

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTTLÁN
INGENIERÍA MECÁNICA Y ELÉCTRICA ÁREA ELÉCTRICA Y ELECTRÓNICA



“Diseño y simulación de un control automático de trasvase de agua cisterna – tinaco para una casa habitación”

TESIS

Que para obtener el título de Ingeniero mecánico electricista

Presenta:

Gerardo Chávez García

Asesor:

Ing. Margarita López López

Cuauttlán Izcalli, Estado de México 2013



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Quiero dedicar este trabajo **A DIOS** por haberme permitido llegar a este punto y haberme dado salud para lograr mis objetivos.

Con mucho amor y cariño **A MIS PADRES** que me dieron la vida y han estado conmigo en todo momento. Gracias por darme una carrera para mi futuro y por creer en mí, por guiarme a ser un hombre de bien.

A MIS HERMANOS Y CUÑADOS que siempre me apoyaron para seguir adelante y que creyeron en mí.

Especialmente **A ELISA BEATRIZ CORONA ESPEJO** que incondicionalmente me apoyó y que ha estado alentándome para seguir adelante.

Agradezco a la **UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO** por abrirme las puertas del conocimiento y a todos mis profesores que influyeron en mi desarrollo profesional, en especial **A MI ASESOR DE TESIS** por sus consejos y quien me apoyó para la culminación de este trabajo.

Muchas gracias a todos.

INDICE

INTRODUCCIÓN -----	1
OBJETIVOS -----	1
METODOLOGÍA -----	2
ANTECEDENTES	
Funcionamiento básico del sistema de trasvase de agua cisterna – tinaco-----	3
Medición de fluidos-----	6
Medición mecánica	
• Medidor por sonda-----	6
• Medidor por cinta y plomada-----	6
• Medidor por flotador-----	6
Medición electrónica	
• Medidor por flotador magnético-----	6
• Medidor conductivo-----	7
• Medidor capacitivo-----	7
• Medidor por ultrasonido-----	7
• Medidor por vibración-----	7
 CAPÍTULO 1. CONDICIONES DE OPERACIÓN DEL PROYECTO	
Funcionamiento del control automático-----	9
Diagrama de flujo-----	10
• Descripción del diagrama de flujo-----	11
Tablas descriptivas de la operación trasvase de agua-----	12
• El proceso de llenado-----	12
• El proceso de uso de la reserva de agua-----	13
• Las fallas-----	14

CAPÍTULO 2. EL HARDWARE

Bloques que conforman el hardware del control automático-----	16
Elementos externos conectados al módulo de control	
• Los interruptores de nivel con flotador magnético-----	17
• El interruptor de flujo tipo paleta-----	20
• La bomba centrífuga con arranque por capacitor-----	22
• La electroválvula-----	25
Elementos internos que conforman el módulo de control	
El microcontrolador PIC 16F873A como elemento principal-----	28
• La arquitectura Harvard-----	30
• Distribución y descripción de pines del microcontrolador-----	31
• El circuito de reset en MCLR (Master Clear Reset)-----	33
• El oscilador externo-----	34
• Características eléctricas-----	37
• La etapa de entrada-----	38
El display LCD alfanumérico-----	39
• El conjunto de instrucciones-----	41
• Descripción de terminales-----	42
• Bus de datos-----	43
La alarma audible-----	45
Etapa de potencia para la bomba centrífuga y la electroválvula-----	47
• El optoacoplador MOC3031-----	47
• El triac BTA24-800BW-----	49
• La red Snubber-----	51
• Etapa de potencia-----	51

La fuente de alimentación-----	53
• La transformación-----	54
• La rectificación de onda completa-----	55
• El filtrado-----	56
• La regulación-----	57
Diagrama electrónico del control automático-----	59

CAPÍTULO 3. EL SOFTWARE

Lenguaje de programación Flowcode para microcontroladores PIC-----	61
Elección y configuración del microcontrolador-----	64
Conexión y propiedades externas de los interruptores-----	68
Conexión y propiedades externas del display LCD-----	73
Conexión y propiedades externas de los indicadores LED-----	78
Las macros de componentes y del programa-----	82
El diagrama de flujo del control automático-----	86
La simulación del programa-----	88
La compilación del programa-----	95
Código en lenguaje C-----	99
Grabación del programa en el microcontrolador-----	134

CAPÍTULO 4. LA SIMULACIÓN FÍSICA Y CONSTRUCCIÓN DEL PROTOTIPO

Simulación en protoboard-----	140
Diseño de la placa de circuito impreso principal-----	143
Diseño de la placa de circuito impreso secundaria-----	152
Construcción de las placas-----	155
Montaje en el gabinete-----	157

CAPÍTULO 5. PRESUPUESTO DEL PROYECTO

Resumen de costos del prototipo-----159

CONCLUSIONES-----161

BIBLIOGRAFÍA-----162

INTRODUCCIÓN

El tema que se tratará en este trabajo, es el de diseñar, simular y construir un control automático de trasvase de agua cisterna – tinaco para una casa habitación.

El agua es muy importante en una casa habitación, sin ella no podríamos realizar varias tareas cotidianas como lavar los trastes, lavar la ropa, bañarse, etc. Y el no tenerla implica graves problemas como la salud o dificultad para realizar dichas tareas.

Por eso en zonas donde el agua escasea, la gente opta por la instalación de un sistema de trasvase de agua cisterna – tinaco en sus viviendas para garantizar que no falte agua, pero para la mayoría de la gente que cuenta con dicho sistema, aun así sigue teniendo problemas graves por el hecho de no contar con un buen control automático que realice de manera óptima, el proceso de trasvase de agua.

A consecuencia de ello, se tiene el primer problema que cuando el sistema no está automatizado, el usuario tiene que activar la bomba centrífuga de una manera manual para llenar el tinaco, pero por realizar otras tareas en casa, se le olvida la desactivación, originándose gasto de recursos por desperdicio de agua, producto del desbordamiento en el tinaco y por consiguiente, gastos económicos innecesarios.

El segundo problema se da cuando el usuario tiene que salir al patio para observar si tiene agua en la cisterna, o subir con una escalera al tinaco, con el riesgo de sufrir un accidente por resbalarse y caer.

El tercer problema implica un mantenimiento a corto plazo de la bomba centrífuga, ya que al activarse continuamente, sus partes internas sufren un desgaste prematuro y en el peor de los casos al accionarse sin agua, se sobrecalentaría hasta dañarse completamente, ocasionando aún más gastos económicos innecesarios, contando los gastos por consumo de energía eléctrica desaprovechada.

OBJETIVOS

Se pretende que estos problemas se resuelvan y en base a mis conocimientos adquiridos en la carrera, diseñar un control automático de trasvase de agua cisterna – tinaco que proveerá al usuario un proceso de trasvase de agua autónomo, o sea sin la necesidad de activar el sistema manualmente y además una reserva de agua adicional para el caso de que cuando no haya suministro de agua potable, esta pueda ser utilizada mientras se restablece el servicio.

El control automático dispone de un display LCD que mostrará mensajes del estado del sistema, informando al usuario el nivel de agua que se tiene tanto en la cisterna, como en el tinaco, también mostrará un mensaje si el sistema falla. Se incluye tres indicadores led, uno parpadea si se ha originado una falla, y los otros dos se iluminan, dependiendo si la bomba centrífuga o la electroválvula está en funcionamiento.

También se incluye una alarma audible para que al usuario se le alerte de que se ha originado una falla en el sistema. Se pretende con este control, proveer al usuario confort, seguridad y economía.

METODOLOGÍA

La Metodología que se emplea para este proyecto, se basa en el método científico – experimental, ya que se parte de la formulación de la problemática a resolver, se investiga entonces la forma de resolver estos problemas, teniendo como base teórica, los conocimientos de electrónica, microelectrónica y de la automatización y el control. El siguiente paso es la de comprobar si funcionará el diseño electrónico que se plantea para solucionar dichos problemas, seguido del desarrollo del proyecto experimental analizando resultados y por último se concluye dando un resumen de lo obtenido.

En el primer capítulo se da a conocer las condiciones de operación del control automático, es decir, como va a reaccionar el control (toma de decisiones) a partir de los pasos del proceso de trasvase de agua cisterna - tinaco. Utilizando para la explicación, el diagrama de flujo y las tablas descriptivas.

En el segundo capítulo se explica detalladamente todos los bloques que conforman el hardware del control automático, sus circuitos, conexión, etc. Así como los elementos externos que se integran al control.

En el tercer capítulo se describe todo lo referente a la programación del microcontrolador en base al lenguaje gráfico de diagramas de flujo Flowcode, incluyendo la simulación del sistema.

En el cuarto capítulo se describe como se construye el prototipo y se prueba su funcionamiento físicamente.

En el quinto capítulo se da a conocer el presupuesto utilizado para la construcción del mismo.

ANTECEDENTES

FUNCIONAMIENTO BÁSICO DEL SISTEMA DE TRASVASE DE AGUA CISTERNA – TINACO

Para poder diseñar el control automático que automatizará el sistema de trasvase de agua cisterna – tinaco, es necesario partir del análisis del funcionamiento manual de dicho sistema que a continuación se describe.

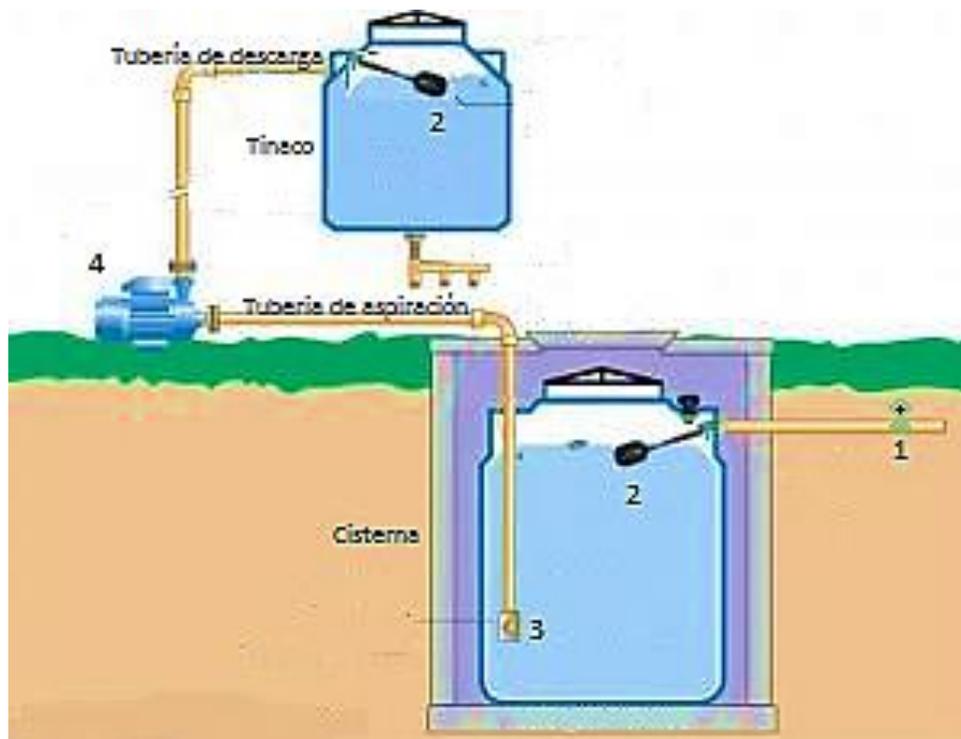


Figura 1. Sistema de trasvase de agua.

- 1) Válvula de compuerta principal que permite abrir o cerrar el paso del agua.
- 2) Válvula con flotador que impide que se sobrellene la cisterna.
- 3) Válvula de retención vertical con pichancha que sirve para retener agua en el tubo de aspiración de la bomba, lo que evita que aire dentro del tubo afecte negativamente al desempeño de la bomba, provocando aspiración de aire en lugar de agua lo que provocaría un sobrecalentamiento y la posible rotura de su sello mecánico. La pichancha retiene partículas sólidas que puede haber flotando en el agua.
- 4) Bomba centrífuga que succiona agua de la cisterna, para depositarla en el tinaco.

Ingresa agua potable de la toma principal proporcionada por el proveedor del servicio de agua potable, llenando así la cisterna. Cuando el nivel de agua en la cisterna llega al máximo de su capacidad, la válvula con flotador se cierra, impidiendo que ingrese más agua y evitando así el desbordamiento de la misma.

Al contener agua en la cisterna y haciendo funcionar la bomba centrífuga, empezará a aspirar agua por medio de la tubería de aspiración, pasando primero por la válvula de retención vertical con pichancha, haciendo subir el agua por la tubería de descarga con dirección al tinaco.

Al llenarse el tinaco, la válvula con flotador impide el ingreso de más agua, evitando el desbordamiento. La bomba centrífuga se tendrá que dejar de funcionar, cumpliendo así el ciclo de trasvase.

Teniendo éste análisis, se podrá automatizar el sistema, remplazando y agregando algunos elementos que a continuación se detallan.

Se remplaza el uso de la válvula con flotador en la cisterna, por una electroválvula que hará la misma función, solo que ésta puede ser controlada por medio de una señal eléctrica.

Se agregaran cinco interruptores de nivel con flotador magnético que censarán el nivel de agua existente en la cisterna y el tinaco.

También se agregará un interruptor de flujo en la toma principal, lo cual nos censará la presencia o no del flujo de agua.

Teniendo estos elementos, incluyendo la bomba centrífuga, solo faltará implementar el control automático que controlará el proceso de trasvase de agua.

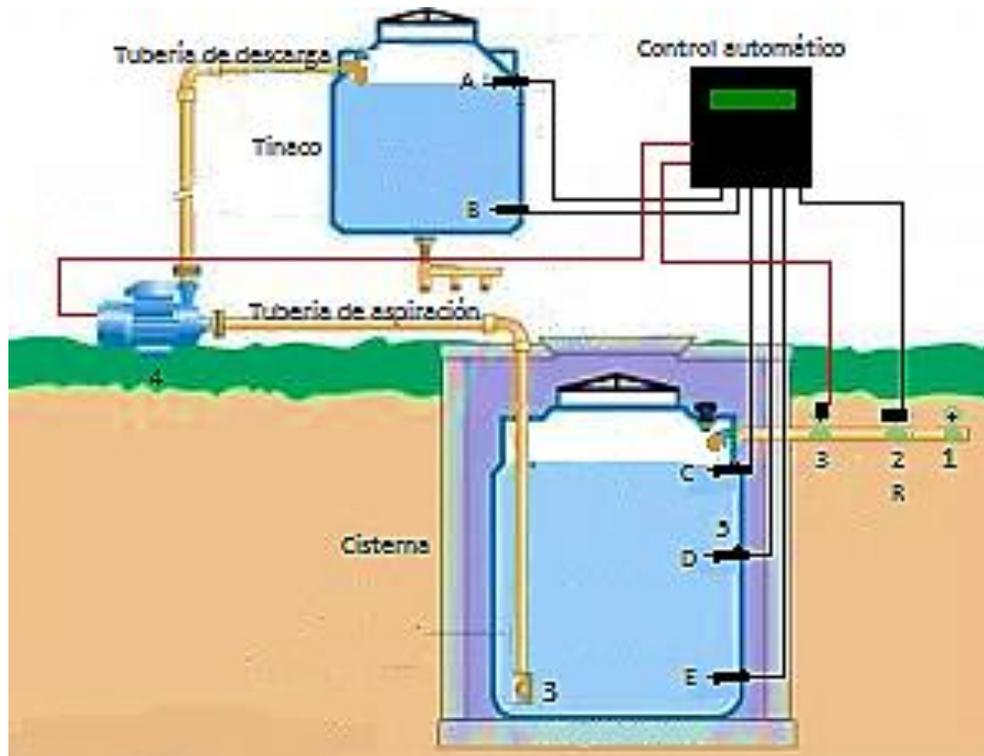


Figura 2. Sistema de trasvase de agua Automatizado.

- 1) Válvula de compuerta principal que permite abrir o cerrar el paso del agua.
- 2) Interruptor de flujo que detectará la presencia del flujo de agua.
- 3) Electroválvula que abrirá o cerrará el paso del agua hacia la cisterna.
- 4) Bomba centrífuga que succiona agua de la cisterna, para depositarla en el tinaco.
- 5) Interruptores de nivel que censarán el nivel del agua.

MEDICIÓN DE FLUIDOS

Hace mucho tiempo, los medios para medir el nivel de líquidos en recipientes, no eran tan precisos como en la actualidad, se tenía que realizar la medición con herramientas mecánicas que podían realizar su trabajo, pero se requería diseñar sistemas más complejos y costosos. Hoy en día disponemos de herramientas electrónicas eficientes para esta tarea.

MEDICIÓN MECÁNICA

Medidor por sonda.

Consiste en una varilla o regla graduada, de la longitud conveniente para introducirla dentro del depósito. La determinación del nivel se efectúa por la lectura directa de la longitud mojada por el líquido.

Medidor por cinta y plomada.

Consta de una cinta graduada y un plomo en la punta. Se emplea cuando es difícil que la varilla tenga acceso al fondo del tanque. Se usa midiendo la distancia desde la superficie del líquido hasta la parte superior del tanque, obteniendo el nivel por diferencia.

Medidor por flotador.

Consiste en un flotador ubicado en la superficie del líquido y conectado al exterior del tanque, indicando directamente el nivel sobre una escala graduada. Es el modelo más antiguo y el más usado en tanques de capacidad grande.

MEDICIÓN ELECTRÓNICA

Medición por flotador magnético.

Consiste en un flotador que acciona un micro interruptor alojado en su base, cuando el nivel del líquido hace desplazar al flotador, llega un momento en que el imán contenido en el flotador hace cerrar el micro interruptor produciéndose la conmutación, logrando activar ya sea relevadores u otro dispositivo de control.

Medidor conductivo.

Consiste en uno o varios electrodos y un relevador eléctrico o electrónico que es excitado cuando el líquido moja a dichos electrodos. El líquido debe ser lo suficientemente conductor como para excitar el circuito electrónico. Cuando el líquido moja los electrodos se cierra el circuito electrónico y circula una corriente segura. El líquido contenido en el tanque debe tener un mínimo de conductividad y si su naturaleza lo exige, la corriente debe ser baja para evitar el deterioro del producto.

Medidor capacitivo.

Mide la capacidad del condensador formado por el electrodo sumergido en el líquido y las paredes del tanque. La capacidad del conjunto depende linealmente del nivel del líquido. En fluidos conductores el electrodo está aislado usualmente con teflón interviniendo las capacidades adicionales entre el material aislante y el electrodo en la zona del líquido.

Medidor por ultrasonido.

Se basa en la emisión de un impulso ultrasónico a una superficie reflectante y la recepción del eco del mismo en un receptor. El retardo en la captación del eco depende del nivel del tanque. La medición se hace desde el exterior del tanque. Los sensores trabajan a frecuencias cercanas a 20 KHz. Estas ondas atraviesan el medio ambiente de gases o vapores con cierto amortiguamiento y se reflejan en la superficie del líquido.

Medidor por vibración.

Dispone de un diapasón que queda introducido en el recipiente, este queda vibrando a una frecuencia fija. Al contacto con el fluido, altera la vibración del diapasón y este cambio es detectado por el circuito de control.

CAPITULO 1

CONDICIONES DE OPERACIÓN DEL PROYECTO

FUNCIONAMIENTO DEL CONTROL AUTOMÁTICO

Los interruptores de nivel con flotador son conectados al módulo de control lo cual estos nos censarán el nivel de agua que se encuentra en la cisterna y/o el tinaco, abriendo o cerrando sus contactos, es decir el nivel de agua que se tiene en estos. También es conectado un interruptor de flujo que censará si hay o no flujo de agua en la toma principal con el propósito de que el módulo de control haga uso de la reserva de agua cuando no se tenga abastecimiento en la toma principal. El módulo de control automático procesará esta información tomando las decisiones para las que es diseñado, activando o desactivando la bomba centrífuga, la electroválvula o en su defecto la alarma audible que nos avisará de algún fallo en el sistema, con todo esto, el Display de LCD mostrará dicha información del proceso.

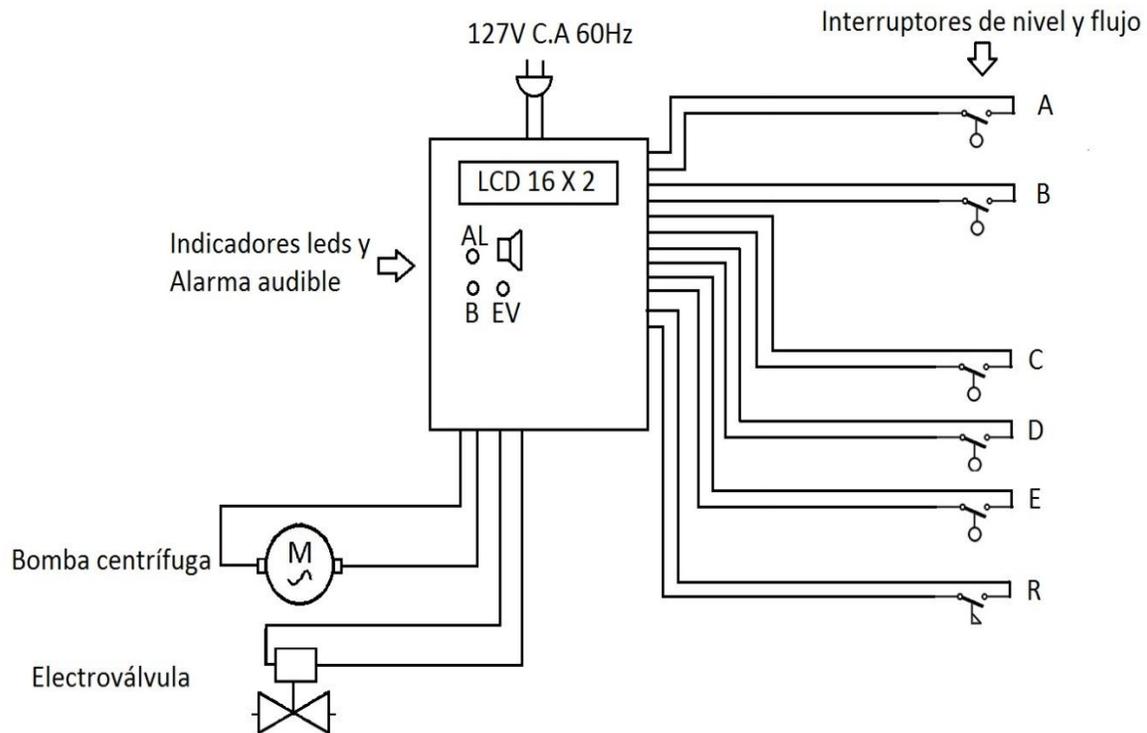


Figura 3. Diagrama de conexiones eléctricas de los dispositivos externos al módulo de control.

DIAGRAMA DE FLUJO

Para poder diseñar correctamente este control automático, se tiene que analizar como interpretará la información proveniente de los sensores para que con ello se tomen las decisiones adecuadas. Se recurre entonces a elaborar el siguiente diagrama de flujo.

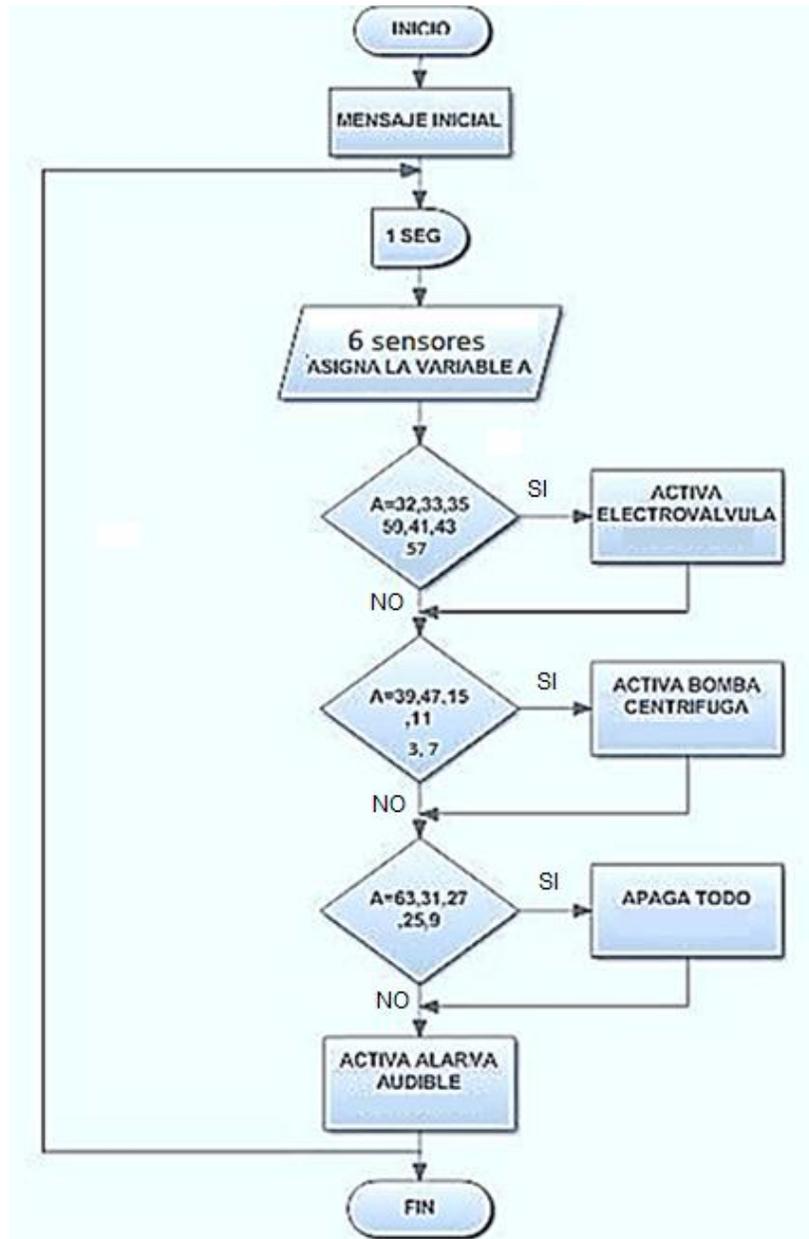


Figura 4. Diagrama de flujo del trasvase de agua.

- **DESCRIPCION DEL DIAGRAMA DE FLUJO**

Inicio del programa.

Se muestra un mensaje inicial.

Retardo de un segundo para eliminar falsas lecturas producidas por los rebotes mecánicos que se tienen cuando son accionados los interruptores, recordemos que estos al conmutar, el contacto se abre y se cierra varias veces antes de quedar en la posición cerrada final.

Se asigna al puerto C del microcontrolador como entradas de 6 bits, la combinación que se tenga en la entrada se guarda en la variable A, cabe destacar que al tener seis interruptores en la entrada, se tienen 64 posibles combinaciones, es por eso que cada combinación que se tenga, es guardada en una variable para su posterior procesamiento.

Dependiendo del valor de la variable A, se compara con las combinaciones que se tienen para el accionamiento de la electroválvula, las combinaciones en decimal para esta acción son la 32, 33, 35, 59, 41, 43, 57. Entonces para tal caso si en la entrada se encuentra con estas combinaciones, inmediatamente se activará la electroválvula, pero si no es ninguna, pasa a comparar con las siguientes que accionan la bomba que son en decimal la 39, 47, 15, 11, 3, 7. Si la combinación es alguna de estas, inmediatamente se activa la bomba pero si no, pasa a comparar las siguientes que paran a la electroválvula y la bomba que son en decimal la 63, 31, 27, 25, 9. Espera a un cambio de estado de cualquier interruptor, Pero en caso de que tampoco sean, el programa interpreta al resto de combinaciones como falla, activando la alarma audible, en este momento el usuario tendrá que revisar el sistema para corregir la falla que se tenga. Después de esto el programa regresa al inicio del bucle y vuelve a repetir las comparaciones.

Por eso es muy importante que se respete el orden en los que se instalan los interruptores y también es fundamental analizar el proceso normal para el trasvase de agua cisterna – tinaco.

TABLAS DESCRIPTIVAS DE LA OPERACIÓN

A continuación se muestran y se describen las tres tablas que son parte del código binario donde se realizó el análisis del proceso normal del trasvase de agua.

- **EL PROCESO DE LLENADO**

Al inicio del proceso de llenado, la electroválvula (EV) ingresa agua a la cisterna ya que los interruptores de nivel (A, B, C, D, E) están desactivados y el interruptor de flujo (R) detecta la presencia de agua en la toma principal.

Al momento en que se va llenando la cisterna se activa el interruptor de nivel E y sigue en funcionamiento la electroválvula (EV) hasta que se active el interruptor de nivel D, sigue en funcionamiento la electroválvula (EV) hasta que se active el interruptor de nivel C que indica que la cisterna está llena, en este momento la electroválvula se desactiva. Como los interruptores de nivel del tinaco A y B están desactivados que quiere decir que no hay agua en el tinaco, la bomba centrífuga (B) se activa para subir agua de la cisterna al tinaco, una vez que los interruptores de nivel A y B están activos, la bomba deja de subir agua, pero al mismo tiempo el interruptor de nivel C se desactiva por el trasvase al tinaco, la electroválvula (EV) se vuelve a activar para completar el llenado de la cisterna, una vez lleno tanto el tinaco como la cisterna se desactivan los dispositivos.

Cuando el usuario consuma el agua del tinaco, se va a estar repitiendo la combinación 47, 59 y 63 hasta que el interruptor de flujo (R) no detecte agua en la toma principal.

PUERTO C							PUERTO A			PUERTO B
pin	RC5	RC4	RC3	RC2	RC1	RC0	RA2	RA1	RA0	
decimal	R	A	B	C	D	E	EV	B	AL	Mensajes en Display LCD
32	1	0	0	0	0	0	1	0	1	INGRESO DE AGUA A LA CISTERNA
33	1	0	0	0	0	1	1	0	0	NIVEL MINIMO EN LA CISTERNA
35	1	0	0	0	1	1	1	0	0	INGRESO DE AGUA A LA CISTERNA
39	1	0	0	1	1	1	0	1	0	CISTERNA LLENA LLENANDO TINACO
47	1	0	1	1	1	1	0	1	0	NIVEL MINIMO EN EL TINACO
59	1	1	1	0	1	1	1	0	0	TINACO LLENO INGRESANDO AGUA
63	1	1	1	1	1	1	0	0	0	CISTERNA LLENA TINACO LLENO

Figura 5. Tabla descriptiva del proceso de llenado.

• **EL PROCESO DE USO DE LA RESERVA DE AGUA**

Inicia cuando el interruptor de flujo (R) no detecta agua en la toma principal, inmediatamente el control automático comprueba en qué estado se encuentran los interruptores de nivel (A, B, C, D y E) entonces se tiene lo siguiente.

Cuando todos los interruptores de nivel (A, B, C, D y E) se encuentren activos, la electroválvula (EV), la bomba centrífuga (B) y la alarma audible (AL) deberán estar desactivados. Cuando el usuario consume agua del tinaco, la bomba centrífuga (B) se activa haciendo uso de la reserva de agua que se tiene en la cisterna. En este momento se desactiva el interruptor de nivel C pero el interruptor D y E siguen activos por la reserva que aún no se consume.

Al consumir el agua del tinaco el interruptor de nivel A se desactiva haciendo que la bomba centrífuga (B) se active nuevamente para hacer uso de la reserva de agua completando el llenado, ahora el interruptor de nivel D se encuentra desactivado lo que significa que la cisterna se encuentra en el nivel mínimo de agua que ya no se puede consumir. Al consumir el agua del tinaco dejando el interruptor de nivel A desactivado lo que significa que se hizo uso total de la reserva de agua, el control automático se queda esperando a que el interruptor de flujo (R) se active nuevamente para hacer funcionar la electroválvula (EV) y así hasta que el interruptor de nivel C se active, por lo que significa que la cisterna está llena. El control automático comprobará el estado de los interruptores de nivel para regresar al proceso de llenado y continuar con el ciclo normal del trasvase de agua

		PUERTO C						PUERTO A			PUERTO B		
binario	R	A	B	C	D	E	EV	B	AL	Mensajes en Display LCD			
31	0	1	1	1	1	1	0	0	0	SIN SUMINISTRO DE AGUA / CISTERNA LLENA TINACO LLENO			
15	0	0	1	1	1	1	0	1	0	SIN SUMINISTRO DE AGUA / LLENANDO TINACO RESERVA COMPLETA			
27	0	1	1	0	1	1	0	0	0	SIN SUMINISTRO DE AGUA / TINACO LLENO RESERVA COMPLETA			
11	0	0	1	0	1	1	0	1	0	SIN SUMINISTRO DE AGUA / USO DE LA RESERVA DE AGUA			
25	0	1	1	0	0	1	0	0	0	SIN SUMINISTRO DE AGUA / TINACO LLENO RESERVA AGOTADA			
9	0	0	1	0	0	1	0	0	0	SIN SUMINISTRO DE AGUA / TINACO MÍNIMO CISTERNA VACIA			
41	1	0	1	0	0	1	1	0	0	INGRESO DE AGUA A LA CISTERNA			
43	1	0	1	0	1	1	1	0	0	INGRESO DE AGUA A LA CISTERNA			

Figura 6. Tabla descriptiva del proceso de uso de la reserva de agua.

• LAS FALLAS

En caso de que alguna combinación no se encuentre dentro de los procesos de trasvase, el control automático clasificara estas combinaciones como erróneas mandando a falla.

decimal	PUERTO C						PUERTO A			Mensaje en Display LCD
	RC3	RC4	RC5	RC2	RC1	RC0	RA2	RA1	RA0	
0	0	0	0	0	0	0	0	0	1	TRASVASE DE AGUA SIN RECU RSOS...
1	0	0	0	0	0	1	0	0	1	TRASVASE DE AGUA SIN RECU RSOS...
2	0	0	0	0	1	0	0	0	1	SE PRODUJO UNA FALLA REVISE
3	0	0	0	0	1	1	0	1	0	SIN SUMINISTRO DE AGUA / USO DE LA RESERVA DE AGUA
4	0	0	0	1	0	0	0	0	1	SE PRODUJO UNA FALLA REVISE
5	0	0	0	1	0	1	0	0	1	SE PRODUJO UNA FALLA REVISE
6	0	0	0	1	1	0	0	0	1	SE PRODUJO UNA FALLA REVISE
7	0	0	0	1	1	1	0	1	0	SIN SUMINISTRO DE AGUA / LLENANDO TINACO RESERVA COMPLETA
8	0	0	1	0	0	0	0	0	1	TRASVASE DE AGUA SIN RECU RSOS...
9	0	0	1	0	0	1	0	0	0	SIN SUMINISTRO DE AGUA / TINACO VACIO CISTERNA VACIA
10	0	0	1	0	1	0	0	0	1	SE PRODUJO UNA FALLA REVISE
11	0	0	1	0	1	1	0	1	0	SIN SUMINISTRO DE AGUA / USO DE LA RESERVA DE AGUA
12	0	0	1	1	0	0	0	0	1	SE PRODUJO UNA FALLA REVISE
13	0	0	1	1	0	1	0	0	1	SE PRODUJO UNA FALLA REVISE
14	0	0	1	1	1	0	0	0	1	SE PRODUJO UNA FALLA REVISE
15	0	0	1	1	1	1	0	1	0	SIN SUMINISTRO DE AGUA / LLENANDO TINACO RESERVA COMPLETA
16	0	1	0	0	0	0	0	0	1	SE PRODUJO UNA FALLA REVISE
17	0	1	0	0	0	1	0	0	1	SE PRODUJO UNA FALLA REVISE
18	0	1	0	0	1	0	0	0	1	SE PRODUJO UNA FALLA REVISE
19	0	1	0	0	1	1	0	0	1	SE PRODUJO UNA FALLA REVISE
20	0	1	0	1	0	0	0	0	1	SE PRODUJO UNA FALLA REVISE
21	0	1	0	1	0	1	0	0	1	SE PRODUJO UNA FALLA REVISE
22	0	1	0	1	1	0	0	0	1	SE PRODUJO UNA FALLA REVISE
23	0	1	0	1	1	1	0	0	1	SE PRODUJO UNA FALLA REVISE
24	0	1	1	0	0	0	0	0	1	TRASVASE DE AGUA SIN RECU RSOS...
25	0	1	1	0	0	1	0	0	0	SIN SUMINISTRO DE AGUA / TINACO LLENO RESERVA AGOTADA
26	0	1	1	0	1	0	0	0	1	SE PRODUJO UNA FALLA REVISE
27	0	1	1	0	1	1	0	0	0	SIN SUMINISTRO DE AGUA / TINACO LLENO RESERVA COMPLETA
28	0	1	1	1	0	0	0	0	1	SE PRODUJO UNA FALLA REVISE
29	0	1	1	1	0	1	0	0	1	SE PRODUJO UNA FALLA REVISE
30	0	1	1	1	1	0	0	0	1	SE PRODUJO UNA FALLA REVISE
31	0	1	1	1	1	1	0	0	0	SIN SUMINISTRO DE AGUA / CISTERNA LLENA TINACO LLENO
32	1	0	0	0	0	0	1	0	1	INGRESO DE AGUA A LA CISTERNA
33	1	0	0	0	0	1	1	0	0	NIVEL MINIMO EN LA CISTERNA
34	1	0	0	0	1	0	0	0	1	SE PRODUJO UNA FALLA REVISE
35	1	0	0	0	1	1	1	0	0	INGRESO DE AGUA A LA CISTERNA
36	1	0	0	1	0	0	0	0	1	SE PRODUJO UNA FALLA REVISE
37	1	0	0	1	0	1	0	0	1	SE PRODUJO UNA FALLA REVISE
38	1	0	0	1	1	0	0	0	1	SE PRODUJO UNA FALLA REVISE
39	1	0	0	1	1	1	0	1	0	CISTERNA LLENA LLENA TINACO
40	1	0	1	0	0	0	0	0	1	SE PRODUJO UNA FALLA REVISE
41	1	0	1	0	0	1	1	0	0	INGRESO DE AGUA A LA CISTERNA
42	1	0	1	0	1	0	0	0	1	SE PRODUJO UNA FALLA REVISE
43	1	0	1	0	1	1	1	0	0	INGRESO DE AGUA A LA CISTERNA
44	1	0	1	1	0	0	0	0	1	SE PRODUJO UNA FALLA REVISE
45	1	0	1	1	0	1	0	0	1	SE PRODUJO UNA FALLA REVISE
46	1	0	1	1	1	0	0	0	1	SE PRODUJO UNA FALLA REVISE
47	1	0	1	1	1	1	0	1	0	NIVEL MINIMO EN EL TINACO
48	1	1	0	0	0	0	0	0	1	SE PRODUJO UNA FALLA REVISE
49	1	1	0	0	0	1	0	0	1	SE PRODUJO UNA FALLA REVISE
50	1	1	0	0	1	0	0	0	1	SE PRODUJO UNA FALLA REVISE
51	1	1	0	0	1	1	0	0	1	SE PRODUJO UNA FALLA REVISE
52	1	1	0	1	0	0	0	0	1	SE PRODUJO UNA FALLA REVISE
53	1	1	0	1	0	1	0	0	1	SE PRODUJO UNA FALLA REVISE
54	1	1	0	1	1	0	0	0	1	SE PRODUJO UNA FALLA REVISE
55	1	1	0	1	1	1	0	0	1	SE PRODUJO UNA FALLA REVISE
56	1	1	1	0	0	0	0	0	1	SE PRODUJO UNA FALLA REVISE
57	1	1	1	0	0	1	1	0	0	TINACO LLENO INGRESANDO AGUA
58	1	1	1	0	1	0	0	0	1	SE PRODUJO UNA FALLA REVISE
59	1	1	1	0	1	1	1	0	0	TINACO LLENO INGRESANDO AGUA
60	1	1	1	1	0	0	0	0	1	SE PRODUJO UNA FALLA REVISE
61	1	1	1	1	0	1	0	0	1	SE PRODUJO UNA FALLA REVISE
62	1	1	1	1	1	0	0	0	1	SE PRODUJO UNA FALLA REVISE
63	1	1	1	1	1	1	0	0	0	CISTERNA LLENA TINACO LLENO

Figura 7. Tabla descriptiva de las fallas en el trasvase de agua.

CAPITULO 2

EL HARDWARE

BLOQUES QUE CONFORMAN EL HARDWARE DEL CONTROL AUTOMÁTICO

En la figura 8. Se muestran los bloques del hardware del módulo de control automático, cabe destacar que los bloques que están fuera de la línea punteada se refieren a los elementos externos conectados al módulo de control automático y lo que está dentro a los elementos internos del módulo de control.

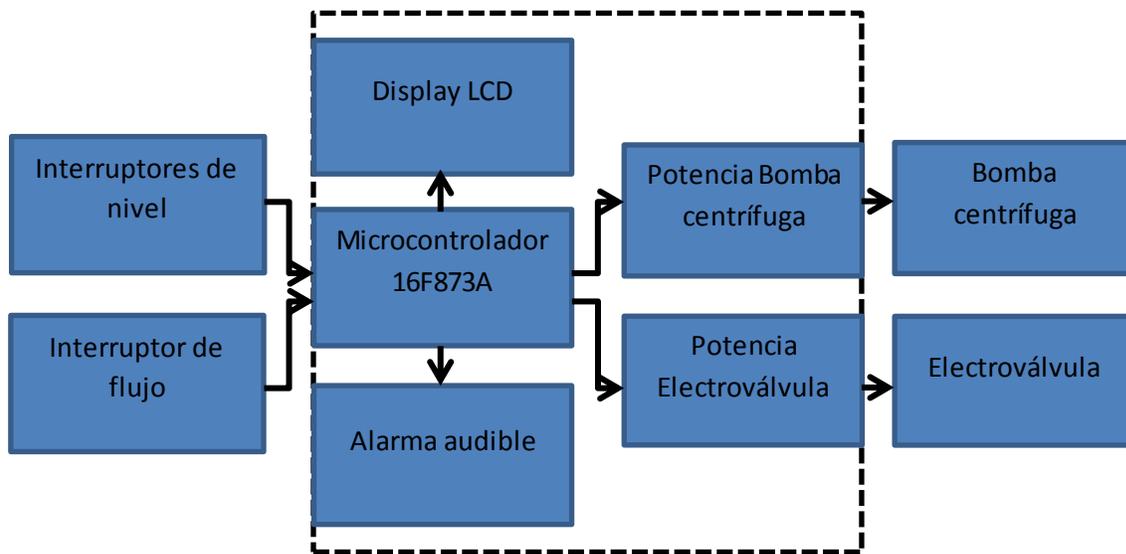


Figura 8. Bloques que componen el hardware del módulo de control automático.

Elementos externos

1. Interruptores de nivel
2. Interruptor de flujo
3. Bomba centrífuga
4. Electroválvula

Elementos internos

5. Microcontrolador 16F873A
6. Display LCD de 16 caracteres por 2 líneas
7. Alarma audible
8. Etapa de potencia de la bomba centrífuga
9. Etapa de potencia de la electroválvula

ELEMENTOS EXTERNOS CONECTADOS AL MÓDULO DE CONTROL

- **LOS INTERRUPTORES DE NIVEL CON FLOTADOR MAGNÉTICO**

Un interruptor de nivel es un dispositivo que logra abrir o cerrar un interruptor magnético contenido en su cuerpo, funciona mediante un flotador que oculta en su base un imán.

Cuando el nivel del líquido aumenta, el imán de la base del flotador se aproxima al interruptor magnético lo cual se atraen por acción del campo magnético entre ellos, haciendo que los contactos del interruptor se cierren. Por el contrario si el líquido desciende, hará que el imán de la base del flotador se separe del interruptor magnético haciendo que se abra sus contactos.



Figura 9. Interruptor de nivel con flotador magnético horizontal.

Dependiendo de la posición del interruptor pueden ser NA (normalmente abiertos) o NC (normalmente cerrados) y también pueden ser verticales u horizontales.



Figura 10. Interruptor de nivel con flotador magnético vertical.

A continuación se muestra la tabla de características de los interruptores elegidos, así como sus dimensiones.

CARACTERÍSTICAS	
MARCA	CEBEK
MODELO	C7235 / C7236
MATERIAL	POLIPROPILENO
CONEXION ELECTRICA	POR CABLE
TIPO DE CONTACTO	NA / NC
TENSION DE CONMUTACIÓN	200V C.D MÁX.
CORRIENTE DE CONMUTACIÓN	0.5A MÁX.
TEMPERATURA	-20°C a 90°C
MONTAJE	HORIZONTAL / VERTICAL

Figura 11. Tabla de características del interruptor de nivel.

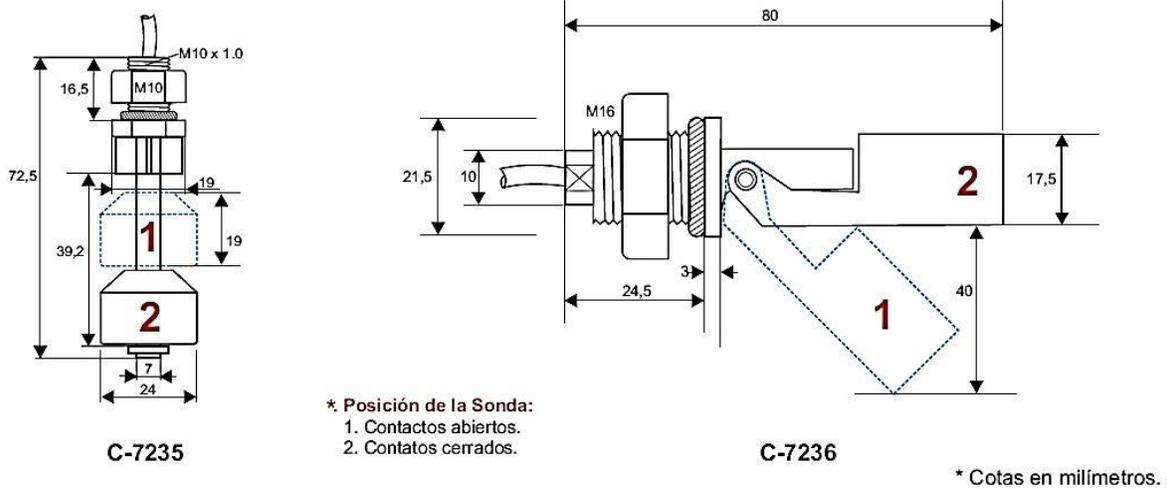


Figura 12. Dimensiones de los interruptores de nivel.

- **EL INTERRUPTOR DE FLUJO TIPO PALETA**

El interruptor de flujo es un dispositivo que es instalado en una tubería y permite detectar si hay o no circulación de líquido, pero no miden el caudal.

Está constituido por una paleta con un imán en su base y de un interruptor magnético. Cuando no existe flujo de líquido en la tubería, el imán de la base de la paleta, se encuentra separada del interruptor abriendo así sus contactos, pero al existir flujo, el imán es desplazado hacia arriba haciendo contacto con el interruptor cerrando sus contactos.

Estos interruptores pueden ser calibrados para ajustar la resistencia al flujo de líquido según se requiera, mediante un tornillo situado en la parte posterior del interruptor.



Figura 13. Interruptor de flujo tipo paleta.

Se elige el interruptor de flujo con las siguientes características.

CARACTERÍSTICAS	
MARCA	CEBEK
MODELO	SF140
CONEXION	ROSCAA 3/4" a 1/2" o 3/4" a 1"
SALIDA	RELÉ SPDT 3A / 250V C.A
TEMPERATURA	-10°C a 110°C
PRESIÓN MÁXIMA	25 bar
REGULACION DE CAUDAL	MEDIANTE TORNILLO

Figura 14. Tabla de características del interruptor de flujo.

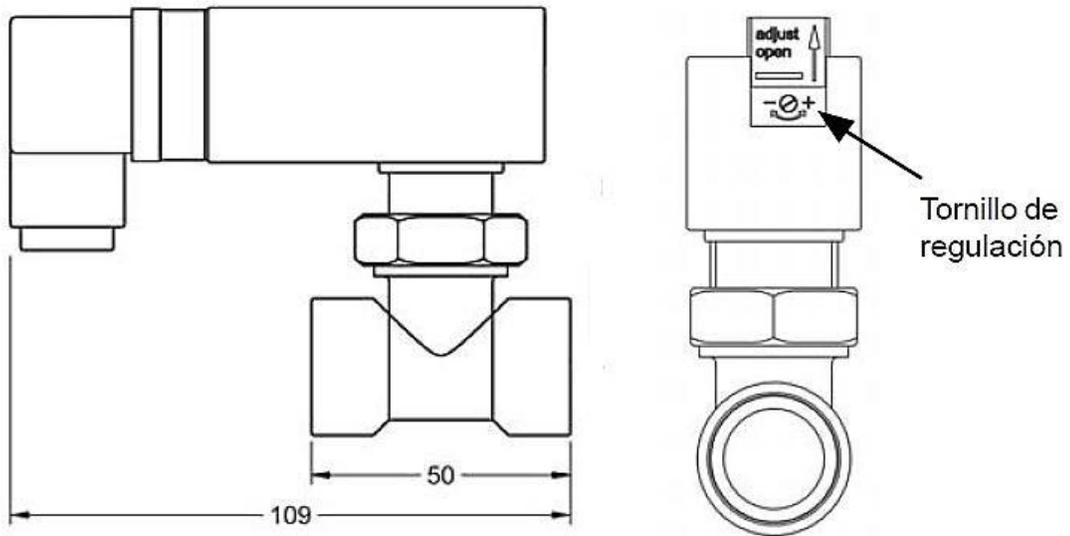


Figura 15. Tornillo de regulación del interruptor de flujo.

- **LA BOMBA CENTRÍFUGA CON ARRANQUE POR CAPACITOR**

La función de la bomba centrífuga es la de mover un cierto volumen de líquido de un punto a otro.

Se constituyen por una brida de succión y un rodete que está unido al eje del rotor, el líquido es succionado mediante el tubo de succión, hasta la entrada del rodete, donde el líquido es sometido a un movimiento de rotación muy rápido, proyectándose éste al exterior por la fuerza centrífuga, ingresando así a la voluta a una velocidad extrema.

La voluta es una parte fija que está colocada alrededor del rodete en forma de caracol, al ingresar el líquido a la voluta, este es impulsado por la brida de impulsión, haciendo circular por la tubería de impulsión líquido a una presión y velocidad creadas por la bomba.

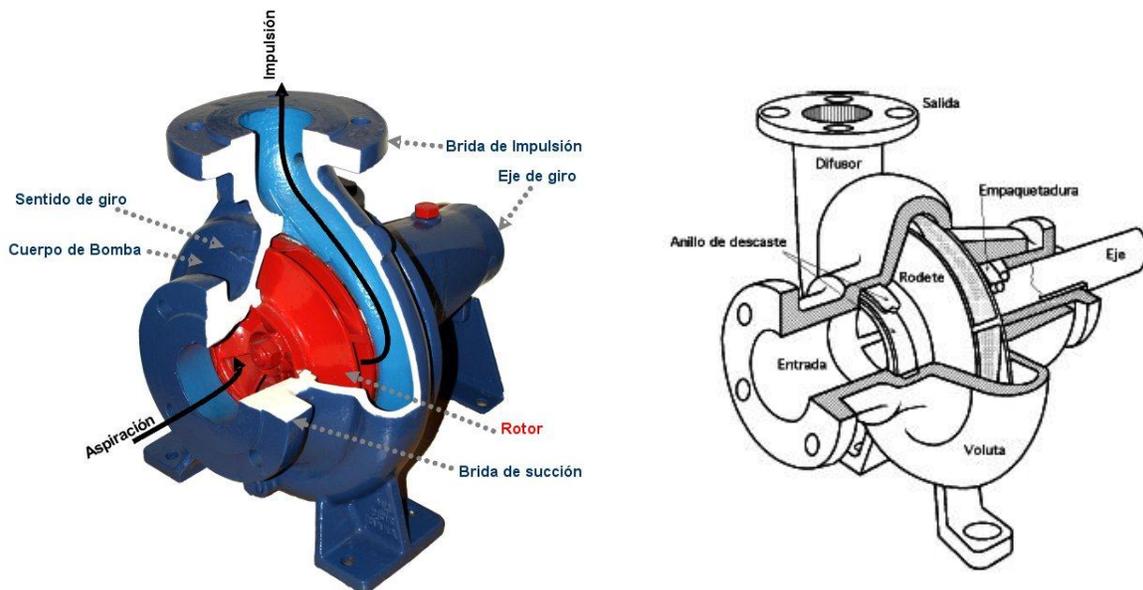


Figura 16. Descripción de componentes de la bomba.

Son motores monofásicos de corriente alterna que están constituidos por un devanado de trabajo y uno de arranque, en este último está conectado un capacitor en serie que al liberar la corriente de arranque le permitirá un mayor par de arranque al motor, también está conectado en serie un interruptor centrífugo que desconectará el devanado de arranque cuando se halla arrancado la bomba.

La corriente en el devanado de arranque que es liberada por el capacitor, proporciona un incremento en el par de arranque, teniendo un incremento en la corriente de arranque con respecto a la corriente nominal que induce el devanado de trabajo, una vez arrancada la bomba, el interruptor centrífugo desconecta el devanado de arranque, dejando solo funcionando al devanado de trabajo.

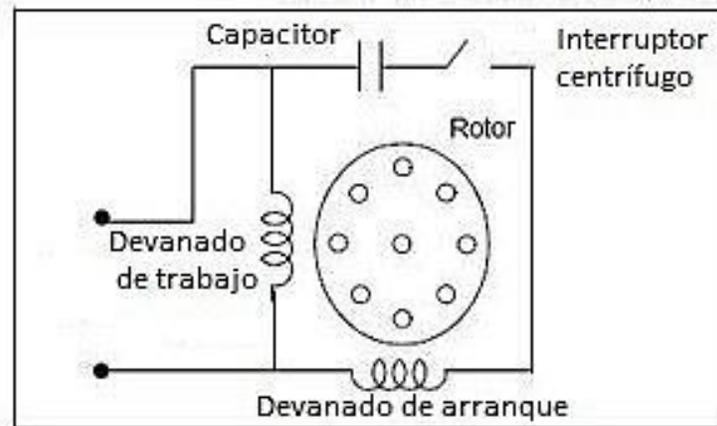


Figura 17. Diagrama eléctrico de los devanados del motor.

Para el diseño del control automático es muy importante conocer la corriente nominal y la corriente de arranque máxima, así como la tensión de la bomba.

Se considera que en un motor de arranque por capacitor, demanda al momento de arrancar máximo una corriente de 3 veces la corriente nominal aunque puede ser menos dependiendo de la bomba.



Figura 18. Bomba centrífuga SIEMENS.

La bomba usada en el sistema hidráulico que se pretende controlar es de 0.5 HP. Suficiente para su uso en una casa. A continuación se muestran sus características.

CARACTERISTICAS	
MARCA	SIEMENS
MODELO	2AN4 254-2YC35
TIPO	MONOFÁSICO DE C.A
POTENCIA	0.5 HP
FRECUENCIA	60 HZ
TENSION	127V
CORRIENTE NOMINAL	8A
VELOCIDAD	3540 rpm
PROTECCIÓN	TÉRMICA
DIAMETRO DE SUCCION	1" ¼
DIAMETRO DE IMPULSION	1"

Figura 19. Tabla de características de la bomba centrífuga.

- **LA ELECTROVÁLVULA**

La electroválvula o válvula con solenoide, es un dispositivo operado eléctricamente y es utilizado para controlar el flujo de líquidos, no regula el flujo aunque puede estar completamente abierta o cerrada.



Figura 20. Electroválvula ORBIT.

La electroválvula es una válvula que se cierra por la acción de un resorte y es abierta por el movimiento de un émbolo operado por la acción magnética de una bobina energizada eléctricamente.

Está integrada por dos partes, un solenoide y el cuerpo de la válvula.

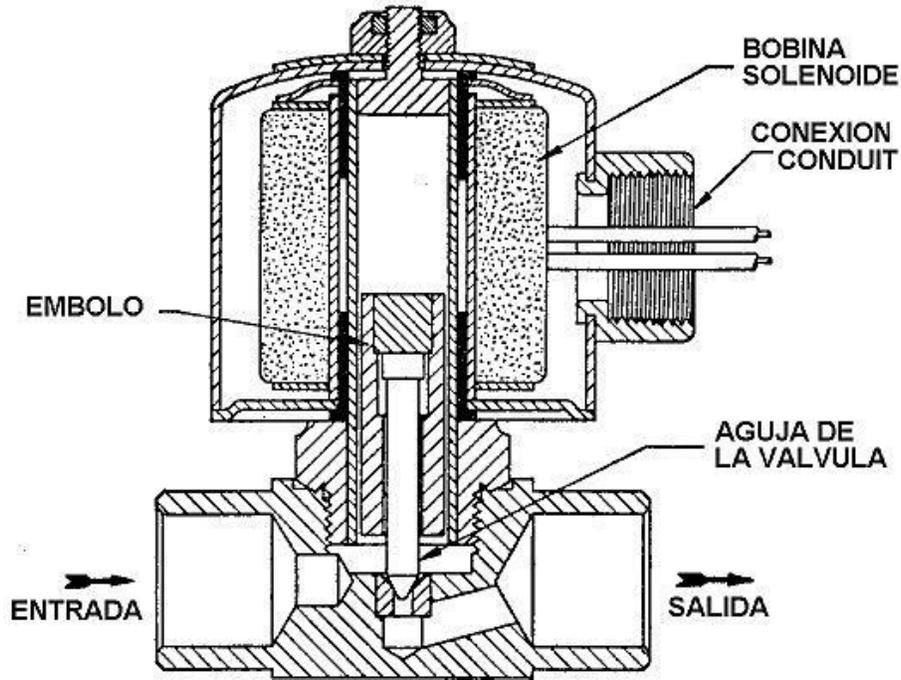


Figura 21. Vista transversal de la electroválvula.

El solenoide actúa como un electro-imán que consiste en una bobina de alambre, el cual está enrollado en espiral alrededor de un carrete o núcleo cilíndrico, cuando se envía corriente eléctrica a través de sus devanados, se crea un campo magnético. Dentro del carrete va un émbolo móvil de acero magnético el cual es desplazado al energizarse la bobina.

Se elige una electroválvula de la marca ORBIT con las siguientes características.

CARACTERÍSTICAS	
MARCA	ORBIT
MODELO	94150
CONEXION	ROSCADO DE 1"
PRESION DE TRABAJO	0.7 a 10 atm
TENSION DEL SOLENOIDE	24V C.A
FRECUENCIA	60HZ
CORRIENTE DEL SOLENOIDE	300 mA

Figura 22. Tabla de características de la electroválvula.

ELEMENTOS INTERNOS QUE CONFORMAN EL MÓDULO DE CONTROL

EL MICROCONTROLADOR PIC16F873A COMO ELEMENTO PRINCIPAL

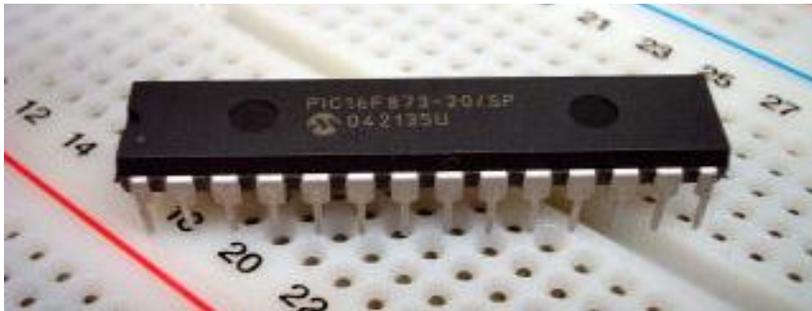


Figura 23. Microcontrolador PIC16F873A

El microcontrolador PIC16F873A es un circuito integrado programable, que contiene en su interior distintos módulos que hacen que este sea versátil para muchas aplicaciones. En la siguiente figura se muestra el interior del microcontrolador.

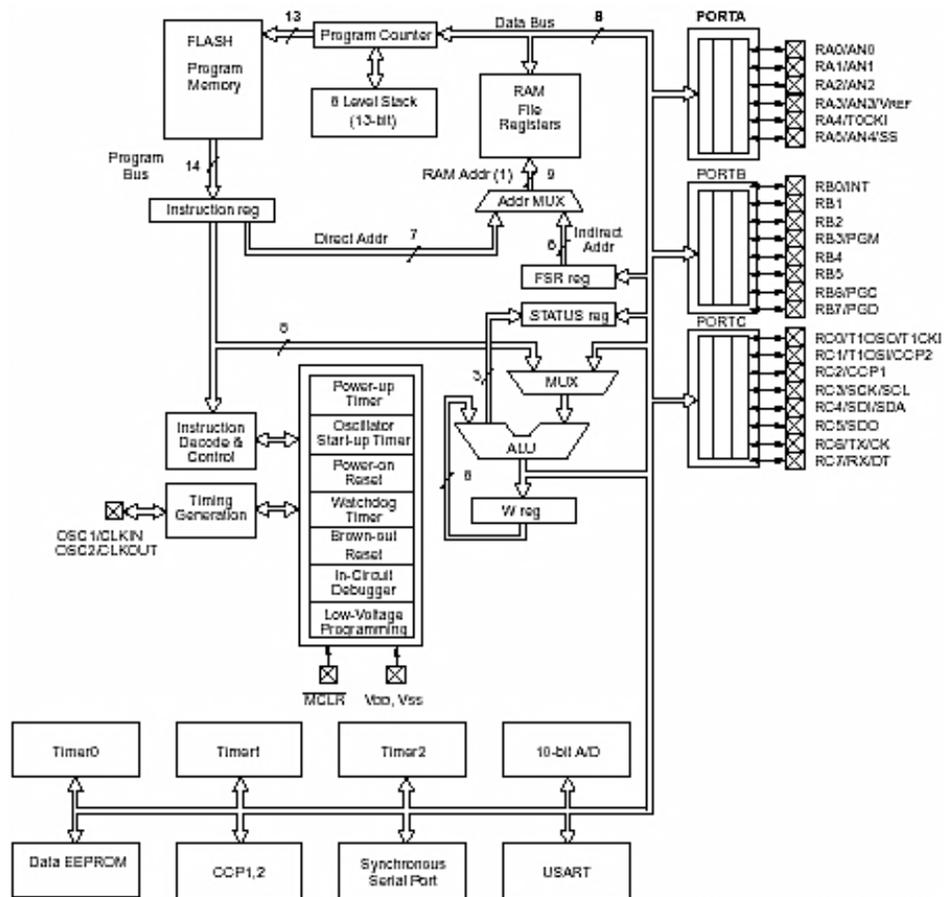


Figura 24. Módulos internos del microcontrolador.

Como se observa en el interior del microcontrolador, consta de un procesador con una unidad aritmética lógica (ALU), un decodificador de instrucciones, una memoria de programa tipo Flash de 4096 instrucciones de 14 bits, una memoria de datos tipo RAM de 192 bytes (8 bits por registro), una memoria EEPROM de 128 bytes para almacenar datos no volátiles, tres puertos Bidireccionales de Entrada / Salida, módulos especiales como temporizadores, un convertidor Analógico / Digital, módulo de comunicación serie, entre otros.

- **LA ARQUITECTURA HARVARD**

La arquitectura de éste microcontrolador se basa en la de tipo Harvard que consiste simplemente en el que el CPU está conectado a dos memorias a través de buses independientes en el cuál la memoria de programa contiene las instrucciones a ejecutar y la memoria de datos almacena datos temporales durante la ejecución del programa.

EL PROCESADOR (CPU)

El procesador responde a la arquitectura RISC, que se identifica porque el juego de instrucciones se reduce a 35, donde la mayoría se ejecutan en un ciclo de reloj, excepto las instrucciones de salto que necesitan dos ciclos. La ALU (Unidad Aritmética Lógica), ubicada dentro del procesador realiza las operaciones lógicas y aritméticas.

MEMORIA DE PROGRAMA

La memoria de programa es del tipo flash. La memoria flash es una memoria no volátil, de bajo consumo que se puede escribir y borrar eléctricamente. Es programable en el circuito como la EEPROM pero de mayor densidad y más rápida. El PIC 16F873 posee una memoria de 4K palabras, es decir permite hasta 4096 instrucciones de 14 bits cada una.

MEMORIA DE DATOS

Se compone de la memoria RAM para almacenar datos temporales durante la ejecución del programa y la memoria EEPROM que es capaz de guardar la información más de 40 años.

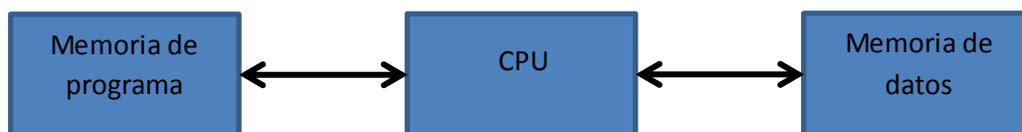


Figura 25. Bloques que componen la arquitectura Harvard.

- **DISTRIBUCIÓN Y DESCRIPCIÓN DE PINES DEL MICROCONTROLADOR 16F873A**

El microcontrolador contiene en su encapsulado 28 pines como se muestra en la figura.

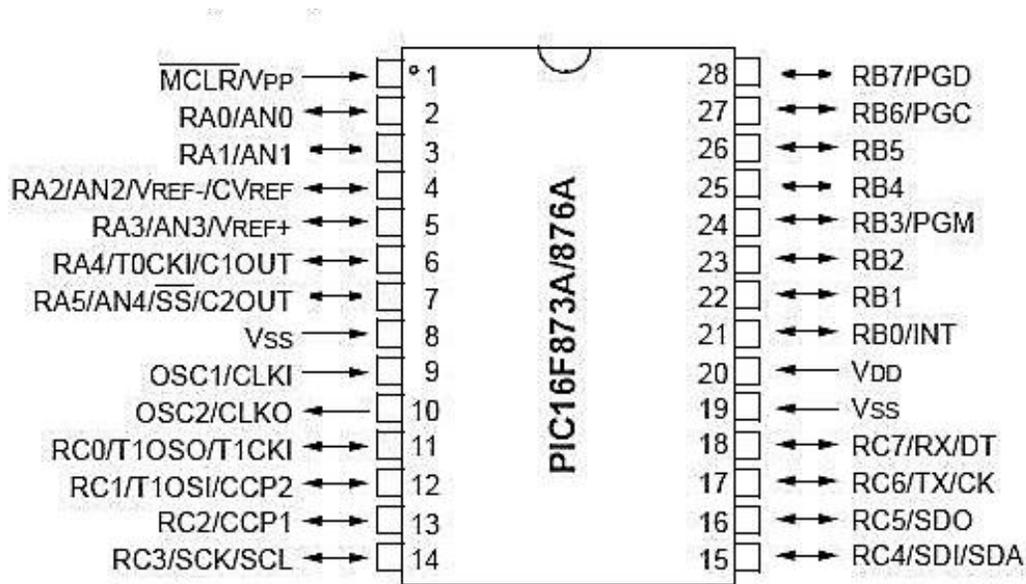


Figura 26. Pines del microcontrolador PIC16F873A.

El puerto A se compone de RA0 a RA5, donde Las terminales RA0-RA3 y RA5 son bidireccionales y manejan señales TTL. Pueden configurarse como entradas analógicas.

Se usará en nuestro diseño RA0, RA1 y RA2 como salidas.

El puerto B se compone de RB0 a RB7, son bidireccionales y manejan señales TTL. Por software se pueden activar las resistencias de pull-up internas, que evitan el uso de resistencias externas en caso de que los terminales se utilicen como entrada (esto permite, en algunos casos, reducir el número de componentes externos necesarios).

En este puerto se conectará el display LCD y la configuración de éste puerto se hace a través de software.

El puerto C se compone de RC0 a RC7, Los terminales RC0-RC7 son bidireccionales y manejan señales TTL. Se utilizan en los módulos de PWM, comparadores y transmisión serial.

En este puerto se conectarán los interruptores de nivel y de flujo y se configurarán por software como entradas digitales.

MCLR/VPP (Master Clear Reset) y su función es mantener al microcontrolador activo o inactivo, dependiendo de su estado lógico. 0= inactivo 1= activo, funciona también para resetear el circuito. También es usada para programar el microcontrolador.

VSS es el negativo de la alimentación.

VDD es el positivo de la alimentación

OSC1/CLKIN es la entrada del oscilador externo.

OSC2/CLKOUT es la salida del oscilador externo.

- **EL CIRCUITO DE RESET EN MCLR (Master Clear Reset)**

El microcontrolador se mantiene activo ya que este pin se encuentra en estado lógico alto, pero si se presiona el pulsador, este pin baja a un estado lógico bajo produciéndose un reseteo del programa cargado en el microcontrolador, desde el comienzo.

Esto es muy útil ya que si en algún momento el sistema no responde adecuadamente, con esto se logra restablecer, la resistencia Pull up de 10K conectada a VCC, mantiene al microcontrolador activo circulando una pequeña corriente aproximada de 0.45mA y cuando se pulsa el interruptor, se desvía la corriente, la resistencia de 1K protege al pin limitando la corriente. Pero si no se usa éste circuito como en nuestro caso, solo se mantiene conectado a VCC por medio de la resistencia de 10K.

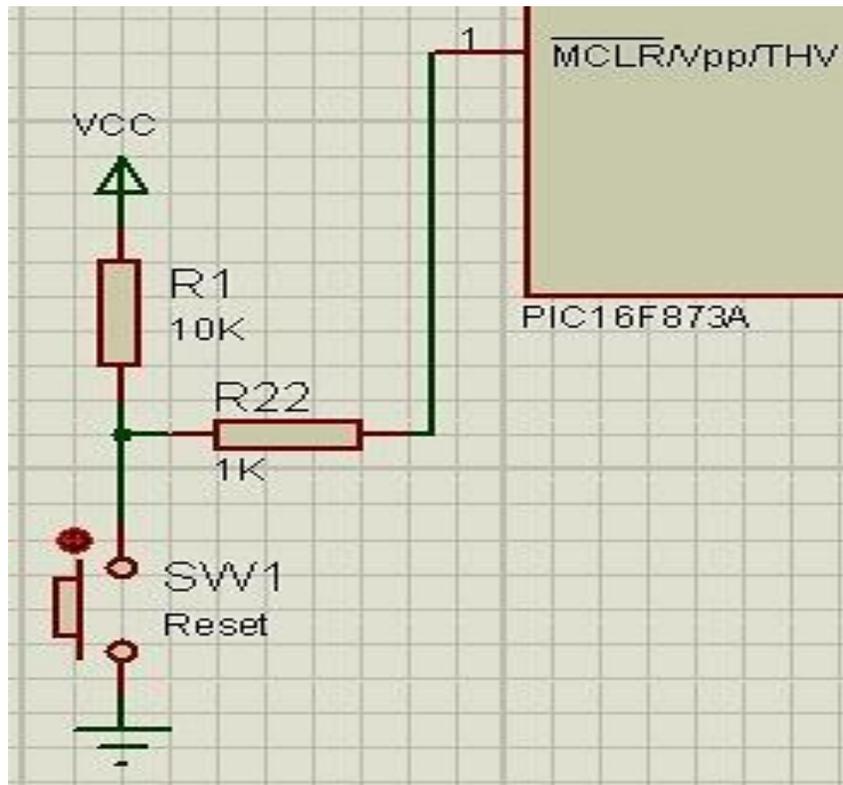


Figura 27. Circuito de reset externo en MCLR

- **EL OSCILADOR EXTERNO**

Para que el microcontrolador pueda procesar instrucciones, necesita de un oscilador cuyos pulsos de reloj determinen la velocidad de trabajo. El oscilador que genera esos pulsos, está implementado dentro del microcontrolador pero para estabilizar y regular la frecuencia de trabajo se hace uso de un oscilador externo conectado a las terminales OSC1 y OSC2.

La frecuencia de reloj principal determina el ciclo de instrucción (tiempo que tarda una instrucción en ejecutarse). En los microcontroladores, un ciclo de instrucción emplea cuatro periodos de oscilación del reloj principal. Por ejemplo si la frecuencia del reloj principal es de 4 MHz, un ciclo de instrucción tardara en realizarse:

Tiempo de oscilación del reloj principal = $1/F = 1/4\text{MHz} = 250\text{ns}$
Ciclo de instrucción = Tiempo de oscilación del reloj principal x 4 = $250\text{ns} \times 4 = 1\mu\text{s}$

Figura 28. Tiempo de cada ciclo de instrucción.

Entonces se tienen cuatro tipos de configuraciones para el oscilador externo.

Tipo RC: Es un oscilador de bajo costo y poca estabilidad. Solo precisa una resistencia y un capacitor externo.

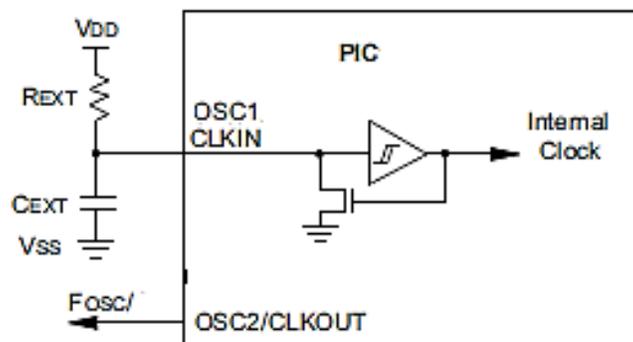


Figura 29. Oscilador en configuración RC.

Tipo HS: Es un oscilador de alta velocidad y muy estable funciona en frecuencias comprendidas entre 4MHz y 20MHz. Utiliza un cristal de cuarzo o un resonador cerámico.

Tipo XT: También emplea el cristal de cuarzo o el resonador cerámico. Trabaja en frecuencias medias, comprendidas entre 200KHz y 4MHz.

Tipo LP: Empleado en aplicaciones de bajo consumo, lo que conlleva una frecuencia pequeña. A más velocidad, mas consumo. Usa cristal o resonador, y las frecuencias de trabajo oscilan entre 32KHz y 200KHz.

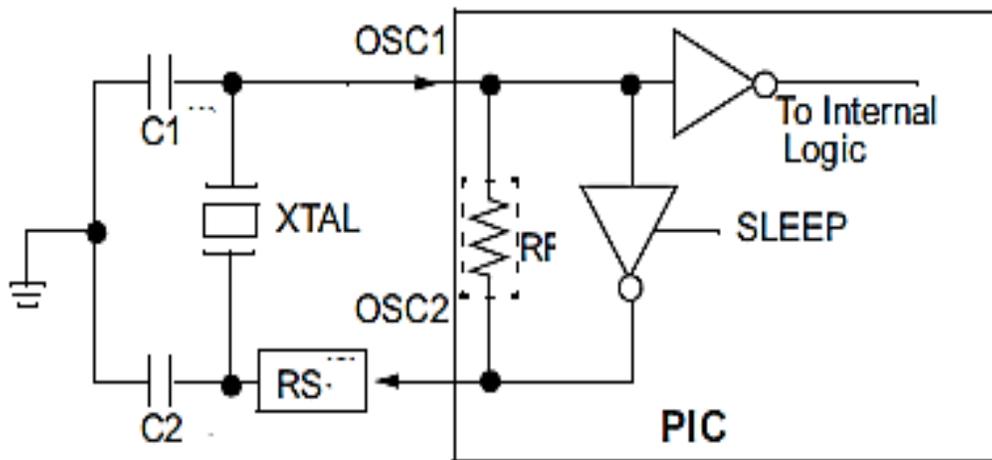


Figura 30. Oscilador en configuraciones LP, XT, HS.

Para otros modos de configuración y dependiendo de la frecuencia de trabajo que se requiera, se elige el valor de los capacitores el cual deben ser del mismo valor. A continuación se muestra la tabla de referencia.

TIPO DE OSCILADOR	FRECUENCIA DEL CRISTAL	RANGO DE VALOR	
		C1	C2
LP	32 kHz	33pF	33pF
	200 kHz	15pF	15pF
XT	200 kHz	47-68pF	47-68pF
	1MHz	15pF	15pF
	4MHz	15pF	15pF
HS	4MHz	15pF	15pF
	8MHz	15-33pF	15-33pF
	20MHz	15-33pF	15-33pF

Figura 31. Tabla de referencias de los capacitores.

En nuestro diseño, se elige un oscilador a cristal del tipo HS de 4MHz y capacitores de 15pF.

- **CARACTERÍSTICAS ELÉCTRICAS**

A continuación se muestra algunas de las características eléctricas más importantes del microcontrolador PIC16F873A.

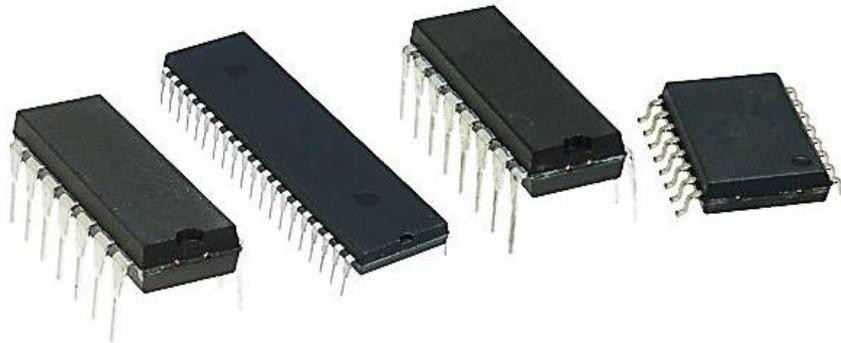


Figura 32. Microcontroladores PIC16F873A.

CARACTERISTICAS ELECTRICAS	
Temperatura ambiente	-55°C a 125°C
Temperatura almacenada	-65°C a +150°C
Voltaje en cualquier pin con respecto a VSS (Excepto VDD, MCLR)	0.3V a (VDD+0.3V)
Voltaje en VDD con respecto a VSS	-0.3V a +7.5V
Voltaje en MCLR con respecto a VSS	0V a +14V
Potencia total de disipación	1W
Máxima corriente en el pin VDD	250mA
Máxima corriente de salida por cualquier pin de E/S	25mA

Figura 33. Tabla de características eléctricas

- **LA ETAPA DE ENTRADA**

Cuando los interruptores conectados a los pines de entrada del microcontrolador se encuentran abiertos, los pines se mantienen a estado lógico bajo (0V), ya que se mandan a tierra a partir de las resistencias pull down de 10K, recordemos que como está configurado el puerto C como entradas TTL, estos pines no pueden quedarse en el aire. Cuando los interruptores se cierran, los pines cambian a un estado lógico alto (5V) y las resistencias de 1K protegen a los pines limitando la corriente aproximada de 5mA. Estos cambios de estado lógicos, el microcontrolador los interpretará, es por eso que los interruptores de nivel censarán el nivel de agua conectando o desconectando, mandando 1 y 0 respectivamente. En el esquema eléctrico estos interruptores se representan con letras.

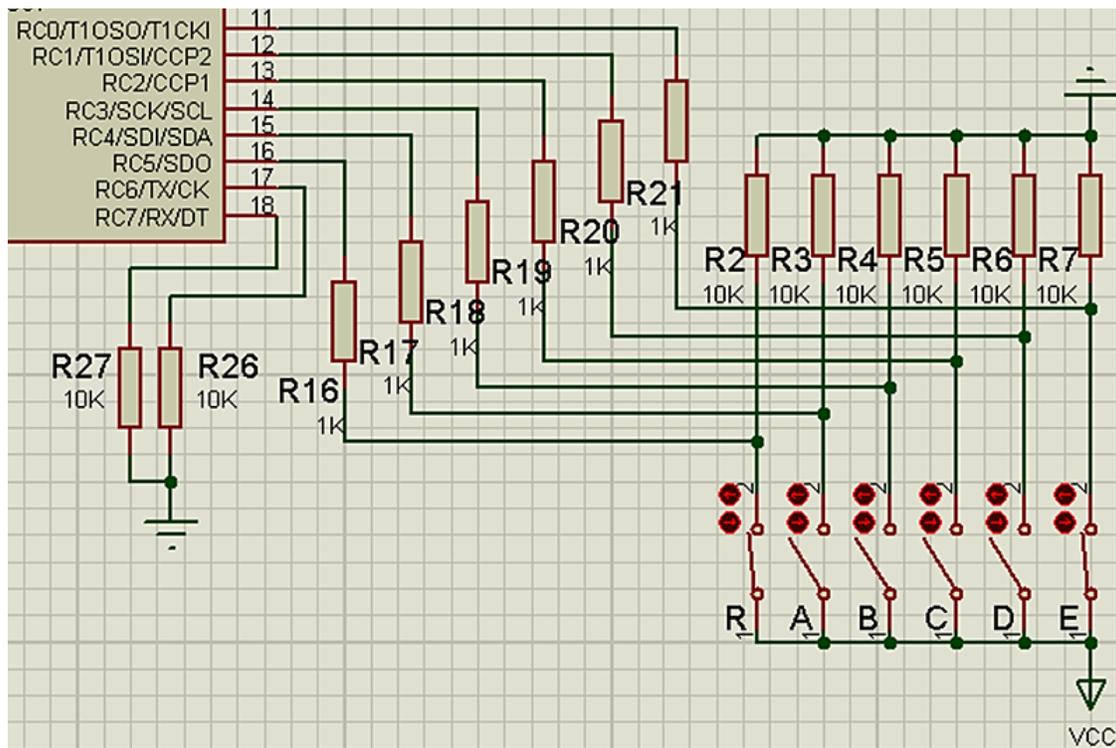


Figura 34. Conexiones de la etapa de entrada.

EL DISPLAY LCD ALFANUMÉRICO

Es una pantalla de cristal líquido capaz de mostrar caracteres alfanuméricos, puede desplegar información y permite una fácil implementación en circuitos microcontrolados.

Existen varios tipos pero su funcionamiento es el mismo. Los hay de 8 caracteres por 2 líneas, de 16 caracteres por 1 línea, de 16 caracteres por 2 líneas y 16 caracteres por 4 líneas.

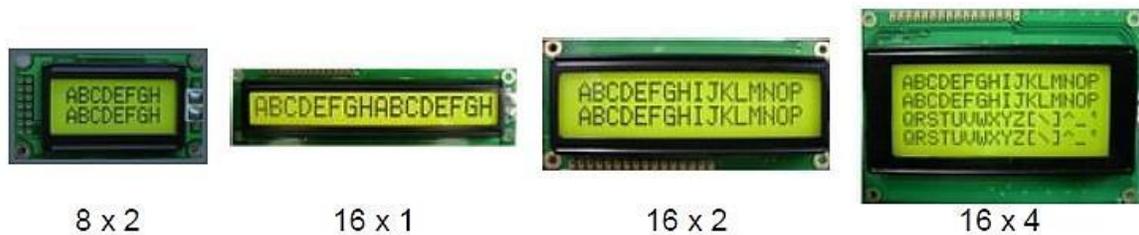


Figura 35. Tipos de Display LCD.

La pantalla representa a una matriz de caracteres de 5 x 8 puntos, distribuidos desde 1 o hasta 4 líneas de 16 caracteres cada una.

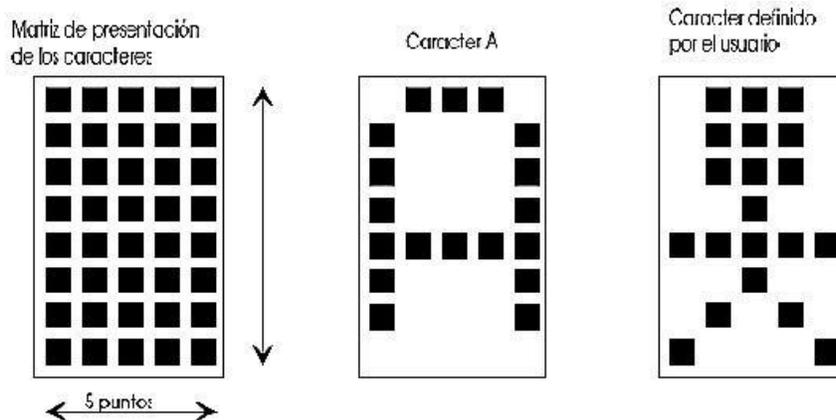


Figura 36. Matriz de caracteres del Display LCD.

CARACTERISTICAS
Su consumo es muy reducido del orden de los 7.5 mW
Desplazamiento de los caracteres en forma horizontal
Memoria de 40 caracteres por línea de pantalla, visualizándose 16 caracteres por línea
Movimiento del cursor y cambio de su aspecto
Intermitencia de los caracteres
Con o sin iluminación (Backlight)
Puede ser gobernado con conexión por bus de 4 bits u 8 bits

Figura 37. Tabla de características del Display LCD.

El display LCD posee una memoria ROM que contiene todos los caracteres que se pueden visualizar ya sea alfanuméricos ASCII, Kanji, Griegos y símbolos matemáticos.

Higher bit Lower bit	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
xxx-0000		0	1	2	3	4	5	6	7	8	9	A	B
xxx-0001		C	D	E	F	G	H	I	J	K	L	M	N
xxx-0010		O	P	Q	R	S	T	U	V	W	X	Y	Z
xxx-0011		[\]	^	_	`	{		}	~		
xxx-0100													
xxx-0101													
xxx-0110													
xxx-0111													
xxx-1000													
xxx-1001													
xxx-1010													
xxx-1011													
xxx-1100													
xxx-1101													
xxx-1110													
xxx-1111													

Figura 38. Tabla de caracteres alfanuméricos.

• **EL CONJUNTO DE INSTRUCCIONES**

El display LCD contiene en su interior un microcontrolador que controla la inicialización y el despliegue de la información por eso existe un conjunto de instrucciones de funcionamiento del módulo que depende directamente de las señales que se tengan desde el bus de datos y control del microcontrolador PIC16F873A que empleamos.

Instrucciones	Código										Descripción	t
	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0		
CLEAR DISPLAY	0	0	0	0	0	0	0	0	0	1	Limpia el display	1,64 mS
RETURN HOME	0	0	0	0	0	0	0	0	1	*	Coloca al display en la posición inicial	1,64 mS
ENTRY HOME SET	0	0	0	0	0	0	0	1	I/O	S	Fija el movimiento del cursor y el desplazamiento del display	40 uS
DISPLAY ON/OFF CONTROL	0	0	0	0	0	0	1	D	C	B	D: Display ON/OFF C: Cursor ON/OFF B: Parpadeo del cursor ON/OFF	40 uS
DISPLAY OR CURSOR SHIFT	0	0	0	0	0	1	S/C	R/L	*	*	Mueve el cursor y desplaza el display	40 uS
FUNCTION SET	0	0	0	0	1	DL	N	F	*	*	Define DL, N y F	40 uS
SET CG RAM ADDRESS	0	0	0	1	ACG						Define ACG	40 uS
SET DD RAM ADDRESS	0	0	1	ADD						Define ADD	40 uS	
READ BUSY FLAG & ADDRESS	0	1	BF	AC						BF: flag de ocupado Lee el contenido de AC	40 uS	
WRITE DATA TO CG OR RAM	1	0	WRITE DATA						Escribe un dato en DD RAM o CG RAM	40 uS		
READ DATA FROM CG OR DD RAM	1	1	READ DATA						Lee un dato de DD RAM o de CG RAM	40 uS		

NOTAS	<p>I/O = 1: Incremento 0: Decremento</p> <p>S = 1: Acompaña al desplazamiento del display</p> <p>S/C = 1: Desplazamiento del display 0: Movimiento del cursor</p> <p>R/L = 1: Desplazamiento a la derecha 0: Desplazamiento a la izquierda</p> <p>DL = 1: 8 BITS 0: 4 BITS</p> <p>N = 1: 2 LÍNEAS 0: 1 LÍNEA</p> <p>F = 1: 5 x 10 puntos 0: 5 x 7 puntos</p> <p>BF = 1: Operación interna 0: Puede aceptar una instrucción</p>	<p>DD RAM: RAM de datos del Display</p> <p>CG RAM: RAM del generador de caracteres</p> <p>ACG: Dirección de CG RAM</p> <p>ADD: Dirección de DD RAM y corresponde con la dirección del cursor.</p> <p>AC: Dirección del contador utilizada por DD RAM y CG RAM</p> <p>* SINEFECTO</p> <p>t: Tiempo de Ejecución</p>
--------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Figura 39. Instrucciones de control de la pantalla.

• **DESCRIPCIÓN DE TERMINALES**

A continuación se muestra la disposición de los pines del display LCD.

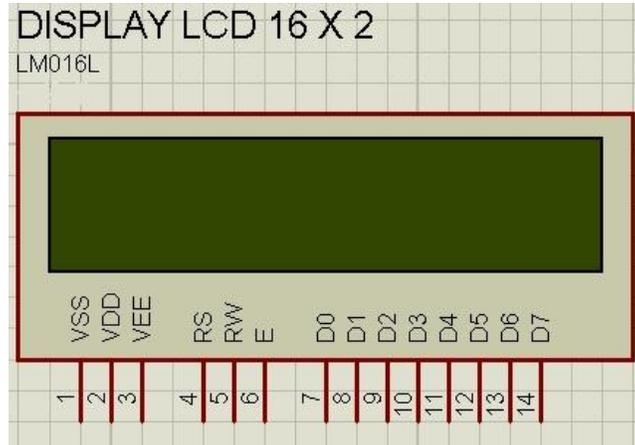


Figura 40. Disposición de pines.

NOMBRE	FUNCION	DESCRIPCION
VSS	ALIMENTACIÓN NEGATIVA	Gnd
VDD	ALIMENTACIÓN POSITIVA	+5V
VEE	AJUSTE DE CONTRASTE	Entre 0 y +5V se puede regular el contraste
RS	SELECTOR DE REGISTRO	RS=0 selección del registro de control (ejecución de las siguientes instrucciones) RS=1 selección del registro de datos (mantiene la visualización de caracteres)
RW	SELECTOR DE LECTURA / ESCRITURA	RW=0 habilita escritura RW=1 habilita lectura
E	SEÑAL DE ACTIVACIÓN	E=0 LCD desconectado E=1 LCD conectado
DO A D7	BUS DE DATOS BIDIRECCIONAL	transferencia de información con el lcd y el microcontrolador

Figura 41. Tabla descriptiva de los pines.

- **BUS DE DATOS**

CONEXIÓN CON BUS DE DATOS DE 8 BITS

Al emplear esta conexión, facilita enormemente la programación en el microcontrolador, pero existe la desventaja de que se ocupan ocho pines.

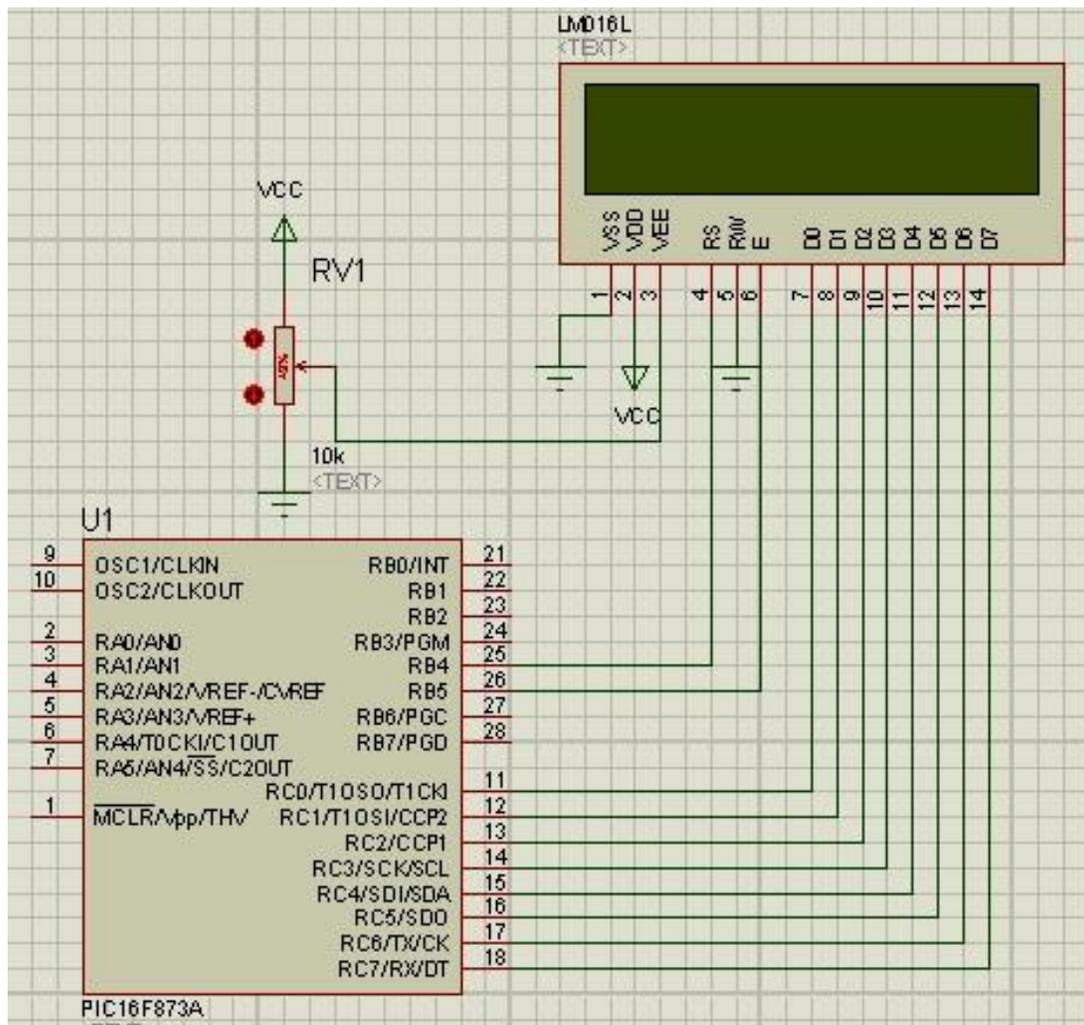


Figura 42. Bus de datos de 8 bits.

CONEXIÓN CON BUS DE DATOS DE 4 BITS

Al emplear esta conexión se ahorran 4 bits en el bus y la transferencia de la información se realiza primero los 4 bits más significativos y luego los 4 bits menos significativos.

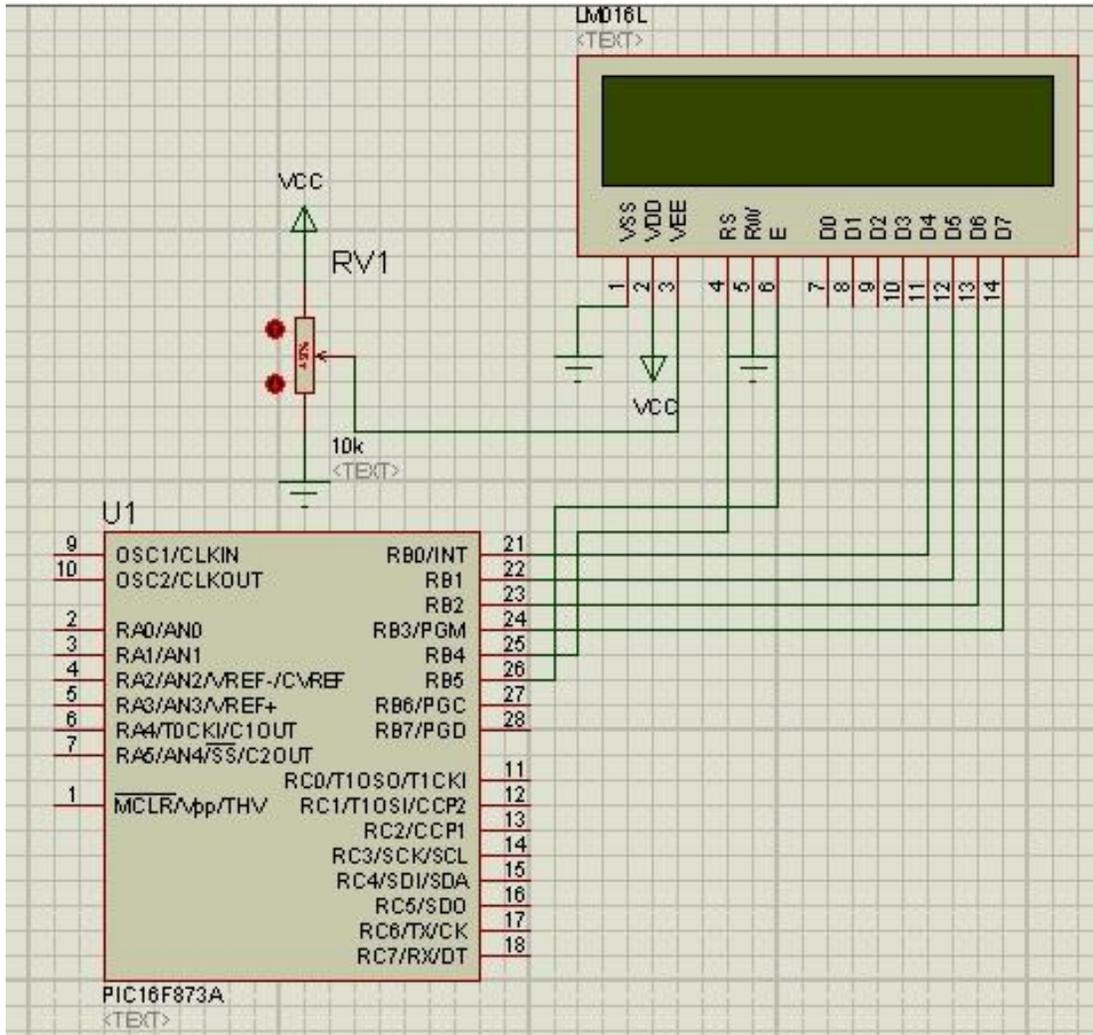


Figura 43. Bus de datos de 4 bits.

LA ALARMA AUDIBLE

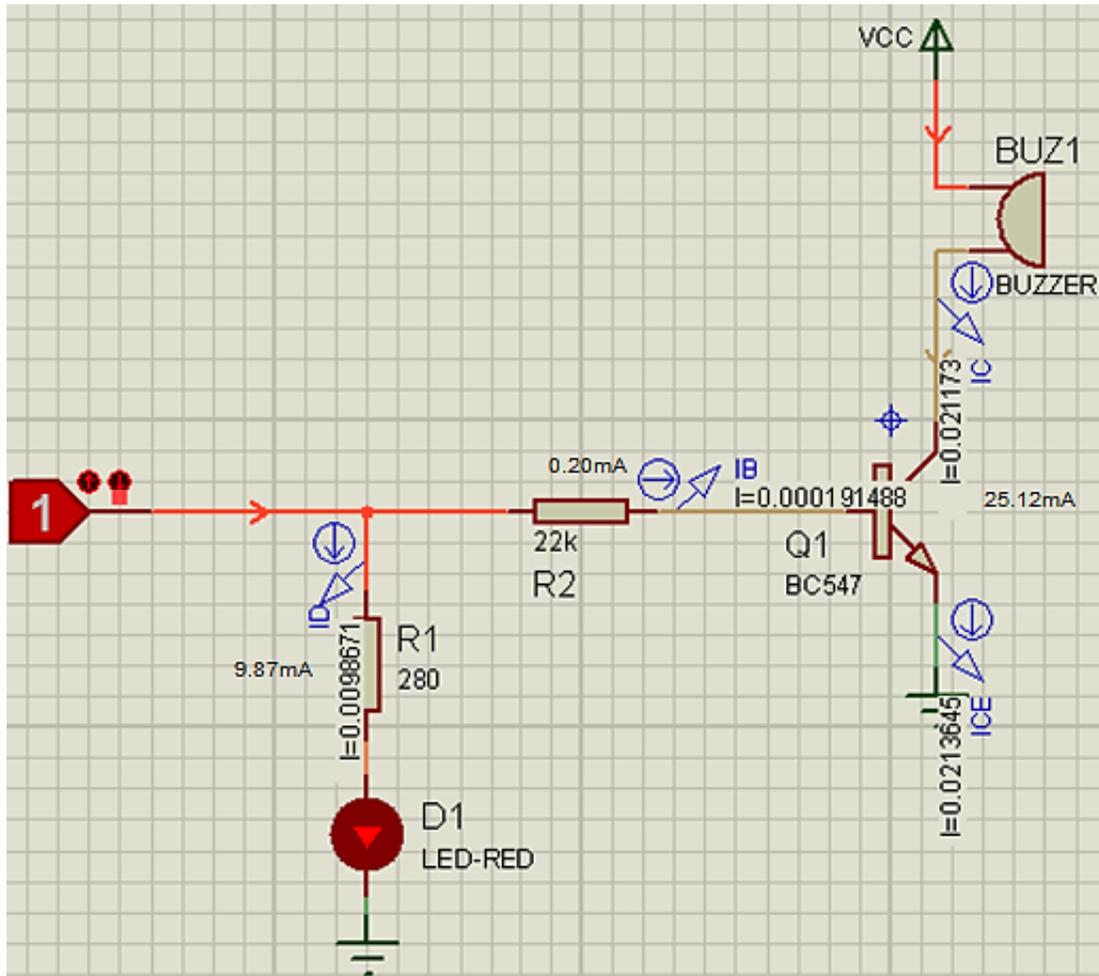


Figura 44. Simulación del circuito.

Se requiere usar un transistor en configuración emisor común debido a que la corriente que entrega el pin del microcontrolador es pequeña, solo 25mA muy por el límite de lo que consume el zumbador y que al conectarse directamente se corre el riesgo de quemar el pin.

Al haber corriente suficiente en la base del transistor, este se satura provocando que exista una circulación de corriente activando el zumbador

Para este diseño se elige al transistor BC 547 ya que en sus características la corriente de colector máxima que puede manejar es de 100mA con un voltaje colector – base de 50V, una ganancia de 200 por lo que es adecuado para la corriente y voltaje que demanda el zumbador. Es conveniente no superar el 50% de la corriente del colector máxima del transistor

Los datos del zumbador son los siguientes:

Voltaje 5V

Resistencia interna 200 Ohm

$$I_C = V/R = 5V/200 = 25mA$$

$$I_B = I_C / \beta = 25mA / 200 = 0.125mA \text{ corriente de saturación mínima necesaria}$$

$$R_B = (V_{CC} - V_{BE}) / I_B = (5V - 0.7V) / 0.125mA = 34.4k \text{ Para fines prácticos } R_B = 33K$$

$$I_{CE} = I_B + I_C = 25.12mA$$

En el diseño se empleó una resistencia de base de 22K puesto que:

$I_B = V/R = 5V/22K = 0.22mA$ que es mayor a la corriente de base calculada por lo tanto se satura.

Se ha agregado un diodo emisor de luz (LED) para que nos muestre la activación de la alarma de una forma también visual. El LED para que funcione necesita un voltaje de 2.2V y una corriente máxima de 10mA para que no se queme, teniendo estas especificaciones, podemos saber el valor de su resistencia, aplicando la ley de ohm se tiene:

$$R = V/I = 5V - 2.2V / 10mA = 280 \text{ Ohm}$$

Este será el valor de la resistencia conectada en serie con el diodo para no sobrepasar los 10mA de corriente y no se queme el LED, se elige una resistencia comercial de 330 Ohm.

Ya teniendo nuestro circuito, estará listo para recibir la señal de alarma del microcontrolador, solo cuando mande un estado lógico alto el zumbador comenzará a emitir un sonido.

ETAPA DE POTENCIA PARA LA BOMBA CENTRÍFUGA Y LA ELECTROVÁLVULA

Para poder controlar adecuadamente a la carga inductiva, en este caso una bomba centrífuga o una electroválvula, es necesario aislar eléctricamente al circuito digital con la etapa de potencia para prevenir que altas corrientes y voltajes de alterna, dañen la circuitería de control.



Figura 45. Acoplamiento de un circuito digital y de potencia.

- **EL OPTOACOPLADOR MOC3031**

Es un circuito integrado optoacoplador que es un dispositivo que contiene en su interior un diodo emisor de infrarrojo y que al activarse, incide sobre un detector de infrarrojo, en este caso un optotriac que contiene un detector de cruce por cero que activa al Triac de potencia solo cuando la señal de alterna pase por cero evitando así que los picos de corriente sean mayores cuando se conecta la carga.

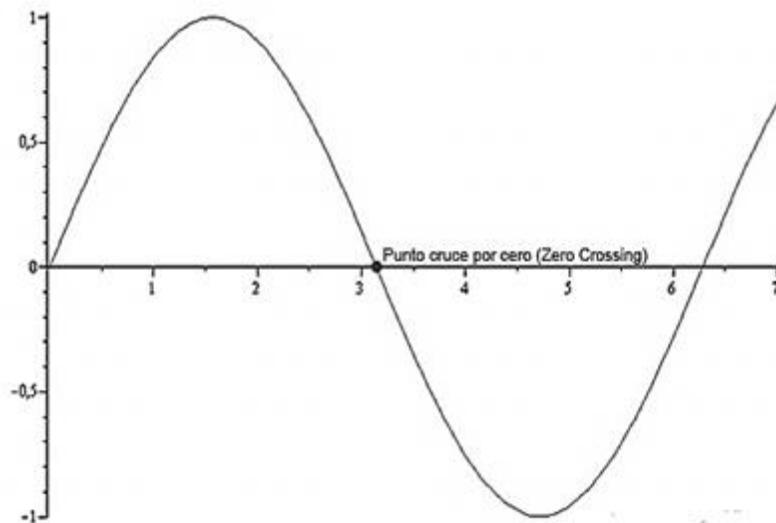


Figura 46. Cruce por cero.

Para este diseño se hace uso del optoacoplador MOC3031

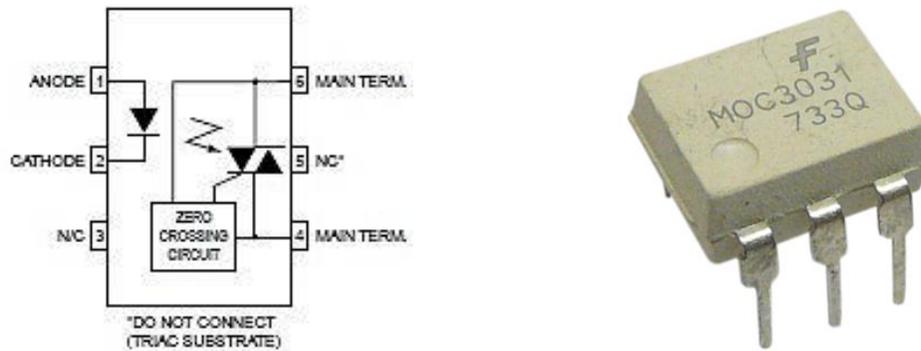


Figura 47. El MOC3031.

CARACTERISTICAS	
Corriente de disparo del led I_F	15mA
Corriente máxima que entrega ITSM	1A
Voltaje en terminal de salida en estado apagado VDRM	127V CA
Detector de cruce por cero	127V CA
Temperatura de operación	-40°C a 85°C
Potencia de disipación	250mW

Figura 48. Tabla de características del MOC3031.

- **EL TRIAC BTA24-800BW.**



Figura 49. Triac BTA24-800BW.

También se necesita un Triac (Triodo de Corriente Alterna) que pueda conectar la carga, y pueda soportar la corriente que consume la bomba centrífuga o la electroválvula.

Un triac es bidireccional debido a que conduce en ambas direcciones. Por el triac la corriente circula en cualquiera de las dos direcciones entre los terminales principales MT1 y MT2. Esto es iniciado por una corriente pequeña de señal aplicada entre el terminal de puerta y MT1

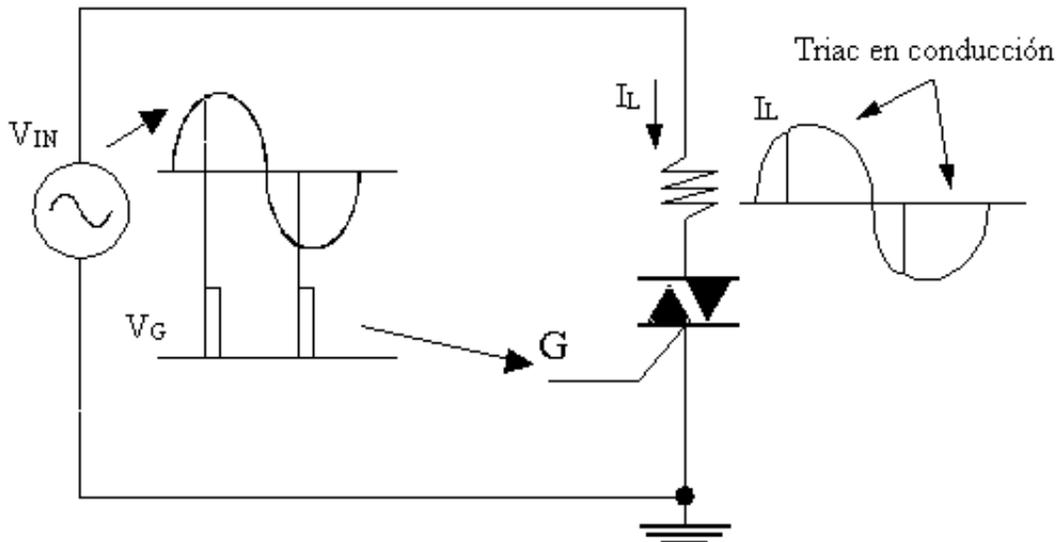


Figura 50. Disparo del Triac.

El triac estándar puede ser disparado por circulación de corriente positiva o negativa entre compuerta y MT1, Esto permite el disparo del triac en cuatro cuadrantes como se muestra.

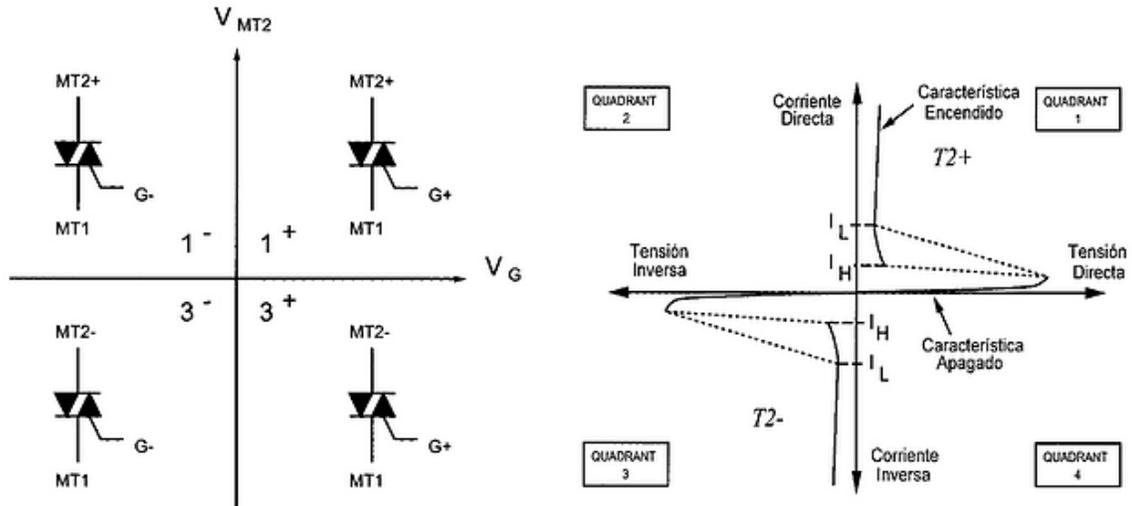


Figura 52. Cuadrantes del disparo del Triac.

Se eligió el Triac BTA24-800BW que puede soportar una carga máxima de 25A suficiente para este propósito, también porque internamente dispone de una red snubber para su protección.

CARACTERISTICAS	
Corriente a onda senoidal completa IT (RMS)	25A
Sensibilidad en IGT	50mA
Voltaje en MT1, MT2 máximo VDRM	800V

Figura 52. Tabla de características del Triac BTA24.

- **RED SNUBBER**

Las redes Snubber se utilizan para suprimir transitorios indeseables en elementos inductivos. La conmutación de estos elementos, pueden generar interferencias electromagnéticas (EMI) lo que conlleva que el triac se ponga en conducción inesperadamente o incluso a su destrucción.

El triac dispone de una red Snubber interna que lo protege y evita en lo más posible la emisión de dichas interferencias, si no se usa un triac de este tipo, es necesaria la conexión con una red Snubber que lo proteja.

- **ETAPA DE POTENCIA**

Este es el circuito de la etapa de potencia, dispone de un led indicador para cuando se active la bomba o en su defecto la electroválvula desde el microcontrolador. La resistencia en serie con el led limita la corriente para que no se queme, al igual que el led interno del optoacoplador, dispone de una resistencia que limita la corriente.

Para el led indicador $R_{12} = V/I = (5V - 2.2V)/10mA = 280 \text{ Ohm}$, se elige un valor de 330 Ohm.

De la hoja de características del MOC3031, el voltaje máximo para que se active el led emisor de infrarrojo es de $V_F = 1.5V$ con una corriente máxima $I_F = 15mA$. Como el voltaje de entrada es de 5V se calcula la resistencia en serie con el led.

$$R_{10} = V/I = (5V - 1.5V)/15mA = 233.3 \text{ Ohm}$$

La resistencia R11 limita la corriente a través del optoacoplador, el máximo valor de la corriente que puede proporcionar el MOC3031 es de $I_{TSM} = 1A$, considerando una tensión de red de 127V determinamos el voltaje pico.

$$V_{pico} = I_{TSM} \times V_{red} \times 1.4142 = 1A \times 127V \times 1.4142 = 179.60V$$

$$\text{Entonces } R11 = V_{pico} / I_{TSM} = 179.60V / 1A = 179.6 \text{ Ohm}$$

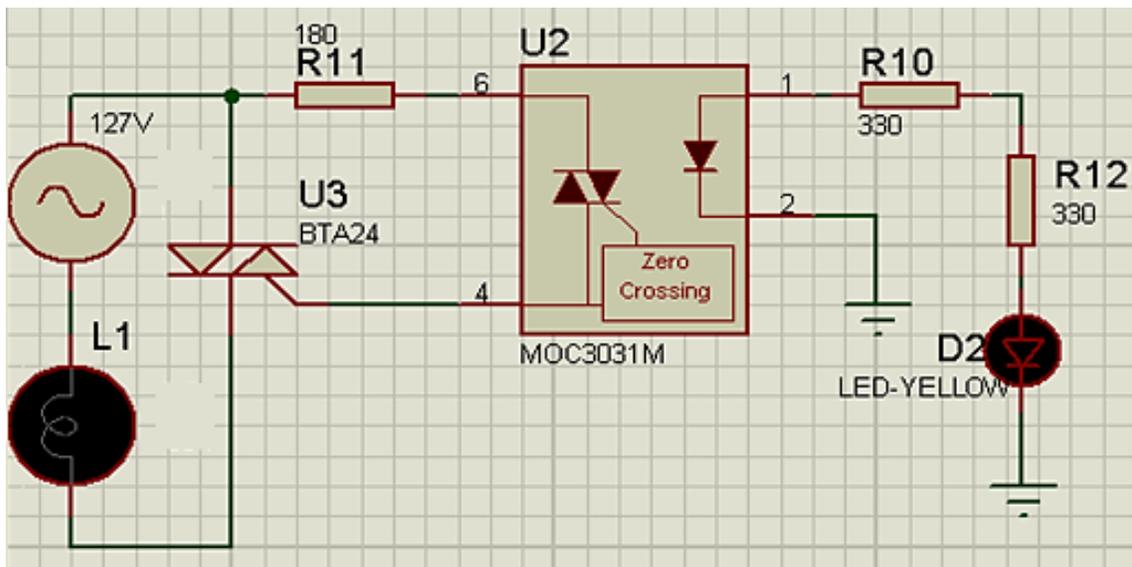


Figura 53. Etapa de potencia

LA FUENTE DE ALIMENTACIÓN

Las características de la fuente de alimentación que se necesita para alimentar los circuitos digitales, así como la electroválvula son de un voltaje de corriente directa de 5V y 24V de corriente alterna respectivamente. Esta fuente proporciona una corriente máxima de 500mA que son suficientes para alimentarlos.

Los bloques que conforman una fuente de alimentación de 5V regulada se detalla a continuación.

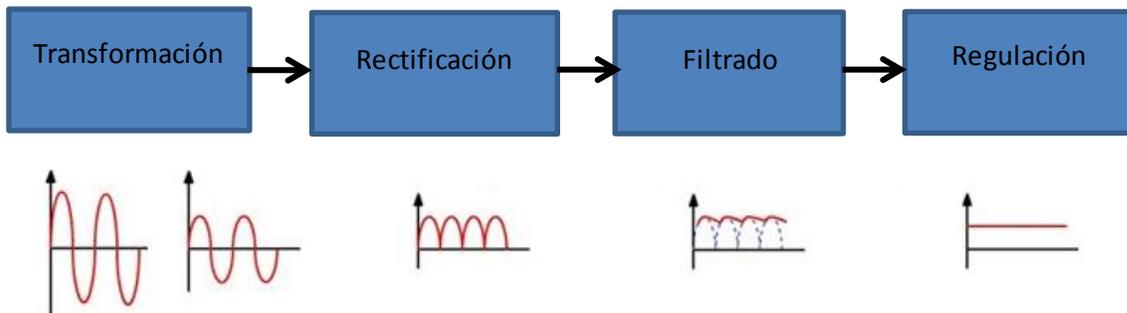


Figura 54. Etapas que conforman la fuente de alimentación.

- **LA TRANSFORMACIÓN**

Se requiere un transformador para reducir la tensión a un nivel adecuado, en este proyecto se eligió un transformador con 127V de corriente alterna y 60 Hz. de frecuencia en el primario y 12+12V 500mA de corriente alterna en el secundario, dando un total de 24V. Para la circuitería digital se requiere una tensión de 5V C.D y para alimentar la electroválvula de 24V C.A 300mA.

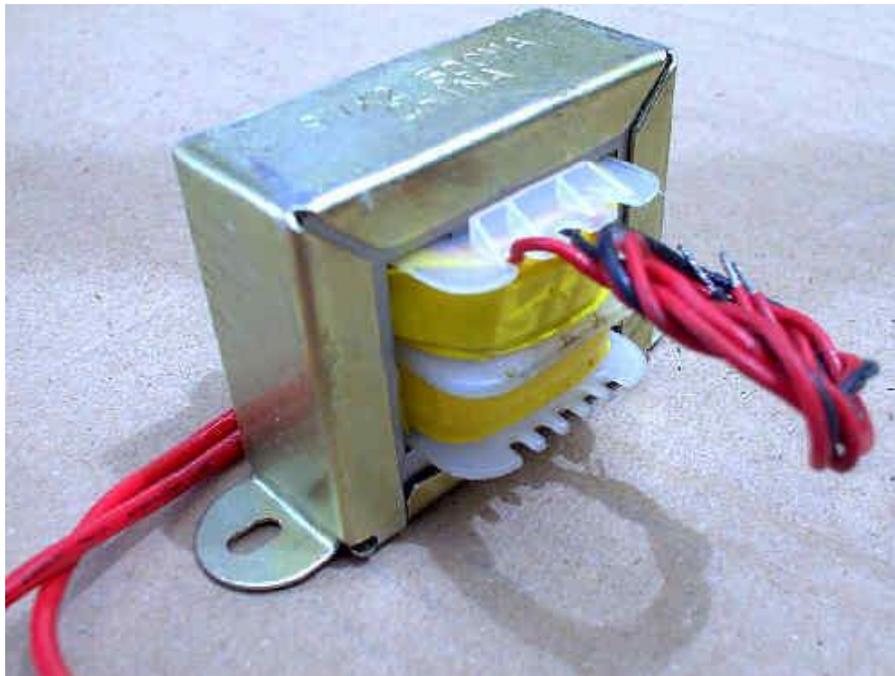


Figura 55. El transformador.

- **LA RECTIFICACIÓN DE ONDA COMPLETA**

Esta etapa se encarga de convertir la corriente alterna que sale en el secundario del transformador, en corriente continua pulsante.



Figura 56. Puente rectificador.

Se hace uso de un puente de diodos, cuando el voltaje de salida del transformador está en el semiciclo positivo, los diodos D2 y D3 conducen, mientras que D1 y D4 no, pero cuando pasa al semiciclo negativo, D1 y D4 conducen y D2y D3 ya no, de tal forma que para ambos semiciclos, la corriente es positiva en la carga, obteniendo así una corriente continua pulsante.

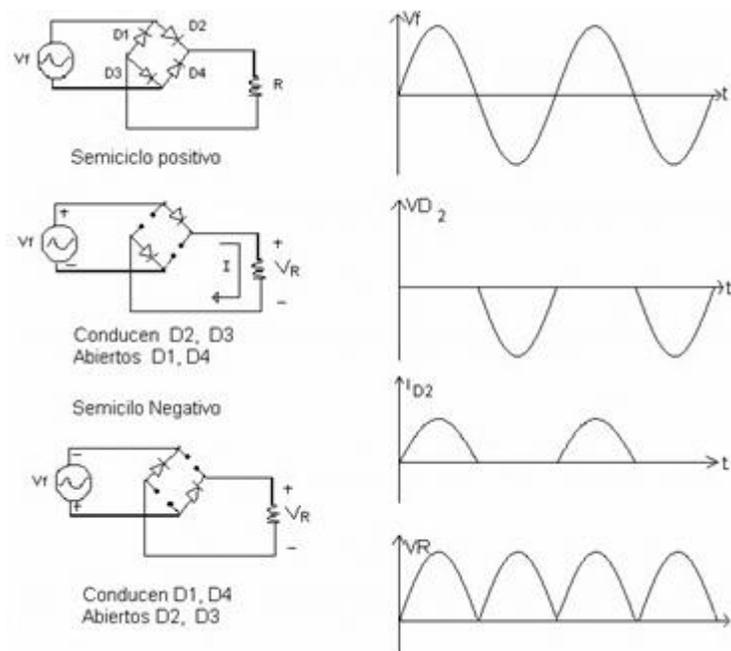


Figura 57. Proceso de rectificación.

Entonces el voltaje pico de salida será:

$$V_{\text{máx}} = 12 (1.4142) - 2 = 14.97\text{V}$$

- **EL FILTRADO**

Como se obtiene de la etapa de rectificación una corriente continua pulsante, los circuitos electrónicos pueden quemarse o producirse fallas ya que este tipo de corriente no es lo correcto para dichos circuitos, lo correcto es que se obtenga una corriente directa y para esto se requiere de un capacitor conectado en paralelo con la carga. Cuando los diodos conducen, el capacitor se carga con el voltaje pico de salida del rectificador, pero cuando no conducen el capacitor se va descargando a través de la carga. Cuando vuelven a conducir los diodos, el capacitor todavía mantiene la tensión en sus terminales y se vuelve a cargar obteniendo el valor de la tensión de la salida del rectificador, obteniendo una corriente de directa pero con un voltaje de rizo originado por la carga y descarga del capacitor.



Figura 58. Capacitor de filtrado.

Para determinar el valor del capacitor en este diseño se hace uso de la siguiente fórmula que consiste en obtener un voltaje de rizo del 10%.

$$C = (5 * I) / (f * V_{\text{máx}})$$

Dónde:

C = Capacidad del capacitor en faradios

I = Corriente que suministrará la fuente

F = Frecuencia de la red eléctrica

$V_{m\acute{a}x}$ = Voltaje pico de salida del rectificador

Entonces:

$$C = (5 * 0.5A) / (60Hz * 14.97V) = 2783.34 \text{ microfaradios}$$

Se elige un valor comercial de 3300 microfaradios, con una tensión de trabajo de 25V.

- **LA REGULACIÓN.**

Esta etapa se encarga de regular la tensión exactamente al valor que se desea, en este caso 5V de corriente directa y sirve también para disminuir el voltaje de rizo.

Para esto se necesita un regulador integrado de tres terminales. Para la elección de este dispositivo, se toma en cuenta el voltaje de salida que se desea obtener y la corriente máxima.

Se elige entonces el regulador integrado de la serie 7805 que proporciona máximo una corriente de 1A con una tensión de salida de 5V. La tensión de entrada al regulador, deberá de ser siempre de 2 a 3 volts superior a la tensión de salida, si son 5V de salida entonces mínimo se requiere 8V y máximo 35V.



Figura 59. Regulador de tensión 7805

La potencia disipada genera calor y depende de la corriente que se entrega en la carga y de la caída de tensión que hay en el regulador.

$$P = (V_{in} - V_{out}) * I \quad P = (15V - 5V) * 0.5A = 5W$$

El regulador que se emplea es capaz de disipar de 2 a 3W máximo y como la potencia disipada en nuestro diseño es de 5W, es necesario de un disipador adecuado, su mica y pasta térmica.

A continuación se muestra la simulación del circuito completo de la fuente de alimentación de 5V, en el secundario del transformador se toman los 24V de corriente alterna que necesita la electroválvula y en la salida del regulador se colocó un diodo led que nos indicará el encendido.

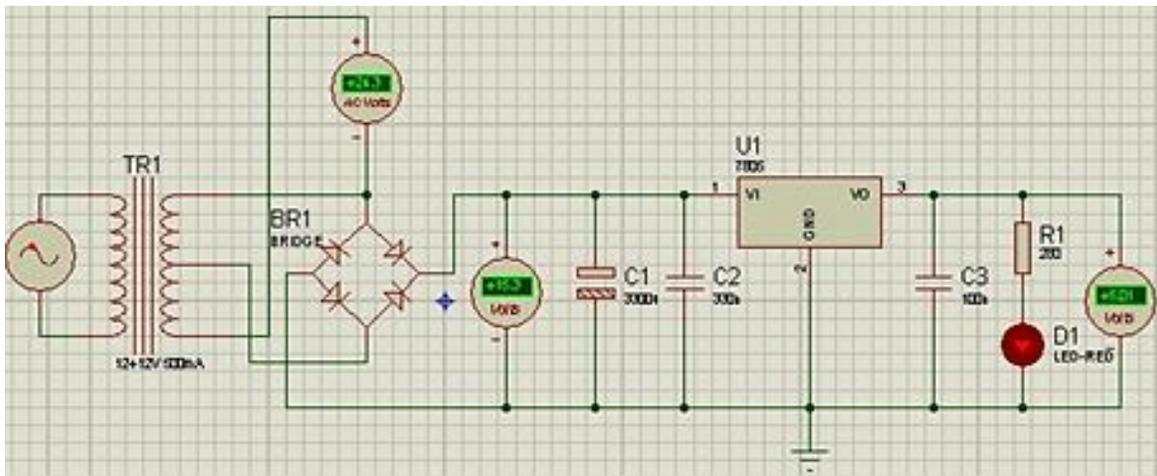


Figura 61. Simulación de la fuente de alimentación.

DIAGRAMA ELECTRÓNICO DEL CONTROL AUTOMÁTICO

Este diagrama se compone de todas las etapas descritas anteriormente.

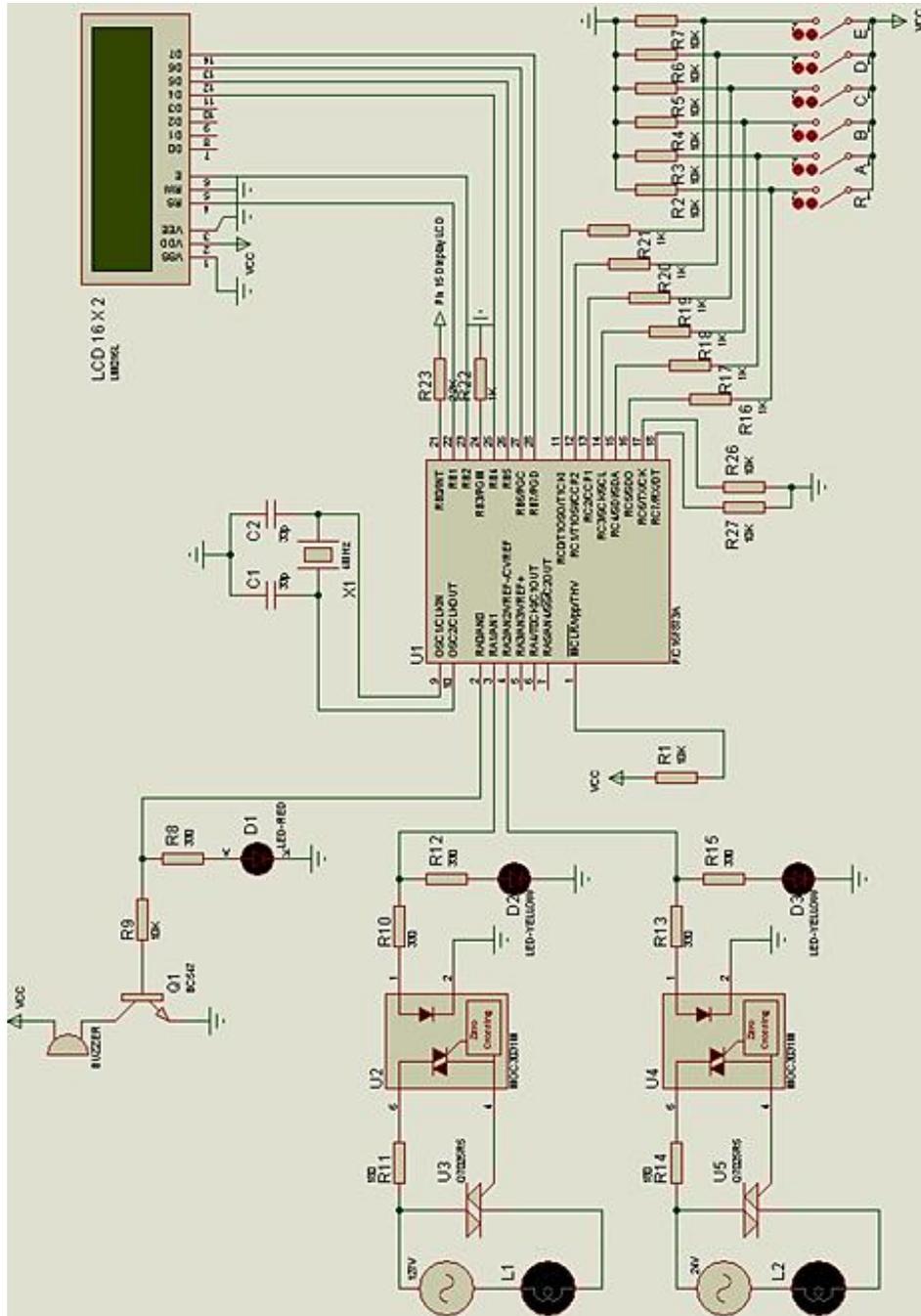


Figura 62. Diagrama electrónico completo.

CAPITULO 3

EL SOFTWARE

LENGUAJE DE PROGRAMACIÓN FLOWCODE PARA MICROCONTROLADORES PIC

El lenguaje de programación proporciona una intuitiva interfaz de programación de diagramas de flujo.

Desarrollado por la empresa Matrix multimedia, se trata de un lenguaje gráfico orientado a iconos con los que el programador diseña su aplicación a partir del diagrama de flujo del mismo.

Flowcode integra un gran número de subrutinas de alto nivel para la gestión de diversos componentes, lo que facilita un rápido desarrollo de cualquier aplicación. Cuando se diseña un programa, su funcionamiento se puede simular totalmente en la computadora.

Flowcode integra las funciones necesarias para el control de los componentes más típicos usados en cualquier aplicación. Led, interruptores, teclados, pantallas, Display LCD, Display de 7 segmentos, memorias EEPROM, dispositivos de comunicación, etc.

Cabe señalar que ya obtenido el diagrama de flujo, Flowcode puede realizar la compilación del código a lenguaje C, a lenguaje Ensamblador y generar el archivo hexadecimal que se grabará en el microcontrolador.

A continuación se describen las secciones más importantes de la interfaz del programa.



Figura 63. Interfaz del programa FLOWCODE.

LA CAJA DE HERRAMIENTAS.

Contiene en forma de botones, algunas de las opciones del menú más utilizadas como Nuevo, Abrir, Guardar, Copiar, Cortar, imprimir, etc. También hay botones para la simulación del programa de la aplicación.

HERRAMIENTAS DE COMANDOS.

Contiene los diferentes iconos con los cuales vamos a confeccionar el diagrama de flujo correspondiente al programa de la aplicación.

ÁREA DE TRABAJO.

Es el área donde se van colocando todos los íconos o comandos que conforman el diagrama de flujo, se conectan entre sí y se establecen sus parámetros y propiedades.

HERRAMIENTAS DE COMPONENTES.

Cada uno de esos botones nos permite seleccionar los diferentes dispositivos periféricos que pueden ser utilizados por nuestra aplicación, ya sea dispositivos de entrada, de salida, de comunicaciones, etc.

PANEL DE MONTAJE

Es el área donde se irán colocando los componentes y periféricos que se emplean en la aplicación.

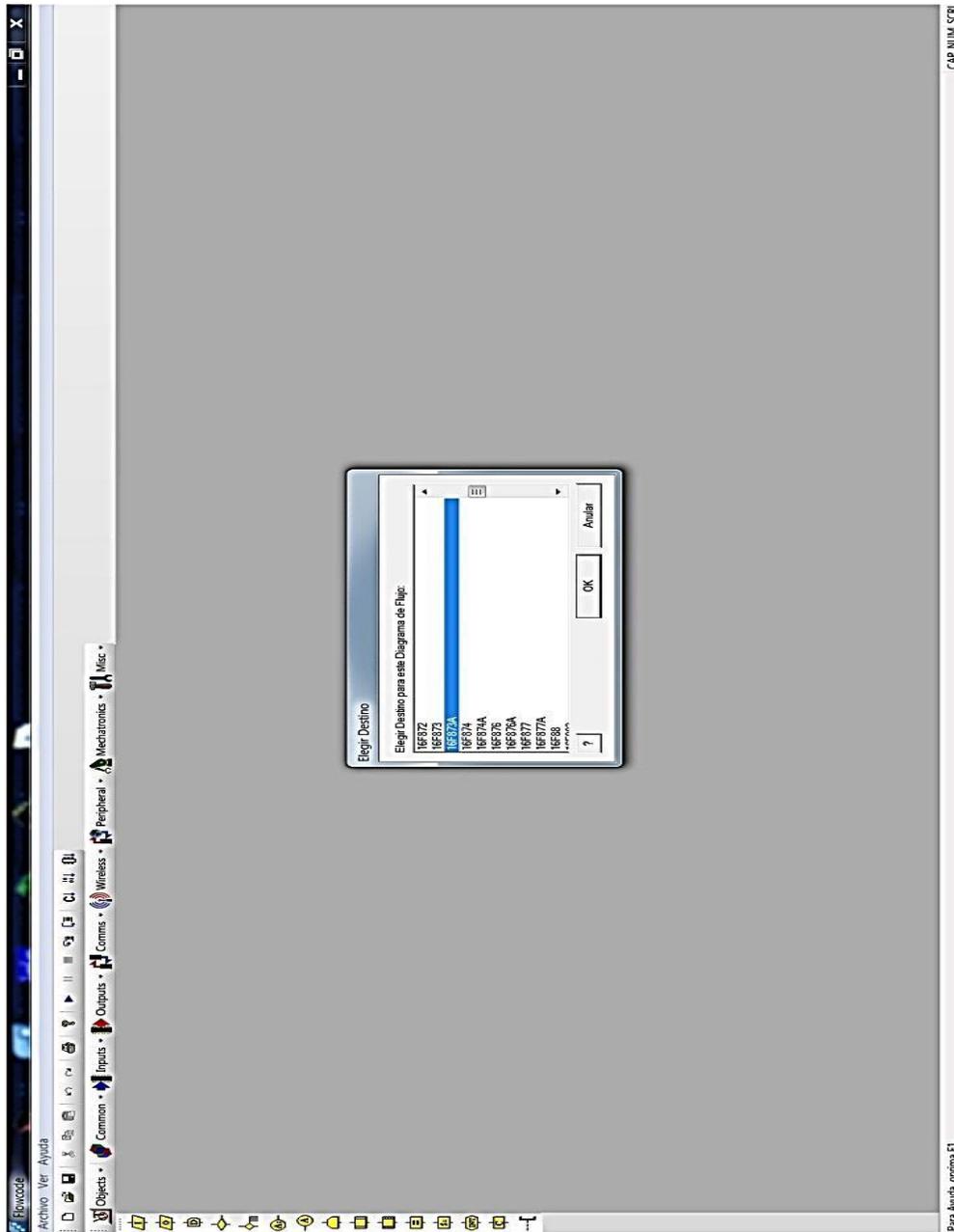
PROPIEDADES DEL PANEL

En esta ventana se editan las propiedades de los objetos que se encuentran en el panel de montaje.

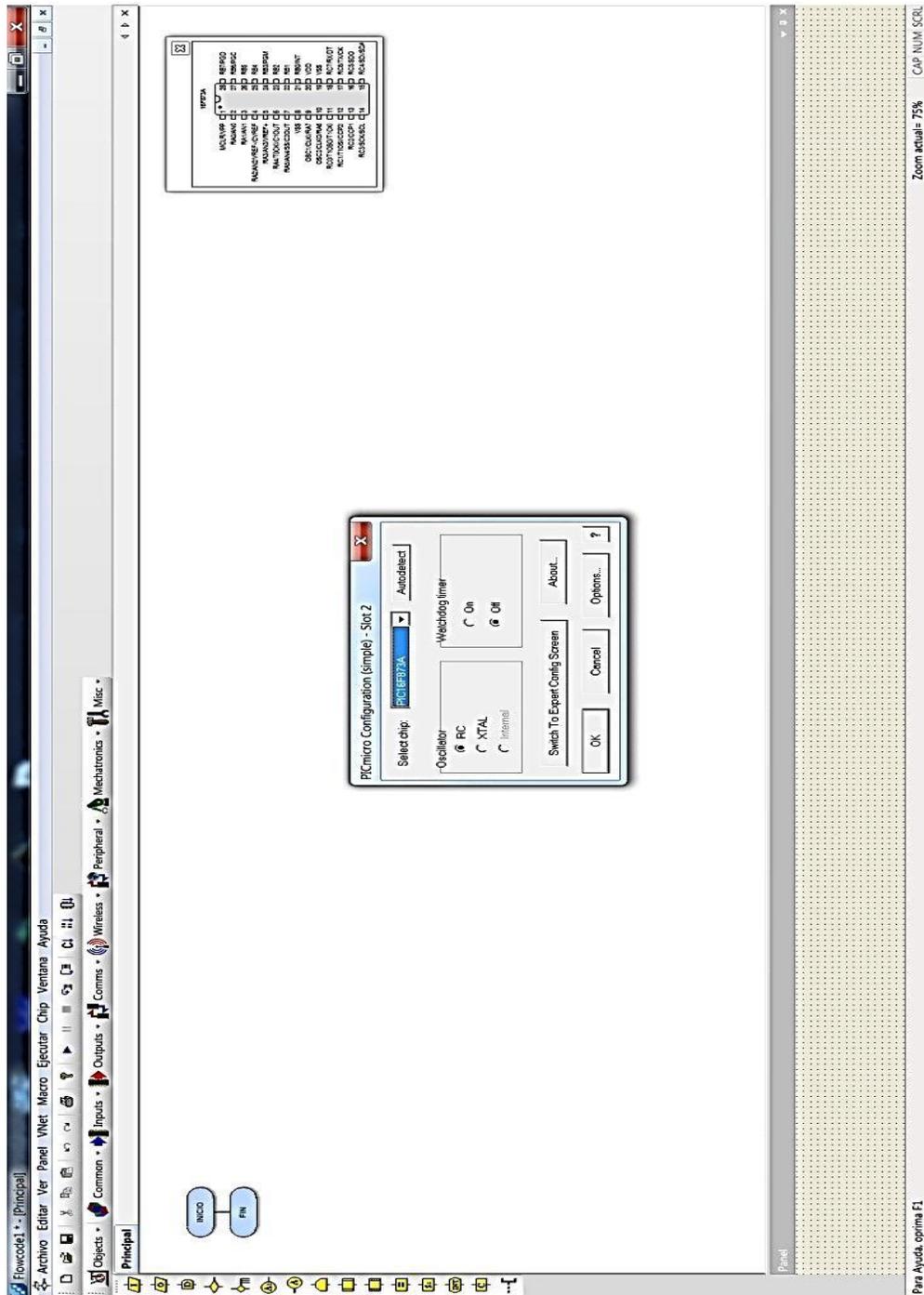
BARRA DE ESTADO

Se visualizan diversos mensajes que informan el estado del sistema.

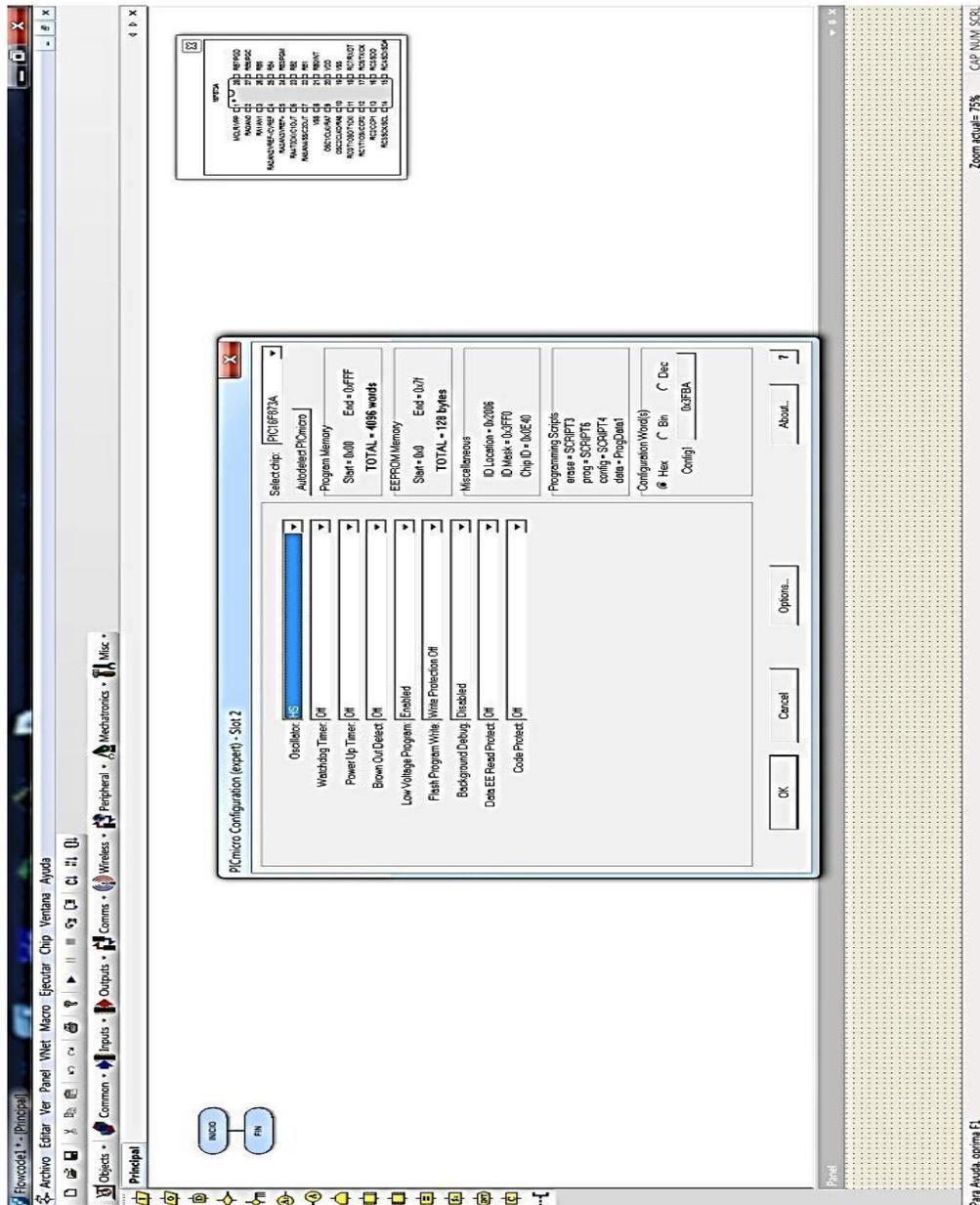
ELECCIÓN Y CONFIGURACIÓN DEL MICROCONTROLADOR



Al abrir el programa Flowcode, se abre la ventana de elección del microcontrolador a usar, en este caso seleccionamos de la lista al PIC 16F873A y pulsamos ok. Con esto el programa Flowcode es preparado para hacer compatible el diagrama de flujo a crear con el dispositivo a programar.

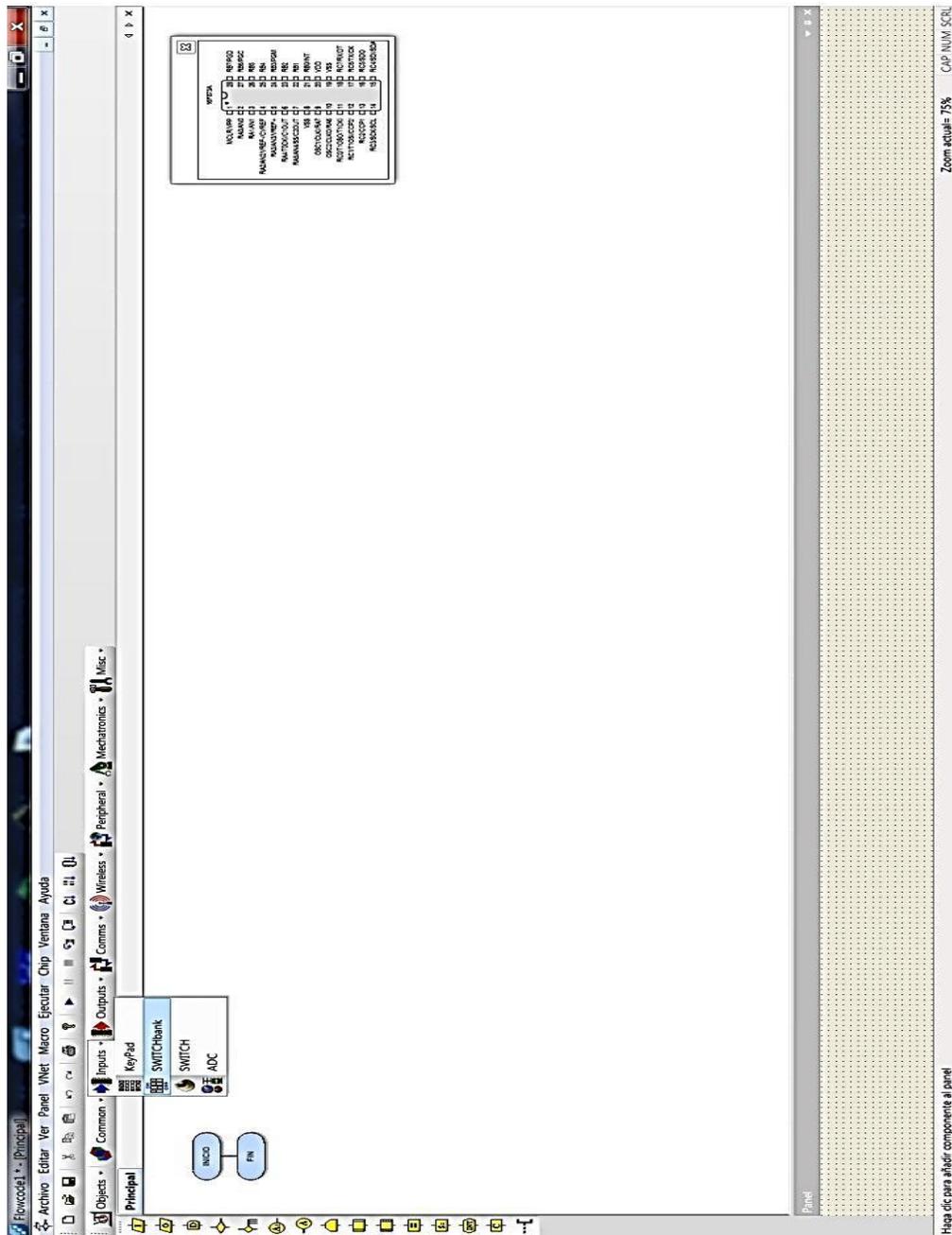


Nos abre una ventana donde podemos configurar el oscilador que se va a emplear, entre otras configuraciones, pulsamos el botón “Switch to Expert config Screen” donde nos abre otra ventana de configuración más avanzada.

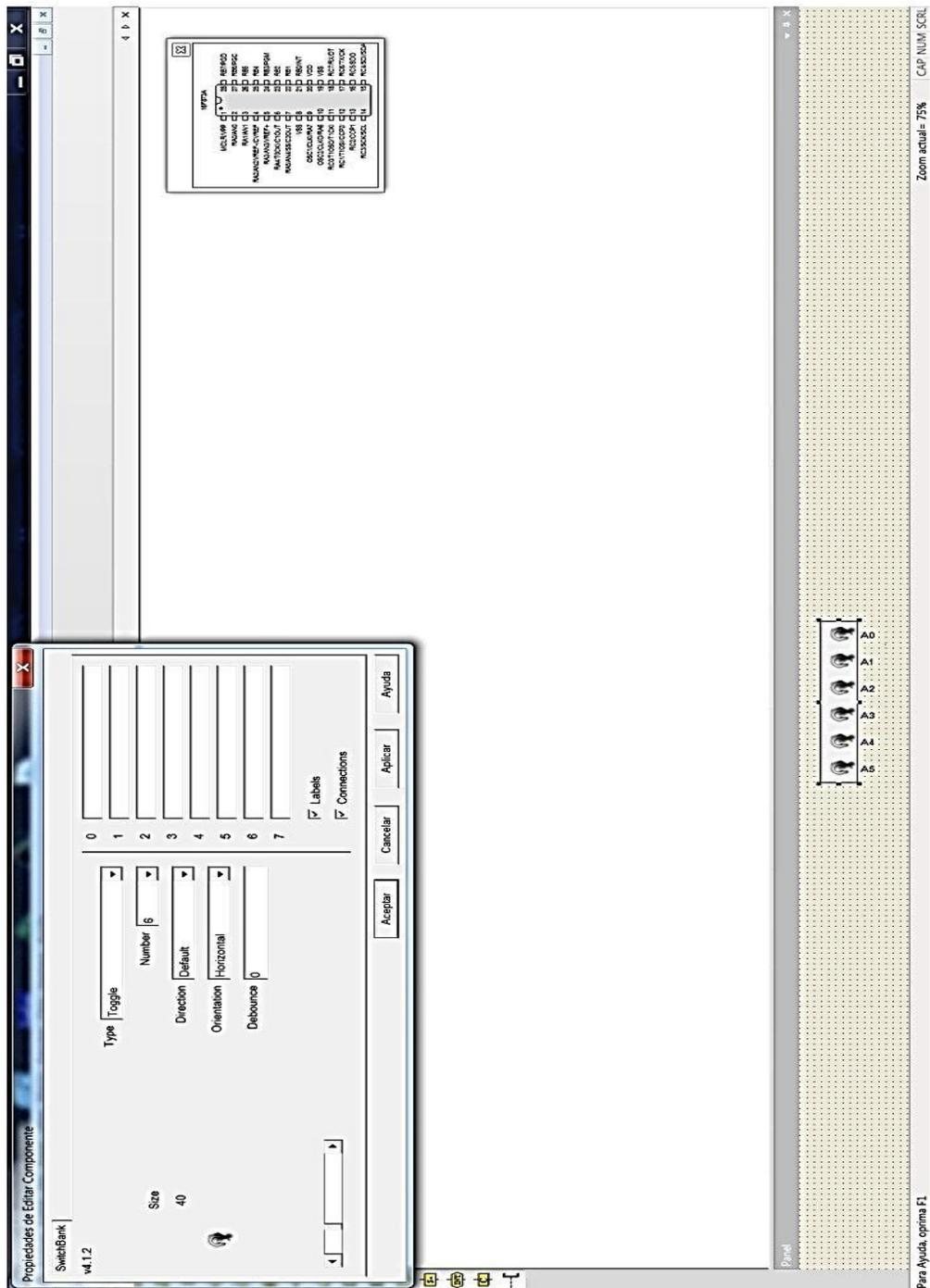


Configuramos el tipo del oscilador, en este caso elegimos HS y habilitamos Low Voltage Program para que cuando se requiera grabar el programa en el microcontrolador, este se pueda grabar en bajo voltaje de programación de 5V sin tener la necesidad de sacarlo del circuito de aplicación, de otro modo se tendrá que sacar del circuito y colocarlo en un circuito programador que funciona para programar, 12V por el pin MCLR del microcontrolador.

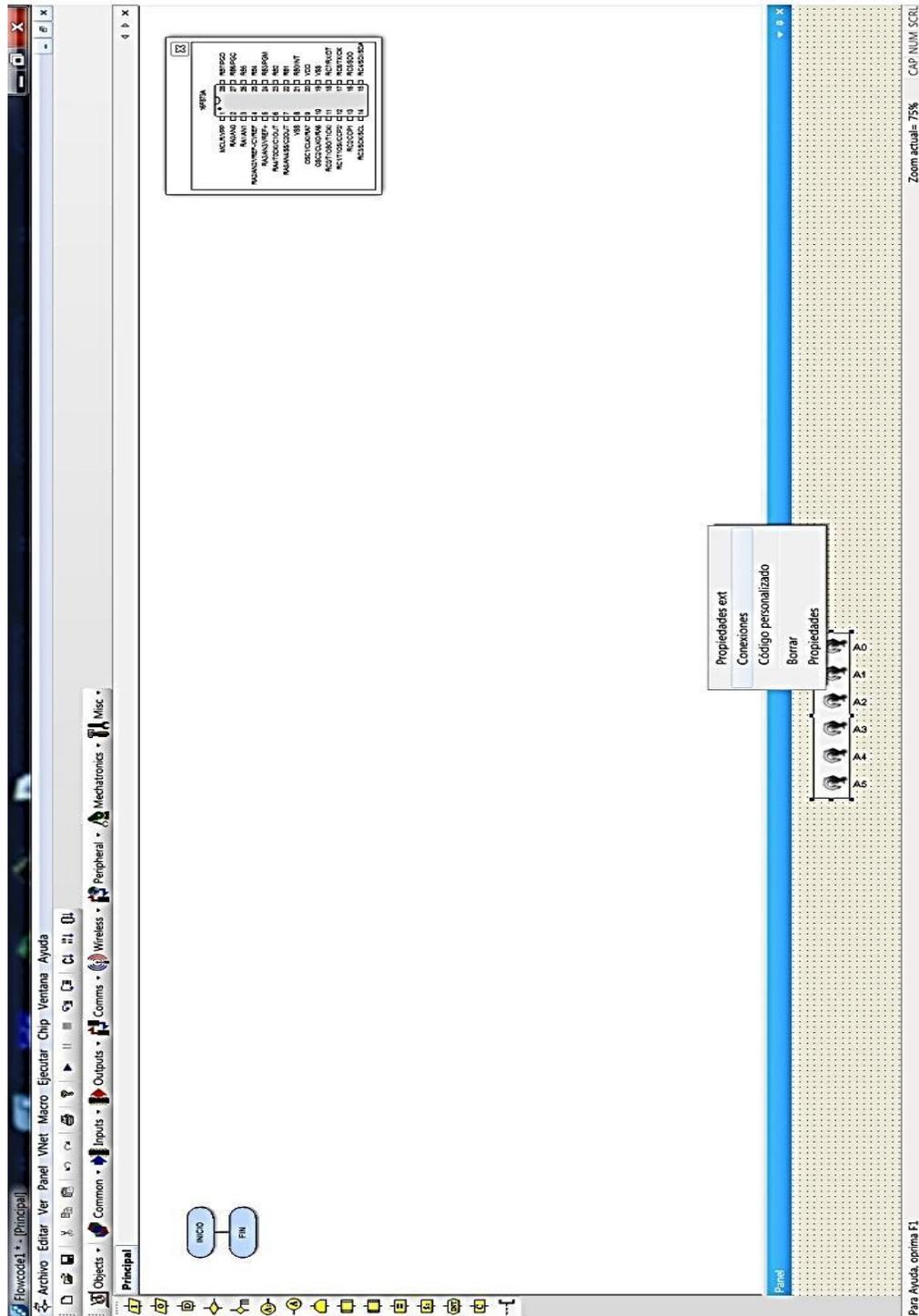
CONEXIÓN Y PROPIEDADES EXTERNAS DE LOS INTERRUPTORES



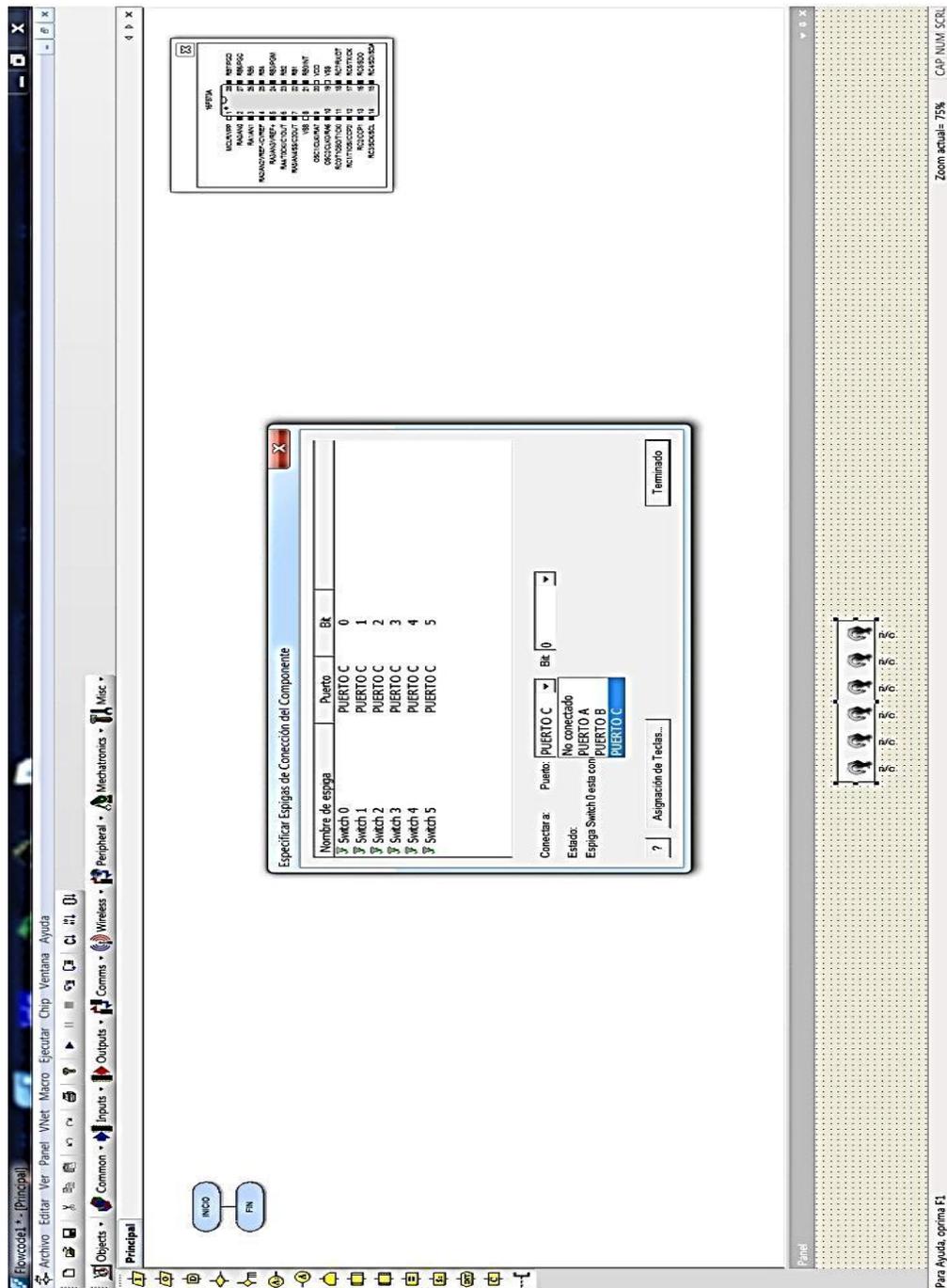
Como necesitamos conectar al puerto de entrada del microcontrolador seis interruptores, desplegamos de la herramienta de componentes, la opción Inputs y seleccionamos SWITCHbank, enseguida se colocan los interruptores en el panel de montaje.



Nos abre la ventana de editar componente, donde podemos cambiar algunas propiedades como el número de interruptores a usar o su posición ya sea vertical u horizontal. En este caso elegimos 6 interruptores en posición horizontal.

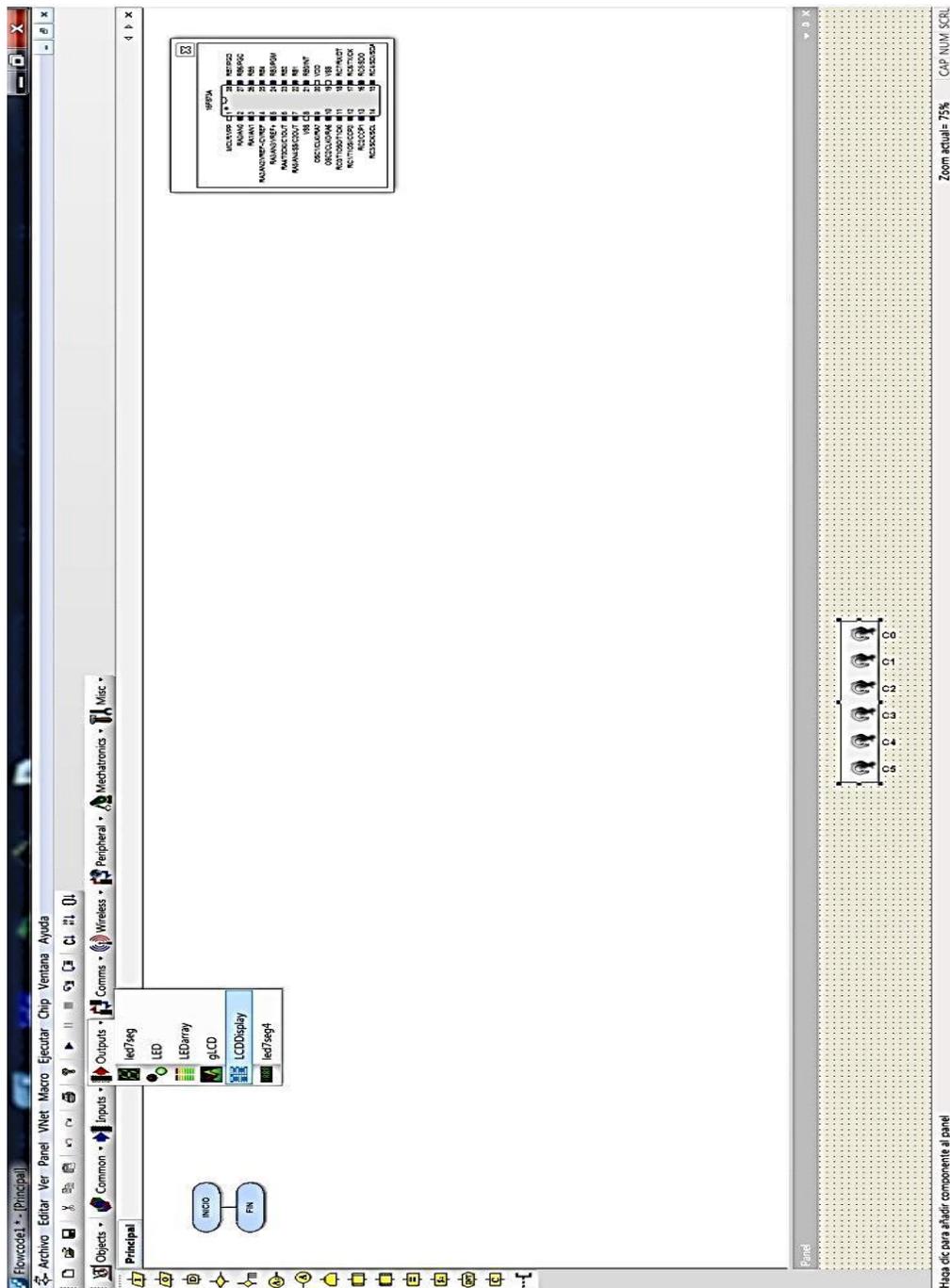


Desplegamos de nuevo el menú contextual en los interruptores y seleccionamos Conexiones.

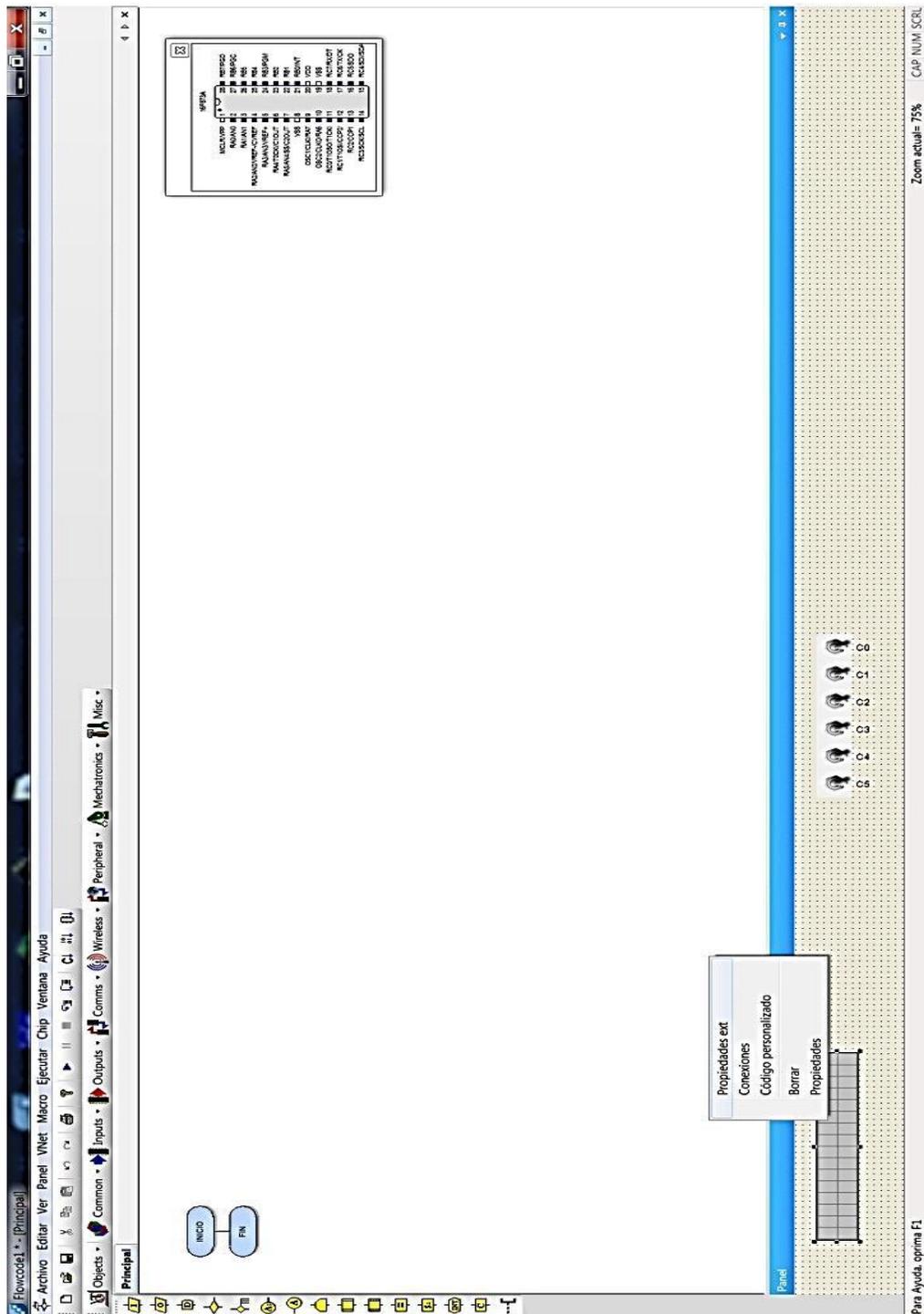


Ahora se abre la ventana de conexiones del componente y se procede a conectarlo al puerto C del microcontrolador, donde RC0 corresponde al interruptor E, RC1 al interruptor D, RC2 al interruptor C, RC3 al interruptor B, RC4 al interruptor A y RC5 al interruptor R.

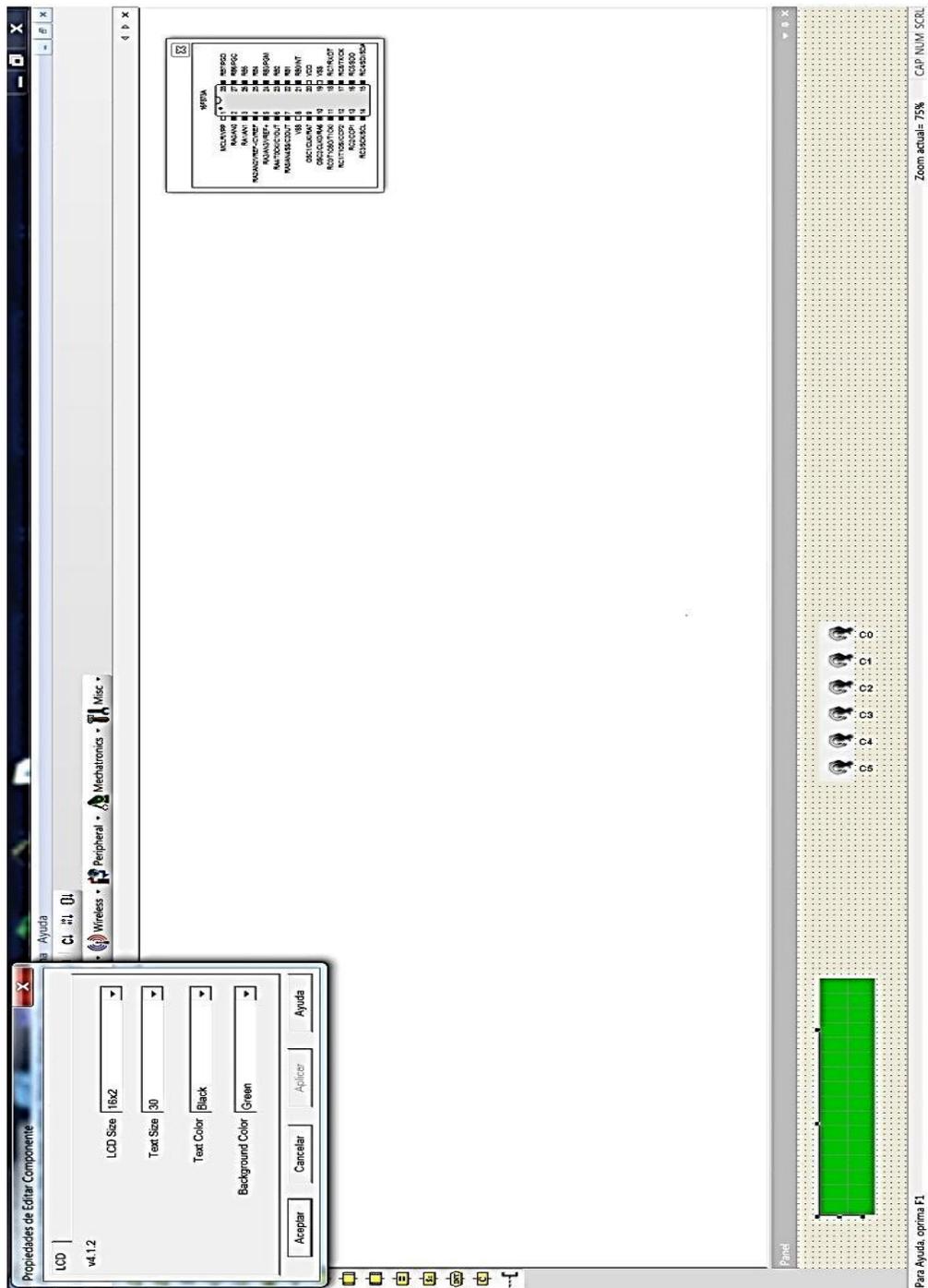
CONEXIÓN Y PROPIEDADES EXTERNAS DEL DISPLAY LCD



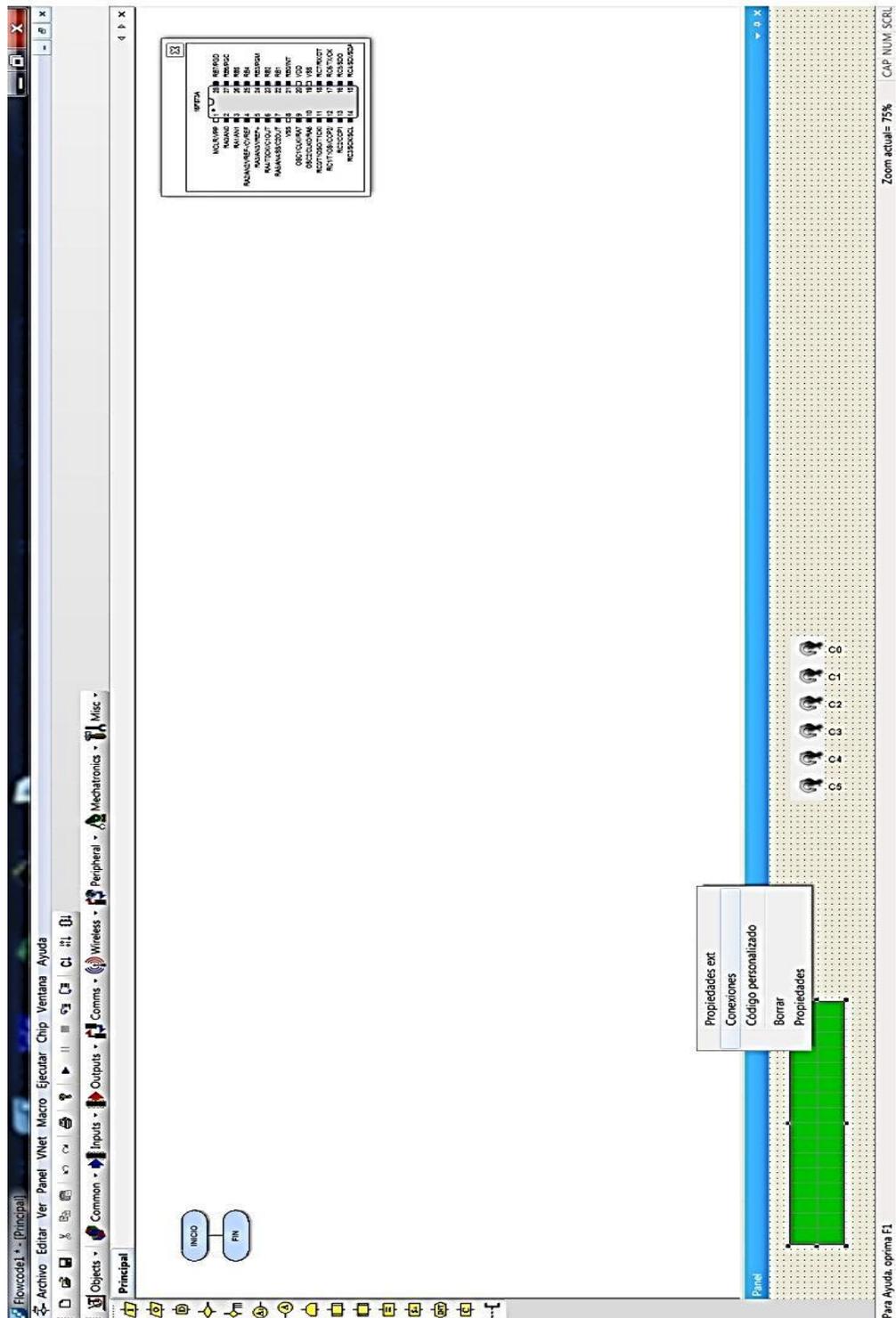
Desde la herramienta de componentes, desplegamos el menú Outputs y seleccionamos LCDDisplay, inmediatamente se coloca el display en el panel de montaje.



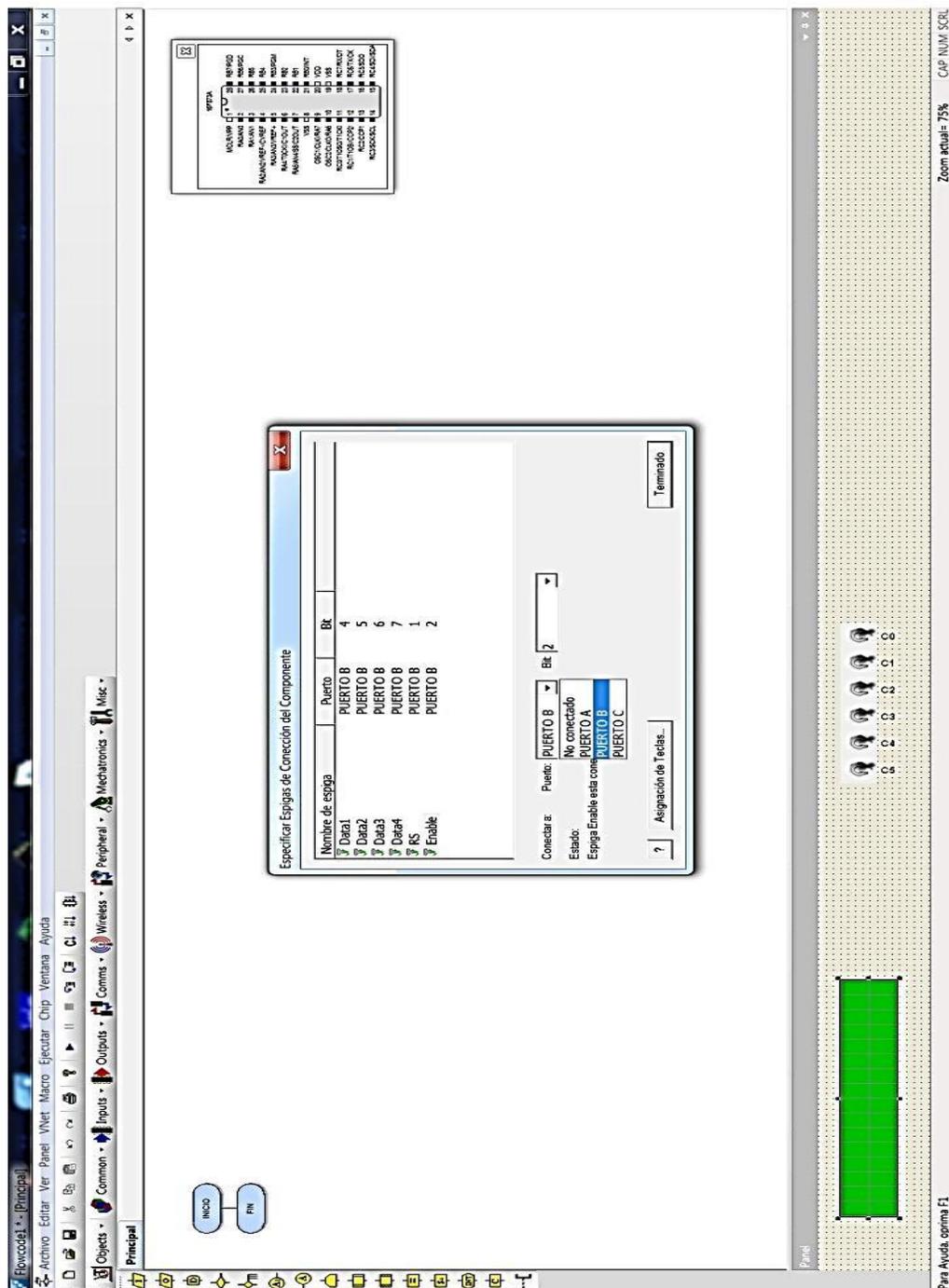
Desplegamos el menú contextual en el componente y seleccionamos Propiedades externas.



Se abre la ventana de edición del componente, donde se puede cambiar el tipo de display que se va a usar en este caso elegimos uno de 16 x 2 con el tamaño de texto en 30 y color de fondo verde.

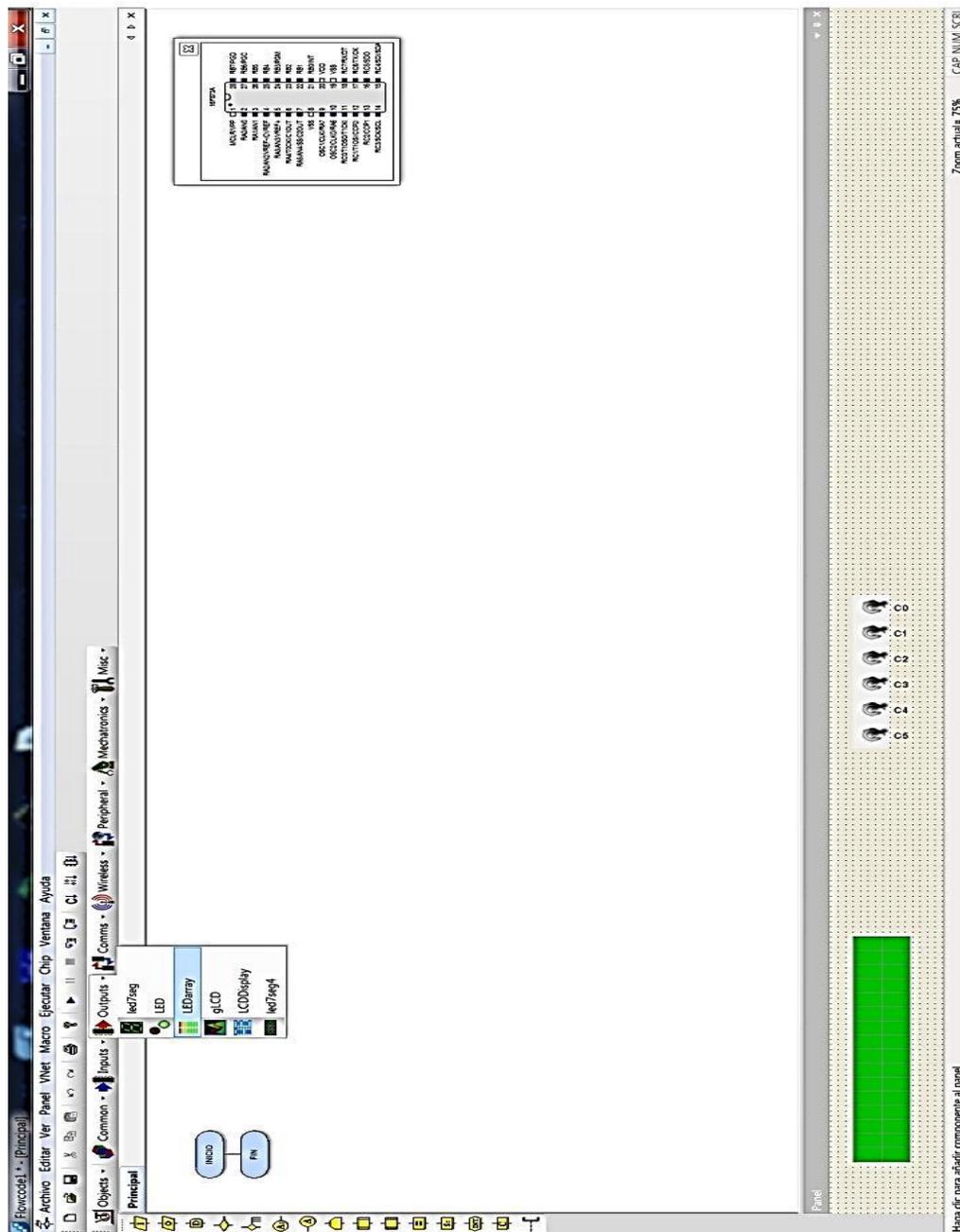


Desplegamos el menú contextual en el display y seleccionamos Conexiones.

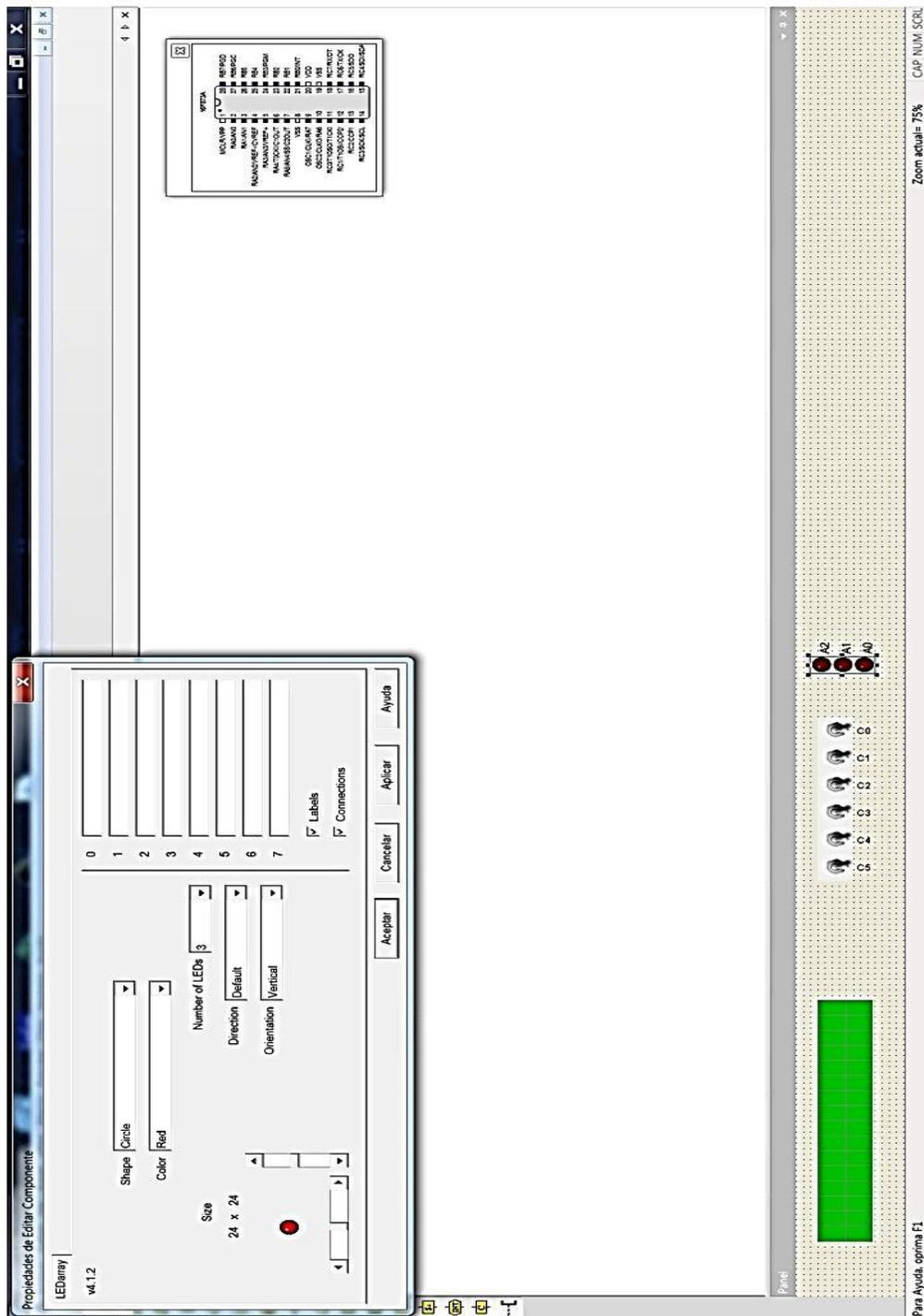


Se abre la ventana de conexiones del componente donde seleccionamos al puerto B del microcontrolador, donde Data 1 le corresponde a RB4, Data 2 a RB5, Data 3 a RB6, Data 4 a RB7, RS a RB1 y E a RB2.

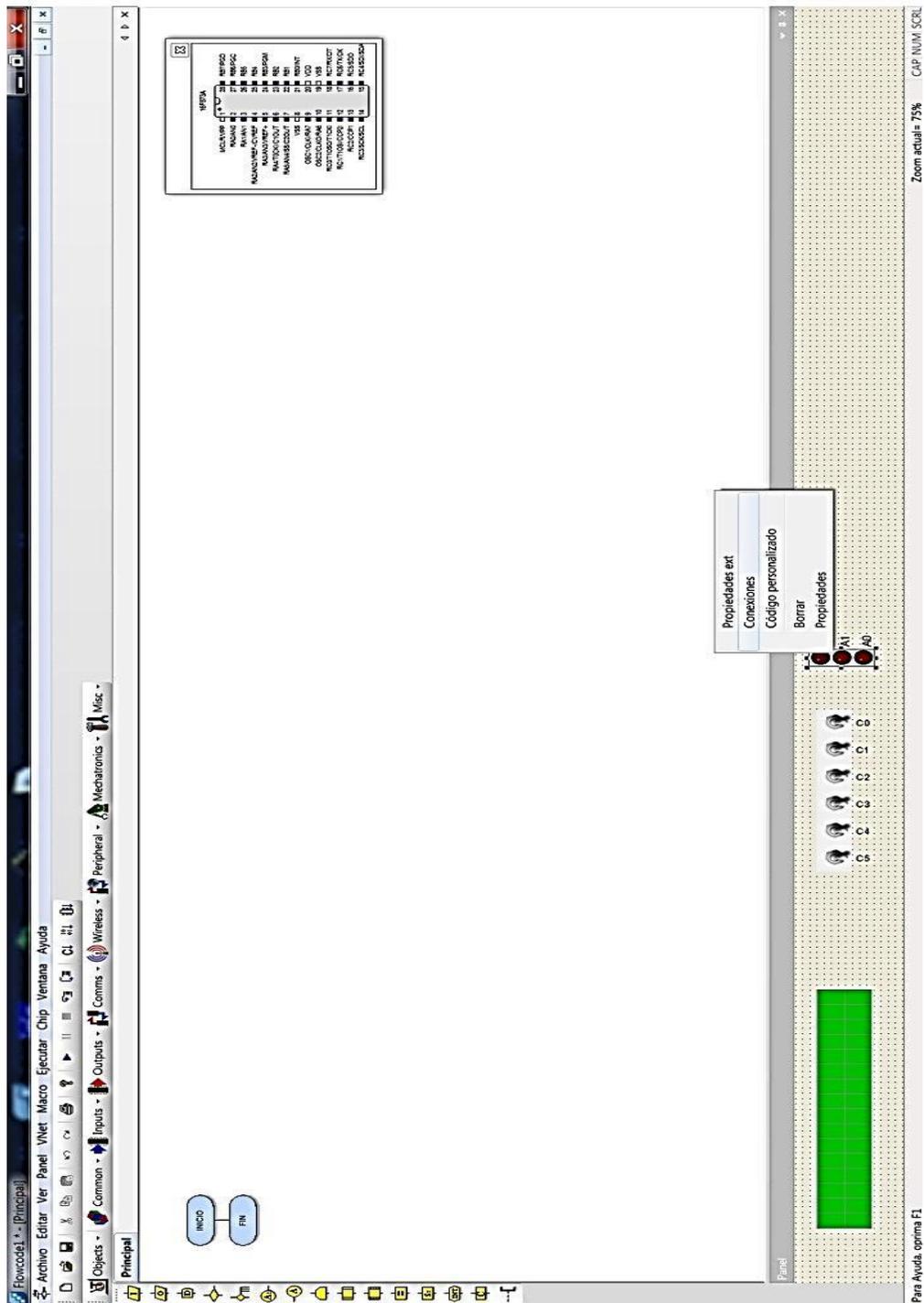
CONEXIÓN Y PROPIEDADES EXTERNAS DE LOS INDICADORES LED



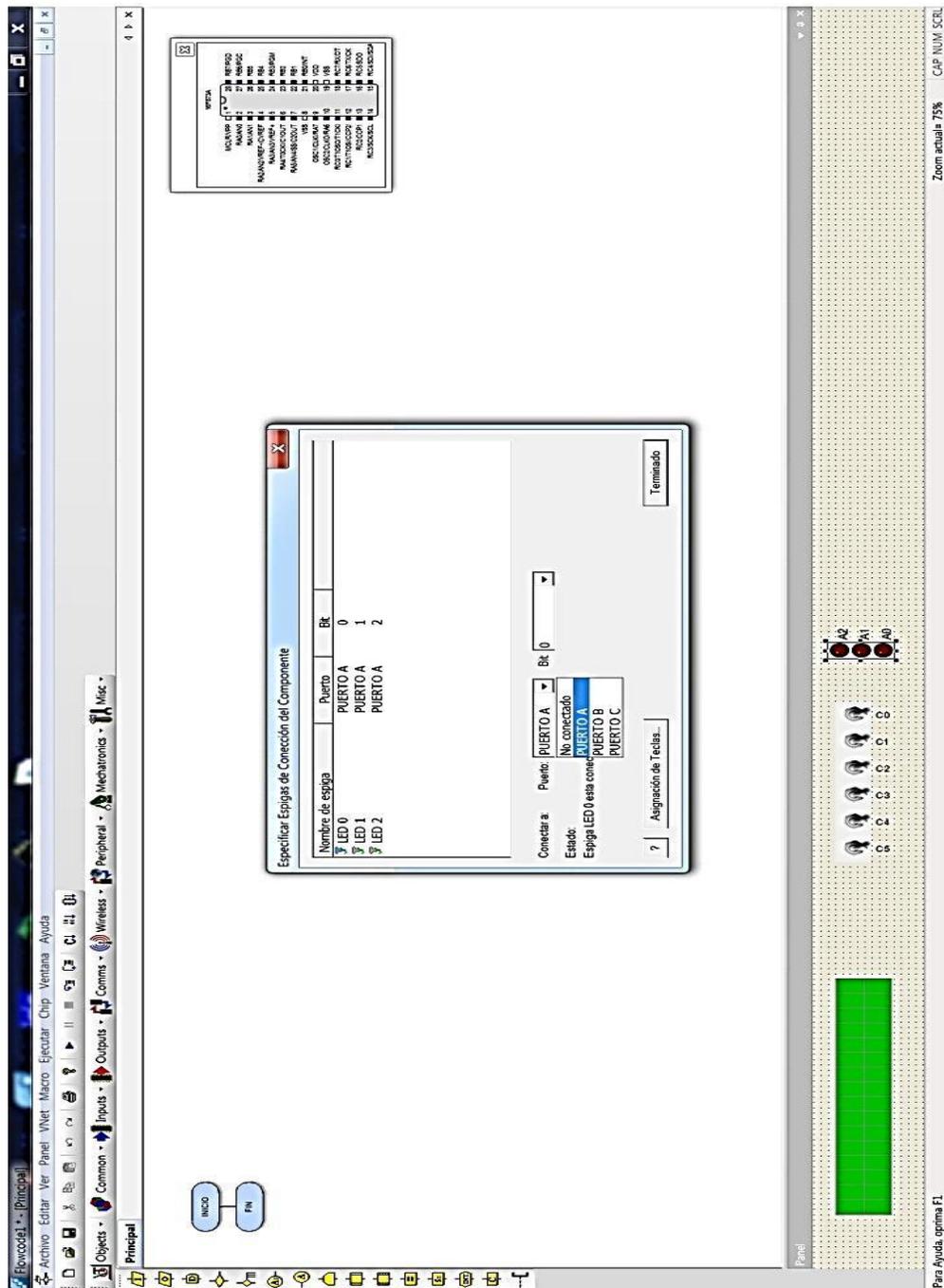
Desde la herramienta de componentes, desplegamos el menú Outputs y seleccionamos LEDarray, inmediatamente se colocan los leds en el panel de montaje. Desplegamos el menú contextual en los leds y seleccionamos Propiedades externas.



Se abre la ventana de edición de componente, donde cambiamos el número de leds a 3 con una orientación en vertical.



Desplegamos de nuevo el menú contextual en el componente y seleccionamos Conexiones.



Se abre la ventana de conexión del componente donde elegimos al puerto A del microcontrolador y asignamos al led indicador de la alarma a RA0, al led indicador de la bomba en uso a RA1 y al led indicador de la electroválvula en uso a RA2.

LAS MACROS DE COMPONENTES Y DEL PROGRAMA

Una macro es una subrutina, están creadas para ahorrar tiempo de programación pero sobre todo mucho código de programa. El programa principal puede llamar a una macro tantas veces sea necesario, sin tener que reescribir el código de las instrucciones que la componen.

En el caso del display LCD, el programa Flowcode ya dispone de macros para este componente, como por ejemplo la macro "Start" que inicializa la pantalla, la macro "Clear" que borra los caracteres visualizados en la pantalla, o la macro PrintASCII que imprime caracteres alfanuméricos sobre la pantalla. Estos son algunos pero no todos, existen más macros para otros componentes.

Sistema de trasvase de agua - (Principal)

Archivo Editar Ver Panel VNet Macro Ejecutar Chip Ventana Ayuda

Objects Common Inputs Outputs Comms Wireless Peripheral Mechatronics Misc

FALLA Principal

Propiedades: Macro de Componente

Mostrar Nombre: Inicializa la pantalla LCD

Componente: LEDarray(0), LCDDisplay(0), SWITCHbank(0), LED(0)

Macro: Start, Clear, PrintASCII, Command, Cursor

Parámetros:

Valor de Retorno:

OK Anular

HPE51A			
INCLAVVP	1	38	RETPGQ
RADANO	2	27	RE6PGQ
RAIANI	3	26	RE5
RADANVREF	4	25	RE4
RADANVREF	5	24	RE3PM
RAITODUCIOUT	6	23	RE2
RASANSICOUT	7	22	RE1
VSS	8	21	RE0INT
OSCCLKAV	9	20	VDD
OSCCLKAV	10	19	VSS
ACPT1000T100	11	18	RC1TRVCT
RC1T00-COP2	12	17	RC5TUXK
RC3COP1	13	16	RC5SD0
RC5CKBGL	14	15	RC4SD0P

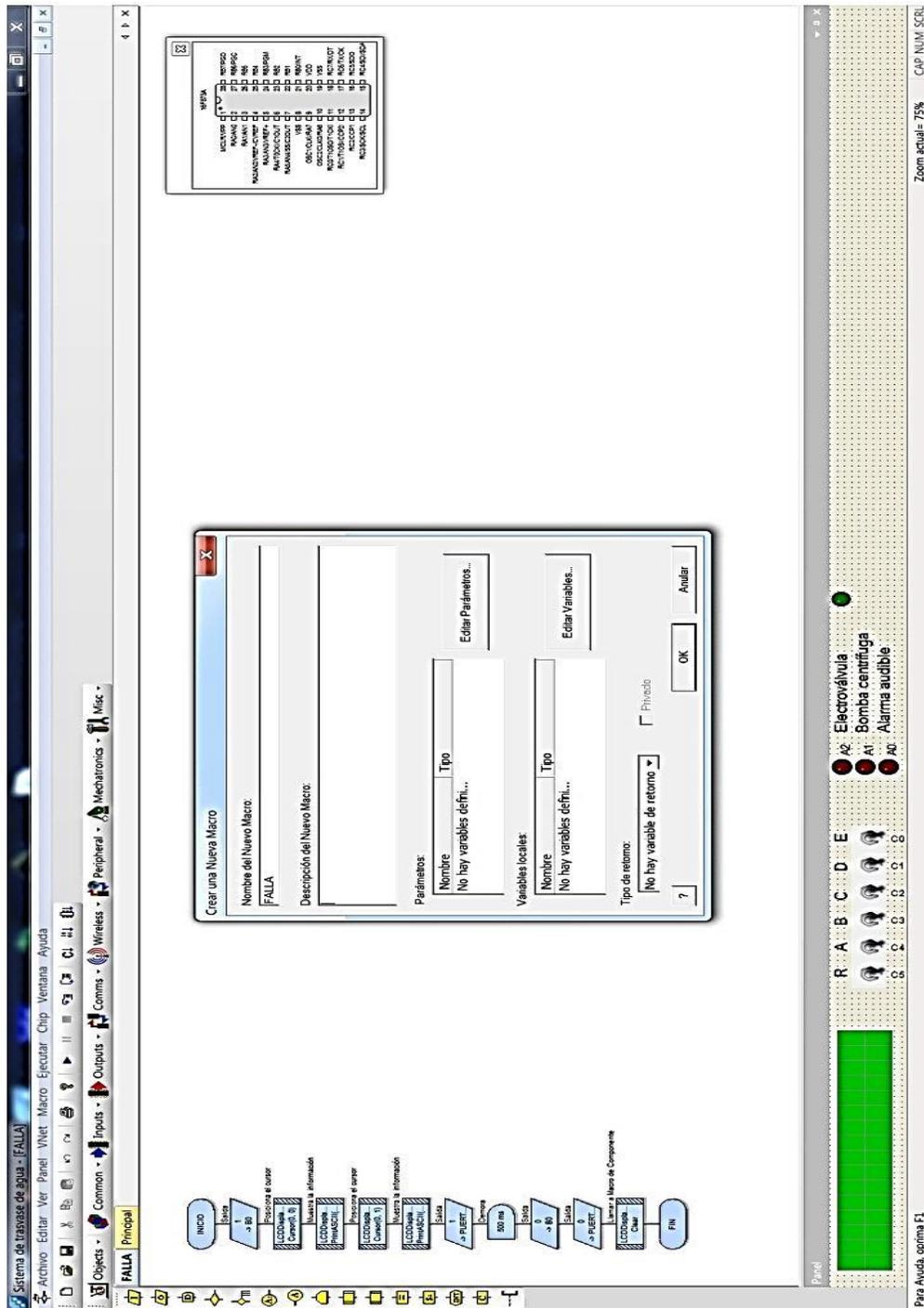
Panel

R A B C D E

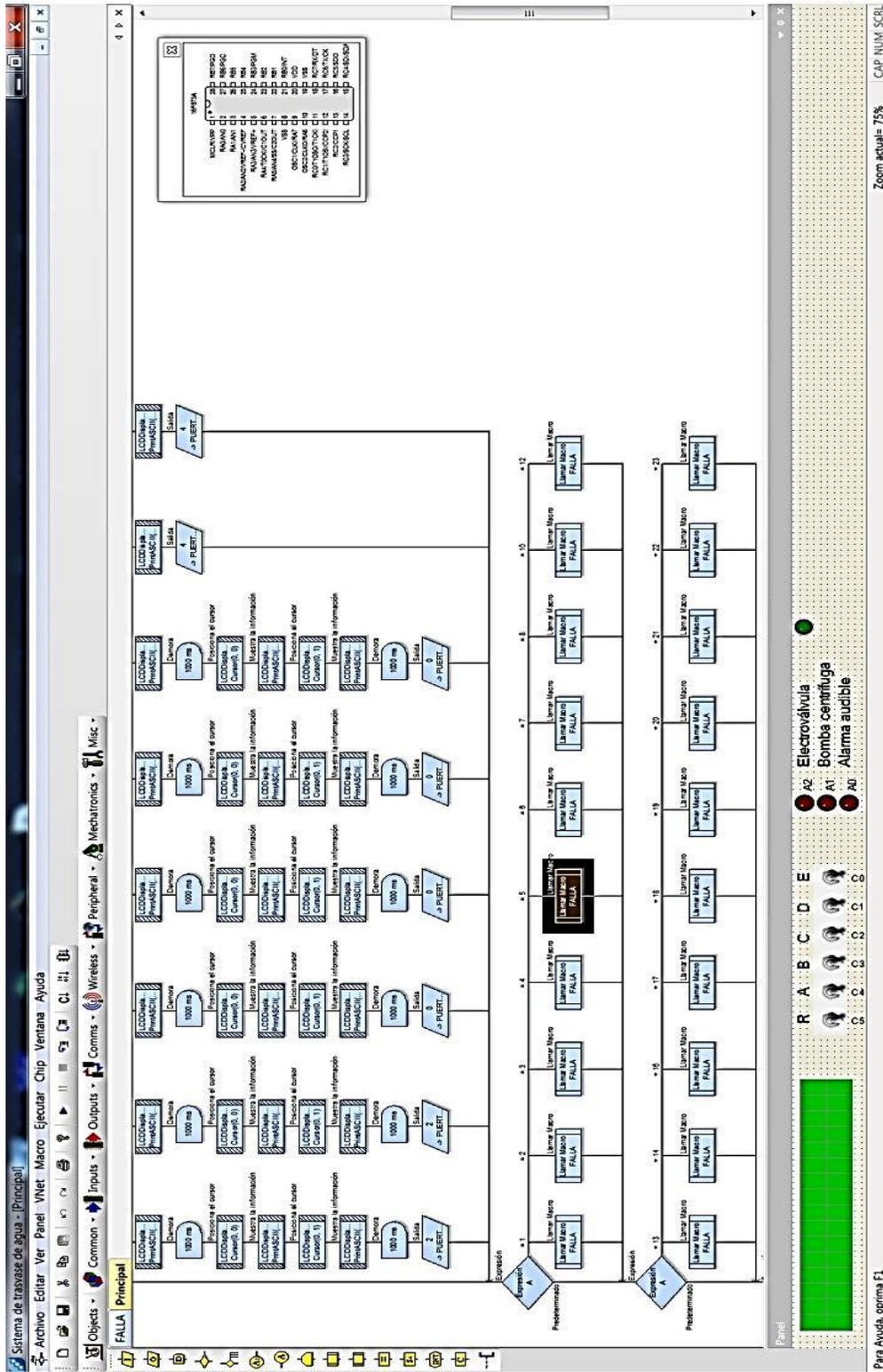
- A2: Electroválvula
- A1: Bomba centrífuga
- A0: Alarma audible

Para Ayuda. oprima F1

Zoom actual= 75% CAP NUM SCRL

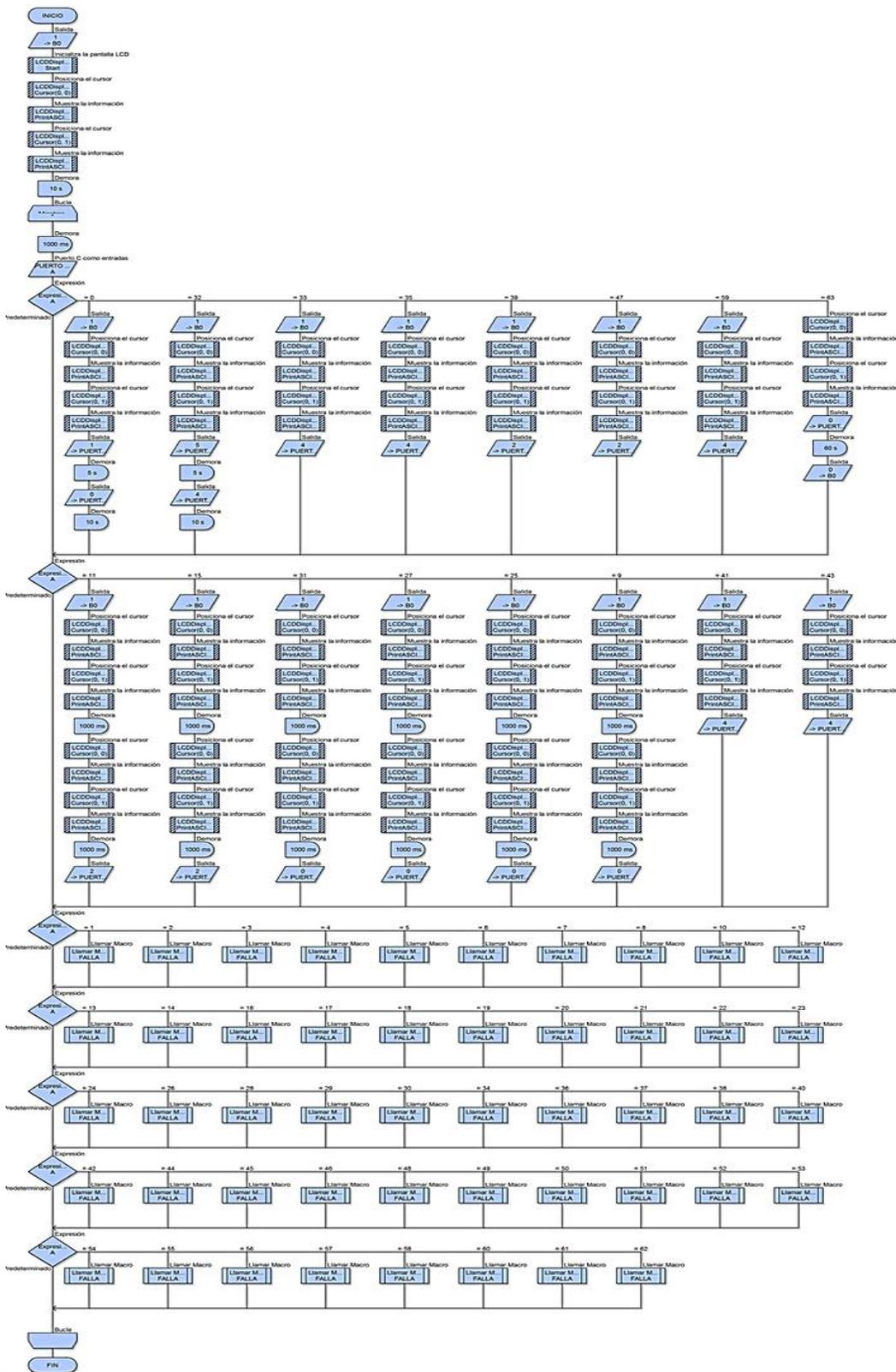


También se puede crear macros, en nuestro caso se creó una macro llamada Falla que el programa principal llamara según corresponda.



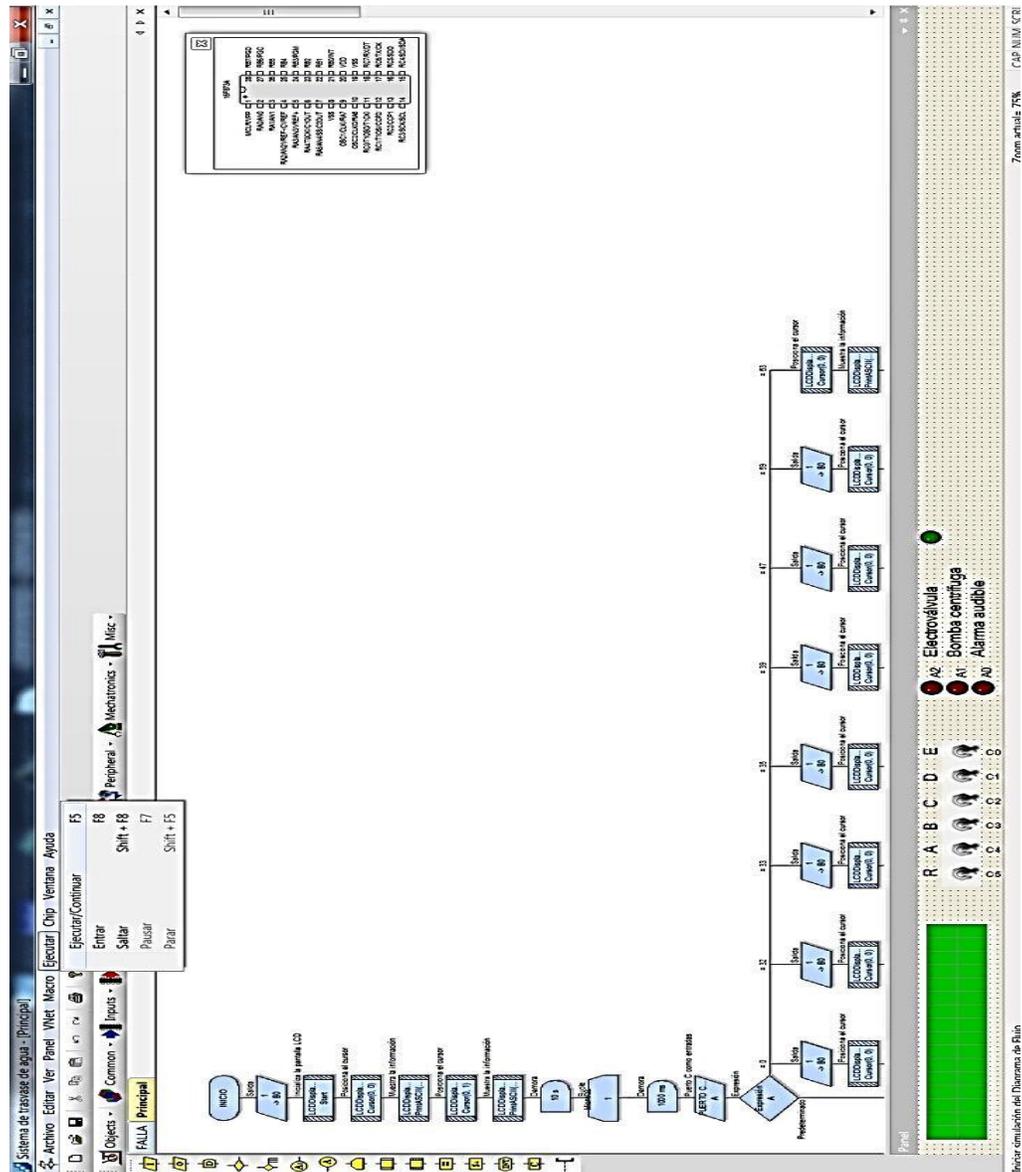
EL DIAGRAMA DE FLUJO DEL CONTROL AUTOMÁTICO

En la siguiente figura se muestra el diagrama de flujo completo del programa principal y lleva la misma lógica que el diagrama de flujo descrito en el capítulo 1 de este trabajo, solo que ahora es creado bajo el entorno de programación Flowcode, el cual se detalló para mostrar correctamente y según corresponda, los mensajes en el display LCD, también se incorporaron demoras para realizar el efecto de intermitencia en los led indicadores, así como la incorporación de las llamadas a macros correspondientes al display para su correcta inicialización y la macro creada llamada “Falla” para el uso de la alarma audible.

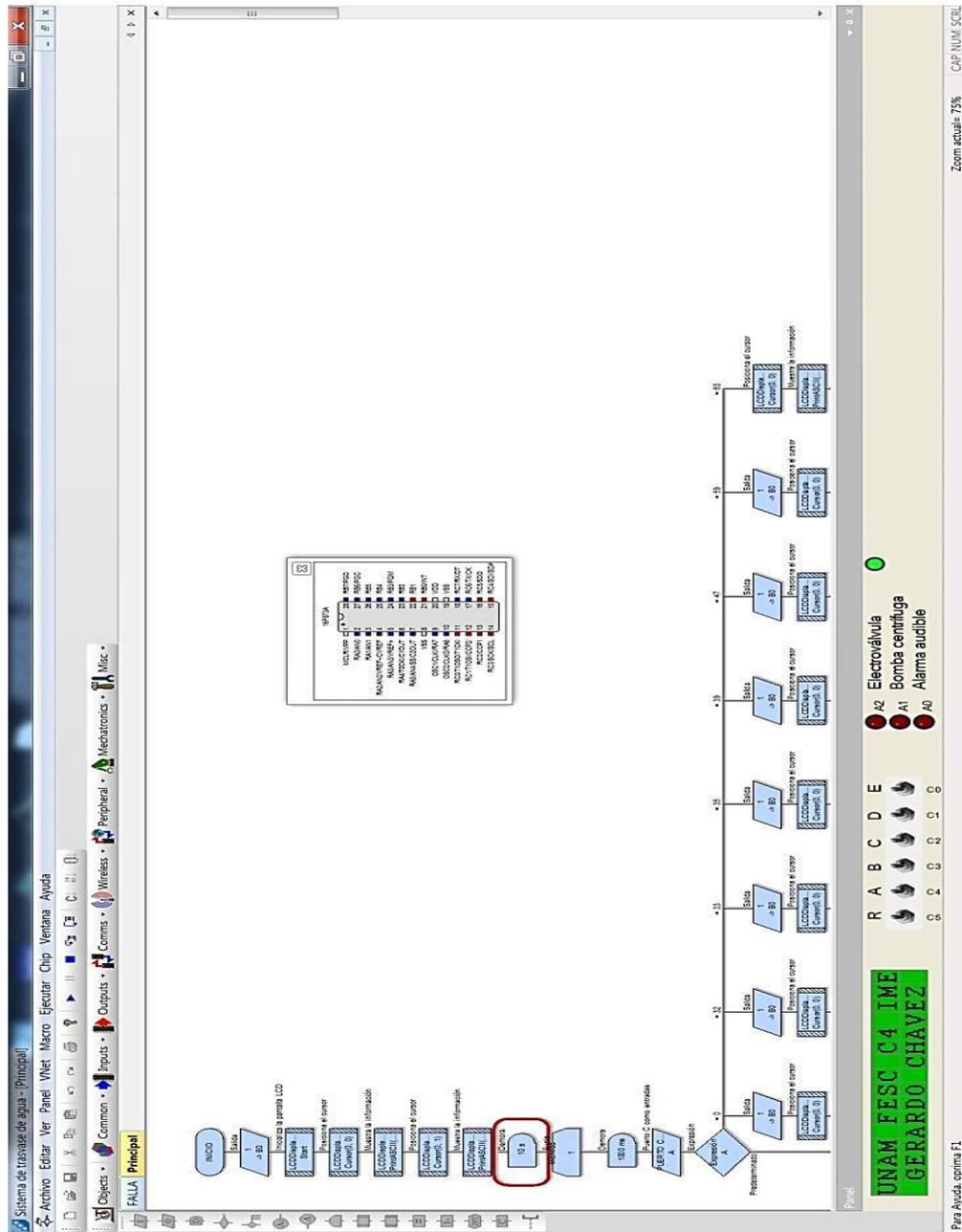


LA SIMULACIÓN DEL PROGRAMA

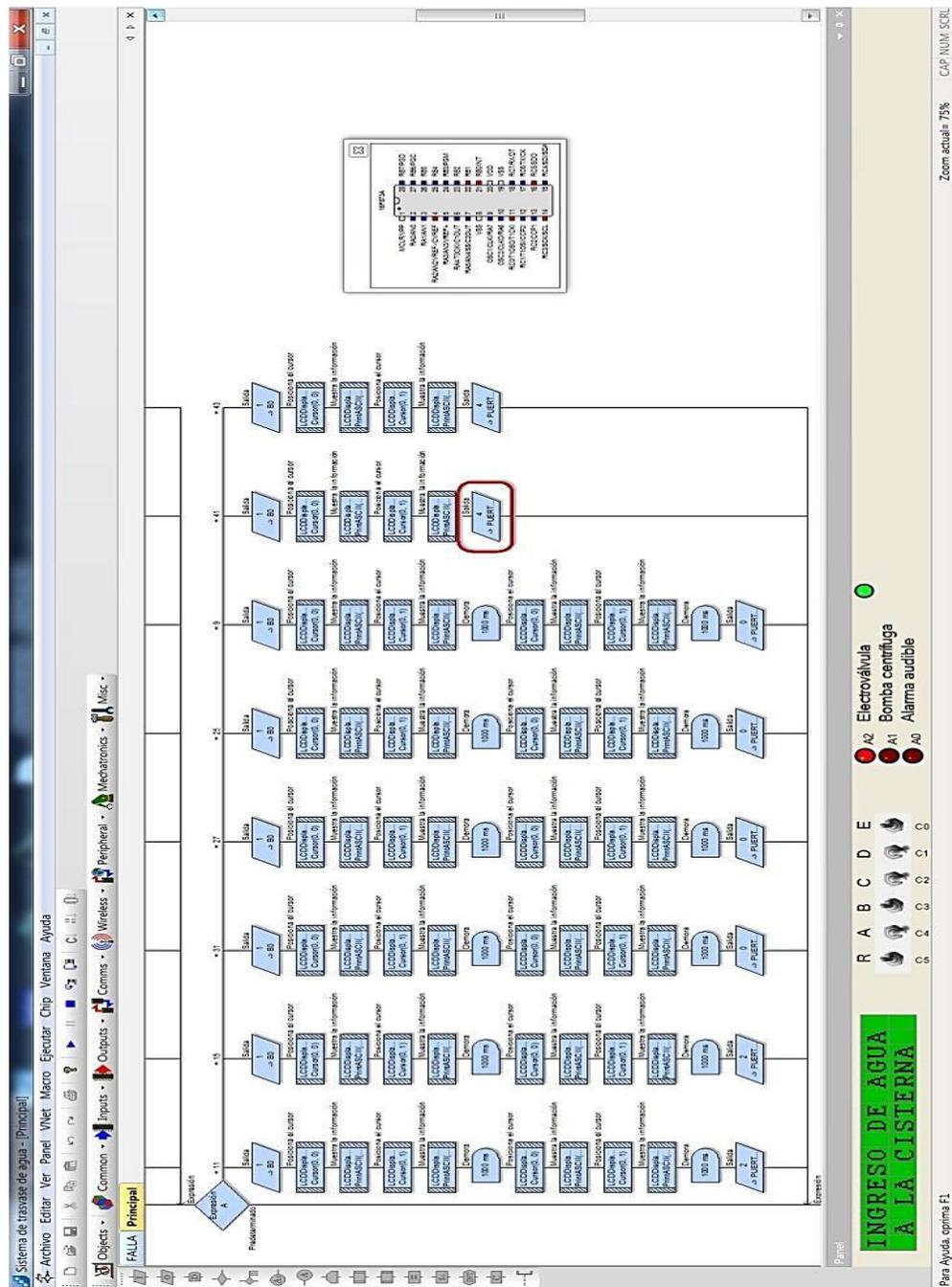
La simulación nos permite comprobar si nuestro programa se ejecuta correctamente. Una vez creado el diagrama de flujo, Flowcode puede simularlo de dos formas, la primera es ejecutar la simulación continuamente o ejecutarla paso a paso.



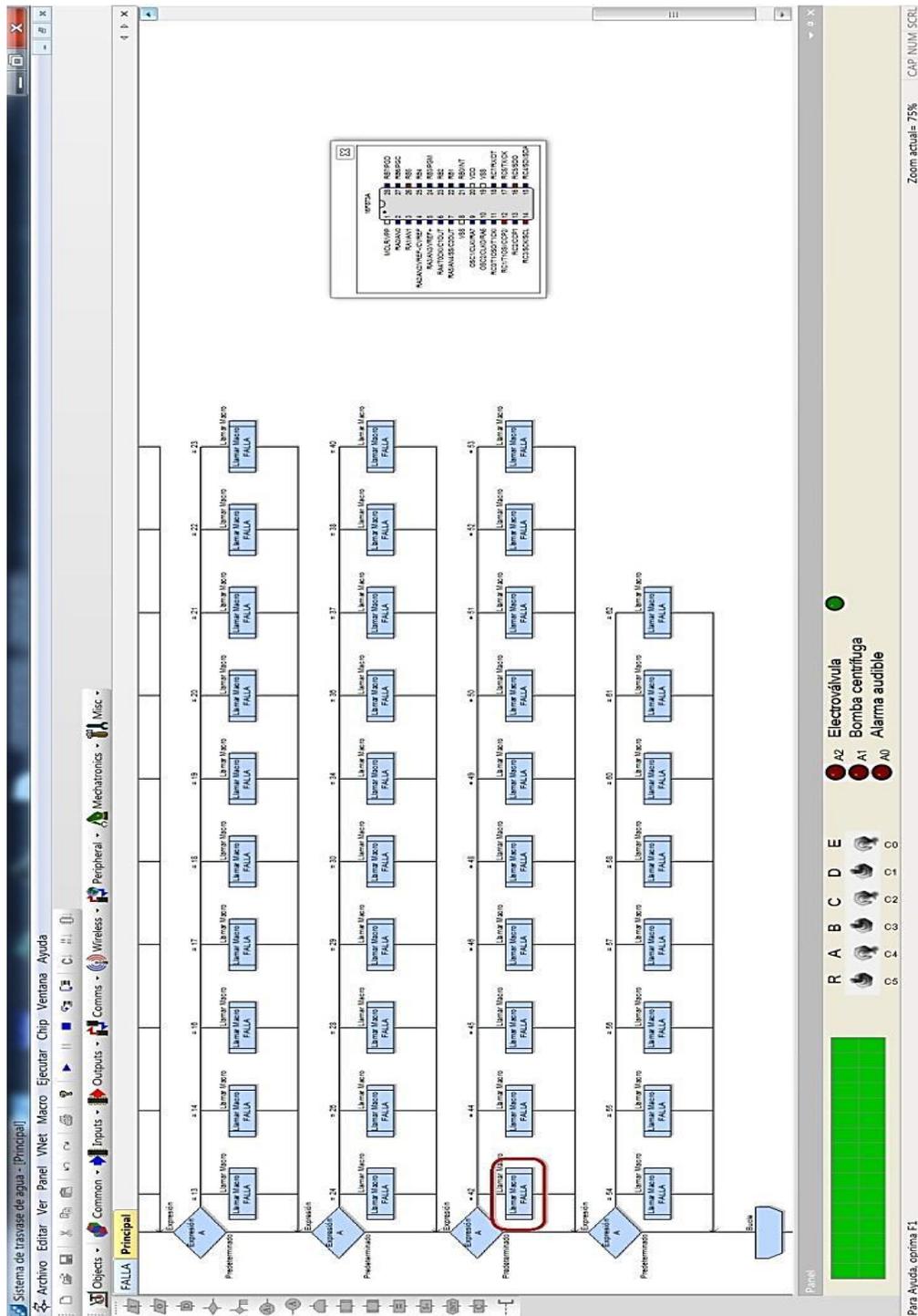
Se despliega el menú “Ejecutar” desde el menú principal y se selecciona “Ejecutar / continuar”, también se puede elegir esta opción con la tecla F5 del teclado de la PC.



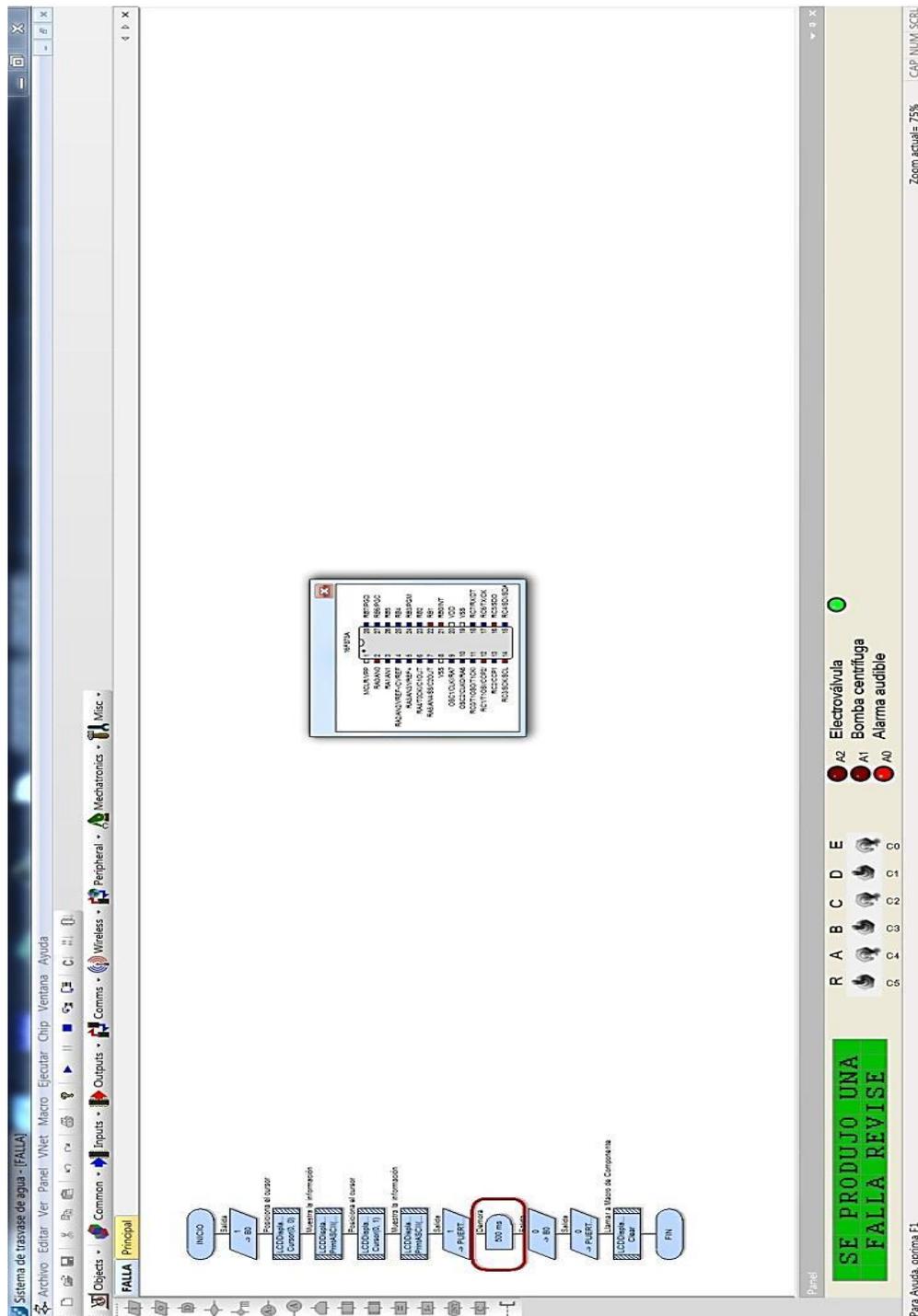
Cuando se está simulando, un rectángulo rojo señala cada instrucción que se está ejecutando en ese momento, también se visualiza los pines del microcontrolador en color rojo cuando su estado es 1 lógico es decir en estado lógico alto, en azul cuando se encuentra en estado lógico bajo y en gris que significa que no se están usando esos pines.



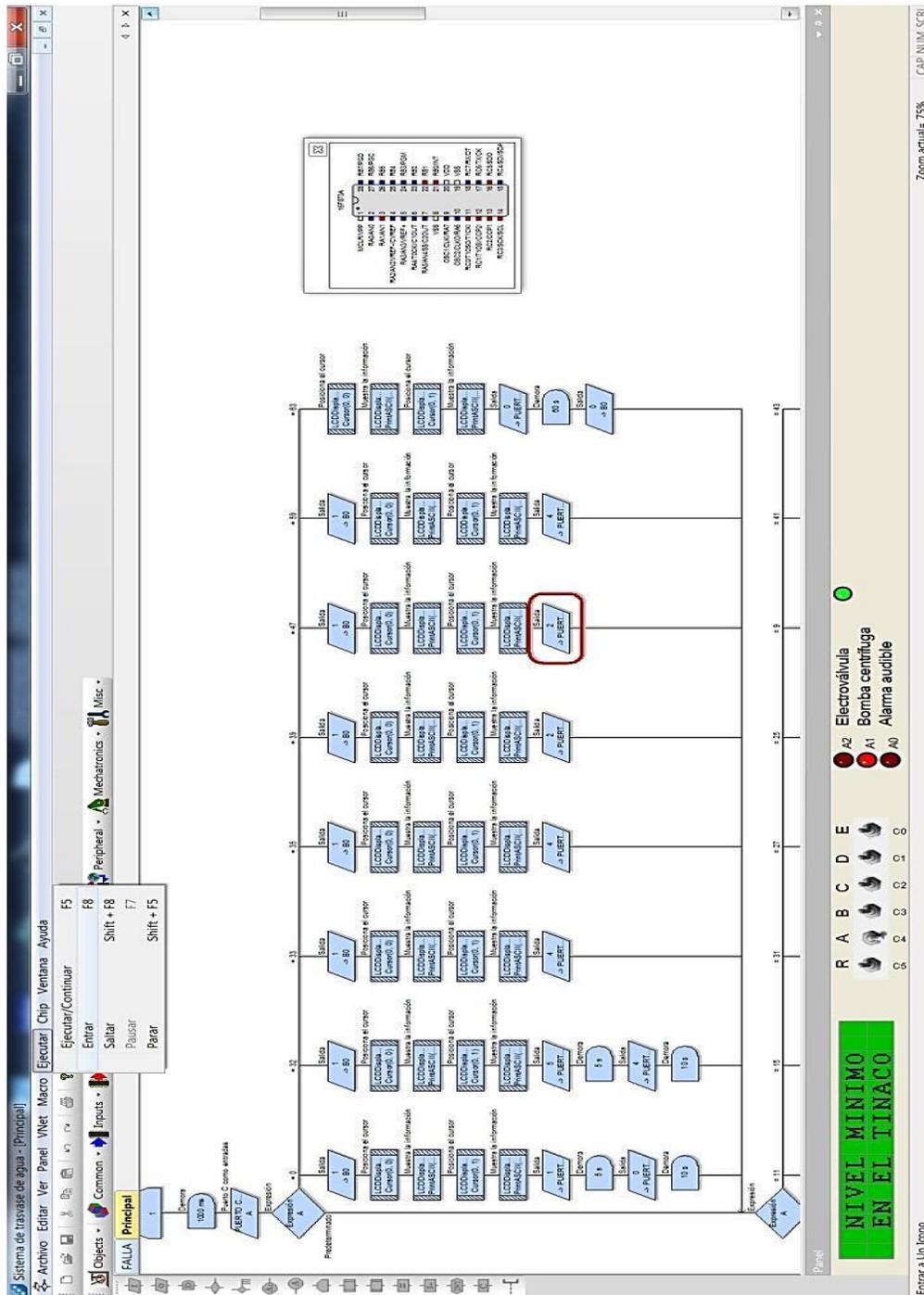
En esta imagen se aprecia un ejemplo de la simulación, donde la electroválvula esta activa y se muestra en el display LCD su mensaje correspondiente, esto es así dependiendo de la combinación que se tenga en la entrada del microcontrolador por medio de los interruptores.



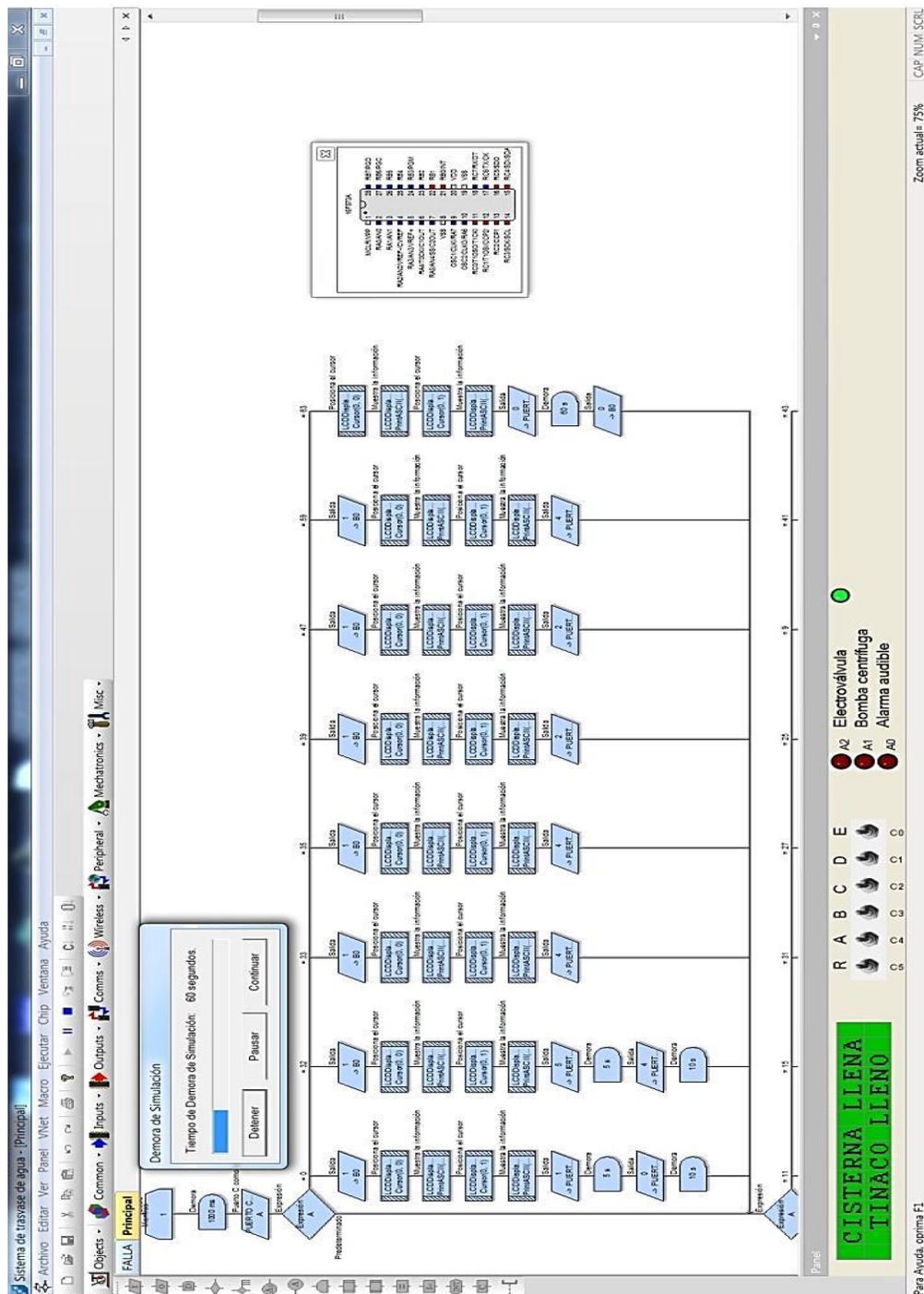
En este otro ejemplo se simuló haber una incongruencia en la combinación de los interruptores, lo que hace que el programa principal llame a la macro “Falla”.



Cuando es ejecutada la macro "Falla", esta misma hace que se active la alarma audible y que se muestre el mensaje intermitente en la pantalla.



En este otro ejemplo, la simulación se hizo paso por paso desplegando el menú “Ejecutar” del menú principal y seleccionando “Entrar” o simplemente con la tecla F8 del teclado de la PC.

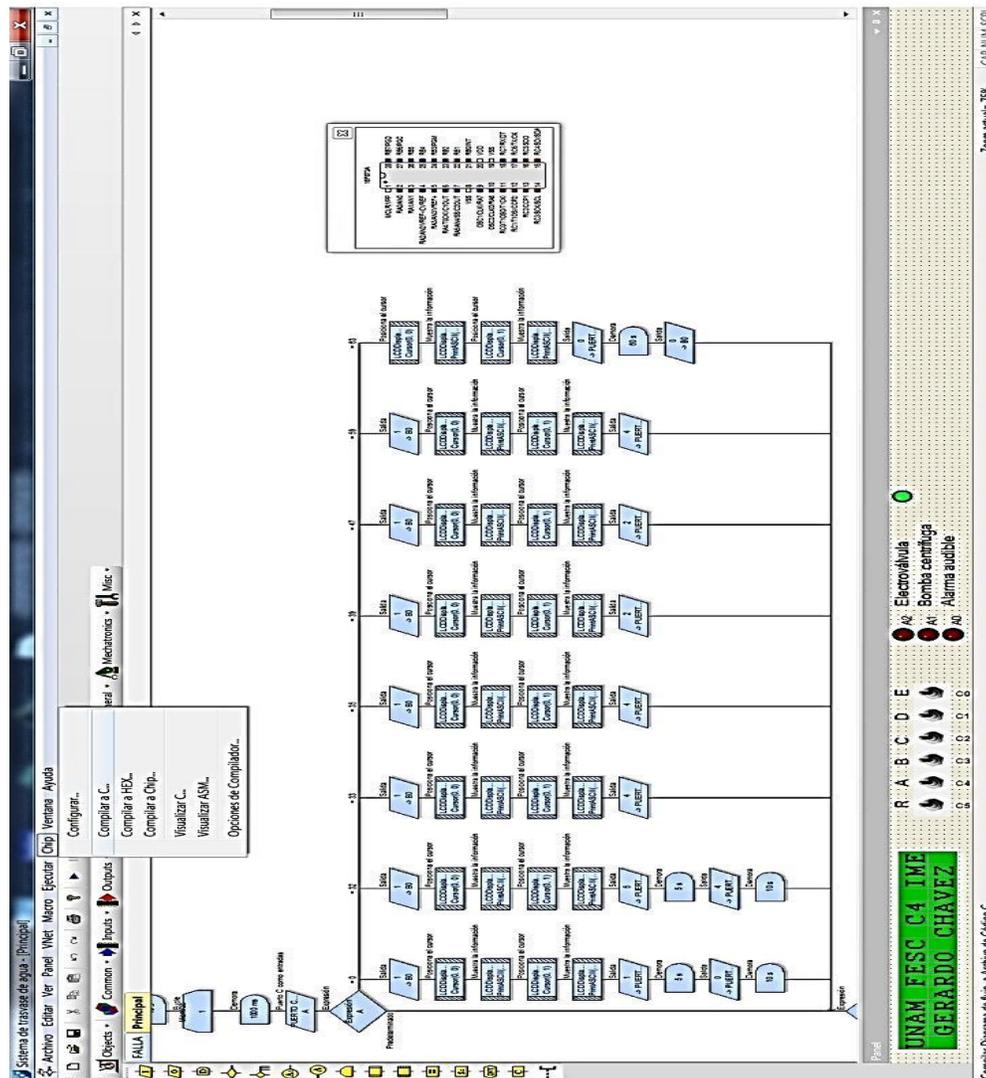


Cabe señalar que cuando se ejecuta una demora, una ventana se abre dónde nos da la opción de detener, pausar la demora o la opción de seguir con la ejecución de la siguiente instrucción.

LA COMPILACIÓN DEL PROGRAMA

La compilación es el proceso en el cual se traducen las instrucciones creadas por un lenguaje de programación en este caso lenguaje Flowcode a un lenguaje máquina.

La compilación convertirá todas las instrucciones del diagrama de flujo a un código como el C o directamente generará un archivo hexadecimal con todas las instrucciones traducidas que se necesita para que el microcontrolador “entienda” lo que va a ejecutar.



Para compilar el diagrama de flujo se despliega la opción “Chip” y seleccionamos compilación a C.

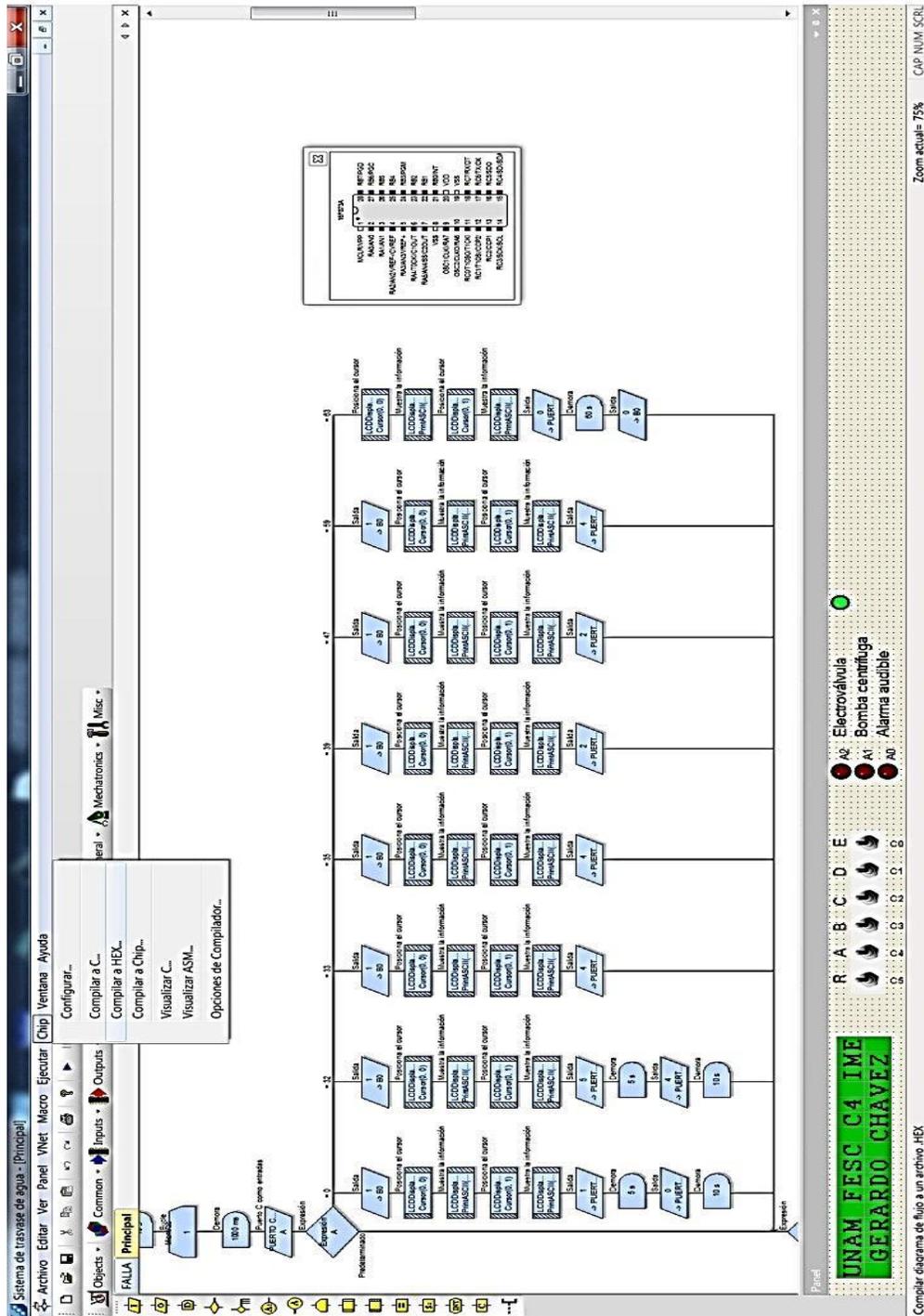
El diagrama de flujo ha sido exitosamente compilado a
C:\Users\Gerardo Chavez\Documents\Tesis\08 - Capitulo
3\Programa pic flowcode\Sistema de traspase de agua.c

UNAM FESC C4 IME
GERARDO CHAVEZ

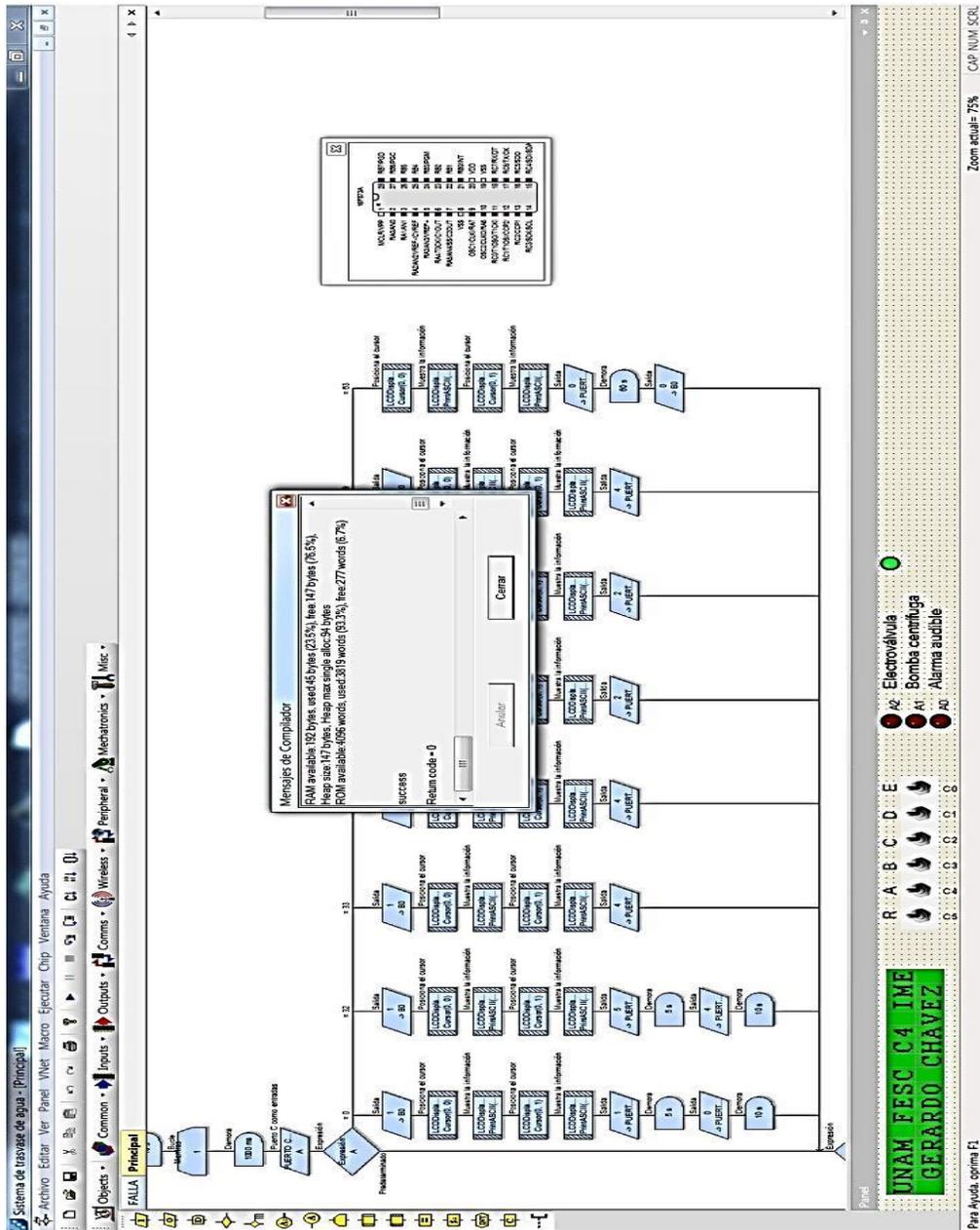
Electroválvula
Bomba centrífuga
Alarma audible

Zoom actual=75% CAP. NUM. SCRL

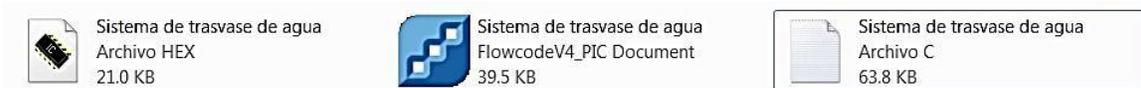
Se abre una ventana donde nos avisa que el diagrama de flujo es correctamente compilado a lenguaje C creándonos un archivo en el disco duro de la computadora.



Para compilar el diagrama de flujo a Hexadecimal, se procede de la misma forma solo que ahora seleccionamos la opción Compilar a HEX.



Se abre una ventana donde se muestra el progreso de la compilación, si no existe ningún error, nos creará un archivo hexadecimal en el disco duro de la computadora, yo lo llamé con el nombre de Sistema de trasvase de agua como se muestra en la siguiente imagen.



CÓDIGO EN LENGUAJE C

A continuación se muestra el código completo generado por la compilación a lenguaje C en Flowcode. Cabe destacar que programar un microcontrolador no es sencillo, se requiere conocimientos de programación avanzados para realizar este programa pero la programación en Flowcode a base de diagramas de flujo, nos facilita crear programas complejos y la reducción en la escritura del código y por consiguiente menos probabilidad de errores ya que se puede simular y corregir antes de compilarlo.

```
/**
*****
** Nombre del archivo: Sistema de trasvase de agua.c
** Fecha: Domingo, 19 de Agosto, 2012 20:13:03
** Realizado por: Gerardo Chávez García
*****
#define MX_PIC
//Definiciones para el microcontrolador
#define P16F873A
#define MX_EE
#define MX_EE_SIZE 128
#define MX_SPI
#define MX_SPI_C
#define MX_SPI_SDI 4
#define MX_SPI_SDO 5
#define MX_SPI_SCK 3
#define MX_UART
#define MX_UART_C
#define MX_UART_TX 6
#define MX_UART_RX 7
#define MX_I2C
#define MX_I2C_C
#define MX_I2C_SDA 4
#define MX_I2C_SCL 3
#define MX_PWM
#define MX_PWM_CNT 2
#define MX_PWM_TRIS1 trisc
#define MX_PWM_1 2
#define MX_PWM_TRIS2 trisc
#define MX_PWM_2 1
//Funciones
#define MX_CLK_SPEED 4000000
#if def _BOOSTC
#include <system.h>
#endif
#if def HI_TECH_C
#include <pic.h>
#endif
//Datos de configuración
#if def _BOOSTC
#pragma DATA 0x2007, 0x3fb2
#endif
#if def HI_TECH_C
__CONFIG(0x3fb2);
#endif
//Funciones internas
#include "C:\Program Files\Matrix Multimedia\Flowcode V4\FCD\internals.h"
//Macro function declarations
void FCM_FALLA();
```

```
//Variable declarations

char FCV_A;
//Defines:
/**** Macro Substitutions ****
portb = D1 Port
trisb = D1 Data Direction
portb = D2 Port
trisb = D2 Data Direction
portb = D3 Port
trisb = D3 Data Direction
portb = D4 Port
trisb = D4 Data Direction
portb = RS Port
trisb = RS Data Direction
portb = E Port
trisb = E Data Direction
4 = Data 1_Pin
5 = Data 2_Pin
6 = Data 3_Pin
7 = Data 4_Pin
1 = RS Pin
2 = Enable Pin
LCD_66260 = Unique Component Reference Number
2 = Row Count
16 = Column Count
*****/

//component connections
#define LCD_66260_PORT0 portb
#define LCD_66260_TRIS0 trisb
#define LCD_66260_PORT1 portb
#define LCD_66260_TRIS1 trisb
#define LCD_66260_PORT2 portb
#define LCD_66260_TRIS2 trisb
#define LCD_66260_PORT3 portb
#define LCD_66260_TRIS3 trisb
#define LCD_66260_PORT4 portb
#define LCD_66260_TRIS4 trisb
#define LCD_66260_PORT5 portb
#define LCD_66260_TRIS5 trisb
#define LCD_66260_BIT0 4
#define LCD_66260_BIT1 5
#define LCD_66260_BIT2 6
#define LCD_66260_BIT3 7
#define LCD_66260_RS 1
#define LCD_66260_E 2
#define LCD_66260_ROWCNT 2
#define LCD_66260_COLCNT 16
#if def _BOOSTC
#define LCD_66260_DELAY delay_10us(10)
#endif
#if def _C2C_
#define LCD_66260_DELAY delay_us(100)
#endif
#if def HI_TECH_C
#define LCD_66260_DELAY __delay_us(120)
#endif
#if ndef LCD_66260_DELAY
#define LCD_66260_DELAY delay_us(100)
#endif

//LCDDisplay0: //Macro function declarations
void FCD_LCDDisplay0_RawSend(char in, char mask);
void FCD_LCDDisplay0_Start();
void FCD_LCDDisplay0_Clear();
```

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN CAMPO 4

```
void FCD_LCDDisplay0_PrintASCII(char Character);
void FCD_LCDDisplay0_Command(char in);
void FCD_LCDDisplay0_Cursor(char x, char y);
void FCD_LCDDisplay0_PrintNumber(short Number);
void FCD_LCDDisplay0_PrintString(char* String, char MSZ_String);

void FCD_LCDDisplay0_ScrollDisplay(char Direction, char Num_Positions);
void FCD_LCDDisplay0_ClearLine(char Line);
void FCD_LCDDisplay0_RAM_Write(char nIdx, char d0, char d1, char d2, char d3, char d4, char d5, char d6, char d7);
//LCDDisplay0: //Macro implementations
void FCD_LCDDisplay0_RawSend(char in, char mask)
{
    unsigned char pt;
    clear_bit(LCD_66260_PORT0, LCD_66260_BIT0);
    clear_bit(LCD_66260_PORT1, LCD_66260_BIT1);
    clear_bit(LCD_66260_PORT2, LCD_66260_BIT2);
    clear_bit(LCD_66260_PORT3, LCD_66260_BIT3);
    clear_bit(LCD_66260_PORT4, LCD_66260_RS);
    clear_bit(LCD_66260_PORT5, LCD_66260_E);
    pt = ((in >> 4) & 0x0f);
    if (pt & 0x01)
        set_bit(LCD_66260_PORT0, LCD_66260_BIT0);
    if (pt & 0x02)
        set_bit(LCD_66260_PORT1, LCD_66260_BIT1);
    if (pt & 0x04)
        set_bit(LCD_66260_PORT2, LCD_66260_BIT2);
    if (pt & 0x08)
        set_bit(LCD_66260_PORT3, LCD_66260_BIT3);
    if (mask)
        set_bit(LCD_66260_PORT4, LCD_66260_RS);
    LCD_66260_DELAY;
    set_bit(LCD_66260_PORT5, LCD_66260_E);
    LCD_66260_DELAY;
    clear_bit(LCD_66260_PORT5, LCD_66260_E);
    pt = (in & 0x0f);
    LCD_66260_DELAY;
    clear_bit(LCD_66260_PORT0, LCD_66260_BIT0);
    clear_bit(LCD_66260_PORT1, LCD_66260_BIT1);
    clear_bit(LCD_66260_PORT2, LCD_66260_BIT2);
    clear_bit(LCD_66260_PORT3, LCD_66260_BIT3);
    clear_bit(LCD_66260_PORT4, LCD_66260_RS);
    clear_bit(LCD_66260_PORT5, LCD_66260_E);
    if (pt & 0x01)
        set_bit(LCD_66260_PORT0, LCD_66260_BIT0);
    if (pt & 0x02)
        set_bit(LCD_66260_PORT1, LCD_66260_BIT1);
    if (pt & 0x04)
        set_bit(LCD_66260_PORT2, LCD_66260_BIT2);
    if (pt & 0x08)
        set_bit(LCD_66260_PORT3, LCD_66260_BIT3);
    if (mask)
        set_bit(LCD_66260_PORT4, LCD_66260_RS);
    LCD_66260_DELAY;
    set_bit(LCD_66260_PORT5, LCD_66260_E);
    LCD_66260_DELAY;
    clear_bit(LCD_66260_PORT5, LCD_66260_E);
    LCD_66260_DELAY;
}
void FCD_LCDDisplay0_Start()
{
    clear_bit(LCD_66260_TRIS0, LCD_66260_BIT0);
    clear_bit(LCD_66260_TRIS1, LCD_66260_BIT1);
    clear_bit(LCD_66260_TRIS2, LCD_66260_BIT2);
    clear_bit(LCD_66260_TRIS3, LCD_66260_BIT3);
    clear_bit(LCD_66260_TRIS4, LCD_66260_RS);
```

```
clear_bit(LCD_66260_TRIS5, LCD_66260_E);
Wdt_Delay_Ms(12);
FCD_LCDDisplay0_RawSend(0x33, 0);
Wdt_Delay_Ms(2);
FCD_LCDDisplay0_RawSend(0x33, 0);
Wdt_Delay_Ms(2);
FCD_LCDDisplay0_RawSend(0x32, 0);
Wdt_Delay_Ms(2);

FCD_LCDDisplay0_RawSend(0x2c, 0);
Wdt_Delay_Ms(2);
FCD_LCDDisplay0_RawSend(0x06, 0);
Wdt_Delay_Ms(2);
FCD_LCDDisplay0_RawSend(0x0c, 0);
Wdt_Delay_Ms(2);
//clear the display
FCD_LCDDisplay0_RawSend(0x01, 0);
Wdt_Delay_Ms(2);
FCD_LCDDisplay0_RawSend(0x02, 0);
Wdt_Delay_Ms(2);
}
void FCD_LCDDisplay0_Clear()
{
    FCD_LCDDisplay0_RawSend(0x01, 0);
    Wdt_Delay_Ms(2);
    FCD_LCDDisplay0_RawSend(0x02, 0);
    Wdt_Delay_Ms(2);
}
void FCD_LCDDisplay0_PrintASCII(char Character)
{
    FCD_LCDDisplay0_RawSend(Character, 0x10);
}
void FCD_LCDDisplay0_Command(char in)
{
    FCD_LCDDisplay0_RawSend(in, 0);
    Wdt_Delay_Ms(2);
}
void FCD_LCDDisplay0_Cursor(char x, char y)
{
    #if (LCD_66260_ROWCNT == 1)
        y=0x80;
    #endif
    #if (LCD_66260_ROWCNT == 2)
        if (y==0)
            y=0x80;
        else
            y=0xc0;
    #endif
    #if (LCD_66260_ROWCNT == 4)
        if (y==0)
            y=0x80;
    else if (y==1)
        y=0xc0;
        #if (LCD_66260_COLCNT == 16)
            else if (y==2)
                y=0x90;
            else
                y=0xd0;
        #endif
        #if (LCD_66260_COLCNT == 20)
            else if (y==2)
                y=0x94;
            else
                y=0xd4;
        #endif
    #endif
}
```

```
#endif
    FCD_LCDDisplay0_RawSend(y + x, 0);
    Wdt_Delay_Ms(2);
}
void FCD_LCDDisplay0_PrintNumber(short Number)
{
    short tmp_int;
    char tmp_byte;
    if (Number < 0)
    {
        FCD_LCDDisplay0_RawSend('-', 0x10);

        Number = 0 - Number;
    }
    tmp_int = Number;
    if (Number >= 10000)
    {
        tmp_byte = tmp_int / 10000;
        FCD_LCDDisplay0_RawSend('0' + tmp_byte, 0x10);
        while (tmp_byte > 0)
        {
            tmp_int = tmp_int - 10000;
            tmp_byte--;
        }
    }
    if (Number >= 1000)
    {
        tmp_byte = tmp_int / 1000;
        FCD_LCDDisplay0_RawSend('0' + tmp_byte, 0x10);
        while (tmp_byte > 0)
        {
            tmp_int = tmp_int - 1000;
            tmp_byte--;
        }
    }
    if (Number >= 100)
    {
        tmp_byte = tmp_int / 100;
        FCD_LCDDisplay0_RawSend('0' + tmp_byte, 0x10);
        while (tmp_byte > 0)
        {
            tmp_int = tmp_int - 100;
            tmp_byte--;
        }
    }
    if (Number >= 10)
    {
        tmp_byte = tmp_int / 10;
        FCD_LCDDisplay0_RawSend('0' + tmp_byte, 0x10);
        while (tmp_byte > 0)
        {
            tmp_int = tmp_int - 10;
            tmp_byte--;
        }
    }
    FCD_LCDDisplay0_RawSend('0' + tmp_int, 0x10);
}
void FCD_LCDDisplay0_PrintString(char* String, char MSZ_String)
{
    char idx = 0;
    for (idx=0; idx<MSZ_String; idx++)
    {
        #if def _BOOSTC
            if (String[idx] == 0)
            {

```

```
                break;
            }
            FCD_LCDDisplay0_RawSend(String[idx], 0x10);
        #endif
        #if def HI_TECH_C
            if (*String == 0)
            {
                break;
            }
            FCD_LCDDisplay0_RawSend(*String, 0x10);
            String++;
        #endif
    }
}
```

```
void FCD_LCDDisplay0_ScrollDisplay(char Direction, char Num_Positions)
{
    char cmd = 0;
    char count;
    //Choose the direction
    switch (Direction)
    {
        case 0:
        case '!':
        case 'L':
            cmd = 0x18;
            break;

        case 1:
        case 'r':
        case 'R':
            cmd = 0x1C;
            break;

        default:
            break;
    }
    //If direction accepted then scroll the specified amount
    if (cmd)
    {
        for (count = 0; count < Num_Positions; count++)
            FCD_LCDDisplay0_Command(cmd);
    }
}
```

```
void FCD_LCDDisplay0_ClearLine(char Line)
{
    char count;
    char rowcount;
    //Define number of columns per line
    #if (LCD_66260_ROW_CNT == 1)
        rowcount=80;
    #endif
    #if (LCD_66260_ROW_CNT == 2)
        rowcount=40;
    #endif
    #if (LCD_66260_ROW_CNT == 4)
        #if (LCD_66260_COL_CNT == 16)
            rowcount=16;
        #endif
        #if (LCD_66260_COL_CNT == 20)
            rowcount=20;
        #endif
    #endif
    #endif
    //Start at beginning of the line
    FCD_LCDDisplay0_Cursor(0, Line);
    //Send out spaces to clear line
    for (count = 0; count < rowcount; count++)
```

```
        FCD_LCDDisplay0_RawSend(' ', 0x10);
        //Move back to the beginning of the line.
        FCD_LCDDisplay0_Cursor(0, Line);
    }
void FCD_LCDDisplay0_RAM_Write(char nIdx, char d0, char d1, char d2, char d3, char d4, char d5, char d6, char d7)
{
    //set CGRAM address
    FCD_LCDDisplay0_RawSend(64 + (nIdx << 3), 0);
    delay_ms(2);
    //write CGRAM data
    FCD_LCDDisplay0_RawSend(d0, 0x10);
    FCD_LCDDisplay0_RawSend(d1, 0x10);
    FCD_LCDDisplay0_RawSend(d2, 0x10);
    FCD_LCDDisplay0_RawSend(d3, 0x10);
    FCD_LCDDisplay0_RawSend(d4, 0x10);
    FCD_LCDDisplay0_RawSend(d5, 0x10);
    FCD_LCDDisplay0_RawSend(d6, 0x10);

    FCD_LCDDisplay0_RawSend(d7, 0x10);
    //Clear the display
    FCD_LCDDisplay0_RawSend(0x01, 0);
    delay_ms(2);
    FCD_LCDDisplay0_RawSend(0x02, 0);
    delay_ms(2);
}
//Macro implementations
void FCM_FALLA()
{
    //Salida
    //Salida: 1 -> B0
    trisb = trisb & 0xfe;
    if (1)
        portb = (portb & 0xfe) | 0x01;
    else
        portb = portb & 0xfe;
    //Posiciona el cursor
    //Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 0)
    FCD_LCDDisplay0_Cursor(0, 0);
    //Muestra la información
    //Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" SE PRODUJO UNA ")
    FCD_LCDDisplay0_PrintASCII(' ');
    FCD_LCDDisplay0_PrintASCII('S');
    FCD_LCDDisplay0_PrintASCII('E');
    FCD_LCDDisplay0_PrintASCII(' ');
    FCD_LCDDisplay0_PrintASCII('P');
    FCD_LCDDisplay0_PrintASCII('R');
    FCD_LCDDisplay0_PrintASCII('O');
    FCD_LCDDisplay0_PrintASCII('D');
    FCD_LCDDisplay0_PrintASCII('U');
    FCD_LCDDisplay0_PrintASCII('J');
    FCD_LCDDisplay0_PrintASCII('O');
    FCD_LCDDisplay0_PrintASCII(' ');
    FCD_LCDDisplay0_PrintASCII('U');
    FCD_LCDDisplay0_PrintASCII('N');
    FCD_LCDDisplay0_PrintASCII('A');
    FCD_LCDDisplay0_PrintASCII(' ');
    //Posiciona el cursor
    //Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 1)
    FCD_LCDDisplay0_Cursor(0, 1);
    //Muestra la información
    //Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" FALLA REVISE ")
    FCD_LCDDisplay0_PrintASCII(' ');
    FCD_LCDDisplay0_PrintASCII(' ');
    FCD_LCDDisplay0_PrintASCII('F');
    FCD_LCDDisplay0_PrintASCII('A');
```

```
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('V');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
//Salida
//Salida: 1 -> PUERTO A
trisa = trisa & 0xf8;
porta = (porta & 0xf8) | (1 & 0x07);
//Demora
//Demora: 500 ms
delay_ms(255);
delay_ms(245);

//Salida
//Salida: 0 -> B0
trisb = trisb & 0xfe;
if (0)
    portb = (portb & 0xfe) | 0x01;
else
    portb = portb & 0xfe;

//Salida
//Salida: 0 -> PUERTO A
trisa = trisa & 0xf8;
porta = (porta & 0xf8) | (0 & 0x07);
//Llamar a Macro de Componente
//Llamar a Macro de Componente: LCDDisplay(0)::Clear
FCD_LCDDisplay0_Clear();
}
void main()
{
    //Inicialización
    adcon1 = 0x07;
    //Inicialización del Código de Interrupción
    option_reg = 0xC0;
    //Salida
    //Salida: 1 -> B0
    trisb = trisb & 0xfe;
    if (1)
        portb = (portb & 0xfe) | 0x01;
    else
        portb = portb & 0xfe;
    //Inicializa la pantalla LCD
    //Llamar a Macro de Componente: LCDDisplay(0)::Start
    FCD_LCDDisplay0_Start();
    //Posiciona el cursor
    //Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 0)
    FCD_LCDDisplay0_Cursor(0, 0);
    //Muestra la información
    //Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII("UNAM FESC C4 IME")
    FCD_LCDDisplay0_PrintASCII('U');
    FCD_LCDDisplay0_PrintASCII('N');
    FCD_LCDDisplay0_PrintASCII('A');
    FCD_LCDDisplay0_PrintASCII('M');
    FCD_LCDDisplay0_PrintASCII(' ');
    FCD_LCDDisplay0_PrintASCII('F');
    FCD_LCDDisplay0_PrintASCII('E');
```

```
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('C');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('C');
FCD_LCDDisplay0_PrintASCII('4');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('M');
FCD_LCDDisplay0_PrintASCII('E');
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 1)
FCD_LCDDisplay0_Cursor(0, 1);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" GERARDO CHAVEZ ")
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('G');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('D');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII(' ');

FCD_LCDDisplay0_PrintASCII('C');
FCD_LCDDisplay0_PrintASCII('H');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('V');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('Z');
FCD_LCDDisplay0_PrintASCII(' ');
//Demora
//Demora: 10 s
delay_s(10);
//Bucle
//Bucle: Mientras 1
while (1)
{
    //Demora
    //Demora: 1000 ms
    delay_ms(255);
    delay_ms(255);
    delay_ms(255);
    delay_ms(235);
    //Puerto C como entradas
    //Ingreso: PUERTO C -> A
    trisc = trisc | 0x3f;
    FCV_A = portc & 0x3f;
    //Expresión
    //Expresión: A?
    switch (FCV_A)
    {
        case 0:
        {
            //Salida
            //Salida: 1 -> B0
            trisb = trisb & 0xfe;
            if (1)
                portb = (portb & 0xfe) | 0x01;
            else
                portb = portb & 0xfe;
            //Posiciona el cursor
            //Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 0)
            FCD_LCDDisplay0_Cursor(0, 0);
            //Muestra la información
```

```
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII("TRASVASE DE AGUA")
FCD_LCDDisplay0_PrintASCII('T');
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('V');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('D');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('G');
FCD_LCDDisplay0_PrintASCII('U');
FCD_LCDDisplay0_PrintASCII('A');
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 1)
FCD_LCDDisplay0_Cursor(0, 1);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII("SIN RECURSOS... ")
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII(' ');

FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('C');
FCD_LCDDisplay0_PrintASCII('U');
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('.');
FCD_LCDDisplay0_PrintASCII('.');
FCD_LCDDisplay0_PrintASCII('.');
FCD_LCDDisplay0_PrintASCII(' ');

//Salida
//Salida: 1 -> PUERTO A
trisa = trisa & 0xf8;
porta = (porta & 0xf8) | (1 & 0x07);
//Demora
//Demora: 5 s
delay_s(5);
//Salida
//Salida: 0 -> PUERTO A
trisa = trisa & 0xf8;
porta = (porta & 0xf8) | (0 & 0x07);
//Demora
//Demora: 10 s
delay_s(10);
break;
}
case 32:
{
//Salida
//Salida: 1 -> B0
trib = trib & 0xfe;
if (1)
    portb = (portb & 0xfe) | 0x01;
else
    portb = portb & 0xfe;
//Posiciona el cursor
```

```
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 0)
FCD_LCDDisplay0_Cursor(0, 0);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII("INGRESO DE AGUA ")
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('G');
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('D');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('G');
FCD_LCDDisplay0_PrintASCII('U');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 1)
FCD_LCDDisplay0_Cursor(0, 1);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" A LA CISTERNA ")
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('L');

FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('C');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('T');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
//Salida
//Salida: 5 -> PUERTO A
trisa = trisa & 0xf8;
porta = (porta & 0xf8) | (5 & 0x07);
//Demora
//Demora: 5 s
delay_(5);
//Salida
//Salida: 4 -> PUERTO A
trisa = trisa & 0xf8;
porta = (porta & 0xf8) | (4 & 0x07);
//Demora
//Demora: 10 s
delay_(10);
break;
}
case 33:
{
//Salida
//Salida: 1 -> B0
trisb = trisb & 0xfe;
if (1)
portb = (portb & 0xfe) | 0x01;
```

```
else
    portb = portb & 0xfe;
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 0)
FCD_LCDDisplay0_Cursor(0, 0);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" NIVEL MINIMO ")
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('V');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('M');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('M');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 1)
FCD_LCDDisplay0_Cursor(0, 1);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" EN LA CISTERNA ")
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII(' ');

FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('C');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('T');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
//Salida
//Salida: 4 -> PUERTO A
trisa = trisa & 0xf8;
porta = (porta & 0xf8) | (4 & 0x07);
break;
}
case 35:
{
//Salida
//Salida: 1 -> B0
trisb = trisb & 0xfe;
if (1)
    portb = (portb & 0xfe) | 0x01;
else
    portb = portb & 0xfe;
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 0)
FCD_LCDDisplay0_Cursor(0, 0);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII("RESERVA COMPLETA")
```

```
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('V');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('C');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII('M');
FCD_LCDDisplay0_PrintASCII('P');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('T');
FCD_LCDDisplay0_PrintASCII('A');
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 1)
FCD_LCDDisplay0_Cursor(0, 1);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" EN LA CISTERNA ")
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('C');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('T');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('N');

FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
//Salida
//Salida: 4 -> PUERTO A
trisa = trisa & 0xf8;
porta = (porta & 0xf8) | (4 & 0x07);
break;
}
case 39:
{
//Salida
//Salida: 1 -> B0
trisb = trisb & 0xfe;
if (1)
    portb = (portb & 0xfe) | 0x01;
else
    portb = portb & 0xfe;
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 0)
FCD_LCDDisplay0_Cursor(0, 0);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" CISTERNA LLENA ")
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('C');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('T');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('R');
```

```
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 1)
FCD_LCDDisplay0_Cursor(0, 1);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" LLENANDO TINACO")
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('D');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('T');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('C');
FCD_LCDDisplay0_PrintASCII('O');
//Salida
//Salida: 2 -> PUERTO A
trisa = trisa & 0xf8;
porta = (porta & 0xf8) | (2 & 0x07);
break;
}
case 47:
{

//Salida
//Salida: 1 -> B0
tribb = tribb & 0xfe;
if (1)
    portb = (portb & 0xfe) | 0x01;
else
    portb = portb & 0xfe;
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 0)
FCD_LCDDisplay0_Cursor(0, 0);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" NIVEL MINIMO ")
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');

FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('V');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('M');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('M');
```

```
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 1)
FCD_LCDDisplay0_Cursor(0, 1);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" EN EL TINACO ")
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('T');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('C');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
//Salida
//Salida: 2 -> PUERTO A
trisa = trisa & 0xf8;
porta = (porta & 0xf8) | (2 & 0x07);
break;
}
case 59:
{
//Salida
//Salida: 1 -> B0
trib = trib & 0xfe;
if (1)
    portb = (portb & 0xfe) | 0x01;
else
    portb = portb & 0xfe;
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 0)

FCD_LCDDisplay0_Cursor(0, 0);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" TINACO LLENO ")
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('T');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('C');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 1)
FCD_LCDDisplay0_Cursor(0, 1);
//Muestra la información
```

```
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII("INGRESANDO AGUA ")
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('G');
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('D');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('G');
FCD_LCDDisplay0_PrintASCII('U');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
//Salida
//Salida: 4 -> PUERTO A
trisa = trisa & 0xf8;
porta = (porta & 0xf8) | (4 & 0x07);
break;
}
case 63:
{
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 0)
FCD_LCDDisplay0_Cursor(0, 0);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" CISTERNA LLENA ")
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('C');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('T');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('N');

FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 1)
FCD_LCDDisplay0_Cursor(0, 1);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" TINACO LLENO ")
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('T');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('C');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('N');
```

```
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
//Salida
//Salida: 0 -> PUERTO A
trisa = trisa & 0xf8;
porta = (porta & 0xf8) | (0 & 0x07);
//Demora
//Demora: 60 s
delay_s(60);
//Salida
//Salida: 0 -> B0
trisb = trisb & 0xfe;
if (0)
    portb = (portb & 0xfe) | 0x01;
else
    portb = portb & 0xfe;
break;
}
}
//Expresión
//Expresión: A?
switch (FCV_A)
{
    case 11:
    {
        //Salida
        //Salida: 1 -> B0
        trisb = trisb & 0xfe;
        if (1)
            portb = (portb & 0xfe) | 0x01;
        else
            portb = portb & 0xfe;
        //Posiciona el cursor
        //Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 0)
        FCD_LCDDisplay0_Cursor(0, 0);
        //Muestra la información
        //Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" SIN SUMINISTRO ")
        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII('S');
        FCD_LCDDisplay0_PrintASCII('I');
        FCD_LCDDisplay0_PrintASCII('N');
        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII('S');
        FCD_LCDDisplay0_PrintASCII('U');
        FCD_LCDDisplay0_PrintASCII('M');
        FCD_LCDDisplay0_PrintASCII('I');

        FCD_LCDDisplay0_PrintASCII('N');
        FCD_LCDDisplay0_PrintASCII('I');
        FCD_LCDDisplay0_PrintASCII('S');
        FCD_LCDDisplay0_PrintASCII('T');
        FCD_LCDDisplay0_PrintASCII('R');
        FCD_LCDDisplay0_PrintASCII('O');
        FCD_LCDDisplay0_PrintASCII(' ');
        //Posiciona el cursor
        //Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 1)
        FCD_LCDDisplay0_Cursor(0, 1);
        //Muestra la información
        //Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" DE AGUA ")
        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII('D');
    }
}
```

```
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('G');
FCD_LCDDisplay0_PrintASCII('U');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
//Demora
//Demora: 1000 ms
delay_ms(255);
delay_ms(255);
delay_ms(255);
delay_ms(235);
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 0)
FCD_LCDDisplay0_Cursor(0, 0);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" USO DE LA ")
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('U');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('D');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 1)
FCD_LCDDisplay0_Cursor(0, 1);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII("RESERVA DE AGUA ")
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('V');
FCD_LCDDisplay0_PrintASCII('A');

FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('D');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('G');
FCD_LCDDisplay0_PrintASCII('U');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
//Demora
//Demora: 1000 ms
delay_ms(255);
delay_ms(255);
delay_ms(255);
```

```
        delay_ms(235);
        //Salida
        //Salida: 2 -> PUERTO A
        trisa = trisa & 0xf8;
        porta = (porta & 0xf8) | (2 & 0x07);
        break;
    }
    case 15:
    {
        //Salida
        //Salida: 1 -> B0
        trisb = trisb & 0xfe;
        if (1)
            portb = (portb & 0xfe) | 0x01;
        else
            portb = portb & 0xfe;
        //Posiciona el cursor
        //Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 0)
        FCD_LCDDisplay0_Cursor(0, 0);
        //Muestra la información
        //Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" SIN SUMINISTRO ")
        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII('S');
        FCD_LCDDisplay0_PrintASCII('I');
        FCD_LCDDisplay0_PrintASCII('N');
        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII('S');
        FCD_LCDDisplay0_PrintASCII('U');
        FCD_LCDDisplay0_PrintASCII('M');
        FCD_LCDDisplay0_PrintASCII('I');
        FCD_LCDDisplay0_PrintASCII('N');
        FCD_LCDDisplay0_PrintASCII('I');
        FCD_LCDDisplay0_PrintASCII('S');
        FCD_LCDDisplay0_PrintASCII('T');
        FCD_LCDDisplay0_PrintASCII('R');
        FCD_LCDDisplay0_PrintASCII('O');
        FCD_LCDDisplay0_PrintASCII(' ');
        //Posiciona el cursor
        //Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 1)
        FCD_LCDDisplay0_Cursor(0, 1);
        //Muestra la información
        //Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" DE AGUA  ")
        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII('D');
        FCD_LCDDisplay0_PrintASCII('E');
        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII('A');
        FCD_LCDDisplay0_PrintASCII('G');
        FCD_LCDDisplay0_PrintASCII('U');
        FCD_LCDDisplay0_PrintASCII('A');

        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII(' ');
        //Demora
        //Demora: 1000 ms
        delay_ms(255);
        delay_ms(255);
        delay_ms(255);
        delay_ms(235);
    }
}
```

```
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 0)
FCD_LCDDisplay0_Cursor(0, 0);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII("LLENANDO TINACO ")
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('D');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('T');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('C');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII(' ');
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 1)
FCD_LCDDisplay0_Cursor(0, 1);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII("RESERVA COMPLETA")
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('V');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('C');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII('M');
FCD_LCDDisplay0_PrintASCII('P');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('T');
FCD_LCDDisplay0_PrintASCII('A');
//Demora
//Demora: 1000 ms
delay_ms(255);
delay_ms(255);
delay_ms(255);
delay_ms(235);
//Salida
//Salida: 2 -> PUERTO A
trisa = trisa & 0xf8;
porta = (porta & 0xf8) | (2 & 0x07);
break;
}
case 31:
{

//Salida
//Salida: 1 -> B0
trisb = trisb & 0xfe;
if (1)
    portb = (portb & 0xfe) | 0x01;
else
    portb = portb & 0xfe;
//Posiciona el cursor
```

```
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 0)
FCD_LCDDisplay0_Cursor(0, 0);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" SIN SUMINISTRO ")
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('U');
FCD_LCDDisplay0_PrintASCII('M');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('T');
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII(' ');
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 1)
FCD_LCDDisplay0_Cursor(0, 1);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" DE AGUA ")
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('D');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('G');
FCD_LCDDisplay0_PrintASCII('U');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
//Demora
//Demora: 1000 ms
delay_ms(255);
delay_ms(255);
delay_ms(255);
delay_ms(235);
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 0)
FCD_LCDDisplay0_Cursor(0, 0);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" CISTERNA LLENA ")
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('C');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('T');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('R');

FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('L');
```

```
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 1)
FCD_LCDDisplay0_Cursor(0, 1);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" TINACO LLENO ")
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('T');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('C');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
//Demora
//Demora: 1000 ms
delay_ms(255);
delay_ms(255);
delay_ms(255);
delay_ms(235);
//Salida
//Salida: 0 -> PUERTO A
trisa = trisa & 0xf8;
porta = (porta & 0xf8) | (0 & 0x07);
break;
}
case 27:
{
//Salida
//Salida: 1 -> B0
trib = trib & 0xfe;
if (1)
    portb = (portb & 0xfe) | 0x01;
else
    portb = portb & 0xfe;
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 0)
FCD_LCDDisplay0_Cursor(0, 0);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" SIN SUMINISTRO ")
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('U');
FCD_LCDDisplay0_PrintASCII('M');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('I');

FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('T');
```

```
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII(' ');
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 1)
FCD_LCDDisplay0_Cursor(0, 1);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" DE AGUA ")
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('D');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('G');
FCD_LCDDisplay0_PrintASCII('U');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
//Demora
//Demora: 1000 ms
delay_ms(255);
delay_ms(255);
delay_ms(255);
delay_ms(235);
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 0)
FCD_LCDDisplay0_Cursor(0, 0);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" TINACO LLENO ")
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('T');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('C');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 1)
FCD_LCDDisplay0_Cursor(0, 1);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII("RESERVA COMPLETA")
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('V');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('C');
```

```
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII('M');
FCD_LCDDisplay0_PrintASCII('P');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('T');
FCD_LCDDisplay0_PrintASCII('A');
//Demora
//Demora: 1000 ms
delay_ms(255);
delay_ms(255);
delay_ms(255);
delay_ms(235);
//Salida
//Salida: 0 -> PUERTO A
trisa = trisa & 0xf8;
porta = (porta & 0xf8) | (0 & 0x07);
break;
}
case 25:
{
//Salida
//Salida: 1 -> B0
trib = trib & 0xfe;
if (1)
    portb = (portb & 0xfe) | 0x01;
else
    portb = portb & 0xfe;
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 0)
FCD_LCDDisplay0_Cursor(0, 0);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" SIN SUMINISTRO ")
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('U');
FCD_LCDDisplay0_PrintASCII('M');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('T');
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII(' ');
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 1)
FCD_LCDDisplay0_Cursor(0, 1);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" DE AGUA ")
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('D');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('G');
FCD_LCDDisplay0_PrintASCII('U');
```

```
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');

FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
//Demora
//Demora: 1000 ms
delay_ms(255);
delay_ms(255);
delay_ms(255);
delay_ms(235);
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 0)
FCD_LCDDisplay0_Cursor(0, 0);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" TINACO LLENO ")
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('T');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('C');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 1)
FCD_LCDDisplay0_Cursor(0, 1);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII("RESERVA AGOTADA ")
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('V');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('G');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII('T');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('D');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
//Demora
//Demora: 1000 ms
delay_ms(255);
delay_ms(255);
delay_ms(255);
delay_ms(235);
//Salida
//Salida: 0 -> PUERTO A
trisa = trisa & 0xf8;
porta = (porta & 0xf8) | (0 & 0x07);
```

```
        break;
    }
    case 9:
    {
        //Salida
        //Salida: 1 -> B0

        trisb = trisb & 0xfe;
        if (1)
            portb = (portb & 0xfe) | 0x01;
        else
            portb = portb & 0xfe;
        //Posiciona el cursor
        //Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 0)
        FCD_LCDDisplay0_Cursor(0, 0);
        //Muestra la información
        //Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" SIN SUMINISTRO ")
        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII('S');
        FCD_LCDDisplay0_PrintASCII('I');
        FCD_LCDDisplay0_PrintASCII('N');
        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII('S');
        FCD_LCDDisplay0_PrintASCII('U');
        FCD_LCDDisplay0_PrintASCII('M');
        FCD_LCDDisplay0_PrintASCII('I');
        FCD_LCDDisplay0_PrintASCII('N');
        FCD_LCDDisplay0_PrintASCII('I');
        FCD_LCDDisplay0_PrintASCII('S');
        FCD_LCDDisplay0_PrintASCII('T');
        FCD_LCDDisplay0_PrintASCII('R');
        FCD_LCDDisplay0_PrintASCII('O');
        FCD_LCDDisplay0_PrintASCII(' ');
        //Posiciona el cursor
        //Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 1)
        FCD_LCDDisplay0_Cursor(0, 1);
        //Muestra la información
        //Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" DE AGUA ")
        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII('D');
        FCD_LCDDisplay0_PrintASCII('E');
        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII('A');
        FCD_LCDDisplay0_PrintASCII('G');
        FCD_LCDDisplay0_PrintASCII('U');
        FCD_LCDDisplay0_PrintASCII('A');
        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII(' ');
        FCD_LCDDisplay0_PrintASCII(' ');
        //Demora
        //Demora: 1000 ms
        delay_ms(255);
        delay_ms(255);
        delay_ms(255);
        delay_ms(235);
        //Posiciona el cursor
        //Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 0)
        FCD_LCDDisplay0_Cursor(0, 0);
        //Muestra la información
        //Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" TINACO VACIO ")
    }
```

```
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('T');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('C');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII(' ');

FCD_LCDDisplay0_PrintASCII('V');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('C');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 1)
FCD_LCDDisplay0_Cursor(0, 1);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" CISTERNA VACIA ")
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('C');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('T');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('V');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('C');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
//Demora
//Demora: 1000 ms
delay_ms(255);
delay_ms(255);
delay_ms(255);
delay_ms(235);
//Salida
//Salida: 0 -> PUERTO A
trisa = trisa & 0xf8;
porta = (porta & 0xf8) | (0 & 0x07);
break;
}
case 41:
{
//Salida
//Salida: 1 -> B0
trisb = trisb & 0xfe;
if (1)
    portb = (portb & 0xfe) | 0x01;
else
    portb = portb & 0xfe;
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 0)
FCD_LCDDisplay0_Cursor(0, 0);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII("INGRESO DE AGUA ")
FCD_LCDDisplay0_PrintASCII('I');
```

```
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('G');
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('D');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII('G');

FCD_LCDDisplay0_PrintASCII('U');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 1)
FCD_LCDDisplay0_Cursor(0, 1);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" A LA CISTERNA ")
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('C');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('T');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('A');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
//Salida
//Salida: 4 -> PUERTO A
trisa = trisa & 0xf8;
porta = (porta & 0xf8) | (4 & 0x07);
break;
}
case 43:
{
//Salida
//Salida: 1 -> B0
trisb = trisb & 0xfe;
if (1)
    portb = (portb & 0xfe) | 0x01;
else
    portb = portb & 0xfe;
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 0)
FCD_LCDDisplay0_Cursor(0, 0);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII("NIVEL DE RESERVA")
FCD_LCDDisplay0_PrintASCII('N');
FCD_LCDDisplay0_PrintASCII('I');
FCD_LCDDisplay0_PrintASCII('V');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('D');
FCD_LCDDisplay0_PrintASCII('E');
```

```
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('S');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('R');
FCD_LCDDisplay0_PrintASCII('V');
FCD_LCDDisplay0_PrintASCII('A');
//Posiciona el cursor
//Llamar a Macro de Componente: LCDDisplay(0)::Cursor(0, 1)
FCD_LCDDisplay0_Cursor(0, 1);
//Muestra la información
//Llamar a Macro de Componente: LCDDisplay(0)::PrintASCII(" COMPLETO ")
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');

FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII('C');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII('M');
FCD_LCDDisplay0_PrintASCII('P');
FCD_LCDDisplay0_PrintASCII('L');
FCD_LCDDisplay0_PrintASCII('E');
FCD_LCDDisplay0_PrintASCII('T');
FCD_LCDDisplay0_PrintASCII('O');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
FCD_LCDDisplay0_PrintASCII(' ');
//Salida
//Salida: 4 -> PUERTO A
trisa = trisa & 0xf8;
porta = (porta & 0xf8) | (4 & 0x07);
break;
}
}
//Expresión
//Expresión: A?
switch (FCV_A)
{
    case 1:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 2:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 3:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 4:
    {
        //Llamar Macro
```

```
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }

    case 5:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 6:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 7:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 8:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 10:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 12:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
}
//Expresión
//Expresión: A?
switch (FCV_A)
{
    case 13:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 14:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
    }
}
```

```
        break;
    }
    case 16:
    {
        //Llamar Macro
        //Llamar Macro: FALLA

        FCM_FALLA();
        break;
    }
    case 17:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 18:
    {
        //Llamar Macro
        //Llamar Macro: FALLA

        FCM_FALLA();
        break;
    }
    case 19:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 20:
    {
        //Llamar Macro
        //Llamar Macro: FALLA

        FCM_FALLA();
        break;
    }
    case 21:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 22:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 23:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
}
//Expresión
//Expresión: A?
```

```
switch (FCV_A)
{
    case 24:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 26:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 28:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 29:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 30:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 34:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 36:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 37:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 38:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
    }
}
```

```
        break;
    }
    case 40:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
}
//Expresión
//Expresión: A?
switch (FCV_A)
{
    case 42:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 44:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 45:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 46:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 48:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 49:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 50:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
}
```

```
case 51:
{
    //Llamar Macro
    //Llamar Macro: FALLA
    FCM_FALLA();
    break;
}
case 52:
{
    //Llamar Macro
    //Llamar Macro: FALLA
    FCM_FALLA();
    break;
}
case 53:
{
    //Llamar Macro
    //Llamar Macro: FALLA
    FCM_FALLA();
    break;
}
}
//Expresión
//Expresión: A?
switch (FCV_A)
{
case 54:
{
    //Llamar Macro
    //Llamar Macro: FALLA

    FCM_FALLA();
    break;
}
case 55:
{
    //Llamar Macro
    //Llamar Macro: FALLA
    FCM_FALLA();
    break;
}
case 56:
{
    //Llamar Macro
    //Llamar Macro: FALLA
    FCM_FALLA();
    break;
}
case 57:
{
    //Llamar Macro
    //Llamar Macro: FALLA
    FCM_FALLA();
    break;
}
case 58:
{
    //Llamar Macro
    //Llamar Macro: FALLA
    FCM_FALLA();
    break;
}
case 60:
{
    //Llamar Macro
```

```
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 61:
    {
        //Llamar Macro
        //Llamar Macro: FALLA
        FCM_FALLA();
        break;
    }
    case 62:
    {
        //Llamar Macro
        //Llamar Macro: FALLA

        FCM_FALLA();
        break;
    }
}
mainendloop: goto mainendloop;
}
void MX_INTERRUPT_MACRO(void)
{
}
```

GRABACIÓN DEL PROGRAMA EN EL MICROCONTROLADOR

Para poder grabar el archivo hexadecimal del programa de trasvase de agua, se requiere de conectar el microcontrolador a una computadora con puerto serial COM1 por medio de un programador de PIC y el software WinPic800, se debe de configurar en el software el modelo del programador que se va a conectar, en este caso elegimos el JDM Programmer.

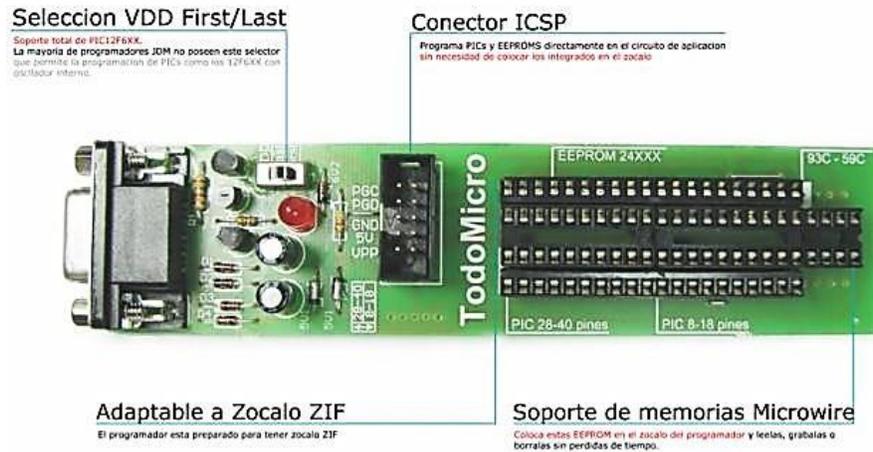


Figura 64. Programador de PIC JDM.

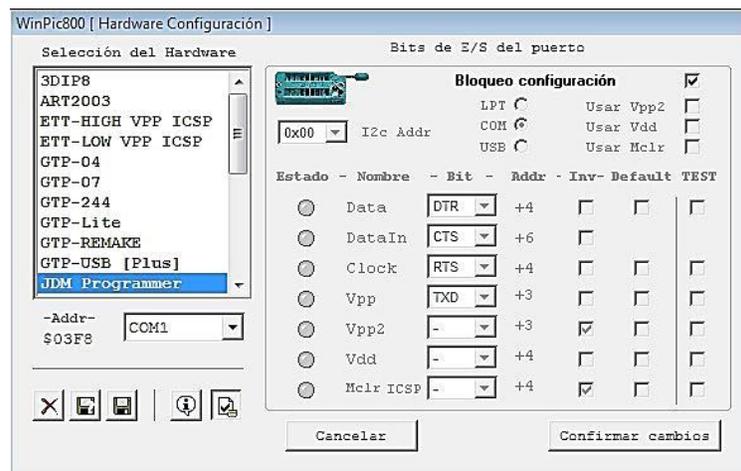
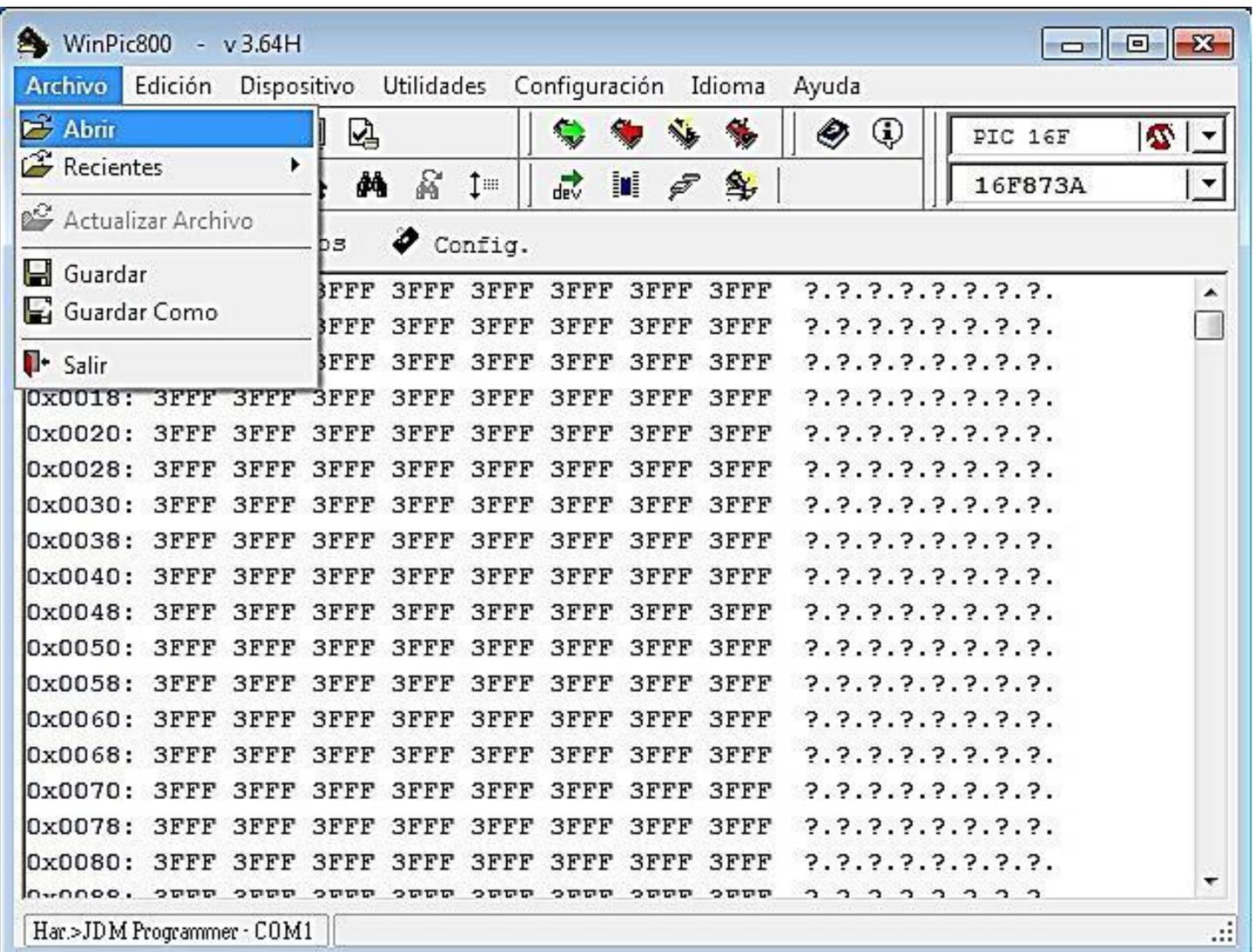
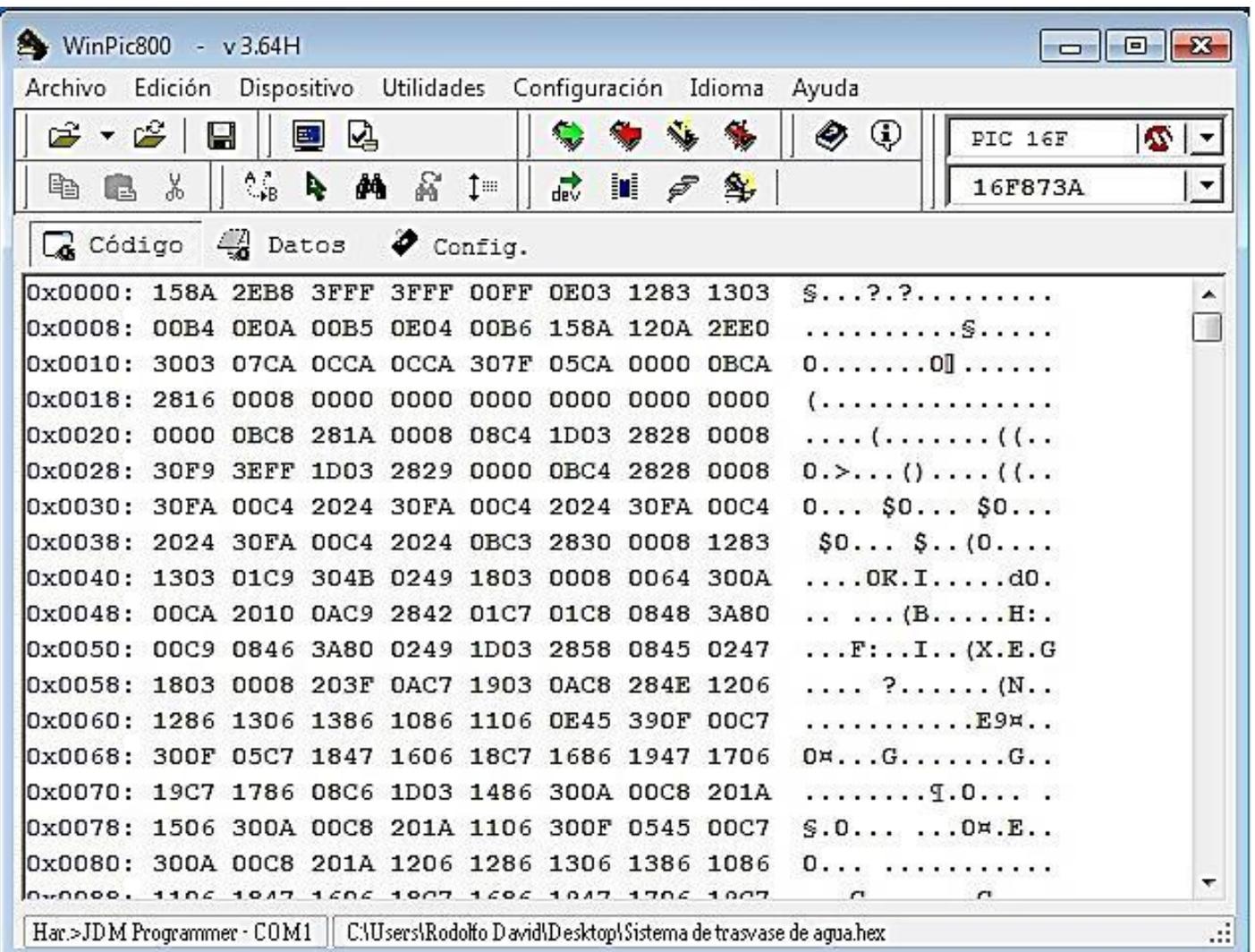


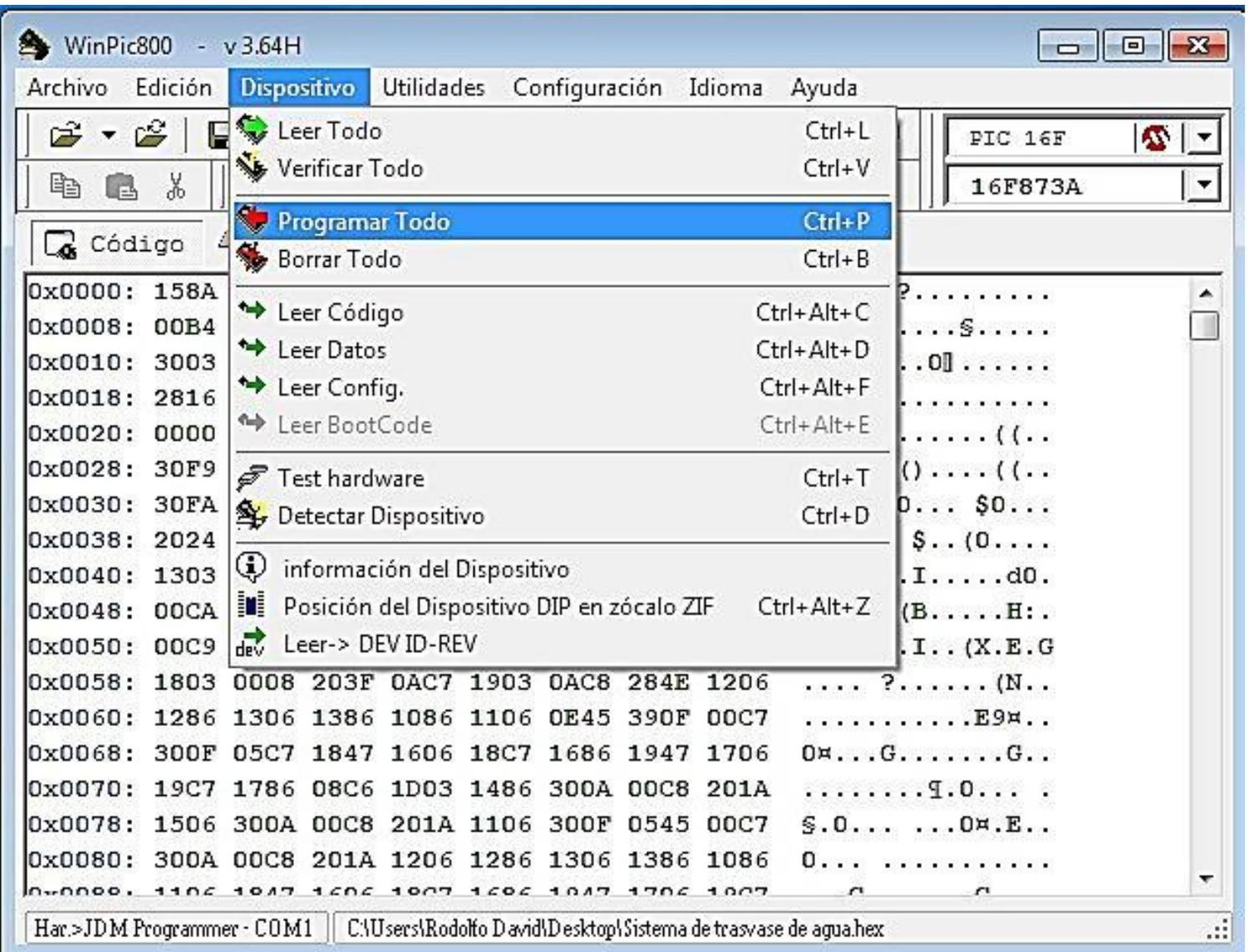
Figura 65. Configuración del programador.



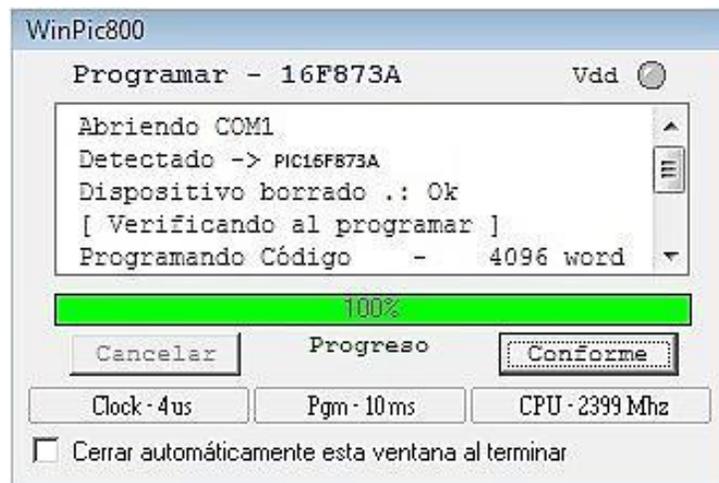
Una vez configurado correctamente el hardware del programador se procede a abrir el archivo hexadecimal que se compiló en Flowcode.



Así es como se muestra ya cargado el archivo hexadecimal, se tiene que elegir en el software WinPic800 el modelo del microcontrolador a usar y tiene que corresponder con el elegido en el programa Flowcode.



Se despliega el menú de la opción "Dispositivo" y se selecciona Programar Todo.



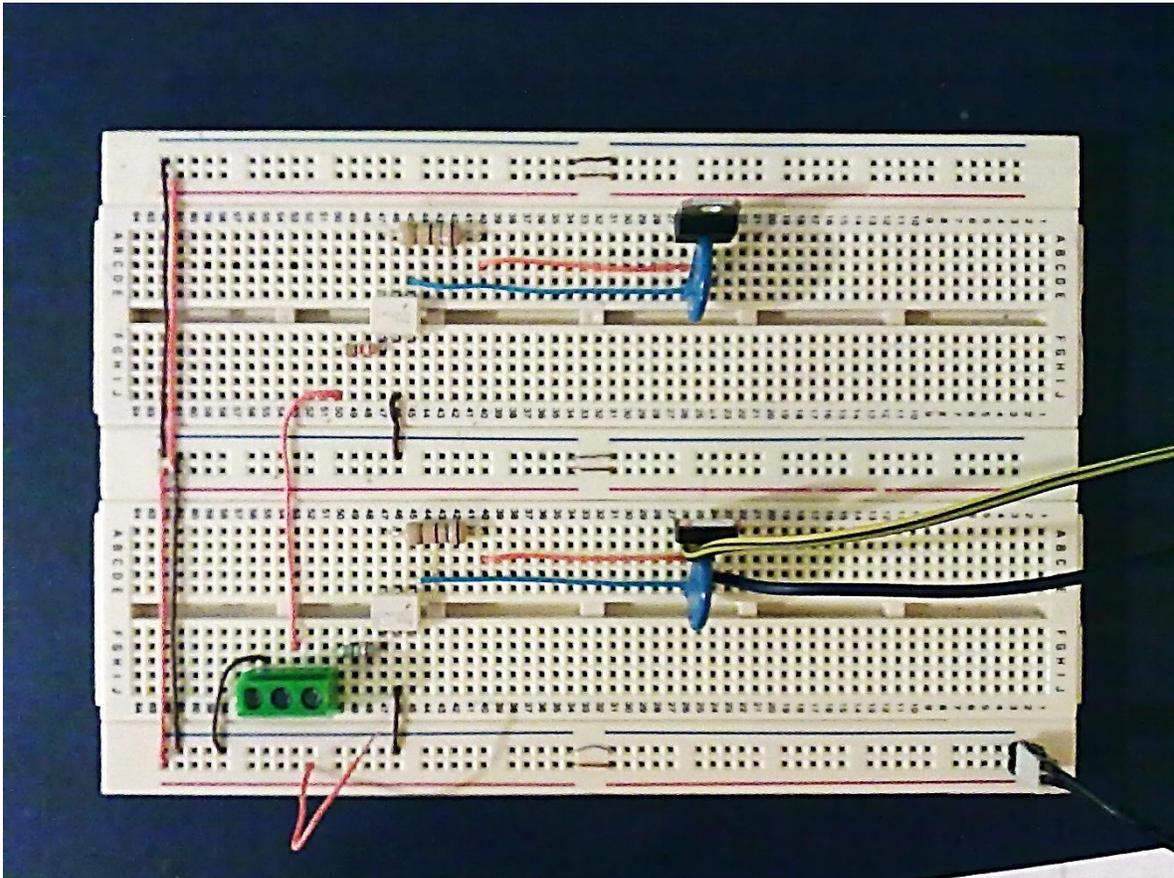
Se abre una ventana informándonos del progreso de la grabación, si no hay errores el microcontrolador estará grabado con el programa creado en Flowcode y estará listo para su funcionamiento.

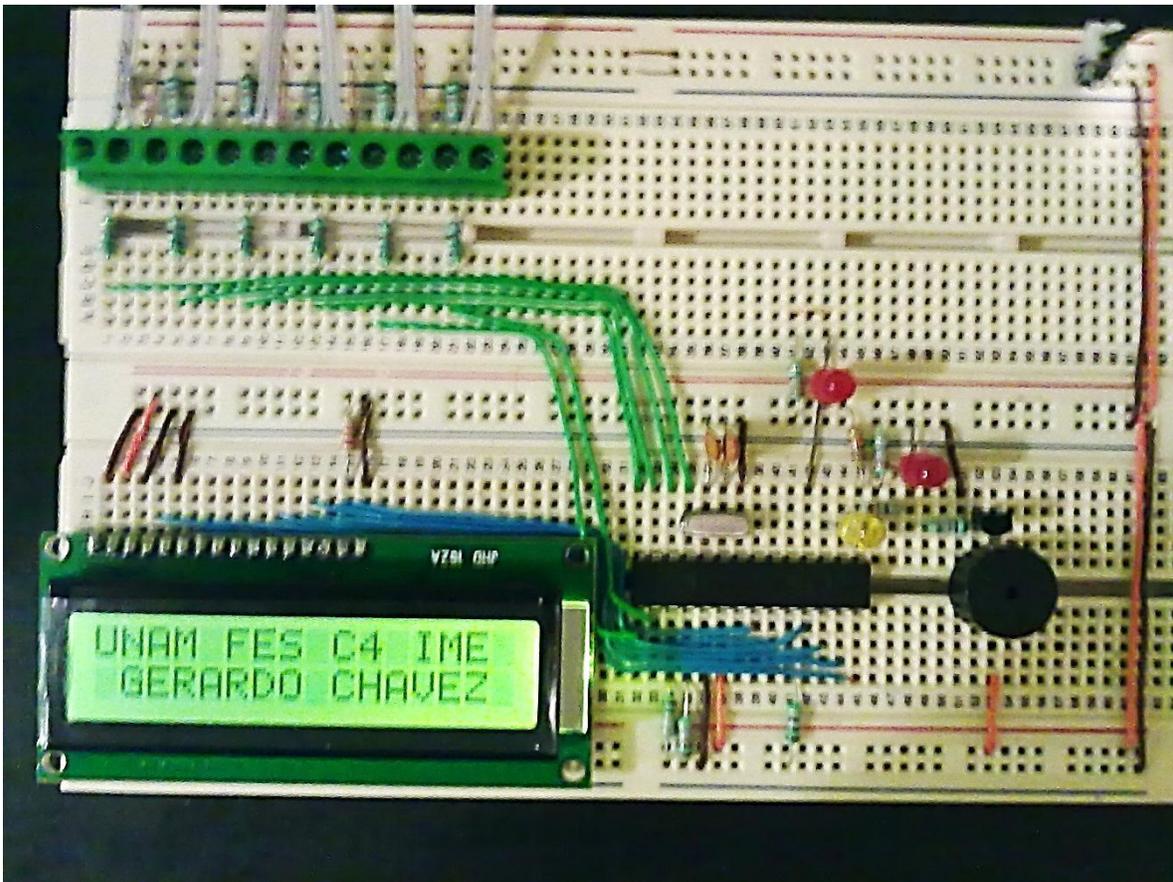
CAPITULO 4

LA SIMULACIÓN FÍSICA Y CONSTRUCCIÓN DEL PROTOTIPO

SIMULACIÓN EN PROTOBOARD.

Se hicieron pruebas de funcionamiento del proyecto, armando el circuito en el protoboard, primero se realizó con la etapa de potencia por ser más sencillo de simular, lo cual el circuito funcionó bien al momento de mandar un estado lógico alto por el MOC3031, se lograba la conmutación del triac con los valores calculados, sin problemas de errores.



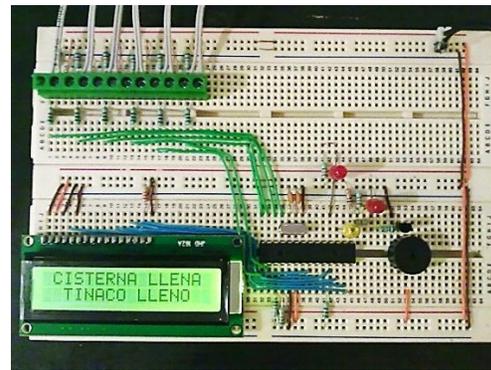
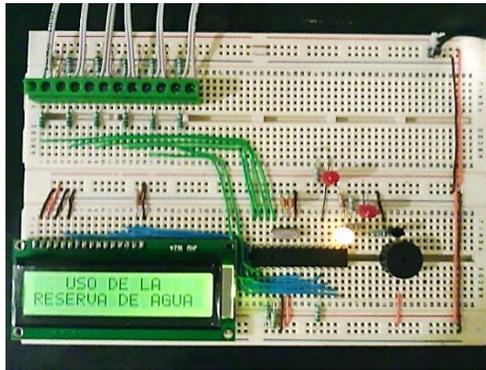
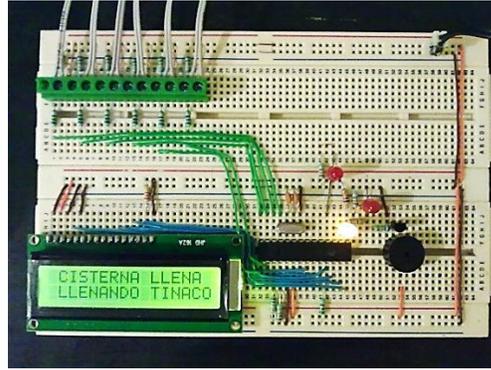
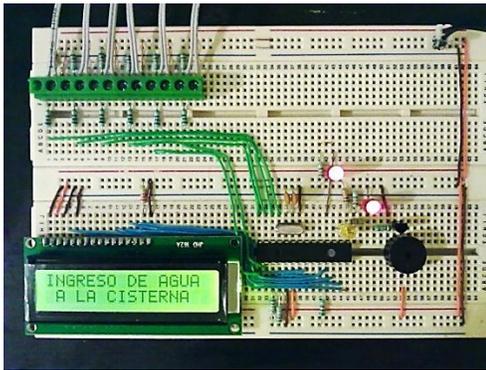
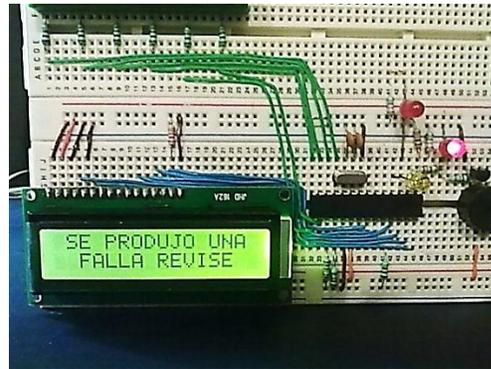
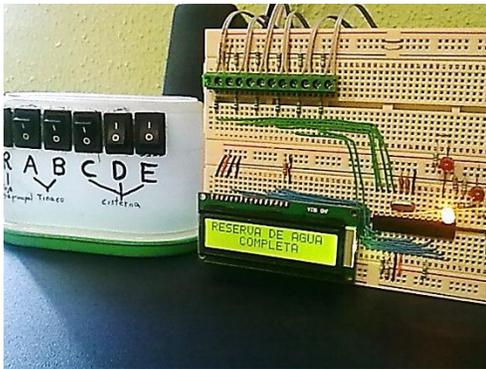


Durante la simulación en el protoboard, resultó que la grabación del microcontrolador fue exitosa, al momento de hacer funcionar el circuito, se mostraba el mensaje inicial del control automático.

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

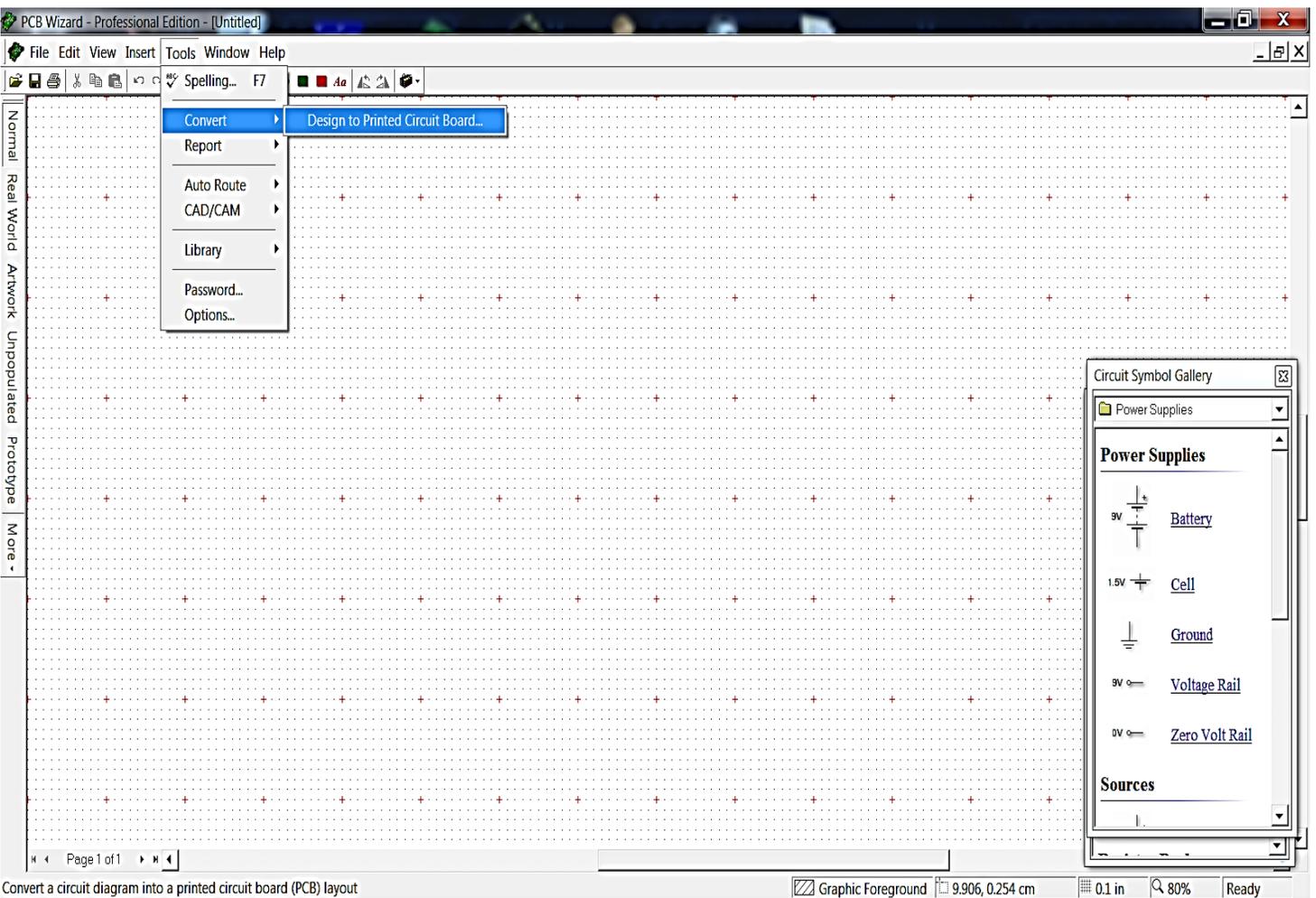
FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN CAMPO 4

Se conectaron en la entrada del microcontrolador unos interruptores para simular a los interruptores de nivel con flotador. Se fue simulando cada una de las combinaciones binarias del todo el proceso, haciendo la comprobación de cada uno de ellos, incluyendo las fallas. Aquí se muestra las imágenes de algunas combinaciones y sus respectivos mensajes desplegados en el display LCD.

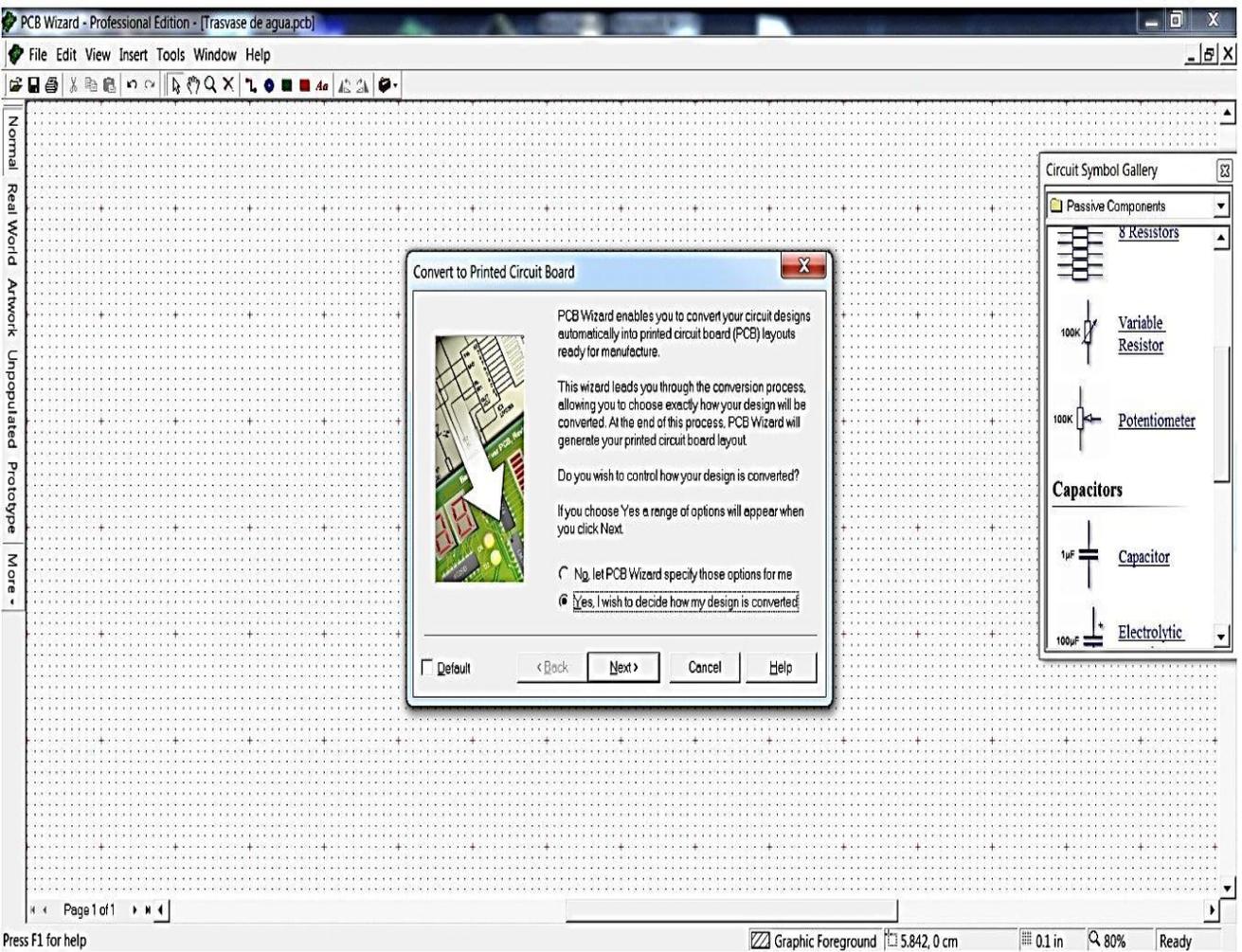


En este esquema se puede ver que solo corresponde al circuito principal y que contiene un conector para su conexión con la placa de circuito impreso secundaria que es la de la etapa de potencia y la fuente de alimentación.

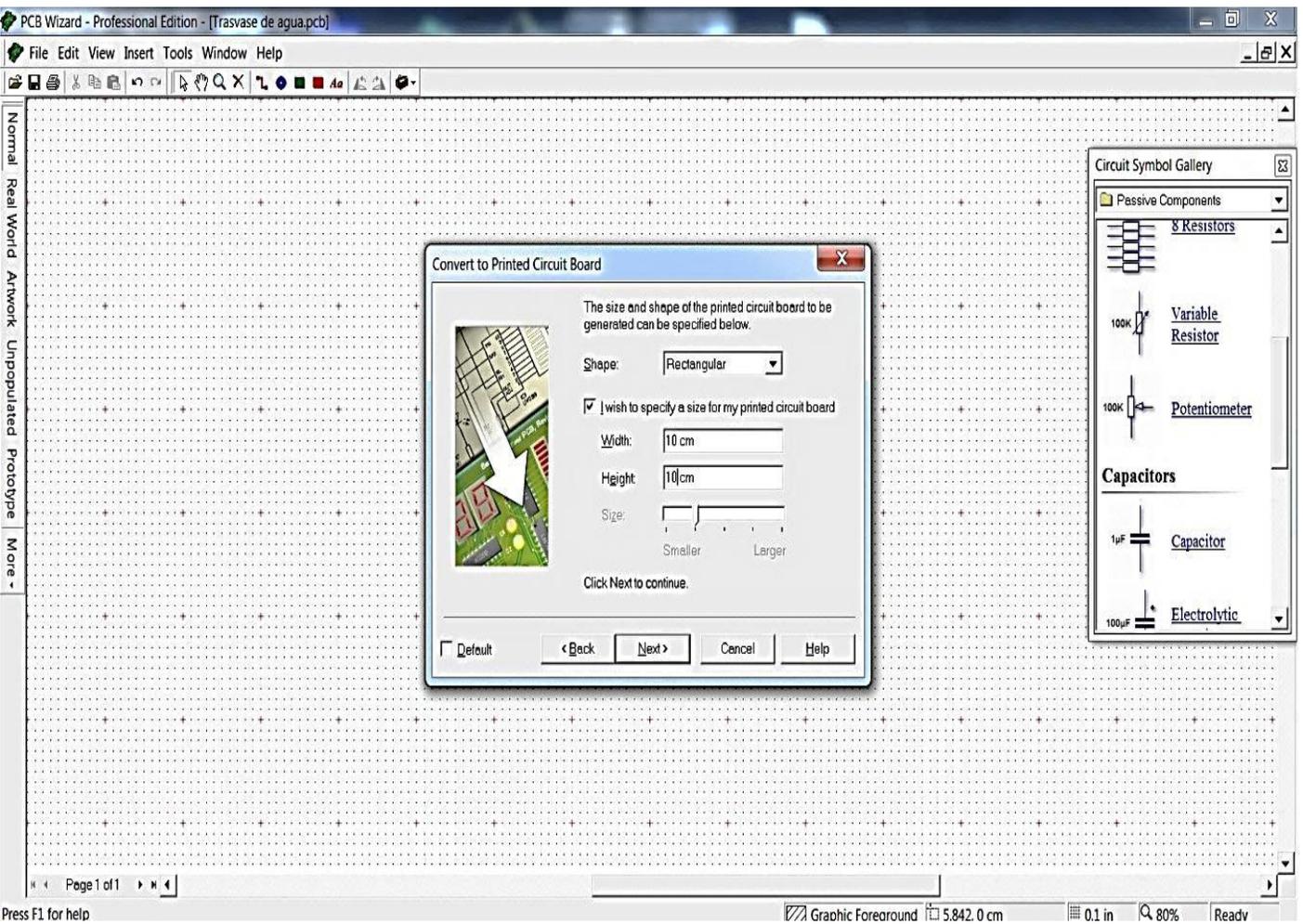
Teniendo listo el esquema, se procede a crear la placa con el asistente de creación de PCB.



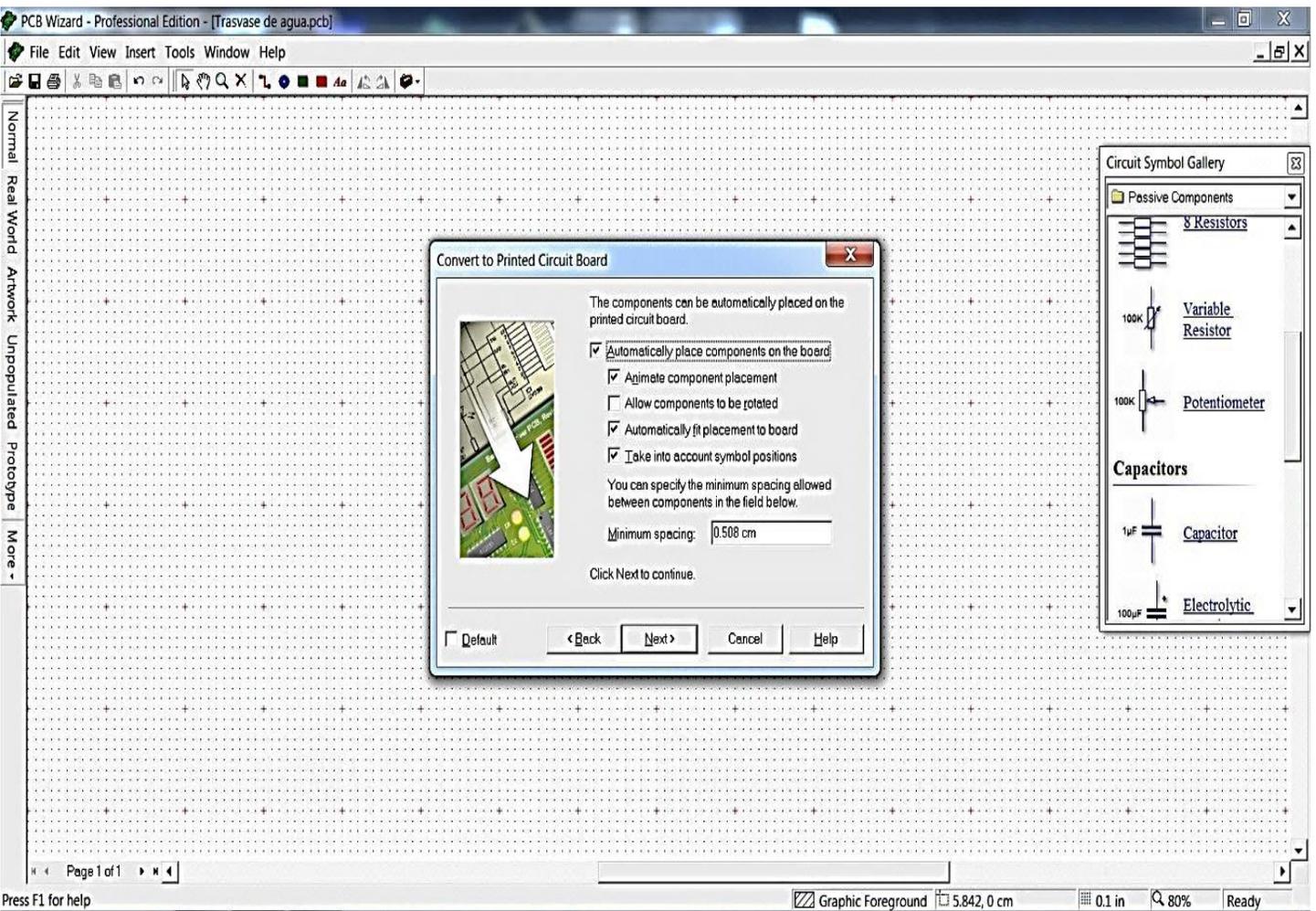
Para abrir la ventana del asistente de creación de placas, desplegamos el menú "Tools" y seleccionamos Convert / Design to circuit printed board.



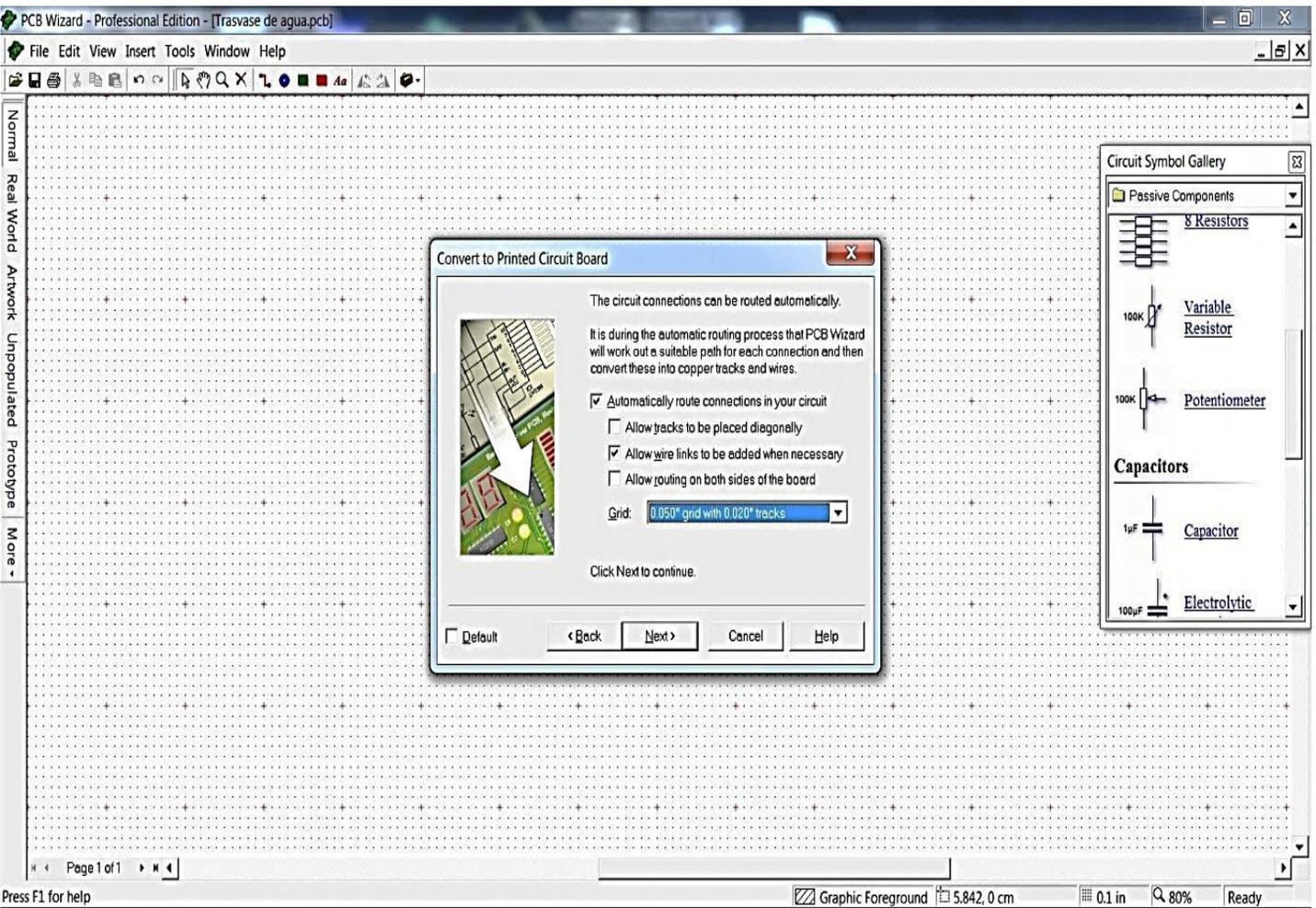
Se abre la ventana con dos opciones la primera opción pregunta si la placa va a ser creada por el asistente a su criterio y omitir los pasos siguientes, y la segunda opción pregunta si va a ser creada por el usuario dando especificaciones de cómo va a ser creada. Para nuestro caso se elige la segunda opción.



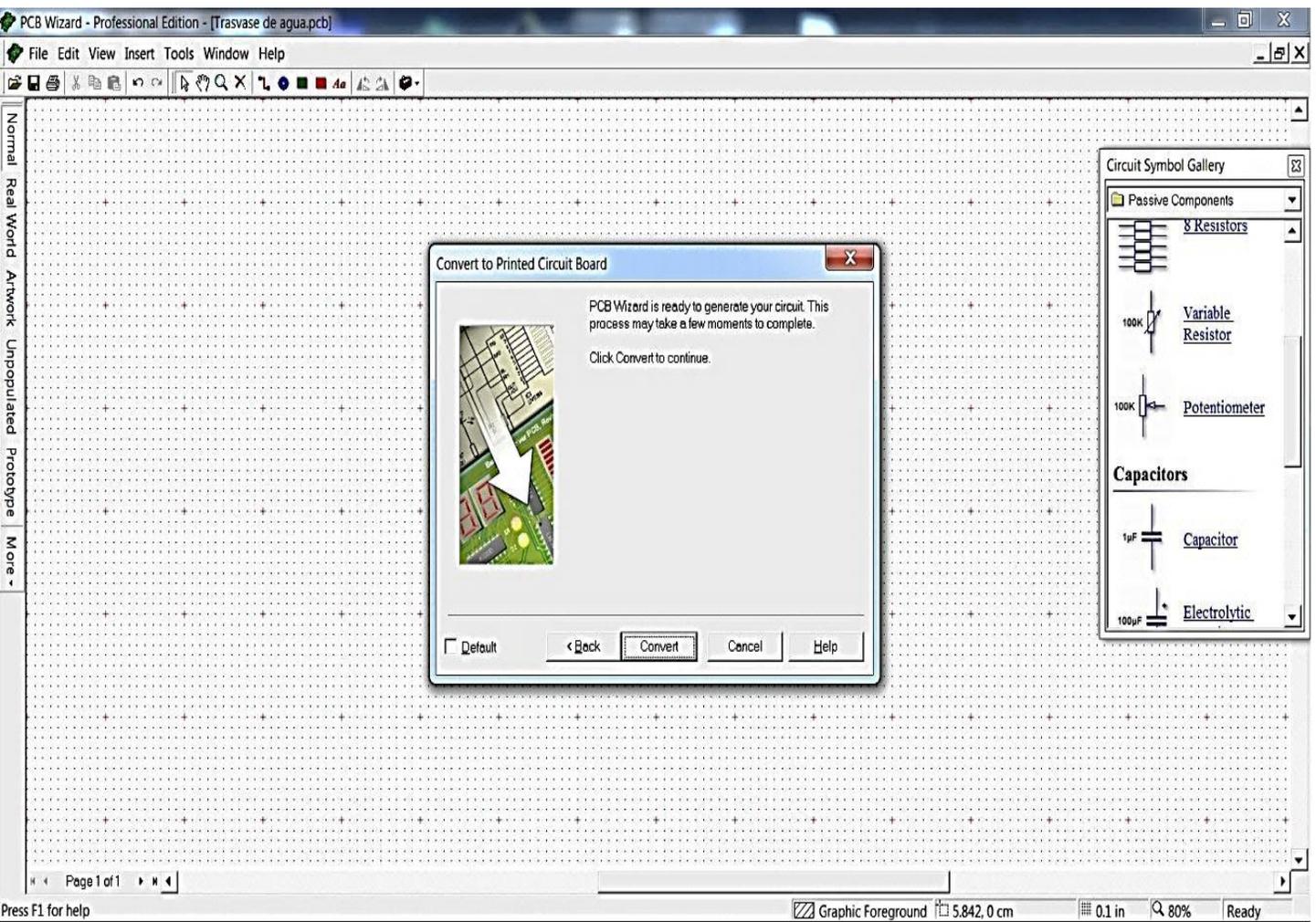
En la siguiente ventana podemos configurar las medidas que tendrá la placa en centímetros. Elegimos de 10 cm horizontal x 10 cm vertical.



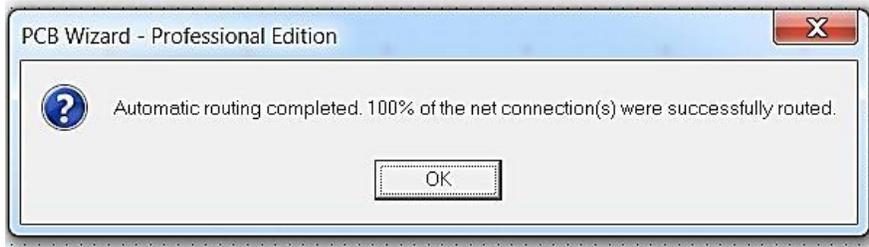
La ventana que sigue nos pregunta si los componentes van a ser colocados de manera automática en la placa por el asistente, tildamos esta opción.



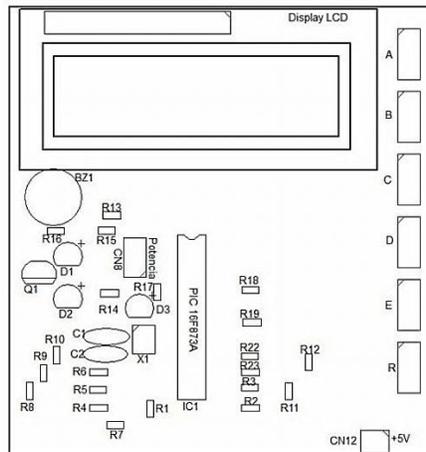
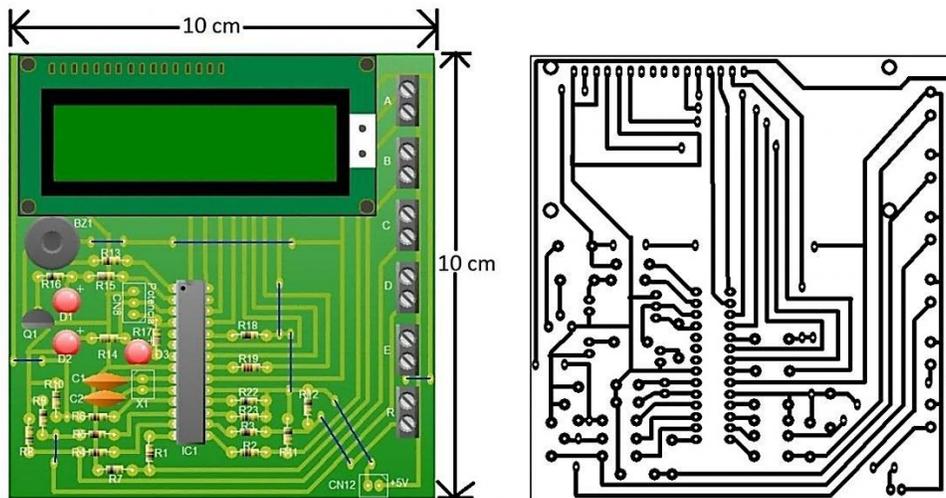
En esta ventana podemos habilitar el ruteo automático y podemos escoger el tamaño de las pistas. Tildamos esta opción.



Por último esta ventana confirma que se procederá a crear la placa cuando se pulse el botón "Convert".



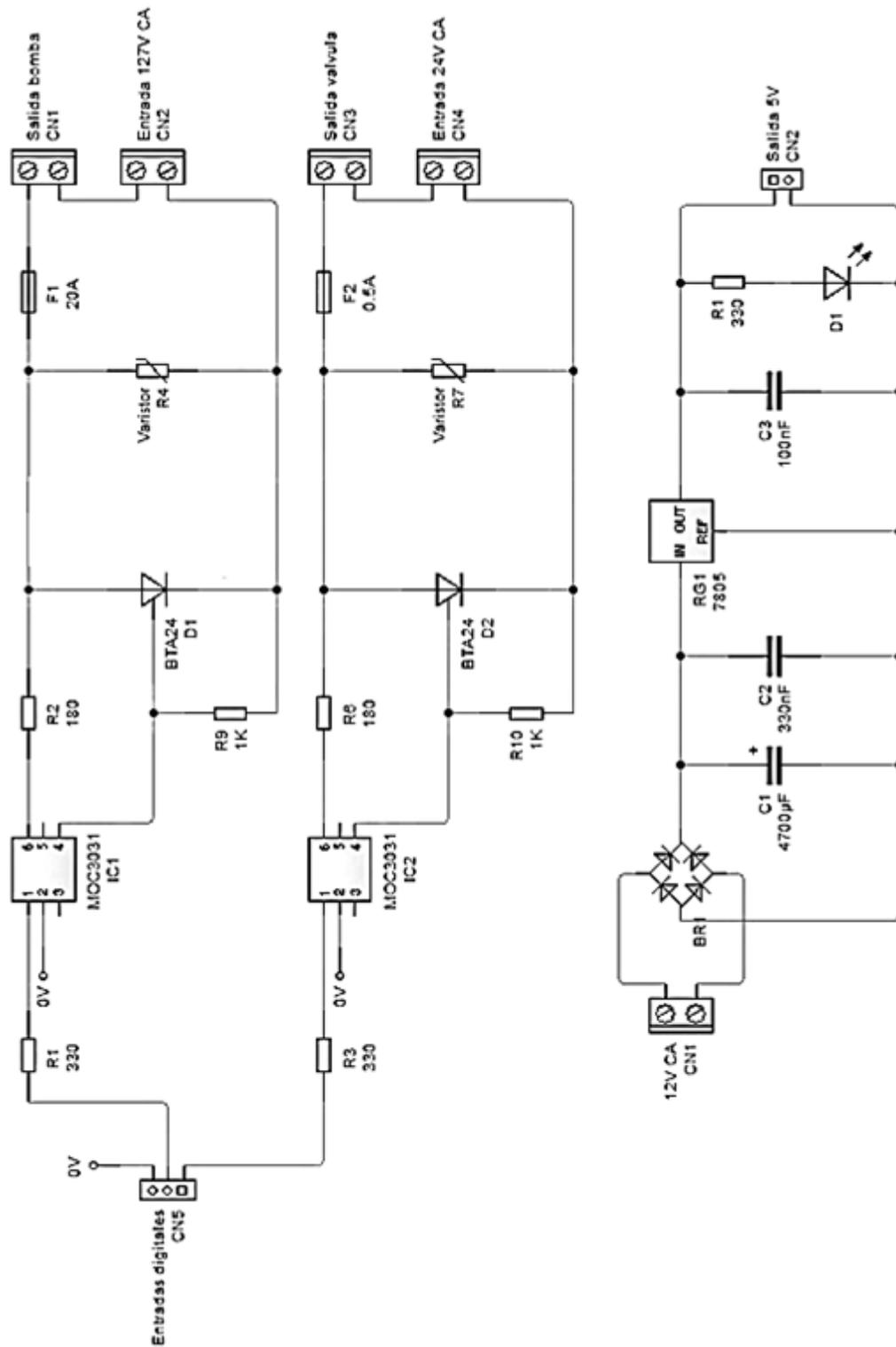
Al terminar el ruteado de todo el circuito, si no hubo errores aparece este cuadro de dialogo, informando que se ha completado el ruteado, conectando satisfactoriamente el 100% de las conexiones. Cabe aclarar que en nuestro caso, se tuvo que colocar los componentes manualmente ya que la posición de algunos componentes, dificultaban el ruteado.

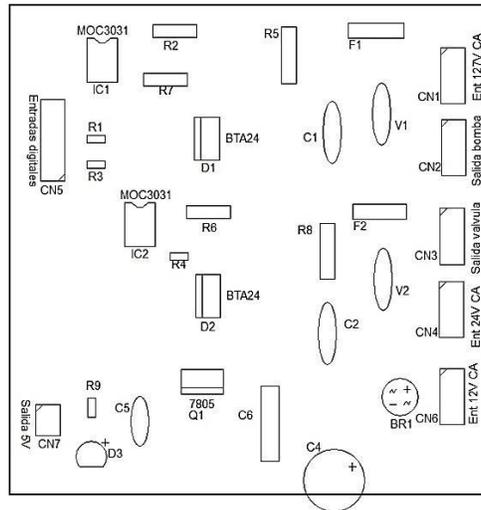
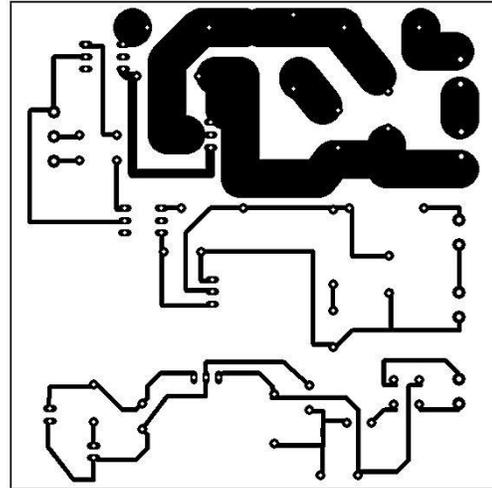
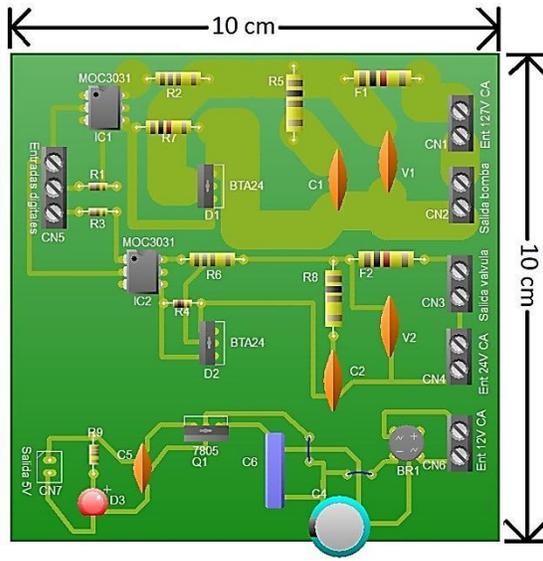


En estas imágenes, se observa la vista real de la disposición de los componentes en la placa, así como el ruteado de las pistas ya listo para transferir y serigrafiar en la placa fenólica.

DISEÑO DE LA PLACA DE CIRCUITO IMPRESO SECUNDARIA.

El proceso para la construcción de esta placa es lo mismo que la placa anteriormente descrita. En la siguiente imagen se visualiza el esquema electrónico de la etapa de potencia para la bomba centrífuga y de la electroválvula, así como la fuente de alimentación que alimentará a la placa principal.





Así es como se muestra el diseño de la placa secundaria, cabe destacar que el grosor de las pistas de la etapa de potencia de la bomba es de 10mm y está estañada para darle cierto grosor a la pista ya que circulará alta corriente, si 2A equivale a 1 mm de espesor, por consiguiente, para 20A equivale a 10 mm de espesor de no tomar estas medidas, se corre el riesgo de que las pistas se calienten en exceso y se desprendan o se abran en la placa.

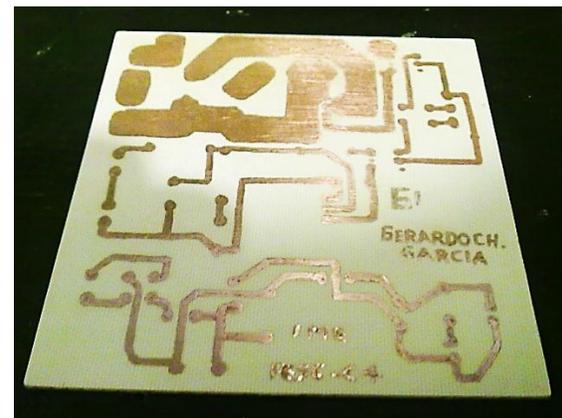
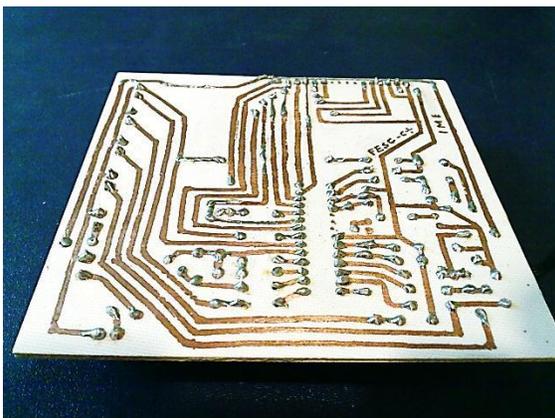
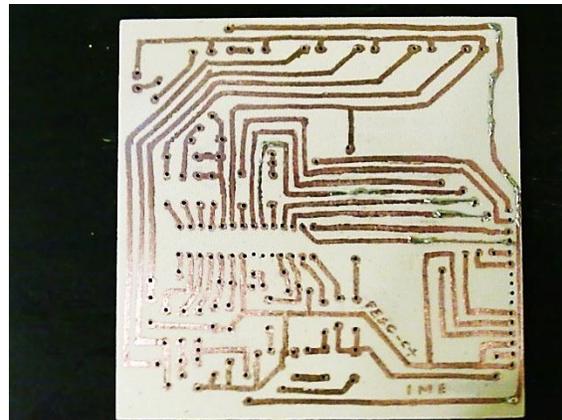
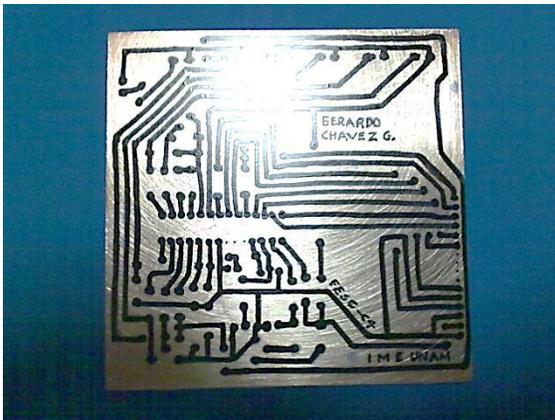
CONSTRUCCIÓN DE LAS PLACAS

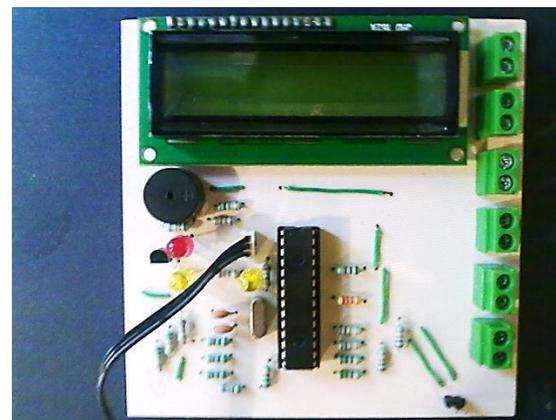
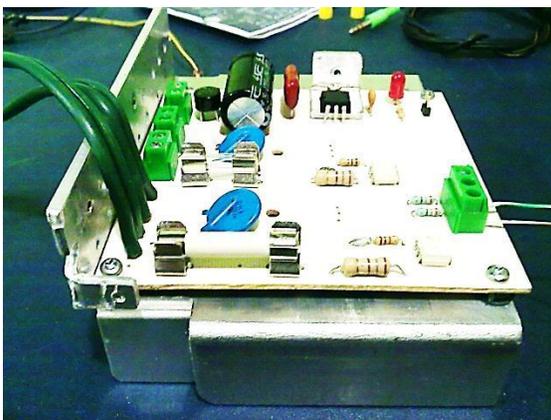
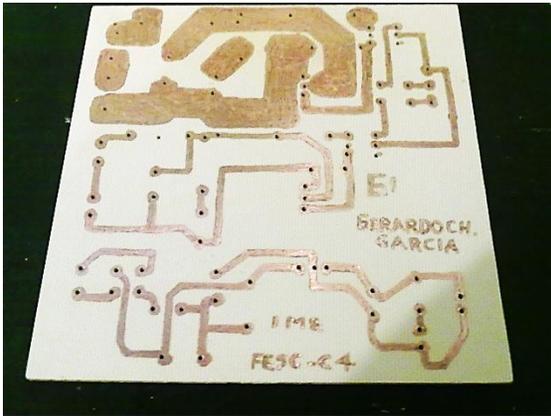
Una vez obtenido el dibujo del lado de las pistas, se procede a transferirlo a la placa de cobre remarcándolo con tinta permanente negra para que el cloruro férrico no las desintegre. El cloruro férrico solo desintegrará el cobre expuesto

El siguiente paso es perforar la placa para colocar los componentes en su respectivo lugar.

Se colocan los componentes y se procede a soldarlos en la placa.

Por último se fijan con tornillo los triacs con el disipador de calor, haciendo uso de pasta térmica para una excelente disipación de calor.





MONTAJE EN EL GABINETE.

Ya con las placas fabricadas se procede a montarlas en el gabinete, el cual en la parte frontal del gabinete se practicaron los tres orificios donde sobresalen los leds indicadores y se recortó un rectángulo para el display LCD.

Una vez montada la placa principal, se conectó a la placa secundaria y se montó de modo que el disipador de calor no estorbara para cerrar el gabinete. El transformador de la fuente de alimentación de 5V y que a su vez alimenta a la electroválvula, se fijó con tornillo a la parte posterior del disipador de calor.

También se perforó todo el costado del gabinete para que existiera ventilación en el interior del gabinete.

El cable de alimentación sale del lado superior y el cable donde se conecta la bomba centrífuga, la electroválvula y los interruptores de nivel, salen del lado inferior del gabinete.



CAPITULO 5

PRESUPUESTO DEL PROYECTO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN CAMPO 4

RESUMEN DE COSTOS DEL PROTOTIPO

MATERIAL	CANTIDAD	PRECIO POR UNIDAD	PRECIO TOTAL
Microcontrolador PIC16F873A-20/S	1	\$80.00	\$80.00
Display LCD JHD162A	1	\$160.00	\$160.00
Optoacoplador MOC3031	2	\$17.00	\$34.00
Transistor BC547-B	1	\$4.00	\$4.00
Triac BTA24-800BW	2	\$65.00	\$130.00
Mini buzzer 4KHz a 1.5 - 16 VCC	1	\$15.00	\$15.00
Cristal de cuarzo de 4Mhz	1	\$16.00	\$16.00
Regulador de tensión MC7805CT	1	\$9.00	\$9.00
Transformador 12+12V 500mA	1	\$104.00	\$104.00
Puente rectificador 200V 2A	1	\$9.00	\$9.00
Led rojo 5mm	2	\$2.00	\$4.00
Led amarillo 5mm	2	\$2.00	\$4.00
Capacitor electrolítico 4700mF/25V	1	\$14.00	\$14.00
Capacitor polyester 330nF/250V	1	\$6.00	\$6.00
Capacitor cerámico 100nF/50V	1	\$3.00	\$3.00
Capacitor ceramico 15pF/50V	2	\$4.00	\$8.00
Resistencia 330 1/4W	6	\$1.00	\$6.00
Resistencia 180 1/2W	2	\$1.00	\$2.00
Resistencia 10K 1/4W	10	\$1.00	\$10.00
Resistencia 1K 1/4W	7	\$1.00	\$7.00
Resistencia 2k2 1/4W	1	\$1.00	\$1.00
Varistor V130K14	2	\$7.00	\$14.00
Fusible americano 250V/20A	1	\$9.00	\$9.00
Fusible europeo 250V/0.3A	1	\$5.00	\$5.00
Porta fusible tipo americano	1	\$5.00	\$5.00
Porta fusible tipo europeo	1	\$5.00	\$5.00
Base para C.I de 28 pines	1	\$5.00	\$5.00
Tira de 20 pines	1	\$6.00	\$6.00
Interruptor tipo balancin 1P1T	6	\$6.00	\$36.00
Terminal doble con tornillo	3	\$5.00	\$15.00
Terminal triple con tornillo	1	\$6.00	\$6.00
Cable plano 14 vías 28 AWG 1m	1	\$16.00	\$16.00
Cable trenzado 4 pares 24 AWG 1m	1	\$9.00	\$9.00
Cable duplex 14 AWG 1m	1	\$25.00	\$25.00
Disipador de calor TO-220	1	\$15.00	\$15.00
Disipador de calor 11x15cm	1	\$50.00	\$50.00
Placa fenólica 1 cara de 10 x 10cm	2	\$9.00	\$18.00
Tubo de soldadura 20grs	1	\$23.00	\$23.00
Cloruro férrico 220mL	1	\$26.00	\$26.00
Pasta térmica base de silicon	1	\$49.00	\$49.00
Gabinete de plastico 11x14x7.5cm	1	\$35.00	\$35.00
			\$998.00

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN CAMPO 4

La tabla anterior se refiere a la lista de precios del material que se utilizó para la construcción del control automático. Los precios incluyen el IVA. Costos en la mano de obra, de diseño, no se consideran ya que fue de construcción propia y con propósitos demostrativos, pero cabe señalar que la fabricación en serie de éste control automático se reduciría enormemente el costo total por unidad.

También en la siguiente tabla se incluyen los precios orientativos de los elementos externos conectados al control automático.

DISPOSITIVO	CANTIDAD	PRECIO POR UNIDAD	PRECIO TOTAL
Interruptor de nivel CEBEK	5	\$230.00	\$1,150.00
Interruptor de flujo CEBEK	1	\$215.00	\$215.00
Bomba centrifuga Siemens 0.5HP	1	\$850.00	\$850.00
Electroválvula Orbit	1	\$325.00	\$325.00
			<u>\$2,540.00</u>

CONCLUSIONES

Para la realización de este proyecto, se describieron todos los bloques que conforman el diseño electrónico del control automático de trasvase de agua cisterna – tinaco, y se ha logrado obtener un control capaz de brindarle a la sociedad en general, confort, seguridad y economía en su vivienda, ya que este diseño cumple con los objetivos propuestos.

Se logró programar un microcontrolador a base de diagramas de flujo y con el empleo de subrutinas preestablecidas para el uso del display LCD, facilitan la programación enormemente y minimizan errores. Nos permitió en gran manera la simulación de todo el sistema desde el entorno de programación gráfico Flowcode de una manera óptima.

Algo muy importante en el momento del diseño, fue que al emplear el microcontrolador PIC 16f873A, se logró reducir en lo posible, el uso de componentes electrónicos como compuertas lógicas, circuitos combinacionales, etc. logrando reducir en lo posible el tamaño de las placas de circuito impreso, también se logró eliminar el uso de relevadores electromecánicos ya que estos al momento de fallar, pueden quedarse pegados sus contactos y ocasionar un desbordamiento, aparte de que son grandes y ruidosos, para este caso se utilizó relevadores de estado sólido que aparte de ser pequeños, son muy confiables y seguros. En este proyecto se utilizaron componentes fáciles de conseguir en el mercado.

El diseño acepta conectar electroválvulas de 24V con un consumo de 300mA de corriente alterna y una bomba centrífuga monofásica de 0.5HP suficiente para una casa habitación. Es de fácil instalación y su mantenimiento es nulo ya que no emplea componentes electromecánicos internos en el módulo.

Se conectaron interruptores tipo balancín de un polo un tiro para simular los interruptores de nivel y probar cada una de las combinaciones del proceso de trasvase de agua.

Este proyecto se construyó y se probó obteniendo resultados favorables en el funcionamiento.

BIBLIOGRAFÍA

José María Angulo Usategui, 2002

Laboratorio de prácticas de microelectrónica Vol.II

Editorial Mc Graw Hill

Mikel Etxebarria Isuskiza, 2011

Microcontroladores PIC: Teoría y práctica, el pic 16F88x y el laboratorio USB-PIC school

Editorial Creaciones Copyright, 428 páginas

Enriquez Harper Gilberto, 2002

El ABC del control electrónico de las máquinas eléctricas

Editorial Limusa, 401 páginas

Fraenkel, Peter y Thake, Jeremy, 2010

Dispositivos de elevación del agua. Manual para usuarios y planificadores 3° Ed.

Editorial Alfaomega, 352 páginas

Dogan Ibrahim, 2007

Programación de microcontroladores PIC

Editorial Marcombo, 327 páginas

Jim Cox, 1995

Motores eléctricos

Editorial Ediciones Gustavo Gilli, 126 páginas

Angulo José María, 2006

Microcontroladores PIC 2° parte: PIC 16F87X Diseño práctico de aplicaciones

Editorial Mc Graw Hill, 304 páginas

Angulo Losé María, 1997

Enciclopedia de electrónica moderna 4° Ed.

Editorial Thomson Paraninfo, 168 páginas

Mikel Etxebarria Isuskiza, (2012, Enero)

El lenguaje gráfico Flowcode y el laboratorio USB-PIC school. Una revolución en la formación y el diseño de aplicaciones con PIC 1° parte

Microsystems Engineering (MSE) Ingeniería de Microsistemas Programados S.L., 50 - 59

John Dobson, (2009, Julio)

Flowcode y PICKit 2. Un arma letal contra los problemas que atenazan al Ingeniero de Diseño

Microsystems Engineering (MSE) Ingeniería de Microsistemas Programados S.L., 28 - 31

Foros de Electrónica Comunidad internacional de electrónicos,
www.forosdeelectronica.com/f13/tutorial-basico-pcb-wizard-17005/, recuperado en la fecha 03/08/12

Fadisel S.L., www.cebek.com/cebek-components/sensor/sensor-nivel-de-liquidos-horizontal-cebek-c-7236_r_270_739.aspx, recuperado en la fecha 25/07/12

Euro-Rain Distribución de sistemas de riego y fuentes ornamentales, <http://www.euro-rain.es/es/catalogo/electrovalvulas-riego/orbit/modelo-94150.html>, recuperado en la fecha 27/07/12

Matrix Multimedia, Curso Flowcode.,
<http://www.matrixmultimedia.com/resources/getresource.php?id=582>, recuperado en la fecha 12/06/12

Datasheet catalog, hojas de características de componentes electrónicos (BC547, L7805, PIC16F873, MOC3031, BTA24-800BW, JHD162A) <http://www.datasheetcatalog.com/>, recuperado en la fecha 16/06/12

Rotoplas, Trasvase de agua cisterna – tinaco, <http://www.rotoplas.com/hogar.html>, recuperado en la fecha 04/07/12