CENTRO DE INGENIERÍA
Y
DESARROLLO INDUSTRIAL

ESPECIALIDAD DE TECNÓLOGO EN MECATRÓNICA

005820

"ABASTECEDORA AUTOMÁTICA DE TAPA"

Presenta: **Rafael Ocampo Martínez**

Marzo del 2006

ÍNDICE.

1. Introducción.

2. Marco Conceptual

3. Definición del problema.

   3.1 Descripción del equipo

4. Propuesta para el abastecimiento automático de tapas

   4.1 Marco teórico del PIC
   4.2 Esquema de control
   4.3 Diagrama de flujo y estructura del programa para el PIC16F84
   4.4 Explicación y funcionamiento del circuito
   4.5 Estructura mecánica para el montaje del contenedor de tapa.

5 Conclusiones.

6 Referencias.

1. INTRODUCCION.

La automatización es un sistema donde se trasfieren tareas de producción, realizadas habitualmente por operadores humanos a un conjunto de elementos tecnológicos.
Un sistema automatizado consta de dos partes principales:

Parte de Mando

Parte Operativa

Los objetivos de la automatización son mejorar la productividad de la empresa, reduciendo los costos de la producción y mejorando la calidad de la misma, mejorar las condiciones de trabajo del personal suprimiendo los trabajos penosos e incrementando la seguridad, mejorar la disponibilidad de los productos pudiendo proveer las cantidades necesarias en el momento preciso, simplificar el mantenimiento e Integrar la gestión y la producción entre otras.

En el presente proyecto se persigue utilizar la tecnologías de componentes de reciente aparición, tales como son los microcontroladores PIC's, los cuales presentan las ventajas de tener un bajo costo de implementación y aprovechando las capacidades de la maquinaría instalada, lograr una mejora al proceso de embotellado teniendo como resultado directo una disminución de la carga de trabajo del operario y un efecto favorable en el incremento de la productividad.

De tal forma que se realizará primero una descripción de la empresa donde se generó la oportunidad de este proyecto, posteriormente se expone el problema a resolver. En la parte de desarrollo de la propuesta se inicia con una referencia a la teoría de microcontroladores haciendo énfasis en la familia 16F84 que fue el PIC utilizado, posteriormente se presenta el desarrollo de la tarjeta electrónica y finalmente se propone el programa que se cargará al PIC.

En la parte final se anexan las hojas de datos relevantes del PIC16F84.

2. MARCO CONCEPTUAL

La empresa la madrileña donde se desarrolla el proyecto, inicia operaciones como negocio familiar en 1911 conociéndose primero como compañía LEON VELASCO y ya para 1935 funciona oficialmente como madrileña S.A. de CV en un terreno industrial en Ixtapalapa D.F. La planta San Juan del Río de la madrileña S.A. de CV inicia sus operaciones oficialmente en agosto de 1969, y las instalaciones que se tenían en ese entonces era un salón de fermentación con 8 tanques de 62 mil litros así como el equipo para realizar la molienda de la uva.

También contaban con un espacio destinado para 10 tanques de concreto de 190 mil litros cada uno, una destilería, un salón para añejamiento y un edificio destinado a ser oficinas, posteriormente en este lugar se acondicionó un pequeño laboratorio. Durante esta época la manufactura y la venta de jerez no rebasaba el millón de litros.

La destilería comenzó a funcionar en 1970 con el propósito de elaborar el brandy, pero debido a situaciones del mercado, para 1974 solo se destilaba alcohol de caña.

Durante la década de los 70 la venta de jerez creció hasta un volumen de 7 millones de litros suficiente para pensar en una ampliación, así que para 1972 y 1973 se construyó lo que se conocía como salón de jerez, la cual tenia instalado en su interior 19 tanques de madera de 70 mil litros cada uno y 20 tanques pequeños de madera con capacidad de 7 mil litros, al finalizar esta obra se construyó la cava.

En la década de los 80 debido a que muchas actividades que realizaba la planta de la madrileña en el D.F. se empezaron a trasladar a San Juan del Río todos los tanques de madera y se empezaron a cambiar por tanques de acero inoxidable.

En 1981 se construyó el salón para llevar acabo las labores de envasado del producto, ya que en este año, se empezó a envasar la bebida tipo jerez en San Juan del Río.

La manufactura de rompope y licor de café llega en 1982 y a finales del mismo año todos los demás licores. Esto se instaló donde operaba la destilería, cabe señalar que todos los licores se elaboraban en la planta de la madrileña en México y que en la producción de bebidas destiladas era mínima.

En el año de 1982 también fue el año en que la planta opera de manera integral, como una sola planta, ya que hasta antes de este año opero como dos plantas separadas, una como destilería y la otra como vinatería. Para 1986 se construyó un nuevo salón para barricas ya que durante un tiempo se tuvieron barricas de reposo a la intemperie. Y aunque la venta del jerez creció mucho durante esta época se tuvieron bastantes años con volumen de 4 a 4.5 millones de litros, estabilizándose a principios de los 90 en 7 a 7.5 millones de litros e incluso en el 2001 llegando a 8.4 millones de litros.

Otro tipo de bebida con gran crecimiento fueron las destiladas, el vodka, ron y sobre todo el tequila provocaron que se produjeran mas ampliaciones a la planta. Los terrenos de la planta San Juan del Río son de aproximadamente 15 hectáreas.

A finales de 1997 se inició la instalación de 4 tanques con capacidad de un millón de litros cada uno. Ya para inicios de 1998 se inició la construcción de la bodega actual de producto terminado, concluyendo dicha obra para el segundo cuatrimestre de 1999. También durante este periodo se derribó un edificio y se construyeron calles, todo lo anterior originado por el crecimiento en ventas de 180 mil cajas de producto a 1.5 millones de cajas, mas la distribución de productos importados por 0.5 millones de cajas.

Durante el año 2000 y 2001 se integro el edificio de producción, el área donde estuvieron los tanques de concreto se limpio como área de producción. En el 2001 se movió el equipo de molienda para construir las nuevas oficinas administrativas, aún así el terreno construido es de 5 hectáreas y quedan 10 como terreno agrícola.

3. DEFINICIÓN DEL PROBLEMA.

De acuerdo con las necesidades de crecimiento que demanda el mercado, el objetivo de este proyecto es hacer que el recipiente que abastece de taparrosca a la engargoladora "CAPMATIC" sea llenado automáticamente por otro recipiente de mayor capacidad, para evitar que el operador este vaciando manualmente la taparrosca, aumentando con ello la productividad del proceso.

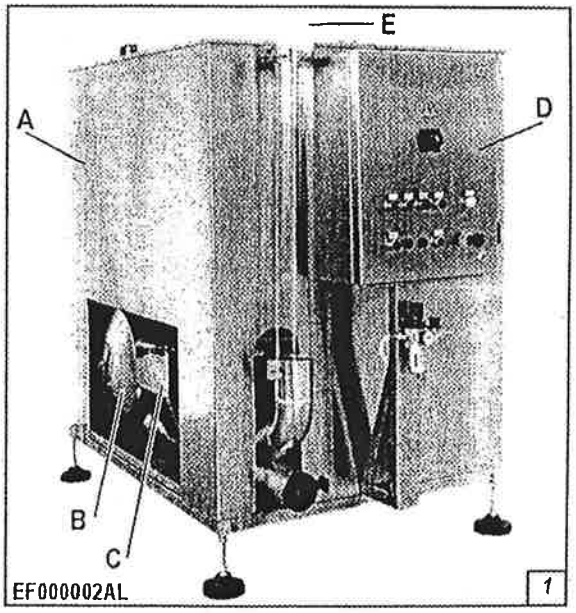
El recipiente que abastece a la máquina taponadora del área de envasado que existe en la empresa tiene que ser llenado a mano cada vez que se termina, lo que origina que aproximadamente cada 20 minutos el operador tiene que subir por unas escaleras manuales para rellenar el recipiente de la máquina, la cual trabaja con un ritmo de envasado de aprox. 2 500 botellas por hora, por lo que resulta poco eficiente el proceso de reposición de tapas para la operación continua de la máquina. Por lo que se pretende aprovechar una maquina abastecedora de tapa que actualmente surte a otra línea de envasado y la cual tiene un depósito de taparrosca de mucho mayor capacidad con lo cual se eliminará el tiempo del operador de subir y rellenar el recipiente de taparoscas cada vez que se termine y solo será necesario reponer aproximadamente una vez al día el nivel del contenedor.

3.1 Descripción del equipo.

Alimentador neumático EF002AL MBF

El alimentador neumático Serie AP de tapones esta construido en acero inoxidable y está equipado de un soplador para el envío de tapones, también cuenta con u vibrador para la caída de los tapones y un tablero eléctrico de mando. Incluye también 10 mts. de tubería en material plástico transparente de 100 mm de diámetro. La tolva de carga tiene una capacidad de 1000 lts.

En la siguientes figuras se muestra la máquina y sus componentes principales:



005820

7

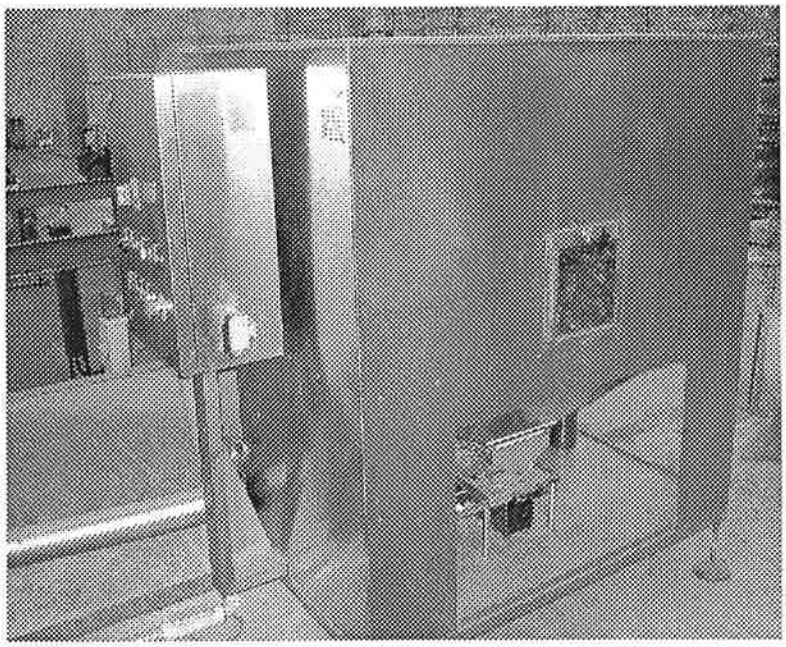


Fig. 3.1 Alimentador Neumático EF00002AL

Ciclo de funcionamiento.

Los tapones que se encuentran en la tolva de carga bajan en el deslizadero por caída que está causada por la conicidad de la tolva. Sólo en algunos casos se activa un vibrador neumático sujeto a la misma tolva. Después los tapones que se caen en el deslizadero son distanciados y encauzados al interior de la tubería de transporte, por medio de un vibrador variable para poder dosificar la cantidad de tapones.
Después los tapones son enviados al distribuidor de tapones de la maquina taponadora por medio del aire que pasa a través del ventilador. El arranque del alimentador estará supeditado a la petición de tapones señalada por el sensor que se ubicará en el distribuidor de tapones de la máquina taponadora..

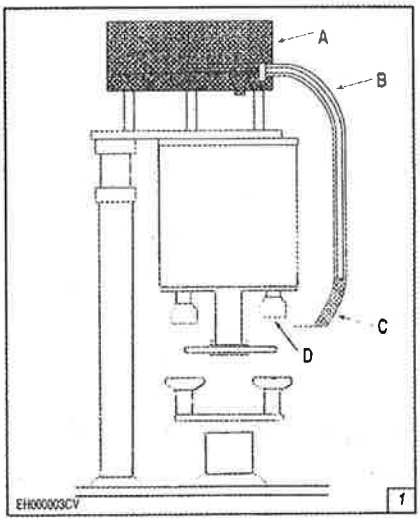
Consideraciones de seguridad:

La máquina se ha construido reuniendo los requisitos fundamentales para el empleo en el sector agrícola-alimenticio.
La máquina estándar no está planteada para trabajar en lugares con atmósfera potencialmente explosiva.
La limpieza, el mantenimiento y la revisión periódica son indispensables para el funcionamiento correcto y la larga duración de la máquina; evítese el empleo de agua o disolventes corrosivos.
EL montaje de las tuberías de transporte de tapones tiene que realizarse de forma que no existan aristas interiores que podrían dañar los tapones y por consiguiente causar inconvenientes en su distribución.

Máquina taponadora MF00013CV MBF

La taponadora CAPMATIC V permite trabajar tapas de aluminio. La máquina tiene una estructura sólida, necesita un mantenimiento reducido y está totalmente construida en acero inox en las piezas que llegan eventualmente a tener contacto con los líquidos.

Sus características permiten emplear la máquina en líneas de mediana y alta producción. La CAPMATIC 240/4V cuenta con los siguientes componentes principales:



A. Distribuidor superior de tapones.

B. Canal de bajada de tapones

C. Distribuidor de tapones en línea

D. Cabezas de taponamiento

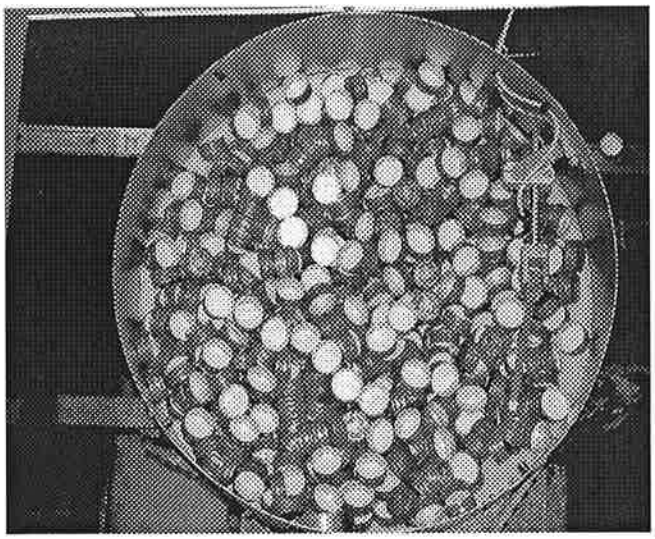


Fig. 3.2 Vista superior de la tolva abastecedora de tapa

La máquina tiene una estructura en acero soldado de cédula gruesa y protegido por capas de barnizado epoxídico. Los tornillos reguladores en los pies de apoyo están construidos totalmente en acero inoxidable. Cuenta también con dos hojas transparentes de policarbonato provistas de un interruptor de seguridad. El cuadro eléctrico de mando esta colocado al lado de la máquina y cuenta con un tablero de pulsadores multifuncionales con display digital para visualizar datos y funciones.

Ciclo de funcionamiento

Durante la rotación de la torreta los tapones bajan oportunamente orientados en el canal de bajada donde en la parte Terminal llegan al contacto con la botella a través de la distribución rápida del tapón.

En la fase de contacto entre la cabeza taponadora y la botella se realiza el taponamiento, en el cual dos pequeños rodillos superiores de la cabeza enroscan la cápsula siguiendo el enroscado que hay en el gollete de la botella, mientras que otros dos rodillos inferiores perfilan la tapa en la parte inferior del diámetro interior para garantizar su cierre. Al mismo tiempo se ejerce una presión vertical en el tapón para que el empaque del tapón se adhiera a la boca de la botella.

Fig. 3.3 Engargoladora

4. PROPUESTA PARA EL ABASTECIMIENTO AUTOMÁTICO DE TAPAS

Tal como se menciona en la introducción la propuesta para la automatización del proyecto se basa en un desarrollo con el microcontrolador PIC16F84 de Microchip, ya que por las necesidades y requerimientos es suficiente el uso de este micro por lo que se desarrollaran la teoría y características de este microcontrolador.

4.1 Marco Teórico del PIC

A continuación se muestra un listado de los componentes electrónicos utilizados para la realización del proyecto.

- Microcontrolador (PIC16F84)
- Led's indicadores
- Transistores (2N2222)
- Reguladores (LM7805, LM7812)
- Capacitores no polarizados
- Resistores
- Relay's
- Diodos (1N4001)
- Conectores
- Cristal (4 MHz)
- Sensor inductivo (24 VDC)
- Push buttons

Definición de un microcontrolador

Un microcontrolador es un computador completo de limitadas prestaciones que está contenido en el chip de un circuito integrado.

El PIC (Programable integrated Circuit) es un circuito integrado programable que contiene todos los componentes de un computador. Se emplea para controlar el funcionamiento de una tarea determinada y, debido a su reducido tamaño, suele ir incorporado en el propio dispositivo al que gobierna. Esta última característica es la que le confiere la denominación de <<controlador incrustado>> (embedded controller).

El microcontrolador es un computador dedicado a realizar una tarea específica. En su memoria sólo reside un programa destinado a gobernar un aplicación determinada; sus líneas de E/S (entradas y salidas) soportan el conexionado de sensores y actuadores del dispositivo a controlar y todos los recursos complementarios disponibles tienen como única finalidad atender a sus requerimiento. Una vez programado y configurado el microcontrolador solamente sirve para gobernar una tarea asignada.

La mayoría de los microcontroladores utilizan un conjunto de instrucciones de tipo RICS (computadores de juego de instrucciones reducido). Con el RISC se suelen ejecutar la mayor parte de las instrucciones en un solo pulso de reloj a diferencia de otros como el CISC (computadores de juego de instrucciones complejo), con el CISC se logran ejecutar instrucciones más poderosas pero se sacrifica la velocidad, ya que una sola instrucción puede ejecutarse en cinco ciclos de reloj.

En realidad un microprocesador y un microcontrolador no son la misma cosa. Los Pic's se integran por una unidad que posee en su interior al microprocesador y a los elementos indispensables para que pueda funcionar como una minicomputadora contenida en un solo chip.

Un microprocesador es solamente la CPU (unidad central de proceso), las líneas de sus buses de direcciones, dados y control para permitir conectarle con la memoria y los módulos de entradas y salidas y configurar un computador debe se implementado por varios circuitos integrados. Se dice que un microprocesador es un circuito abierto porque su configuración es variable de acuerdo con la aplicación a la que se destine.

Un microcontrolador integra la CPU y todos los periféricos en un mismo chip y su capacidad de memoria y características no pueden alterarse.

Arquitectura interna

Los microcontroladores poseen una memoria de programa que es el lugar donde deben alojarse los datos para decirle al chip que es lo que debe de hacer; una memoria de datos por donde ingresen las señales que debe procesar el programa, una unidad aritmética y lógica donde se desarrollen todas las tareas, una unidad de control que se encargue de supervisar todos los proceso y los puertos de entrada y salida para que el PIC tenga contacto con el exterior. En la figura 4.1 se observa la arquitectura simplificada del PIC 16F84.

Fig. 4.1 Arquitectura interna simplificada del PIC16F84

La memoria de programa es una EEPROM (electrical eraser memory only read), memoria de sólo lectura, que se programa por tensión; es decir que basta con introducir los datos con cierto nivel de tensión para que éstos borren el programa anterior y graben uno nuevo.

La memoria de datos es una RAM (random acces memory) es una memoria de lectura y escritura de 128 posiciones pero que sólo tiene implementado las primeras 48 posiciones (desde 00 a 2F en hexadecimal). De estos 48 registros, los primeros 12 son fijos y cumplen un propósito determinado, en tanto que desde el 13 hasta el 48 son registros de propósito general, en donde el programa puede indicar que se almacene un dato para ser tomado más tarde.

Características del PIC 16F84

El PIC 16F84 pertenece a la familia de microcontroladores 16CXX fabricados por Microchip cuya arquitectura, capacidades y juego de instrucciones y especialmente su bajo costo lo hacen muy útil en pequeñas aplicaciones así como parte de otras aplicaciones de mayor complejidad sustituyendo a gran cantidad de circuitos lógicos convencionales.

Principales características:

- Encapsulado plástico Dip de 18 pines.
- Memoria EEPROM de 1K X 14 (1 kbyte con palabras de 14 bit's) tipo flash.
- Memoria de datos RAM de 68 bytes.
- Pila stack de 8 niveles.
- Cuatro tipos de interrupciones.
- Juego de 35 instrucciones.
- Frecuencia de trabajo de 10 MHz máxima.
- Procesador tipo RISC.

- Un temporizador TMR0.
- Perro guardián (WTD).
- 13 líneas de entradas y salidas digitales (5 puerto A y 8 puerto B).
- Voltaje de alimentación (V_{DD}) de 2 a 6 VDC.
- Voltaje de grabación (V_{PP}) de 12 a 14 VDC.
- Tecnología CMOS.

Puertos de entradas y salidas

El PIC6F84 tiene trece pines que se pueden programar como entrada o como salida. Estos pines (patas) están organizados formando dos puertos paralelos de entrada y salida llamados: puerto A que cuenta con 5 pines disponibles y el B con 8 pines como se puede ver en la figura 4.2. Cada pin puede ser dispuesto por el programa para operar como entrada o como salida. Cada pin tiene un resistor de pull-up (resistor conectado a fuente) interno que puede ser desconectado mediante el programa. Estos resistores se desconectan automáticamente si un pin se dispone como pin de salida debido a que las salidas ya tiene la posibilidad de entregar corriente desde una fuente con un transistor. Todos los resistores de pull-up se conectan o desconectan al mismo tiempo (no existe un comando que los conecte independientemente.

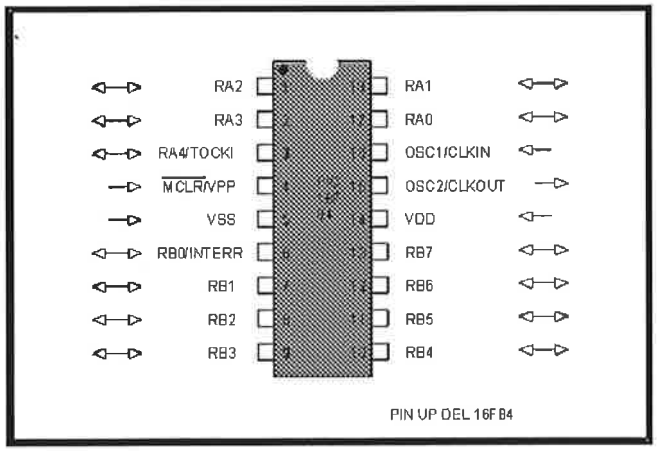


Fig. 4.2 PIC16F84

Los puertos de salida o de entrada trabajan con una tensión normalizada de 0 ó 5 V. Es decir que en la terminal correspondiente a un uno lógico se obtendrán 5 V y en el correspondiente a un cero lógico se obtendrá 0 V. Cuando un pin de un puerto se utiliza como salida puede proveer hasta 20 mA y cuando se utiliza como entrada puede consumir 25 mA. También se debe tener en cuenta que el puerto A en su totalidad sólo puede entregar 80 mA de corriente máxima si se utiliza como salida y se emplea como entrada puede consumir 50 mA de corriente máxima. Estos valores se incrementan a 100 y 150 mA respectivamente para el puerto B.

El reloj oscilador (Clock)

Los PIC poseen un oscilador configurable por programa de características muy amplias. Cuando no se requiere de mucha precisión se puede trabajar con un oscilador RC (resistivo capacitivo) conectado como en la figura 4.3

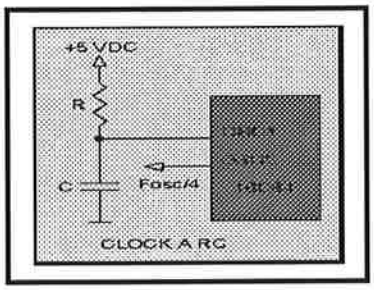


Fig. 4.3 Oscilador RC

Para circuitos que requieran una gran precisión se puede trabajar con un cristal de frecuencia baja, media o alta, el arreglo de este tipo de oscilador es el que muestra la figura 4.4

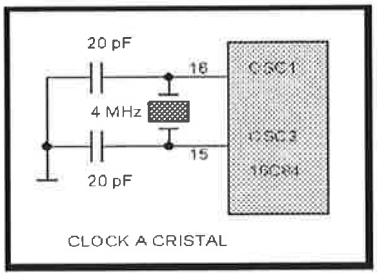


Fig. 4.4 Oscilador de cristal

Usualmente se utiliza un cristal de 4MHz, que internamente la frecuencia del cristal se divide entre cuatro obteniendo una frecuencia de trabajo de 1 MHz, asegurando que cada instrucción se ejecute exactamente cada 1ms (milisegundo).

Reset del microcontrolador

El PIC se resetea cuando el pin 4 (MCLR es negado), es decir, cuando se conecta a tierra. Para simplificar el circuito de reset el PIC incorpora un temporizador interno que permite realizar un reset automático cuando se aplica tensión de 5V. En estos casos el circuito externo de reset sólo implica el uso de un resistor de 10KΩ entre la pata y la fuente como se muestra en la figura 4.5.

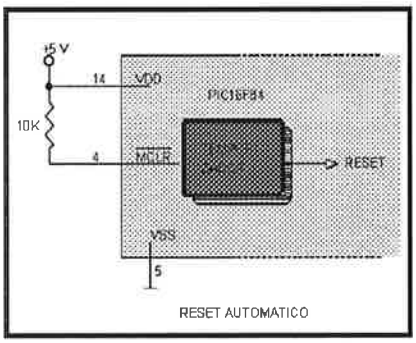


Fig. 4.5 Reset automático

En muchos circuitos es necesario realizar un reset manual como es el caso del proyecto y para ello existen dos posibilidades, una de ellas es utilizar sólo el temporizador interno (por programa) y la otra es agregar una constante de tiempo exterior como se muestra en la figura 4.6, en el segundo circuito C1 provee un retardo al encendido o posterior al pulsado de reset porque C1 se cargará lentamente a través de R1 con una constante de tiempo de (22KΩ x 100μ = 220ms). El resistor R3 limita la corriente de descarga de C1 a valores

compatibles con sus características de corriente de pico máxima. D1 descarga a C1 cuando la tensión de fuente decae para permitir un reset inmediato cuando la fuente se apaga y se enciende rápidamente. R2 limita la corriente de reset, tomada desde el microprocesador. Este segundo sistema se suele utilizar cuando se requiere un reseteado a través de varios metros de cable que podrían captar ruido, o alguna perturbación que pueda disparar el circuito en falso, (C1 reduce la impedancia del circuito de reset) eliminando los problemas de ruido.

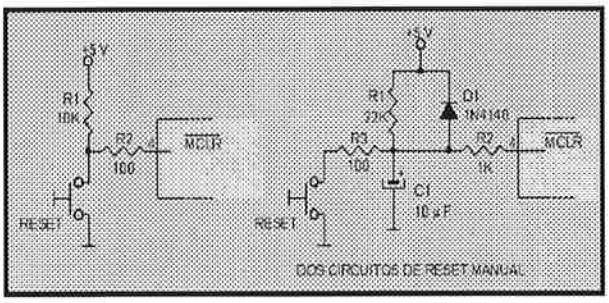


Fig. 4.6 Reset manual

Programación del microcontrolador

La utilización de los lenguajes más cercanos a la máquina (de bajo nivel) representan un considerable ahorro de código en la confección de los programas, lo que es muy importante dada la estricta limitación de la capacidad de la memoria de instrucciones. Los programas bien realizados en lenguaje ensamblador optimizan el tamaño de la memoria que ocupan y su ejecución es mucho más rápida que un programa grande.

Los PIC basan su programación en un set reducido de instrucciones que permiten editar el programa generando un archivo con extensión "asm". Existen programas que se encargan de traducir el programa editado en lenguaje ensamblador "asm" a lenguaje máquina o hexadecimal, a estos programas se les conoce como traductores o ensambladores.

Por lo tanto para editar un programa y cargarlo en un PIC se necesita:

Un utilitario o programa que nos permita escribir el juego de sentencias a cargar en la memoria de programa del PIC. Este utilitario es un "editor" y como ejemplo tenemos el MPLAB.

- Un utilitario o programa que nos permita "traducir" el programa escrito en ensamblador a lenguaje de máquina para que podamos cargarlo en el PIC. A ese programa se lo denomina "ensamblador" y como ejemplo se encuentra al MPASM.
- Un utilitario para "cargar" el archivo que tiene el pro-grama en lenguaje de máquina (archivo con extensión hexadecimal) en la memoria del PIC. A estos programas se los de-nomina cargadores" y como ejemplo tenemos al NOPPP," al PIC PRO, PICSHELL y a muchos otros que existen en el mercado.

Pero también existen otros utilitarios que permiten "simular" y "emular" un programa para ver cómo se comporta antes de cargarlo en el PIC. El simulador permite detectar errores en el programa (el MPLAB permite simular) para poder corregirlos. El emulador permite que "una computadora se comporte como PIC" siendo las patas del puerto, las correspondientes a las del PIC. Esto quiere decir que si se requiere hacer circuitos con PIC's, para poder trabajar adecuadamente se necesitará:

- Programa MPLAB: Para editar (escribir) un programa en lenguaje ensamblador y así generar el archivo con extensión asm. El MPLAB también permite "simular" el funcionamiento del programa para detectar errores y corregirlos.
- Programa MPASM: Para "ensamblar" o convertir el archivo asm en archivo hexadecimal con el objeto de cargarlo en el PIC.
- Programa NOPPP: O cualquier otro cargador que permita ingresar el archivo con extensión hexadecimal en la memoria de programa del PIC. También se puede emplear cualquier otro cargador. Todos los programas cargadores precisarán de un circuito "eléctrico" o hardware para poder realizar la función de carga.

Un cargador de PIC debe poseer por un lado el circuito que permita cargar el programa desde una computadora (hardware) y el programa que permita manejar a dicho circuito para realizar la operación de carga(software). El circuito debe ser muy simple.

El PICSHELL es un programa ampliamente difundido que sirve para cargar PIC's del tipo 16C84, 16F83 y 16F84.

Todos los editores de programas y programadores para los PIC's de Microchip circulan libremente para todos los usuarios, a través de internet en la página (http://www.microchip.com).

Las instrucciones para la programación de la gama de PICS 16FXX se divide en tres grupos, los cuales se presentan en las siguientes tablas:

Instrucciones orientadas a Bits

CODIGO MNEMONICO	DESCRIPCION	CODIGO DE OPERACION	MODIFICACION DE LOS FLAGS	NOTAS
BCF f,b	Pone a 0 el bit b del registro f	01 00BB BFFF FFFF	Ninguno	1,2
BSF f,b	Pone a 1 el bit b del registro f	01 01BB BFFF FFFF	Ninguno	1,2
BTFSC f,b	Skip si el bit b del reg. f es 0	01 10BB BFFF FFFF	Ninguno	3
BTFSS f,b	Skip si el bit b del reg. f es 1	01 11BB BFFF FFFF	Ninguno	3

Tabla 4.1 Instrucciones orientadas a bits

Instrucciones con literales y de control

CODIGO MNEMONICO	DESCRIPCION	CODIGO DE OPERACION	MODIFICACION DE LOS FLAGS	NOTAS
ADDLW K	(W)+ K → (W)	11 111x KKKK KKKK	C,DC,Z	
ANDLW K	(W) AND K → (W)	11 1001 KKKK KKKK	Z	
CALL K	Llamada a subrutina	10 0KKK KKKK KKKK		
CLRWDT	Clear del temporizador del WD	00 0000 0110 0100	Ninguno	
GOTO K	Go To dirección	10 1KKK KKKK KKKK	Z	
IORLW K	(W) OR K → (W)	11 1000 KKKK KKKK	Ninguno	
MOVLW K	K → (W)	11 00xx KKKK KKKK	Ninguno	
RETFIE	Retorno de una interrupción	00 0000 0000 1001	Ninguno	
RETLW K	Retorno con un literal en W	11 01xx KKKK KKKK	Ninguno	
RETURN	Retorno de una subrutina	00 0000 0000 1000	C,DC,Z	
SLEEP	Modo Standby	00 0000 0110 0011	Z	
SUBLW K	K – (W) → W	11 110x KKKK KKKK		
XORLW K	(W) XOR K → (W)	11 1010 KKKK KKKK		

Instrucciones Orientadas a Registros

CODIGO MNEMONICO	DESCRIPCION	CODIGO DE OPERACION	MODIFICACION DE LOS FLAGS	NOTAS
ADDWF f,d	(W)+(f) → (destino)	00 0111 DFFF FFFF	C, DC, Z	1,2
ANDWF f,d	(W) AND (f) → (destino)	00 0101 DFFF FFFF	Z	1,2
CLRF f	00 → (f)	00 0001 1fff ffff	Z	2
CLRW	00 → (W)	00 0001 0000 0011	Z	1,2
COMF f,d	Complemento de f [(#f)→ (destino)]	00 1001 DFFF FFFF	Z	1,2
DECF f,d	(f)-1 → destino	00 0011 DFFF FFFF	Z	1,2,3
DECFSZ f,d	(f)-1 → destino y si resultado es 0 salta	00 1011 DFFF FFFF	Ninguno	1,2
INCF f,d	(f)+1 → destino	00 1010 DFFF FFFF	Z	1,2,3
INCFSZ f,d	(f)+1 → destino y si resultado es 0 salta	00 1111 DFFF FFFF	Ninguno	1,2
IORWF f,d	(W) OR (f) → destino	00 0100 DFFF FFFF	Z	1,2
MOVF f,d	Mueve f → destino	00 1000 DFFF FFFF	Z	
MOVWF f	(W) → (f)	00 0000 1fff ffff	Ninguno	1,2
NOP	No operación	00 0000 0xx0 0000	Ninguno	1,2
RLF f,d	Rota f a la izquierda a través del carry → destino	00 1101 DFFF FFFF	C	1,2
RRF f,d	Rota f a la derecha a través del carry → destino	00 1100 DFFF FFFF	C	1,2
SUBWF f,d	(f)–(W) → (destino)	00 0010 DFFF FFFF	C,DC,Z	1,2
SWAPF f,d	Intercambia los niveles de f → destino	00 1110 DFFF FFFF	Ninguno	
XORWF f,d	(W) XOR (f) → (destino)	00 0110 DFFF FFFF	Z	

Tabla 4.2 Instrucciones orientadas a registros

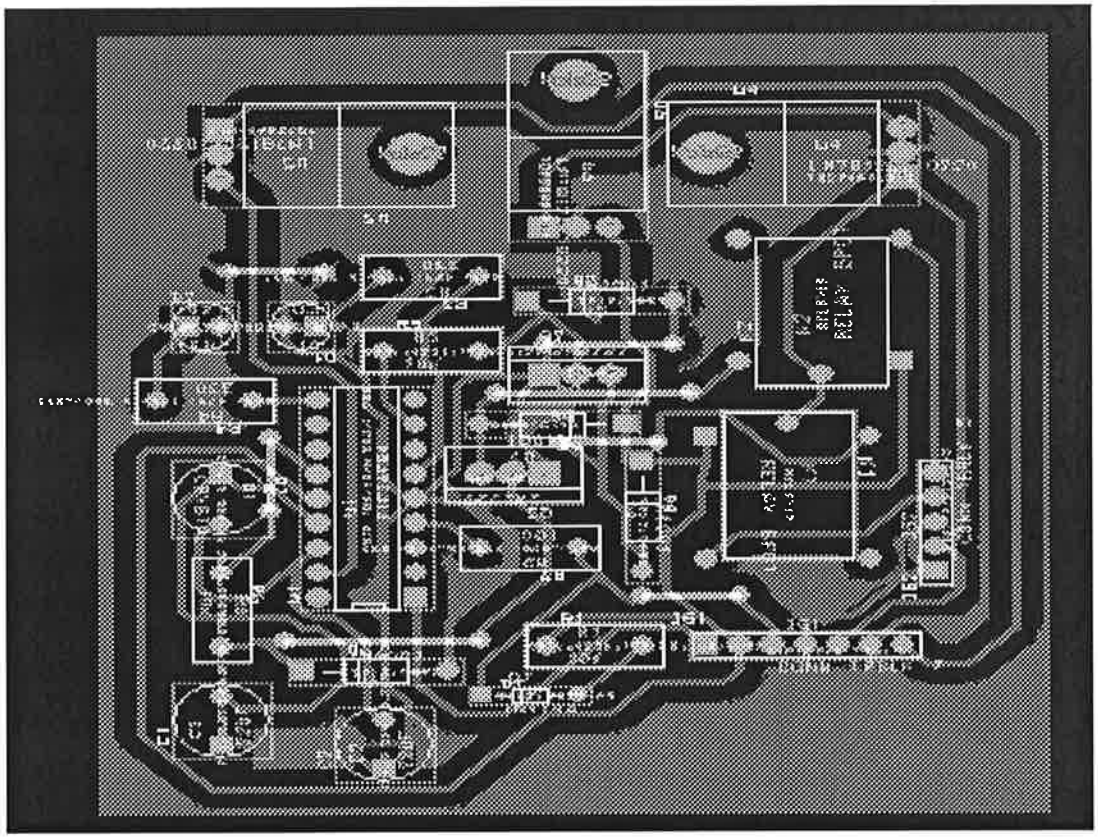


Fig. 4.8 Diagrama del circuito destinado a la placa de cobre

Las siguientes imágenes del circuito ya terminado

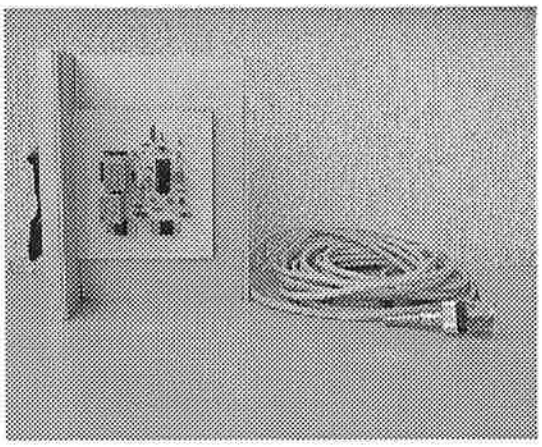


Fig. 4.9a Circuito ya terminado

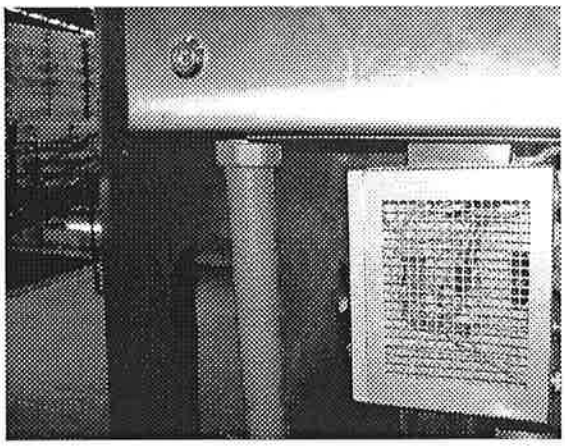
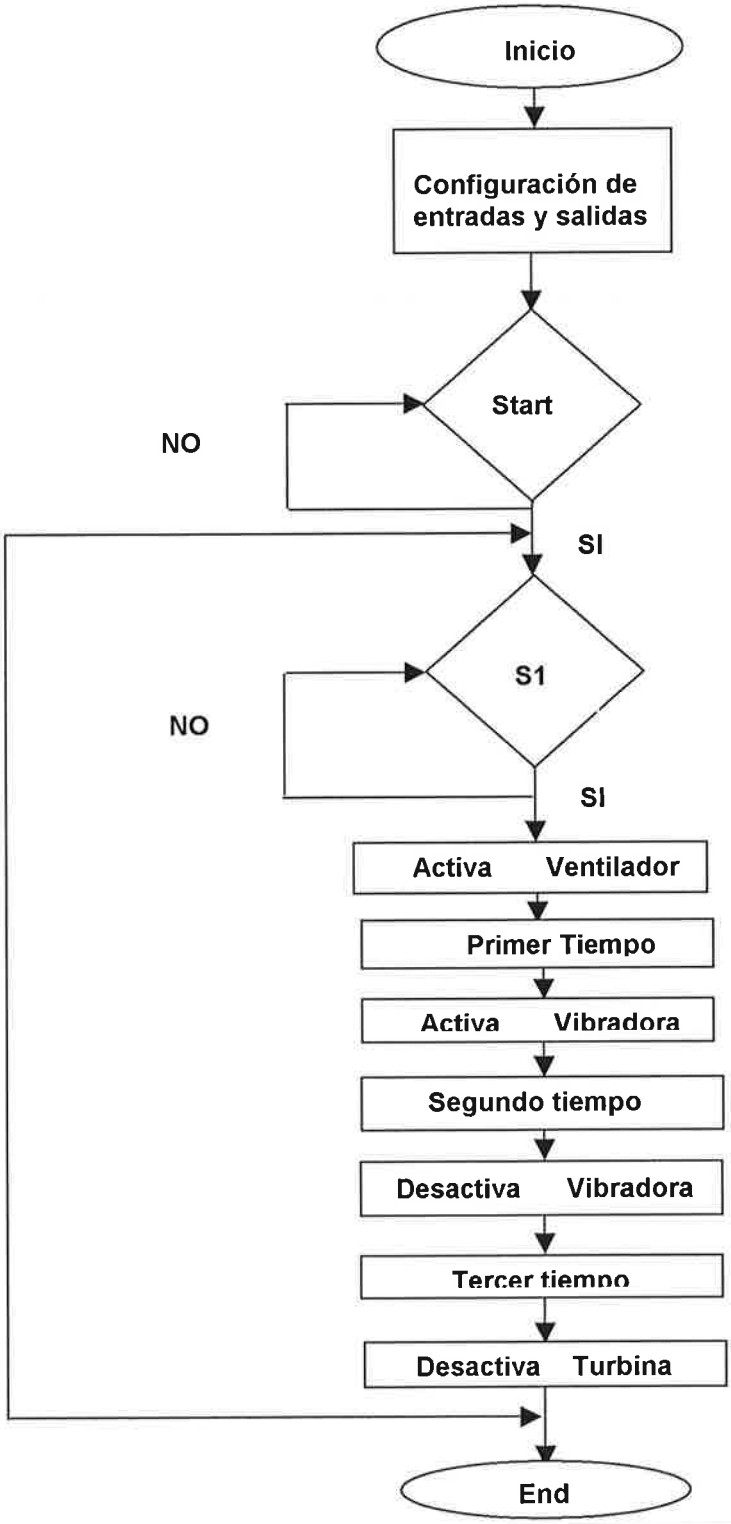


Fig. 4.9b Ubicación del circuito

4.3 Diagrama de flujo y estructura del programa para el PIC 16F84

A continuación se presenta el diagrama de flujo y el listado del programa original que se grabó en el PIC para la implementación del circuito:



4.4 Explicación y funcionamiento del circuito.

Funcionamiento del PIC 16F84

Esta parte del circuito, como se muestra en el anterior diagrama de flujo el PIC solo tiene que registrar las señales de señales de entrada y enviar las señales de salida.

La función del PIC una vez arrancado el sistema es monitorear la señal de arranque y una vez que se habilita esta orden, se pasa a monitorear la señal del sensor que detecta la falta de tapas; una vez que se ha activado el sensor se manda activar la salida primeramente del ventilador y mediante un retardo posteriormente se activa el vibrador, esta tarea la va realizar con un intervalo de tiempo que sea suficiente para llenar el recipiente de tapas; por lo que sea hará uso del temporizador interno que maneja el PIC.

La disposición de las señales es de la siguiente manera: La señal del sensor es recibida en el pin RA0, del puerto A del PIC, mientras que en el pin RA2 se recibe la señal del botón de start (botón que sirve para iniciar la operación de la máquina).

Por medio del puerto B del PIC y a través de los pines RB2 y RB0 se activa el soplador y un led indicador respectivamente. A través de los pines RB7 y RB4 se activa el vibrador y su indicador respectivamente.

Estructura del programa para el PIC 16F84

A continuación se presenta el listado del programa original que se grabó en el PIC para la implementación del circuito.

```
; UNIVERSIDAD TECNOLÓGICA DE SAN JUAN DEL RIO
; PROGRAMA PARA EL ABASTECIMIENTO AUTOMÀTICO DE TAPAS
; EN LA LINEA DE LLENADO DE LA PLANTA "LA MADRILEÑA"

    processor 16F84
      include<p16F84.inc>
      __config _XT_OSC &_WDT_OFF&_PWRTE_ON


;-----------------------------------------------------------------------------------------
;                DECLARACIÓN DE VARIABLES
;-----------------------------------------------------------------------------------------

#define START                    btfss  PORTA,2
#define SENSOR                   btfss  PORTA,0
#define SENSOR1                  btfsc  PORTA,0
#define ACTTURBINA               bsf  PORTA,2
#define ACTVIBRADORA             bsf  PORTA,7
#define DESACTTURBINA            bcf  PORTA,2
#define DESACTVIBRADORA          bcf  PORTA,7
#define VERDE_1                  bsf  PORTA,0
#define ROJO_1                   bsf  PORTA,4
#define VERDE_0                  bcf  PORTA,0
#define ROJO_0                   bcf  PORTA,4

CONT equ 0x0c
cont    equ 0x0d

;-----------------------------------------------------------------------------------------
;                CONFIGURACIÓN DE PUERTOS
;-----------------------------------------------------------------------------------------
org 0                            ;comienza en la dirección de memoria 0
goto start

org5
      start
              bsf        STATUS,5
```

```
            movlw   b'00011111'
            movwf   TRISA
            clrf    TRISB
            movlw   0x07
            movwf   OPTION_REG
            bcf     STATUS,5
            clrf    PORTA
            clrf    PORTB

;-----------------------------------------------------------------------------
;           PROGRAMA PRINCIPAL
;-----------------------------------------------------------------------------

INICIO      START
            goto    INICIO

s1          SENSOR
            goto    s1
            ACTTURBINA
            VERDE_1
            ROJO_0
            call    dosseg
            ACTVIBRADORA

s2          SENSOR1
            goto    s2
            DESACTTURBINA
            VERDE_0
            ROJO_1
            goto    s1

dosseg      clrf    INTCON
            movlw   .2
            movwf   cont
Loop3       movlw   0x10
            movwf   CONT
Loop1       clrf    INTCON
            movlw   b'00001100'
            movwf   TMR0
Loop2       btfss   INTCON,2
            goto    Loop2
            decfsz  CONT,F
            goto    Loop1
            decfsz  cont,F
            goto    Loop3
```

```
                return
end
```

Al principio del programa se declara el tipo de PIC que se va utiliza para poder utilizar las librerías que contienen las instrucciones, enseguida se define que tipo de oscilador a utilizará, en este caso es un oscilador XT (Cristal de cuarzo), también en el mismo renglón se especifica que el WTD_OFF (perro guardián) está apagado y por último se indica que el PWRTE_ON (reset de encendido) se encuentra activado.

Declaración de variables. En este bloque se habilitan en los registros todas las variables que se utilizarán en el programa, para guardar valores, cambiarlos o moverlos, también son empleadas para poder tener acceso a los registros y modificarlos o bien para entrar a un determinado banco de memoria.

Posteriormente se pasa a la configuración de puertos, en esta parte del programa se le indica al micro la configuración de sus puertos, estos pueden operar como entrada o salidas.

En el programa principal, una vez que ya se han declarado variables y se han configurado los puertos se comienza por editar el programa principal.

4.5 Estructura mecánica para el montaje del contenedor de tapa.

Es importante mencionar que para la implementación del circuito, se monto la placa en una caja de aluminio debidamente aterrizada para eliminar la captación de ruido y además se propone una estructura que consta de un soporte fijo al piso y un contenedor cónico de material malla al cual llegará la línea de envío de tapa y el cual servirá para minimizar el choque de las tapas de manera que se acopla tangencialmente y además cuenta con un deflector de goma en la parte superior que reduce el choque de la tapa cuando entra al contenedor. A continuación se muestran las imágenes de la estructura:

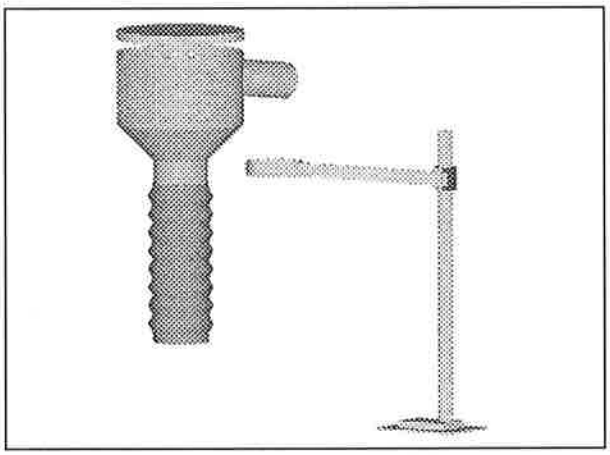


Fig. 4.10a Contenedor cónico

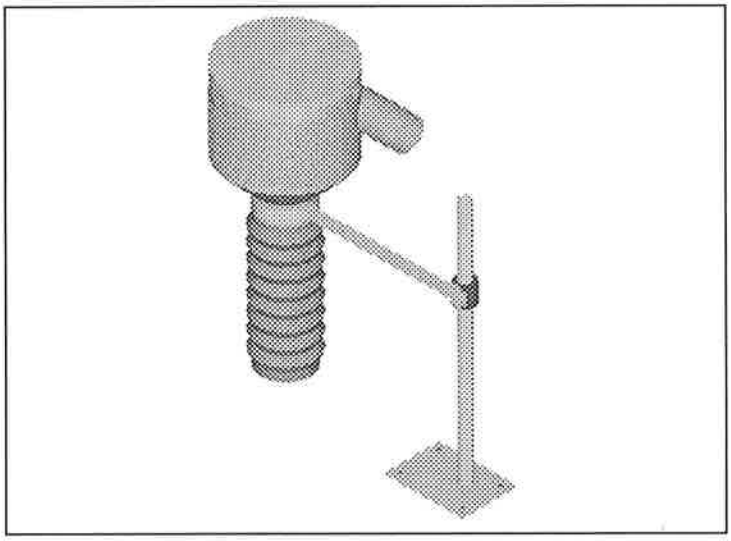


Fig. 4.10b Estructura ensamblada

5. CONCLUSIONES

La etapa de implementación será complementada con el diseño y construcción de la línea de transporte de las tapas y el contenedor que abastecerá al "magazine" distribuidor de la máquina, así como también la etapa de conexión dentro del tablero de control de la tolva alimentadora de tapas utilizando parte de su sistema de fuerza necesario para activar los equipos.

Aunque la aplicación presentada es sencilla, una vez implementada deberá satisfacer una necesidad de un ambiente industrial tal como sucedió en la empresa vitivinícola "La Madrileña" además me permite poner en práctica los conocimientos adquiridos durante la especialidad de Mecatrónica así como también tener un concepto integral de lo que puede abarcar un proyecto mecatrónico.

El proyecto presentado sienta las bases para poder seguir utilizando este tipo de dispositivos de control (PIC's) en aplicaciones cada vez más completas y novedosas, esto solo es el principio de las muchas aplicaciones que se pueden implementar con la integración de la diversas disciplinas con las cuales cuenta la Mecatrónica.

6. REFERENCIAS.

[1] Angulo Usastegui José Ma. / Angulo Martínez Ignacio, Microcontroladores PIC Diseño práctico de aplicaciones, 2ª Edición, McGraw Hill.

[2] Manual de empleo y mantenimiento ALIMENTATORE AP, MBF Tecnologie Per L'imbottigliamento.

[3] Manual de empleo y mantenimiento CAPMATIC V, MBF Tecnologie Per L'imbottigliamento

[4] www.webelectronica.com.ar Curso de PIC´s para estudiantes y aficionados (Mar'04)

[5] Saber electrónica no. 161, "Ahora programar es fácil"...p.7.