

# **ESCUELA POLITÉCNICA NACIONAL**

## **ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

### **CONSTRUCCIÓN DE UN PROTOTIPO DE DETECCIÓN Y CONTROL ELECTRÓNICO DE TEMPERATURA E ILUMINACIÓN PARA EL ECOSISTEMA DE UN ACUARIO DE PECES TROPICALES EN UN CLIMA TEMPLADO**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN  
ELECTRÓNICA Y TELECOMUNICACIONES**

**SALA CARLOSAMA LEONEL ANDRÉS**  
[leonel.sala@est.epn.edu.ec](mailto:leonel.sala@est.epn.edu.ec)

**DIRECTOR: ING. ALFREDO PLUTARCO ARCOS LARA**  
[alfredo.arcos@epn.edu.ec](mailto:alfredo.arcos@epn.edu.ec)

**QUITO, noviembre, 2014**

## **DECLARACIÓN**

Yo, Leonel Andrés Sala Carlosama declaro bajo juramento que el trabajo aquí escrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual por su Reglamento y por normativa institucional vigente.

---

Leonel Andrés Sala Carlosama

## CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Leonel Andrés Sala Carlosama, bajo mi supervisión.

---

Ing. Alfredo Arcos  
**DIRECTOR**

## **AGRADECIMIENTO**

A Dios. A mis padres y hermanos por el apoyo incondicional en todo momento, a los profesores que compartieron su conocimiento en las aulas de clases, a mi tutor por la ayuda y consejos para el proceso de este proyecto, a mis compañeros y amigos que me acompañaron en las aulas, y al último y no menos importante, mi novia que me apoyo en el tiempo que hice esta tesis, por su paciencia y ayuda, te agradezco mucho Diana Yunga.

## **DEDICATORIA**

A mis padres por el amor y cariño que los tengo, y también como demostración de agradecimiento en el apoyo a mis estudios, a mi novia que por toda la ayuda, paciencia e insistencia he podido hacer este proyecto.

## ÍNDICE DE CONTENIDO

DECLARACIÓN .....	i
CERTIFICACIÓN .....	ii
AGRADECIMIENTO.....	iii
DEDICATORIA.....	iv
ÍNDICE DE CONTENIDO.....	v
LISTA DE FIGURAS .....	x
Lista de tablas .....	xii
RESUMEN .....	xiii
INTRODUCCIÓN .....	xiv
CAPÍTULO 1 .....	1
MARCO TEÓRICO.....	1
1.1 MICROCONTROLADOR.....	1
1.1.1 MICROPROCESADOR .....	2
1.1.1.1 Funcionamiento.....	2
PreFetch.....	3
Fetch .....	3
Decodificación de la instrucción .....	3
Lectura de operandos .....	3
Ejecución.....	3
Escritura .....	3
1.1.2 DIFERENCIAS ENTRE MICROPROCESADOR Y MICROCONTROLADOR .....	3
1.1.3 ARQUITECTURA DE LOS PROCESADORES.....	4
1.1.3.1 Arquitectura Von Neumann. ....	4
1.1.3.2 Arquitectura Harvard. ....	5
1.2 SENSORES .....	6
1.2.1 CARACTERÍSTICAS GENERALES DE LOS SENSORES.....	6
1.2.1.1 Características estáticas. ....	6
a) Rango de medida:.....	6
b) Resolución: .....	6
c) Precisión: .....	6

d) Repetibilidad:.....	7
e) Linealidad:.....	7
f) Sensibilidad.....	7
g) Ruido:.....	7
1.2.1.2 Características dinámicas.....	7
a) Velocidad de respuesta:.....	7
b) Respuesta en frecuencia:.....	7
c) Estabilidad:.....	7
1.2.2 Sensores de Temperatura.....	7
1.2.3 Termistores: .....	8
1.2.4 RTD:.....	8
1.2.5 Termopar.....	9
1.3 LCD.....	9
1.3.1 FUNCIONAMIENTO.....	10
1.4 BUZZER.....	10
1.5 TECLADO .....	12
1.6 CIRCUITO DE RELOJ EN TIEMPO REAL.....	13
1.7 COMUNICACIÓN I2C .....	14
1.7.1 Descripción de las señales.....	14
1.7.2 Los estados del bus I2C.....	14
1.7.2.1 Libre.....	15
1.7.2.2 Inicio.....	15
1.7.2.3 Cambio.....	15
1.7.2.4 Dato.....	16
1.7.2.5 Parada.....	16
1.7.3 Protocolo de comunicación del bus I2C .....	16
1.7.3.1 Definición de términos.....	16
1.7.3.1.1 Maestro (Master): .....	16
1.7.3.1.2 Esclavo (Slave): .....	17
1.7.3.1.3 Bus libre (Bus Free):.....	17
1.7.4 Formato de una transacción.....	17
1.8 CIRCUITO CONTROLADOR DE POTENCIA.....	17
1.8.1 Triac.....	18
1.8.1.1 Característica tensión – corriente.....	19

1.8.2 Aplicaciones .....	19
1.9 OPTOACOPLADORES .....	20
1.9.1 Funcionamiento del Optoacoplador .....	20
1.9.2 Diferentes tipos de Optoacopladores .....	21
1.9.3 Aplicaciones con optoacopladores .....	21
1.10 LUCES ARTIFICIALES PARA PECERA .....	22
1.10.1 Características de la luz en un proceso biológico. ....	23
1.10.1.1 Consejos de iluminación.....	23
1.11 CALENTADORES .....	24
1.11.1 Tipos de calentadores .....	25
1.11.2 Colocación del calentador .....	25
1.11.3 potencia del calentador .....	26
1.12 LENGUAJE DE PROGRAMACIÓN.....	26
1.12.1 LENGUAJE ENSAMBLADOR .....	27
1.12.2 Ventajas de lenguajes de programación de alto nivel .....	28
1.12.3 Lenguaje C .....	30
1.12.4 CARACTERÍSTICAS PRINCIPALES DEL MIKROC.....	31
1.12.5 FASES DE COMPILACIÓN.....	31
CAPÍTULO 2 .....	33
CONSTRUCCIÓN DEL HARDWARE .....	33
2.1 DESCRIPCIÓN GENERAL .....	33
2.1.1 CONTROL DE LA LUZ ARTIFICIAL.....	33
2.1.1.1 Configuración de actualización de la hora de reloj .....	34
2.1.2 CONTROL DE CALENTADOR DE AGUA .....	35
2.1.2.1 Configuración de temperatura .....	36
2.2 DESCRIPCIÓN POR BLOQUES .....	36
2.2.1 DESCRIPCIÓN DEL BLOQUE MICROCONTROLADOR .....	37
2.2.1.1 PIC 16F877A.....	37
a. Arquitectura interna .....	38
b. Configuración de pines PIC16F877A .....	38
c. El oscilador externo.....	40
d. Reset.....	41
e. Distribución de pines por bloques .....	42
f. Módulo del convertidor Analógico a Digital .....	43



2.2.2 DESCRIPCIÓN DEL BLOQUE SENSOR DE TEMPERATURA.....	45
2.2.2.1 LM35 .....	45
2.2.2.2 Filtración de la señal del LM35.....	46
2.2.2.3 Impermeabilización del LM35.....	47
2.2.3 DESCRIPCIÓN DEL BLOQUE CIRCUITO DE RELOJ .....	48
2.2.3.1 DS1307 .....	48
a. Descripción. ....	48
b. Características. ....	49
c. Descripción de la señal. ....	50
2.2.4 DESCRIPCIÓN DEL BLOQUE VISUALIZADOR .....	51
2.2.4.1 LCD 16X2.....	51
a. Descripción: .....	51
b. Características principales: .....	52
c. Descripción de pines .....	52
2.2.5 DESCRIPCIÓN DEL BLOQUE TECLADO.....	52
2.2.5.1 Pulsador .....	53
2.2.5.2 Teclado matricial .....	54
2.2.6 DESCRIPCIÓN DEL BLOQUE SWITCHING .....	54
2.2.6.1 Acoplamientos de la señal para el control .....	54
2.2.6.1.1 CI MOC 3010 .....	55
2.2.6.1.2 TRIAC BT136 .....	55
2.2.6.2 Descripción del bloque luz artificial .....	56
2.2.6.3 Descripción del bloque calentador de agua.....	57
2.3 CIRCUITOS DEL PROTOTIPO.....	57
CAPÍTULO 3 .....	61
DESARROLLO DE LA PROGRAMACIÓN.....	61
3.1 FUNCIONAMIENTO DEL PROGRAMA POR DIAGRAMA DE FLUJO .....	61
3.1.1 DIAGRAMA DE FLUJO DE OPERACIÓN NORMAL .....	61
3.1.2 DIAGRAMA DE FLUJO DE CONFIGURACIÓN DE TEMPERATURA.....	63
3.1.3 DIAGRAMA DE FLUJO DE ENCENDIDO DE LUZ MANUAL .....	64
3.1.4 DIAGRAMA DE FLUJO DE CONFIGURACIÓN DE LA HORA .....	65
3.2 CÓDIGO DEL PROGRAMA.....	69
3.2.1 LÍNEAS DE CÓDIGO .....	69
3.2.1.1 Asignación de variables.....	74

3.2.1.2 Comunicación con I2C al DS1307 .....	76
3.2.1.3 Comunicación con el LCD .....	76
3.2.1.4 Lecturas de los datos del DS1307 .....	76
3.2.1.5 Codificación de datos leídos del DC1307 .....	76
3.2.1.6 Presentación de la hora minutos y segundos en el LCD .....	77
3.2.1.7 Programas secundarios.....	77
3.2.1.8 Datos del sensor de temperatura .....	77
3.2.1.9 Sonidos aviso .....	78
3.2.1.10 Barrido del teclado.....	78
3.2.1.11 Asignación del valor de comparación de temperatura.....	79
3.2.1.12 Asignación de los valores de la hora .....	79
3.2.1.13 Lógica de comparaciones.....	80
3.2.1.14 Encendido manual de la luz artificial .....	80
3.1.1.15 Funcionamiento del programa principal.....	80
CAPÍTULO 4 .....	81
PRUEBAS Y RESULTADOS.....	81
CAPÍTULO 5 .....	84
CONCLUSIONES Y RECOMENDACIONES .....	84
5.1 CONCLUSIONES.....	84
5.2 RECOMENDACIONES .....	85
Bibliografía .....	86

## LISTA DE FIGURAS

Figura 1.1 - Estructura interna de un microcontrolador. ....	2
Figura 1.2 - Microprocesador. ....	3
Figura 1.3 - Arquitectura Von Neumann.....	4
Figura 1.4 - Arquitectura Harvard.....	5
Figura 1.5 - Sensores de temperatura.....	8
Figura 1.6 – Buzzer. ....	11
Figura 1.7 - Símbolo electrónico del buzzer. ....	11
Figura 1.8 - Esquema típico de un teclado matricial.....	12
Figura 1.9 - Condición de bus Libre. ....	15
Figura 1.10 - Condición de inicio.....	15
Figura 1.11 - Condiciones o estados de cambio y de dato.....	15
Figura 1.12 - Condición de parada. ....	16
Figura 1.13 - Símbolo electrónico del Triac.....	18
Figura 1.14 - Curva característica del Triac.....	19
Figura 1.15 - Manejando un SCR.....	21
Figura 1.16 – Ejemplo de un acuario.....	22
Figura 1.17- Calentador de agua.....	24
Figura 1.18 - Sistema hexadecimal en el microcontrolador.....	27
Figura 1.19 - Instrucciones del ensamblador.....	28
Figura 1.20 - Lenguaje C compilado en ensamblador.....	29
Figura 1.21- compilacion de un lenguaje de alto nivel a bajo nivel.....	30
Figura 1.22 - Pasos de complilacion del lenguaje C.....	31
Figura 1.23 - Proceso de compilación del lenguaje C. ....	32
Figura 2.1- Diagrama de bloques del proyecto.....	33
Figura 2.2 – Representación tecla hora. ....	35
Figura 2.3 – Representación tecla temperatura. ....	36
Figura 2.4 - Distribución de pines PIC 16F877A .....	38
Figura 2.5- Arquitectura del microcontrolador PIC 16F877A.....	39
Figura 2.6-Conexión oscilador HS y los capacitores .....	40

Figura 2.7-Conexión del botón reset .....	41
Figura 2.8 - Circuito de conexión de los elementos al microcontrolador .....	44
Figura 2.9 - Distribución de pines LM35 vista inferior.....	46
Figura 2.10 - Circuito de filtro para el LM35 .....	47
Figura 2.11 - Circuito del sensor LM35 y filtro impermeabilizado .....	47
Figura 2.12 – Distribución de pines DS1307 .....	49
Figura 2.13 – LCD 16x2. ....	51
Figura 2.14 – Teclado y pulsadores del bloque teclado. ....	53
Figura 2.15 - Funcionamiento pulsador.....	53
Figura 2.16 – Distribución de pines del C.I. MOC3010. ....	55
Figura 2.17 – Distribución de pines del TRIAC BT136. ....	56
Figura 2.18 – Lámpara fluorescente.....	57
Figura 2.19 – Calentador de agua.....	57
Figura 2.20 - Pistas de la baquelita principal.....	58
Figura 2.21 - Baquelita principal.....	59
Figura 2.22 - Pistas del bloque switching .....	60
Figura 2.23 - Baquelita bloque switching.....	60
Figura 3.1 – Pantalla LCD en función normal.....	62
Figura 3.2 – Pantalla del menú de configuración de temperatura .....	64
Figura 3.3 – Pantalla del menú de configuración de la hora. ....	65
Figura 3.4 – Diagrama de flujo primera parte (a) Inicio (b) Ciclo principal. (c) Inicio de menús y menú encendido de lámpara. ....	66
Figura 3.5 – Diagrama de flujo menú configuración temperatura.....	67
Figura 3.6 – Diagrama de flujo menú configuración de la hora. ....	68
Figura 3.7 – Teclado 4x3.....	79

## LISTA DE TABLAS

Tabla 2.1- Características del microcontrolador utilizado .....	37
Tabla 2.2- Pines especiales del microcontrolador .....	40
Tabla 2.3- Distribucion de pines del microcontroladro por bloques de funcionamiento .....	42
Tabla 2.4 - Distribución de pines LCD 16x2. ....	52
Tabla 2.5 – Configuración de los pines del TRIAC BT136 .....	56
Tabla 3.1 – Variables utilizadas en las líneas del código .....	74
Tabla 3.2 - Caracteres utilizados en la primera fila del LCD 16x2.....	77
Tabla 3.3 - Caracteres utilizados en la segunda fila del LCD 16x2 .....	78
Tabla 4.1- Comportamiento de la lámpara .....	81
Tabla 4.2- Comportamiento del calentador de agua .....	82

## **RESUMEN**

El desarrollo de un prototipo de detección y control automático de temperatura e iluminación en un acuario se basa en la evaluación de la temperatura del agua y la determinación de la hora del día, con un sensor de temperatura para controlar el calentador, un reloj en tiempo real para el control de iluminación del acuario respectivamente. El proyecto es dividido en 5 capítulos: El primer capítulo es el análisis conceptual de todos los elementos que forman parte del prototipo, como son un microcontrolador que se encargará de ser el cerebro del proyecto, un sensor capaz de detectar la temperatura del agua, un LCD que informará de manera gráfica los valores de las variables de control, un buzzer capaz de generar una advertencia en casos impropios al correcto funcionamiento, un teclado facilitando elegir los parámetros de control convenientes, un reloj en tiempo real, para saber qué hora del día es y realizar la iluminación, y elementos capaces de acoplar todo al sistema de control. El segundo capítulo es la descripción específica de los dispositivos electrónicos a ser usados en la implementación del hardware del proyecto. El tercer capítulo presenta el desarrollo del programa encargado de realizar el manejo de los circuitos implementados. El cuarto capítulo indica las pruebas realizadas, comprobando que funciona correctamente y que cumple el objetivo principal de este proyecto. El quinto capítulo presenta las conclusiones y recomendaciones.

## INTRODUCCIÓN

En la acuariofilia, la gran mayoría de los peces tropicales, requieren temperaturas estables y superiores a temperaturas templadas, como la de Quito, y también, por beneficio del ecosistema, requieren 12 horas de luz y 12 horas de sombra. En la actualidad existen diversos tipos de calefactores para calentar el agua, que por el tipo de funcionamiento de estos y el ajuste del termostato del calefactor hasta la temperatura deseada (manualmente) y usando un termómetro, se pueden considerar que son muy inestables y no confiables o tiene componentes mecánicos que puedan desgastarse se vuelvan potencialmente inexactos con el uso, y además de verificar la temperatura del agua diariamente.

También debe iluminarse el acuario durante al menos 10 horas, hasta un máximo de 12 horas, tanto para los peces como para las plantas especialmente, con luz artificial adecuada debido a que la luz natural directa pueden llevar a una rápida proliferación de algas en el acuario.

Si se sigue controlando manualmente este tipo de elementos, habrá momentos, que por el descuido humano, en que no son controlados, tiendan a no solventar la idoneidad del acuario para los peces que vivirán en su interior.

Mediante la construcción de un prototipo de detección y control electrónico de temperatura e iluminación para el ecosistema de un acuario de peces tropicales en un clima templado se cuenta con un equipo con un ajuste automático, seguro, preciso, rápido y confiable, liberando de la preocupación del control continuo a la persona que practica la acuariofilia con los peces tropicales en el clima templado de Quito y manteniendo apto el ambiente del acuario para los peces.

Todo esto es realizado por un elemento idealmente “inteligente”, como lo es un microcontrolador PIC, aplicando lógicas reales y controles electrónicos, generados por un lenguaje de programación de alto nivel.

# CAPÍTULO 1

## MARCO TEÓRICO

### 1.1 MICROCONTROLADOR

Un microcontrolador es un dispositivo electrónico capaz de llevar a cabo procesos lógicos que cuentan con un microprocesador o unidad de procesamiento central (CPU), una memoria para almacenar el programa, una memoria para almacenar datos y puertos de entrada salida.

El funcionamiento de los microcontroladores está determinado por el programa almacenado en su memoria. Este puede escribirse en distintos lenguajes de programación.

La memoria generalmente está constituida por memoria RAM compuesta por registros que almacena datos temporales, memoria EEPROM para el almacenamiento del programa que se debe ejecutar.

Las unidades de entrada/salida se refieren a los puertos que tiene el micro para recibir o enviar datos en forma serie o en forma paralela. Cuenta además con módulos especiales para convertir señales analógicas a digitales o de digitales a analógicas.<sup>1</sup>

Los parámetros más importantes en un microcontrolador son:

- Bus de datos: 8, 16, 32 bits
- Capacidad de memoria: Tamaño de la memoria RAM y de la memoria EEPROM en kilobytes KB
- Velocidad: Numero de instrucciones a ejecutar por segundo. Depende de la frecuencia del oscilador del micro <sup>2</sup>.

---

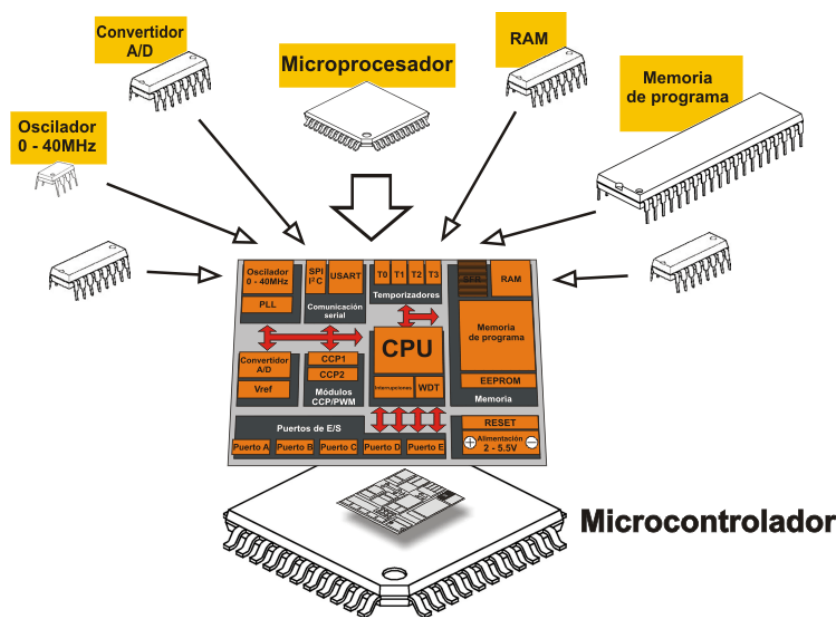
<sup>1</sup> Ceduvirt. (s.f.). Obtenido de <http://www.ceduvirt.com/resources/Microcontroladores.pdf>

<sup>2</sup> Torriti, M. T. (s.f.). Obtenido de [http://web.ing.puc.cl/~mtorrest/downloads/pic/tutorial\\_pic.pdf](http://web.ing.puc.cl/~mtorrest/downloads/pic/tutorial_pic.pdf)



- Puertos: Puertos de entrada salida de forma paralela y serial para comunicación externa.<sup>3</sup>
- Módulos: Para conversión A/D, D/A, PWM, USB, CAN, I2C, SPI, UART, USART, etc.

En la figura 1.1 se indica la estructura interna de un microcontrolador.



**Figura 1.1** - Estructura interna de un microcontrolador.

## 1.1.1 MICROPROCESADOR

Un microprocesador, también conocido como procesador, micro, chip o microchip, es un circuito lógico que responde y procesa las operaciones lógicas y aritméticas que hacen funcionar a nuestras computadoras. En definitiva, es su cerebro.

### 1.1.1.1 Funcionamiento.

El microprocesador ejecuta instrucciones almacenadas como números binarios organizados secuencialmente en la memoria principal. La ejecución de las instrucciones se puede realizar en varias fases:

<sup>3</sup> S, P. A. (s.f.). *olimex*. Obtenido de <http://www.olimex.cl/tutorial/tutorial1.pdf>

*PreFetch*, pre lectura de la instrucción desde la memoria principal.

*Fetch*, envío de la instrucción al decodificador

*Decodificación de la instrucción*, es decir, determinar qué instrucción es y por tanto qué se debe hacer.

*Lectura de operandos* (si los hay).

*Ejecución*, lanzamiento de las máquinas de estado que llevan a cabo el procesamiento.

*Escritura* de los resultados en la **memoria principal** o en los registros. <sup>4</sup>

En la figura 1.2 se puede observar un ejemplo de microprocesador de computador.



**Figura 1.2** - Microprocesador.

### **1.1.2 DIFERENCIAS ENTRE MICROPROCESADOR Y MICROCONTROLADOR**

Un microcontrolador no es lo mismo que un microprocesador. Difieren uno del otro en muchos sentidos. La primera y la más importante diferencia es su funcionalidad. El microprocesador permite conectarle la memoria y los módulos de entrada y salida, estos dispositivos se conectan a los pines del microprocesador que ponen en el exterior las líneas de sus buses de direcciones, datos y control.

Un microcontrolador es un circuito integrado que contiene todos los elementos electrónicos que se utilizan para hacer funcionar un sistema basado con un microprocesador; es decir contiene en un solo integrado la Unidad de Proceso, la

---

<sup>4</sup> *INGENIATIC*. (s.f.). Obtenido de <http://ingeniatic.euitt.upm.es/index.php/tecnologias/item/519-microprocesador>

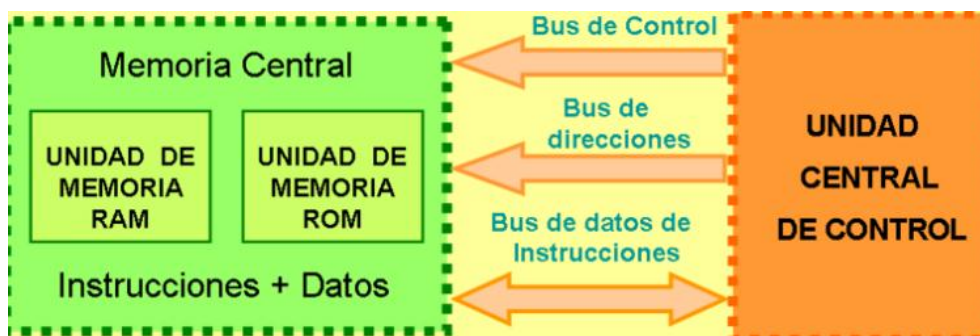
memoria RAM, memoria ROM , puertos de entrada, salidas y otros periféricos, con la consiguiente reducción de espacio.<sup>5</sup>

### 1.1.3 ARQUITECTURA DE LOS PROCESADORES

#### 1.1.3.1 Arquitectura Von Neumann.

En un sistema con arquitectura Von Neumann el tamaño de la unidad de datos o instrucciones está fijado por el ancho del bus que comunica la memoria con la CPU. Así un microprocesador de 8 bits con un bus de 8 bits, tendrá que manejar datos e instrucciones de una o más unidades de 8 bits (bytes) de longitud. Si tiene que acceder a una instrucción o dato de más de un byte de longitud, tendrá que realizar más de un acceso a la memoria.

En la figura 1.3 presenta una arquitectura Von Neumann que al tener un único bus hace que el microprocesador sea más lento en su respuesta, ya que no puede buscar en memoria una nueva instrucción mientras no finalicen las transferencias de datos de la instrucción anterior.



**Figura 1.3** - Arquitectura Von Neumann.

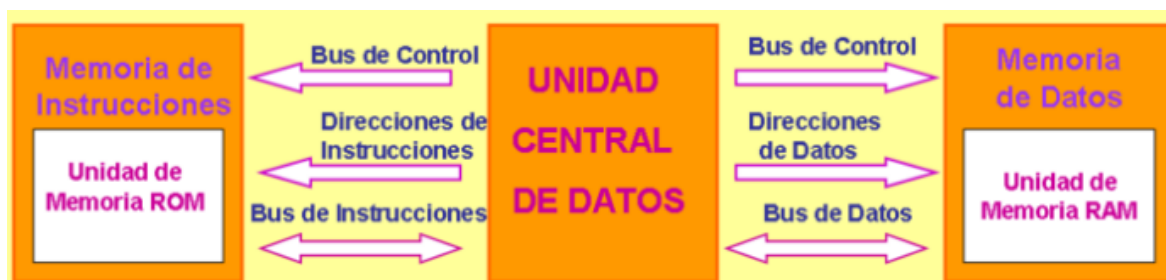
- Las principales limitaciones que nos encontramos con la arquitectura Von Neumann son:

<sup>5</sup> S, P. A. (s.f.). *olimex*. Obtenido de <http://www.olimex.cl/tutorial/tutorial1.pdf>

- La limitación de la longitud de las instrucciones por el bus de datos, que hace que el microprocesador tenga que realizar varios accesos a memoria para buscar instrucciones complejas.
- La limitación de la velocidad de operación a causa del bus único para datos e instrucciones que no deja acceder simultáneamente a unos y otras, lo cual impide superponer ambos tiempos de acceso.

### 1.1.3.2 Arquitectura Harvard.

Este modelo, que utilizan los microcontroladores PIC, como se observa en la figura 1.4, tiene la unidad central de proceso (CPU) conectada a dos memorias (una con las instrucciones y otra con los datos) por medio de dos buses diferentes.



**Figura 1.4 -** Arquitectura Harvard.

Una de las memorias contiene solamente las instrucciones del programa (Memoria de Programa), y la otra sólo almacena datos (Memoria de Datos).

Ambos buses son totalmente independientes lo que permite que la CPU pueda acceder de forma independiente y simultánea a la memoria de datos y a la de instrucciones. Como los buses son independientes éstos pueden tener distintos contenidos en la misma dirección y también distinta longitud. También la longitud de los datos y las instrucciones puede ser distinta, lo que optimiza el uso de la memoria en general.

Para un procesador de Set de Instrucciones Reducido, o RISC (Reduced Instrucción Set Computer), el set de instrucciones y el bus de memoria de programa pueden diseñarse de tal manera que todas las instrucciones tengan una sola posición de memoria de programa de longitud.

Además, al ser los buses independientes, la CPU puede acceder a los datos para completar la ejecución de una instrucción, y al mismo tiempo leer la siguiente instrucción a ejecutar.<sup>6</sup>

## 1.2 SENSORES

Un sensor es un dispositivo diseñado para recibir información de una magnitud del exterior y transformarla en otra magnitud, normalmente eléctrica, que seamos capaces de cuantificar y manipular.

### 1.2.1 CARACTERÍSTICAS GENERALES DE LOS SENSORES.

El sensor ideal es aquel cuya relación entre la magnitud de entrada y la magnitud de salida es proporcional.

Sin embargo, la respuesta real de los sensores nunca es del todo lineal, tiene un rango limitado de validez, suele estar afectada por perturbaciones del entorno exterior y tiene un cierto retardo en la respuesta.

Las características de los sensores se pueden agrupar en dos grandes bloques: Características estáticas, que describen la actuación del sensor en régimen permanente o con cambios muy lentos de la variable a medir.

Características dinámicas, que describen el comportamiento del sensor en régimen transitorio.

#### 1.2.1.1 Características estáticas.

- a) *Rango de medida*: el conjunto de valores que puede tomar la señal de entrada comprendidos entre el máximo y el mínimo detectados por el sensor con una tolerancia de error aceptable.
- b) *Resolución*: indica la capacidad del sensor para discernir entre valores muy próximos de la variable de entrada. Indica que variación de la señal de entrada produce una variación detectable en la señal de salida.
- c) *Precisión*: define la variación máxima entre la salida real obtenida y la salida teórica dada como patrón para el sensor.

---

<sup>6</sup> <http://www.fing.edu.uy/tecnoinf/mvd/cursos/arqcomp/material/teorico/arq-teorico05.pdf>

- d) *Repetitibilidad*: Indica la máxima variación entre valores de salida obtenidos al medir varias veces la misma entrada con el mismo sensor y en idénticas condiciones ambientales.
- e) *Linealidad*: un transductor es lineal si existe una constante de proporcionalidad única que relaciona los incrementos de la señal de salida con los respectivos incrementos de la señal de entrada en todo el rango de medida.
- f) *Sensibilidad*: indica la mayor o menor variación de la señal de salida por unidad de la magnitud de entrada. Cuanto mayor sea la variación de la señal de salida producida por una variación en la señal de entrada, el sensor es más sensible.
- g) *Ruido*: cualquier perturbación aleatoria del propio sistema de medida que afecta la señal que se quiere medir.

#### **1.2.1.2 Características dinámicas.**

- a) *Velocidad de respuesta*: mide la capacidad del sensor para que la señal de salida siga sin retraso las variaciones de la señal de entrada.
- b) *Respuesta en frecuencia*: mide la capacidad del sensor para seguir las variaciones de la señal de entrada a medida que aumenta la frecuencia, generalmente los sensores convencionales presentan una respuesta del tipo pasabajos.
- c) *Estabilidad*: indica la desviación en la salida del sensor con respecto al valor teórico dado, al variar parámetros exteriores distintos al que se quiere medir (condiciones ambientales, alimentación, etc.).<sup>7</sup>

#### **1.2.2 SENSORES DE TEMPERATURA.**

Los sensores de temperatura son dispositivos que transforman los cambios de temperatura en cambios de señales eléctricas que son procesados por equipo eléctrico o electrónico.

---

<sup>7</sup> Artemisa. (s.f.). Obtenido de <http://artemisa.unicauca.edu.co/~gavasquez/res/Sensores.pdf>

Como se ve en la figura 1.5, hay tres tipos de sensores de temperatura, los termistores, los RTD y los termopares.



**Figura 1.5** - Sensores de temperatura.

### **1.2.3 TERMISTORES:**

El termistor está basado en que el comportamiento de la resistencia de los semiconductores es variable en función de la temperatura.

Existen los termistores tipo NTC y los termistores tipo PTC. En los primeros, al aumentar la temperatura, disminuye la resistencia. En los PTC, al aumentar la temperatura, aumenta la resistencia.

El principal problema de los termistores es que no son lineales según la temperatura por lo que es necesario aplicar fórmulas complejas para determinar la temperatura según la corriente que circula y son complicados de calibrar.

### **1.2.4 RTD:**

Un RTD es un sensor de temperatura basado en la variación de la resistencia de un conductor con la temperatura.

Los metales empleados normalmente como RTD son platino, cobre, níquel y molibdeno.

De entre los anteriores, los sensores de platino son los más comunes por tener mejor linealidad, más rapidez y mayor margen de temperatura.

El elemento sensor es típicamente un fino alambre de platino o una delgada película del mismo material aplicada a un sustrato cerámico. Otros metales comúnmente

utilizados como elementos sensores son: el níquel, el cobre y el molibdeno. Las RTD, principalmente las versiones de platino, se caracterizan principalmente por su precisión y su amplio rango de temperaturas de operación, el cual se extiende desde  $-250^{\circ}\text{C}$  hasta  $+850^{\circ}\text{C}$ . Tienen también una sensibilidad, estabilidad y repetitividad muy altas, y ofrecen una respuesta.

### 1.2.5 TERMOPAR

El termopar, también llamado termocupla y que recibe este nombre por estar formado por dos metales, es un instrumento de medida cuyo principio de funcionamiento es el efecto termoeléctrico.

Un material termoeléctrico permite transformar directamente el calor en electricidad, o bien generar frío cuando se le aplica una corriente eléctrica.

El termopar genera una tensión que está en función de la temperatura que se está aplicando al sensor. Midiendo con un voltímetro la tensión generada, conoceremos la temperatura.

Los termopares tienen un amplio rango de medida, son económicos y están muy extendidos en la industria. El principal inconveniente estriba en su precisión, que es pequeña en comparación con sensores de temperatura RTD o termistores.<sup>8</sup>

Existen varios tipos de termocupla, puesto que cualquier par de metales conformaría un tipo determinado. Sin embargo, la empírica ha llevado al uso de ciertos tipos estandarizados, a los que se les cita por una letra (las más típicas son las tipo J, K). Cada tipo difiere en el material de los metales A y B. Al diferir los materiales de construcción difieren los rangos de trabajo, el voltaje generado por unidad de grado y la máxima temperatura útil (antes que se funda).<sup>9</sup>

## 1.3 LCD

Una pantalla de cristal líquido o LCD (liquid crystal display) es una pantalla delgada y plana formada por un número de píxeles en color o monocromos colocados delante de una fuente de luz o reflectora. A menudo se utiliza en dispositivos

---

<sup>8</sup> medirtemperatura.com. (s.f.). Obtenido de <http://medirtemperatura.com/sensor-temperatura.php>

<sup>9</sup> Bagueño, M. A. (s.f.). Obtenido de <http://snsoresdetemperatura.blogspot.com/>



electrónicos de pilas, ya que utiliza cantidades muy pequeñas de energía eléctrica.

### **1.3.1 FUNCIONAMIENTO**

El LCD modifica la luz que lo incide. Dependiendo de la polarización que se esté aplicando, el LCD reflejará o absorberá la luz.

Cuando un segmento recibe la tensión de polarización adecuada no reflejará la luz y aparecerá en la pantalla del dispositivo como un segmento oscuro. Un ejemplo de este fenómeno se puede visualizar en calculadoras, relojes, etc.

El líquido de un display LCD está entre dos placas de vidrio paralelas con una separación de unos micrones. Estas placas de vidrio tienen unos electrodos especiales que definen, con su forma, los símbolos, caracteres, etc. que se visualizarán.

La superficie del vidrio que hace contacto con el líquido es tratada de manera que induzca la alineación de los cristales en dirección paralela a las placas. Esta alineación permite el paso de la luz incidente sin ninguna alteración.

Cuando se aplica la polarización adecuada entre los electrodos, aparece un campo eléctrico entre estos electrodos (campo que es perpendicular a las placas) y esto causa que las moléculas del líquido se agrupen en sentido paralelo a este (el campo eléctrico) y causen que aparezca una zona oscura sobre un fondo claro (contraste positivo). De esta manera aparece la información que se desea mostrar.

## **1.4 BUZZER**

En la figura 1.6 se muestra un transductor electroacústico que produce un sonido o zumbido continuo o intermitente de un mismo tono. Sirve como mecanismo de señalización o aviso, y son utilizados en múltiples sistemas como en automóviles o en electrodomésticos, incluidos los despertadores.



Figura 1.6 – Buzzer.

El símbolo del buzzer se muestra en la figura 1.7. El corazón de los buzzer piezoeléctricos es un simple disco piezo, que consiste de una placa cerámica con una capa metálica. Si el disco es controlado por un circuito oscilante externo se habla de un transductor piezo eléctrico. Si el circuito oscilador está incluido en la carcasa, se le denomina zumbador piezoeléctrico. <sup>10</sup>



Figura 1.7 - Símbolo electrónico del buzzer.

La transducción o transformación de energía, se hace en dos fases. El modelo teórico de un transductor electroacústico, se basa en un transductor electromecánico y un transductor mecánico-acústico. Esto significa, que se estudia por un lado la transformación de la energía eléctrica en mecánica, ya que se genera un movimiento, y por otro lado se estudia la transformación de la energía mecánica en acústica, ya que el movimiento genera energía acústica. <sup>11</sup>

<sup>10</sup> reparo, Y. (s.f.). Obtenido de <http://electronica.yoreparo.com/electronica/1075528.html>

<sup>11</sup> Palace, B. (s.f.). Obtenido de <http://cosmicdevices.tradeindia.com/electronic-buzzers-179625.html>

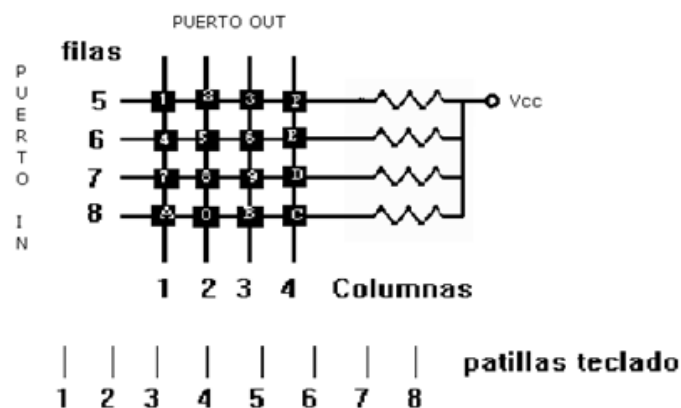
## 1.5 TECLADO

El teclado matricial proporciona una interfaz sencilla de entrada de datos. Sus usos pueden ser tan variados como aplicaciones. Desde una calculadora, pasando por una cerradura codificada, un termostato programable, etc.

Un teclado matricial es un arreglo de botones conectados en filas y columnas, de tal manera que se pueden leer varios botones con el mínimo número de entradas requeridas. Un teclado matricial 4x4 solamente ocupa 4 líneas de un puerto para leer las filas y otras 4 líneas para las columnas, de este modo se pueden leer 16 teclas utilizando solamente 8 entradas (un solo puerto) de un microcontrolador como se muestra en la figura 1.8.

El funcionamiento básico del teclado matricial consiste en rastrear la unión entre una fila y una columna, mediante un algoritmo que descifra cual tecla fue presionada.

En el mercado es habitual encontrar teclados matriciales pasivos de 3 x 4 y de 4 x 4 teclas. Esencialmente están constituidos por filas y columnas conductoras en cuyo cruce se encuentra un pulsador mecánico o de membrana que, al ser pulsado, establece el contacto eléctrico entre la fila y la columna correspondiente.<sup>12</sup>



**Figura 1.8** - Esquema típico de un teclado matricial.

Para su conexión al microcontrolador se deben colocar unas resistencias de *pull-up* en las filas (o las columnas). De esta forma si se pone a masa (cero lógico) una

<sup>12</sup> Alanoca, G. M. (s.f.). Scribd. Obtenido de <http://es.scribd.com/doc/217825101/Teclado-Matricial>

columna (fila) se obtendrá un cero en la fila (columna) correspondiente cuando se pulse una tecla.

Al pulsar pueden producirse rebotes mecánicos e interpretarlos como varias repeticiones de la misma tecla. Para evitarlo basta con esperar un tiempo superior a la duración de los mismos antes de seguir leyendo datos del teclado.<sup>13</sup>

## **1.6 CIRCUITO DE RELOJ EN TIEMPO REAL**

Se denomina reloj a un instrumento que permite medir el tiempo. Existen diversos tipos, y cada uno se adecua según el propósito, este es uno de los instrumentos más populares que existen actualmente.

La más importante y esencial de las funciones de un reloj común es proporcionar la hora actual, que debe ser la misma para todas las personas que viven en la misma zona horaria. Por convenio, el día se divide en veinticuatro horas. Cada hora comprende sesenta minutos y cada minuto comprende sesenta segundos. El segundo es la unidad básica de tiempo y se mide igual en todo el mundo, es decir, no existen diferentes unidades de tiempo en diversas regiones. Un reloj normal marca siempre la hora y los minutos, en algunos casos los segundos.

La exactitud del reloj depende de la base de tiempos, que puede consistir en un oscilador o en un adaptador que, a partir de una referencia, genera una señal periódica

El divisor de frecuencia es un circuito digital formado por una sucesión de contadores hasta obtener una frecuencia de 1 Hz, que permite mostrar segundos. Si se quiere mostrar décimas, la división se detiene al llegar a los 10 Hz. Esta frecuencia pasa al módulo de presentación, que puede ser electrónico o mecánico, donde otros divisores van separando los segundos, minutos y horas para presentarlas mediante algún tipo de display.<sup>14</sup>

---

<sup>13</sup>Alanoca, G. M. (s.f.). Scribd. Obtenido de <http://es.scribd.com/doc/217825101/Teclado-Matricial>

<sup>14</sup> Plan Ceibal. (s.f.). Obtenido de [http://www.ceibal.edu.uy/contenidos/areas\\_conocimiento/mat/midiendoeltiempo/el\\_reloj.html](http://www.ceibal.edu.uy/contenidos/areas_conocimiento/mat/midiendoeltiempo/el_reloj.html)

## **1.7 COMUNICACIÓN I2C**

El bus I2C, es un estándar que facilita la comunicación entre microcontroladores, memorias y otros dispositivos con cierto nivel de "inteligencia", sólo requiere de dos líneas de señal y un común. Permite el intercambio de información entre muchos dispositivos a una velocidad aceptable, de unos 100 Kbits por segundo, aunque hay casos especiales en los que el reloj llega hasta los 3,4 MHz.

La metodología de comunicación de datos del bus I2C es en serie y sincrónica. Una de las señales del bus marca el tiempo (pulsos de reloj) y la otra se utiliza para intercambiar datos.

### **1.7.1 DESCRIPCIÓN DE LAS SEÑALES**

SCL (System Clock) es la línea de los pulsos de reloj que sincronizan el sistema. SDA (System Data) es la línea por la que se mueven los datos entre los dispositivos. GND (Tierra) común de la interconexión entre todos los dispositivos "enganchados" al bus.

Las líneas SDA y SCL son del tipo drenaje abierto, es decir, un estado similar al de colector abierto, pero asociadas a un transistor de efecto de campo (o FET). Se deben polarizar en estado alto (conectando a la alimentación por medio de resistores "pull-up") lo que define una estructura de bus que permite conectar en paralelo múltiples entradas y salidas.<sup>15</sup>

### **1.7.2 LOS ESTADOS DEL BUS I2C.**

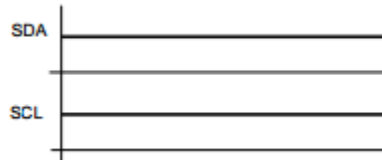
El bus I2C puede encontrarse en distintos estados, que la norma define como los siguientes:

---

<sup>15</sup> Carletti, E. J. (n.d.). Robots. Retrieved from [http://robots-argentina.com.ar/Comunicacion\\_busI2C.htm](http://robots-argentina.com.ar/Comunicacion_busI2C.htm)

### 1.7.2.1 Libre.

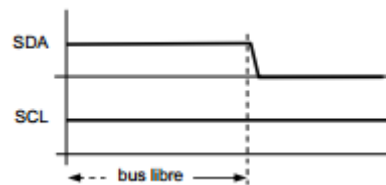
Este estado se caracteriza por encontrarse las líneas SDA y SCL a 1, como se ve en la figura 1.9, sin que se esté realizando ningún tipo de transferencia.



**Figura 1.9** - Condición de bus Libre.

### 1.7.2.2 Inicio.

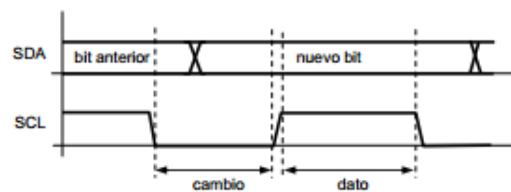
Se produce una condición de inicio cuando un maestro inicia una transacción. En concreto, consiste en un cambio de alta a baja en la línea SDA mientras SCL permanece a alta, mostrado en la figura 1.10.



**Figura 1.10** - Condición de inicio.

### 1.7.2.3 Cambio.

En la figura 1.11 se produce una condición de cambio cuando, estando a baja la línea SCL, la línea SDA puede cambiar de estado.



**Figura 1.11** - Condiciones o estados de cambio y de dato.

#### 1.7.2.4 Dato.

Este estado es el que, una vez iniciada una transacción, queda definido por la fase alta de la señal de sincronía SCL.

#### 1.7.2.5 Parada.

En la figura 1.12 se produce una condición de fin o parada cuando, estando la línea SCL a alta, se produce un cambio de baja a alta en la línea SDA.



**Figura 1.12** - Condición de parada.

### 1.7.3 PROTOCOLO DE COMUNICACIÓN DEL BUS I2C

Es lógico que para establecer una comunicación a través de él bus I2C se deba respetar un protocolo. En primer lugar, lo más importante es que: existen dispositivos maestros y dispositivos esclavos. Sólo los dispositivos maestros pueden iniciar una comunicación.

#### 1.7.3.1 Definición de términos

*1.7.3.1.1 Maestro (Master):* Dispositivo que determina los tiempos y la dirección del tráfico en el bus. Es el único que aplica los pulsos de reloj en la línea SCL. Cuando se conectan varios dispositivos maestros a un mismo bus la configuración obtenida se denomina "multi-maestro".<sup>16</sup>

<sup>16</sup> Fernández-Caparrós, A. M. (s.f.). Obtenido de <http://www.uco.es/~el1mofer/Docs/IntPerif/Bus%20I2C.pdf>

*1.7.3.1.2 Esclavo (Slave):* Todo dispositivo conectado al bus que no tiene la capacidad de generar pulsos de reloj. Los dispositivos esclavos reciben señales de comando y de reloj generados desde el maestro.

*1.7.3.1.3 Bus libre (Bus Free):* Estado en el que ambas líneas (SDA y SCL) están inactivas, presentando un estado lógico alto. Es el único momento en que un dispositivo maestro puede comenzar a hacer uso del bus.

#### **1.7.4 FORMATO DE UNA TRANSACCIÓN**

En el formato de transacción se cumple los siguientes criterios.

**- El carácter de ocho bits es la unidad de transferencia.**

El orden de emisión de los bits de un carácter es empezando por el más significativo, siguiendo en orden decreciente de pesos y terminando por el menos significativo (como se ve, es un criterio contrario al que siguen las UART)

**- Tras cada carácter se debe producir un reconocimiento por parte del receptor**

El motivo es dotar al protocolo de un mecanismo de seguridad. El primer carácter transferido lo emite siempre el maestro; sus siete bits más significativos indican la dirección del esclavo al que se dirige, y el bit de menor peso indica el sentido de la transferencia de los subsiguientes caracteres (0=escritura, 1=lectura, siempre desde el punto de vista del maestro).<sup>17</sup>

## **1.8 CIRCUITO CONTROLADOR DE POTENCIA**

Las facilidades que se tiene al controlar dispositivos con un microcontrolador como encender un led o controlar otros dispositivos electrónicos, son limitadas al momento de hacer un control de un elemento de potencia alta, es por eso que utiliza elementos capaces de soportar esa cantidad de potencia y adaptarla a la señal que emite el microcontrolador.

---

<sup>17</sup> Carletti, E. J. (s.f.). *Robots*. Obtenido de [http://robotsargentina.com.ar/Comunicacion\\_busI2C.htm](http://robotsargentina.com.ar/Comunicacion_busI2C.htm)



### 1.8.1 TRIAC.

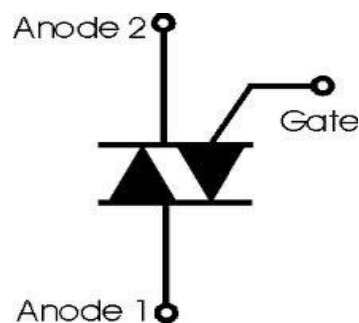
El Triac es un dispositivo semiconductor de tres terminales que se usa para controlar el flujo de corriente promedio a una carga, con la particularidad de que conduce en ambos sentidos y puede ser bloqueado por inversión de la tensión o al disminuir la corriente por debajo del valor de mantenimiento.

El Triac puede ser disparado independientemente de la polarización de puerta, es decir, mediante una corriente de puerta positiva o negativa.

Cuando el triac conduce, hay una trayectoria de flujo de corriente de muy baja resistencia de una terminal a la otra, dependiendo la dirección de flujo de la polaridad del voltaje externo aplicado.

Cuando el voltaje es más positivo en ánodo 2, la corriente fluye del ánodo 2 al ánodo 1 en caso contrario fluye del ánodo 1 al ánodo 2. En ambos casos el triac se comporta como un interruptor cerrado. Cuando el triac deja de conducir no puede fluir corriente entre las terminales principales sin importar la polaridad del voltaje externo aplicado por tanto actúa como un interruptor abierto.

En la figura 1.13 se representa a un triac que posee tres electrodos: A1, A2 (en este caso pierden la denominación de ánodo y cátodo) y puerta. El disparo del TRIAC se realiza aplicando una corriente al electrodo puerta.<sup>18</sup>

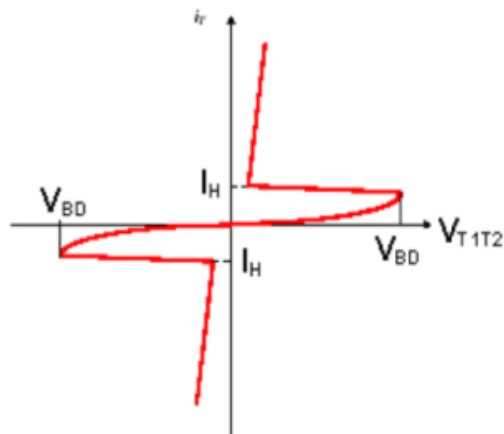


**Figura 1.13** - Símbolo electrónico del Triac.

<sup>18</sup> Jimenez, O. (s.f.). *Triac*. Obtenido de <http://triac.blogia.com/>

### 1.8.1.1 Característica tensión – corriente.

La figura 1.14 describe la característica tensión corriente del Triac. Muestra la corriente a través del Triac como una función de la tensión entre los ánodos 2 y 1. El punto  $V_{BD}$  (tensión de ruptura) es el punto por el cual el dispositivo pasa de una resistencia alta a una resistencia baja y la corriente, a través del Triac, crece con un pequeño cambio en la tensión entre los ánodos.



**Figura 1.14** - Curva característica del Triac

El Triac permanece en estado ON hasta que la corriente disminuye por debajo de la corriente de mantenimiento  $I_H$ . Esto se realiza por medio de la disminución de la tensión de la fuente. Una vez que el Triac entra en conducción, la compuerta no controla más la conducción, por esta razón se acostumbra dar un pulso de corriente corto y de esta manera se impide la disipación de energía sobrante en la compuerta.<sup>19</sup>

### 1.8.2 APLICACIONES.

- Su versatilidad lo hace ideal para el control de corrientes alternas.
- Una de ellas es su utilización como interruptor estático ofreciendo muchas ventajas sobre los interruptores mecánicos convencionales y los relés.
- Funciona como switch electrónico y también a pila.

<sup>19</sup> Jimenez, O. (s.f.). *Triac*. Obtenido de <http://triac.blogia.com/>

- Se utilizan TRIACs de baja potencia en muchas aplicaciones como atenuadores de luz, controles de velocidad para motores eléctricos, y en los sistemas de control computarizado de muchos elementos caseros. No obstante, cuando se utiliza con cargas inductivas como motores eléctricos, se deben tomar las precauciones necesarias para asegurarse que el TRIAC se apaga correctamente al final de cada semiciclo de la onda de Corriente alterna.<sup>20</sup>

## 1.9 OPTOACOPLADORES

Los Optoacopladores son dispositivos que podemos encontrar en múltiples aplicaciones dentro de un equipo electrónico, cuando una señal debe ser transmitida desde un circuito específico a otro, *sin que exista conexión eléctrica entre ambos*.

Un optoacoplador combina un dispositivo semiconductor formado por un fotoemisor, un fotoreceptor y entre ambos hay un camino por donde se transmite la luz. Todos estos elementos se encuentran dentro de un encapsulado que por lo general es del tipo DIP o Dual in-line package, una forma de encapsulamiento común en la construcción de circuitos integrados.

### 1.9.1 FUNCIONAMIENTO DEL OPTOACOPLADOR

La señal de entrada es aplicada al fotoemisor y la salida es tomada del fotoreceptor. Los optoacopladores son capaces de convertir una señal eléctrica en una señal luminosa modulada y volver a convertirla en una señal eléctrica. La gran ventaja de un optoacoplador reside en el aislamiento eléctrico que puede establecerse entre los circuitos de entrada y salida.

Los fotoemisores que se emplean en los optoacopladores de potencia son diodos que emiten rayos infrarrojos (IRED) y los fotoreceptores pueden ser tiristores o transistores.

---

<sup>20</sup> Biomédica, E. E. (s.f.). *Electrónica de Potencia*. Obtenido de <http://www.udb.edu.sv/udb/archivo/guia/electronica-tecnologico/electronica-de-potencia/2014/i/guia-4.pdf>

Cuando aparece una tensión sobre los terminales del diodo IRED, este emite un haz de rayos infrarrojo que transmite a través de una pequeña guía-ondas de plástico o cristal hacia el fotorreceptor. La energía luminosa que incide sobre el fotorreceptor hace que este genere una tensión eléctrica a su salida. Este responde a las señales de entrada, que podrían ser pulsos de tensión.

### 1.9.2 DIFERENTES TIPOS DE OPTOACOPLADORES

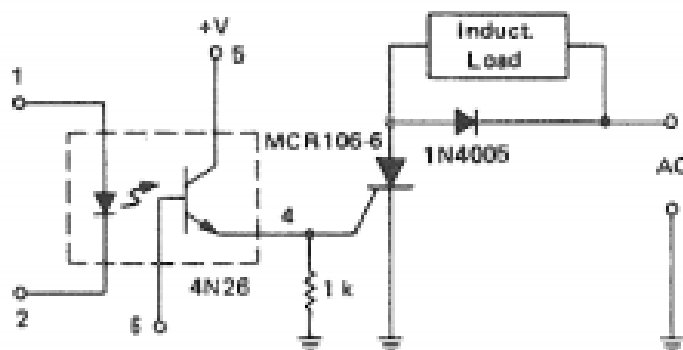
*Fototransistor:* se compone de un optoacoplador con una etapa de salida formada por un transistor BJT.

*Fototriac:* se compone de un optoacoplador con una etapa de salida formada por un triac

*Fototriac de paso por cero:* Optoacoplador en cuya etapa de salida se encuentra un triac de cruce por cero. El circuito interno de cruce por cero conmuta al triac sólo en los cruces por cero de la corriente alterna.

### 1.9.3 APLICACIONES CON OPTOACOPLADORES

Los modos básicos de operación de los optoacopladores son: por pulsos y lineal, en pulsos el LED se switchea on-off. En el modo lineal, la entrada es polarizada por una fuente de directa, y la entrada se monta a esta señal. En la figura 1.15 se ve un ejemplo de aplicación del optoacoplador.



**Figura 1.15** - Manejando un SCR.

## 1.10 LUCES ARTIFICIALES PARA PECERA

A la hora de instalar una pecera o acuario, como el de la figura 1.16, la luz es uno de los elementos más importantes, lo primero que se tiene que tomar en cuenta antes de iluminar la pecera es el tipo de animales que lo va a habitar. Este elemento es indispensable ya que es fuente de vida y genera la cadena alimentaria que sustenta a este tipo de ecosistemas. Por ello se debe tener mucho cuidado a la hora de elegir el tipo de iluminación a instalar.



**Figura 1.16** – Ejemplo de un acuario

Cuando se tiene una mala iluminación del acuario en el tanque de peces, los peces de bellos colores no se puede ver correctamente y puede incluso desaparecer. Para la iluminación del acuario se puede elegir entre las luces fluorescentes e incandescentes. Luces fluorescentes son, básicamente, las luces ultravioletas. Las luces incandescentes son como las luces que la mayoría de la gente utiliza en los hogares.

Si la iluminación del acuario es demasiado brillante puede causar que el agua se vea muy verde. Si no hay suficiente iluminación la vida de las plantas pueden tener problemas de crecimiento y de vida. También puede decidir si desea utilizar la iluminación natural o artificial, o puede optar por utilizar los dos.<sup>21</sup>

---

<sup>21</sup> leelo.co. (s.f.). Obtenido de [http://leelo.co/acuario-iluminación-es-importante-para-la-salud-de-tu-amigo-con-aletas-de\\_13b68f.html](http://leelo.co/acuario-iluminación-es-importante-para-la-salud-de-tu-amigo-con-aletas-de_13b68f.html)

### **1.10.1 CARACTERÍSTICAS DE LA LUZ EN UN PROCESO BIOLÓGICO.**

La luz viaja en el vacío a 300.000 kilómetros por segundo. En la atmósfera, desciende un poco. Sin embargo, al ser el agua un medio 1000 veces más denso que el aire, la velocidad en el medio acuoso de la luz desciende a 200.000 km/s. Esto se debe a que la luz penetra en el agua 2000 veces peor que en el aire. La luz visible que se recibe del Sol va desde el rojo (700nm), hasta el violeta (400nm). No obstante, en el medio marino las ondas más cortas y más largas son atenuadas rápidamente ya que el agua es muy opaca con respecto al ultravioleta y especialmente a la luz roja, pero bastante transparente a la luz azul y verde. Esta característica hace que en los océanos conforme se va profundizando, se van convirtiendo en monocromáticos. Esta característica mencionada tiene especial importancia ya que en los fondos coralinos la luz que impera de forma mayoritaria es la luz azul. Por debajo de los 10 a 15 primeros metros de profundidad, prácticamente nada de rojo y amarillo queda, siendo la luz imperante únicamente la verde y mayoritariamente la azul.<sup>22</sup>

#### **1.10.1.1 Consejos de iluminación**

La mayoría de las plantas requiere 12 horas diarias de iluminación o luz de tubos fluorescentes. Los cambios repentinos de luz, pueden estresar a los peces. Cuando encienda o apague las luces de la cubierta, es beneficioso tener encendidas las luces de la habitación por al menos 30 minutos. Los peces alimentados de día deben tener 30 minutos de luz antes y después de alimentarse.

Se utiliza sincronizadores cuando sea posible, las plantas y peces responden mejor a los períodos consistentes de luz. Plantas y peces se adaptan a los cambios de luz. Cuando se cambie los tubos en una instalación múltiple, se debe hacerlo escalonadamente con una o dos semanas de diferencia. El balastro remoto debe ser montado en áreas donde la ventilación sea adecuada para disipar el calor eficazmente. Esto es especialmente importante en los tipos de balastro que generan más calor. El cableado eléctrico que vaya al balastro siempre debe tener un bucle de goteo. Se debe considerar la instalación de una barra de potencia con

---

<sup>22</sup> Acuario Marino. (s.f.). Obtenido de <http://www.acuariomarino.com.ar/es/iluminacion.html>

interruptor de fallas a tierra o regleta eléctrica como una medida de seguridad de bajo costo para prevenir posibles accidentes eléctricos.

No se debe dejar las luces encendidas las 24 horas del día, como en la naturaleza los peces y plantas también necesitan períodos de oscuridad.

## 1.11 CALENTADORES

La temperatura del acuario es vital para el desarrollo y el mantenimiento correcto de los animales y plantas que habitan en el ecosistema marino.

La gran mayoría de los peces tropicales, que se ponen en un acuario pueden vivir de manera correcta en un acuario con una temperatura del agua que oscila entre los 24 y 28 grados centígrados. Incluso algunos animales puedan necesitar aguas más cálidas que esta, por lo que es importante que se conozca los peces que se tiene ahí adentro y sus necesidades.

Es importante que se tenga en cuenta que para lograr la temperatura necesaria, para el desarrollo correcto de los animales acuáticos es necesario un calentador o un calefactor que se pueda graduar para lograr la temperatura requerida tal como la figura 1.17.<sup>23</sup>



**Figura 1.17-** Calentador de agua

---

<sup>23</sup> Saldarriaga, V. (2 de enero de 2012). De Peces. Obtenido de <http://www.depeces.com/tipos-de-calentadores-para-acuarios.html>

Los calentadores de agua son utilizados comúnmente en todo tipo de acuario, y su uso, aplicación y beneficios son conocidos por todos en ese sentido, por lo que no requiere que se le haga mayor presentación. Los calentadores tradicionales son de vidrio o cerámica y son muy útiles para acuarios de peces tropicales, discos y goldfish, en donde no se requiere de mucha precisión o seguridad dentro del sistema. En el caso de un acuario de arrecife es necesario adoptar mayores medidas de seguridad y sería bastante deseable tener un control más preciso sobre la temperatura.<sup>24</sup>

Para el mantenimiento de peces tropicales o de agua caliente es completamente imprescindible el calentador. Este aparato consiste en una resistencia conectada a un termostato. Dicho termostato se activa el calentador cuando la temperatura del agua es más baja de la que hayamos indicado.

El calentador no es un calefactor que regula la temperatura a modo de climatizador, sino que evita que la temperatura del acuario baje por debajo de un límite peligroso. Si se va a mantener una especie que soporta un rango de 24° C a 28°C, se tiene que poner entonces el calentador a 24°C.

Esto es muy útil durante todo el año para evitar caídas de temperatura especialmente durante la noche y en invierno es un salvavidas para los animales.

### **1.11.1 TIPOS DE CALENTADORES**

Existen dos tipos: los sumergibles y los parcialmente sumergibles. Los primeros permanecen siempre dentro del agua y son los más usuales y cómodos. Los parcialmente sumergibles consisten en un tubo en donde va la resistencia y los controles quedan fuera del agua. No hay que tomar este dato de forma literal para identificarlos, ya que pueden ser muy parecidos unos a otros, hay que consultar las especificaciones en la caja.

### **1.11.2 COLOCACIÓN DEL CALENTADOR**

Cuanto más cerca del fondo mejor y si se puede poner en horizontal mejor todavía. Esto es por la siguiente razón. Cuando se cambia el agua el calentador puede llegar

---

<sup>24</sup> Acuario de Arrecife. (s.f.). Obtenido de <http://www.acuariodearrecife.com/parametros/temperatura-2.htm>



a tener una parte en el agua y otra parte fuera, y debido a la diferencia de grados con el aire, puede estallar si está encendido. Por lo tanto, siempre hay que desenchufar el calentador antes de un cambio de agua. Si se lo pone en el fondo y de forma horizontal se reducirá este riesgo, ya que es muy inusual que por un descuido vaciar tanto el acuario.

### **1.11.3 POTENCIA DEL CALENTADOR**

Existen en el mercado varias opciones de potencia disponible, 25 w, 50 w, 150 w, 200 w etc. En condiciones de habitaciones templadas, esto es que la temperatura exterior no bajará de 7 grados sobre el acuario, lo normal es contar con un watio por litro.

Esto es en un acuario de 50 litros en un ambiente templado el calentador será de 50 watt. Si la temperatura suele bajar mucho, entonces se tiene que considerar el proporcionar dos y hasta tres wattios por litro.

Si se puede entonces dividir la potencia deseada entre varios calentadores mucho mejor. Es preferible tener 3 de 100 watt. que 1 de 300 watt. Por qué, muy sencillo, si uno se estropea los demás pueden evitar el desastre y el desgaste de los componentes es menor si el esfuerzo es compartido.

De la misma forma, puede situar así varios calentadores camuflados en varias partes del acuario y no concentrarlos en un sólo sitio.<sup>25</sup>

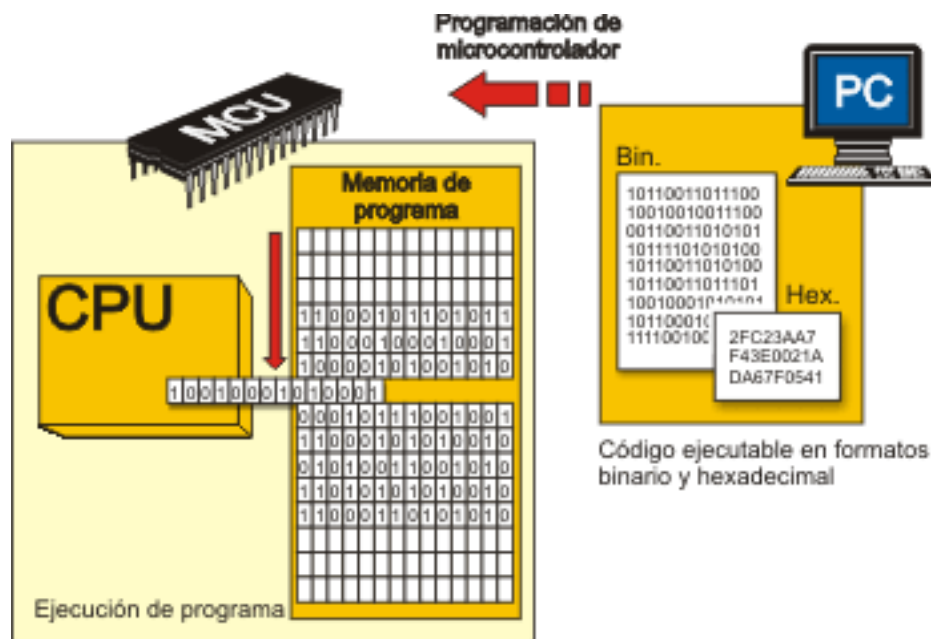
## **1.12 LENGUAJE DE PROGRAMACIÓN**

El microcontrolador ejecuta el programa cargado en la memoria Flash. Esto se denomina el código ejecutable y está compuesto por una serie de ceros y unos, aparentemente sin significado. Dependiendo de la arquitectura del microcontrolador, el código binario está compuesto por palabras de 12, 14 o 16 bits de anchura. Cada palabra se interpreta por la CPU como una instrucción a ser ejecutada durante el funcionamiento del microcontrolador. Todas las instrucciones que el microcontrolador puede reconocer y ejecutar se les denominan

---

<sup>25</sup> Aquaterraria.com. (20 de enero de 2011). Obtenido de <http://www.aquaterraria.com/content/section/197-calentadores.html>

colectivamente Conjunto de instrucciones. Como es más fácil trabajar con el sistema de numeración hexadecimal, el código ejecutable se representa con frecuencia como una serie de los números hexadecimales denominada código Hex, como se puede ver ejemplarizado en la figura 1.18. En los microcontroladores PIC con las palabras de programa de 14 bits de anchura, el conjunto de instrucciones tiene 35 instrucciones diferentes.<sup>26</sup>



**Figura 1.18** - Sistema hexadecimal en el microcontrolador.

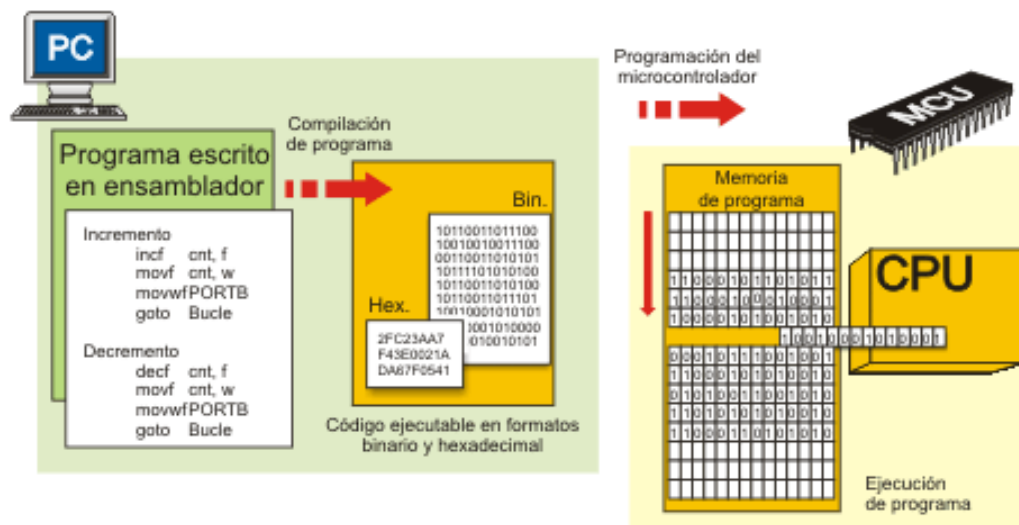
### 1.12.1 LENGUAJE ENSAMBLADOR

Como el proceso de escribir un código ejecutable era considerablemente arduo, en consecuencia fue creado el primer lenguaje de programación denominado ensamblador (ASM). Siguiendo la sintaxis básica del ensamblador, era más fácil escribir y comprender el código. Las instrucciones en ensamblador consisten en las abreviaturas con significado y a cada instrucción corresponde una localidad de memoria un ejemplo de estas instrucciones se ve en la figura 1.19. Un programa

<sup>26</sup> MikroElektronika. (s.f.). Obtenido de <http://www.mikroe.com/chapters/view/80/capitulo-2-programacion-de-los-microcontroladores/>

denominado ensamblador compila (traduce) las instrucciones del lenguaje ensamblador a código máquina (código binario).

Este programa compila instrucción a instrucción sin optimización. Como permite controlar en detalle todos los procesos puestos en marcha dentro del chip, este lenguaje de programación todavía sigue siendo popular.<sup>27</sup>



**Figura 1.19** - Instrucciones del ensamblador.

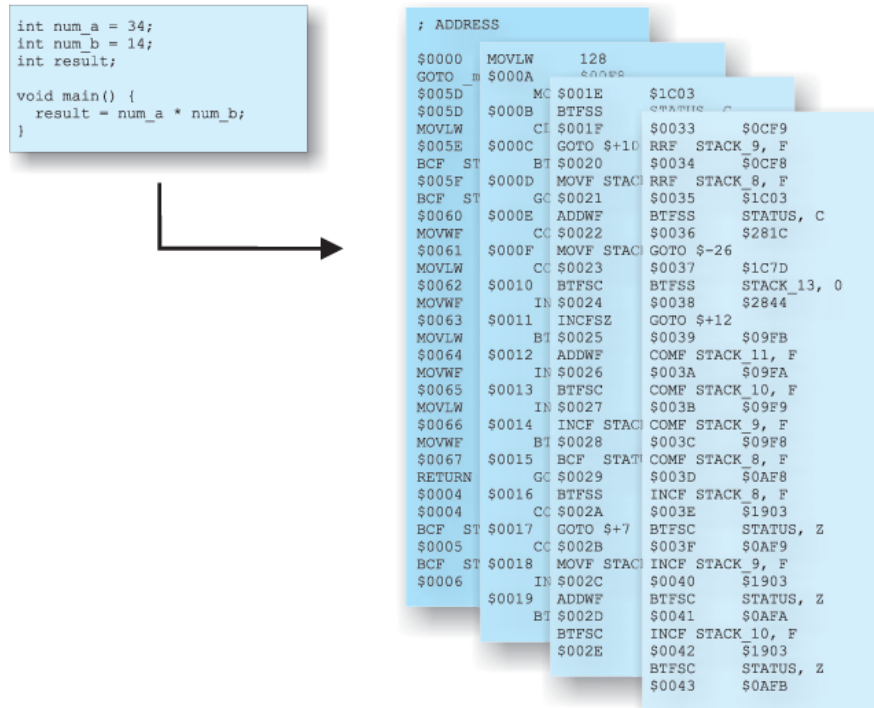
### 1.12.2 VENTAJAS DE LENGUAJES DE PROGRAMACIÓN DE ALTO NIVEL

A pesar de todos los lados buenos, el lenguaje ensamblador tiene algunas desventajas:

- Incluso una sola operación en el programa escrito en ensamblador consiste en muchas instrucciones, haciéndolo muy largo y difícil de manejar.
- Cada tipo de microcontrolador tiene su propio conjunto de instrucciones que un programador tiene que conocer para escribir un programa
- Un programador tiene que conocer el hardware del microcontrolador para escribir un programa

<sup>27</sup> MikroElektronika. (s.f.). Obtenido de <http://www.mikroe.com/chapters/view/80/capitulo-2-programacion-de-los-microcontroladores/>

En la figura 1.20 se puede notar como un Programa escrito en C es el mismo programa compilado al código ensamblador:<sup>28</sup>



**Figura 1.20** - Lenguaje C compilado en ensamblador.

Los lenguajes de programación de alto nivel (Basic, Pascal, C etc.) fueron creados con el propósito de superar las desventajas del ensamblador. En lenguajes de programación de alto nivel varias instrucciones en ensamblador se sustituyen por una sentencia. El programador ya no tiene que conocer el conjunto de instrucciones o características del hardware del microcontrolador utilizado. Ya no es posible conocer exactamente cómo se ejecuta cada sentencia, de todas formas ya no importa. Aunque siempre se puede insertar en el programa una secuencia escrita en ensamblador.

Por ejemplo, no hay instrucción apropiada para multiplicar dos números. No hay que preocuparse al utilizar uno de estos lenguajes de programación de alto nivel

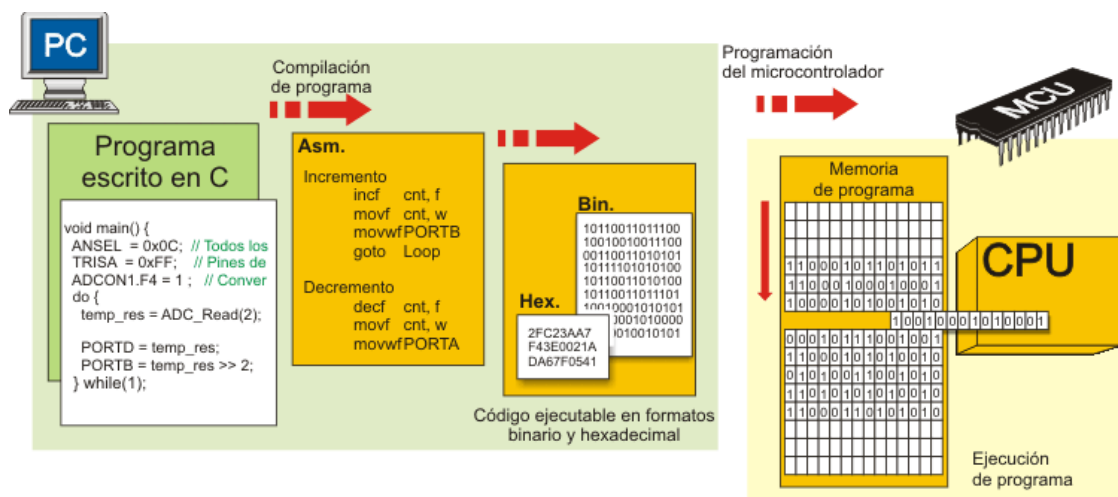
<sup>28</sup> MikroElektronika. (s.f.). Obtenido de <http://www.mikroe.com/chapters/view/80/capitulo-2-programacion-de-los-microcontroladores/>

como es C, porque el compilador encontrará automáticamente la solución a éste problema y otros similares. Para multiplicar los números a y b, basta con escribir  $a*b$ .

### 1.12.3 LENGUAJE C

El lenguaje C dispone de todas las ventajas de un lenguaje de programación de alto nivel (anteriormente descritas) y le permite realizar algunas operaciones tanto sobre los bytes como sobre los bits (operaciones lógicas, desplazamiento etc.). Las características de C pueden ser muy útiles al programar los microcontroladores. Además, C está estandarizado (el estándar ANSI), es muy portable, así que el mismo código se puede utilizar muchas veces en diferentes proyectos. Lo que lo hace accesible para cualquiera que conozca este lenguaje sin reparar en el propósito de uso del microcontrolador. C es un lenguaje compilado, lo que significa que los archivos fuentes que contienen el código C se traducen a lenguaje máquina por el compilador. Todas estas características hicieron al C uno de los lenguajes de programación más populares.<sup>29</sup>

La figura 1.21 es un ejemplo general de lo que sucede durante la compilación de programa de un lenguaje de programación de alto nivel a bajo nivel.



**Figura 1.21-** compilacion de un lenguaje de alto nivel a bajo nivel

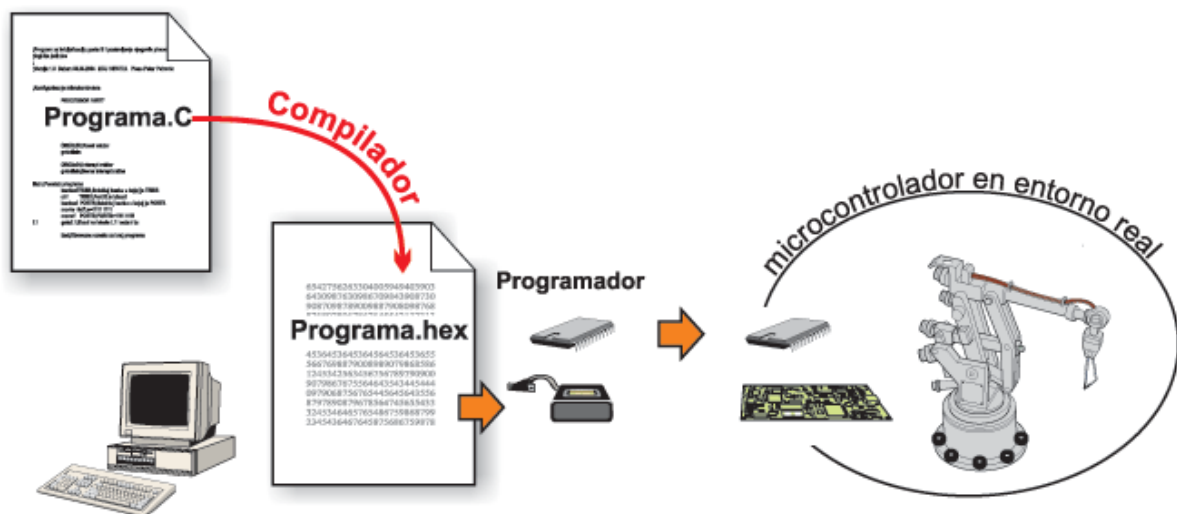
<sup>29</sup> MikroElektronika. (s.f.). Obtenido de <http://www.mikroe.com/chapters/view/80/capitulo-2-programacion-de-los-microcontroladores/>

### 1.12.4 CARACTERÍSTICAS PRINCIPALES DEL MIKROC

Este lenguaje es muy similar al C estándar, no obstante en determinados aspectos difiere del ANSI estándar en algunas características. Algunas de estas diferencias se refieren a las mejoras, destinadas a facilitar la programación de los microcontroladores PIC, mientras que las demás son la consecuencia de la limitación de la arquitectura de hardware de los PIC. El término C se utilizará para referirse a las características comunes de los lenguajes C y mikroC.<sup>30</sup>

### 1.12.5 FASES DE COMPILACIÓN

Como se ve en la figura 1.22, el proceso de compilación consiste en varios pasos y se ejecuta automáticamente por el compilador. Con un conocimiento básico del funcionamiento puede ser útil para entender el concepto del lenguaje mikroC.

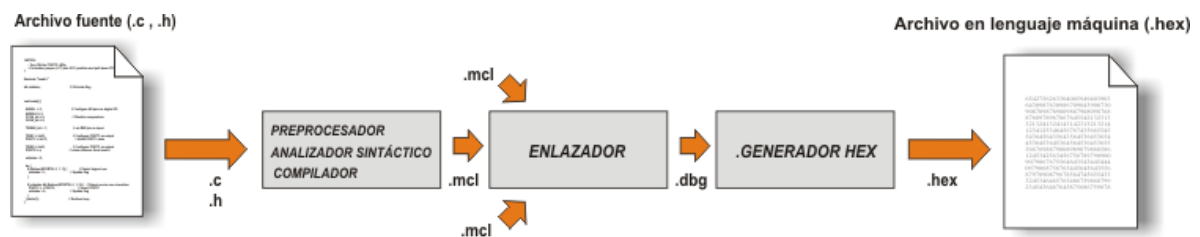


**Figura 1.22** - Pasos de compilación del lenguaje C.

El archivo fuente contiene el código en mikroC que se escribe para programar el microcontrolador. El preprocesador se utiliza automáticamente por el compilador al iniciarse el proceso de la compilación. El compilador busca las directivas del preprocesador (que siempre empiezan por '#') dentro del código y modifica el código fuente de acuerdo con las directivas. En esta fase se llevan a cabo inclusión

<sup>30</sup>MikroElektronika. (s.f.). Obtenido de <http://www.mikroe.com/chapters/view/80/capitulo-2-programacion-de-los-microcontroladores/>

de archivos, definición de constantes y macros etc, lo que facilita el proceso. El **analizador sintáctico** (parser) elimina toda la información inútil del código (comentarios, espacios en blanco). Luego, el **compilador** traduce el código a un archivo binario denominado archivo .mcl. El **enlazador** (linker) recupera toda la información requerida para ejecutar el programa de los archivos externos y la agrupa en un solo archivo (.dbg). Además, un proyecto puede contener más de un archivo fuente y el programador puede utilizar funciones predefinidas y agrupadas dentro de los archivos denominados librerías. Por último, el **generador .hex** produce un archivo .hex. Es el archivo que se va a cargar en el microcontrolador. La figura 1.23 muestra el proceso explicado anteriormente<sup>31</sup>



**Figura 1.23** - Proceso de compilación del lenguaje C.

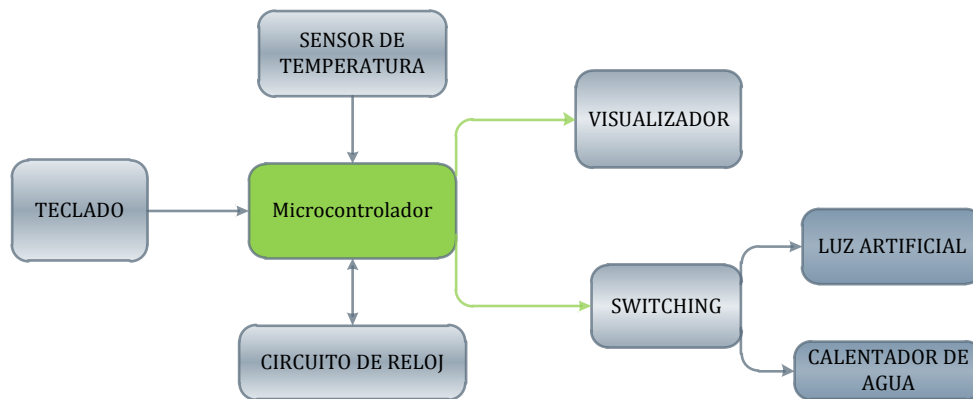
<sup>31</sup> MikroElektronika. (s.f.). Obtenido de <http://www.mikroe.com/chapters/view/80/capitulo-2-programacion-de-los-microcontroladores/>

## CAPÍTULO 2

### CONSTRUCCIÓN DEL HARDWARE

#### 2.1 DESCRIPCIÓN GENERAL

En la figura 2.1 se presenta el esquema en bloques de la funcionalidad del prototipo de detección y control electrónico de temperatura e iluminación para el ecosistema de un acuario de peces tropicales en un clima templado.



**Figura 2.1-** Diagrama de bloques del proyecto

El correcto funcionamiento del proyecto radica en el trabajo de cada uno de los bloques presentados en la figura 2.1. La finalidad de todo el proyecto es de controlar electrónicamente la luz artificial y el calentador del agua en un acuario, para esto se va a dividir la funcionalidad en dos partes, una para el control de la luz artificial y otra para el control del calentador del agua

##### 2.1.1 CONTROL DE LA LUZ ARTIFICIAL

El bloque luz artificial es controlada por un circuito que converge la señal electrónica a la señal eléctrica en el bloque switching, éste es a si mismo controlado por el microcontrolador, el cual interpreta la señal recibida del bloque de circuito de



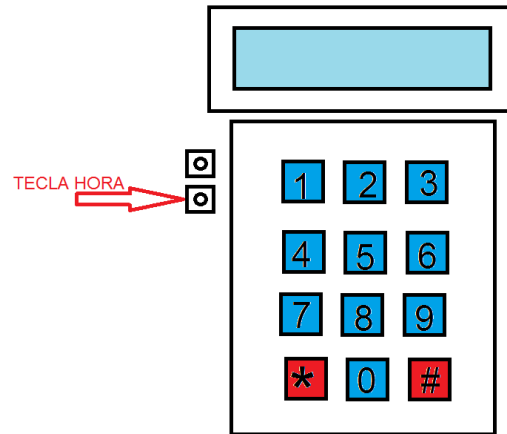
reloj y emite una señal para encender o apagar, en conclusión la luz artificial dependerá explícitamente del bloque reloj. Entonces el control del encendido y apagado de la luz artificial depende de la hora, es decir, “las luces deben estar encendidas entre 10 y 12 horas al día”<sup>32</sup>. De seis de la mañana a seis de la tarde la luz artificial se mantiene encendida y el resto de la horas se mantiene apagada. La hora actualizada es presentada por el bloque visualizador, para tener una interpretación personal de la hora, al mismo hecho de que la hora puede ser igualada con las opciones que presenta el bloque teclado.

#### **2.1.1.1 Configuración de actualización de la hora de reloj**

La configuración de reloj es necesario cuando por algún motivo se haya desconfigurado la hora del sistema. Para esto las opciones que presenta el bloque teclado para igualar la hora son fáciles y prácticos. Como primer paso es presionar la tecla “hora”, como indica la figura 2.2, luego de eso se presenta en la pantalla del bloque visualizador la palabra “HORA:” seguido de los valores de “00:00:\*\*”, estos valores aparecen si es que es la primera vez que se quiere ingresar los valores de actualización, y si no, los valores de la última actualización. El registro de los números de actualización deben ser ingresados por el teclado numérico, tecleando uno por uno la cifra que se desea ingresar, las dos primeras cifras que se ingresan son de la hora, las dos siguientes son los de los minutos, si se continua tecleando se vuelve a ingresar desde los valores de la hora. Para finalizar la actualización de la hora se debe presionar la tecla numeral “#”. Los asteriscos que se presentan en la pantalla representan a los segundos, estos no pueden ser ingresados por el usuario ya que luego de presionar la tecla numeral inmediatamente se regresa a la pantalla principal y los segundos quedan con el valor de cero.

---

<sup>32</sup> hola.com. (22 de octubre de 2012). Obtenido de <http://mascotas.hola.com/peces/acuario-peces-tropicales/244>



**Figura 2.2** – Representación tecla hora.

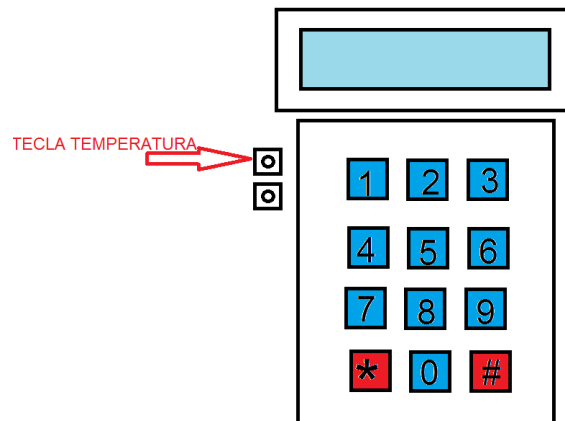
### 2.1.2 CONTROL DE CALENTADOR DE AGUA

El bloque calentador de agua es controlado por un circuito que converge la señal electrónica a la señal eléctrica en el bloque switching, éste es a si mismo controlado por el microcontrolador, el cual interpreta la señal recibida del bloque de sensor de temperatura, ya que la temperatura del agua va a ser acondicionada por el calentador que se enciende y se apaga según las condiciones que se desee, según el registro del bloque sensor de temperatura y el análisis del bloque microcontrolador, es decir, que si el agua se está enfriando, el calentador se enciende automáticamente para calentar el agua, y se mantiene encendido hasta que el agua tenga una temperatura idónea, para luego apagarse permitiendo que no exista un sobrecalentamiento del agua, y al momento que se vuelva a enfriar el agua, se repita el ciclo de calentamiento. La temperatura que tiene el acuario es presentada por el bloque de visualización ayudando a tener una referencia y control de la temperatura. La temperatura idónea del acuario va a depender “de una guía de peces, clasificados de acuerdo a sus características”<sup>33</sup>, que se encuentre en el acuario, es por eso que por medio del bloque de teclado se puede configurar la temperatura para que el control de la temperatura sea apta.

<sup>33</sup> Dotti, A. (s.f.). TARINGA. Obtenido de <http://www.taringa.net/posts/mascotas/3198525/Guia-de-peces-tropical.html>

### 2.1.2.1 Configuración de temperatura

Por la variedad que existe de peces tropicales, la temperatura de ambiente de cada uno de estos no serán las mismas. Para la facilidad de tener la temperatura adecuada en el ecosistema del pez en el acuario existe la opción de ingresar un registro promedio para que el microcontrolador pueda interpretar y controlar la temperatura del acuario, esta opción práctica es validada por el bloque de teclado, presionando la tecla “temperatura” como indica la figura 2.3, para que, en el bloque de visualización, se presente “CONFIGURAR TEMP” en la primera línea, y en la segunda línea aparece la opción de ingresar dos cifras para la temperatura, estas dos cifras representan al valor promedio que se desee tener en el acuario, este valor se debe ingresar por el teclado numérico cifra por cifra, al igual que el registro de la hora, si se sigue presionado los números en el teclado se volverá a registrar desde la primera cifra, para finalizar el ingreso del valor promedio de la temperatura se debe presionar la tecla numeral “#”, y se volverá a la pantalla principal en donde se apreciará la temperatura de la pecera y el valor que se registró.



**Figura 2.3** – Representación tecla temperatura.

## 2.2 DESCRIPCIÓN POR BLOQUES

Si bien el proyecto trabaja en conjunto con todos los bloques tomando en relación uno con otro, es necesario detallar que elementos trabajan en de cada uno de

estos bloques para tener un entendimiento más claro del funcionamiento de cada proceso.

## 2.2.1 DESCRIPCIÓN DEL BLOQUE MICROCONTROLADOR

Este es el bloque principal de toda la funcionalidad del proyecto, ya que comprende la parte “inteligente”, porque en relación a los valores que recibe de otros bloques los interpreta, analiza y controla los bloques de salida, es por eso que todo se centra en este bloque, ahora detallando físicamente el elemento principal del proyecto es el PIC16F877A

### 2.2.1.1 PIC 16F877A

El PIC 16F877A es un microcontrolador de Microchip Technology fabricado en tecnología CMOS, su consumo de potencia es muy bajo y además es completamente estático, esto quiere decir que el reloj puede detenerse y los datos de la memoria no se pierden. El encapsulado más común para este microcontrolador es el DIP (Dual In-line Pin) de 40 pines, propio para usarlo en experimentación.<sup>34</sup>

Las características que tiene este microcontrolador son las necesarias para el trabajo de este proyecto. En la tabla 2.1 se indica algunas características del microcontrolador.

**Tabla 2.1-** Características del microcontrolador utilizado

Frecuencia de operación	DC-20MHz
Memoria de programa (flash) (palabras de 14-bits)	8k
Memoria de datos (bytes)	368
Puertos de entrada y salida	Puertos A, B, C, D, E
Comparadores analógicos	2
Puerto serial sincrónico	I2C

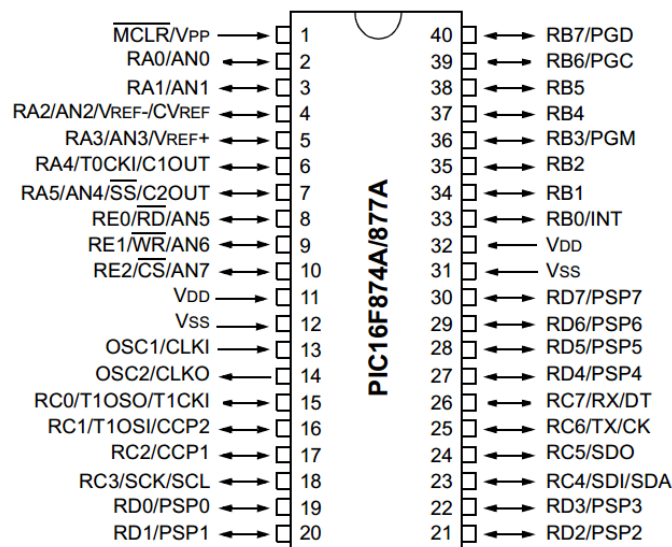
<sup>34</sup> <http://www.utp.edu.co/~eduque/arquitect/PIC16F877.pdf>

a. *Arquitectura interna*

Este término se refiere a los bloques funcionales internos que conforman el microcontrolador y la forma en que están conectados, por ejemplo la memoria FLASH (de programa), la memoria RAM (de datos), los puertos, la lógica de control que permite que todo el conjunto funcione, etc.<sup>35</sup> En la figura 2.5 se presenta la Arquitectura del microcontrolador

b. *Configuración de pines PIC16F877A*

En la figura 2.4 se indica la distribución de pines del microcontrolador, se puede notar que cada pin está diferenciado por puerto al que pertenece y por cada función que cumple el pin, la distribución de pines empleado para el proyecto se dará a conocer en la sección e de este subcapítulo, si se requiere más detalles de cada pin se puede referenciar en la hoja de datos del microcontrolador.

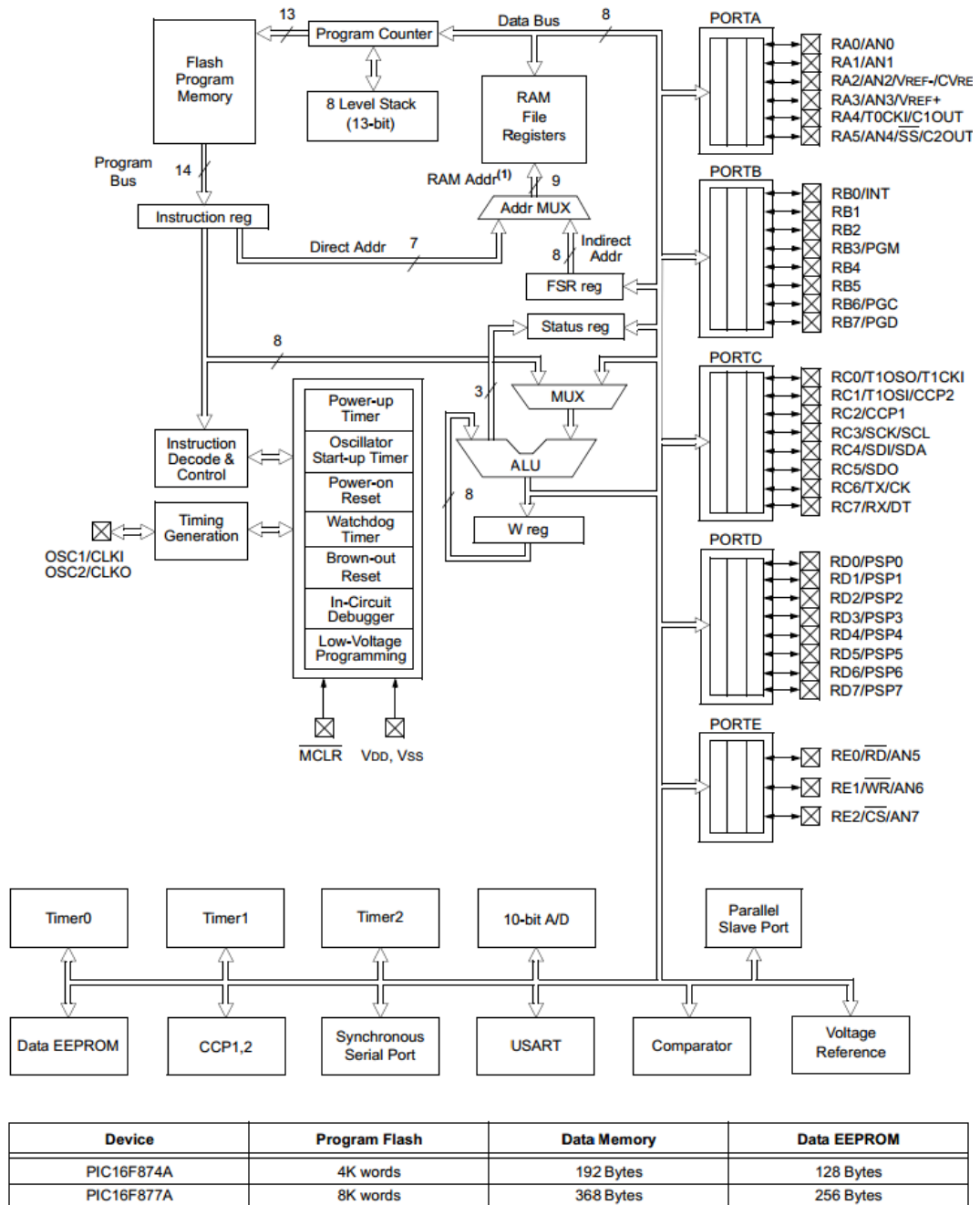


**Figura 2.4** - Distribución de pines PIC 16F877A

Los pines que no pertenecen a algún de puertos son los de polarización, o el pin para el reseteo del PIC, o los pines para las entradas del oscilador

<sup>35</sup> <http://www.utp.edu.co/~eduque/arquitect/PIC16F877.pdf>

externo, siendo estos pines utilizados para el propio uso y funcionamiento del PIC y no para funciones de control.



**Note 1:** Higher order bits are from the Status register.

**Figura 2.5-** Arquitectura del microcontrolador PIC 16F877A.

En la tabla 2.2 se especifica la distribución de pines para el funcionamiento del PIC.

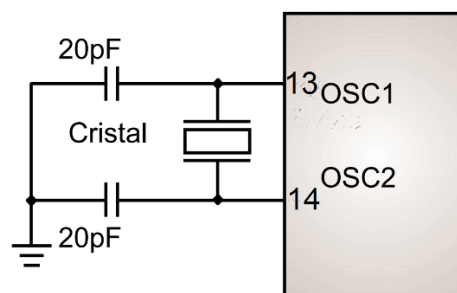
**Tabla 2.2-** Pines especiales del microcontrolador

Nombre del PIN	Pin	descripción
MCLR	1	Entrada de RESET al microcontrolador. Voltaje de entrada durante la programación. En nivel bajo resetea el microcontrolador
VDD	11,32	Voltaje de alimentación DC (+)
VSS	12,31	Referencia de voltaje (GND).
OSC1/CLKIN	13	Entrada oscilador cristal oscilador / Entrada fuente de reloj externa.
OSC2/CLKOUT	14	Salida oscilador cristal. Oscilador RC: Salida con un ¼ frecuencia OSC1

c. *El oscilador externo*

Todo microcontrolador requiere un circuito externo que le indique la velocidad a la que debe trabajar. Este circuito, que se conoce como oscilador o reloj, es muy simple pero de vital importancia para el buen funcionamiento del sistema.<sup>36</sup>

En este proyecto se utiliza el oscilador HS el cual conforma un cristal de 20 MHz y dos capacitores de 20 picofaradios. Como indica la figura 2.6 la conexión del cristal debe ir acompañada por dos capacitores. Con el oscilador HS garantiza una mayor precisión que otros tipos de osciladores externos.

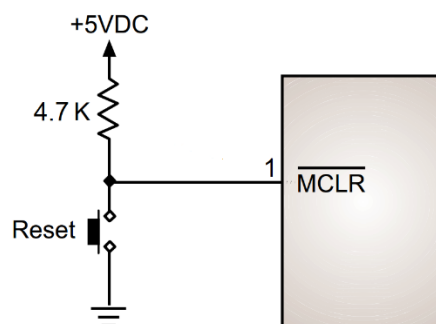
**Figura 2.6-** Conexión oscilador HS y los capacitores

<sup>36</sup> <http://www.utp.edu.co/~eduque/arquitect/PIC16F877.pdf>

d. *Reset*

En los microcontroladores se requiere un pin de reset para reiniciar el funcionamiento del sistema cuando sea necesario, ya sea por una falla que se presente o porque así se haya diseñado el sistema. El pin de reset en los PIC es llamado MCLR (master clear). Uno de los reset es al encendido (Power On Reset) y para precautelar el funcionamiento del proyecto se utiliza el pulsador en el pin MCLR.

El reset al encendido se consigue gracias a dos temporizadores. El primero de ellos es el OST (Oscillator Start-Up Timer: Temporizador de encendido del oscilador), orientado a mantener el microcontrolador en reset hasta que el oscilador del cristal es estable. El segundo es el PWRT (Power-Up Timer: Temporizador de encendido), que provee un retardo fijo de 72 ms (nominal) en el encendido únicamente, diseñado para mantener el dispositivo en reset mientras la fuente se estabiliza. Para utilizar estos temporizadores, basta con conectar el pin MCLR a la fuente de alimentación. El reset por MCLR se consigue llevando momentáneamente este pin a un estado lógico bajo, mientras que el watchdog WDT produce el reset cuando su temporizador rebasa la cuenta, o sea que pasa de 0FFh a 00h. Como se muestra en la figura 2.7, cuando se quiere tener control sobre el reset del sistema se puede conectar un botón<sup>37</sup>



**Figura 2.7-**Conexión del botón reset

<sup>37</sup> <http://www.utp.edu.co/~eduque/arquitect/PIC16F877.pdf>



e. *Distribución de pines por bloques*

Los pines de entrada/salida de este microcontrolador están organizados en cinco puertos, el puerto A con 6 líneas, el puerto B con 8 líneas, el puerto C con 8 líneas, el puerto D con 8 líneas y el puerto E con 3 líneas.<sup>38</sup> Los pines de esos puertos están configurados para funciones en específico e interconectados con la mayoría de los elementos que conforman el proyecto. En la tabla 2.3 se indica una distribución de los pines en función al puerto que pertenecen, al bloque que conforman y a los elementos que van conectados.

**Tabla 2.3-** Distribucion de pines del microcontroladro por bloques de funcionamiento

PIN		PUERTO	BLOQUE	CONEXIÓN A ELEMENTOS	
2	RA0	A	Sensor de temperatura	Vout LM35	
33	RB0	B	TECLADO	R 10 kΩ   fila 1 teclado numérico	
34	RB1			R 10 kΩ   fila 2 teclado numérico	
35	RB2			R 10 kΩ   fila 3 teclado numérico	
36	RB3			R 10 kΩ   fila 4 teclado numérico	
37	RB4			Columna 1 teclado numérico	
38	RB5			Columna 2 teclado numérico	
39	RB6			Columna 3 teclado numérico	
40	RB7			Pulsadores hora y temperatura	
15	RC0	C	MICROCONTROLADOR	R 10K	
18	RC3		CIRCUITO DE RELOJ	SCL PIN 6 DS1307	
23	RC4			SDA PIN 5 DS1307	
24	RC5		MICROCONTROLADOR	BUZZER	
25	RC6		SWITCHING		PIN 1 MOC3010 – CONTROL DE LUZ
26	RC7				PIN 1 MOC3010 – CONTROL CALEFACTOR
21	RD2	D	VISUALIZADOR	PIN 4 LCD – RS Selección dato/instrucción	
22	RD3			PIN 6 LCD – E Habilitación o activación	
27	RD4			PIN 11 LCD – DATOS 4	
28	RD5			PIN 12 LCD – DATOS 5	
29	RD6			PIN 13 LCD – DATOS 6	
30	RD7			PIN 14 LCD – DATOS 7	

<sup>38</sup> <http://ww1.microchip.com/downloads/en/DeviceDoc/39582b.pdf>

Las funciones de cada pin señalado son programados en MickleC (en el capítulo 3 se especifica todo el programa de funcionalidad), en la tabla 2.3 solo están especificados los pines utilizados para el proyecto, a excepción de los pines datallados anteriormente como son los de reset, polarización y del cristal, y los otros pines del microcontrolador sin asignar funcionalidades. El pin conectado al buzzer, es utilizado para generar una alerta sonora en caso de que la temperatura del agua, que detecta el sensor de temperatura, se está sobrecalentando o está demasiado fría por falla del circuito switching. La intercomunicación del reloj con el microcontrolador se da por los pines SCL, línea de reloj serial asíncrono en el modo I<sup>2</sup>C, y SDA, línea de datos en el modo I<sup>2</sup>C. La señal que recibe del bloque sensor de temperatura es una señal analógica, por esta razón está conectado al pin RA2 de entrada analógica y necesita de un módulo convertidor analógico digital para que el microcontrolador pueda procesar esa señal.

*f. Módulo del convertidor Analógico a Digital*

Este módulo permite la conversión de una señal de entrada analógica a su correspondiente valor numérico de 10 bits. El módulo tiene ocho entradas analógicas, las cuales son multiplexadas dentro de un circuito de muestreo y retención. La salida del multiplexor es la entrada al convertidor, el cual genera el resultado por medio de aproximaciones sucesivas.<sup>39</sup>

Como se ha explicado anteriormente el PIC 16F877A es el principal elemento en todo el proyecto y en la figura 2.8 se observa cómo están conectados la mayoría de los bloques al microcontrolador y también se detalla cómo están interconectados los componentes de cada uno de los bloques.

---

<sup>39</sup> <http://www.utp.edu.co/~eduque/arquitec/PIC16F877.pdf>

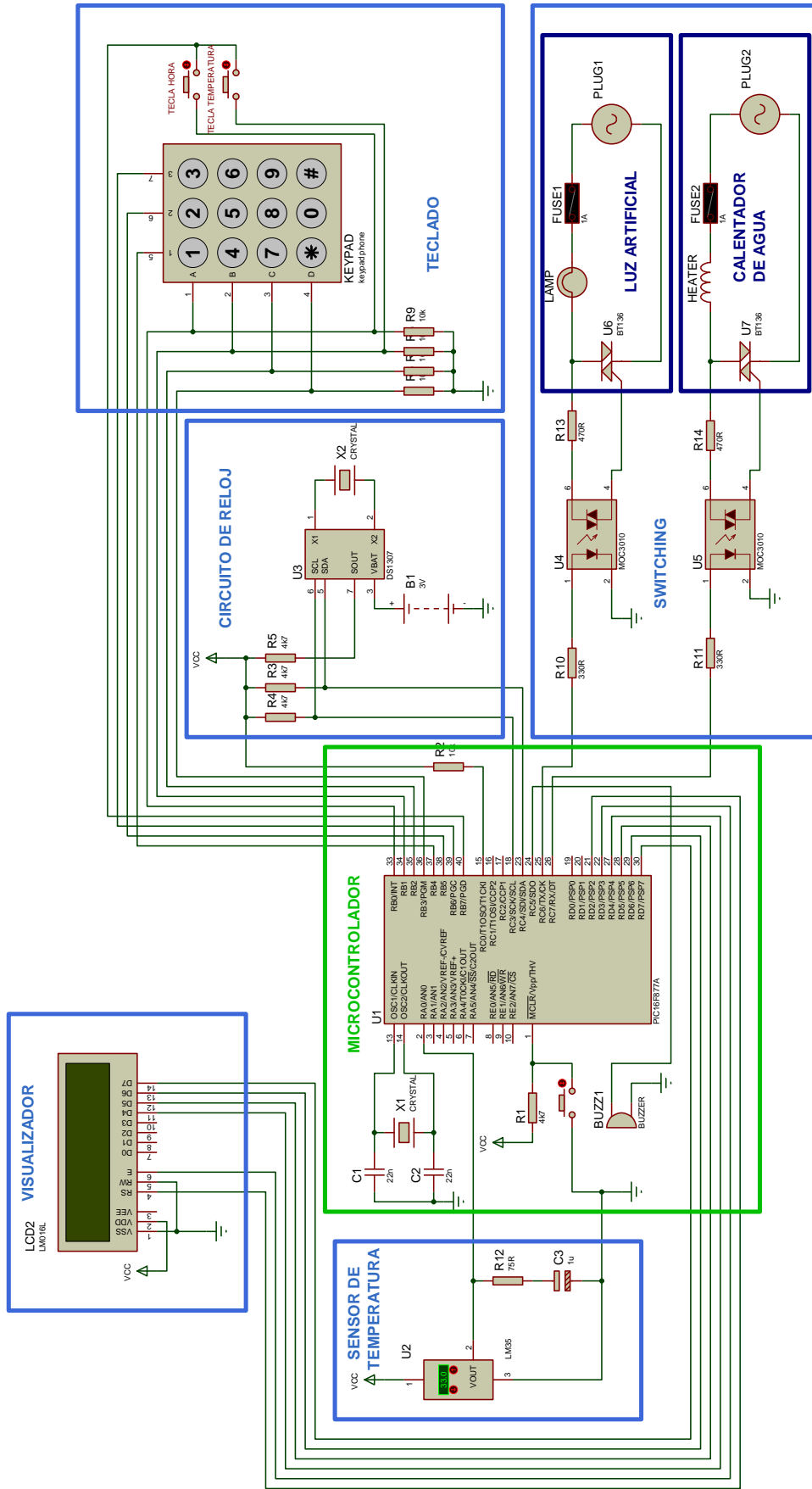


Figura 2.8 - Circuito de conexión de los elementos al microcontrolador

## 2.2.2 DESCRIPCIÓN DEL BLOQUE SENSOR DE TEMPERATURA

La necesidad de saber qué temperatura tiene el acuario para poder calentar o dejar de calentar el agua es importante porque los peces requieren unos cuidados básicos a los que se tiene que prestar atención. Una buena temperatura del agua, será la clave para tener a los peces en perfecto estado.<sup>40</sup> Es por eso que la opción de tener un elemento capaz de solucionar esta necesidad y usarlo electrónicamente es la del sensor de temperatura LM35, ya que es conocido en su funcionamiento en las aulas de clases, y ayuda a convertir la temperatura ambiente en una señal eléctrica para que el microcontrolador la analice.

### 2.2.2.1 LM35

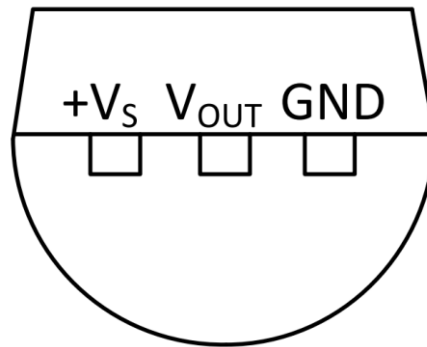
El circuito integrado LM35 es un sensor de temperatura, con el voltaje de salida linealmente proporcional a la temperatura en grados centígrados. Este circuito integrado no requiere de ninguna calibración externa para proporcionar una precisión de  $\pm \frac{1}{4}^{\circ} \text{C}$  en temperatura ambiente y  $\pm \frac{3}{4}^{\circ} \text{C}$  entre el rango de temperatura  $-55^{\circ} \text{C}$  y  $+150^{\circ} \text{C}$ . La baja impedancia de salida, la salida lineal, y la inherente calibración precisa del LM35, hacen que la lectura de conexión sea especialmente fácil. El LM35 es usado con una fuente de poder simple, consume sólo  $60 \mu\text{A}$ , y tiene una disipación de calor de  $0,1^{\circ} \text{C}$  en aire quieto. El LM35 está calificado para operar en un rango de temperatura de  $-55^{\circ} \text{C}$  a  $+150^{\circ} \text{C}$ , mientras que el LM35C está clasificado para un rango de temperatura de  $-40^{\circ} \text{C}$  a  $+110^{\circ} \text{C}$  ( $-10^{\circ} \text{C}$  con una mayor precisión).<sup>41</sup>

En la figura 2.9 se observa la distribución de los pines del sensor de temperatura

---

<sup>40</sup> hola.com. (22 de octubre de 2012). Obtenido de <http://mascotas.hola.com/peces/acuario-peces-tropicales/244>

<sup>41</sup> Instruments, T. (s.f.). ti. Obtenido de <http://www.ti.com/lit/ds/symlink/lm35.pdf>



**Figura 2.9** - Distribución de pines LM35 vista inferior

La aplicación que tiene este elemento electrónico es dentro de la pecera, por lo que se presentan algunas desventajas de funcionamiento, tal como la distancia de comunicación que tiene el sensor hasta el microcontrolador susceptible al ruido generado por la lámpara del acuario o por otras fuentes de ruido cercanas, evitando que la señal que emite el LM35 sea la correcta, y la desventaja de permanecer dentro del agua e impedir que trabaje correctamente. Para estas dos desventajas se dan las soluciones como filtrar la señal del LM35 e impermeabilizar el circuito de filtro con el sensor de temperatura.

#### 2.2.2.2 Filtración de la señal del LM35

De acuerdo con la hoja de datos del sensor de temperatura, como la mayoría de los circuitos de micropotencia, el LM35 tiene una capacidad limitada para manejar cargas capacitivas pesadas. El LM35 solo es capaz de manejar 50 pF sin precauciones especiales.<sup>42</sup> Existen circuitos de filtración para solucionar este problema. La figura 2.10 indica el circuito empleado para la filtración de la señal, habilitando la posibilidad de trabajar con el LM35 distantes del microcontrolador.

<sup>42</sup> Instruments, T. (s.f.). ti. Obtenido de <http://www.ti.com/lit/ds/symlink/lm35.pdf>



### 2.2.3 DESCRIPCIÓN DEL BLOQUE CIRCUITO DE RELOJ

La función del proyecto de establecer un ambiente idóneo para los peces tropicales se centra también en la iluminación que debe tener el acuario ya que los peces provienen de zonas tropicales en donde tienen 12 horas de luz y 12 de oscuridad, debe iluminarse el acuario durante al menos 10 horas, hasta un máximo de 12 horas, tanto por los peces como por las plantas especialmente. Es muy importante respetar el ciclo del día y la noche, caso contrario los habitantes del acuario tendrán síntomas de alteraciones en su salud.<sup>43</sup>

La referencia que tiene el microcontrolador de saber qué hora es del día y controlar el encendido y apagado de la luz artificial, parte del circuito integrado DS1307 que cumple la función de un reloj, generando información de las horas y minutos del día para que el microcontrolador las pueda leer y así encender o apagar la luz artificial.

#### 2.2.3.1 DS1307

##### *a. Descripción.*

El DS1307 es un reloj en tiempo real de bajo consumo de energía, completo con código binario decimal (BCD), reloj/calendario más 56 bytes de NV SRAM. Dirección y datos son transferidos a través de 2 hilos serie, bus bi-direccional. El reloj/calendario provee información de, segundos, minutos, horas, día, fecha, mes y año. El final de fecha de mes se ajusta automáticamente durante meses menores de 31 días, incluyendo correcciones para el año bisiesto. El reloj funciona en cualquiera formato de 24 horas o en 12 horas con indicador AM/PM. El DS1307 tiene incorporado un circuito de sensor de tensión que detecta fallas de energía y cambia automáticamente al suministro de batería de respaldo.

---

<sup>43</sup> Blanco, A. (s.f.). Obtenido de [http://peces-tropicales.idoneos.com/index.php/Generalidades/Iluminacion\\_acuario](http://peces-tropicales.idoneos.com/index.php/Generalidades/Iluminacion_acuario)





SCL – reloj del puerto I2C

SQW/OUT - Onda cuadrada / Driver de salida

c. Descripción de la señal.

**VCC, GND** - La alimentación DC del dispositivo se ofrece en estos pines. VCC es entrada de +5 V. Cuando se aplican 5V dentro de límites normales, el dispositivo es totalmente accesible y los datos pueden ser escritos y leídos. Cuando una batería de 3V se conecta al dispositivo y VCC es inferior a  $1,25 \times V_{BAT}$ , se inhiben lectura y escritura. Sin embargo, la función de la hora normal no se ve afectada por la baja tensión de entrada.

Como VCC caiga por debajo de VBAT, la RAM y el cronometro se cambian a la fuente de energía externa (nominal 3.0V DC) en VBAT.

**VBAT** - Entrada para cualquier batería de litio estándar de 3V u otra fuente de energía. El voltaje de la batería debe ser mantenido entre 2,0 V y 3,5 V para su correcto funcionamiento. La tensión nominal de protección de escritura punto de disparo en el cual el acceso al RTC y la memoria RAM de usuario es denegado, es fijado por el circuito interno como nominal  $1,25 \times V_{BAT}$ . Una batería de litio con 48mAh o mayor mantendrá copia de seguridad del DS1307 durante más de 10 años en ausencia de energía a 25 ° C. Reconocimiento UL asegura contra inversión de corriente de carga cuando se utiliza junto con una batería de litio.

**SCL** (Serial Clock Input) - SCL se utiliza para sincronizar el movimiento de datos en la interfaz serie, requiere una RPA (Resistencia de Polarización a Alto externa).

**SDA** (Serial Data Input/Output) - SDA es el pin entrada/salida para el interfaz 2-hilos serie. El SDA es el pin de drenaje abierto, que requiere una RPA (Resistencia de Polarización a Alto externa).

**SQW/OUT** (Onda Cuadrada/controlador de Salida) - Cuando se activa, el bit SQWE se establece en 1, el pin SQW/OUT es la salida de una de las cuatro frecuencias de onda cuadrada (1 Hz, 4 kHz, 8 kHz, 32 kHz). El pin SQW/OUT es de drenaje abierto y requiere una RPA (Resistencia de Polarización a Alto externa). SQW/OUT funcionará con cualquiera Vcc o Vbat aplicada.

**X1, X2** - Conexiones para un cristal de cuarzo estándar 32.768kHz<sup>44 45</sup>

## 2.2.4 DESCRIPCIÓN DEL BLOQUE VISUALIZADOR

Para poder entender con qué valores se están referenciando la lógica del proyecto, el bloque visualizador nos presenta en la pantalla del LCD los valores de la hora, de la temperatura de referencia y la temperatura que se encuentra en el acuario. También ayuda a la configuración de la hora y de la temperatura como se había explicado antes en este capítulo. El LCD que se está utilizando es un LCD de 16X2. La figura 2.13 muestra el LCD utilizado en el proyecto.



Figura 2.13 – LCD 16x2.

### 2.2.4.1 LCD 16X2

#### a. Descripción:

La pantalla de cristal líquido o LCD (Liquid Crystal Display) es un dispositivo microcontrolado de visualización gráfica para la presentación de caracteres, símbolos o incluso dibujos (en algunos modelos), en este caso dispone de 2 filas de 16 caracteres cada una, y cada carácter dispone de una matriz de 5x7 puntos (píxeles). Este dispositivo está gobernado internamente por un microcontrolador y regula todos los parámetros de presentación.<sup>46</sup>

<sup>44</sup> García, V. (s.f.). hispavila. Obtenido de [http://hispavila.com/3ds/ics/ds1307\\_esp.pdf](http://hispavila.com/3ds/ics/ds1307_esp.pdf)

<sup>45</sup> SEMICONDUCTOR, D. (s.f.). Obtenido de <https://www.sparkfun.com/datasheets/Components/DS1307.pdf>

<sup>46</sup> Rosero, L. (s.f.). SCRIBD. Obtenido de <http://es.scribd.com/doc/44252680/LCD-16X2#download>

*b. Características principales:*

- Pantalla de caracteres ASCII, además de los caracteres Kanji y Griegos.
- Desplazamiento de los caracteres hacia la izquierda o la derecha.
- Proporciona la dirección de la posición absoluta o relativa del caracter.
- Memoria de 40 caracteres por línea de pantalla.
- Movimiento del cursor y cambio de su aspecto.
- Conexión a un procesador usando un interfaz de 4 u 8 bits

*c. Descripción de pines*

En la tabla se muestra la distribución de pines del LCD 16x2

**Tabla 2.4** - Distribución de pines LCD 16x2.

PIN	NOMBRE DE PPIN	DESCRIPCION
1	Vss	Tierra de alimentación GND
2	vdd	Alimentación de + 5v cc
3	vo	Contraste del cristal líquido. (0 a + 5v)
4	Rs	selección del registro de control/ registro de datos
5	R/W	Señal de lectura escritura
6	E	Habilitación del modulo
7-14	D0-D7	Bus de datos bidireccional

### 2.2.5 DESCRIPCIÓN DEL BLOQUE TECLADO

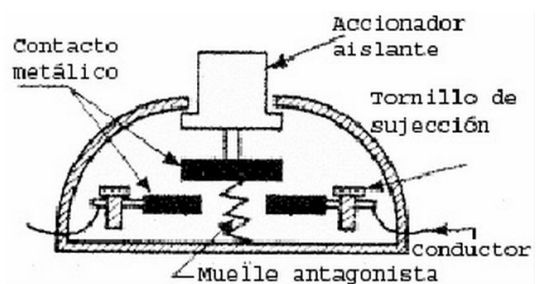
Ya que el proyecto trabaja en base a la información de la temperatura y a la hora del día, la configuración de cada uno es vital para que el proyecto funcione de acuerdo a las necesidades del acuario. La configuración de actualización de hora y temperatura promedio es llevada a cabo del bloque teclado trabajando de forma práctica y fácil de utilizar. Este bloque se conforma por un teclado matricial de 4x3 (numérico) y dos pulsadores como muestra la figura 2.14.



**Figura 2.14** – Teclado y pulsadores del bloque teclado.

### 2.2.5.1 Pulsador

Elemento que permite el paso o interrupción de la corriente mientras es accionado. Cuando ya no se actúa sobre él vuelve a su posición de reposo. Puede ser el contacto normalmente cerrado en reposo NC, o con el contacto normalmente abierto NA. Consta del botón pulsador; una lámina conductora que establece contacto con los dos terminales al oprimir el botón y un muelle que hace recobrar a la lámina su posición primitiva al cesar la presión sobre el botón pulsador<sup>47</sup> como se indica en la figura 2.15.



**Figura 2.15** - Funcionamiento pulsador.

En la programación del microcontrolador el puerto que está configurado el bloque teclado, está designado para un teclado matricial 4x4, como los teclados son solo

<sup>47</sup> CXmat, L. (s.f.). SCRIBD. Obtenido de <http://es.scribd.com/doc/94364908/Pulsadores>

pulsadores conectados de forma matricial, la última columna de 4 pulsadores es separada para los dos pulsadores de configuración, un pulsador para la configuración de la hora y otro pulsador para la configuración de la temperatura, sin perder la funcionalidad como un teclado matricial 4x4. Según la programación del microcontrolador la funcionalidad del pulsador “hora” es solo ingresar en un menú de ingreso de la hora si se está en la pantalla principal, y la funcionalidad del pulsador “temperatura” es solo ingresar en un menú de ingreso de la temperatura de referencia si se está en la pantalla principal.

#### **2.2.5.2 Teclado matricial**

El uso de un teclado numérico matricial facilita a la configuración que ofrece el proyecto, solo presionando las teclas de los números según sea lo necesario.

La programación que tiene el microcontrolador especifica que las teclas del 0 al 9 son solo para el ingreso de los datos de temperatura y hora, es decir que a no más que se presione la tecla de configuración de hora o configuración de temperatura no tenga funcionalidad. También es especificado que la tecla de numeral “#” sea solo para confirmar la configuración ingresada en los menús de configuración de la hora y temperatura, y regresar a la pantalla principal, y luego de eso no cumple función. La última especificación es encender la luz artificial por 5 minutos si ésta se encuentra apagada y en el visualizador esté en la pantalla principal. Dentro del ciclo de 5 minutos de encendido de la luz artificial no se puede realizar ninguna configuración.

### **2.2.6 DESCRIPCIÓN DEL BLOQUE SWITCHING**

#### **2.2.6.1 Acoplamiento de la señal para el control**

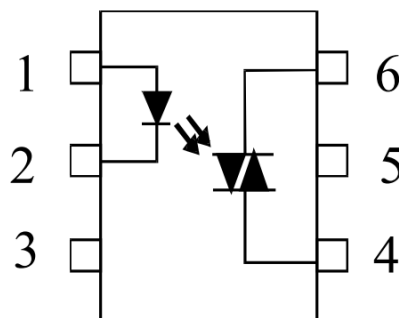
Este bloque es bastante necesario para el control del calentador de agua y de la luz artificial ya que estos dos elementos trabajan a 120 voltios de corriente alterna, y las señales que da el microcontrolador es entre 0 y 5 voltios de corriente continua. Por esta razón se elige este circuito para controlar a partir de una señal digital 1 o

0, cumpliendo la función de encender o apagar respectivamente el calentador de agua y la luz artificial. Tanto el calentador de agua como la luz artificial necesitan un circuito independiente conformado con un optoacoplador MOC3010, una resistencia de 40 ohmios, un triac BT136 y un fusible como muestra la figura 2.8. Los 2 circuitos que controlan a la luz artificial y al calentador de agua funcionan de la siguiente manera. El optoacoplador recibe la señal del microcontrolador si es 1 lógico (5 vcc) y activa el triac interno, fluyendo corriente alterna de la resistencia de 470 ohmios al Gateway del triac BT136, activando el flujo de corriente hacia los dispositivos que trabajan con corriente alterna. La corriente que fluye por el triac del optoacoplador es muy pequeña para poder encender a la lámpara es por eso que se trabaja con un TRIAC de mayor potencia como es el BT136.

#### 2.2.6.1.1 CIMOC 3010

El MOC 3010 es un circuito integrado aislador que acopla ópticamente y consiste de un infrarrojo de arsenuro de galio emitido por el diodo, acoplado con la luz a un switch bilateral de silicio realizando las funciones de un triac montado en un estándar de 6 pines dual in line.<sup>48</sup>

La distribución de pines del optoacoplador se indica en la siguiente figura 2.16.



**Figura 2.16** – Distribución de pines del C.I. MOC3010.

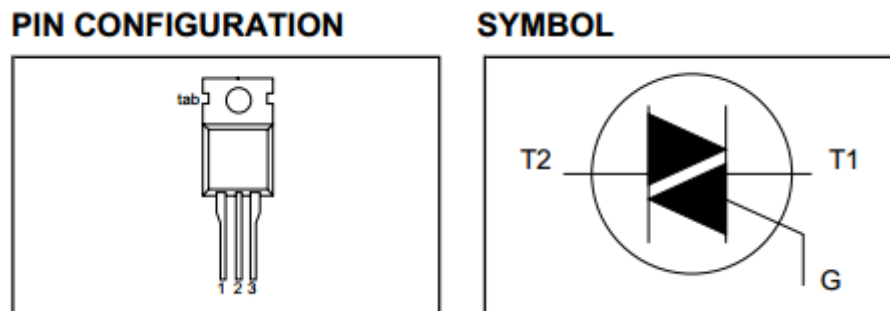
#### 2.2.6.1.2 TRIAC BT136

Triacs pasivado de cristal en una envoltura de plástico, destinados a su uso en aplicaciones que requieren alta transición bidireccional y que bloquean la capacidad del voltaje y de alto rendimiento de ciclos térmicos. Las aplicaciones

<sup>48</sup> ISICOM. (s.f.). Datasheetcatalog. Obtenido de [http://pdf.datasheetcatalog.com/datasheet\\_pdf/isocom/MOC3009\\_to\\_MOC3012X.pdf](http://pdf.datasheetcatalog.com/datasheet_pdf/isocom/MOC3009_to_MOC3012X.pdf)

típicas incluyen control de motores, iluminación industrial y doméstica, calefacción y conmutación estática.<sup>49</sup>

La distribución de pines del Triac se presenta en la Figura 2.17 y la función de cada pin en la tabla 2.5.



**Figura 2.17** – Distribución de pines del TRIAC BT136.

**Tabla 2.5** – Configuración de los pines del TRIAC BT136

Pin		Descripción
1	T1	Terminal principal 1
2	T2	Terminal principal 2
3	G	compuerta
tab	T2	Terminal principal 2

### 2.2.6.2 Descripción del bloque luz artificial

Como se ha explicado anteriormente el acuario debe ser iluminado entre 10 y 12 horas, a más de que el acuario debe ser iluminado con luz artificial adecuada debido a que la luz natural posee una intensidad excesiva y promovería un crecimiento exagerado de las algas que a su vez mermarán el crecimiento de las plantas<sup>50</sup> se puede utilizar cualquier luz artificial para acuario, y en este caso se utiliza como luz artificial una lámpara sumergible de 12w , como muestra la figura 2.18, que trabaja con corriente alterna, teniendo la función de iluminar el acuario según el control que establece el bloque microcontrolador.

<sup>49</sup> Philips. (s.f.). Futurlec. Obtenido de <https://www.futurlec.com/Diodes/BT136.shtml>

<sup>50</sup> aquarium.lapipadelindio.com. (12 de mayo de 2011). Obtenido de <http://aquarium.lapipadelindio.com/tecnica/metodos-iluminacion-acuario>



**Figura 2.18** – Lámpara fluorescente

### 2.2.6.3 Descripción del bloque calentador de agua

Los peces tropicales suelen encontrarse en aguas que están a 23-26 grados centígrados y deben ser mantenidos a esta temperatura en un acuario.<sup>51</sup> Y si el acuario está en un ambiente de clima templado es necesario que el agua sea calentada y para poderlo calentar se usa un calentador sumergible. El calentador de agua utilizado en este proyecto es un calentador modelo HT-250 JAD, que tiene las especificaciones de trabajar con corriente alterna, desactivación automática según la temperatura a la que está y es de 50 W, como se muestra en la figura 2.19. Este calentador de agua es adaptado a la funcionalidad de este proyecto, por el hecho de su desactivación automática, el ajuste que se realizó en este calentador es excluir la función de desactivación automática teniendo una conexión directa con la corriente alterna y así controlarlo por el microcontrolador.



**Figura 2.19** – Calentador de agua

## 2.3 CIRCUITOS DEL PROTOTIPO

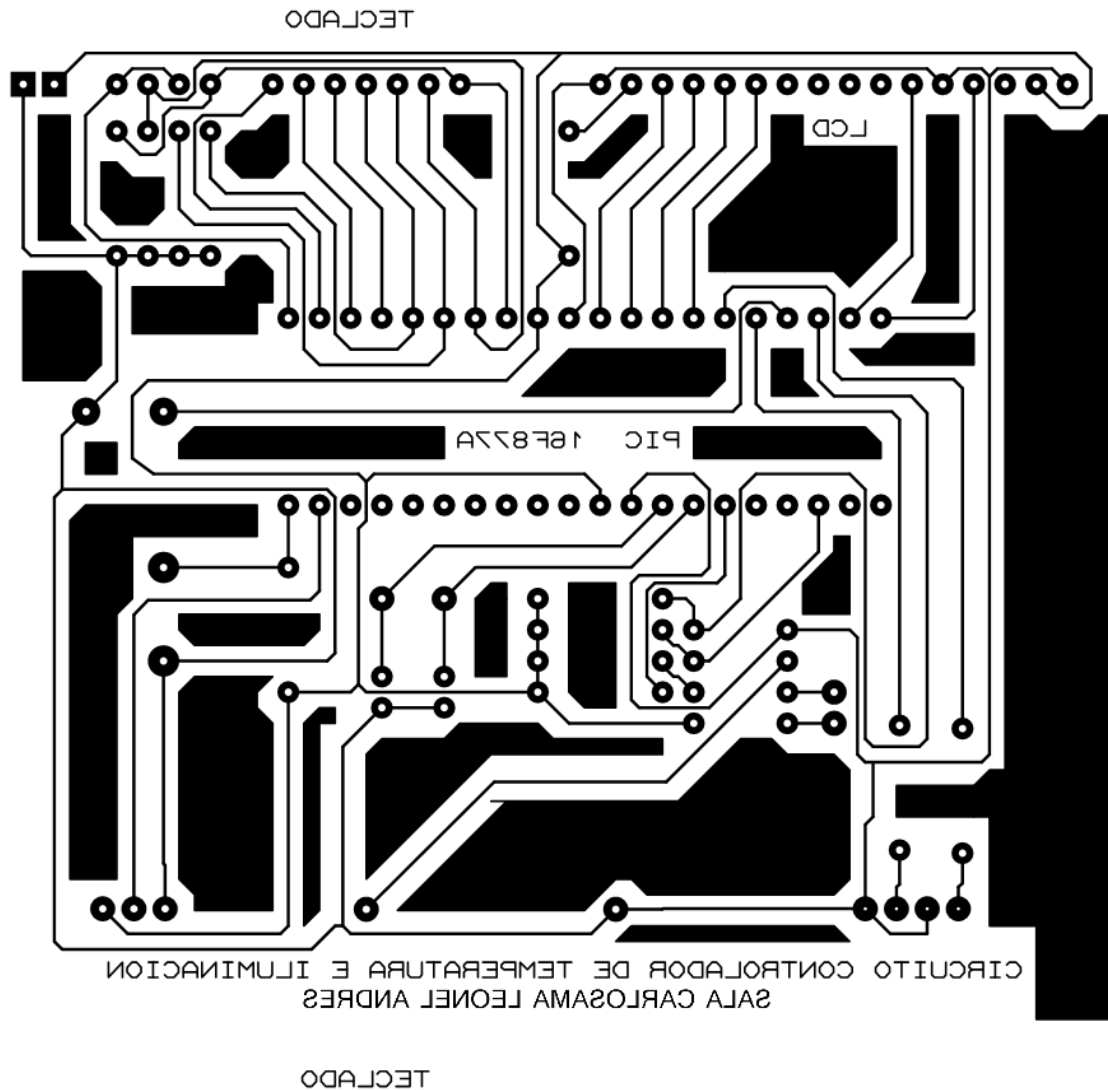
Luego de la descripción de cada elemento que compone el proyecto, se indica cómo van interconectados en las baquelitas.

La figura 2.20 se presenta las pistas impresas en la baquelita del circuito principal, y en la figura 2.21 se muestra los elementos soldados a la baquelita. Esta baquelita

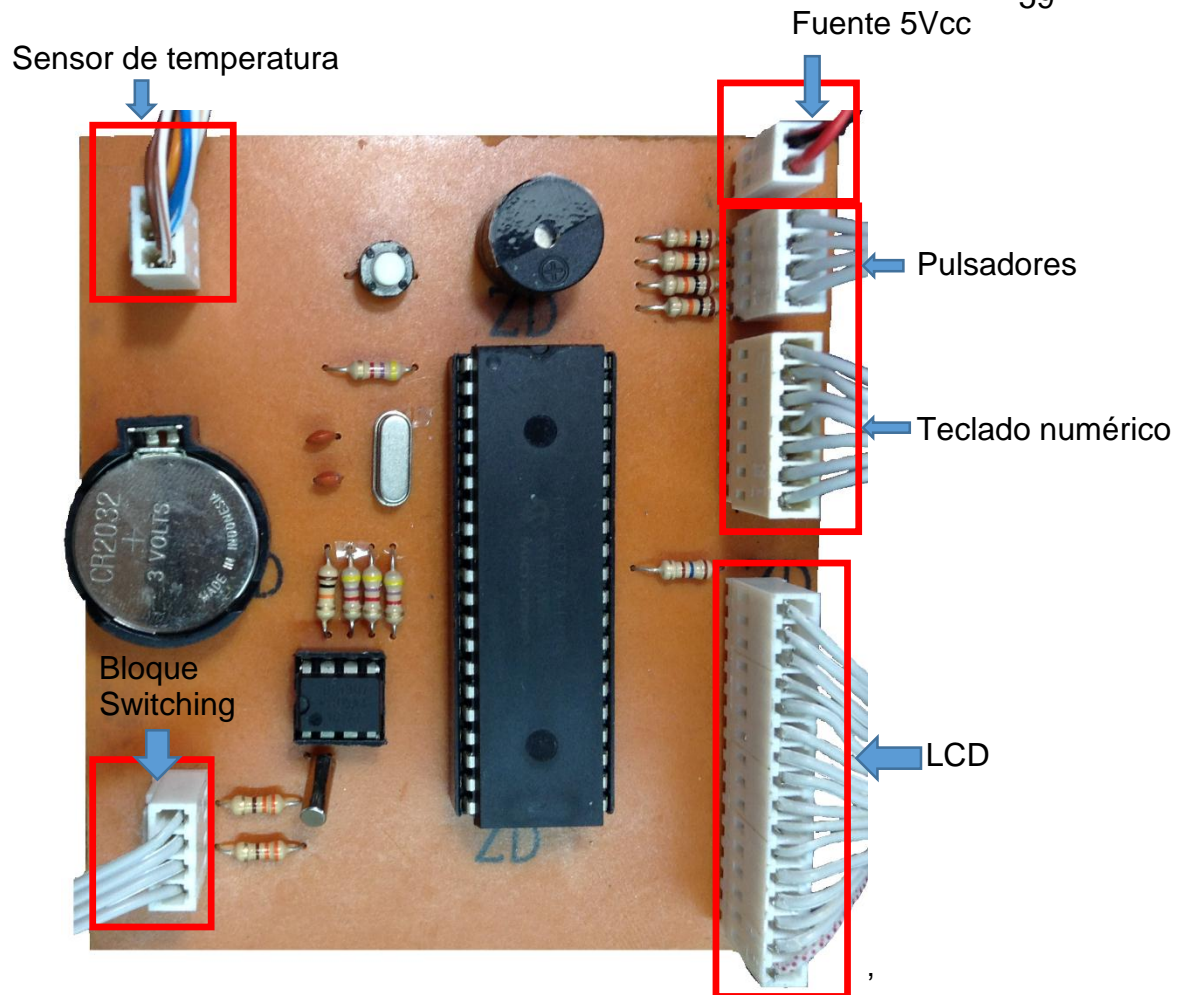
<sup>51</sup> Alibaba.com. (s.f.). Obtenido de <http://spanish.alibaba.com/product-gs/aquarium-thermostat-986462979.html>



tiene los elementos del bloque microcontrolador y del bloque de reloj también tiene las conexiones para los bloques: visualizador, sensor de temperatura, teclado, circuito de reloj y switching. Todo este circuito es implementado en función a la distribución de pines que se presentó en la tabla 2.3 y a las conexiones de configuración de cada elemento.



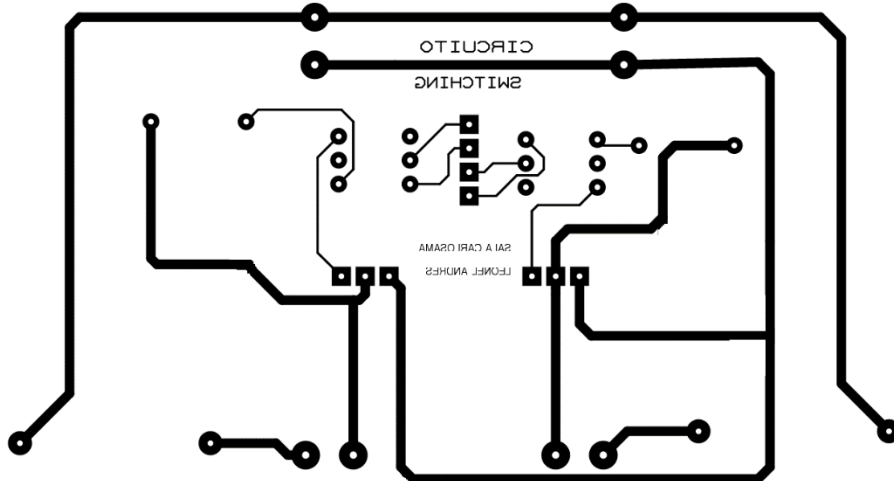
**Figura 2.20** - Pistas de la baquelita principal



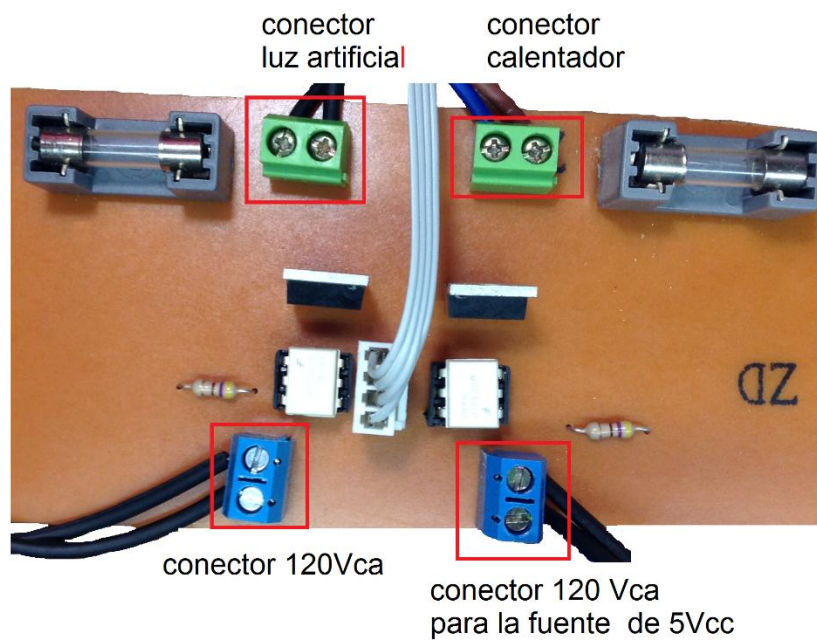
**Figura 2.21** - Baquelita principal

Para el bloque visualizador, sensor de temperatura y teclado no hay necesidad de una baquelita ya que sus elementos son conectados directamente a la baquelita del circuito principal

En la figura 2.22 se indica las pistas del circuito del bloque switching, conjuntamente con la figura 2.23 que indica los elementos soldados a la baquelita.



**Figura 2.22** - Pistas del bloque switching



**Figura 2.23** - Baquelita bloque switching

De los bloques luz artificial y calentador de agua, no se hacen baquelitas ya que los elementos de cada bloque son conectados directamente a la baquelita switching.

## CAPÍTULO 3

### DESARROLLO DE LA PROGRAMACIÓN

La lógica con la que trabaja el bloque microcontrolador dentro del proyecto, explicado en el capítulo anterior, está en función a una programación desarrollada en el compilador mikroC. Esta lógica es realizada en base a las necesidades y posibilidades de funcionamiento, siguiendo un ciclo de pasos bajo condiciones establecidas.

#### 3.1 FUNCIONAMIENTO DEL PROGRAMA POR DIAGRAMA DE FLUJO

El programa desarrollado en MikroC con el que el microcontrolador trabaja, está representado por los siguientes diagramas de flujo observado en las figuras 3.4, 3.5, 3.6.

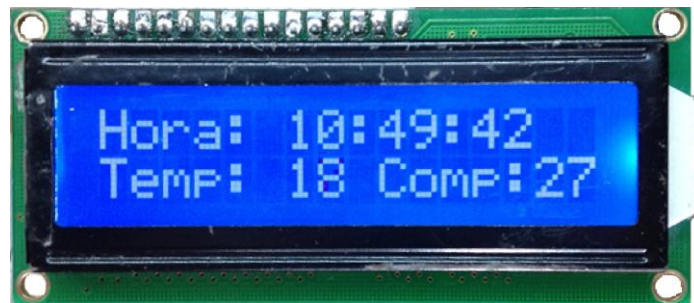
##### 3.1.1 DIAGRAMA DE FLUJO DE OPERACIÓN NORMAL

El inicio del diagrama de flujo se simboliza en la figura 3.4(a) y representa cuando el microcontrolador es conectado a la fuente de energía.

Para empezar las funciones de lógica, envío y recepción de datos por los puertos del microcontrolador, es necesario declarar los programas de control que el compilador facilita para el uso de los periféricos como es el LCD el DS1307, etc., también son declaradas las variables que se asocian a una lógica de función, y la declaración de los puertos para que sean de salida y/o entrada de datos, además una variable usada para la comparación de encender o apagar el calefactor y es asignada con el valor de 27 (luego puede ser modificada en la configuración de temperatura) .

Seguido, en la figura 3.4 (b), el microcontrolador entra a un ciclo indefinido de funcionamiento, al inicio del ciclo se asigna 0 al contador ya declarado al inicio de funcionamiento del microcontrolador, utilizado para el ingreso de datos en los

modos de configuración, luego se declara la espera de una tecla que posteriormente será aceptada para entrar a los modos de configuración. Hasta que el microcontrolador tenga la posibilidad de admitir el ingreso de una tecla, lee los datos del circuito de reloj por los terminales SCL y SDA, los codifica a decimal y los presenta en la pantalla de visualización, luego lee los datos del sensor de temperatura por medio del puerto de entrada analógica, multiplica por un factor para transformar el valor leído a grados centígrados, asigna ese valor a una variable para la comparación de temperatura, y presenta en la pantalla de visualización con el valor de comparación como se ve en la figura 3.1



**Figura 3.1** – Pantalla LCD en función normal

El ciclo continua con la comparación de la temperatura, si la temperatura es menor al comparador menos 1, es decir que si el agua está fría en referencia a la temperatura promedio, el calefactor se enciende, para que luego el microcontrolador continúe con el paso de comparación de la hora. Si en la comparación anterior el agua no está fría, comprueba si la temperatura es mayor al valor de comparación más uno, es decir si la temperatura está caliente en relación a la temperatura promedio, el calefactor se apaga, para que luego el microcontrolador continua con el paso de comparación de la hora.

La comparación de la hora es asociada a un valor ya predefinido para el encendido y apagado de la luz artificial. Primero compara si la hora leída del DS1307 es mayor a las 6 de la tarde o menor a las 6 de la mañana, es decir, las horas de la noche, para apagar la luz artificial y simular la noche y luego proseguir con siguiente paso. Si la comparación no cumple con las horas de la noche, se compara si la hora leída se encuentra entre las 6 de la mañana y las 6 de la tarde, es decir si son las horas

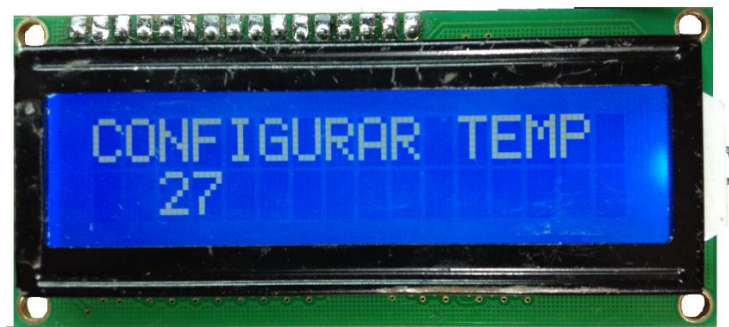
del luz solar, la luz artificial se enciende simulando el día y continua con el paso de comparación de la temperatura en casos de calentamiento o enfriamiento excesivos. La temperatura que se detecta a través del sensor puede ser mayor o menor al idóneo para los peces tropicales si es que existen casos de excepción, para evitar ese caso, la temperatura es permanentemente comparada entre los valores de 18 grados centígrados y 33 grados centígrados, como rango de temperatura idónea, si se sobrepasa este rango se emitirá una alarma de advertencia emitida por cada segundo.

Luego de ese paso se comprueba si una tecla ha sido presionada. La comparación de, si una tecla es presionada o no, reside en empezar el ciclo principal ya antes expuesto o identificar si las teclas son para ingresar a los menús de configuración. Si no se ha presionado ninguna tecla se vuelve al principio del ciclo principal, si una tecla es presionada y no es alguna de las teclas de configuración se vuelve al ciclo principal y si una tecla es presionada y el microcontrolador detecta como teclas ya designadas para el ingreso a los diferentes menús, cumple con las siguientes funciones.

### **3.1.2 DIAGRAMA DE FLUJO DE CONFIGURACIÓN DE TEMPERATURA**

En la figura 3.4(c) se observa que el diagrama de flujo compara si la tecla presionada es la tecla de temperatura, si es así el caso, el siguiente paso es el ingreso al menú de configuración de temperatura. El diagrama de flujo del menú de configuración de temperatura es expuesto en la figura 3.5. El inicio de este menú borra los datos que se presentan en la pantalla para poder observar la configuración. Luego entra en un ciclo el cual empieza con la asignación de las teclas numéricas (para la facilidad de comparación de que teclas se presionan y esperar la pulsación de una). Seguido se imprime en la pantalla "CONFIGURAR TEMP". Si se presiona una tecla, la lógica del funcionamiento comprueba si es una tecla numérica, si no es una tecla numérica continua para comprobar si es la tecla numeral "#", si no es la tecla numeral, se regresa al principio de este ciclo para esperar otra tecla, y si es la tecla numeral "#", la pantalla del LCD borra los datos de este menú, y se regresa a el ciclo principal.

Si la tecla que se presionó es una tecla numérica, es decir del 0 al 9, se continua con una comparación del contador, este contador si es cero la tecla se asigna al valor de decena del comparador, y si es uno el valor del contador, la tecla se asigna al valor de la unidad y luego se da el valor de -1 al contador. Luego de la asignación del valor de la tecla, se suma los valores de decena y unidad, se suma uno al contador, se imprime en el LCD el valor de comparación, se usa ese valor a la comparación de la temperatura y se regresa al inicio de este ciclo. La figura 3.2 presenta el menú de configuración de temperatura en el LCD.



**Figura 3.2** – Pantalla del menú de configuración de temperatura

### **3.1.3 DIAGRAMA DE FLUJO DE ENCENDIDO DE LUZ MANUAL**

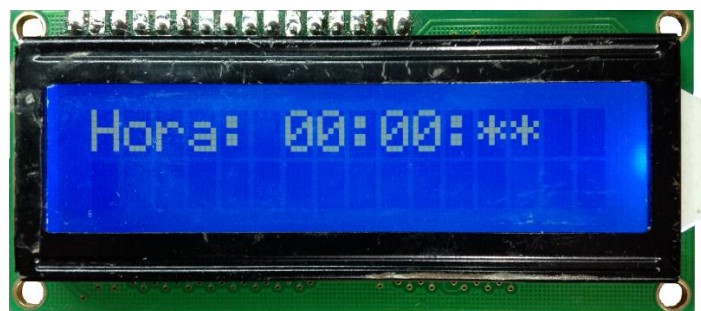
En la figura 3.4 (c) se observa que el diagrama de flujo compara si la tecla presionada es la tecla de temperatura, si no es así el caso, el siguiente paso es la comparación de si es la tecla de encendido de lámpara. Si se presionó la tecla de encendido o sea la tecla "\*" se continua a comparar si la luz artificial está encendida o apagada. Si la luz está encendida no se realiza ninguna acción y se regresa al ciclo principal, y si la luz está apagada, la pantalla del LCD presenta "LUZ ENCENDIDA FAVOR ESPERE" y se enciende la luz por 5 minutos. Luego de los 5 minutos se regresa al ciclo principal.



### 3.1.4 DIAGRAMA DE FLUJO DE CONFIGURACIÓN DE LA HORA

En la figura 3.4 (c) indica que si la tecla presionada no es la tecla de temperatura comprueba si es la tecla “\*”, y si no es la tecla “\*”, continua con el siguiente paso que se presenta en la figura 3.6.

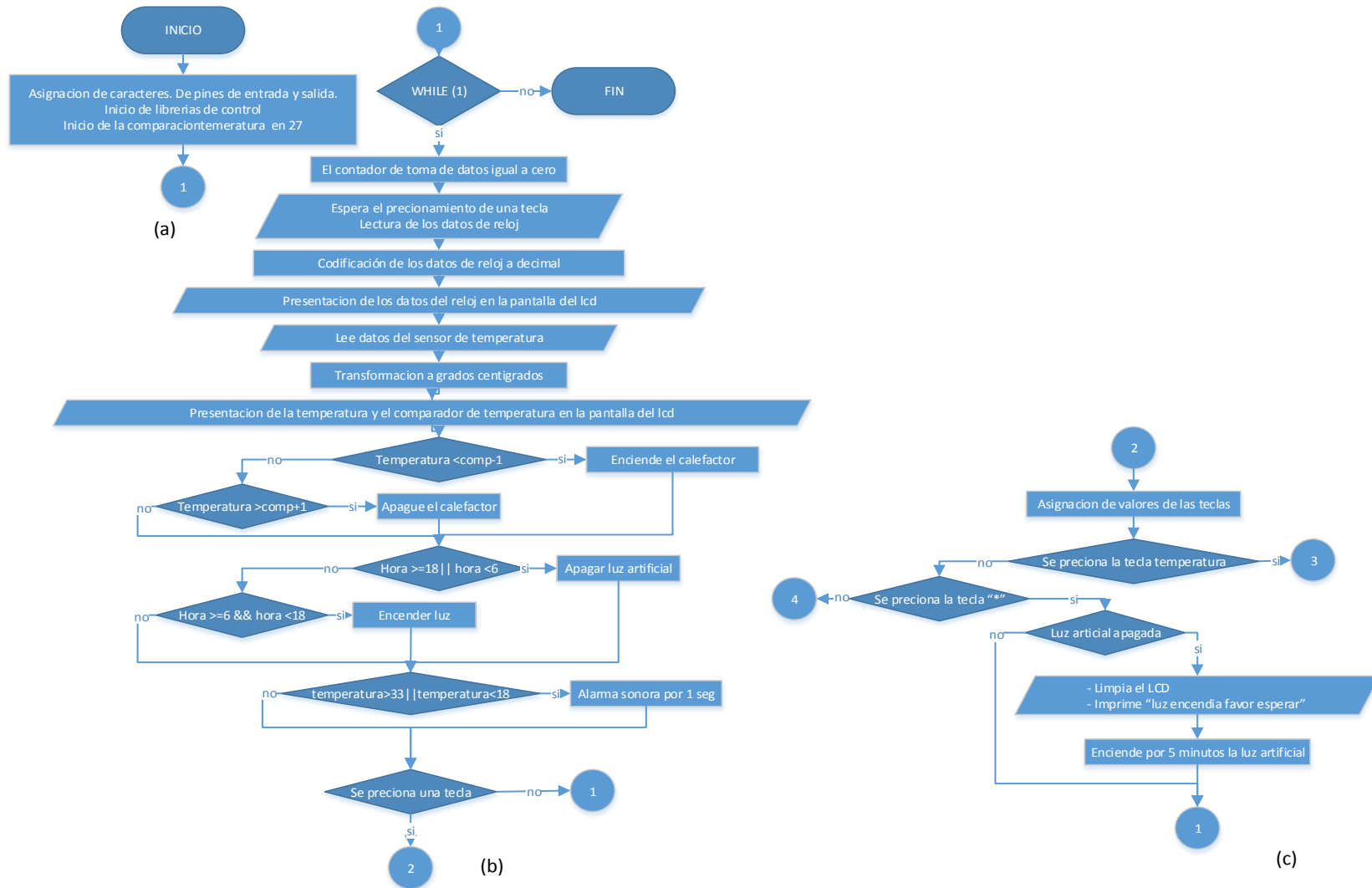
En la figura 3.6 comprueba si la tecla que se presiona es la tecla de hora, si no es la tecla de hora, se vuelve al ciclo principal. Si la tecla es hora, la lógica que se continúa es similar a la del menú de configuración de temperatura, en este caso se imprime la hora que se desea igualar, y las teclas que se presionó como datos numéricos se les asignan a la hora y minutos respectivamente. Con la excepción de que se aumenta un paso más que es de enviar esos datos al DS1307 codificados en BCD. La presentación de el menú de configuracion de temperatura se ve en la figura 3.3.



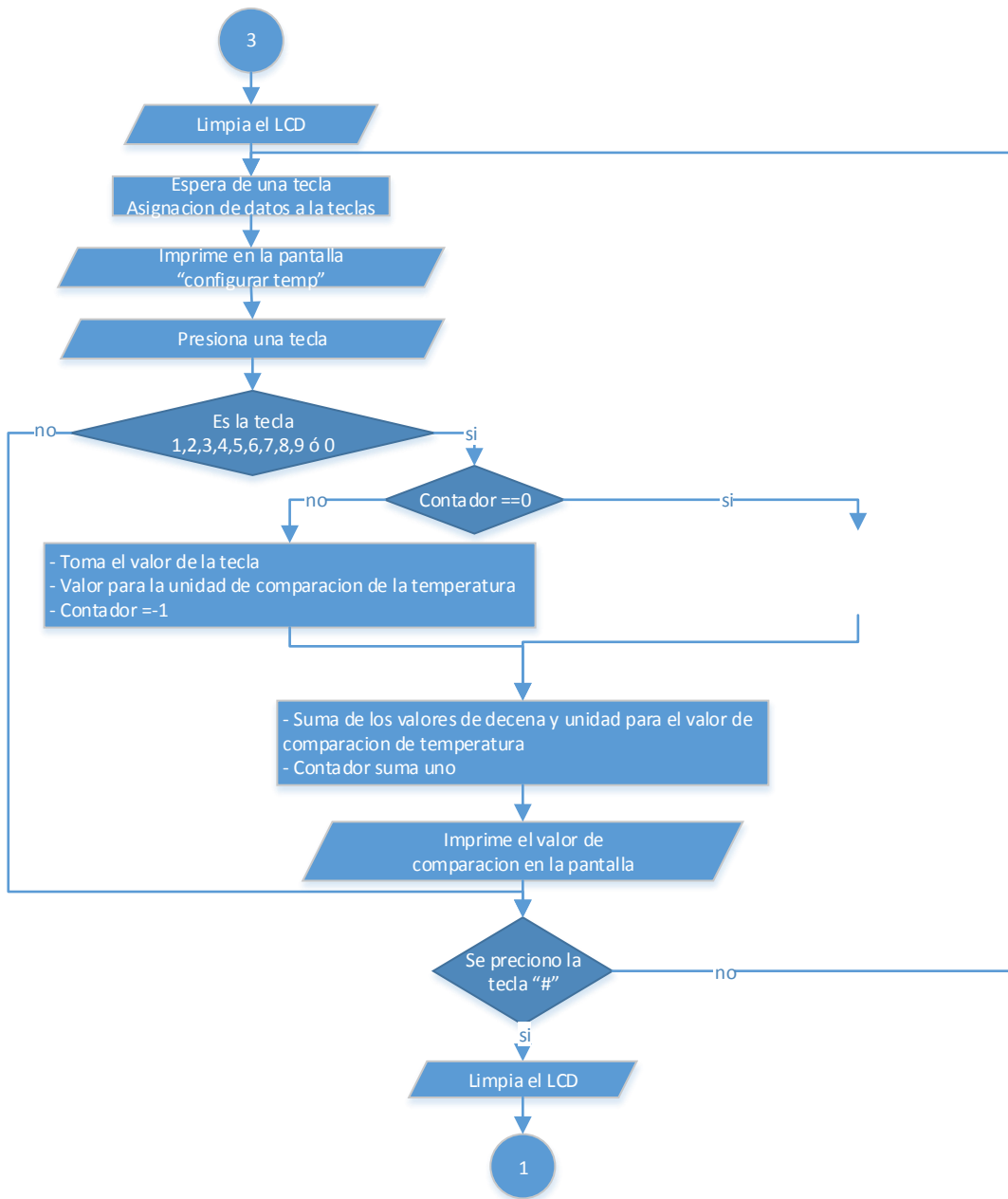
**Figura 3.3** – Pantalla del menú de configuración de la hora.

Cumpliendo así el funcionamiento completo del proyecto representado en diagrama de flujo.

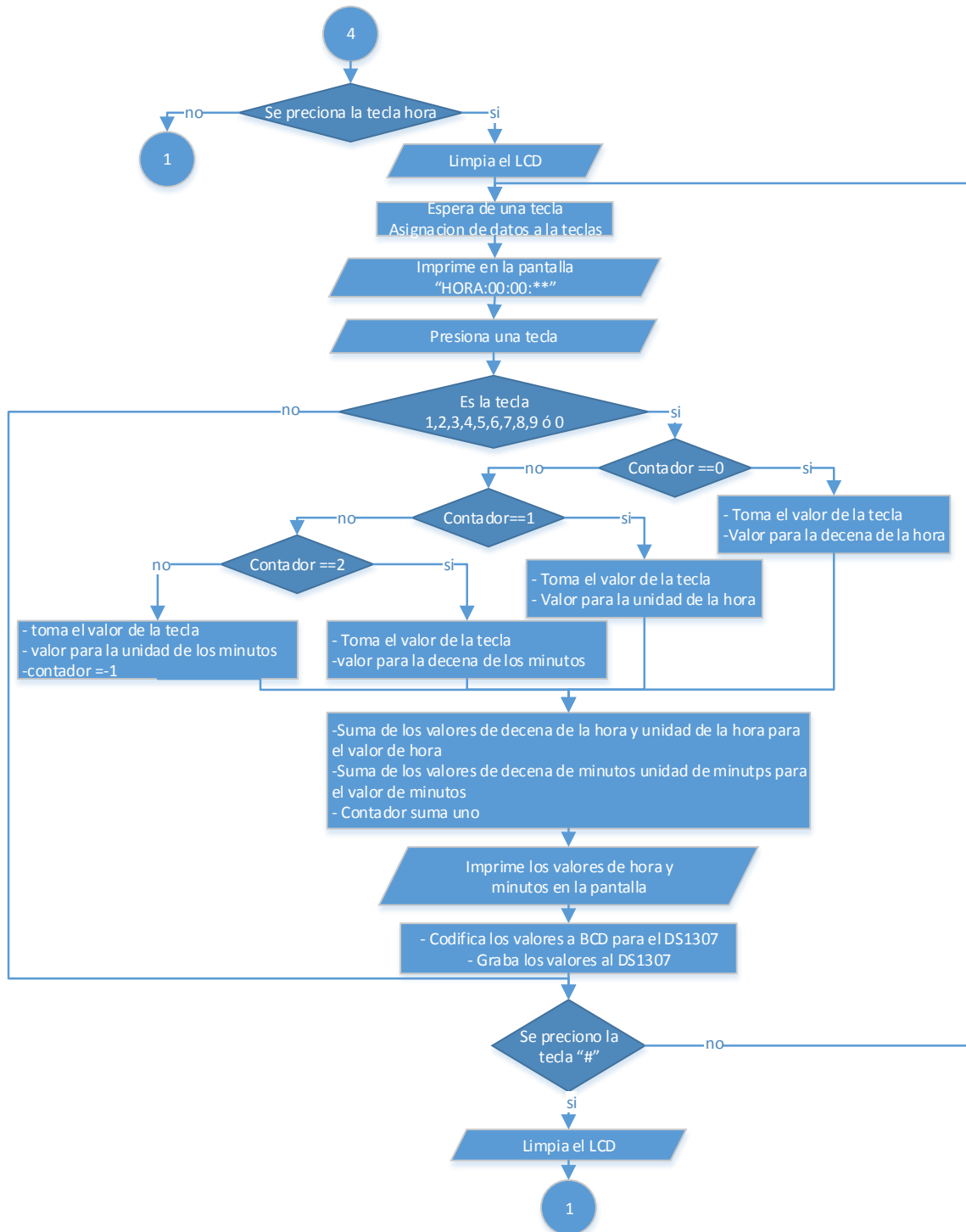




**Figura 3.4** – Diagrama de flujo primera parte (a) Inicio (b) Ciclo principal. (c) Inicio de menús y menú encendido de lámpara.



**Figura 3.5** – Diagrama de flujo menú configuración temperatura.



**Figura 3.6** – Diagrama de flujo menú configuración de la hora.

## 3.2 CÓDIGO DEL PROGRAMA

Toda la lógica expuesta en los diagramas de flujo es programada en el compilador MikroC. Esta programación comprende 294 líneas de código, representando 127 bytes de los 368 bytes de la memoria de datos del microcontrolador y 3268 palabras de las 8000 palabras de 14 bits de programación (datos que presenta el compilador luego de la compilación).

### 3.2.1 LÍNEAS DE CÓDIGO

La programación realizada en el compilador MikroC se presenta en las siguientes líneas de comando. Están numeradas acorde como en el compilador se muestra para generar un mejor entendimiento en las posteriores explicaciones sobre el programa.

```

1: char seconds,minutes,hours;
2: char arreglo[4];
3: short a1,c1,kp,valor,unidad,decena,u_comp,d_comp,comp,hora1=48,
    hora2=48,h1,h2,hora,min1=48,min2=48,m1,m2,minu;
4: signed short int contador;
5: char keypadPort at PORTB;
6:
7: sbit Soft_I2C_Scl at RC3_bit;
8: sbit Soft_I2C_Sda at RC4_bit;
9: sbit Soft_I2C_Scl_Direction at TRISC3_bit;
10: sbit Soft_I2C_Sda_Direction at TRISC4_bit;
11:
12: sbit LCD_RS at RD2_bit;
13: sbit LCD_EN at RD3_bit;
14: sbit LCD_D4 at RD4_bit;
15: sbit LCD_D5 at RD5_bit;
16: sbit LCD_D6 at RD6_bit;
17: sbit LCD_D7 at RD7_bit;
18:
19: sbit LCD_RS_Direction at TRISD2_bit;
20: sbit LCD_EN_Direction at TRISD3_bit;
21: sbit LCD_D4_Direction at TRISD4_bit;
22: sbit LCD_D5_Direction at TRISD5_bit;
23: sbit LCD_D6_Direction at TRISD6_bit;
24: sbit LCD_D7_Direction at TRISD7_bit;
25:
26: void Read_Time ()
27: {
28:   Soft_I2C_Start ();
29:   Soft_I2C_Write (0xD0);
30:   Soft_I2C_Write (0);

```

```

31: Soft_I2C_Start();
32: Soft_I2C_Write(0xD1);
33: seconds = Soft_I2C_Read(1);
34: minutes = Soft_I2C_Read(1);
35: hours = Soft_I2C_Read(0);
36: Soft_I2C_Stop();
37: }
38: void Transform_Time()
39: {
40:   seconds = ((seconds & 0xF0) >> 4)*10 + (seconds & 0x0F);
41:   minutes = ((minutes & 0xF0) >> 4)*10 + (minutes & 0x0F);
42:   hours = ((hours & 0xF0) >> 4)*10 + (hours & 0x0F);
43: }
44: void Display_Time()
45: {
46:   Lcd_Out(1,1,"Hora:");
47:   Lcd_Chr(1,7,(hours / 10) + 48);
48:   Lcd_Chr(1,8,(hours % 10) + 48);
49:   Lcd_Chr(1,9,':');
50:   Lcd_Chr(1,10,(minutes / 10) + 48);
51:   Lcd_Chr(1,11,(minutes % 10) + 48);
52:   Lcd_Chr(1,12,':');
53:   Lcd_Chr(1,13,(seconds / 10) + 48);
54:   Lcd_Chr(1,14,(seconds % 10) + 48);
55: }
56: void Init_Main()
57: {
58:   Soft_I2C_Init();
59:   Lcd_Init();
60:   Lcd_Cmd(_LCD_CLEAR);
61:   Lcd_Cmd(_LCD_CURSOR_OFF);
62:   Keypad_Init();
63:   Sound_Init(&PORTC, 5);
64: }
65: void temperatura()
66: {
67:   a1= ADC_READ(0);
68:   c1=(a1*0.49);
69:   ByteToStr(c1,arreglo);
70:   Lcd_Out(2,1,"Temp:");
71:   Lcd_Out(2,6,arreglo);
72:   Lcd_Out(2,10,"Comp:");
73: }
74: void Melody()
75: {
76:   Sound_Play(100, 50);
77:   Sound_Play(1000, 50);
78:   delay_ms(900);
79: }
80: void teclado()
81: {
82:   switch (kp)
83:   {
84:     case 1: kp = 49; valor = 1;break; //1
85:     case 2: kp = 52; valor = 4;break; //4
86:     case 3: kp = 55; valor = 7;break; //7
87:     case 4: kp = 42;break; // *
88:     case 5: kp = 50; valor = 2;break; //2
89:     case 6: kp = 53; valor = 5;break; //5
90:     case 7: kp = 56; valor = 8;break; //8

```

```

91:   case 8: kp = 48; valor = 0; break; //0
92:   case 9: kp = 51; valor = 3; break; //3
93:   case 10: kp = 54; valor = 6; break; //6
94:   case 11: kp = 57; valor = 9; break; //9
95:   case 12: kp = 35; break; //#
96:   case 13: kp = 104; break; //h
97:   case 14: kp = 116; break; //t
98: }
99: }
100: void tomardatos ()
101: {
102:   kp=Keypad_Key_click();
103:   teclado();
104:   Lcd_out(1,1,"CONFIGURAR TEMP");
105:   if (kp==49 || kp==52 || kp==55 || kp==50 || kp==53 || kp==56 || kp==48 || kp==51 ||
      kp==54 || kp==57)
106:   {
107:     if (contador==0)
108:     {
109:       decena=kp;
110:       d_comp=valor*10;
111:     }
112:     else
113:     {
114:       unidad=kp;
115:       u_comp=valor;
116:       contador=-1;
117:     }
118:     comp=d_comp+u_comp;
119:     contador++;
120:   }
121:   lcd_chr(2,3,decena);
122:   lcd_chr(2,4,unidad);
123: }
124: void igualarreloj ()
125: {
126:   kp=Keypad_Key_click();
127:   teclado();
128:   Lcd_Out(1,1,"Hora:");
129:   Lcd_Chr(1,7,hora1);
130:   Lcd_Chr(1,8,hora2);
131:   Lcd_Chr(1,9,':');
132:   Lcd_Chr(1,10,min1);
133:   Lcd_Chr(1,11,min2);
134:   Lcd_Chr(1,12,':');
135:   Lcd_Out(1,13,"**");
136:   if (kp==49 || kp==52 || kp==55 || kp==50 || kp==53 || kp==56 || kp==48 || kp==51 ||
      kp==54 || kp==57)
137:   {
138:     if (contador==0)
139:     {
140:       hora1=kp;
141:       h1=valor*10;
142:     }
143:     else
144:     {
145:       if(contador==1)
146:       {
147:         hora2=kp;
148:         h2=valor;

```

```

149:     }
150:     else
151:     {
152:         if(contador==2)
153:         {
154:             min1=kp;
155:             m1=valor*10;
156:         }
157:         else
158:         {
159:             min2=kp;
160:             m2=valor;
161:             contador=-1;
162:         }
163:     }
164: }
165: hora = Dec2Bcd(h1+h2);
166: minu=Dec2Bcd(m1+m2);
167:
168: Lcd_Chr(1,7,hora1);
169: Lcd_Chr(1,8,hora2);
170: Lcd_Chr(1,9,':');
171: Lcd_Chr(1,10,min1);
172: Lcd_Chr(1,11,min2);
173: Lcd_Chr(1,12,':');
174: Lcd_Out(1,13,"**");
175:
176: Soft_I2C_Start();
177: Soft_I2C_Write(0xD0);
178: Soft_I2C_Write(0x00);
179: Soft_I2C_Write(0x00);
180: Soft_I2C_Stop();
181:
182: Soft_I2C_Start();
183: Soft_I2C_Write(0xD0);
184: Soft_I2C_Write(0x01);
185: Soft_I2C_Write(minu);
186: Soft_I2C_Stop();
187:
188: Soft_I2C_Start();
189: Soft_I2C_Write(0xD0);
190: Soft_I2C_Write(0x02);
191: Soft_I2C_Write(hora);
192: Soft_I2C_Stop();
193: contador++;
194: }
195: }
196: void comparaciones()
197: {
198:     if(c1<comp-1)
199:     {
200:         PORTC.F6=1;
201:     }
202:     else
203:     {
204:         if(c1>comp+1)
205:         {
206:             PORTC.F6=0;
207:         }
208:     }

```

```

209:  if (hours >= 18 || hours < 6)
210:  {
211:    portc.f7 = 0;
212:  }
213:  else
214:  {
215:    if (hours >=6 && hours < 18)
216:    {
217:      portc.f7 = 1;
218:    }
219:  }
220: }
221: void luz ()
222: {
223:  if(portc.f7==0)
224:  {
225:    lcd_Cmd(_LCD_CLEAR);
226:    Lcd_Out(1,2,"LUZ ENCENDIDA");
227:    Lcd_Out(2,2,"FAVOR ESPERAR");
228:    portc.f7=1;
229:    delay_ms(300000);
230:    lcd_Cmd(_LCD_CLEAR);
231:  }
232: }
233: void main ()
234: {
235:  TRISB = 0xFF;
236:  TRISC.F6=0;
237:  TRISC.F7=0;
238:  adcon1=0X82;
239:  adcon0=0B00000001;
240:  Init_Main();
241:  comp=27;decena=50;unidad=55;
242:  while (1)
243:  {
244:    contador=0;
245:    kp=0;
246:    do
247:    {
248:      kp=Keypad_Key_press();
249:      Read_Time();
250:      Transform_Time();
251:      Display_Time();
252:      temperatura();
253:      lcd_chr(2,15,decena);
254:      lcd_chr(2,16,unidad);
255:      comparaciones();
256:      if (c1>33||c1<18)
257:      {
258:        Melody();
259:      }
260:    }
261:    while (!kp);
262:    teclado();
263:    if (kp==116)
264:    {
265:      lcd_Cmd(_LCD_CLEAR);
266:      do
267:      {
268:        tomardatos();

```



```

269:  }
270:  while (kp!=35);
271:  lcd_Cmd(_LCD_CLEAR);
272:  }
273:  else
274:  {
275:    if (kp==42)
276:    {
277:      luz();
278:    }
279:    else
280:    {
281:      if (kp==104)
282:      {
283:        lcd_Cmd(_LCD_CLEAR);
284:        do
285:        {
286:          igualarreloj();
287:        }
288:        while (kp!=35);
289:        lcd_Cmd(_LCD_CLEAR);
290:      }
291:    }
292:  }
293: }
294: }

```

### 3.2.1.1 Asignación de variables

En las primeras 5 líneas de código del programa están determinadas las variables que serán utilizadas, y dependiendo del valor que tiene asignado, cumplan con la lógica que se implementa, dependiendo del tipo de variable será la utilización que se tiene, en la tabla 3.1 se presenta para qué se utiliza la variable y el tipo.

**Tabla 3.1** – Variables utilizadas en las líneas del código

Nombre de variable	Tipo	Uso
seconds	char	Asignar el valor leído de la dirección de segundos del DS1307 para la presentación de los segundos en el LCD
minutes	char	Asignar el valor leído de la dirección de minutos del DS1307 para la presentación de los minutos en el LCD
hours	char	Asignar el valor leído de la dirección de hora del DS1307 para la presentación de la hora en el LCD
a1	short	Asignar el valor de lectura del sensor de temperatura.
c1	short	Asignar el valor de lectura del sensor de temperatura transformado en grados centígrados, para las operaciones de comparación.
kp	short	Representación del valor en ASCII de cada tecla para la presentación en LCD
valor	short	Representación del valor numérico que representa cada tecla del teclado numerico.

Nombre de variable	Tipo	Uso
unidad	Short	Asignar el valor en ASCII de la tecla presionada a la cifra unidad del valor de comparación de la temperatura (solo para la presentación en el LCD).
decena	Short	Asignar el valor en ASCII de la tecla presionada a la cifra decena del valor de comparación de la temperatura (solo para la presentación en el LCD).
u_comp	Short	Asignar el valor numérico de la tecla presionada a la cifra unidad del valor de comparación de la temperatura
d_comp	short	Asignar el valor numérico de la tecla presionada a la cifra decena del valor de comparación de la temperatura
comp	Short	Asignar la suma de los valores de u_comp y d_comp para el uso de comparación.
hora1	Short	Asignar el valor en ASCII de la tecla presionada a la cifra decena del valor de hora (solo para la presentación en el LCD). Establecido con valor predeterminado en ASCII igual a cero para el menú de ingreso de datos de la hora.
hora2	short	Asignar el valor en ASCII de la tecla presionada a la cifra unidad del valor de hora (solo para la presentación en el LCD). Establecido con valor predeterminado en ASCII igual a cero para el menú de ingreso de datos de la hora.
h1	short	Asignar el valor numérico de la tecla presionada a la cifra unidad de hora.
h2	Short	Asignar el valor numérico de la tecla presionada a la cifra unidad de la hora.
hora	Short	Asignar la codificación de decimal a BCD de la suma de h1 y h2 para registrarlo al DS1307.
min1	Short	Asignar el valor en ASCII de la tecla presionada a la cifra decena del valor de minutos (solo para la presentación en el LCD). Establecido con valor predeterminado en ASCII igual a cero para el menú de ingreso de datos de los minutos.
min2	Short	Asignar el valor en ASCII de la tecla presionada a la cifra unidad del valor de minutos (solo para la presentación en el LCD). Establecido con valor predeterminado en ASCII igual a cero para el menú de ingreso de datos de los minutos.
m1	Short	Asignar el valor numérico de la tecla presionada a la cifra decena de los minutos.
m2	Short	Asignar el valor numérico de la tecla presionada a la cifra unidad de los minutos.
minu	Short	Asignar la codificación de decimal a BCD de la suma de h1 y h2 para registrarlo al DS1307.
arreglo[4]	Char	Carácter de 4 espacios de arreglo para asignación del valor de temperatura en grados centígrados y presentación en LCD
contador	Signed short int	Variable utilizada para el conteo de las cifras de ingreso en los menús de configuración de temperatura y hora
keypadport at PORTB	Char	Asignación de los caracteres de ingreso de datos del teclado al puerto B

### **3.2.1.2 Comunicación con I2C al DS1307**

Entre las líneas de código 7 y 10 son las declaraciones de los pines I2C para la comunicación con el DS1307, declarando los pines RC3 y RC4 (18 y 23 de la distribución de pines respectivamente) del PIC16F877A.

### **3.2.1.3 Comunicación con el LCD**

Los pines que se usan para la comunicación con el LCD se especifican entre las líneas de código 12 y 24 declarando explícitamente los pines que se conectarán del LCD y así enviar los datos correctamente.

### **3.2.1.4 Lecturas de los datos del DS1307**

Los datos que se presentan en el LCD y utilizados para las comparaciones lógicas en el encendido y apagado de la luz artificial dependen de la sentencia de lectura del DS1307 establecida en el void Read\_time entre las líneas de código 26 y 37, empezando con el arranque de comunicación con el DS1307, luego si no tiene valores almacenados escribe un registro de 0, para en seguida establecer, con la dirección de lectura, la asignación de los valores leídos a la hora, minutos y segundos y finalmente cerrando la comunicación.

### **3.2.1.5 Codificación de datos leídos del DS1307**

Como los valores que se leen del DS1307 y asignados a la hora minuto y segundos son valores en BCD, es necesario codificarlos a decimal para la utilización y sin equivocación en las sentencias de comparación. En el void Transform\_time entre líneas de código 38 y 43, son las responsables de la codificación de los datos leídos del DS1307.

### 3.2.1.6 Presentación de la hora minutos y segundos en el LCD

En el LCD solo se puede presentar valores de tipo caracter según su valor en ASCII , o tipo string, es por eso que en en void Display\_Time entre las líneas de código 44 y 55 se declara la salida de la hora al LCD 16x2.

En esta declaración se establece los lugares asignados para cada variable signo o palabra y también la declaración de una codificación de los valores de la hora, minutos y segundos de decimal a carácter ASCII para poderlos presentar. En la tabla 3.2 se muestra los caracteres utilizados del LCD.

**Tabla 3.2** - Caracteres utilizados en la primera fila del LCD 16x2.

Valor o palabra	Carácter del LCD
"Hora:"	Del 1 al 5
Espacio vacío	6
hora	7 y 8
“.”	9
Minutos	10 y 11
“.”	12
segundos	13 y 14

### 3.2.1.7 Programas secundarios.

Las comunicaciones con el DS1307, el LCD, el teclado y la salida de sonido utilizan cada uno un programa controlador y están declarados en el void Init\_Main entre las líneas de código 56 y 64. Estos programas tienen funciones de declaración para la comunicación que se desee con los periféricos.

En la línea 58 se declara el inicio del programa controlador para la comunicación con el DS1307, en la línea 59 el inicio del programa controlador la comunicación con el LCD, en la línea 62 el inicio del programa para la comunicación con las funcionalidades del teclado, y en la línea 63 la declaración del inicio del programa del sonido asignando el pin rc5 (pin 24) donde está conectado el buzzer.

### 3.2.1.8 Datos del sensor de temperatura

La lectura de la señal del sensor de temperatura y la presentación en la pantalla del LCD es declarada en el void\_temperatura entre las líneas de código 65 y 73,

asignando el valor leído a la variable a1 en la línea 67, la transformación a grados centígrados en la línea 68 y codificación a valor de string para que el LCD pueda presentarlo.

La presentación de la temperatura ocupa la fila 2 del LCD y en la tabla 3.3 se muestra los caracteres utilizados del LCD.

**Tabla 3.3** - Caracteres utilizados en la segunda fila del LCD 16x2

Valor o palabra	Carácter del LCD
"Temp:"	Del 1 al 5
Arreglo[4]	6 al 8
"Comp:"	Del 10 al 14

### 3.2.1.9 Sonidos aviso

El sonido que se emite si existen valores de temperatura extremos está declarado en el void Melody entre las líneas de código 74 y 79. Para emitir el sonido de aviso se usa 2 tonos con frecuencias y tiempos de sonidos. El primer tono, declarado en la línea 76, tiene una frecuencia de 100 Hz y un tiempo de 50 milisegundos, el segundo, sonido declarado en la línea 77, tiene una frecuencia de 1000 Hz y 50 milisegundos, para completar el tiempo de 1 segundo se declara un retraso de 900 milisegundos.

### 3.2.1.10 Barrido del teclado

Los valores que se presentan en la pantalla del LCD y los valores que se usan para las comparaciones de lógica que se asume al presionar una tecla del teclado numérico son diferentes, en el void teclado, entre las líneas de código 80 y 99, se asigna estos valores según sea el caso de la tecla presionada. El orden que se sigue para los casos de la tecla presionada es empezando por la columna izquierda del teclado, es decir, como se ve en la figura 3.7, el primer caso es el número 1 el segundo el número 4 el tercero el número 7, el cuarto el asterisco, el quinto el número 2, etc., hasta el caso 12 que es el numeral. Como el programa de teclado es declarado para un teclado de 4x4, el resto de casos sería en otra columna de teclas, en las que solo 2 teclas son asignadas, siendo el caso subsiguiente 13 para

la tecla de configuración de la hora y el caso 14 para la tecla de configuración de la temperatura.



**Figura 3.7** – Teclado 4x3.

Dependiendo del caso es la asignación del valor en ASCII y el valor en número de la tecla presionada, como se ha explicado el valor del carácter en ASCII es para la presentación en la pantalla y el valor en número para las funciones lógicas de comparación.

#### **3.2.1.11 Asignación del valor de comparación de temperatura**

El valor ingresado por el teclado en el menú de configuración de temperatura está declarado en el void tomardatos entre las líneas de código 100 y 123. Todo el procedimiento que cumple este void es el de la lógica del diagrama de flujo ya explicado anteriormente, tomando el valor de temperatura según los casos de la tecla presionada utilizando el contador.

#### **3.2.1.12 Asignación de los valores de la hora**

Los valores ingresados por el teclado en el menú de configuración de la hora usado para igualar el reloj, está declarado en el void igualarreloj entre las líneas de código 124 y 195. Todo el procedimiento que cumple este void es el de la lógica del diagrama de flujo de configuración de la hora ya explicado anteriormente, tomando el valor de la hora según los casos de la tecla presionada y utilizando el contador.

En el proceso de enviar los datos al DS1307 depende primero, de la codificación del valor de la hora y minutos declarado en las líneas de comandos 165 y 166 respectivamente para luego enviarlos al DS1307 con las direcciones que se asigna entre la línea de código 182 y 186 para los minutos y entre las líneas de código 188 y 192 para la hora.

#### **3.2.1.13 Lógica de comparaciones**

Lo explicado en el diagrama de flujo de operación normal, en la sección donde se compara si se prenden o apagan la luz artificial y el calentador de agua dependiendo de los valores asignados, está declarado en el void comparaciones entre la línea de comandos 196 y 220.

#### **3.2.1.14 Encendido manual de la luz artificial**

La funcionalidad de encender la luz por 5 minutos si ésta está apagada, es declarada entre la línea de comandos 221 y 232 cumpliendo las condiciones desarrolladas en el diagrama de flujo.

#### **3.1.1.15 Funcionamiento del programa principal**

La recopilación de todas las declaraciones y cargos explicados en las líneas de código anteriores, utilizadas según la lógica ya explicada en las figuras del diagrama de flujo, son declaradas en el void main entre las líneas de comando 233 y 294 empezando con las declaraciones de los puertos y continuando con las funciones principales del proyecto. Siendo el ciclo principal entre las líneas de código 248 y 260, el ingreso al menú de configuración de temperatura entre las líneas de código 263 y 272, el encendido de la luz artificial entre las líneas de código 275 y 278, y el ingreso al menú de configuración de la hora entre las líneas de código 281 y 290.

## CAPÍTULO 4

### PRUEBAS Y RESULTADOS

Luego del desarrollo del programa y compilado al PIC, el acoplamiento según su funcionamiento de cada elemento dentro del proyecto y el montaje de los elementos en las baquelitas, es importante validar que todo en conjunto funcione de manera correcta según lo planteado. Para estos se generan diferentes escenarios con cada posibilidad que el proyecto se pueda enfrentar. Las posibilidades de cada escenario tienen relación a la función del proyecto, es decir, que se evalúa en el día y en la noche y si el agua está fría o caliente.

Teniendo como referencia que el día empieza a las seis de la mañana (seis horas) y termina a las 6 de la tarde (18 horas).

La tabla 4.1 indica cual es el comportamiento deseado y el comportamiento real de la lámpara ante las posibilidades.

**Tabla 4.1-** Comportamiento de la lámpara

<b>HORA DEL DÍA</b>	<b>Comportamiento deseado</b>	<b>Comportamiento real</b>
0	Apagada	Apagada
1	Apagada	Apagada
De 2 a 5	Apagada	Apagada
6	Se enciende	Se enciende
7	Encendida	Encendida
8	Encendida	Encendida
De 9 a 15	Encendida	Encendida
16	Encendida	Encendida
17	Encendida	Encendida
18	Se apaga	Se apaga
19	Apagada	Apagada
20	Apagada	Apagada
De 20 a 23	Apagada	Apagada



Los datos que presenta la tabla 4.1 tienen referencia al proyecto funcionando en su totalidad, es decir, que también el sensor de temperatura y el control del calentador del agua estaban funcionando. También los datos son tomados en el transcurso de las horas y no por configuraciones directas en la hora del sistema.

Como bien es sabido el calentador de agua es activado y desactivado en relación al valor de comparación que se ingresa, y el valor que detecta el sensor.

Los datos de la tabla 4.2 son tomados desde que el agua es vertida en el acuario, esta agua es fría y no afecta a que la prueba de funcionamiento sea la correcta. Lo que si toca tener en cuenta es cuando el pez es colocado en el acuario, ya que dependiendo del calentador, el tiempo en que el agua se caliente es amplio.

La tabla 4.2 muestra el comportamiento del calentador del agua según los datos que se presenten

**Tabla 4.2-** Comportamiento del calentador de agua

<b>Temperatura que detecta el sensor</b>	<b>Valor de comparación</b>	<b>Comportamiento deseado</b>	<b>Comportamiento real</b>
De 15 a 19	20	Encendido	Encendido
20	20	Encendido	Encendido
21	20	Apagado	Apagado
20	20	Apagado	Apagado
19	20	Encendido	Encendido
20	21	Encendido	Encendido
21	21	Encendido	Encendido
22	21	Apagado	Apagado
21	21	Apagado	Apagado
20	21	Apagado	Apagado
19	21	Encendido	Encendido
20	19	Apagado	Apagado
19	19	Apagado	Apagado
18	19	Encendido	Encendido
19	19	Encendido	Encendido
20	19	Apagado	Apagado

Los valores que presenta la tabla son tomados en el transcurso de su funcionalidad y sin verter agua caliente o agua fría. Esta tabla también refleja que el proyecto funciona correctamente y que con cualquier valor de comparación que se configure, el funcionamiento es el deseado.

Las funciones secundarias que han sido programadas en el PIC, como el encendido de luz por cinco minutos cuando está apagada, el sonido de aviso en casos extremos de temperatura, la configuración de actualización de la hora del sistema y la configuración del valor de comparación, han sido puesto en práctica y generando un buen resultado de prueba a lo planteado a los capítulos anteriores.

## CAPÍTULO 5

### CONCLUSIONES Y RECOMENDACIONES

#### 5.1 CONCLUSIONES

- Se mejora el cuidado de los peces tropicales por medio de la construcción del prototipo de detección y control electrónico de temperatura e iluminación para el ecosistema de un acuario de peces tropicales en un clima templado, evitando totalmente el descuido por el control manual
- La construcción del prototipo es aplicado a las necesidades de los peces tropicales que se mantienen en un clima templado, especialmente activando y desactivando el calentador de agua.
- Si el prototipo es usado en un clima más caliente, en referencia al ambiente de los peces tropicales, no funcionaría correctamente, ya que el prototipo no enfría el agua, solo la calienta.
- El sensor LM35 funciona correctamente a la necesidad de detectar la temperatura dentro del agua, ya que tiene un circuito de filtración de ruido y además una permeabilización que evita el cortocircuito.
- La dependencia de la potencia del calentador por la cantidad de agua en el acuario se mantiene, pero el ajuste que se debe dar para calentar el agua se elimina a razón del funcionamiento que tiene el proyecto.
- Ya no habrá problema si se coloca un calentador que sobrepase la potencia de referencia que se tiene en función a los litros de agua, porque con el proyecto la referencia es la temperatura del agua del acuario.
- El microcontrolador PIC16f877A cumple con los parámetros para acoplar los elementos al sistema referenciando la lógica del programa desarrollado.

- Se comprueba que la función de encender y apagar el calentador de agua, depende de la temperatura que detecta el LM35.
- Usar el CI DS1307, que emite la hora del día, para encender o apagar la luz artificial, es más eficiente porque se usan valores reales mejorando la precisión del control.

## **5.2 RECOMENDACIONES**

- En el acuario usar un oxigenador alado del calentador de agua, porque el movimiento ayuda a que el agua caliente fluya por el acuario y tener más equitativa la temperatura del agua
- También para que haya un mejor calentamiento del agua, colocar en la parte inferior del acuario al calentador, para que el agua caliente, por propiedades químicas, suba, y el agua fría se caliente.
- Para tener una buena detección de la temperatura del agua y así un mejor funcionamiento del calentamiento de agua, colocar el sensor de temperatura al lado contrario del que está ubicado el calentador de agua
- Es bastante recomendable no colocar el acuario en lugares que tengan luz natural directa, porque se evita que el agua se sobrecaliente y que haya una rápida proliferación de algas en el acuario.

## BIBLIOGRAFÍA

*Acuario de Arrecife*. (s.f.). Obtenido de

<http://www.acuariodearrecife.com/parametros/temperatura-2.htm>

*Acuario Marino*. (s.f.). Obtenido de

<http://www.acuariomarino.com.ar/es/iluminacion.html>

Alanoca, G. M. (s.f.). *Scribd*. Obtenido de

<http://es.scribd.com/doc/217825101/Teclado-Matrical>

*Alibaba.com*. (s.f.). Obtenido de <http://spanish.alibaba.com/product-gs/aquarium-thermostat-986462979.html>

*aquarium.lapipadelindio.com*. (12 de mayo de 2011). Obtenido de

<http://aquarium.lapipadelindio.com/tecnica/metodos-iluminacion-acuario>

*Aquaterraria.com*. (20 de enero de 2011). Obtenido de

<http://www.aquaterraria.com/content/section/197-calentadores.html>

Artemisa. (s.f.). Obtenido de

<http://artemisa.unicauca.edu.co/~gavasquez/res/Sensores.pdf>

Biomédica, E. E. (s.f.). *Electrónica de Potencia*. Obtenido de

<http://www.udb.edu.sv/udb/archivo/guia/electronica-tecnologico/electronica-de-potencia/2014/i/guia-4.pdf>

Blanco, A. (s.f.). Obtenido de [http://peces-](http://peces-tropicales.idoneos.com/index.php/Generalidades/Iluminacion_acuario)

[tropicales.idoneos.com/index.php/Generalidades/Iluminacion\\_acuario](http://peces-tropicales.idoneos.com/index.php/Generalidades/Iluminacion_acuario)

Bugueño, M. A. (s.f.). Obtenido de <http://snsordesde temperatura.blogspot.com/>

Carletti, E. J. (s.f.). *Robots*. Obtenido de [http://robots-](http://robots-argentina.com.ar/Comunicacion_busI2C.htm)

[argentina.com.ar/Comunicacion\\_busI2C.htm](http://robots-argentina.com.ar/Comunicacion_busI2C.htm)

Ceduvirt. (s.f.). Obtenido de

<http://www.ceduvirt.com/resources/Microcontroladores.pdf>

CXmat, L. (s.f.). *SCRIBD*. Obtenido de

<http://es.scribd.com/doc/94364908/Pulsadores>

Dotti, A. (s.f.). *TARINGA*. Obtenido de

<http://www.taringa.net/posts/mascotas/3198525/Guia-de-peces-tropical.html>

Fernández-Caparrós, A. M. (s.f.). Obtenido de

<http://www.uco.es/~el1mofer/Docs/IntPerif/Bus%20I2C.pdf>

García, V. (s.f.). *his pavila*. Obtenido de

[http://his pavila.com/3ds/ics/ds1307\\_esp.pdf](http://his pavila.com/3ds/ics/ds1307_esp.pdf)

*hola.com*. (22 de octubre de 2012). Obtenido de

<http://mascotas.hola.com/peces/acuario-peces-tropicales/244>

<http://www.fing.edu.uy/tecnoinf/mvd/cursos/arqcomp/material/teorico/arq-teorico05.pdf>

<http://www.utp.edu.co/~eduque/arquitect/PIC16F877.pdf>

<http://www.utp.edu.co/~eduque/arquitect/PIC16F877.pdf>

Inc, M. T. (s.f.). *microchip.com*. Obtenido de

<http://ww1.microchip.com/downloads/en/DeviceDoc/39582b.pdf>

*INGENIATIC*. (s.f.). Obtenido de

<http://ingeniatic.euitt.upm.es/index.php/tecnologias/item/519-microprocesador>

Instruments, T. (s.f.). *ti*. Obtenido de <http://www.ti.com/lit/ds/symlink/lm35.pdf>

ISICOM. (s.f.). *Datasheetcatalog*. Obtenido de

[http://pdf.datasheetcatalog.com/datasheet\\_pdf/isocom/MOC3009\\_to\\_MOC3012X.pdf](http://pdf.datasheetcatalog.com/datasheet_pdf/isocom/MOC3009_to_MOC3012X.pdf)

Jimenez, O. (s.f.). *Triac*. Obtenido de <http://triac.blogia.com/>

*leelo.co*. (s.f.). Obtenido de [http://leelo.co/acuario-iluminación-es-importante-para-la-salud-de-tu-amigo-con-aletas-de\\_13b68f.html](http://leelo.co/acuario-iluminación-es-importante-para-la-salud-de-tu-amigo-con-aletas-de_13b68f.html)

*medirtemperatura.com*. (s.f.). Obtenido de <http://medirtemperatura.com/sensor-temperatura.php>

*MikroElektronika*. (s.f.). Obtenido de

<http://www.mikroe.com/chapters/view/80/capitulo-2-programacion-de-los-microcontroladores/>

Palace, B. (s.f.). Obtenido de <http://cosmicdevices.tradeindia.com/electronic-buzzers-179625.html>

Philips. (s.f.). *Futurlec*. Obtenido de <https://www.futurlec.com/Diodes/BT136.shtml>

*Plan Ceibal*. (s.f.). Obtenido de

[http://www.ceibal.edu.uy/contenidos/areas\\_conocimiento/mat/midiendoeltiempo/el\\_reloj.html](http://www.ceibal.edu.uy/contenidos/areas_conocimiento/mat/midiendoeltiempo/el_reloj.html)

reparo, Y. (s.f.). Obtenido de

<http://electronica.yoreparo.com/electronica/1075528.html>

Rosero, L. (s.f.). *SCRIBD*. Obtenido de <http://es.scribd.com/doc/44252680/LCD-16X2#download>

S, P. A. (s.f.). *olimex*. Obtenido de <http://www.olimex.cl/tutorial/tutorial1.pdf>

Saldarriaga, V. (2 de enero de 2012). *De Peces*. Obtenido de

<http://www.depeces.com/tipos-de-calentadores-para-acuarios.html>

SEMICONDUCTOR, D. (s.f.). Obtenido de

<https://www.sparkfun.com/datasheets/Components/DS1307.pdf>

TORRITI, M. T. (s.f.). Obtenido de

[http://web.ing.puc.cl/~mtorrest/downloads/pic/tutorial\\_pic.pdf](http://web.ing.puc.cl/~mtorrest/downloads/pic/tutorial_pic.pdf)