos que salen de su cuerpo, debido al par de bobinas que no tienen toma central, como se observa en la siguiente Figura 3.19.

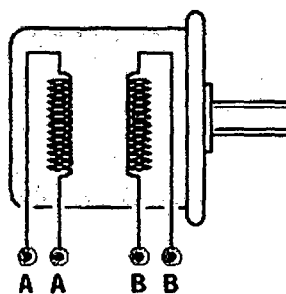


Figura 3.19 Motores bipolares

Los motores unipolares son llamados así porque, al tener un doble arrollamiento en sus bobinas, no es necesario invertir la polaridad de la alimentación. Estos motores se reconocen por los 5 o 6 hilos que salen de su cuerpo, cuando salen 6 hilos, quiere decir que el par de bobinas tienen una toma central, cada una; cuando salen 5 hilos las bobinas A y B están conectadas entre ellas, directamente en su interior por una sola toma. Como se observa en la siguiente Figura 3.20.

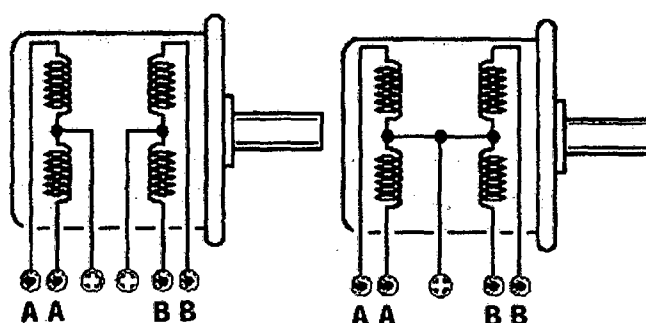


Figura 3.20 Motores unipolares

Los motores bipolares son más eficientes, debido a que, a igual potencia, tienen unas dimensiones menores que los unipolares, es por esta razón, por la que se prefiere utilizar los motores bipolares, porque suponen menor problema de espacio en el interior de cualquier dispositivo.

El primer problema al que se enfrenta uno que quiere utilizar estos motores, es al control de un motor paso a paso. Aún teniendo un motor con tan solo 4 hilos, frente a los 5 o 6 hilos, muchos ignoran como conectarlos y con qué tensión alimentarlos. En la mayoría de los casos, se encuentra en el cuerpo de motor etiquetas con datos incomprensibles, pero raramente, el valor de la tensión de trabajo ni la corriente máxima admisible ya que, la corriente varia al mismo tiempo que la tensión de alimentación.

Los motores paso a paso pueden ser alimentados no importa con que tensión, es decir 9, 12, 15, 18, 24 o 28 voltios, a condición que el circuito de control disponga de un sistema que limite la corriente que circula en sus bobinas. Sin este control, corrientes muy elevados atravesarán los arrollamientos de las bobinas, en este caso no solo el circuito de control se dañará rápidamente, sino que simultáneamente los arrollamientos de las bobinas del motor se calentarán en exceso y terminarán dañándose.

Si tenemos un motor en el que la tensión de alimentación es de 12 voltios, preparado para absorber una corriente máxima de 0,3 amperios, y medimos el valor óhmico de sus bobinas, podremos comprobar que esta es de 2,5 Ohm. Si este motor es alimentado con una tensión de 12 voltios sin limitación de corriente, según la ley de Ohm bien conocida:

$$\text{Corriente (amperios)} = \text{Voltaje (voltios)} / \text{resistencia (ohm)}$$

Tendremos que por las bobinas circulará una corriente de:

$$12 \text{ voltios} / 2,5 \text{ Ohm} = 4,8 \text{ amperios.}$$

Si a continuación lo alimentamos con una tensión de 24 voltios, la corriente aumentará de esta manera:

$$24 \text{ voltios} / 2,5 \text{ Ohm} = 9,6 \text{ amperios}$$

Para evitar que el circuito controlador o las bobinas del motor (o ambos) puedan dañarse, es suficiente con limitar la corriente a un máximo de unos 0,3 amperios, siempre para un voltaje de 12 voltios, o limitar la corriente a un valor inferior, para una tensión de 24 voltios.

Si abrimos uno de estos motores, nos encontraremos, algo similar a una simple dinamo de bicicleta, un imán con múltiples polos, conectado al eje del motor (ver Figura 2.21-2), y un cierto número de bobinas excitadoras situadas sobre la pared del motor (ver Figura 2.21-1). En los motores bipolares siempre nos encontramos con 4 hilos, mientras que en los motores unipolares serán 5 o 6.

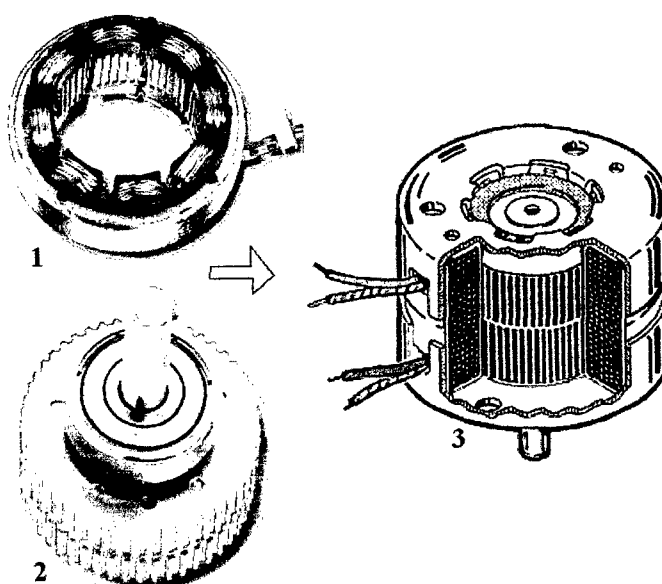


Figura 2.21 Estructura interna de un motor paso a paso

Para comprender aún mejor como se procede para hacer girar un paso al eje del motor, analicemos un motor teórico provisto de 4 bobinas excitadoras únicamente, excitadas por un imán.

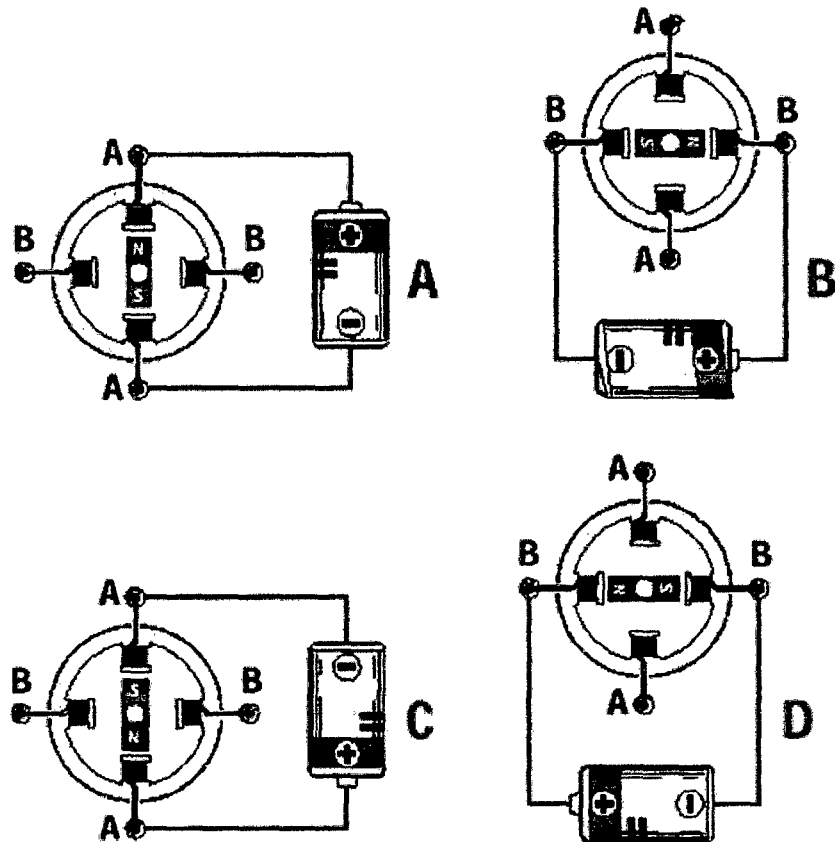


Figura 3.22 Secuencia y polaridad para hacer girar un paso al motor

Si aplicamos una tensión sobre las 2 bobinas A-A, el imán será atraído hacia sus 2 bobinas, y se obtendrá entonces la rotación de un paso (ver Figura 3.22A). Si retiramos la alimentación de las 2 bobinas A-A para aplicarla sobre las bobinas B-B, el imán será atraído hacia éstas 2 bobinas y se obtendrá de nuevo la rotación de otro paso (ver Figura 3.22B). Si aplicamos la tensión sobre las bobinas A-A, pero con una polaridad inversa, el imán será de nuevo atraído hacia estas 2 bobinas y se obtendrá de nuevo otro avance de

paso (ver Figura 3.22C). Para obtener todavía otro paso de rotación, se deberá aplicar una tensión de polaridad inversa sobre las bobinas B-B (ver figura 3.22D).

De manera que, para realizar un giro completo del eje del motor, tendremos que aplicar de manera secuencial una tensión sobre las bobinas A-A, luego sobre las bobinas B-B, luego de nuevo, pero con una polaridad inversa, sobre las bobinas A-A y sobre las bobinas B-B.

Cuanto mayor sea el número de imanes presentes sobre el rotor y la cantidad de bobinas presentes en el estator, mayor será el número de pasos necesarios para efectuar un giro completo del eje del motor.

Existen motores que ejecutan un giro completo con 20 pasos, otros con 48 pasos y otros incluso, con 100 ó 200 pasos.

Podemos encontrarnos a veces una etiqueta sobre el cuerpo del motor, sobre la que figura un número, por ejemplo 1,8°, 7,5°, 15°, etc. Indicando el número de grados que gira el eje en cada paso.

Conociendo el número de grados recorridos por el eje del motor a cada paso, podemos calcular cuantos pasos son necesarios para que el eje del motor efectúe un giro completo, utilizando la siguiente fórmula:

$$\text{Total de pasos} = 360 / \text{grados en cada paso}$$



Conociendo el número total de pasos necesarios para obtener un giro completo de eje del motor, podemos conocer el valor de la rotación de un paso en grados, utilizando la siguiente fórmula:

$$\text{Grados por paso} = 360 / \text{número total de pasos}$$

No todo el mundo sabe que los motores paso a paso pueden igualmente girar medio paso, si se aplica sobre las bobinas A-A y B-B una tensión según la secuencia de la Figura 3.23.

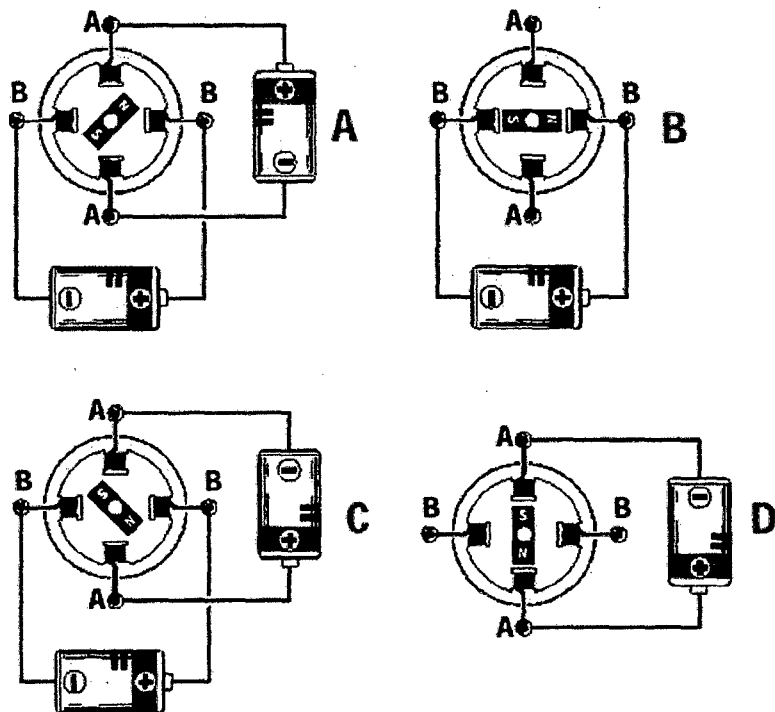


Figura 3.23 Secuencia y polaridad para hacer girar medio paso al motor

Para comprender como es posible hacer girar medio paso el eje del motor, observemos la Figura 3.23A.

Si aplicamos una tensión sobre las bobinas A-A y simultáneamente sobre las bobinas B-B, el imán se posicionará en el centro de las 2 bobinas A-B, y se obtendrá de esta manera medio paso de giro (ver Figura 3.23A).

Si retiramos la tensión de las bobinas A-A solamente, el imán será atraído hacia las 2 bobinas B-B, y se obtendrá de nuevo una rotación de otro medio paso (ver Figura 3.23B).

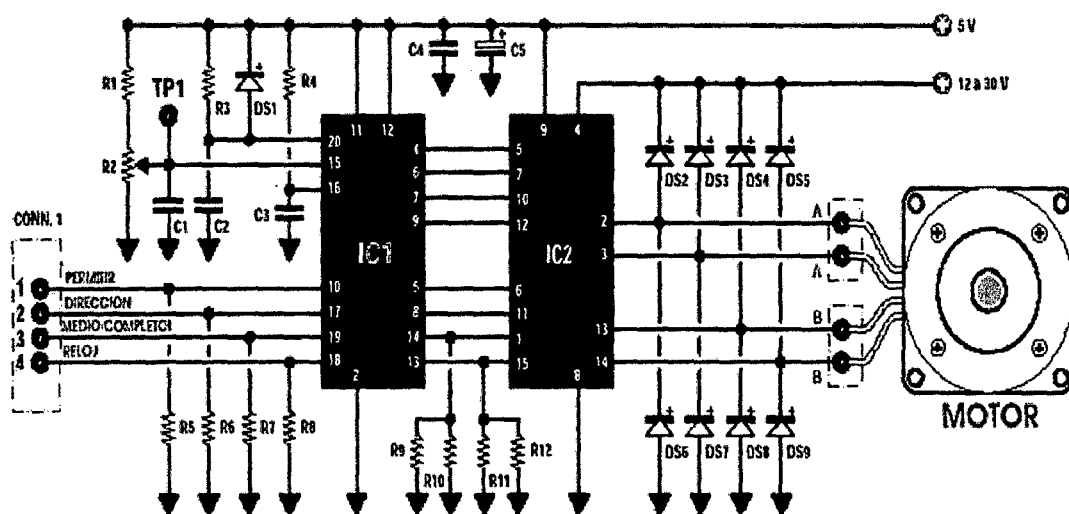
Si aplicamos de nuevo una tensión, esta vez con una polaridad inversa, sobre las bobinas A-A, el imán se posicionará en el centro de las bobinas B-A, y obtendremos entonces otro medio paso de rotación (ver Figura 3.23C).

Para obtener otro medio paso, será suficiente con retirar la tensión presente sobre las bobinas B-B (ver Figura 3.23D).

Hay varias formas de controlar los motores paso a paso. En el mercado se encuentran varios circuitos controladores, pero se deben de tener en cuenta, que torque va a manejar el motor, por lo tanto que amperaje por fase va a necesitar ese motor, de acuerdo a eso vamos a seleccionar los puentes H y los controladores de los motores, en nuestro caso de acuerdo a las secciones 2.2.2 y 2.2.3, se seleccionaron motores paso a paso con corrientes de fase de 1.5 y 2 amperios respectivamente, de acuerdo a esto les mostraremos los controladores y los puente H escogidos para el control de los motores.

### 3.6.2. Controlador de Motores Paso a Paso

El esquema eléctrico que podemos ver en la Figura 3.24, que utiliza dos integrados, IC1 e IC2, es el circuito de potencia que entrega en los bornes A-A y B-B las combinaciones necesarias para hacer girar cualquier tipo de motor paso a paso. Este esquema, será un módulo aparte del circuito de control, se harán 8 de estos esquemas para controlar los 8 motores.



#### Lista de componentes

|   |                      |
|---|----------------------|
| R1=   | 8k2                  |
| R2=   | 2k2_trimmer          |
| R3,R5,R6,R7,R8=   | 10k                  |
| R4=   | 22k                  |
| R9,R10,R11,R12=   | 10hm_0,5W            |
| C1,C2,C4 =  | 100nf_poliéster      |
| C3=   | 3n3_poliéster        |
| C5=   | 100uF_electrolítico  |
| DS1=  | diodo_1N4148         |
| DS2-DS9=  | diodo_Schottky_GI852 |
| IC2=  | L297                 |
| IC3=  | L298/IN              |
| * todas las resistencias (salvo indicación) de 1/4w 5%. |                      |

Figura 3.24 Esquema eléctrico de la etapa de potencia, lista de componentes

Comenzamos la explicación del funcionamiento con el esquema de Figura 3.24, el primer circuito integrado que nos encontramos es un L297, identificado como IC1. Este circuito integrado, gracias a su lógica interna (ver Figura 3.25), presenta sobre sus salidas 4 a 9 todas las combinaciones necesarias para manejar las parejas de las bobinas A-A y B-B.

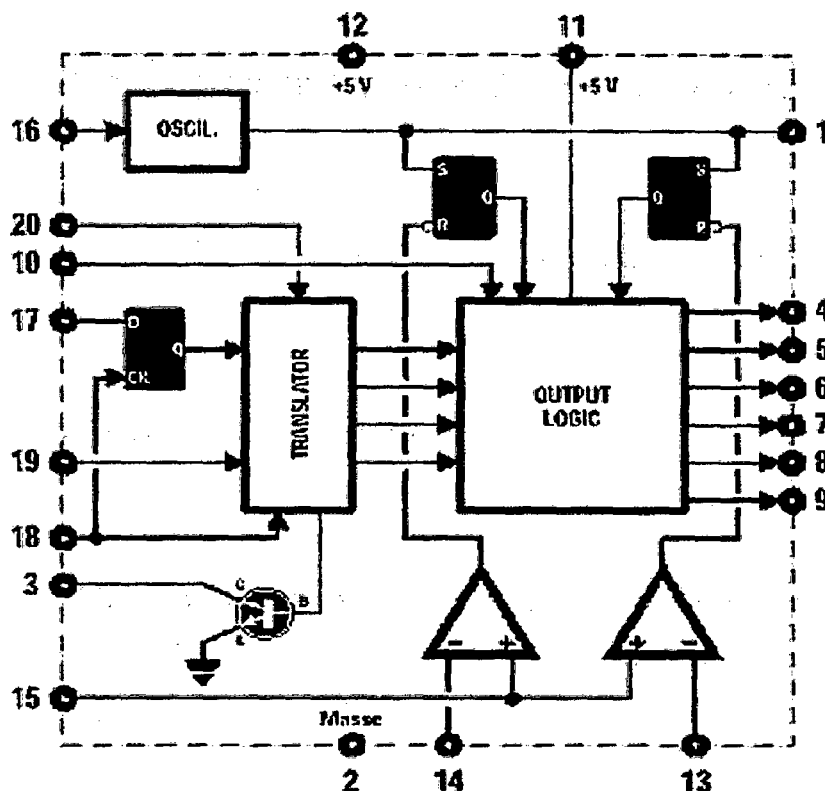


Figura 3.25 Esquema sinóptico del circuito integrado L297 (IC1)

Sus salidas controlan el manejo y sentido de rotación del motor, mientras que los pines 14 y 13 se utilizan para controlar la corriente máxima que pueden absorber las bobinas del motor.

El potenciómetro R2, conectado al pin 15 de IC1, sirve para variar la corriente de manejo del motor de un mínimo de 1mA a un máximo de 2A. Es

gracias a esta posibilidad, que podemos utilizarlo para controlar nuestros motores paso a paso seleccionados.

Las otras entradas en este circuito integrado tienen las siguientes funciones:

- Pin 10 "Permitir" = aplicando una tensión de 5V. En este pin, le permite al integrado IC2 manejar el motor.
- Pin 17 "Dirección" = Si ponemos pin a nivel lógico 0, el motor gira en sentido contrario a las agujas del reloj. Si ponemos un nivel lógico alto (5V), girará en el sentido de las agujas del reloj. Por lo tanto si nos equivocamos al conectar las bobinas A-A y B-B, y las invertimos, no tendremos problema alguno, ya que podremos invertir el sentido de giro al actuar sobre este pin.
- Pin 19 "medio/completo" = si la ponemos a nivel lógico 0, el eje del motor gira en pasos completos, mientras que si se pone a nivel lógico 1, el eje del motor gira a medios pasos.
- Pin 18 "Reloj" = en este pin, aplicaremos una onda cuadrada, en la que la frecuencia determina la velocidad de giro del motor.

En nuestro sistema estos pines son los que van a ir conectados a los pines de salidas digitales para el control de motores del dsPIC, claro que nosotros solo tenemos pines de salida de señal de reloj y sentido de giro, esto se debe a que para nuestro diseño solo necesitamos estas dos señales, el pin 10 "permitir" lo conectaremos a 5V para que este apto el motor a girar y solo esté esperando la señal de reloj, que es el que va a hacer girar el motor de



salida de IC2, para evitar que pueda dañarse, protegiendo igualmente al motor.

Los diodos schottky, DS2 a DS9, que se conectan a las salidas A-A y B-B, tanto sobre el positivo como sobre la masa, sirven para proteger al circuito integrado de tensiones peligrosas, siempre presentes en las fases de conmutación.

Las patillas 11 y 12 de IC1 y la patilla 9 de IC2, están alimentadas con una tensión estabilizada de 5V, mientras que en la patilla 4 de IC3, se aplica una tensión continua de 24V, que servirá para alimentar las bobinas del motor.

En la figura 3.27 se muestra el esquema de nuestro controlador de motores paso a paso, donde solo se va a controlar la habilitación del controlador, la dirección de giro del motor y el reloj que es el que va hacer girar el motor, cada pulso de reloj va hacer girar  $1.8^\circ$  al motor. Cabe resaltar que es necesaria el pin de habilitación del motor, es el que activa al controlador L298, el cual maneja toda la potencia de nuestro circuito controlador de motores paso a paso, cuando se habilita el controlador del motor, se energiza las bobinas del motor y esto hace que se muestre el torque del motor, se debe de habilitar al controlador siempre en cuando se va a utilizar el motor, porque si habilitas el controlador y no le envías pulsos para hacer girar el motor, el controlador L298 empieza a calentarse y disipa mucho calor, en cambio si el motor empieza a girar todo esa energía se va en el movimiento del motor.

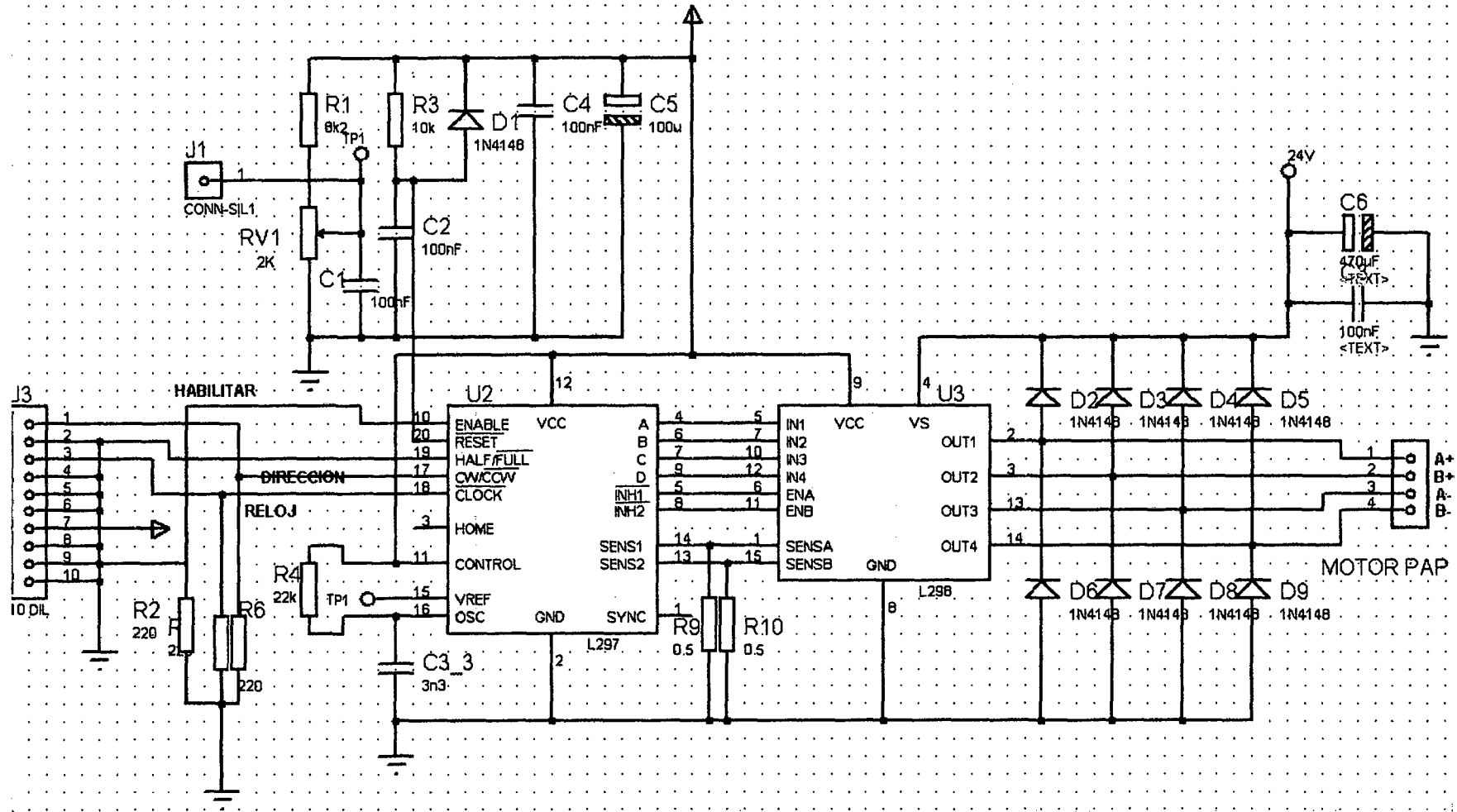


Figura 3.27 Esquema del controlador de motores paso a paso



Ahora que ya hemos descrito todos los circuitos que compondrán nuestro sistema electrónico, nos falta solo integrarlos. El sistema será como lo presenta la industria electrónica actual, la fuente de poder compacto en un solo dispositivo, el PLC compacto como otro dispositivo, sus módulos de expansión compactos en otros dispositivos, los controladores de los motores compactos en otros dispositivos y todo esto integrado en un tablero eléctrico con todas sus protecciones.

Ya se ha mostrado la fuente de poder, como un circuito completo con sus salidas para alimentar al circuito de control y a los controladores de los motores paso a paso, esta fuente de poder nos entregará 24VDC, 12VDC, 5VDC y 3.3VDC, para alimentar los distintos circuitos.

También hemos mostrado el circuito que gobernará nuestro sistema, el circuito del dsPIC, con las conexiones necesarias para funcionar correctamente.

También se ha mostrado el circuito de los controladores de los motores paso a paso que requieren alimentación de 24VDC y 5VDC, así como también la llegada de las señales desde el circuito de control, para controlar la dirección, la habilitación del controlador y el giro de los motores paso a paso.

Prácticamente hemos mostrado todo el sistema desde el controlador dsPIC hasta los acondicionamientos de las entradas y salidas digitales, lo único que faltaría es la unión de todo el sistema en un tablero eléctrico.

También tenemos los circuitos impresos de todos los circuitos electrónicos del sistema y estos se muestran en el Anexo C.

## CAPÍTULO 4

### DISEÑO DEL SISTEMA DE CONTROL

#### 4.1. Entradas y Salidas del dsPIC para el Control del Sistema

El control del sistema es netamente digital, tanto las entradas como las salidas son digitales, en esta parte vamos a mostrar todas las entradas y las salidas que interactuarán con el dsPIC a programar para controlar el sistema, ya se han mencionado estas entradas y salidas digitales en la parte del Diseño electrónico del sistema, pero es necesario volverlos a mencionar, ponerles un símbolo y describir a que pines del dsPIC van asignados, para su identificación en la programación del dsPIC.

#### Entradas digitales del sistema

Tabla 4.1 Entradas digitales del sistema

| Dispositivo       | Función                           | Abreviación | pin  |
|-------------------|-----------------------------------|-------------|------|
| pulsador          | selección jugo surtido            | J1          | RB0  |
| pulsador          | selección jugo papaya             | J2          | RB1  |
| pulsador          | selección jugo piña               | J3          | RB2  |
| pulsador          | selección jugo manzana            | J4          | RB3  |
| pulsador          | selección jugo plátano            | J5          | RB4  |
| switch            | selección manual/automático       | M/A         | RB5  |
| switch            | encender electroválvula manual    | EV_M        | RG9  |
| monedero          | monedero de 2 soles               | C           | RE7  |
| monedero          | monedero de 1 sol                 | B           | RE6  |
| monedero          | monedero de 0.5 soles             | A           | RE5  |
| sensor capacitivo | reconocer si falta vaso           | FV          | RG8  |
| termistor         | mide temperatura de licuadora     | TL          | RD11 |
| sensor capacitivo | reconocer vaso listo para servido | LV          | RG6  |

## Salidas digitales del sistema

Tabla 4.2 salidas digitales del sistema.

| <b>_Dispositivo</b> | <b>Función</b>                      | <b>Abreviación</b> | <b>pin</b> |
|---------------------|-------------------------------------|--------------------|------------|
| LCD                 | transferencia de información        | D0                 | RD0        |
| LCD                 | transferencia de información        | D1                 | RD1        |
| LCD                 | transferencia de información        | D2                 | RD2        |
| LCD                 | transferencia de información        | D3                 | RD3        |
| LCD                 | transferencia de información        | D4                 | RD4        |
| LCD                 | transferencia de información        | D5                 | RD5        |
| LCD                 | transferencia de información        | D6                 | RD6        |
| LCD                 | transferencia de información        | D7                 | RD7        |
| LCD                 | selección de registro control/datos | RS                 | RD9        |
| LCD                 | señal de lectura/escritura          | R/W                | RD8        |
| LCD                 | señal activación modulo LCD         | E                  | RD10       |
| motor 1             | señal giro motor 1                  | C-M1               | RF0        |
| motor 1             | señal ENABLE motor1                 | E-M1               | RF1        |
| motor 2             | señal giro motor 2                  | C-M2               | RE0        |
| motor 2             | señal ENABLE motor 2                | E-M2               | RE1        |
| motor 3             | señal giro motor 3                  | C-M3               | RC13       |
| motor 3             | señal ENABLE motor 3                | E-M3               | RC14       |
| motor 4             | señal giro motor 4                  | C-M4               | RC12       |
| motor 4             | señal ENABLE motor 4                | E-M4               | RC15       |
| motor 5             | señal giro motor 5                  | C-M5               | RG2        |
| motor 5             | señal ENABLE motor 5                | E-M5               | RG3        |
| motor 6             | señal giro motor 6                  | C-M6               | RE2        |
| motor 6             | señal ENABLE motor 6                | E-M6               | RE3        |
| motor 6             | señal dirección motor 6             | D-M6               | RE4        |
| motor 7             | señal giro motor 7                  | C-M7               | RF3        |
| motor 7             | señal ENABLE motor 7                | E-M7               | RF2        |
| motor 7             | señal dirección motor 7             | D-M7               | RF6        |
| motor 8             | señal giro motor 8                  | C-M8               | RF4        |
| motor 8             | señal ENABLE motor 8                | E-M8               | RF5        |
| motor 8             | señal dirección motor 8             | D-M8               | RG7        |
| licuadora           | encendido licuadora                 | EL                 | RB8        |
| electroválvula      | encendido electroválvula            | EEV                | RB9        |
| alarma              | encendido alarma por fallas         | A                  | RB10       |
| Motor fruta 1       | para asegurar caída de fruta        | MF1                | RB11       |
| Motor fruta 2       | para asegurar caída de fruta        | MF2                | RB12       |
| Motor fruta 3       | para asegurar caída de fruta        | MF3                | RB13       |
| Motor fruta 4       | para asegurar caída de fruta        | MF4                | RB14       |
| Motor fruta 5       | para asegurar caída de fruta        | MF5                | RB15       |

#### 4.2. Diagrama de Flujo del Sistema

Todo proceso requiere de un diagrama de flujo para luego programar el controlador en cualquier tipo de lenguaje de programación, es por esto que primero tenemos que mostrar el diagrama de flujo para nuestro sistema, pero antes mostramos en la Figura 4.1 el sistema completo con todos los sensores y actuadores para un mejor entendimiento del diagrama de flujo que se mostrará más adelante.

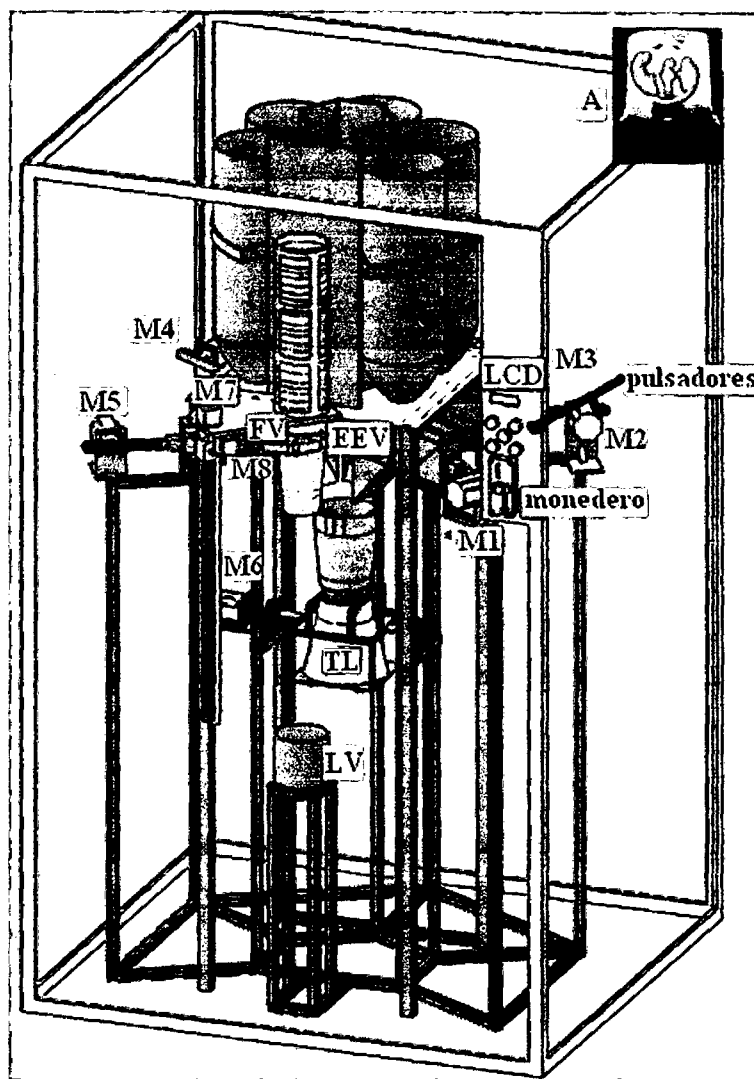
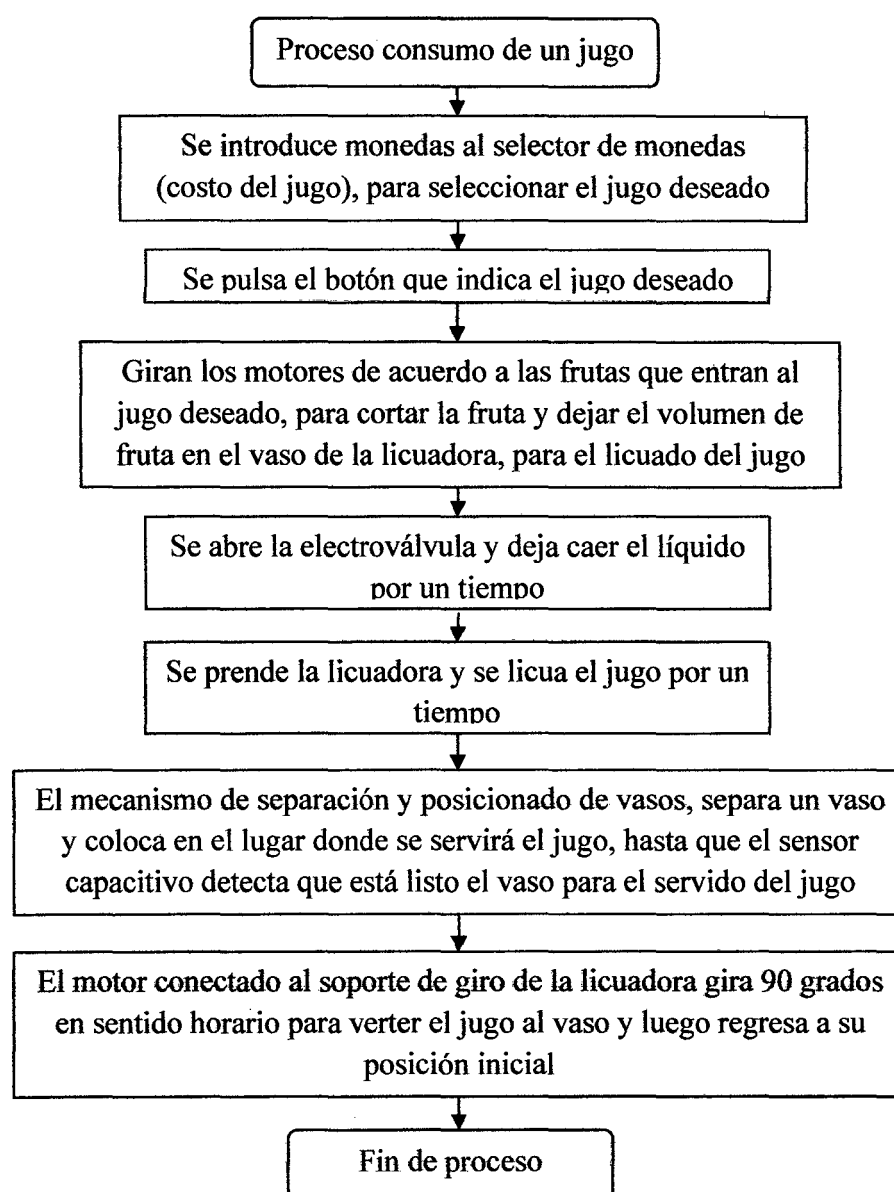


Figura 4.1 Sistema completo con posicionado de sensores y actuadores

Veamos primero un diagrama de flujo en macro sobre lo que realizará el sistema para un funcionamiento normal y luego veremos un diagrama de flujo en micro, más detallado y utilizando las abreviaturas de las entradas y salidas digitales y con los pines de los dsPIC.

### Diagrama de flujo del sistema en macro



### **4.3. Programación del dsPIC en Lenguaje C MPLAB IDE**

MPLAB IDE es un programa de software que corre en una PC convencional para desarrollar aplicaciones para los microcontroladores Microchip. Es denominado Entorno de Desarrollo Integrado (Integrated Development Environment), IDE en adelante, debido a que provee un entorno simple para desarrollar código para microcontroladores.

Se puede descargar el software MPLAB de [www.microchip.com](http://www.microchip.com), después de descargar el archivo ejecutable o comprimido, descomprimir y/o ejecutar el archivo setup.exe para instalar el programa.

#### **4.3.1 Componentes del MPLAB IDE**

Los principales componentes del MPLAB IDE son los siguientes

- Administrador de Proyectos (Project Manager)

El administrador de proyectos provee integración y comunicación entre el entorno de desarrollo (simulación, configuración, emulación y depuración) con las herramientas de programación (lenguaje de programación máquina, C30).

- Editor

El editor es un editor de texto se utiliza para escribir los códigos fuente con muchas características para programación y también sirve como ventana para la depuración de los programas.

- **Ensamblador/Enlazador y Herramientas de Lenguaje**

El ensamblador puede ser usado solo sin ayuda para ensamblar un archivo, o puede ser usado como enlazador o linker para construir un proyecto de diferentes archivos fuente, librerías y objetos recompilados. El linker es responsable de posicionar el código compilado en áreas de memoria de microcontrolador destino.

- **Motores de Ejecución (Execution Engines)**

Existe software de simulación en MPLAB IDE para todos los dispositivos MCU PICmicro y dsPIC DSC. Estos simuladores utilizan una PC para simular las instrucciones y algunas funciones periféricas de los dispositivos MCU PICmicro y dsPIC DSC.

- **Herramientas del Lenguaje Compilador**

Los compiladores de MPLAB C18 y MPLAB C30 de Microchip proveen código totalmente integrado y optimizado. Los compiladores de HI-TECH, IAR, micro Engineering Labs, CCS y Byte Craft, pueden ser invocados por el administrador de proyectos de MPLAB IDE para compilar código que es automáticamente cargado en el debugger destino para instancias de prueba y verificación.

- **Debugger o Depurador**

El depurador Microchip permite breakpoints, correr el programa paso a paso, ventanas de visualización (Watch Windows) y todas las

características de un depurador moderno. Trabaja en conjunto con el editor para referirse a la información destino siendo el código fuente depurado en realimentación.

- Programador

PICSTART Plus, PICKIT 1 y 2, PRO MATE II, MPLAB PM3 tanto como el MPLAB ICD 3 pueden programar código en los dispositivos destino. MPLAB IDE ofrece control total sobre programación de código y datos, tanto como de bits de configuración para definir los diferentes modos de operación de los microcontroladores destino o dsPIC.

#### **4.3.2 El Lenguaje de Programación C30**

MPLAB C30 es una adaptación al ANSI x3.159-1989, optimizando el compilador C que incluye extensiones para aplicaciones de control con el dsPIC DSC. El compilador es una aplicación para entorno Windows que provee una plataforma de desarrollo en código C.

MPLAB C30 compila archivos fuente en C, produciendo archivos en lenguaje ensamblador. Estos archivos compilados y producidos por la compilación se enlazan con otros archivos objeto y librerías para producir la aplicación final en ejecutable en formato COFF y ELF. Los archivos COFF o ELF pueden ser cargados por MPLAB IDE, donde pueden ser probados y depurados, o también utilizar el utilitario de conversión para pasar de COFF o ELF a formato HEX de Intel, adecuado para cargarlo en la línea de comandos



para la simulación o el respectivo programador, en la Figura 4.2 se muestra el diagrama de flujo de desarrollo del software.

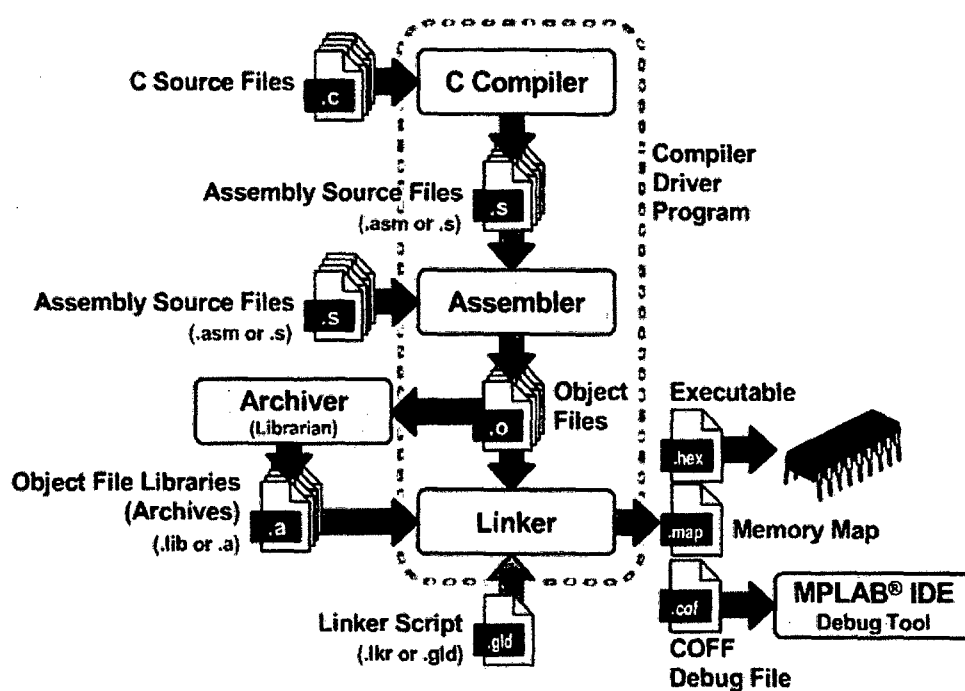


Figura 4.2 diagrama de flujo de desarrollo del software

- Estándar ANSI C

El compilador MPLAB C30 es un compilador totalmente validado que conforma el estándar ANSI C como fue definido por la especificación ANSI descrita por Kernighan y Ritchie's. El estándar ANSI C incluye extensiones a la definición original C, estas extensiones incrementan la portabilidad y ofrece una mayor capacidad.

- Optimización

El compilador utiliza un conjunto sofisticado de optimización que emplea muchas técnicas avanzadas para generar eficiente y compacto código desde código fuente C. Esta optimización toma ventaja determinadas particulares del dsPIC.

- Soporte de librerías del Estándar ANSI C

MPLAB C30 es distribuido con una completa librería del estándar ANSI C. Todas las funciones de la librería han sido validadas, y conforman el estándar de librería ANSI C. Las librerías incluyen funciones para manejo de caracteres, asignación de memoria dinámica, conversión de datos y funciones matemáticas. Las funciones estándar de Entrada/Salida para la manipulación de archivos también están incluidas y distribuidas, las cuales soportan acceso total al sistema de archivos usando el simulador de línea de comandos.

- Driver del compilador

MPLAB C30 incluye un potente driver de programa de línea de comandos. Usando el Driver de Programa, programas de aplicación pueden ser compilados, ensamblados y enlazados en un simple paso.

### 4.3.3 Creando un Proyecto en MPLAB IDE

Primeramente tenemos que instalar el MPLAB IDE que se puede descargar desde <http://www.microchip.com>, también tenemos que instalar el compilador C que se puede descargar también de la misma página, pero se descarga la versión estudiantil del compilador C30, esta ofrece 60 días de pruebas optimas, luego de estos días le quita algunas optimizaciones al software. En esta misma página se encuentra mucha información sobre la utilización de este software como manuales y tutoriales.

Después de esto ya podemos ejecutar el MPLAB IDE para construir nuestro proyecto.

Lo primero que tenemos que hacer es crear una carpeta en cualquier parte del disco de la PC para guardar todos los archivos con los que se trabaja.

Luego creamos el proyecto usando el Asistente de Proyectos (Project Wizard). El Asistente de Proyectos nos ayuda a crear el proyecto, así como organizar los archivos internos a ser compilados y ensamblados. Para abrir el asistente se debe ejecutar: `Project> Project Wizard` como se muestra en la siguiente Figura 4.3.

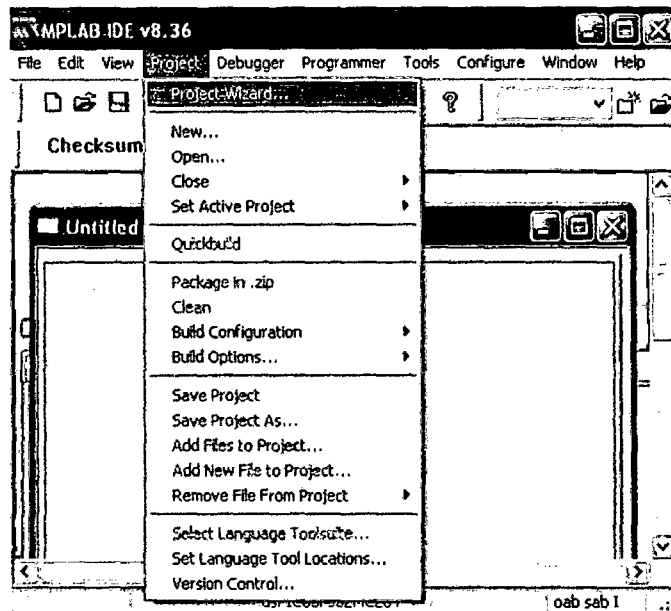


Figura 4.3 Utilizamos el asistente de creación de proyectos

Desde el dialogo de bienvenida, presione siguiente> para continuar.

En la siguiente ventana, que se muestra en la Figura 4.4 seleccionamos el modelo del dsPIC a usar y presionamos siguiente>, en nuestro caso va a ser el dsPIC33FJ64GS606.

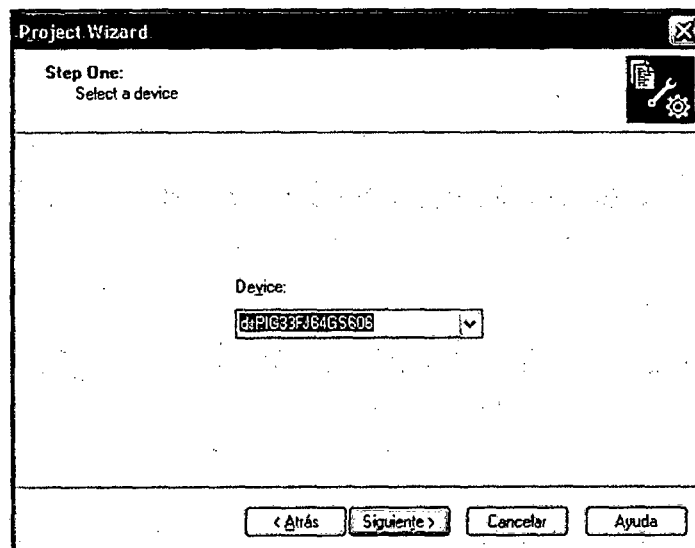


Figura 4.4 Seleccionamos el modelo de dsPIC a usar

Luego en la siguiente ventana, que se muestra en la Figura 4.5 seleccionamos en **Active toolsuite: Microchip C30 toolsuite** y en **toolsuite contents: MPLAB C30 C Compiler**, después presionamos siguiente.

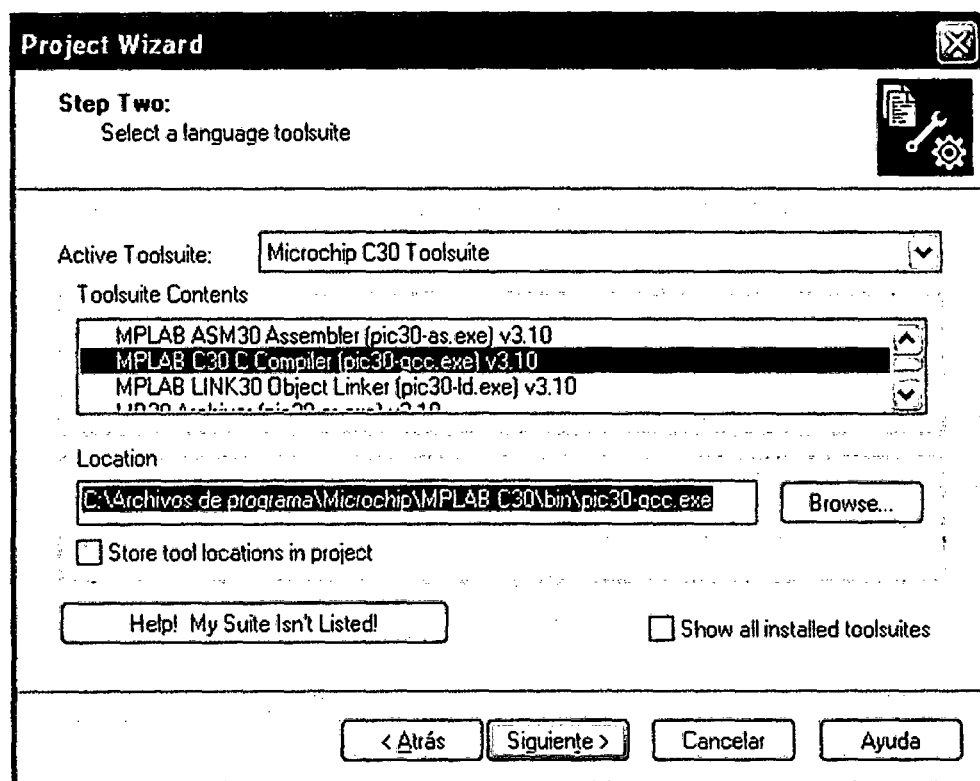


Figura 4.5 Ventana de selección del compilador a utilizar

Luego en la siguiente ventana que se muestra en la Figura 4.6, nombramos el proyecto, presionamos el botón browse... y buscamos la carpeta que creamos primero y le ponemos un nombre al proyecto, en nuestro caso control del proceso, luego presionamos Guardar y después siguiente.

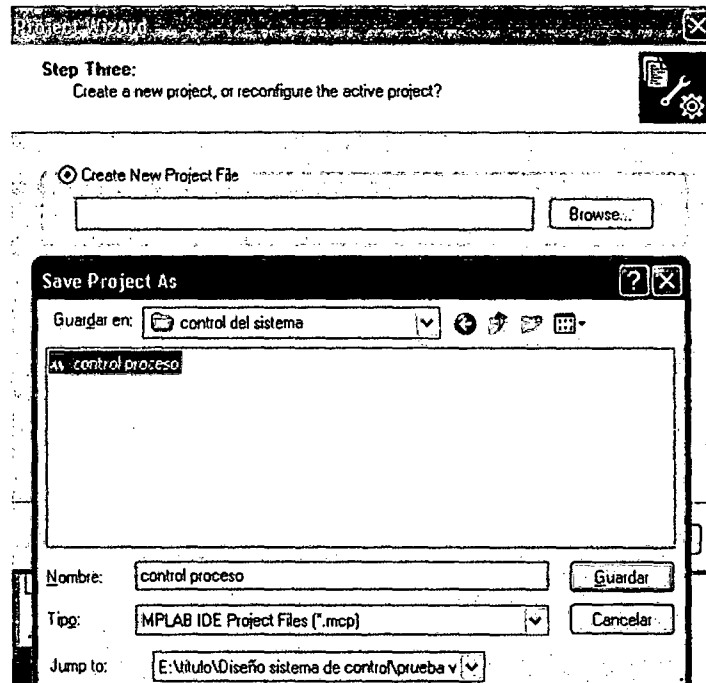


Figura 4.6 Colocamos un nombre al proyecto

En la siguiente ventana que se muestra en la Figura 4.7, el asistente de proyectos no brinda la posibilidad de adicionar archivos al nuevo proyecto, en nuestro caso vamos a comenzar de cero, presionamos siguiente>.

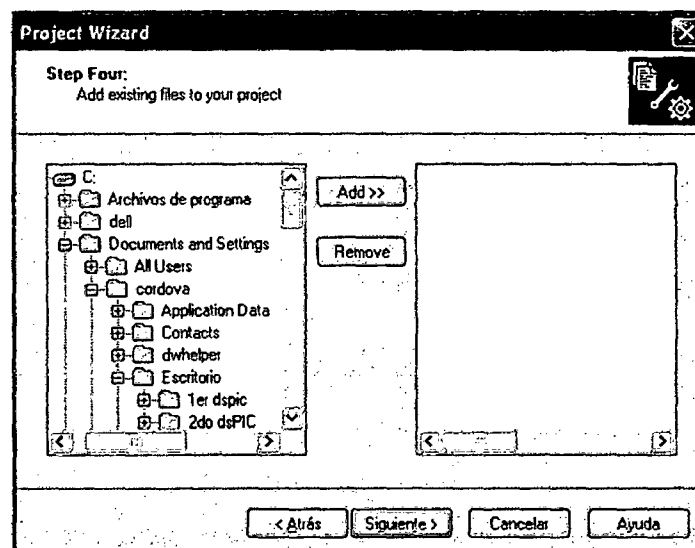


Figura 4.7 Ventana para poder adicionar archivos al programa

En la siguiente ventana que se muestra en la Figura 4.8, al final de los pasos el Asistente del Proyecto nos brinda un resumen con la configuración dada en los pasos anteriores, le das siguiente y luego finalizar.

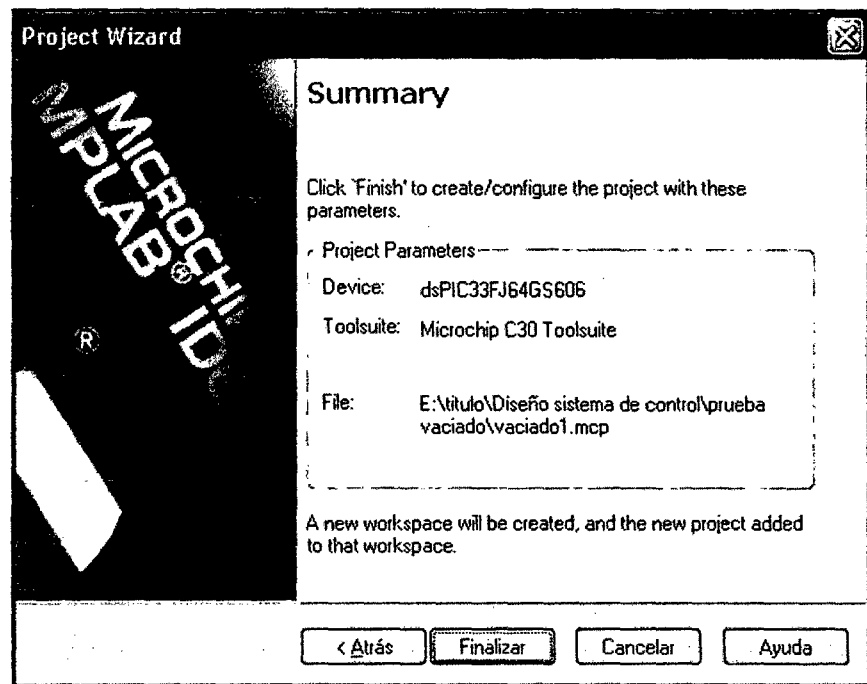


Figura 4.8 Ventana resumen de la configuración del proyecto

Ahora nos encontramos en el entorno MPLAB, es una ventana tal como se muestra en la Figura 4.9, en esta ventana hacemos click en la pestaña **View** de la barra de menú y activamos las ventanas de Project y Output.

- Project: permite ver que archivos hay en el proyecto
- Output: permite ver mensajes de compilación, errores presentados y otros.

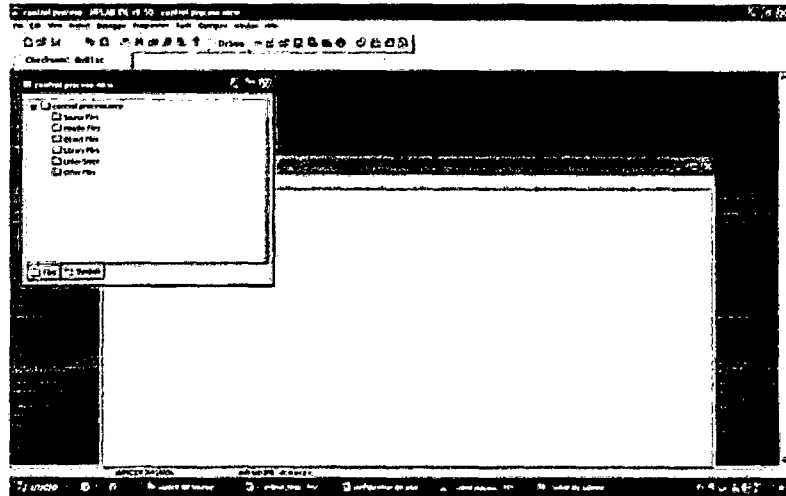


Figura 4.9 Entorno de desarrollo del MPLAB IDE

Luego se añaden archivos al programa que son necesarios para el funcionamiento del programa.

- En la ventana de **Project**, click derecho en **Header Files** y click en **Add Files**, agregamos el archivo `p33FJ64GS606.h` ubicado en:

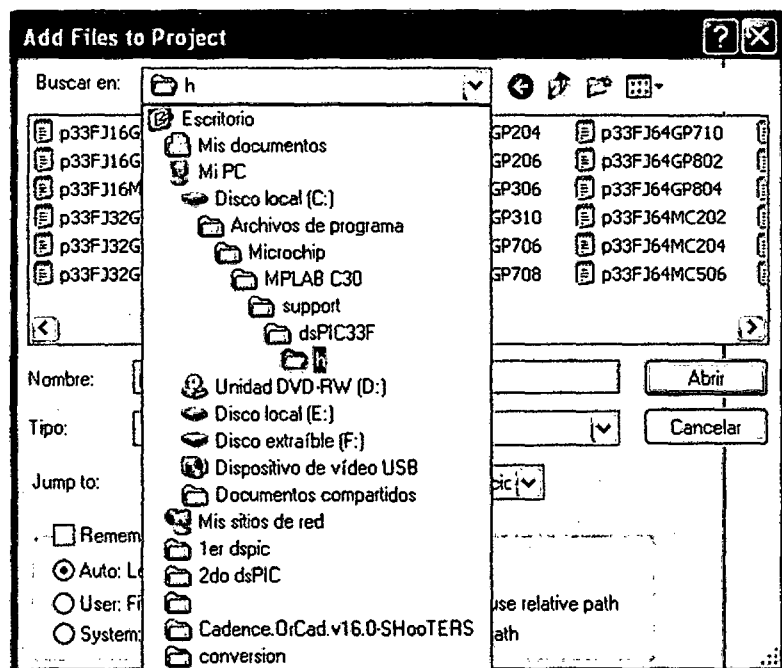


Figura 4.10 Dirección donde encontramos los archivos .h



- En la ventana de **Project**, click derecho en **Linker Script** y click en **Add Files**, agregamos el archivo p33FJ64GS606.gld ubicado en:

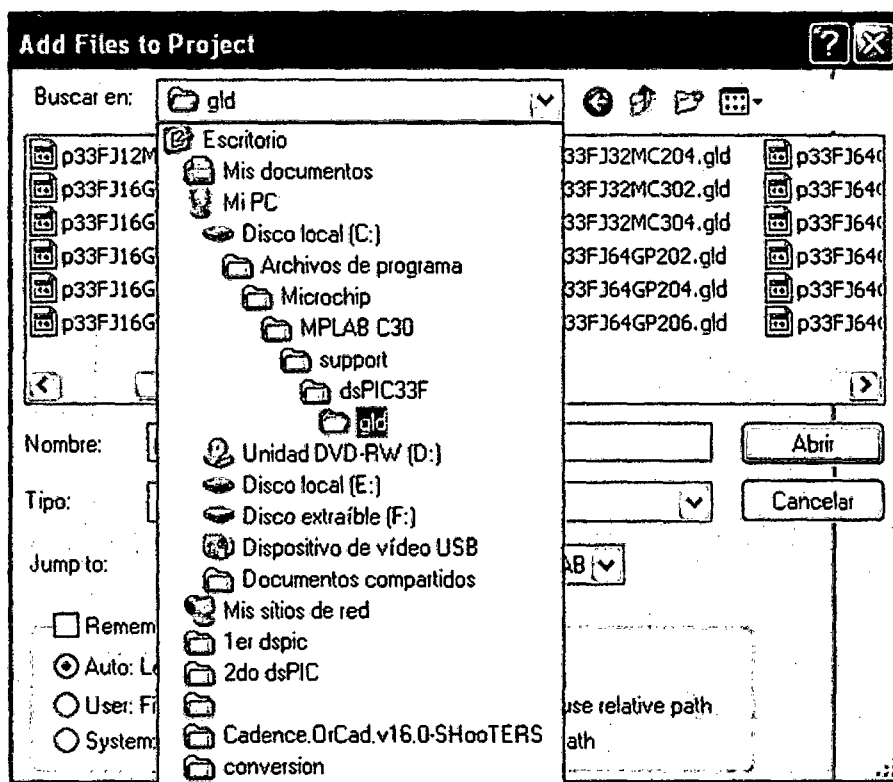


Figura 4.11 Dirección donde encontramos los archivos .gld

Ahora solo nos falta agregar nuestro código en .c; para ello vamos a la Barra de Menú y damos click en **file**→**New**. Aparece un nuevo documento en blanco, el cual lo guardaremos con la extensión .c (en nuestro caso principal.c) en la misma carpeta donde guardamos nuestro proyecto. El código fuente a escribir lo mostraremos en el ANEXO D, una vez guardado el archivo, lo añadimos al proyecto, vamos a la ventana de **Project**, click derecho en **Source Files** y click en **Add Files**, agregamos el archivo .c que acabamos de crear y ya pertenece al proyecto.

Nuestro proyecto se muestra en la Figura 4.12, donde podemos ver los archivos adicionados.

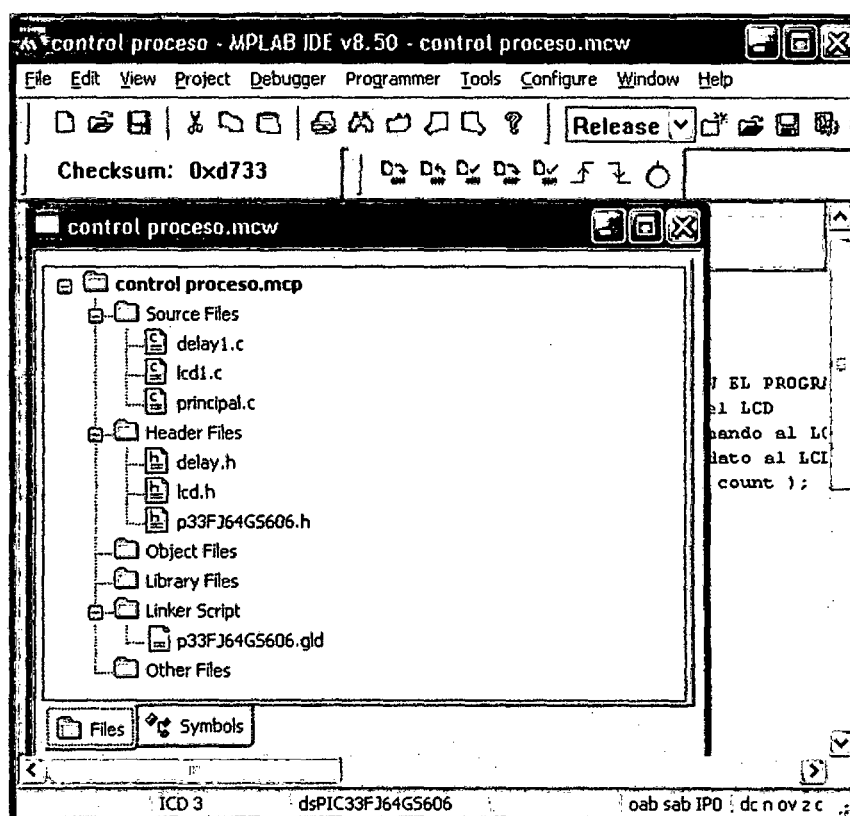


Figura 4.12 Proyecto con la adición de los archivos .h, .gld y .c

Donde los archivos .c son archivos donde se edita el código fuente del programa principal del controlador, ya sea en assembler o en código C. Los archivos .h son las cabeceras donde se declaran algunas variables o constantes que se utilizan en el archivo .c, muchas veces por orden en la programación y manejo de datos se separan la declaración de variables y constantes del programa en un archivo aparte llamado archivo de header (cabecera), con extensión .h y estos archivos son asociados en el archivo .c mediante la directiva `#include <file.h>`. El archivo .gld es una archivo asociado a los

registros de manejo de datos de cada microcontrolador, por lo tanto es muy importante adicionar este archivo para cada tipo de dsPIC a utilizar en un proyecto.

Cabe mencionar que en este proyecto que hice para el control del sistema utilice algunos archivos .c aparte del principal con sus respectivos archivos .h para un mejor manejo de programación, como archivos para controlar el display LCD, donde utilice el archivo **lcd1.c** con su archivo de cabecera **lcd.h** y para los retardos utilice el archivo **delay1.c** con su archivo de cabecera **delay.h**, tales archivos los adicione al proyecto como se muestra en la Figura 4.13.

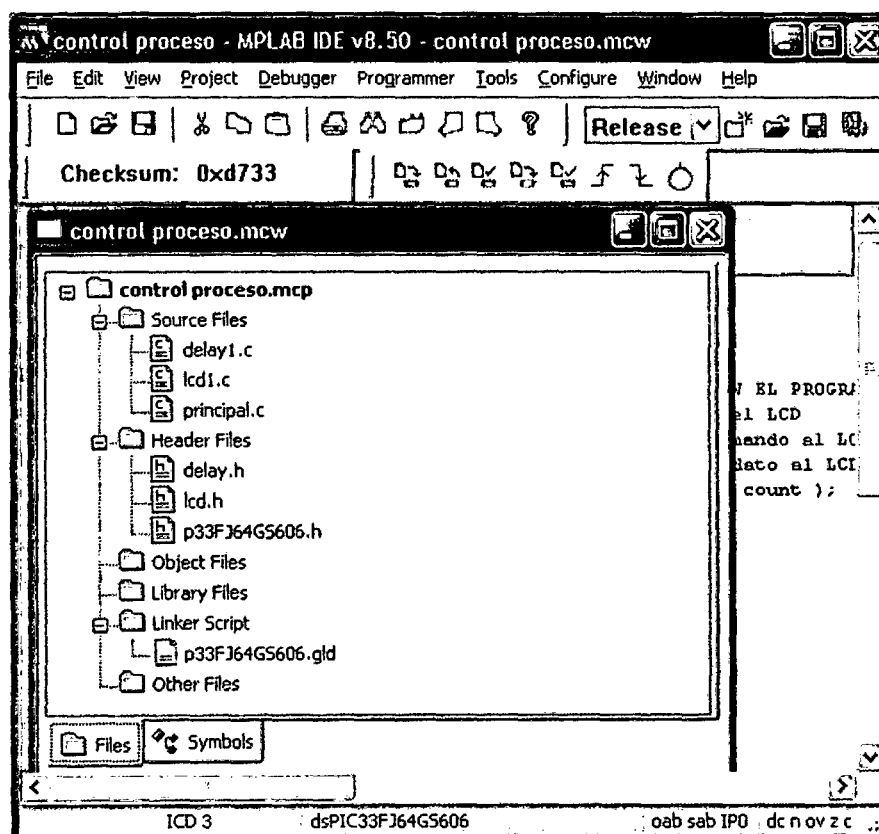


Figura 4.12 Proyecto completo con la adición de los archivos importantes

Para verificar si el programa está bien hecho y si compila vamos a la pestaña **Project** de la Barra de Menú y click en **Build All**. Si está bien compilado en la ventana **Output** debería aparecer el mensaje: “BUILD SUCCEEDED” tal como se muestra en la Figura 4.13 y si está mal saldrá el mensaje “BUILD FAILED”, cuyos errores se mostraran en la misma ventana Output.

```

Output
Build Version Control Find in Files
Dynamic Memory Usage
region          address          maximum length (dec)
-----
heap            0                0 (0)
stack          0x804           0x7fc (204)
Maximum dynamic memory (bytes): 0x7fc (204)
Executing: "C:\Archivos de programa\Microchip\MPLAB C30\bin\pic30-bin2hex.exe" "E:\titulo\Diseño sis
Loaded E:\titulo\Diseño sistema de control\simulacion de software\2do dsPIC\dsPIC2.cof.
Debug build of project 'E:\titulo\Diseño sistema de control\simulacion de software\2do dsPIC\dsPIC2.m
Language tool versions: pic30-as.exe v3.10, pic30-gcc.exe v3.10, pic30-ld.exe v3.10, pic30-ar.exe v3.10
Preprocessor symbol '___DEBUG' is defined.
Mon Aug 02 16:55:01 2010
BUILD SUCCEEDED

```

Figura 4.13 Ventana de salida donde muestra si esta correcto la programación

En el ANEXO D se muestra todo el código fuente de los archivos mencionados con explicaciones línea por línea, tanto las extensiones .c como las extensiones .h.

#### **4.4. Funcionamiento del Sistema para el Preparado de un Jugo**

En esta parte se va a mostrar paso a paso como es el funcionamiento normal de nuestro sistema, como es que va a interactuar con el usuario cuando desee consumir un tipo de jugo, es necesario mostrar entonces lo que resulta de nuestro diseño del sistema de control, nuestra programación en el controlador.

En la Figura 4.14 se muestra el panel que es la parte de nuestro sistema que interactuara con el usuario, como podemos ver, es ahí donde están los 3 selectores comparadores de monedas, estos selectores detectan solo un tipo de moneda, colocamos para que detecten 50 céntimos de nuevo sol, 1 nuevo sol y 2 nuevos soles, esto para hacer más funcional al sistema con respecto a nuestra moneda, debemos resaltar que cada vez que insertemos una moneda, el sistema nos dirá cuanto le falta insertar para completar el costo del jugo mediante el display LCD que se muestra en el mismo panel, es mediante este display LCD por donde el usuario será guiado para consumir un jugo, también se muestra los pulsadores que representan los jugos a elegir, en este caso solo se puede elegir 5 tipos de jugos, el primero que es el surtido, que es la combinación de las 5 frutas, el segundo que es un jugo sólo de papaya, el tercero que un jugo sólo de piña, el cuarto que es un jugo sólo de manzana y el ultimo que es un jugo sólo de plátano.

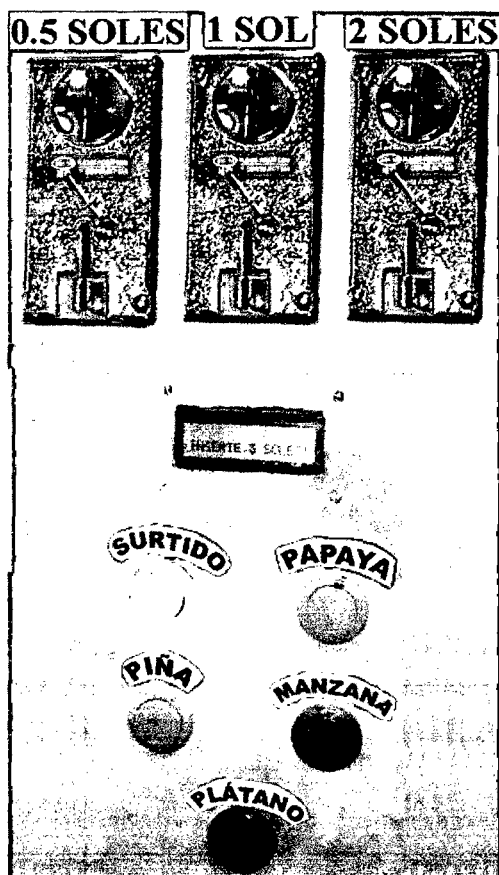


Figura 4.14 panel de interacción con el usuario

El proceso comienza cuando el usuario ve el display LCD donde le indica que inserte tres soles, tal como se muestra en la Figura 4.15.

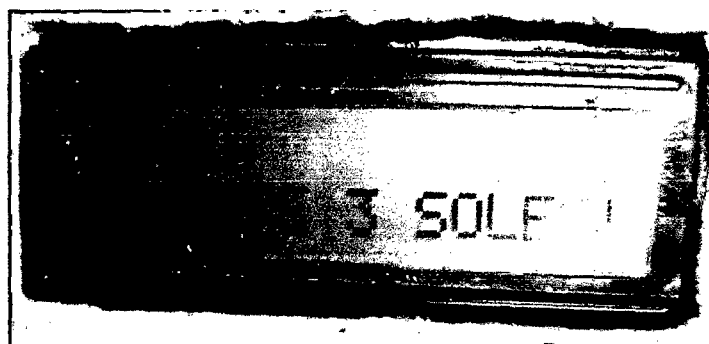


Figura 4.15 Display LCD indicando que se inserte 3 soles

Luego el usuario inserta una moneda, en este caso inserta 1 nuevo sol, donde después de insertar la moneda el display LCD siempre le va a indicar cuanto le falta insertar de dinero, tal como se muestra en la Figura 4.16,

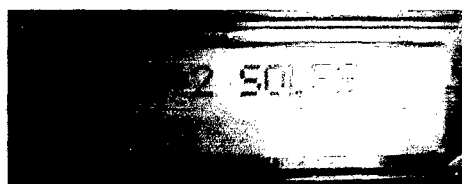


Figura 4.16 Display LCD indicando que le falta colocar 2 soles

Ahora seguimos metiendo las monedas y él nos indicará cuando le falta, por ejemplo ahora le insertamos otra moneda de 1 sol, como se muestra en la Figura 4.17.

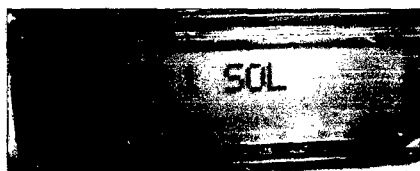


Figura 4.17 Display LCD indicando que le falta todavía insertar 1 sol

Por último le insertamos otra moneda de 1 sol para poder seleccionar un jugo y cuando el sistema cuenta que ya han insertado tres soles, sale el mensaje de "PULSE EL JUGO QUE DESEA SERVIRSE", tal como se muestra en la Figura 4.18.



Figura 4.18 Display LCD indicando que puede elegir un jugo

En nuestro caso vamos a seleccionar para el preparado de un jugo surtido, entonces presionamos el pulsador de surtido tal como se muestra en la Figura 4.19.

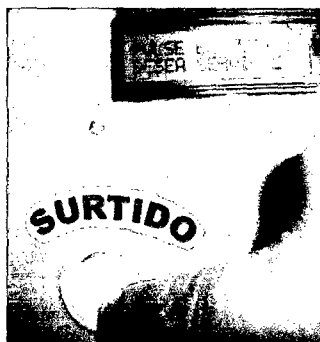


Figura 4.19 Presionamos el pulsador de jugo surtido

Luego de presionar jugo surtido el display LCD nos indicará el jugo que hemos elegido, tal como se observa en la Figura 4.20

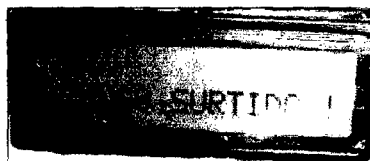


Figura 4.20 Display LCD indicando el jugo elegido

Después de pulsar el pulsador de jugo surtido, lo primero que hace el sistema es hacer girar el primer motor para cortar la papaya (ver Figura 4.21-1), luego el segundo motor para cortar la piña (ver Figura 4.21-2), luego el tercer motor para cortar la manzana(ver Figura 4.22-3), luego el cuarto motor para cortar el plátano y por último el quinto motor para cortar el melón (ver Figura 4.22-5), todo estos cortes se muestran en las siguientes pasos del 1 al 5 de la Figura 4.21 y Figura 4.22.



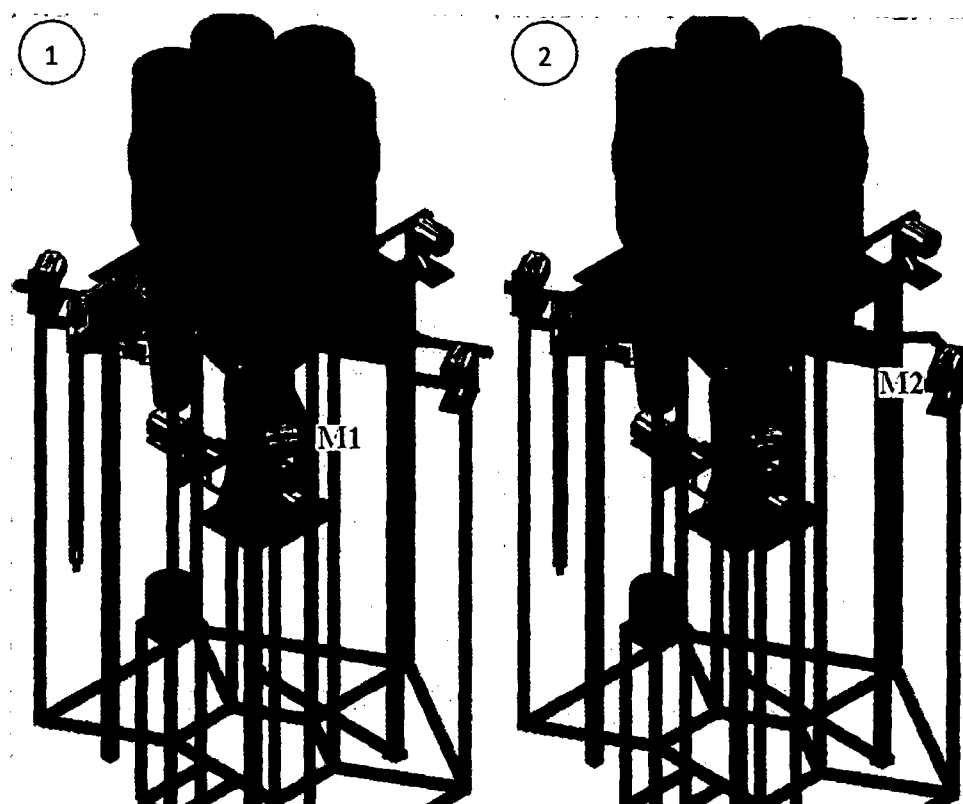


Figura 4.21 Giro motores 1 y 2 (papaya, piña)

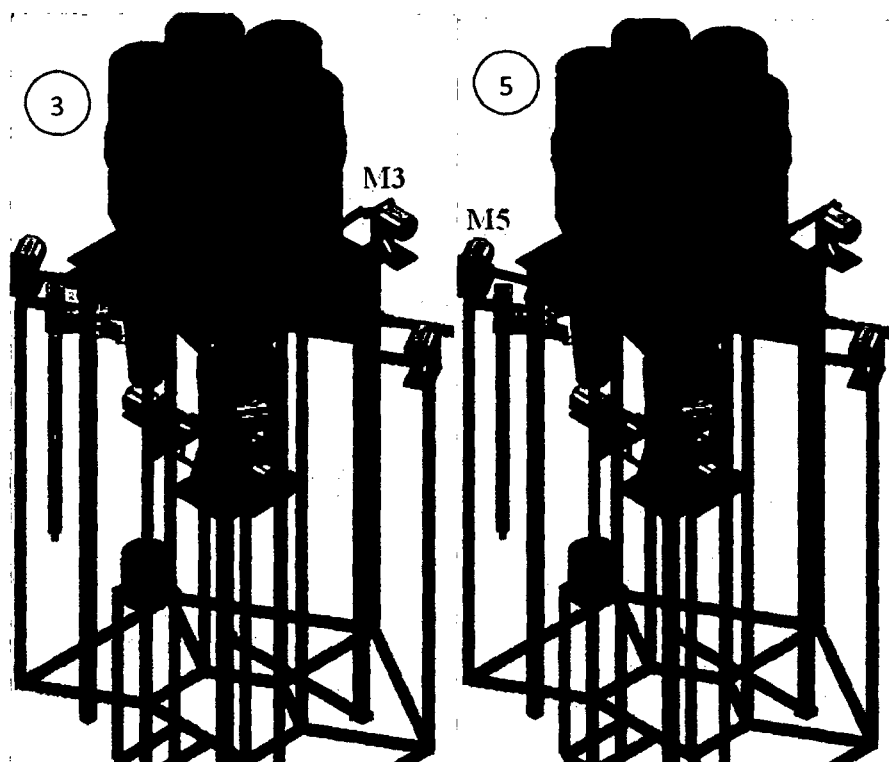


Figura 4.22 Giro motores 3 y 5 (manzana, melón)

Una vez que todas las frutas están en la licuadora se apertura la electroválvula por un tiempo para que entre el líquido para el combinado del jugo, luego se enciende la licuadora para el licuado del jugo por un tiempo, pero mientras el jugo se licua, el mecanismo de separación y posicionado de vasos empieza a separar un vaso y colocar en el punto donde se verterá el jugo, todo esto lo hace de esta forma, el eje del motor8 está acoplado a un tornillo sinfín que se une con dos engranajes posicionados al lado derecho e izquierdo, estos engranajes están unidos a dos dedos especiales que cuando el motor gira en un sentido se juntan los dedos y presionan el vasos y cuando el motor gira en sentido contrario se abren los dedos para soltar el vaso.

En la figura 4.23 se muestra cuando el motor8 gira en sentido horario para sujetar el vaso.

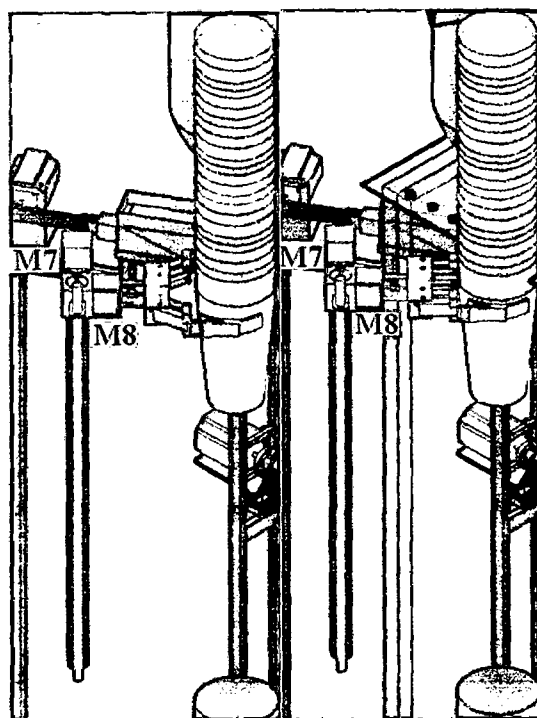


Figura 4.23 Giro horario motor8 para sujetar vaso

Después de sujetar el vaso, se activa el motor7 que también esta acoplado a un tornillo sinfín, al girar hace bajar a la mano que sujeta el vaso, separando el vaso de los demás y lo lleva a su posición final y lo suelta ahí, para luego regresar a su posición inicial, para comenzar un nuevo proceso, todo esto se muestra en los pasos del 1 a 3 de la Figura 4.24, desde que baja el brazo, hasta que regrese a su posición inicial.

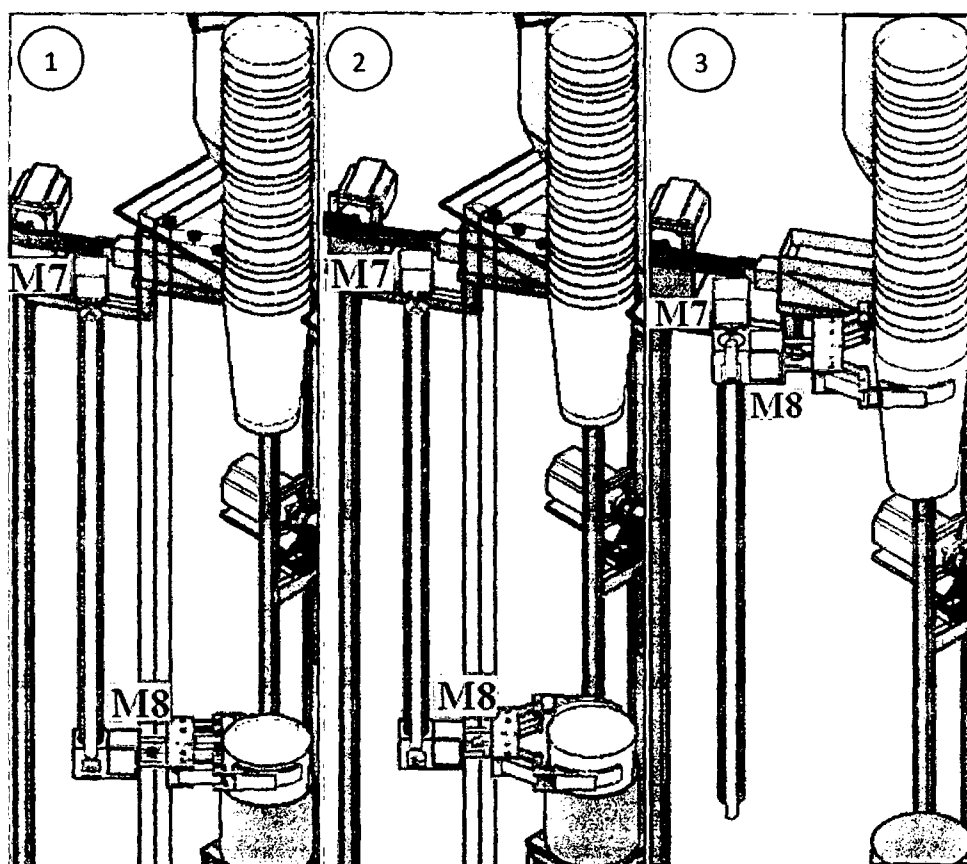


Figura 4.24 Giro horario motor7 para bajar el vaso, giro antihorario motor8 para soltar el vaso y giro antihorario motor7, para regresar posición inicial

Ojo que si en toda esa operación el mecanismo no logra separar un vaso, el sistema vuelve a repetir lo mismo hasta que lo logre, en todo caso, el usuario puede colocar el vaso para que el sistema finalice con el servido del jugo.

Cuando el sensor capacitivo detecte que el vaso está listo para el servido del jugo, entonces se apaga la licuadora y se activa el motor6 para un giro horario de 90 grados, para verter el jugo, después de un tiempo, se activa el mismo motor6 pero esta vez para girar 90 grados en sentido antihorario para regresar a su posición inicial, todo esto se muestra en los pasos del 1 al 4 de la Figura 4.25.

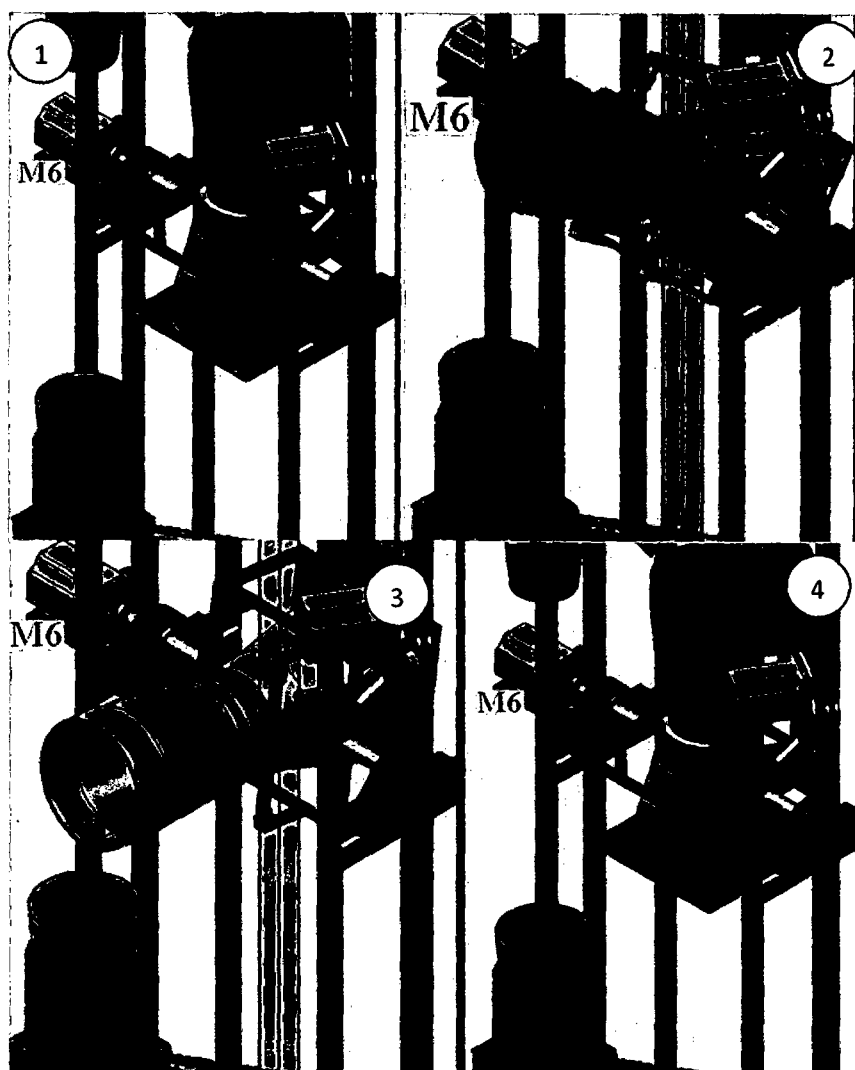


Figura 4.25 Giro motor6 horario y antihorario para verter el jugo y regresar

Con esto termina el proceso con una indicación en el display LCD de “BUEN PROVECHO”, tal como muestra la Figura 4.26.



Figura 4.26 Display LCD, con el cual da por concluido el proceso.

Luego de esto el sistema vuelve a indicar mediante el display LCD que inserte 3 soles, para un nuevo proceso de preparado de jugo.

Cabe indicar que este sistema también se le puede utilizar en modo manual, podemos girar los motores paso a paso, abrir la electroválvula, etc. En este modo es más que todo para la limpieza luego de acabado el día, en caso de no haberse acabado la fruta ni el líquido o para darle un mantenimiento.

## CAPÍTULO 5

### ESTUDIO ECONÓMICO - FINANCIERO

#### 5.1. Introducción

Cuando alguien diseña un nuevo sistema, para crear empresa con ese sistema, su mayor preocupación radica básicamente en tres aspectos que son:

- Verificar que exista un mercado potencial y que es viable, desde el punto de vista operativo, introducir a ese mercado el sistema diseñado.
- Demostrar que tecnológicamente es posible producirlo, una vez que se verifico que no existe impedimento alguno en el abasto de todos los insumos necesarios para su producción.
- Demostrar que es económicamente rentable llevar a cabo su realización.

En relación a nuestro sistema diseñado y basado en estos tres aspectos básicos, tenemos:

Haciendo un análisis de mercado nuestro diseño está hecho para un mercado de competencia perfecta, donde hay muchos vendedores y muchos compradores, pero nuestro diseño tiene un valor agregado por ser un sistema novedoso, rápido y de fácil uso, por tal motivo será muy fácil introducirlo al mercado sin ningún problema.

Tecnológicamente es posible producirlo porque se tiene la infraestructura, es posible la importación de todos los dispositivos y tenemos la capacidad para desarrollarlo.

Demostrar que económicamente es rentable, es el gran reto, porque de esto depende el éxito (implementación del diseño) o el fracaso del sistema diseñado, es por eso que tenemos la obligación de demostrar que el resultado de esta tesis es económicamente rentable, para nuestro beneficio y el beneficio del inversionista. Para tal motivo nosotros vamos a basar esta evaluación económica primeramente en un costo total del sistema implementado y luego mostrar un estudio económico financiero, para una persona natural que desea hacer empresa con este sistema.

## **5.2. Costo del Sistema Implementado**

### **Manufactura del Diseño Mecánico**

El costo de producir toda la parte mecánica entre soportes, recipientes, mecanismos, mostrador, etc. Según la empresa PRODERMET tiene un costo de US \$3000, esto debido a que la mayoría de las partes del sistema están hechos de acero inoxidable AISI 304 y AISI316, también porque la estructura es un poco compleja y requiere de mucha experiencia en el corte y soldadura del acero inoxidable. Además del apoyo en la instalación de los sensores y los actuadores del sistema.

**Total manufactura: US \$3000**

### **Sensores**

- Pulsadores: se requiere 12 pulsadores industriales cada uno de los pulsadores cuesta US \$15, por lo tanto seria US \$180.
- Sensores capacitivos: requerimos dos sensores capacitivos, cada sensor capacitivo cuesta US \$60, por lo tanto seria US \$120.
- Selector de monedas: se requiere de un selector de monedas, cuyo costo es US \$120.

Total costo sensores: US \$420

### **Actuadores**

- Motores paso a paso: se requiere de 7 motores paso a paso de un torque de 280 oz\*in cuyo NEMA es 23, el costo por unidad de este motor es US\$60, por lo tanto serian US\$420, también requerimos de un motor paso a paso de 400 oz\*in cuyo NEMA es 24, el costo de este motor es US\$70.
- Licuadora: se requiere una licuadora de 600Watts de potencia cuyo costo es de US\$200.
- Electroválvula: se requiere de una electroválvula normalmente cerrada (NC), con cuerpo de acero inoxidable, con 0 bar de presión mínima cuyo costo es de US \$500.
- Display LCD: se requiere un display LCD cuyo costo es US \$15.
- Circulina de indicación de alarma: se requiere uno, cuyo costo es US\$15.

Total costo actuadores: US \$1220



### Costo Implementación de Circuitos Electrónicos

- Fuente de Poder: el costo para armar este circuito es de US\$42.
- Circuito de control de motores paso a paso: el costo de armar uno de estos circuitos es de US\$31, pero como necesitamos 8 de estos circuitos para controlar los 8 motores del sistema, entonces serian US\$248.
- Circuito de control: el costo para armar este circuito es de US\$120.

Total costo implementación de circuitos electrónicos: US\$410

Vamos a mostrar en la Tabla 5.1 un resumen hasta este punto.

Tabla 5.1 Costo manufactura e insumos de implementación sistema

| <b>Manufactura del diseño mecánico</b>                       |                              |                 |                     |
|--|------------------------------|-----------------|---------------------|
| <b>item</b>  | <b>costo unitario (US\$)</b> | <b>cantidad</b> | <b>total (US\$)</b> |
| manufactura  | 3000                         | 1               | 3000                |
| costo total manufactura                                      |                              |                 | 3000                |
| <b>Sensores</b>  |                              |                 |                     |
| <b>item</b>  | <b>costo unitario (US\$)</b> | <b>cantidad</b> | <b>total (US\$)</b> |
| pulsadores   | 15                           | 12              | 180                 |
| sensores capacitivos   | 60                           | 2               | 120                 |
| selector de monedas  | 120                          | 1               | 120                 |
| costo total sensores   |                              |                 | 420                 |
| <b>Actuadores</b>  |                              |                 |                     |
| <b>item</b>  | <b>costo unitario (US\$)</b> | <b>cantidad</b> | <b>total (US\$)</b> |
| motor pap NEMA23   | 60                           | 7               | 420                 |
| motor pap NEMA24   | 70                           | 1               | 70                  |
| licuadora 600Watts   | 200                          | 1               | 200                 |
| electroválvula (NC)  | 500                          | 1               | 500                 |
| Display LCD  | 15                           | 1               | 15                  |
| circulina  | 15                           | 1               | 15                  |
| costo total actuadores                                       |                              |                 | 1220                |
| <b>Implementación de circuitos electrónicos</b>              |                              |                 |                     |
| <b>item</b>  | <b>costo unitario (US\$)</b> | <b>cantidad</b> | <b>total (US\$)</b> |
| fuelle de poder  | 42                           | 1               | 42                  |
| circuito de control de motor pap                             | 31                           | 8               | 248                 |
| circuito de control  | 120                          | 1               | 120                 |
| costo total implementación de circuitos electrónicos         |                              |                 | 410                 |
|  |                              |                 | <b>total (US\$)</b> |
| <b>costo manufactura e insumos de implementación sistema</b> |                              |                 | <b>5050</b>         |

En la Tabla 5.1 se muestra el costo de la manufactura e insumos para producir un sistema autónomo inteligente selector de frutas, cabe resaltar que a este costo le falta agregar, el sueldo a los que lo desarrollen por el tiempo de trabajo, la patente, el transporte y algunos costos de producción.

También quisiera acotar que pasaría si fuéramos nosotros los que hagamos empresa con este sistema y lo desarrollemos nosotros, nos costaría US\$5050, para desarrollar un sistema como este.

Pero si desarrolláramos este sistema para venderlo, tendríamos que cobrar algunos costos más como por ejemplo, nuestro sueldo por producir este sistema, la patente de este sistema y gastos adicionales como los costos fijos.

#### **Sueldo de producción**

Este sistema se desarrolla en un mes de trabajo, la manufactura se desarrolla en 2 semanas (también en estas semanas se producen los circuitos electrónicos) y las otras dos semanas para instalación de sensores y pruebas del sistema, entonces el sueldo por producir un sistema así, le ponemos US\$950, para redondear hasta ahora los US\$6000 de costo total.

#### **Patentes y otros costos fijos**

En cuestión de patentes y otros costos fijos por producir un sistema le adicionamos US\$1000, con esto llegaríamos a US\$7000 de costo total del sistema implementado, siendo este el precio de venta de un sistema implementado.

Ahora veamos en la Tabla 5.2 el resumen total de los costos y el precio final con el que saldría a la venta este sistema.

Tabla 5.2 resumen total de costos y precio de venta del sistema

| <b>Manufactura del diseño mecánico</b>               |                              |                 |                     |
|--|------------------------------|-----------------|---------------------|
| <b>ítem</b>  | <b>costo unitario (US\$)</b> | <b>cantidad</b> | <b>total (US\$)</b> |
| manufactura  | 3000                         | 1               | 3000                |
| costo total manufactura                              |                              |                 | 3000                |
| <b>Sensores</b>                                      |                              |                 |                     |
| <b>ítem</b>  | <b>costo unitario (US\$)</b> | <b>cantidad</b> | <b>total (US\$)</b> |
| pulsadores   | 15                           | 12              | 180                 |
| sensores capacitivos                                 | 60                           | 2               | 120                 |
| selector de monedas                                  | 120                          | 1               | 120                 |
| costo total sensores                                 |                              |                 | 420                 |
| <b>Actuadores</b>                                    |                              |                 |                     |
| <b>ítem</b>  | <b>costo unitario (US\$)</b> | <b>cantidad</b> | <b>total (US\$)</b> |
| motor pap NEMA23                                     | 60                           | 7               | 420                 |
| motor pap NEMA24                                     | 70                           | 1               | 70                  |
| licuadora 600Watts                                   | 200                          | 1               | 200                 |
| electroválvula (NC)                                  | 500                          | 1               | 500                 |
| Display LCD  | 15                           | 1               | 15                  |
| circulina  | 15                           | 1               | 15                  |
| costo total actuadores                               |                              |                 | 1220                |
| <b>Implementación de circuitos electrónicos</b>      |                              |                 |                     |
| <b>ítem</b>  | <b>costo unitario (US\$)</b> | <b>cantidad</b> | <b>total (US\$)</b> |
| fuelle de poder                                      | 42                           | 1               | 42                  |
| circuito de control de motor pap                     | 31                           | 8               | 248                 |
| circuito de control                                  | 120                          | 1               | 120                 |
| costo total implementación de circuitos electrónicos |                              |                 | 410                 |
| <b>Sueldo personal</b>                               |                              |                 |                     |
| <b>ítem</b>  | <b>costo unitario (US\$)</b> | <b>cantidad</b> | <b>total (US\$)</b> |
| sueldo personal                                      | 950                          | 1               | 950                 |
| <b>Patente y otros costos fijos</b>                  |                              |                 |                     |
| <b>ítem</b>  | <b>costo unitario (US\$)</b> | <b>cantidad</b> | <b>total (US\$)</b> |
| patente y otros costos fijos                         | 1000                         | 1               | 1000                |
|  |                              |                 | <b>total (US\$)</b> |
| <b>precio de venta sistema</b>                       |                              |                 | <b>7000</b>         |

Es este precio de venta al cual venderíamos el sistema implementado a las personas que quisieran hacer empresa con este sistema. En la siguiente sección les mostrare un estudio de cómo hacer empresa con este sistema.

### **5.3. La Empresa y el Sistema Selector de Frutas**

Vamos a desarrollar este estudio para cuando el sistema solo prepare jugo surtido.

#### **5.3.1 Oferta del Producto**

El producto a ofrecer son jugos de frutas, como: jugo de papaya, jugo de piña, jugo de manzana, jugo de plátano y un jugo surtido de todas las frutas mencionadas además de una porción de melón, todo estos jugos preparados por un novedoso sistema expendedor de jugos de fruta, que no requieren de un personal para la venta de los jugos, solo se requiere el personal para llenar de frutas peladas y picadas al sistema.

#### **5.3.2 Factores de Producción**

Para los jugos que se desea vender, lo que se requieren son:

##### **Insumos**

Para el cálculo de la cantidad de estos insumos, lo datos lo sacamos de la sección 2.2.1 del volumen del pistón de corte de frutas por la densidad de la fruta pelado y picado.

- Papaya: para un jugo de papaya se requiere de 5 porciones de 0,11Kg de papaya pelado y picado, por tanto se requiere para hacer un jugo de papaya 0.55Kg de papaya pelado y picado, pero para hacer un jugo surtido, entra 0,11kg de papaya pelado y picado.

- piña: para un jugo de piña se requiere de 5 porciones de 0,11Kg de piña pelado y picado, por tanto se requiere para hacer un jugo de piña 0.55Kg de piña pelado y picado, pero para hacer un jugo surtido, entra 0,11kg de piña pelado y picado.
- manzana: para un jugo de manzana se requiere de 5 porciones de 0,06Kg de piña pelado y picado, por tanto se requiere para hacer un jugo de piña 0.3Kg de manzana pelado y picado, pero para hacer un jugo surtido, entra 0,06kg de manzana pelado y picado.
- plátano: para un jugo de plátano se requiere de 4 porciones de 0,11Kg de plátano pelado y picado, por tanto se requiere para hacer un jugo de plátano 0.44Kg de plátano pelado y picado, pero para hacer un jugo surtido, entra 0,11kg de plátano pelado y picado.
- Melón: solo entra en el jugo surtido con una porción de 0.06Kg
- Líquido: es un líquido preparado con agua hervida, azúcar, canela, clavo, etc.

Para preparar un jugo surtido se requiere de 0.11Kg de papaya, 0.11Kg de piña, 0.06Kg de manzana, 0.11Kg de plátano 0.06Kg de melón y 0.5 litros de líquido.

#### **Sistema, equipos de corte y pelado de frutas**

- Sistema autónomo inteligente selector de frutas
- Cuchillo
- Tablero de corte

- Recipientes de manejo de las frutas

### **Accesorios de despacho del jugo**

- Vasos de polipapel con sus tapas
- Sorbetes

### **Infraestructura**

En lo que respecta a infraestructura puedes alquilar un local donde no solo lo utilices para dejar al sistema trabajando, puedes utilizar el local para vender otras productos como: sándwiches, pasteles, tortas, kekes, etc.

Nosotros solo vamos a sacar un costo de alquiler del local solo como para el uso del sistema

Lo más importante es la ubicación de este local ya que es un factor muy influyente en el número de ventas que se realizara y tiene que estar ubicada en una zona comercial.

### **Personal**

Solo se requiere de un personal, para que pele y pique la fruta y para que al final del día le dé una limpieza, prácticamente eso lo realizara en 3 horas de trabajo así es que el costo de pagar al personal solo será por 3 horas.

### 5.3.3 Inversión

- Sistema autónomo inteligente selector de frutas: tiene un costo de US\$7000 más el 19% por IGV nos da US\$8330 que en soles según una tasa de cambio actual de S/.2.85 el dólar nos da en promedio S/.23740.
- Cuchillo y tablero para picar frutas: S/.100
- Recipientes para el manejo de las frutas: S/.100
- Gastos legalización empresa(registros públicos), publicidad, otros: S/.1060

Inversión total: S/.25000

### Crédito para la inversión

Aproximamos un préstamo alrededor de S/.25000 que se pagaran en un periodo de 4 años, según la entidad crediticia consultada la cuota mensual a pagar es de S/.800.

### 5.3.4 Cálculo de la Estructura de Costos y Precio de Referencia

El precio de venta de un jugo surtido está basado esencialmente en tres factores: costos fijos (CF), costo marginal ( $\alpha$ ) y el precio de referencia (p) del mercado en el cual será ofertado el producto.

#### Costos Fijos (CF)

- Local: alquiler mensual (independiente del pago de agua y luz) → S/.300.00.

- Salarios: para un personal que pele, pique, coloque las frutas y realice la limpieza al final del día, es un trabajo de 3 horas por día, el pago es de S/.300 mensuales.
- Servicios:
  - Agua: S/.50 mensuales
  - Luz: S/.100 mensuales
- Transporte: para la compra de las frutas y otros insumos, costo mensual de S/.50.
- Mantenimiento sistema: S/.50 mensuales
- Cuota bancaria: S/.800 mensuales

Sumando todo tenemos el costo fijo (CF) mensual=S/.1650 y multiplicando por 12 para sacar un costo fijo anual tenemos: CF anual= S/.19800.

### **Costo Marginal ( $\alpha$ )**

Para poder determinar nuestro costo marginal requerimos conocer nuestros costos variables durante la venta de los jugos mensualmente.

El sistema está hecho para vender 50 vasos de jugo surtido diarios, esto con solo colocar al sistema 5.5Kg de papaya pelado y picado, 5.5Kg de piña pelado y picado, 3Kg de manzana pelado y picado, 5,5Kg de plátano pelado y picado, 3 Kg de melón pelado y picado y 25Kg de líquido.

En la Tabla 5.3 se muestra el gasto total en insumos por un mes, esto es por los 1500 jugos surtidos que se prepararán en 30 días (1 mes).



Tabla 5.3 Gasto total en insumos por un mes

| insumo                        | cantidad(Kg)/mes    | precio x mayor (S/.) el kilo(pelado y picado) | Gasto mensual (S/.)        |
|-------------------------------|---------------------|---|----------------------------|
| papaya                        | 165,00              | 1,50  | 247,50                     |
| piña                          | 165,00              | 1,50  | 247,50                     |
| manzana                       | 90,00               | 1,20  | 108,00                     |
| plátano                       | 165,00              | 1,00  | 165,00                     |
| melón                         | 90,00               | 1,00  | 90,00                      |
| líquido                       | 750                 | 0,5   | 375,00                     |
|                               | <b>cantidad (u)</b> | <b>precio x millar (S/.)</b>                  | <b>Gasto mensual (S/.)</b> |
| vasos con tapas               | 1500                | 200   | 300,00                     |
| sorbetes                      | 1500                | 4   | 6                          |
| <b>gasto total en insumos</b> |                     |   | <b>1539,00</b>             |

Por lo tanto nuestro costo marginal va a ser la división del gasto total en insumos por un mes entre la cantidad de jugos surtidos producidos en un mes, esto es  $1539/1500=1.026$ , entonces:

$$\alpha = S/.1.026$$

### Precio de referencia del mercado

Observando en los principales establecimientos de ventas de jugos, notamos que un litro de jugo surtido varía entre 3 y 4 soles, dependiendo de la localidad, para nuestro sistema le vamos a poner un costo de 3 soles por 946ml de jugo (32 onzas), que es un costo bajo para un sistema novedoso y automático, depende de cada empresa ponerle el costo del jugo, entonces:

$$p = S/.3$$

### 5.3.5 Estimación de la Producción de Equilibrio (Qe)

Ya tenemos el costo fijo (CF), el costo marginal ( $\alpha$ ) y el precio de referencia (p); con estos podemos hallar las ventas de equilibrio (Qe), según:

$$Q_e = CF / (p - \alpha) = 836 \text{ Vasos de jugo surtido mensuales}$$

Esto quiere decir que vendiendo 836 vasos de jugo surtido al mes o un promedio de 28 vasos de jugo surtido al día no tendríamos ni ganancias ni pérdidas, vendiendo más de esta cantidad diaria, empezaríamos a ganar.

El sistema está diseñado para preparar 50 jugos surtidos, si todos sus recipientes de las frutas y el líquido están llenos. Con vender los 50 jugos surtidos, estaríamos ganando el costo de 22 vasos de jugos, según nuestro costo del producto lo hemos puesto S/.3, por lo tanto estarías ganando S/.66 diarios, S/.1980 mensuales y si fuera uno mismo quien pele y pique las frutas y los deje listo para el preparado de los jugos, le aumentaríamos a esta ganancia S/.300 mas, porque es lo que pusimos en el costo fijo del sueldo del empleado. Con esto sería en total un aproximado de S/.2300 mensuales de ganancia y esto solo hasta los 4 años cuando se termine de pagar la inversión (cuotas bancarias), luego serán mayores las ganancias.

### **5.3.6 Meta de Utilidades o Ganancias**

Vamos a analizar las ganancias anuales que nos puede dejar al trabajar con este sistema, también vamos a ver algunos indicadores para mostrar si la inversión es buena o mala.

Sabemos que nuestro costo fijo anual sin la cuota bancaria es S/.10200 y nuestra cuota bancaria anual es S/.800 x 12meses = S/.9600 anuales, vamos a analizar todos estos costos, primero a partir del año 1 al 4to año que es donde terminamos de pagar las cuotas bancaria y luego del 5to año a más.

En la Tabla 5.4 se muestra en resumen la cantidad estimada a vender y las ganancias que generaría anualmente.

Tabla 5.4 Meta de utilidades anuales

|  |                  |
|--|------------------|
| <b>Estado del 1er año al 4to año</b>   | <b>S/.</b>       |
| Costo fijo infraestructura, personal , servicios, transporte y mantenimiento ANUAL | 10200,000        |
| cuota bancaria ANUAL   | 9600,000         |
| <b>COSTO FIJO ANUAL</b>  | <b>19800,000</b> |
|  |                  |
| <b>Estado del 5to año a mas</b>  | <b>S/.</b>       |
| Costo fijo infraestructura, personal , servicios, transporte y mantenimiento ANUAL | 10200,000        |
| cuota bancaria ANUAL   | 0,000            |
| <b>COSTO FIJO ANUAL</b>  | <b>10200,000</b> |
|  |                  |
|  | <b>S/.</b>       |
| <b>COSTO MARGINAL O VARIABLE</b>   | <b>1,026</b>     |
| <b>PRECIO DE VENTA</b>   | <b>3,000</b>     |
|  |                  |
| <b>PUNTO DE EQUILIBRIO</b>   | <b>CANTIDAD</b>  |
| PUNTO DE EQUILIBRIO ANUAL DE 1er a 4to año   | 10030,400        |
| <b>PUNTO DE EQUILIBRIO ANUAL DEL 5to año a mas</b>                                 | <b>5167,173</b>  |
|  |                  |
| <b>META DE PRODUCCION</b>  | <b>CANTIDAD</b>  |
| PRODUCCION META ANUAL DE 1er a 4to año   | 18250,000        |
| <b>PRODUCCION META ANUAL DEL 5to año a mas</b>                                     | <b>18250,000</b> |
|  |                  |
| <b>UTILIDADES ANUALES SEGÚN META</b>   | <b>S/.</b>       |
| META UTILIDADES ANUAL DE 1er a 4to año   | 16225,500        |
| <b>META UTILIDADES ANUAL DEL 5to año a mas</b>                                     | <b>25825,500</b> |

Con esta tabla nos podemos dar cuenta de las ganancias anuales que genera trabajar con este sistema y más todavía desde el inicio del 5to año porque ya no pagamos las cuotas bancarias, también nuestras metas de producción son justas, pero podríamos ponernos unas metas de producción más grandes y las ganancias serian aun mejor.

### **5.3.7 Evaluación Económica**

#### **Valor Actual Neto (VAN)**

Es la diferencia entre todos los ingresos y los egresos actualizados al período actual. Según el criterio del valor actual neto el proyecto debe aceptarse si su valor es positivo. El tiempo evaluado es de 5 años y la tasa de descuento es de 12%.

#### **Tasa Interna de Retorno (TIR)**

Es aquella tasa que hace que el valor actual neto sea igual a cero. El proyecto debe aceptarse cuando la tasa interna de retorno sea mayor a la tasa de descuento. El tiempo a evaluar se mantiene igual.

#### **Tiempo de Recuperación ( $t_R$ )**

Consiste en determinar el período para la recuperación de la inversión inicial. A menor tiempo se considera al proyecto más rentable.

En la Tabla 5.5 se muestra la evaluación económica para el proyecto de crear una empresa con el sistema diseñado en esta tesis, es muy importante para el inversionista saber si es factible invertir en este proyecto y justamente para eso se calculan estos indicadores.

Tabla 5.5 Evaluación económica

|                 |           | S/.        |
|-----------------|-----------|------------|
| INVERSION       |           | -25000,000 |
| AÑO             | UTIL/AÑO  | FLUJO UTIL |
| 1               | 16225,500 | 14487,05   |
| 2               | 16225,500 | 12934,87   |
| 3               | 16225,500 | 11549      |
| 4               | 16225,500 | 10311.6    |
| 5               | 25825,500 | 14654,08   |
| <b>VAN</b>      |           | 28625,000  |
| <b>TIR</b>      |           | 22%        |
| <b>tr(años)</b> |           | 1,54       |

Según el estudio económico revela que la evaluación posee un VAN > 0 y el TIR > 12%. Se concluye que para las consideraciones realizadas, el proyecto de crear empresa con este sistema se presenta muy atractivo, debido a que todo empieza desde un posible préstamo por una entidad bancaria y luego buscar un local donde posicionar este sistema.

## CONCLUSIONES

- ❖ El diseño e implementación del sistema cumple con el objetivo principal de esta tesis, el de desarrollar un sistema autónomo e inteligente para el preparado de jugos de frutas frescas y se venda sin la presencia de un dependiente para cobrar por el jugo; a la vez que se demuestra que es económicamente rentable, debido a que evaluado en un periodo de 5 años nuestro VAN >> 0, nuestro TIR > 12% (tasa de descuento) y nuestro tiempo de recuperación de la inversión es de 1 año y 6 meses, este tiempo es corto y es menor al tiempo del periodo de pago del préstamo bancario, además que toda la inversión es un préstamo bancario, hasta se podría vender el sistema financiado, se cobraría 800 soles mensuales en un periodo de 4 años a la persona que desea hacer un negocio con este sistema, luego de los 4 años sería completamente suyo.
- ❖ Se demuestra que nuestro diseño del sistema mecatrónico de selección y corte de frutas, garantiza que las porciones de frutas lleguen a la licuadora para el licuado de las frutas y cada porción de fruta se puede llevar a la licuadora con el uso de un motor paso a paso y toda su electrónica que éste requiere.
- ❖ El diseño mecánico del sistema selector de frutas, es un diseño que se puede utilizar para otras máquinas, con la misma función pero para diferentes productos, por ejemplo podría cortar frutas para hacer ensaladas de frutas, separar arroz cocido, etc. Además el sistema es higiénico, útil y novedoso, por todo esto tiene un valor agregado.

- ❖ Se demuestra que nuestro diseño del sistema mecatrónico de giro de la licuadora es capaz de verter el jugo a un vaso sin la utilización de muchos actuadores, simplemente con el uso de un motor paso a paso y toda su electrónica que éste requiere.
- ❖ Se redujeron los costos de implementación con la utilización de motores paso a paso en vez de pistones, que requieren de electroválvulas, un compresor y mayor espacio. También eliminando un sensor ultrasonido para medir el volumen del líquido vaciado a la licuadora y solucionándolo con el control de tiempos de apertura de la electroválvula. Así como también la utilización de un microcontrolador de bajo costo en vez de controladores industriales como PLC's y otros dispositivos de alto costo, además se utiliza selectores de monedas comparadores, un simple display LCD de 16x2 y unos pulsadores para que el usuario pueda interactuar con nuestro sistema sin mucha dificultad, el cual no genera un alto costo de implementación.
- ❖ La tecnología dsPIC intenta ser una respuesta eficaz a las necesidades de las modernas aplicaciones que combinan las funciones típicas de los microcontroladores con las del procesamiento digital de señales de los DSP, para nuestro sistema no es necesario utilizar las prestaciones del DSP de esta tecnología, se podría concluir que era suficiente un microcontrolador PIC, pero el hecho de que esta tecnología tenga el mismo costo de un microcontrolador PIC, que además se programa en un lenguaje C de alto nivel y nos abra un mundo de posibilidades de aplicaciones, hace que sea posible olvidarnos de la existencia de los microcontroladores PIC.

- ❖ Según el diagrama de tiempos mostrado el sistema puede preparar un jugo surtido en 90 segundos que es mucho menor a la preparación artesanal de un jugo surtido en cualquier juguería que normalmente se demoran más de 3 minutos.
  
- ❖ Para el estudio económico le pusimos una meta de 50 vasos de jugo diarios, pero podríamos ser más ambiciosos y ponernos una meta mayor, dependiendo del lugar donde dejemos este sistema, inclusive podemos bajar el precio referencial y ponernos una meta aun mayor, el estudio económico está hecho como para que se analice muchos casos.



## RECOMENDACIONES

A este diseño se le pueden agregar algunas mejoras, como por ejemplo, colocarles sensores de fines de carrera a cada mecanismo de corte de las frutas, para realizar el conteo de giros (vueltas) del mecanismo de corte y según esto determinar la cantidad de fruta que se consumió, si en caso el sensor haya contado 50 vueltas, para una determinada fruta, el sistema tenga la capacidad de comunicar que ya no puede preparar un jugo de la fruta que se terminó.

También el controlador del sistema, el dsPIC, tiene la capacidad de interactuar con un puerto Ethernet y a través de este, introducimos al internet, para monitorear nuestro sistema, en caso se acabe una fruta, mediante este monitoreo podremos saber el lugar y la fruta que tenemos que reponer.

Inclusive le podemos adicionar un panel, una pantalla, en donde mientras se esté preparando el jugo, se vaya visualizando un chiste, una broma, una cámara escondida, etc. Para distraer al que va a consumir el jugo y no esté pendiente del sonido de la licuadora.

Si se desea llevar nuestro sistema para un lugar donde la temperatura es mayor a 20 grados, como la costa o la selva, entonces se le puede adicionar un sistema de refrigeración para que las frutas se conserven.

**BIBLIOGRAFÍA**

- [1] Calero Pérez, Roque y Carta González, José, “*Fundamentos de mecanismos y máquinas para ingenieros*”, McGRAW-HILL/INTERAMERICANA DE ESPAÑA, S.A.U., Primera edición, 1999.
- [2] Ferdinand P. Beer; E. Russell Johnston, Jr. y Elliot R.Eisenberg, “*Mecánica vectorial para ingenieros, Estática*”, McGRAW-HILL/INTERAMERICANA DE ESPAÑA, S.A.U., Octava edición, 2007.
- [3] Ferdinand P. Beer; E. Russell Johnston, Jr. y William E. Clausen, “*Mecánica vectorial para ingenieros, Dinámica*”, McGRAW-HILL/INTERAMERICANA DE ESPAÑA, S.A.U., Octava edición, 2007.
- [4] A.L. Casillas, “*Máquinas cálculos de taller*” COPYRIGHT BY EDICIONES <MÁQUINAS>, España.
- [5] Coughlin, Robert y Driscoll, Frederick, “*Amplificadores operacionales y circuitos integrados lineales*”, PRENTICE-HALL HISPANOAMERICANA, S.A., 1993.
- [6] Angulo Usategui, José; Etxebarría Ruiz, Aritza; Angulo Martínez, Ignacio y Trueba Parra, Iván, “*dsPIC Diseño práctico de aplicaciones*”, McGRAW-HILL/INTERAMERICANA DE ESPAÑA, S.A.U., Primera edición, 2008.

- [7] O G POPA, “*LEARN HARDWARE FIRMWARE AND SOFTWARE DESIGN*” Corollary Theorems Ltd, Primera edición, 2005.
- [8] Microchip, “*Soluciones rápidas de Microchip*”, Microchip Technology Inc., 2009.
- [9] Bruno Saravia, Bruno, “*Programación en dsPIC*”, [clases audiovisuales], Programa certificado por Microchip Technology Inc. 36 horas de clases.
- [10] “*Máquina expendedora*”, Wikipedia  
Página web: [http://es.wikipedia.org/wiki/M%C3%A1quina\\_expendedora](http://es.wikipedia.org/wiki/M%C3%A1quina_expendedora)
- [11] “*Manual de sistema de motores paso a paso*”, SureStep, catalogue.  
Página web:  
<http://www.automationdirect.com/static/manuals/surestepmanualsp/surestepmanualsp.pdf>
- [12] “*Selecting a stepping motor*” y “*An introduction to stepping motors*”, Oriental Motor, General catalogue.  
Página web:  
<http://www.orientalmotor.de/de/uploads/documents/21122005145147.pdf>
- [13] “*Motores paso a paso*”, Departamento de tecnología, Gonzalo Correas.  
Página web:  
<http://jmdiezm.iespana.es/documentos/motorelectric/motorespap.pdf>
- [14] “*12 Volt 30Amp Power Suply*”, Circuit Exchange international.  
Página web: <http://www.zen22142.zen.co.uk/Circuits/Power/1230psu.htm>
- [15] “*Control automático de temperatura*”, Oscar Montoya y Alberto Franco.  
Página web:  
<http://www.videosistemas.com/proyectos/Control%20Automatico%20de%20Temperatura.pdf>

**ANEXO A**  
**PLANO DEL DISEÑO MECÁNICO**

**ANEXO B**

**PINADO DEL DSPIC32FJ64GS606**

## Pines de los Microcontroladores

Está claro que el dsPIC tienen muchas funciones atractivas integradas en sus pines, y lo primero que se debe hacer es identificar a que pines son asignados, para luego diseñar los circuitos conectados a los pines en particular. Ahora vamos a analizar los pines y sus funciones del dsPIC seleccionado, veamos lo que tiene en general el dsPIC33FJ64GS606 y esto se muestra en la Tabla B1

| Device           | Pins | Program Flash Memory (Kbytes) | RAM (Bytes)       | 16-bit Timer | Input Capture | Output Compare | UART | Quadrature Encoder Interface | SPI | ECAN™ | DMA Channels | PWM | Analog Comparator | External Interrupts | DAC Output | I <sup>2</sup> C™ | ADC  |                               |                          | I/O Pins | Packages |
|------------------|------|-------------------------------|-------------------|--------------|---------------|----------------|------|------------------------------|-----|-------|--------------|-----|-------------------|---------------------|------------|-------------------|------|-------------------------------|--------------------------|----------|----------|
|                  |      |                               |                   |              |               |                |      |                              |     |       |              |     |                   |                     |            |                   | SARs | Sample and Hold (S&H) Circuit | Analog-to-Digital Inputs |          |          |
| dsPIC33FJ64GS606 | 64   | 64                            | 9K <sup>(1)</sup> | 5            | 4             | 4              | 2    | 2                            | 2   | 1     | 4            | 6x2 | 4                 | 5                   | 1          | 2                 | 2    | 6                             | 16                       | 58       | PT, MR   |

Tabla B1 familia de controladores dsPIC33FJ64GS606

Vamos a analizar los pines del dsPIC33FJ64GS606 y como ellos están conectados a los módulos de diferentes funciones, note en la Figura B1 que cada pin tiene un número y algunos códigos, que describen sus funciones, por ejemplo el pin número 11 tiene 4 funciones, de los mas resaltantes tiene por ejemplo una entrada analógica llamado AN5, también una entrada o salida digital llamado RB5. Cuando un pin tiene múltiples funciones se tiene que habilitar esas funciones con el cual uno quiere trabajar y deshabilitar otros.

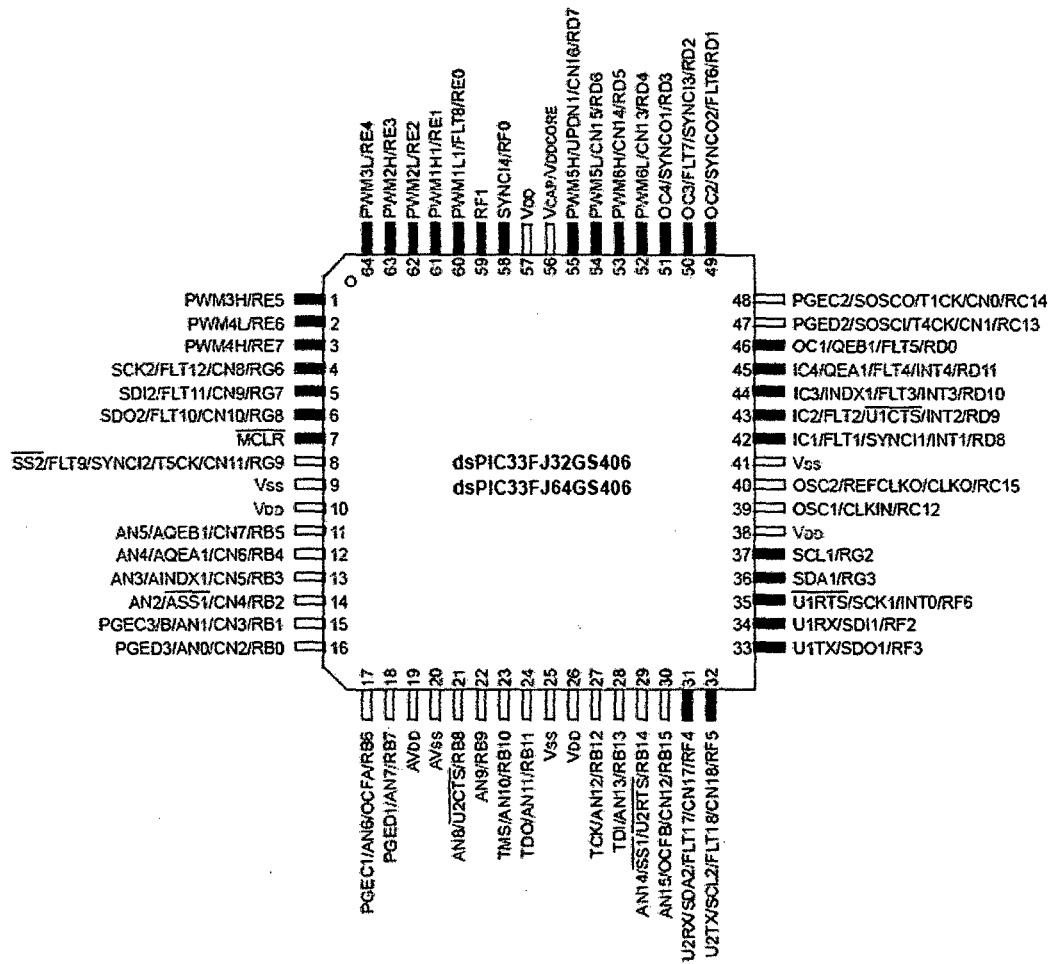


Figura B1 Diagrama de Pines dsPIC33FJ32MC204

TABLA 3.2 Descripción de los pines del dsPIC33FJ64GS606

| Nombre de pin       | Tipo de pin | Descripción  |
|---------------------|-------------|--|
| AN0-AN8             | I           | Canales de entrada analógica   |
| CLKI                | I           | Fuente externa de entrada de reloj, siempre asociado a la función pin OSC1.        |
| CLKO                | O           | Oscilador de cristal de salida, siempre asociado a la función pin OSC2             |
| OSC1                | I           | Oscilador de cristal de entrada.   |
| OSC2                | I/O         | Oscilador de cristal de salida. Opcionalmente funciona como CLKO en RC y modos CE. |
| SOSCI               | I           | 32,768 kHz oscilador de cristal de baja potencia de entrada.                       |
| SOSCO               | O           | 32,768 kHz oscilador de cristal de baja potencia de salida                         |
| CN0-CN30            | I           | Notificación del cambio de entradas.   |
| IC1-IC2             | I           | Captura de entradas 1/2.   |
| IC7-IC8             | I           | Captura de entradas 7/8.   |
| OCFA                | I           | Comparar error de entrada A (para Comparar los canales 1 y 2).                     |
| OC1-OC2             | O           | Comparar los resultados del 1 al 2   |
| INT0                | I           | Interrupción externa 0.  |
| INT1                | I           | Interrupción externa 1.  |
| INT2                | I           | Interrupción externa 2.  |
| RA0-RA4<br>RA7-RA10 | I/O         | PORTA es un puerto bidireccional de entrada y salida                               |
| RBO-RB15            | I/O         | PORTB es un puerto bidireccional de entrada y salida                               |
| RCO-RC9             | I/O         | PORTC es un puerto bidireccional de entrada y salida                               |
| T1CK                | I           | Timer1 de entrada de reloj externa.  |
| T2CK                | I           | Timer2 de entrada de reloj externa.  |
| T3CK                | I           | Timer3 de entrada de reloj externa.  |
| $\overline{U1CTS}$  | I           | UART1 limpiar para enviar.   |
| $\overline{U1RTS}$  | O           | UART1 listo para enviar.   |
| U1RX                | I           | UART1 recibir.   |
| U1TX                | O           | UART1 transmitir.  |
| SCK1                | I/O         | Serie síncrona de entrada y salida de reloj para SPI1                              |
| SDI1                | I           | SPI1 datos de entrada  |
| SDO1                | O           | SPI1 datos de salida   |
| $\overline{SS1}$    | I/O         | SPI1 esclavo de sincronización o marco de pulso de E/S                             |
| SCL1                | I/O         | Serie síncrona de entrada y salida de reloj para I2C1                              |
| SDA1                | I/O         | Serie síncrona de entrada y salida de datos para I2C1                              |
| ASCL1               | I/O         | Serie síncrona alterna de entrada y salida de reloj para I2C1                      |
| ASDA1               | I/O         | Serie síncrona alterna de entrada y salida de datos para I2C1                      |
| TMS                 | I           | JTAG Prueba modo de selección de pin.  |
| TCK                 | I           | JTAG Prueba pin de entrada de reloj.   |
| TDI                 | I           | JTAG Prueba pin de entrada de datos.   |



|                    |     |  |
|--------------------|-----|--|
| TDO                | O   | JTAG Prueba pin de salida de datos.  |
| INDX               | I   | Quadrature Encoder índice del pulso de entrada.  |
| QEA                | I   | Quadrature Encoder fase A de entrada en modo QEI.<br>Temporizador externo auxiliar de entrada reloj/puerta en modo temporizador. |
| QEB                | I   | Quadrature Encoder fase B de entrada en modo QEI.<br>Temporizador externo auxiliar de entrada reloj/puerta en modo temporizador. |
| UPDN               | O   | Posición arriba/abajo del contador de dirección de estado  |
| $\overline{FLTA1}$ | I   | PWM1 Falla de entrada A  |
| PWM1L1             | O   | PWM1 salida en nivel bajo 1  |
| PWM1H1             | O   | PWM1 salida en nivel alto 1  |
| PWM1L2             | O   | PWM1 salida en nivel bajo 2  |
| PWM1H2             | O   | PWM1 salida en nivel alto 2  |
| PWM1L3             | O   | PWM1 salida en nivel bajo 3  |
| PWM1H3             | O   | PWM1 salida en nivel alto 3  |
| $\overline{FLTA2}$ | I   | PWM2 Falla de entrada A  |
| PWM2L1             | O   | PWM2 salida en nivel bajo 1  |
| PWM2H1             | O   | PWM2 salida en nivel alto 1  |
| PGD1/EMUD1         | I/O | Pin de E/S de datos para programación/grabado canal de comunicación 1.   |
| PGD1/EMUD1         | I   | Pin de entrada de reloj para programación/grabado canal de comunicación 1.   |
| PGD2/EMUD2         | I/O | Pin de E/S de datos para programación/grabado canal de comunicación 2.   |
| PGD2/EMUD2         | I   | Pin de entrada de reloj para programación/grabado canal de comunicación 2.   |
| PGD3/EMUD3         | I/O | Pin de E/S de datos para programación/grabado canal de comunicación 3.   |
| PGD3/EMUD3         | I   | Pin de entrada de reloj para programación/grabado canal de comunicación 3.   |
| $\overline{MCLR}$  | I/P | Para resetear el dsPIC   |
| AVDD               | P   | Fuente positiva para los módulos analógicos, este pin siempre debe estar conectado   |
| AVSS               | P   | Referencia a tierra para los módulos analógicos.   |
| VDD                | P   | Fuente positiva para la lógica periférica y los pines de entrada y salida  |
| VDDCORE            | P   | Conexión de capacitor filtro para la lógica del CPU  |
| VSS                | P   | Referencia a tierra para la lógica y pines de entrada y salida.  |
| VREF+              | I   | Voltaje de referencia analógica (alto) entrada.  |
| VREF-              | I   | Voltaje de referencia analógica (bajo) entrada.  |

Legenda: I/O: entrada/ salida; P: poder.

En la Tabla 3,2 vemos las descripciones de cada función de los pines respectivos del dsPIC que se utilizan en esta parte del diseño electrónico.

**ANEXO C**

**DISEÑO DEL CIRCUITO IMPRESO DEL SISTEMA ELECTRÓNICO**

## **Diseño del Circuito Impreso**

Una vez terminado todo el diseño del circuito electrónico que controlara el sistema, se tiene que llevar todo ese diseño a un circuito impreso para realizar el montaje de todos los dispositivos que operarán para controlar el sistema.

En nuestro sistema electrónico tendremos varios circuitos impresos que cumplirán distintas funciones, como la fuente de poder que irá solo con varias salidas de 3.3V, 5V, 12V y 24V, el circuito de control con los dspIC y las entradas y salidas digitales. Por último, los 8 circuitos de control y potencia de los motores paso a paso.

El Proteus no solo es un software de simulación de circuitos también tiene un creador de circuitos impresos que es el ARES del Proteus, con el cual podemos crear circuitos impresos de los diseños de los circuitos creados en el Proteus.

De hecho que el diseño de circuitos electrónicos en el Proteus para simulación es diferente que un diseño de circuitos impresos, porque acá ya tenemos que ver en que encapsulado quieres que vayan tus circuitos integrados, el espacio para tus disipadores de calor, los terminales para conectarlos con otros circuitos impresos, el tamaño de todo el circuito impreso, si lo vas a realizar en placas de doble cara o una cara, el grosor adecuado de cada pista, etc.

Veamos entonces nuestro circuitos impreso de nuestra fuente poder y se muestra en la Figura C1.

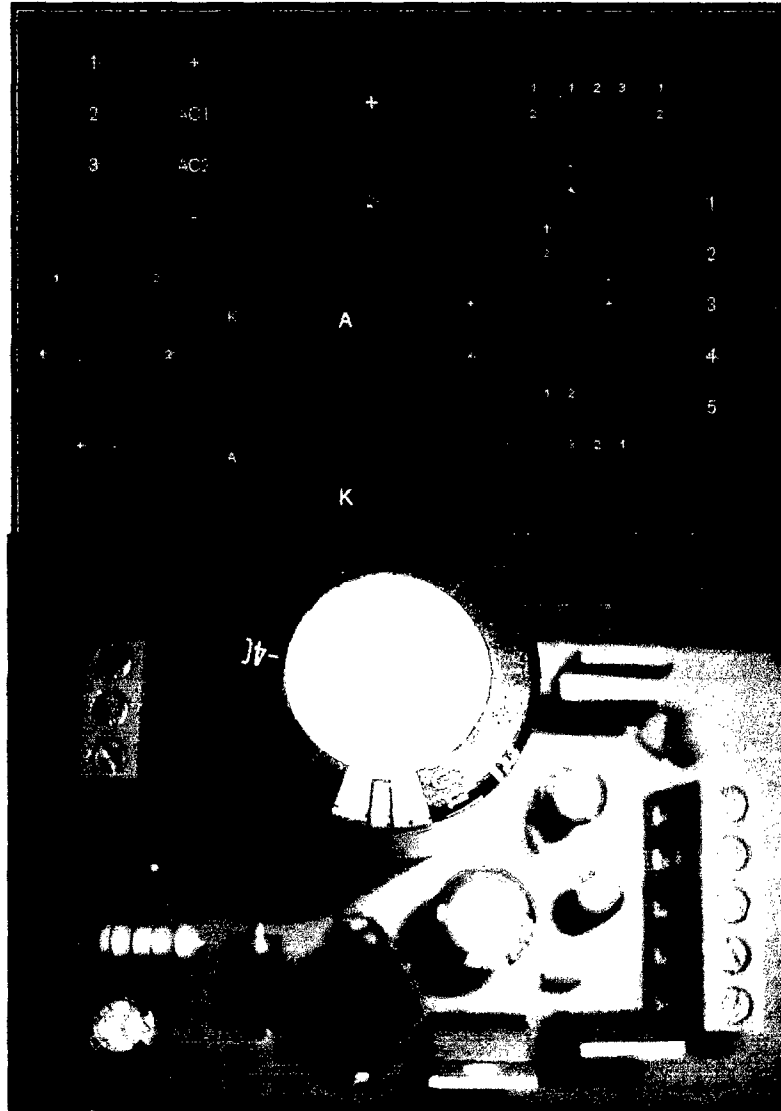


Figura C1 Circuito impreso fuente de poder y y su foto real implementado

El siguiente circuito impreso a mostrar es el circuito impreso del microcontrolador dsPIC, pero este circuito se muestra en dos partes el primero es la ampliacion de los pines de dsPIC y la otra es la conexion necesaria para un funcionamiento correcto, en la Figura C2 se muestra el circuito impreso de la ampliacion de los pines del dsPIC que tiene 64 pines, en realidad si no fuera tan complicado soldar estos micro pines, no necesitaria de esta ampliacion, pero es necesaria dada la tecnologia que tenemos en Perú.

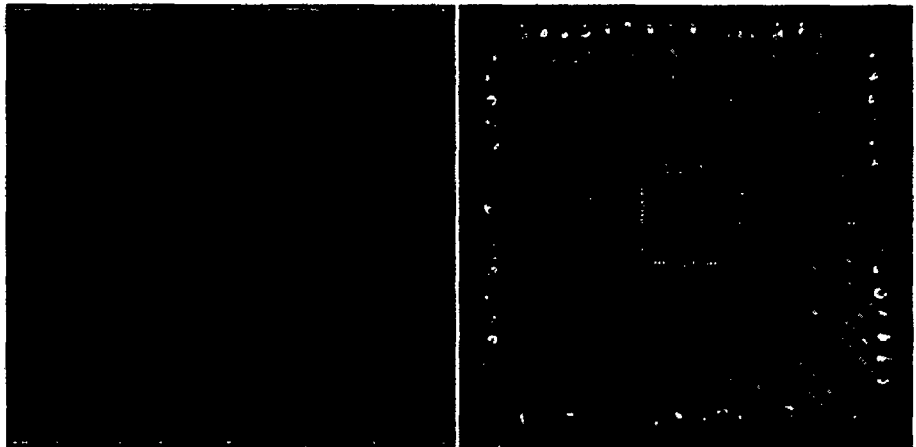


Figura C2 Circuito impreso e implementacion de la amplicion de pines dsPIC

En la Figura C3 mostramos el circuito impreso de la parte donde está la conexión necesaria para el correcto funcionamiento del dsPIC, en este circuito camos a ver el socket donde entrará la ampliacion del dsPIC, la fuente de 3.3V, el circuito del oscilador, el socket RJ11 hembra por donde se grabará mediante el ICD3 y unos socket aparte de 64 pines libres por donde saldrán las entradas o salidas digitales mediante cables hacia los demas circuitos.

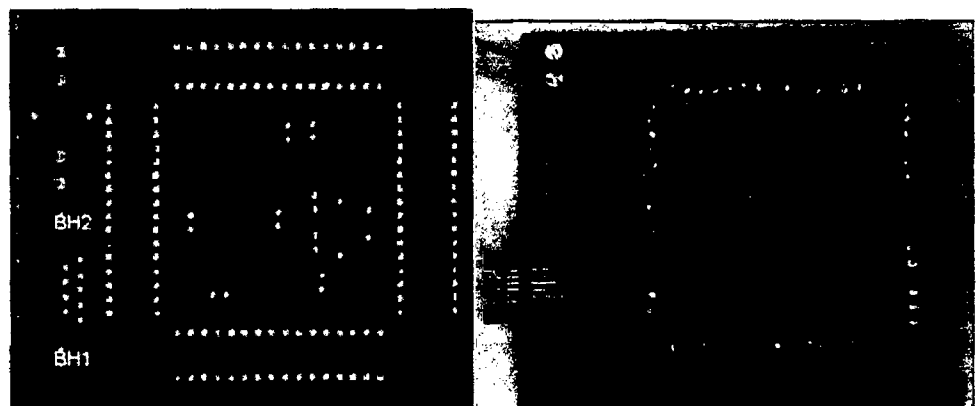


Figura C3 Circuito impreso e implementación del controlador dsPIC

Ahora veamos algunos circuitos impresos como entradas y salidas digitales que utilizamos en nuestro sistema. En la Figura C4 se muestra el circuito electrónico, el circuito impreso y la implementación de las entradas de los selectores de moneda.

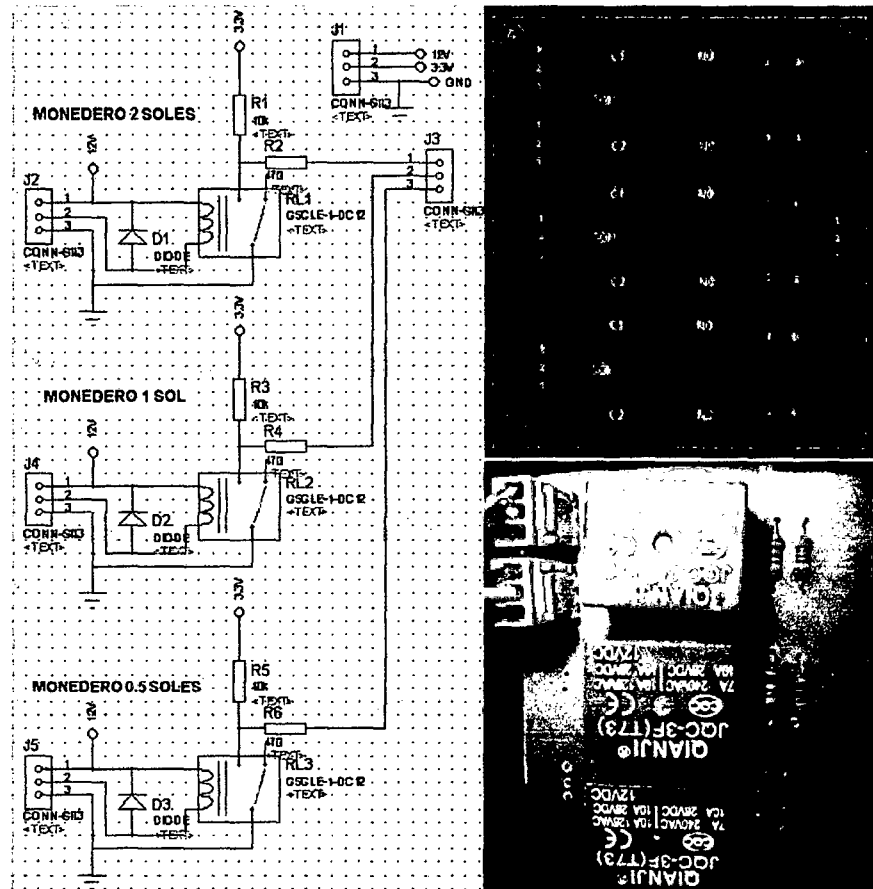


Figura C4 Circuito electrónico, PCB e implementación de monederos

En la Figura C5 se muestra el circuito electrónico, circuito impreso e implementación de los pulsadores e interruptores hacia el dsPIC, en realidad la conexión para un pulsador y un interruptor son lo mismo y son las conexiones típicas de los pulsadores, donde trabajan con 3.3V.

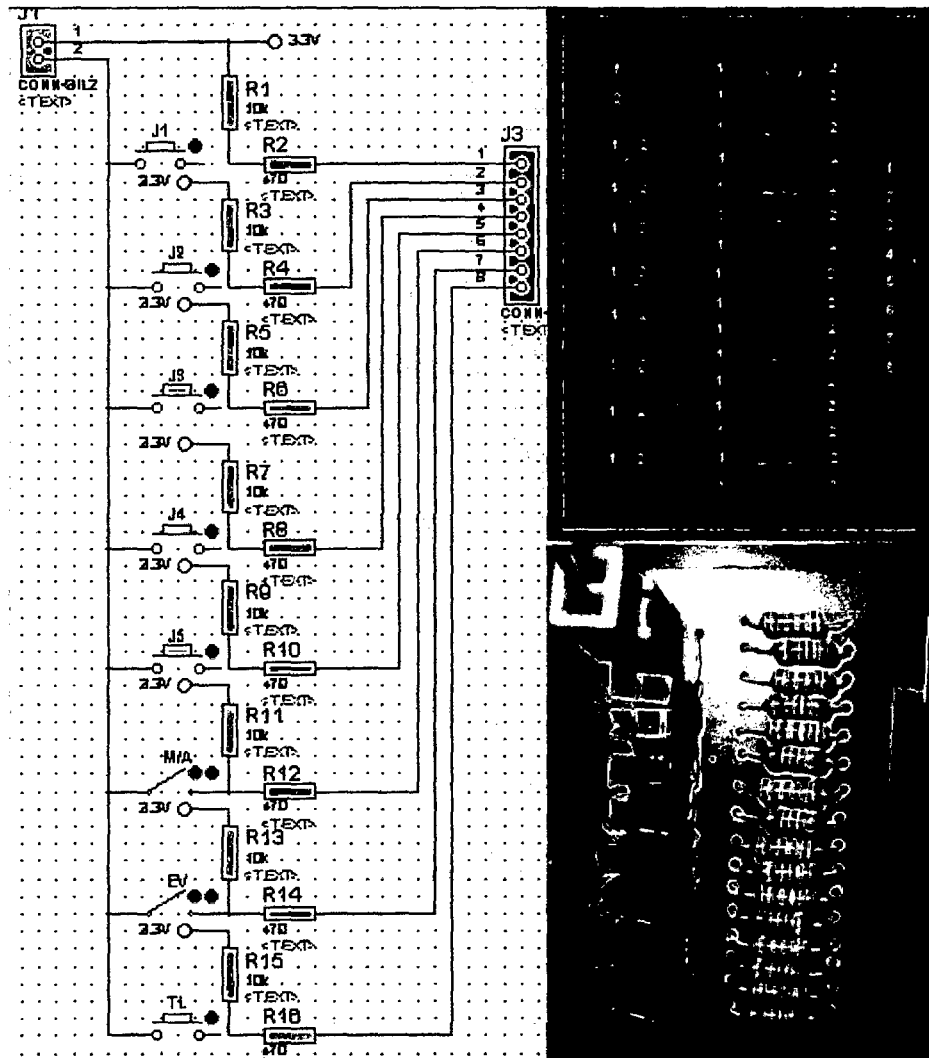


Figura C5 Circuito electrónico, PCB e implementación de pulsadores

Ahora veamos algunas salidas que empleamos para controlar nuestra licuadora, electroválvula, alarma, motores AC, motores paso a paso, etc.

En la Figura C6 se muestra el acondicionamiento de las salidas para encender la licuadora, la electroválvula y un motor AC que sirvira para garantizar la caída de las frutas.

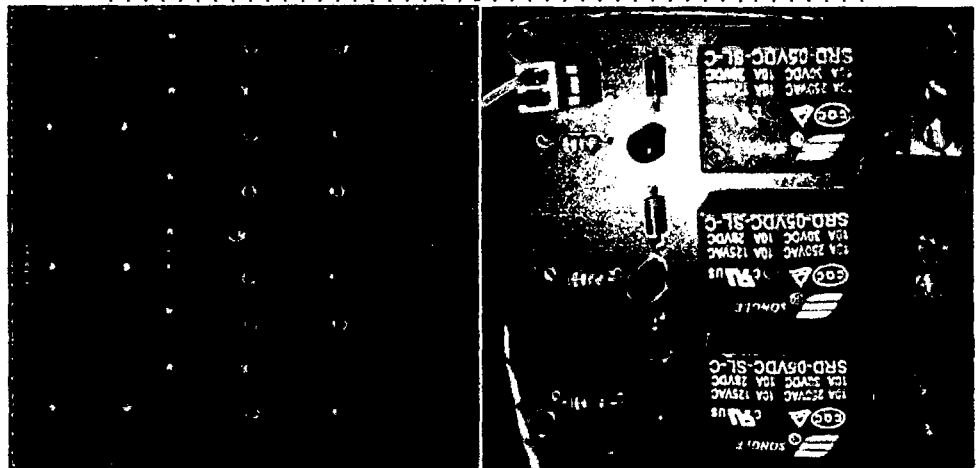
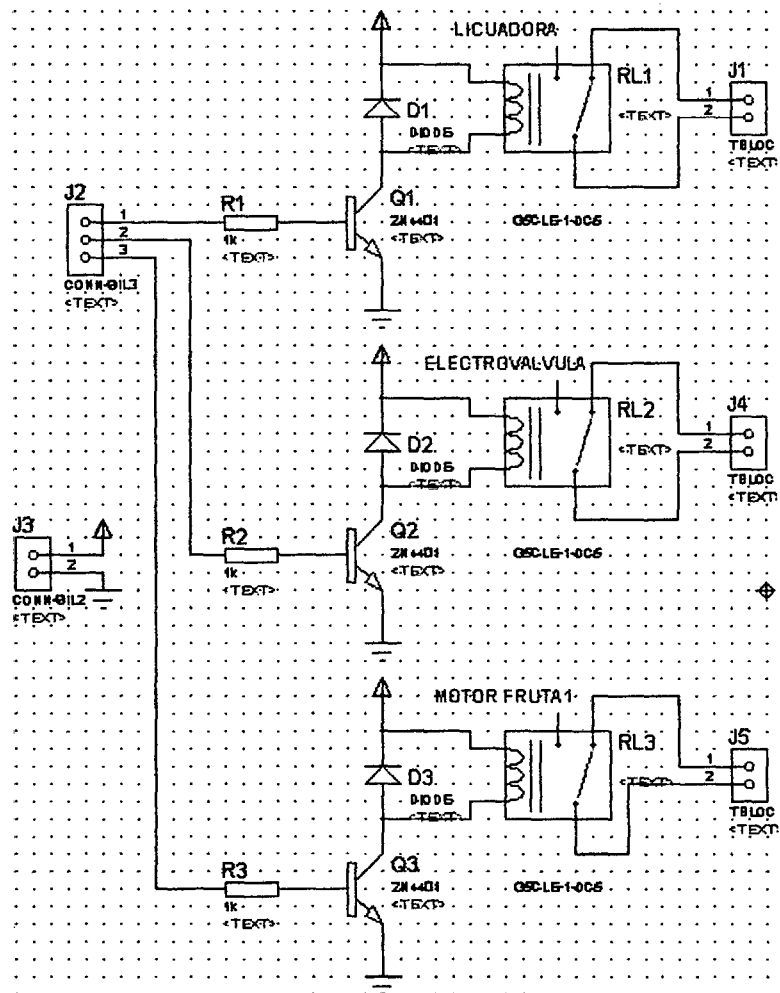


Figura C6 circuito electrónico, PCB e implementación de salidas con relay

La Figura C7 muestra el acondicionamiento de las salidas que servirán para interactuar con los controladores de los motores paso a paso, con este



circuito se controlarán la habilitación, el sentido de giro y los pulsos para hacer girar un motor paso a paso, este circuito esta diseñado para controlar dos motores paso a paso, debido a que para cada motor son tres señales y como se utiliza un 74HC14 que es un inverter y tiene para invertir 6 señales, justo para dos motores.

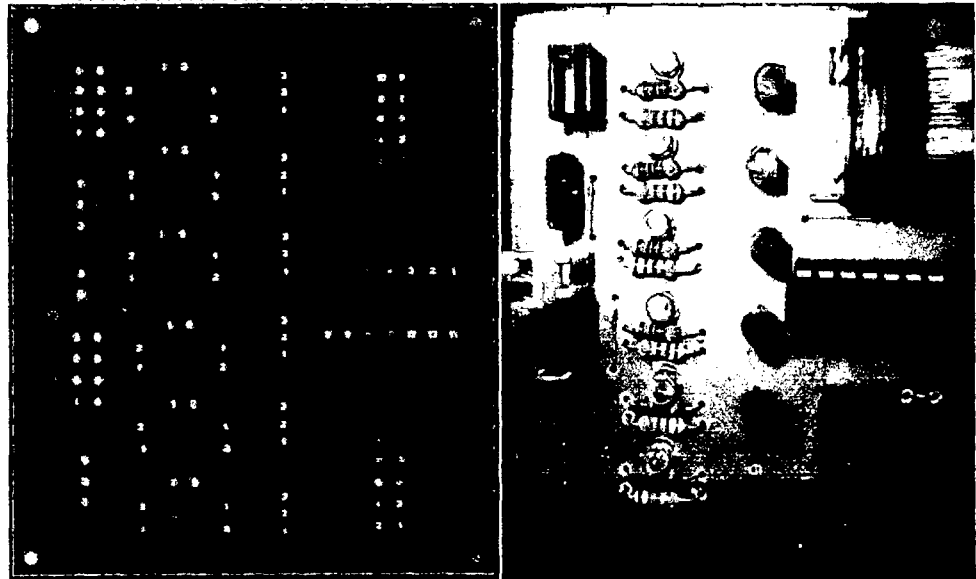
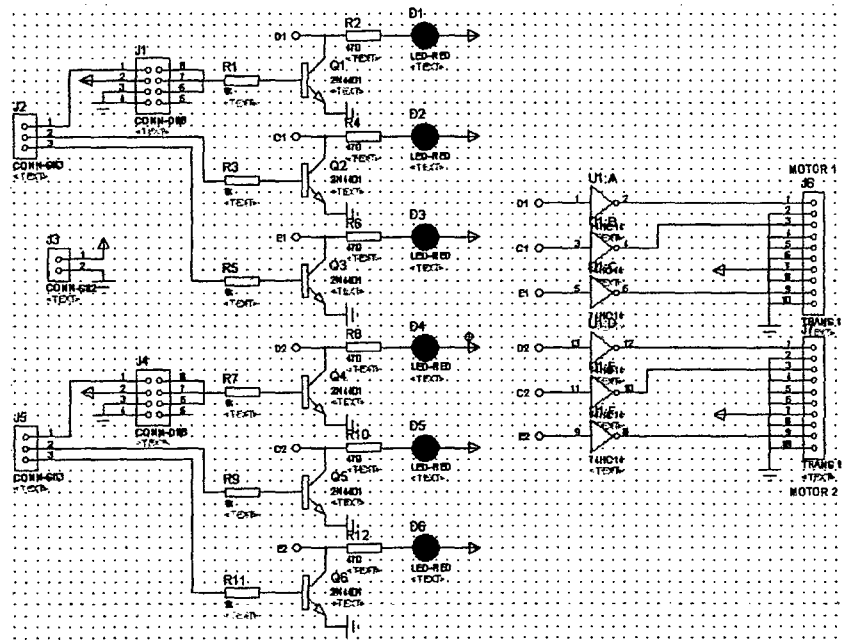


Figura C7 Circuito acondicionador hacia los controladores de los motores pap

El último circuito impreso que mostramos es el circuito de control de motores paso a paso, se tienen que realizar 8 de estos circuitos impresos, para controlar los 8 motores paso a paso que tiene nuestro sistema, en la Figura C8 mostramos el circuito impreso y la implementación de circuito de control de motores paso, debido que el circuito electrónico ya se muestra en la sección 3.6.

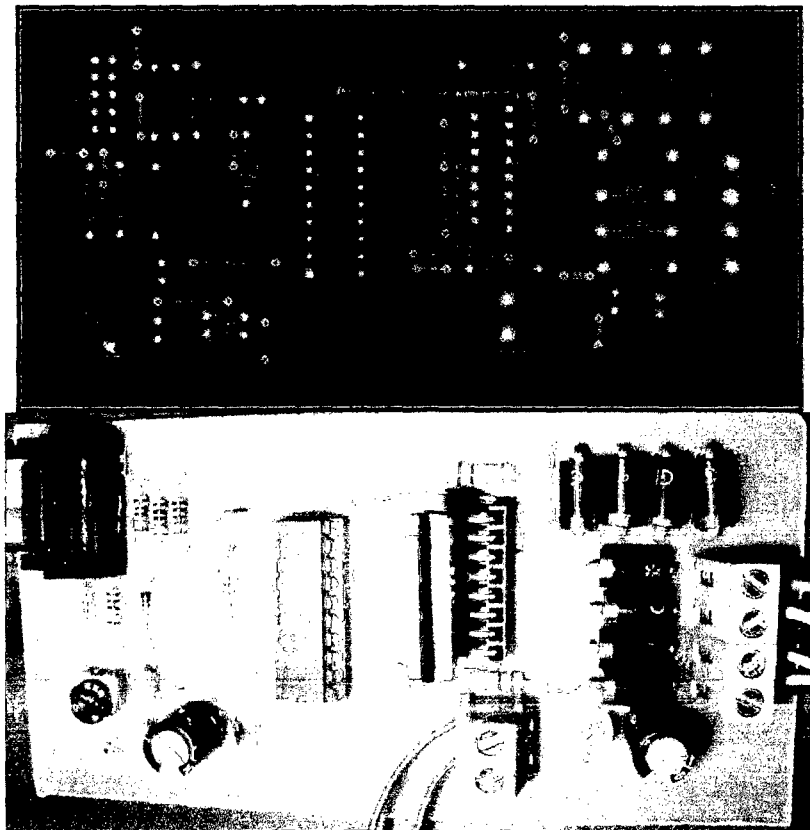


Figura C8 Circuito impreso e implementación del control de motores pap

Existen muchas formas de realizar un circuito impreso, pero si se tiene todos los materiales y las herramientas necesarias resulta muy sencillo realizar estos circuitos impresos, en la mayoría de los casos hemos utilizado el método de la plancha.

**ANEXO D**

**CÓDIGO FUENTE PROGRAMA PRINCIPAL**

En esta parte vamos a mostrar el código fuente de los archivos que componen el proyecto para un óptimo desarrollo del proceso. Tenemos dos clases de archivos unos con extensión .c y otros con extensión .h. Donde los archivos .c son archivos donde se edita el código fuente del programa principal del controlador, ya sea en assembler o en código C. Los archivos .h son las cabeceras donde se declaran algunas variables o constantes que se utilizan en el archivo .c.

### **Código Fuente Programa Principal**

**lcd.h:** cabecera del archivo lcd1.c, este archivo te provee el fabricante para una correcta utilización del display LCD

```
/*
*****
/* ARCHIVO DE CABECERA PARA EL PROGRAMA lcd1.c */
/* *****
/* Definiendo las funciones que se van a emplear en el programa*/
void Init_LCD( void );           // Inicializa el LCD
void lcd_cmd( char cmd );       // Envía un comando al LCD
void lcd_data( char data );     // escribe un dato al LCD
void puts_lcd ( unsigned char *data, unsigned char count ); // escribe un texto al LCD

/***** Prototipos de funciones de comandos LCD *****/
#define cursor_right()  lcd_cmd( 0x14 )
#define cursor_left()   lcd_cmd( 0x10 )
#define display_shift() lcd_cmd( 0x1C )
#define home_clr()      lcd_cmd( 0x01 )
#define home_it()       lcd_cmd( 0x02 )
#define line_2()        lcd_cmd( 0xC0 ) // (0xC0)
```

**lcd1.c:** archivo de configuración de pines y control de datos del LCD, este archivo te provee el fabricante del LCD para una correcta utilización del display LCD

```

/*****
/* ARCHIVO LCD.C*/
*****/

#include <p33FJ64GS606.h>
#include "lcd.h" /*incluimos la cabecera del lcd.c*/
#include "delay.h" /*incluimos la cabecera del delay1.c*/

/* CONFIGURAMOS LOS PINES PARA EL CONTROL DE DATOS DEL LCD
RS -> RD9
E -> RD10
RW -> RD8
DATA -> RD0 - RD7 */

// CONTROL DE DATOS DE LOS PINES DE SEÑAL
#define RW LATDbits.LATD8 // Modo lectura o escritura
#define RS LATDbits.LATD9 // Modo comando o caracter
#define E LATDbits.LATD10 // Habilita envío de datos a LCD

// SEÑAL DE CONTROL DE LOS PINES DE DIRECCION
#define RW_TRIS TRISDbits.TRISD8
#define RS_TRIS TRISDbits.TRISD9
#define E_TRIS TRISDbits.TRISD10

// SEÑAL DE DATOS Y PINES DE DIRECCION
#define DATA LATD // Puerto para enviar datos al LCD
#define DATAPORT PORTD
#define TRISDATA TRISD // Configuración de Entrada/Salida para el puerto

// *****
// ***** SUBROUTINAS PARA MANEJAR EL LCD *****
// *****
// SUBROUTINA PARA INICIALIZAR EL LCD
void Init_LCD( void ) // initialize LCD display
{

    Delay(Delay_15mS_Cnt); // 15ms delay

    /* Establecemos los valores iniciales de los pines de datos
    y control */
    LATD &= 0xFF00;
    RW = 0; // RW en 0 (Modo ESCRITURA)
    RS = 0; // RS en 0 (Modo COMANDO)
    E = 0; // No se valida nada

    /* Establecemos que pines son de entrada y de salida */
    TRISD &= 0xFF00; // Los 8 LSB como SALIDA y los 8 MSB sin cambio
    RW_TRIS = 0; // El pin EN como SALIDA
    RS_TRIS = 0; // El pin RS como SALIDA
    E_TRIS = 0; // El pin EN como SALIDA

    /* PRIMERA SECUENCIA DE INICIALIZACIÓN */
    DATA &= 0xFF00; // Limpia el puerto de datos
    DATA |= 0x0038; // El dato a enviar es 0x38
    E = 1; // Se envía el dato (En este caso es un comando)
}

```

```

Nop(); // Pequeño retardo
Nop(); // (Debe ser de, por lo menos, 0.5us)
Nop();
E = 0; // Se termina de enviar el dato
Delay(Delay_5mS_Cnt); // Retardo de 5ms

/* SEGUNDA SECUENCIA DE INICIALIZACIÓN */
DATA &= 0xFF00; // Limpia el puerto de datos
DATA |= 0x0038; // El dato a enviar es 0x38
E = 1; // Se envía el dato (En este caso es un comando)
Nop();
Nop();
Nop();
E = 0; // Se termina de enviar el dato
Delay_Us( Delay200uS_count ); // 200uS RETARDO

/* TERCERA SECUENCIA DE INICIALIZACIÓN */
DATA &= 0xFF00;
DATA |= 0x0038;
E = 1;
Nop();
Nop();
Nop();
E = 0; // Se termina de enviar el dato
Delay_Us( Delay200uS_count ); // 200uS RETARDO

lcd_cmd( 0x38 ); // Configuración
lcd_cmd( 0x0C ); // Encendemos el LCD, con el cursor apagado
lcd_cmd( 0x06 ); // Modo entrada
}

```

```

void lcd_cmd( char cmd ) // SUBROUTINA PARA LOS COMANDOS DEL LCD
{
// TRISD &= 0xFF00; // Nos aseguramos de que el puerto esté como salida
DATA &= 0xFF00; // prepare RD0 - RD7
DATA |= cmd; // command byte to lcd
RW = 0; // Nos aseguramos que RW esté en 0
RS = 0; // Modo COMANDO
E = 1; // Se envía el dato
Nop();
Nop();
Nop();
E = 0;
Delay(Delay_5mS_Cnt); // retardo de 5ms
}

```

```

void lcd_data( char data ) // subrutina para los datos del LCD
{
RW = 0; // Nos aseguramos de que RW esté en 0
RS = 1; // Modo CHARACTER
DATA &= 0xFF00; // Limpia el puerto de datos
DATA |= data; // Escribimos el carácter en los 8 bits menos significativos
E = 1; // Se envía el dato
Nop();
Nop();
}

```

```

    Nop();
    E = 0;          // Terminamos de enviar el dato
    RS = 0;        // Volvemos al modo COMANDO
    Delay_Us( Delay200uS_count ); // Retardo 200us
    Delay_Us( Delay200uS_count ); // Retardo 200us
}

```

```

void puts_lcd( unsigned char *data, unsigned char count )
{
    while ( count )
    {
        lcd_data( *data++ );
        count --;
    }
}

```

**delay.h:** cabecera del archivo delay1.c

```

#define Fcy 4000000
void Delay( unsigned int delay_count );
void Delay_Us( unsigned int delayUs_count );
#define Delay200uS_count (Fcy * 0.0002) / 1080
#define Delay_1mS_Cnt (Fcy * 0.001) / 3200
#define Delay_2mS_Cnt (Fcy * 0.002) / 3200
#define Delay_5mS_Cnt (Fcy * 0.005) / 3200
#define Delay_15mS_Cnt (Fcy * 0.015) / 3200
#define Delay_1S_Cnt (Fcy * 1) / 3200
#define Delay_1er_Cnt (Fcy * 3.928) / 3200
#define Delay_2do_Cnt (Fcy * 3.968) / 3200
#define Delay_3ro_Cnt (Fcy * 4.009) / 3200
#define Delay_4to_Cnt (Fcy * 4.052) / 3200
#define Delay_5to_Cnt (Fcy * 4.097) / 3200
#define Delay_6to_Cnt (Fcy * 4.142) / 3200
#define Delay_7mo_Cnt (Fcy * 4.190) / 3200
#define Delay_8vo_Cnt (Fcy * 4.239) / 3200
#define Delay_9no_Cnt (Fcy * 4.289) / 3200
#define Delay_10mo_Cnt (Fcy * 4.342) / 3200
#define Delay_11vo_Cnt (Fcy * 4.397) / 3200
#define Delay_12vo_Cnt (Fcy * 4.453) / 3200
#define Delay_13vo_Cnt (Fcy * 4.512) / 3200
#define Delay_14vo_Cnt (Fcy * 4.574) / 3200
#define Delay_15vo_Cnt (Fcy * 4.638) / 3200
#define Delay_16vo_Cnt (Fcy * 4.705) / 3200
#define Delay_17vo_Cnt (Fcy * 4.774) / 3200
#define Delay_18vo_Cnt (Fcy * 4.847) / 3200
#define Delay_19vo_Cnt (Fcy * 4.924) / 3200
#define Delay_20vo_Cnt (Fcy * 5.004) / 3200
#define Delay_21vo_Cnt (Fcy * 5.088) / 3200
#define Delay_22vo_Cnt (Fcy * 5.176) / 3200
#define Delay_23vo_Cnt (Fcy * 5.270) / 3200
#define Delay_24vo_Cnt (Fcy * 5.368) / 3200
#define Delay_25vo_Cnt (Fcy * 5.472) / 3200
#define Delay_26vo_Cnt (Fcy * 5.583) / 3200
#define Delay_27vo_Cnt (Fcy * 5.700) / 3200
#define Delay_28vo_Cnt (Fcy * 5.826) / 3200

```

```

#define Delay_29vo_Cnt (Fcy * 5.960) / 3200
#define Delay_30vo_Cnt (Fcy * 6.103) / 3200
#define Delay_31vo_Cnt (Fcy * 6.258) / 3200
#define Delay_32vo_Cnt (Fcy * 6.425) / 3200
#define Delay_33vo_Cnt (Fcy * 6.606) / 3200
#define Delay_34vo_Cnt (Fcy * 6.803) / 3200
#define Delay_35vo_Cnt (Fcy * 7.020) / 3200
#define Delay_36vo_Cnt (Fcy * 7.258) / 3200
#define Delay_37vo_Cnt (Fcy * 7.522) / 3200
#define Delay_38vo_Cnt (Fcy * 7.817) / 3200
#define Delay_39vo_Cnt (Fcy * 8.150) / 3200
#define Delay_40vo_Cnt (Fcy * 8.530) / 3200
#define Delay_41vo_Cnt (Fcy * 8.968) / 3200
#define Delay_42vo_Cnt (Fcy * 9.482) / 3200
#define Delay_43vo_Cnt (Fcy * 10.096) / 3200
#define Delay_44vo_Cnt (Fcy * 10.846) / 3200
#define Delay_45vo_Cnt (Fcy * 11.795) / 3200
#define Delay_46vo_Cnt (Fcy * 13.046) / 3200
#define Delay_47vo_Cnt (Fcy * 14.808) / 3200
#define Delay_48vo_Cnt (Fcy * 17.565) / 3200
#define Delay_49vo_Cnt (Fcy * 22.891) / 3200
#define Delay_50vo_Cnt (Fcy * 55.265) / 3200

```

**delay1.c:** archivo de conteo para los retardos proporcionado por la empresa

Microchip para distintas aplicaciones.

```
#include "delay.h"
```

```
unsigned int temp_count;
```

```
void Delay( unsigned int delay_count )
```

```
{
    temp_count = delay_count + 1;
    asm volatile("outer: dec_temp_count");
    asm volatile("cp0_temp_count");
    asm volatile("bra z, done");
    asm volatile("do #3200, inner");
    asm volatile("nop");
    asm volatile("inner: nop");
    asm volatile("bra outer");
    asm volatile("done:");
}
```

```
void Delay_Us( unsigned int delayUs_count )
```

```
{
    temp_count = delayUs_count + 1;
    asm volatile("outer1: dec_temp_count");
    asm volatile("cp0_temp_count");
    asm volatile("bra z, done1");
    asm volatile("do #1500, inner1");
    asm volatile("nop");
    asm volatile("inner1: nop");
    asm volatile("bra outer1");
}
```



```

asm volatile("done1:");
}

```

**principal.c:** archivo principal de control del sistema, en este archivo realizamos todo el control del sistema, desde configuración del oscilador, inicialización de entradas y salidas y control digital de todo el proceso de nuestro diseño.

```

/*****
/*  PROGRAMA DE CONTROL DEL SISTEMA  */
/*****
#include "p33FJ64GS606.h"          /*Modelo del dsPIC a usar*/
#include "lcd.h"                  /*incluimos la cabecera del lcd1.c*/
#include "delay.h"               /*incluimos la cabecera del delay1.c*/
#include <math.h>
_FOSCSEL(IESO_OFF & FNOSC_PRIPLL); // Oscilador primario + PLL
_FOSC(FCKSM_CSDCMD & OSCIOFNC_OFF & POSCMD_XT); // Conmutación de Clock y Fail-
Safe Clock des-habilitados, OSC2 Clout, Cristal XT.
_FWDT(FWDTEN_OFF); // Watchdog des-habilitado.
_FPOR(FPWR2_PWR2); // Power-on Reset 2ms.
/* INICIALIZAMOS LAS VARIABLES A USAR */
int x;                          /*inicializamos la variable x*/
int i;                          /*inicializamos la variable i*/
int j;                          /*inicializamos la variable j*/
int k;                          /*inicializamos la variable k*/
int y;                          /*inicializamos la variable y*/
/* CADENAS DE TEXTOS QUE SE UTILIZARAN EN EL LCD */
const char mytext[] = "PULSE EL JUGO Q";
const char mytext1[] = "DESEA SERVIRSE";
const unsigned char __attribute__((space(psv))) Cadena0[] = " JUGO SURTIDO !";
const unsigned char __attribute__((space(psv))) Cadena1[] = " JUGO DE PAPAYA!";
const unsigned char __attribute__((space(psv))) Cadena2[] = " JUGO DE PINNA ";
const unsigned char __attribute__((space(psv))) Cadena3[] = "JUGO DE MANZANA!!";
const unsigned char __attribute__((space(psv))) Cadena4[] = "JUGO DE PLATANO!!";
const unsigned char __attribute__((space(psv))) Cadena6[] = " MODO MANUAL !!";
const unsigned char __attribute__((space(psv))) Cadena7[] = "FALLA SNSR TEMP!!";
const unsigned char __attribute__((space(psv))) Cadena8[] = " BUEN PROVECHO!!";
const unsigned char __attribute__((space(psv))) Cadena9[] = "INSERTE 3 SOLES!";
const unsigned char __attribute__((space(psv))) Cadena10[] = "FALTA 2,5 SOLES";
const unsigned char __attribute__((space(psv))) Cadena11[] = "FALTA 2 SOLES";
const unsigned char __attribute__((space(psv))) Cadena12[] = "FALTA 1,5 SOLES";
const unsigned char __attribute__((space(psv))) Cadena13[] = "FALTA 1 SOL";
const unsigned char __attribute__((space(psv))) Cadena14[] = "FALTA 0,5 SOLES";
const unsigned char __attribute__((space(psv))) Cadena15[] = "FALTA VASOS";
/*****
/*  FUNCION PRINCIPAL  */
/*****
int main (void)
{
    //CONFIGURARAMOS EL OSCILADOR Y EL PLL PARA 40MIPS
    //COMO SE MUESTRA ABAJO:
    //      Crystal Frequency * (M)

```

```

// Fcy = -----
//      (N1) * (N2) * 2
PLLFBFD = 0x0026;
CLKDIV = 0x0000;

    _INT4IP=7;          /* Le damos prioridad 4 a la interrupción Externa 4*/
    _INT4IF=0;         /* Reseteo el Flag de la int Externa 4 */
    _INT4IE=1;        /* Habilito la interrupción Externa 4 */
    _IPL=0;           /* Prioridad del procesador=0 */
    _INT4EP=0;        /* Polaridad positiva*/
    IOInit();         /*Inicializo los puertos de Entrada y Salida*/
    Init_LCD();       /* Inicializo el LCD */
    PSVInit();        /* Inicializo los registros de las cadenas */
    k=0;              /* Inicializo la constante k */
    y=0;              /* Inicializo la constante y */
    while(1)
    {
        home_clr();   /* limpiamos la pantalla del LCD */
        manual();     /* si bit RB5=0 entramos a modo manual*/
        falta_vaso(); /* si falta vaso no opera el sistema */
        reconocer_moneda(); /* reconoce dinero para funcionamiento sistema */
        /* indicamos en el LCD el jugo que desea */
        puts_lcd( (char*) &mytext[0], sizeof(mytext) -1 );
        line_2();     /* para escribir en la segunda línea del LCD */
        line_2();
        line_2();
        puts_lcd( (char*) &mytext1[0], sizeof(mytext1) -1 );
        automatico(); /* entramos al proceso automático */
    }
}
/*****/
/* retardo principal */
/*****/
DelayMain()
{
    int i;
    for(i = 0; i < 0xFFFF; i++);}
/*****/
/* Inicializamos los puertos de Entrada y Salida */
/*****/
IOInit()
{
    ADPCFG=0xffff;
    // Poniendo AD1PCFGH bit a "0" configura su respectivo pin del puerto (AN) como una entrada
    analogica
    // Poniendo AD1PCFGH bit a "1" configura el pin del puerto correspondiente (AN) como una entrada
    digital

    /* Configuro como entradas los pulsadores, sensores, monederos y Switchs */
    TRISBbits.TRISB0 = 1; /* jugo1 */
    TRISBbits.TRISB1 = 1; /* jugo2 */
    TRISBbits.TRISB2 = 1; /* jugo3 */
    TRISBbits.TRISB3 = 1; /* jugo4 */
    TRISBbits.TRISB4 = 1; /* jugo5 */
    TRISBbits.TRISB5 = 1; /* Manual/Automático */
    TRISGbits.TRISG9 = 1; /* electroválvula manual */
    TRISGbits.TRISG8 = 1; /* Falta vaso sistema */
    TRISDbits.TRISD11 = 1; /* temperatura licuadora */
    TRISGbits.TRISG6 = 1; /* vaso posicionado listo para servirse */
}

```

```

TRISEbits.TRISE5 = 1; /* monedero 50 céntimos */
TRISEbits.TRISE6 = 1; /* monedero 1 sol */
TRISEbits.TRISE7 = 1; /* monedero 2 soles */

/* Seteamos las salidas a un nivel bajo */
LATFbits.LATF0 = 0;
LATFbits.LATF1 = 0;
LATEbits.LATE0 = 0;
LATEbits.LATE1 = 0;
LATCbits.LATC13 = 0;
LATCbits.LATC14 = 0;
LATCbits.LATC12 = 0;
LATCbits.LATC15 = 0;
LATGbits.LATG2 = 0;
LATGbits.LATG3 = 0;
LATEbits.LATE2 = 0;
LATEbits.LATE3 = 0;
LATEbits.LATE4 = 0;
LATFbits.LATF3 = 0;
LATFbits.LATF2 = 0;
LATFbits.LATF6 = 0;
LATFbits.LATF4 = 0;
LATFbits.LATF5 = 0;
LATGbits.LATG7 = 0;
LATBbits.LATB8 = 0;
LATBbits.LATB9 = 0;
LATBbits.LATB10 = 0;
LATBbits.LATB11 = 0;
LATBbits.LATB12 = 0;
LATBbits.LATB13 = 0;
LATBbits.LATB14 = 0;
LATBbits.LATB15 = 0;
/* Configuro las salidas */
TRISFbits.TRISF0 = 0; /* clock M1 */
TRISFbits.TRISF1 = 0; /* enable M1 */
TRISEbits.TRISE0 = 0; /* clock M2 */
TRISEbits.TRISE1 = 0; /* enable M2 */
TRISCbits.TRISC13 = 0; /* clock M3 */
TRISCbits.TRISC14 = 0; /* enable M3 */
TRISCbits.TRISC12 = 0; /* clock M4 */
TRISCbits.TRISC15 = 0; /* enable M4 */
TRISGbits.TRISG2 = 0; /* clock M5 */
TRISGbits.TRISG3 = 0; /* enable M5 */
TRISEbits.TRISE2 = 0; /* clock M6 */
TRISEbits.TRISE3 = 0; /* enable M6 */
TRISEbits.TRISE4 = 0; /* Dirección M6 */
TRISFbits.TRISF3 = 0; /* clock M7 */
TRISFbits.TRISF2 = 0; /* enable M7 */
TRISFbits.TRISF6 = 0; /* Dirección M7 */
TRISFbits.TRISF4 = 0; /* clock M8 */
TRISFbits.TRISF5 = 0; /* enable M8 */
TRISGbits.TRISG7 = 0; /* Dirección M8 */
TRISBbits.TRISB8 = 0; /* encendido licuadora */
TRISBbits.TRISB9 = 0; /* encendido electroválvula */
TRISBbits.TRISB10 = 0; /* alarma */
TRISBbits.TRISB11 = 0; /* Giro M_F1 */
TRISBbits.TRISB12 = 0; /* Giro M_F2 */

```

```

TRISBbits.TRISB13 = 0; /* Giro M_F3 */
TRISBbits.TRISB14 = 0; /* Giro M_F4 */
TRISBbits.TRISB15 = 0; /* Giro M_F5 */
}
/*****/
/* AUTOMATICO */
/*****/
void automatico()
{
    while (PORTBbits.RB5==1) /* mientras esta en automático */
    {
        /* espera un pulso para el jugo que desea */
        if (PORTBbits.RB0==0) /* si pulsa RB0 prepara jugo surtido */
        {
            home_clr(); /* limpio pantalla LCD */
            line_2(); /* escribo jugo surtido en el LCD */
            puts_lcd( (char*) &Cadena0[0], sizeof(Cadena0) -1 );
            papaya(); /* doy un giro completo del motor1 */
            pinna(); /* doy un giro completo del motor2 */
            manzana(); /* doy un giro completo del motor3 */
            platano(); /* doy un giro completo del motor4 */
            melon(); /* doy un giro completo del motor5 */
            licuado_servido(); /* realiza el licuado y servido del jugo*/
            break; /* rompo bucle infinito y salgo del while*/
        }
        else if (PORTBbits.RB1==0) /* si pulsa RB1 prepara jugo papaya */
        {
            home_clr(); /* limpio pantalla LCD */
            line_2(); /* escribo jugo papaya en el LCD */
            puts_lcd( (char*) &Cadena1[0], sizeof(Cadena1) -1 );
            for (j=0;j<=4;j++) /* bucle para dar 5 giros del motor1 */
            {
                papaya();
                Delay(Delay_1S_Cnt);
            }
            licuado_servido(); /* realiza el licuado y servido del jugo*/
            break; /* rompo bucle infinito y salgo del while*/
        }
        else if (PORTBbits.RB2==0) /* si pulsa RB2 prepara jugo piña */
        {
            home_clr(); /* limpio pantalla LCD */
            line_2(); /* escribo jugo de piña en el LCD */
            puts_lcd( (char*) &Cadena2[0], sizeof(Cadena2) -1 );
            for (j=0;j<=4;j++) /* bucle para dar 5 giros del motor2 */
            {
                pinna();
                Delay(Delay_1S_Cnt);
            }
            licuado_servido(); /* realiza el licuado y servido del jugo*/
            break; /* rompe bucle infinito y sale del while*/
        }
        else if (PORTBbits.RB3==0) /*si pulsa RB3 prepara jugo manzana*/
        {
            home_clr(); /* limpio pantalla LCD */
            line_2(); /* escribo jugo de manzana en el LCD */
            puts_lcd( (char*) &Cadena3[0], sizeof(Cadena3) -1 );
            for (j=0;j<=4;j++) /*bucle para dar 5 giros del motor3*/
            {
                manzana();
                Delay(Delay_1S_Cnt);
            }
            licuado_servido(); /* realiza el licuado y servido del jugo*/
        }
    }
}

```

```

        break;                /*rompe bucle infinito y sale del while*/
    }

else if (PORTBbits.RB4==0)    /*si pulsa RB4 prepara jugo plátano*/
{
    home_clr();              /* limpio pantalla LCD */
    line_2();                /* escribo jugo de plátano en el LCD */
    puts_lcd( (char*) &Cadena4[0], sizeof(Cadena4) -1 );
    for (j=0;j<=4;j++)      /*bucle para dar 5 giros del motor4*/
    {
        platano();
        Delay(Delay_1S_Cnt);
    }
    licuado_servido();      /* realiza el licuado y servido del jugo*/
    break;                  /*rompe bucle infinito y sale del while*/
}
DelayMain();
}

}
/*****
/* modo manual */
*****/
void manual()
{
    line_2();                /* escribo modo manual en LCD */
    puts_lcd( (char*) &Cadena6[0], sizeof(Cadena6) -1 );
    while (PORTBbits.RB5==0) /* mientras RB5=0 modo manual*/
    {
        if (PORTBbits.RB0==0) /* si pulso RB0 giro motor 1 */
        {
            papaya();        /* doy un giro completo del motor1 */
        }
        else if (PORTBbits.RB1==0) /* si pulso RB1 giro motor 2 */
        {
            pinna(); /* doy un giro completo del motor2 */
        }
        else if (PORTBbits.RB2==0) /*si pulso RB2 giro motor 3*/
        {
            manzana(); /* doy un giro completo del motor3 */
        }
        else if (PORTBbits.RB3==0) /*si pulso RB3 giro motor 4*/
        {
            platano(); /* doy un giro completo del motor4 */
        }
        else if (PORTBbits.RB4==0) /*si pulso RB4 giro motor 5*/
        {
            melon(); /* doy un giro completo del motor5 */
        }
        else if (PORTGbits.RG9==0) /*si pulso RG9 enciendo EV*/
        {
            while (PORTGbits.RG9==0)
            {
                PORTBbits.RB9=1;
            }
        }
    }
    DelayMain();
}
}

```

```
}
```

```
/******  
/* RECONOCER MONEDA */  
/******  
void reconocer_moneda()  
{  
    line_2();/*escribo INSERTE 3 SOLES*/  
    puts_lcd( (char*) &Cadena9[0], sizeof(Cadena9) -1 );  
    x=0; /*inicializo x=0*/  
    while (1) /*bucle infinito*/  
    {  
        /*pregunto si ingresaron 50 céntimos*/  
        if(PORTEbits.RE5==0)  
        {  
            Delay(Delay_15mS_Cnt);  
            Delay(Delay_15mS_Cnt);  
            Delay(Delay_15mS_Cnt);  
            x=x+1; /*50 céntimos es x=1*/  
            if (x==6) /*pregunto si x es 3 soles*/  
            {  
                break; /*si x=6 rompo bucle infinito*/  
            }  
            else /*si no*/  
            {  
                /*esta función pregunta cuánto le falta para 3*/  
                //Delay(Delay_1mS_Cnt);  
                cuanto_falta_para_3soles();  
            }  
        }  
        /*Pregunto si ingresaron 1 sol*/  
        else if(PORTEbits.RE6==0)  
        {  
            Delay(Delay_15mS_Cnt);  
            Delay(Delay_15mS_Cnt);  
            Delay(Delay_15mS_Cnt);  
            x=x+2; /*1 sol es x=2 y se le suma a x*/  
            if (x==6) /*pregunto si x es 3 soles*/  
            {  
                break; /*si x=6 rompo bucle infinito*/  
            }  
            else /*si no*/  
            {  
                /*esta función pregunta cuánto le falta para 3*/  
                //Delay(Delay_1mS_Cnt);  
                cuanto_falta_para_3soles();  
            }  
        }  
        /*pregunto si ingresaron 2 soles*/  
        else if(PORTEbits.RE7==0)  
        {  
            Delay(Delay_15mS_Cnt);  
            Delay(Delay_15mS_Cnt);  
            Delay(Delay_15mS_Cnt);  
            x=x+4; /*2soles es x=4 y se le suma a x*/  
            if (x==6) /*pregunto si x es 3 soles*/
```

```

        {
            break; /*si x=6 rompo bucle infinito*/
        }

        else /*si no*/
        {
            /*esta función pregunta cuánto le falta para 3*/
            //Delay(Delay_1mS_Cnt);
            cuanto_falta_para_3soles();
        }
    }
    //Delay(Delay200uS_count);
    DelayMain();
}
home_clr(); /*cuando llega a 3 soles limpia LCD*/
}
/*****
/* CUANTO FALTA PARA 3 SOLES */
*****/
void cuanto_falta_para_3soles()
{while (1)
{    if(x==1) /*si x=50 céntimos indico que falta 2.5soles*/
    {
        home_clr();
        home_clr();
        home_clr();
        //line_2();
        puts_lcd( (char*) &Cadena10[0], sizeof(Cadena10) -1 );
        break;
    }
    else if (x==2) /*si x=1sol indico que falta 2soles*/
    {
        home_clr();
        home_clr();
        home_clr();
        //line_2();
        puts_lcd( (char*) &Cadena11[0], sizeof(Cadena11) -1 );
        break;
    }
    else if (x==3) /*si x=1.5soles indico que falta 1.5soles*/
    {
        home_clr();
        home_clr();
        home_clr();
        //line_2();
        puts_lcd( (char*) &Cadena12[0], sizeof(Cadena12) -1 );
        break;
    }
    else if (x==4) /*si x=2soles indico que falta 1sol*/
    {
        home_clr();
        home_clr();
        home_clr();
        //line_2();
        puts_lcd( (char*) &Cadena13[0], sizeof(Cadena13) -1 );
        break;
    }
}
}

```

```

else if (x==5) /*si x=2.5soles indico que falta 0.5soles*/
{
    home_clr();
    home_clr();
    home_clr();
    //line_2();
    puts_lcd( (char*) &Cadena14[0], sizeof(Cadena14) -1 );
    break;
}
}
}
/*****/
/* FALTA VASO */
/*****/
void falta_vaso()
{
    while (1) /*bucle infinito*/
    {
        /*Pregunto si no hay vaso para el servido del jugo*/
        if((PORTGbits.RG8==0))
        {
            line_2();/*indico que no hay vaso en el LCD*/
            puts_lcd( (char*) &Cadena15[0], sizeof(Cadena15) -1 );
        }
        else /*si no*/
        {
            home_clr(); /*limpio LCD*/
            break;} /*rompo bucle infinito y salgo del while*/
    }
}
/*****/
/* giro de motores de las frutas */
/*****/
void papaya()
{
    PORTBbits.RB11=1; /* activo giro motor asegura caída fruta 1*/
    Delay(Delay_1S_Cnt);
    Delay(Delay_1S_Cnt);
    Delay(Delay_1S_Cnt);
    PORTBbits.RB11=0; /* desactivo giro motor asegura caída fruta 1*/
    /* con este bucle doy 200 pulsos al motor1 para 1 giro completo */
    PORTFbits.RF1=1; /* habilito motor1*/
    for (i=0; i<=199;i++)
    {
        PORTFbits.RF0=1; /* activo clock motor1*/
        Delay(Delay_5mS_Cnt); /* retardo para dar la vuelta en 4 segundos*/
        Delay(Delay_5mS_Cnt);
        PORTFbits.RF0=0; /* desactivo clock motor1*/
        Delay(Delay_5mS_Cnt); /* retardo para dar la vuelta en 4 segundos*/
        Delay(Delay_5mS_Cnt);
    }
    PORTFbits.RF1=0; /* deshabilito motor1*/
    DelayMain();
}
}

```



```

void pinna()
{
    PORTBbits.RB12=1;          /* activo giro motor asegura caída fruta 2*/
    Delay(Delay_1S_Cnt);
    Delay(Delay_1S_Cnt);
    Delay(Delay_1S_Cnt);
    PORTBbits.RB12=0;         /* desactivo giro motor asegura caída fruta 2*/
    /* con este bucle doy 200 pulsos al motor2 para 1 giro completo */
    PORTEbits.RE1=1;         /* habilito motor2*/
    for (i=0; i<=199;i++)
    {
        PORTEbits.RE0=1;     /* activo clock motor2*/
        Delay(Delay_5mS_Cnt); /* retardo para dar la vuelta en 4 segundos*/
        Delay(Delay_5mS_Cnt);
        PORTEbits.RE0=0;     /* desactivo clock motor2*/
        Delay(Delay_5mS_Cnt);
        Delay(Delay_5mS_Cnt);
    }
    PORTEbits.RE1=0;         /* deshabilito motor2*/
    DelayMain();
}

void manzana()
{
    PORTBbits.RB13=1;         /* activo giro motor asegura caída fruta 3*/
    Delay(Delay_1S_Cnt);
    Delay(Delay_1S_Cnt);
    Delay(Delay_1S_Cnt);
    PORTBbits.RB13=0;         /* desactivo giro motor asegura caída fruta 3*/
    /* con este bucle doy 200 pulsos al motor3 para 1 giro completo */
    PORTCbits.RC14=1;         /* habilito motor3*/
    for (i=0; i<=199;i++)
    {
        PORTCbits.RC13=1;     /* activo clock motor3*/
        Delay(Delay_5mS_Cnt); /* retardo para dar la vuelta en 4 segundos*/
        Delay(Delay_5mS_Cnt);
        PORTCbits.RC13=0;     /* desactivo clock motor3*/
        Delay(Delay_5mS_Cnt);
        Delay(Delay_5mS_Cnt);
    }
    PORTCbits.RC14=0;         /* deshabilito motor3*/
    DelayMain();
}

void platano()
{
    PORTBbits.RB14=1;         /* activo giro motor asegura caída fruta 4*/
    Delay(Delay_1S_Cnt);
    Delay(Delay_1S_Cnt);
    Delay(Delay_1S_Cnt);
    PORTBbits.RB14=0;         /* desactivo giro motor asegura caída fruta 4*/
    /* con este bucle doy 200 pulsos al motor4 para 1 giro completo */
    PORTCbits.RC15=1;         /* habilito motor4*/
    for (i=0; i<=199;i++)
    {
        PORTCbits.RC12=1;     /* activo clock motor4*/
        Delay(Delay_5mS_Cnt); /* retardo para dar la vuelta en 4 segundos*/
        Delay(Delay_5mS_Cnt);
        PORTCbits.RC12=0;     /* desactivo clock motor4*/
        Delay(Delay_5mS_Cnt);
        Delay(Delay_5mS_Cnt);
    }
    PORTCbits.RC15=0;         /* deshabilito motor4*/
}

```

```

        DelayMain();
    }
void melon()
{
    PORTBbits.RB15=1;          /* activo giro motor asegura caída fruta 5*/
    Delay(Delay_1S_Cnt);
    Delay(Delay_1S_Cnt);
    Delay(Delay_1S_Cnt);
    PORTBbits.RB15=0;          /* desactivo giro motor asegura caída fruta 5*/
    /* con este bucle doy 200 pulsos al motor5 para 1 giro completo */
    PORTGbits.RG3=1;          /* habilito motor5*/
    for (i=0; i<=199;i++)
    {
        PORTGbits.RG2=1;          /* activo clock motor5*/
        Delay(Delay_5mS_Cnt); /* retardo para dar la vuelta en 4 segundos*/
        Delay(Delay_5mS_Cnt);
        PORTGbits.RG2=0;          /* desactivo clock motor5*/
        Delay(Delay_5mS_Cnt);
        Delay(Delay_5mS_Cnt);
    }
    PORTGbits.RG3=0;          /* deshabilito motor5*/
    DelayMain();
}
/*****
/* giro de motores con dirección */
*****/
void horario_licuadora()
{
    PORTEbits.RE4=1;          /* activo dirección horario motor6*/
    Delay(Delay_15mS_Cnt);
    PORTEbits.RE3=1;          /* habilito motor6*/
    /* con este bucle doy 50 pulsos al motor6 para 1/4 giro*/
    for (i=0; i<=49;i++)
    {
        PORTEbits.RE2=1;          /* activo clock motor6*/
        Delay(Delay_15mS_Cnt);/* retardo para girar en 1.5 segundos*/
        PORTEbits.RE2=0;          /* desactivo clock motor6*/
        Delay(Delay_15mS_Cnt);
    }
    PORTEbits.RE3=0;          /* deshabilito motor6*/
    DelayMain();
}

void antihorario_licuadora()
{
    PORTEbits.RE4=0;          /* activo dirección antihorario motor6*/
    Delay(Delay_15mS_Cnt);
    PORTEbits.RE3=1;          /* habilito motor6*/
    /* con este bucle doy 50 pulsos al motor6 para 1/4 giro*/
    for (i=0; i<=49;i++)
    {
        PORTEbits.RE2=1;          /* activo clock motor6*/
        Delay(Delay_15mS_Cnt);/* retardo para girar en 1.5 segundos*/
        PORTEbits.RE2=0;          /* desactivo clock motor6*/
        Delay(Delay_15mS_Cnt);
    }
    PORTEbits.RE3=0;          /* deshabilito motor6*/
}

```

```

        DelayMain();
    }

void horario_brazo()
{
    PORTFbits.RF6=1;      /* activo dirección horario motor7*/
    Delay(Delay_15mS_Cnt);
    PORTFbits.RF2=1;      /* habilito motor7*/
    /* con este bucle doy 50 pulsos al motor6 para 1/4 giro*/
    for (i=0; i<=49;i++)
    {
        PORTFbits.RF3=1;      /* activo clock motor6*/
        Delay(Delay_15mS_Cnt);/* retardo para girar en 1.5 segundos*/
        PORTFbits.RF3=0;      /* desactivo clock motor7*/
        Delay(Delay_15mS_Cnt);
    }
    PORTFbits.RF2=0;      /* deshabilito motor7*/
}

void antihorario_brazo()
{
    PORTFbits.RF6=0;      /* activo dirección antihorario motor7*/
    Delay(Delay_15mS_Cnt);
    PORTFbits.RF2=1;      /* habilito motor7*/
    /* con este bucle doy 50 pulsos al motor7 para 1/4 giro*/
    for (i=0; i<=49;i++)
    {
        PORTFbits.RF3=1;      /* activo clock motor7*/
        Delay(Delay_15mS_Cnt);/* retardo para girar en 1.5 segundos*/
        PORTFbits.RF3=0;      /* desactivo clock motor7*/
        Delay(Delay_15mS_Cnt);
    }
    PORTFbits.RF2=0;      /* deshabilito motor7*/
    DelayMain();
}

void horario_mano()
{
    PORTGbits.RG7=1;      /* activo dirección horario motor8*/
    Delay(Delay_15mS_Cnt);
    PORTFbits.RF5=1;      /* habilito motor8*/
    /* con este bucle doy 50 pulsos al motor6 para 1/4 giro*/
    for (i=0; i<=49;i++)
    {
        PORTFbits.RF4=1;      /* activo clock motor8*/
        Delay(Delay_15mS_Cnt);/* retardo para girar en 1.5 segundos*/
        PORTFbits.RF4=0;      /* desactivo clock motor8*/
        Delay(Delay_15mS_Cnt);
    }
    PORTFbits.RF5=0;      /* deshabilito motor8*/
    DelayMain();
}

void antihorario_mano()
{
    PORTGbits.RG7=0;      /* activo dirección antihorario motor8*/
    Delay(Delay_15mS_Cnt);
    PORTFbits.RF5=1;      /* habilito motor8*/
    /* con este bucle doy 50 pulsos al motor6 para 1/4 giro*/
    for (i=0; i<=49;i++)
    {
        PORTFbits.RF4=1;      /* activo clock motor8*/
        Delay(Delay_15mS_Cnt);/* retardo para girar en 1.5 segundos*/
        PORTFbits.RF4=0;      /* desactivo clock motor8*/
        Delay(Delay_15mS_Cnt);
    }
    PORTFbits.RF5=0;      /* deshabilito motor8*/
}

```

```

        DelayMain();
    }

    /*******
    /* licuado y servido del jugo */
    /*******
void licuado_servido()
{
    PORTBbits.RB9=1;           //enciendo electroválvula
    tiempo_vaciado();         //por un tiempo X
    PORTBbits.RB9=0;         //apago electroválvula
    Delay(Delay_15mS_Cnt);
    PORTBbits.RB8=1;         //enciendo licuadora

    while(1)
    {
        horario_mano();           //si es no giro motor mano para sujetar vaso
        Delay(Delay_15mS_Cnt);
        horario_brazo();         //si es no giro motor brazo para bajar vaso
        Delay(Delay_15mS_Cnt);
        antihorario_mano();     //si es no giro motor mano para soltar vaso
        Delay(Delay_15mS_Cnt);
        antihorario_brazo();    //si es no giro motor brazo para regresar punto inicio
        Delay(Delay_15mS_Cnt);
        if (PORTGbits.RG6==0) //pregunto si el vaso esta listo para servido
        {
            //si es si realizo todo esto
            for (k=0;k<=19;k++) /* bucle para que siga licuando x 20segundos */
            {Delay(Delay_1S_Cnt);}
            PORTBbits.RB8=0;     //apago licuadora

            Delay(Delay_1S_Cnt);
            horario_licuadora(); //giro horario motor6 para verter jugo
            Delay(Delay_1S_Cnt);
            antihorario_licuadora();//giro antihorario motor6 para regresar punto inicio
            Delay(Delay_1S_Cnt);
            line_2();/*escribo BUEN PROVECHO en LCD*/
            puts_lcd( (char*) &Cadena8[0], sizeof(Cadena8) - 1 );
            Delay(Delay_1S_Cnt);
            break;             /*rompo bucle infinito y salgo del while*/
        }
        else
        {
            //y si no esta el vaso listo vuelve a realizar la misma operación
            for (k=0;k<=19;k++) /* bucle para que siga licuando x 20segundos */
            {Delay(Delay_1S_Cnt);}
            PORTBbits.RB8=0;     //apago licuadora y vuelvo a realizar de nuevo
        }
        DelayMain();
    }
}

```

```
void tiempo_vaciado()
{
    y=y+1;
    switch(y)
    {
        case 1: Delay(Delay_1er_Cnt);
        break;
        case 2: Delay(Delay_2do_Cnt);
        break;
        case 3: Delay(Delay_3ro_Cnt);
        break;
        case 4: Delay(Delay_4to_Cnt);
        break;
        case 5: Delay(Delay_5to_Cnt);
        break;
        case 6: Delay(Delay_6to_Cnt);
        break;
        case 7: Delay(Delay_7mo_Cnt);
        break;
        case 8: Delay(Delay_8vo_Cnt);
        break;
        case 9: Delay(Delay_9no_Cnt);
        break;
        case 10: Delay(Delay_10mo_Cnt);
        break;
        case 11: Delay(Delay_11vo_Cnt);
        break;
        case 12: Delay(Delay_12vo_Cnt);
        break;
        case 13: Delay(Delay_13vo_Cnt);
        break;
        case 14: Delay(Delay_14vo_Cnt);
        break;
        case 15: Delay(Delay_15vo_Cnt);
        break;
        case 16: Delay(Delay_16vo_Cnt);
        break;
        case 17: Delay(Delay_17vo_Cnt);
        break;
        case 18: Delay(Delay_18vo_Cnt);
        break;
        case 19: Delay(Delay_19vo_Cnt);
        break;
        case 20: Delay(Delay_20vo_Cnt);
        break;
        case 21: Delay(Delay_21vo_Cnt);
        break;
        case 22: Delay(Delay_22vo_Cnt);
        break;
        case 23: Delay(Delay_23vo_Cnt);
        break;
        case 24: Delay(Delay_24vo_Cnt);
        break;
        case 25: Delay(Delay_25vo_Cnt);
        break;
    }
}
```

```
case 26: Delay(Delay_26vo_Cnt);
break;
case 27: Delay(Delay_27vo_Cnt);
break;
case 28: Delay(Delay_28vo_Cnt);
break;
case 29: Delay(Delay_29vo_Cnt);
break;
case 30: Delay(Delay_30vo_Cnt);
break;
case 31: Delay(Delay_31vo_Cnt);
break;
case 32: Delay(Delay_32vo_Cnt);
break;
case 33: Delay(Delay_33vo_Cnt);
break;
case 34: Delay(Delay_34vo_Cnt);
break;
case 35: Delay(Delay_35vo_Cnt);
break;
case 36: Delay(Delay_36vo_Cnt);
break;
case 37: Delay(Delay_37vo_Cnt);
break;
case 38: Delay(Delay_38vo_Cnt);
break;
case 39: Delay(Delay_39vo_Cnt);
break;
case 40: Delay(Delay_40vo_Cnt);
break;
case 41: Delay(Delay_41vo_Cnt);
break;
case 42: Delay(Delay_42vo_Cnt);
break;
case 43: Delay(Delay_43vo_Cnt);
break;
case 44: Delay(Delay_44vo_Cnt);
break;
case 45: Delay(Delay_45vo_Cnt);
break;
case 46: Delay(Delay_46vo_Cnt);
break;
case 47: Delay(Delay_47vo_Cnt);
break;
case 48: Delay(Delay_48vo_Cnt);
break;
case 49: Delay(Delay_49vo_Cnt);
break;
case 50: Delay(Delay_50vo_Cnt);
break;
```

```
}
```

```
}
```

```

void _ISR_INT4Interrupt(void)
{
    //desactivamos todas las salidas
    PORTFbits.RF0 = 0;
    PORTFbits.RF1 = 0;
    PORTEbits.RE0 = 0;
    PORTEbits.RE1 = 0;
    PORTCbits.RC13 = 0;
    PORTCbits.RC14 = 0;
    PORTCbits.RC12 = 0;
    PORTCbits.RC15 = 0;
    PORTGbits.RG2 = 0;
    PORTGbits.RG3 = 0;
    PORTEbits.RE2 = 0;
    PORTEbits.RE3 = 0;
    PORTEbits.RE4 = 0;
    PORTFbits.RF3 = 0;
    PORTFbits.RF2 = 0;
    PORTFbits.RF6 = 0;
    PORTFbits.RF4 = 0;
    PORTFbits.RF5 = 0;
    PORTGbits.RG7 = 0;
    PORTBbits.RB8 = 0;
    PORTBbits.RB9 = 0;
    PORTBbits.RB10 = 0;
    PORTBbits.RB11 = 0;
    PORTBbits.RB12 = 0;
    PORTBbits.RB13 = 0;
    PORTBbits.RB15 = 0;
    home_clr();
    line_2();
    puts_lcd( (char*) &Cadena7[0], sizeof(Cadena7) - 1 );
    PORTBbits.RB10=1;          //enciendo alarma
    for (k=0;k<=4;k++)
        {Delay(Delay_1S_Cnt);}
    PORTBbits.RB10=0;          //apago alarma
    _INT0IF=0;          //Ponemos a 0 el Flag de la Int Externa 2
    while (1){}
}

```

```

/*****
/* Configuración del PSV      */
*****/

```

```

PSVInit()
{
    /* Usando una función predefinida, cargamos PSVPAG */
    /* con la dirección de la página donde el array Cadena */
    /* está ubicado */
    PSVPAG = __builtin_psvpage(Cadena15);
    /* Inicializamos los bits correspondientes */
    CORCONbits.PSV = 1;
}

```