



Universidad Austral de Chile

Facultad de Ciencias de la Ingeniería
Escuela de Ingeniería Civil Electrónica

DISEÑO Y CONSTRUCCIÓN DE UN EQUIPO SIMULADOR DE ESTUARIO

Trabajo para optar al título de:
Ingeniero Electrónico.

Profesor Patrocinante:
Sr. Pedro Ry Clericus.
Ingeniero Electrónico
Licenciado en Ciencias de la Ingeniería
Diplomado en Ciencias de la Ingeniería

FELIPE ANDRÉS VARGAS MILLALONCO
VALDIVIA – CHILE
2009

Comisión Revisora

Pedro J. Rey Clericus

Franklin M. Castro Rojas

Luis M. Pardo Suazo



Universidad Austral de Chile

Facultad de Ciencias de la Ingeniería
Escuela de Ingeniería Civil Electrónica

DISEÑO Y CONSTRUCCIÓN DE UN EQUIPO SIMULADOR DE ESTUARIO

Trabajo para optar al título de:
Ingeniero Electrónico

Profesor Patrocinante:
Sr. Pedro Ry Clericus.
Ingeniero Electrónico
Licenciado en Ciencias de la Ingeniería
Diplomado en Ciencias de la Ingeniería

FELIPE ANDRÉS VARGAS MILLALONCO
VALDIVIA – CHILE
2009

Agradecimientos

Agradezco a mis padres por el apoyo incondicional durante toda mi vida.

A mis profesores por su paciencia, la formación profesional y los conocimientos que me han entregado durante estos años, y los que aun sigo recibiendo.

A mis amigos que me acompañaron a estudiar, trabajar, carretear y expresar nuestras opiniones frente al público.

Para la tía Edith y la tía Clarita, por haberme acogido en su casa, apoyarme con trabajo y sobre todo su amistad.

A todas aquellas personas que me brindaron apoyo intelectual, facilitaron información y materiales de todo tipo para la elaboración de mi trabajo de titulación.

Índice

Portada	i
Comisión Revisora	ii
Dedicatoria	iii
Agradecimientos	iv
Índice	v
Resumen	viii
Abstract	ix
Objetivo	x
Introducción	x
Capítulo I: Definición y características de un estuario.	1
1.1 Definición de Estuario	1
1.2 Clasificación de los Estuarios	1
1.3 Breve antecedentes estuarinos en el sur de Chile	3
Capítulo II: Propiedades y características físicas.	6
2.1 Medición de salinidad	6
2.1.1 Conductividad eléctrica/Salinidad	6
2.1.2 Sensores de conductividad	6
2.1.3 Tipos de sensores de conductividad	7
2.2 Medición de Temperatura	12
2.2.1 Sensores de temperatura	14
Capítulo III: Microcontrolador PIC16F877A	21
3.1 Microcontrolador 16F877A	21
3.1.1 Arquitectura De VON NEUMANN	21
3.1.2 Arquitectura Harvard	22
3.2 Puertos	23
3.2.1 Puerto A (PORTA)	23
3.2.2 El Registro ADCON1 (9FH)	24

3.2.3 Puerto B (PORTB).....	25
3.3 Conversor analógico digital (ADC).....	27
3.3.1 Técnicas de cuantificación.....	30
3.3.2 Módulo ADC del PIC 16F877A.....	31
3.3.3 Configuración del conversor A/D.....	34
3.3.4 Registros De Resultado A/D.....	36
3.4 Modulo de Entrada y Salida USART	37
3.4.1 Protocolo De Comunicación Serie	39
3.5 La USART del PIC 16f877A	42
3.5.1 Modo Asíncrono	43
3.5.2 El Circuito De Muestreo.....	43
3.5.3 El Generador De Baud Rate (BRG)	44
3.5.4 El Transmisor Asíncrono.....	45
3.5.5 El Receptor Asíncrono.....	47
Capitulo IV: Software Utilizados	50
4.1 Microchip Mplab IDE	50
4.1.1 Ficheros Resultantes Del Ensamblado	51
4.2 Compilador PCW de CCS	52
4.3 ICprog.....	54
4.4 Proteus	55
4.4.1 ISIS	55
4.4.2 ARES.....	56
4.4.3 VSM.....	57
4.4.4 ProSPICE.....	58
4.5 PIC Simulator IDE	59
4.6 HyperTerminal.....	60
Capitulo V: Protocolos De Comunicación Analizados Y Aplicados	61
5.1 TIA/EIA RS-232 Recommended Standard 232C.....	61
5.1.1 Descripción de las señales	62

5.2 Protocolo One Wire	63
5.2.1 Señales De Un Bus De Una Línea.....	63
5.2.2 Inicialización: Pulsos De Reset Y Presence	64
5.3 Protocolo I2C (Inter Integred Circuit Bus).....	65
5.3.1 Condiciones de START y STOP	66
Capitulo VI: Diseño, Construcción Y Pruebas Del Prototipo	67
6.1 Etapa de entrada.....	68
6.1.1 Sensor de temperatura DS18B20.....	69
6.1.2 Sensor de Salinidad	70
6.1.3 Reloj de Tiempo Real DS1307	71
6.2 Etapa de proceso.....	72
6.2.1 Software.....	73
6.3 Etapa de Salida	80
6.4 Proyección y mejoramiento	84
Conclusiones.....	85
Bibliografía.....	86
Referencia Electrónica	87

Resumen

Hoy en día, la electrónica ha sido y tal vez siga siendo de gran ayuda para muchas personas, aportando a facilitar el desarrollo tecnológico, a mejorar las condiciones de trabajo en muchas áreas de las ciencias como así también en los ámbitos industriales. Es una realidad poder realizar equipos que ayudan a entender y observar el medio ambiente, y es aquí donde la electrónica, como una técnica, a otorgado mucho.

La tesis se fundamenta en el diseño y construcción de un equipo simulador de estuario, el cual será de gran ayuda tanto para estudiantes como profesores de la carrera de Biología Marina interesados en investigar en este ecosistema. En particular se plantea analizar los efectos que se producen en las especies animales que habitan este medio ambiente, principalmente jaibas.

Para esto se planifica la creación de un equipo de forma tal que se pueda variar salinidad y temperatura del agua, a través de un control de flujo de entrada y salida de agua de mar y dulce. Las variables manejadas son, según la literatura (Kinne 1966), las con mayor influencia en los organismos acuáticos que coexisten en este medio.

Los estuarios son cuerpos de agua costeros semi-cerrados, los cuales tienen una conexión libre con el océano y dentro del cual el agua de mar es considerablemente diluida con agua dulce proveniente del drenaje terrestre (Dyer 1998). La dinámica estuarina está dada principalmente por cambios de la marea, la pluviosidad y la descarga de los tributarios.

Abstract

Nowadays, electronics has been a great and helpful tool for many people, contributing to facilitating technological development, improving working conditions in many areas of science as well as in industrial areas. Now we can create equipments that may and will help to watch and understand the environment, and all this is possible thanks to electronic.

The thesis is based on the design and construction of a computer simulator estuary, which will help students and teachers of the marine biology career interested in environment and ecosystem research. Mostly, it's to analyze the effects that occur in animal species that inhabit this environment, especially the crabs.

For this purposes, the creation of equipment that can change the salinity and water temperature is a great tool. This will be possible through a control flow of entry and exit of seawater and freshwater. The variables are, according to the literature (Kinne 1966), the ones that have the most influence on aquatic organisms that coexist in this environment.

Estuarine particles are semi-enclosed coastal water, which have a free connection with the ocean and within which seawater is significantly diluted with freshwater derived from land drainage (Dyer 1998). Estuarine dynamics happens mostly by changes in the tide, rainfall and discharge of the tributaries.

Objetivo

El objetivo principal es diseñar y construir un equipo capaz de simular la dinámica de la columna de agua de un estuario, en donde se puedan manejar las principales variables (salinidad y temperatura) que influyen en este ecosistema; el equipo a construir debe ser de fácil manipulación, para facilitar la obtención de datos continuos; las variables a utilizar serán dispuestas en un rango que refleje las variaciones ambientales naturales.

Introducción

El trabajo que a continuación se presenta es el diseño e implementación de un equipo capaz de simular las condiciones del estuario de valdivia, para satisfacer una necesidad en la investigación de la reproducción de jaibas. La realización de este equipo estará basada en un sistema electrónico para así lograr hacer la medición de temperatura y el censado de la salinidad, siendo estos dos parámetros los más importantes.

El sistema electrónico estará basado en el estudio realizado de los niveles de temperatura y salinidad que se presentan en el estuario de valdivia. para la medición de temperatura se utilizara el sensor DS18B20, para la medición de la salinidad se implementara una sonda con características especiales con los que se obtendrán resultados inmediatos, para la adquisición de datos y el manejo de estas se utiliza el microcontrolador 16F877A que finalmente esta conectado a computador para la correcta visualización y almacenamiento de los datos.

Capítulo I: Definición y características de un estuario.

1.1 Definición de Estuario

“An estuary is a semi-closed coastal body of water which has a free connection with the open sea and within which sea water is measurably diluted with fresh water derived from land drainage”

PRITCHAR, Donald W. *Estuaries*. Washington DC. 1967, "What is estuary: Physical View Point" p3.

“Un estuario es un cuerpo de agua costero semi-cerrado, el cual tiene una conexión libre con el océano y dentro del cual el agua salada es considerablemente diluida con agua dulce proveniente del drenaje terrestre”

Los estuarios son de gran importancia tanto para la vida y desarrollo de especies de origen marino, tales como peces y crustáceos. Así también para el desarrollo de la actividad pesquera, polo productivo y económico de nuestro país.

1.2 Clasificación de los Estuarios

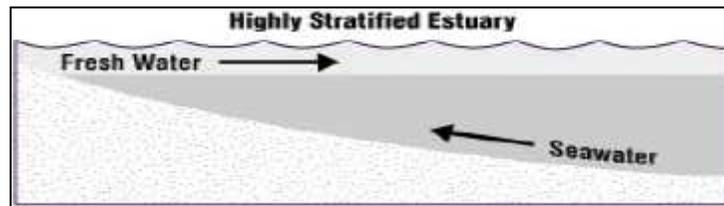
La mezcla de las aguas dulces del río y del agua marina provoca la formación de gradientes horizontal y vertical de salinidad, cuyas características dependen del caudal del río y de sus variaciones estacionales, de la amplitud de las mareas, de la energía de las olas y de la morfología del estuario.

Tanto en el mundo, como en la costa de Chile hay una gran cantidad de estuarios por lo cual los podemos clasificar de la siguiente manera:

- 1.- Desde el punto de vista geológico
 - a) Valles inundados por ríos.
 - b) Tipo Fiordos
 - c) Estuarios que construyen barras
 - d) Estuarios formados por procesos tectónicos

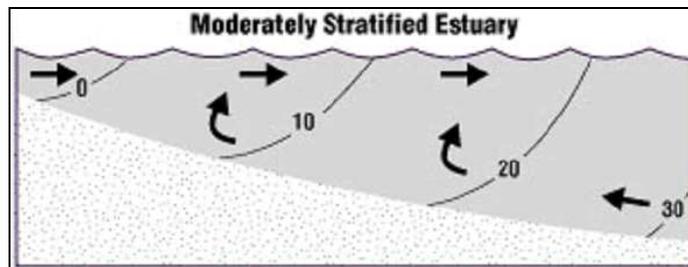
2.- Desde el punto de vista de la circulación

a) Estuario de cuña de sal: Tipo A



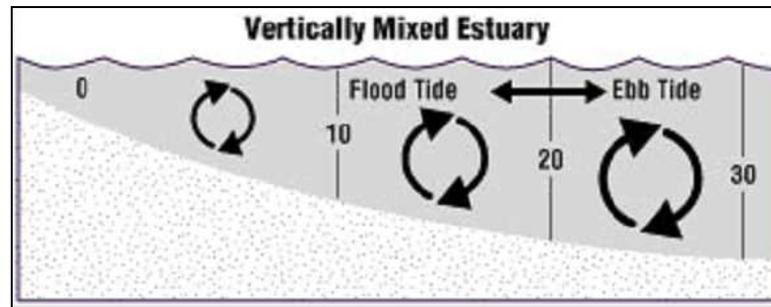
Estos se desarrollan cuando un río desemboca en un mar sin mareas (micromareal). Aquí como se observa en la figura el agua dulce que es menos densa circula por encima del agua salada que es más densa.

b) Estuario parcialmente inundado: Tipo B



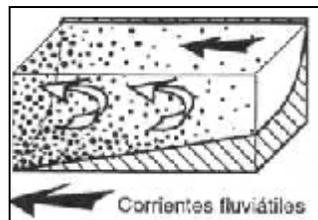
Estos ocurren donde un río desemboca en un mar con rango mareal moderado (mesomareal). Aquí se producen fenómenos de mezcla debidos a la fricción entre las aguas, las corrientes mareales, y la fricción con el fondo.

c) Estuario verticalmente homogéneo: Tipo C



Estos estuarios son anchos, con rango mareal alto donde las corrientes mareales son más fuertes comparadas con el caudal del río. Su columna de agua se encuentra totalmente mezclada, no habiendo variaciones en la vertical de salinidad.

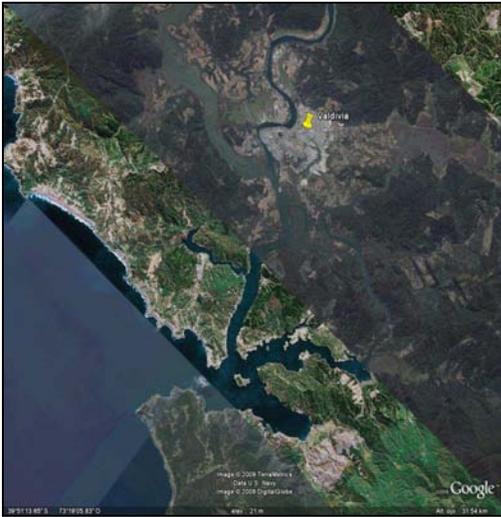
d) Estuario lateralmente homogéneo: Tipo D



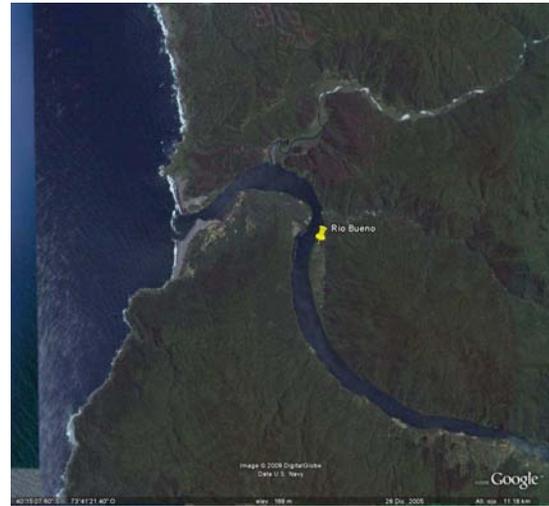
En este tipo de estuario la mezcla entre aguas saladas y dulces es casi total, y solo permanece un débil gradiente horizontal de salinidad

1.3 Breve antecedentes estuarinos en el sur de Chile

En las imágenes 1 y 2 se muestran dos formaciones estuarinas, ambas pertenecientes a la región de los Ríos, ambos están dentro de los que se consideran estuarios vivos.



Img.1 Río Valdivia



Img.2 Río Bueno

(Imágenes de Google Earth)

La costa centro sur de Chile, se caracteriza por la presencia de numerosos estuarios micromareales. Estos se conectan a hoyas hidrográficas originadas en el sistema andino o preandino (Cordillera de la costa) y presentan regímenes hídricos mixtos o exclusivamente fluviales.

El complejo estuarial Valdivia, Tornagaleones incluye los ríos Valdivia, Tornagaleones, Ensenada San Juan y bahía de Corral, perteneciente a una hoya hidrográfica de tipo andino. El río Valdivia nace en el lago Riñihue (con el nombre de río San Pedro) y desemboca en la bahía de corral luego de un largo recorrido. La zona estuarina propiamente tal tiene una extensión de cerca de 15 Km., pero la influencia de las mareas puede alcanzar hasta 42 Km. río arriba. El ancho medio del estuario es de aproximadamente 700 m con un mínimo de 450 m en la Isla San Francisco y un máximo de 1000 m cerca de la desembocadura del río Cutipay. La boca tiene un ancho aproximado de 600 m. Las profundidades alcanzan entre 3 y 5 m, aunque se han registrado profundidades de hasta 10 m en el canal de navegación.

El caudal del río en el estuario alcanza a 1000 m³/seg. en un estuario abierto y no forma barra en su desembocadura. El estuario del río Tornagaleones presenta profundidades de 3 a 5 m, aunque se han registrado profundidades de 11 m. En la ensenada de san Juan, la profundidad aumenta gradualmente desde el sur hacia la Bahía de Corral, con máximos de 9 m en las cercanías de la isla Mancera. Además de los ríos Valdivia y Tornagaleones, en este complejo

estuarial desembocan otros ríos de menor orden originados en cuencas costeras, entre los que se encuentran los ríos San Juan, de los Llanos y Catrilelfu, que también presentan régimen microestuarial. Además, es destacable una serie de pequeños riachuelos repartidos por todo este sistema estuarial (Arcos , 2002).

ARCOS, Dagoberto. Proyecto FIP 2000-09. *Determinación de la capacidad de carga de las zonas estuarinas de los Ríos Valdivia y Bueno X Región (Río Valdivia)* Talcahuano, Chile. INPESCA, (Instituto de investigación pesquera). 2002

Capítulo II: Propiedades y características físicas

2.1 Medición de salinidad

2.1.1 Conductividad eléctrica/Salinidad

Para el entendimiento general, se define conductividad como “la habilidad o poder de conducir o transmitir calor, temperatura o electricidad”, siendo su unidad de medida siemens/m (S/m) en el sistema de medición SI y micromhos por cm en unidades estándar de EE.UU.

La conductividad eléctrica, se produce por el movimiento de partículas cargadas eléctricamente y como respuesta a las fuerzas que actúan en estas partículas debido a un campo eléctrico aplicado.

Es así como podemos explicar sobre la conductividad del agua, como ya sabemos, es un buen conductor de la electricidad, debido a que se las sales se disuelven en el agua y se descomponen en iones cargados positivamente y negativamente, es así, que los iones más positivos encontramos son Sodio (Na^+), Calcio (Ca^{+2}), Potasio (K^+) y Magnesio (Mg^{+2}). Los iones más negativos encontramos al Cloruro (Cl^-), sulfato (So_4^{-2}), Carbonato y Bicarbonato. Los nitratos y fosfatos no contribuyen de forma apreciable a la conductividad aunque son muy importantes biológicamente.

La salinidad es una medida de la cantidad de sales disueltas en agua. La salinidad y la conductividad están relacionadas porque la cantidad de iones disueltos aumentan los valores de ambas.

Para realizar esta medición se escogerá un sensor adecuado a las exigencias que se tienen dentro del estuario de Valdivia.

2.1.2 Sensores de conductividad

En el mercado hoy en día podemos encontrar distintos modelos de sensores de conductividad, diseñados para medir en forma muy precisa la salinidad a través de los iones disueltos en el agua o en distintos fluidos, es así que podemos encontrar desde una gama para trabajos en laboratorios hasta una gama destinada a los procesos productivos de una industria.

Las diferencias que podemos encontrar en estos dispositivos de detallan a continuación, aquí podremos ver los rangos a los cuales trabajan como también los tamaños que tienen y para que medios fueron diseñados, estos datos serán de gran importancia al momento de escoger nuestro dispositivo que medirá la cantidad de sal que se encuentre disuelta en la pileta que simulara nuestro estuario.

2.1.3 Tipos de sensores de conductividad

Sensor de conductividad DT035A

El DT035A esta diseñado para medir conductividad en líquidos y soluciones. Este dispositivo detecta la conductividad entre el rango de 0 – 20 mS (milisiemens). El DT035 consiste en un electrodo de conductividad, un adaptador en forma de huevo y un cable de conexión.



Como trabaja

El electrodo utiliza dos anillos y cada uno es de acero inoxidable de alto grado. El tamaño del anillo y la distancia entre ellos son fijos. Cuando una corriente eléctrica viaja desde el anillo, al electrodo ubicado en el líquido, con algo de conductancia, el segundo anillo recibe algo de esta corriente y la solución que esta entre el anillo actúa como resistencia. La caída de tensión en esta resistencia es medida y ajustada a un rango de 0-5 V para que sea aceptado por el convertidor análogo/digital de el Data-logger.

Área de Utilización

El DT035A es usado para varios experimentos en Biología y Química, el DT035A se pueden demostrar cambios en la conductividad al disolver sales en agua, o monitorear cuerpos de agua para contaminación, también se puede utilizar para el examen de salinidad en el agua.

Datos Técnicas

- Rango: 0 a 20 mS.
- 12-bit de Resolución (TriLog): 5 μ S
- 10-bit de Resolución (MultiLogPRO, MultiLog): 20 μ S
- Total de error: menos que 1.2% sobre todo el rango.
- Resistencia de entrada típica: <100 Ω .
- Equipado con un tornillo de calibración del desbalance.

Conductímetro PCE-CM41

El conductímetro PCE-CM 41 le hace posible la medición del valor de conductividad, la suma de toda la materia sólida o sales disueltas (TDS - Total Dissolved Solids) y al temperatura. Debido a su tamaño reducido puede transportar este conductímetro en cualquier sitio. La carcasa resistente al agua protege la electrónica de forma óptima (también protege contra el polvo). La medición de conductividad EC y TDS tiene la compensación de temperatura automática. El ámbito de uso del conductímetro es muy amplio, pasando por la industria, la piscicultura, horticultura y otros ámbitos. En el siguiente enlace puede encontrar el conductímetro que se ajuste a sus necesidades.



Datos Técnicos

Conductividad

Selección de rango	Rango de medición	Resolución	Precisión
2000uS	2-2000uS	1uS	±3% del rango total
20ms	2-20ms	0,01ms	

TDS

Selección de rango	Rango de medición	Resolución	Precisión
2000ppm	132-1320ppm	1ppm	±3% del rango total
20000ppm	1320-13200ppm	10ppm	

Temperatura

Unidad	Rango de medición	Resolución	Precisión
°C	0-60 °C	0,1 °C	± 0,8 °C

Tabla 1.-

Área de Utilización

- Acondicionamiento del agua
- Piscicultura
- Acuarios
- Piscinas
- Control de calidad
- Agricultura y horticultura
- Cultivos hidropónicos
- Torres de refrigeración
- Calderas
- Aguas residuales

Conductímetro EC-215

El conductímetro 215 está pensado principalmente para su uso en laboratorios. Cuatro campos de medición hacen de este conductímetro de mesa un medidor de flexible aplicación (en industria, agricultura y también en el ámbito comunal). La sonda anular que trabaja potenciométricamente se puede emplear en todos los campos de medición. El aparato dispone de compensación automática de temperatura y regulación variable del coeficiente (entre 0 y 2,5%).



Datos técnicos

Campos de medición	0,0 ...199,9 $\mu\text{S}/\text{cm}$; 0,0 ... 1999 $\mu\text{S}/\text{cm}$ 0,00 ... 19,99 mS/cm ; 0,0 ... 199,9 mS/cm
Resolución	0,1 $\mu\text{S}/\text{cm}$; 1 $\mu\text{S}/\text{cm}$; 0,01 mS/cm ; 0,1 mS/cm
Precisión (a 20°C)	+/- 1 % del campo de medición
Calibración	manual, 1 punto por medio del teclado
Compensación de temperatura	automática de 0...50 °C
Electrodo	Sonda de 4 anillos con sensor térmico y cable de 1m
Alimentación	Adaptador 12 V

Tabla 2.-

oxígeno (opcional)	100 ppm
temperatura (opcional)	0,1 mg/l
	0,1 °C
Precisión	
valor del pH	$\pm 0,02$ pH + 2 dígitos
conductividad	± 2 % del rango de medición + 1 dígito
TDS	± 2 % del rango de medición + 1 dígito
oxígeno (opcional)	$\pm 0,4$ mg/l
temperatura (opcional)	$\pm 0,8$ °C
Calibración	
valor del pH	3 puntos de calibración (pH4, pH7 y pH10)
conductividad	con 1413 μ S/cm
oxígeno (opcional)	en el aire
Compensación de temperatura	automática con sensor de temperatura adicional (0 ... a +65 °C) o manual de 0 ... a +100 °C
Alimentación	4 pilas x 1,5 V AA (o con un adaptador

Tabla 3.-

2.2 Medición de Temperatura

El concepto de temperatura esta muy relacionado con el diario vivir. Tenemos un concepto intuitivo de algo más caliente o mas frío. Este concepto es solo cualitativo y aplicable solo en un espacio limitado.

Se hace necesario establecer una escala que permita clasificar las temperaturas por orden creciente. Para ello basta un fenómeno físico que sea una función constantemente creciente o decreciente en un rango de temperaturas utilizables.

Entre los fenómenos físicos mas conocidos que se han empleado para clasificar temperaturas y hacer termómetros podemos citar:

- **Dilatación y contracción:** de sólidos, líquidos o gases. Con ello se han construido desde los clásicos termómetros de columna líquida, hasta los termómetros bimetalicos.
- **Variación de resistencia eléctrica:** la variación de resistencia eléctrica con la temperatura se usa en termómetros en caso a termistores y termómetros de resistencia eléctrica (resistencia de platino, PT100).
- **Potencial termoeléctrico:** si la unión de dos metales diferentes se somete a un gradiente de temperatura, se genera una fuerza electromotriz (fem). Este es el llamado efecto Seebeck es la base en que se sustentan las termocupla.
- **Radiación Electromagnética:** tanto los pirómetros infrarrojos como los pirómetros ópticos se basan en los fenómenos de la radiación para medir temperatura. Ambos tienen la ventaja de que pueden medir a distancia. los primeros se utilizan para las temperaturas muy bajas y los segundos para altas temperaturas (hornos, metales en fusión).

Puntos termométricos

En la práctica se necesita una serie de puntos de referencias fácilmente replicables para poder calibrar diferentes tipos de termómetros en diferentes rangos de temperatura. Algunos puntos de referencia importantes son:

Producto Puro	Temperatura a 1 atm.	Producto Puro	Temperatura a 1 atm.
Solidificación agua	0°C	Solidificación aleación Ag y Cu (72% Ag, 28% Cu)	779°C
Ebullición Agua	100°C	Ebullición Zinc	907°C
Ebullición Clorobenceno	132°C	Fusión Plata	960°C
Ebullición Naftalina	217,96°C	Fusión Cobre	1083°C
Fusión de Estaño	231,9°C	Fusión Níquel	1455°C
Fusión del Plomo	327,3°C	Fusión Hierro	1530°C
Ebullición del Mercurio	356,95°C	Fusión Platino	1773°C
Fusión del Zinc	419,4°C	Fusión Aluminio	658°C
Ebullición del Azufre	444,6°C		
Solidificación Antimonio	630,5°C		

Tabla 4.-

2.2.1 Sensores de temperatura

DS18B20

El DS18B20 es un sensor de temperatura que transmite su valor digital de la temperatura leída en el lugar donde se encuentre el dispositivo. Utiliza una única línea serie que se comunica con otros dispositivos mediante el protocolo para bus de una línea diseñado por Dallas Semiconductors. Sus principales características son:

- Lo podemos encontrar en diferentes encapsulados TO-92, SO y μ SOP.
- La temperatura es leída como un valor digital de 2 bytes que incluye el signo.
- Medida de temperatura desde $-55\text{ }^{\circ}\text{C}$ hasta $+125\text{ }^{\circ}\text{C}$.
- Requiere una única línea de comunicación mediante el 1-wire Bus.
- Apenas Requiere componentes externos
- Se puede seleccionar la resolución de temperatura entre 9 a 12 Bits.
- Ha sido incluido en el control de temperatura, sistemas industriales, productos de consumo, termómetros y sistemas de medición sensibles a calentamiento.
- Para su funcionamiento se puede utilizar una alimentación desde los 3.3 V a 5 V.
- Se puede alimentar en forma parasita.
- Cuenta con un código único serial de 64 bit, el cual es el nombre identificativo de este dispositivo.

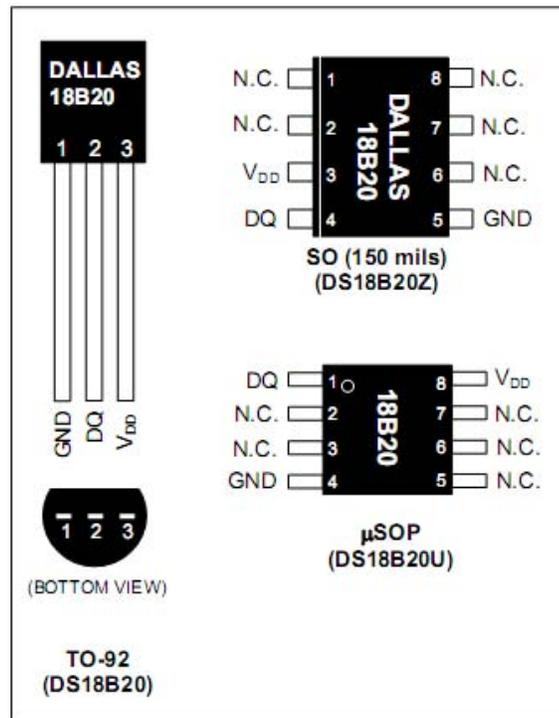


Fig.1.- Pin out DS18B20 y encapsulados.

Diagrama en bloques

En la figura 2 podemos ver el diagrama en bloques de este sensor, se aprecia que es bastante sencillo ya que facilita el entendimiento lógico al momento de programar.

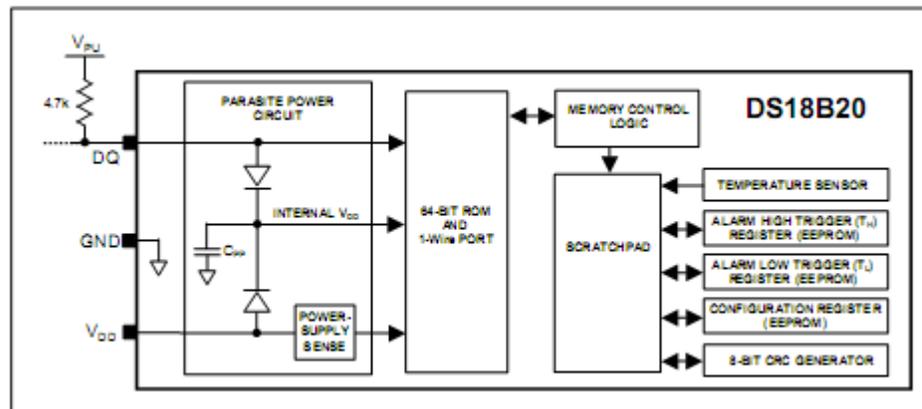


Fig.2.- Diagrama en bloques del DS18B20

- *64-Bit ROM*. Zona de memoria donde se graba un código identificativo de 64 bits y que es único para cada dispositivo. Cada dispositivo tiene un código único asociado grabado en ROM que sirve para identificarlo en el caso de que varios de ellos se encuentren conectados a la misma línea de datos. Esta especial característica Hace que no sea necesario conectar dispositivos adicionales ni aumentar el cableado para la selección de uno de estos dispositivos entre varios.
- *Parasite Power Circuit*. Una característica importante de estos dispositivos es la posibilidad que tienen de alimentarse sin necesidad de una fuente de alimentación externa. Para ello, aprovecha la potencia de la línea DQ Cuando se encuentra a nivel alto. Esta energía es almacenada en el Condensador Cpp, que actúa como alimentación del chip cuando la línea DQ se encuentra en nivel bajo. Cuando el DS18B20 utiliza el modo parasito el pin 3 (Vdd) debe conectarse a masa.
- *Scrachpad Memory*. Esta memoria auxiliar contiene principalmente los dos bytes con la temperatura medida (Temperature Sensor).

Lectura de temperatura

Para realizar la lectura de temperatura según el formato que se muestra en la tabla 5, aquí podemos observar claramente que:

- El formato es de dos bytes.
- En e primer Byte se encuentra incluido el signo, "0" para valores positivos y "1" para valores negativos.
- En los últimos bits del segundo byte se encuentra la parte decimal de la lectura.
- Las temperaturas vienen en complemento a 2. Hay que negar las temperaturas negativas para obtener su valor absoluto.

TEMPERATURE (°C)	DIGITAL OUTPUT (BINARY)	DIGITAL OUTPUT (HEX)
+125	0000 0111 1101 0000	07D0h
+85*	0000 0101 0101 0000	0550h
+25.0625	0000 0001 1001 0001	0191h
+10.125	0000 0000 1010 0010	00A2h
+0.5	0000 0000 0000 1000	0008h
0	0000 0000 0000 0000	0000h
-0.5	1111 1111 1111 1000	FFF8h
-10.125	1111 1111 0101 1110	FF5Eh
-25.0625	1111 1110 0110 1111	FE6Fh
-55	1111 1100 1001 0000	FC90h

Tabla.5.- Relación entre Temperatura /Dato

Como ya hemos mencionado el formato de temperatura viene dada en dos bytes las cuales se detallan a continuación.

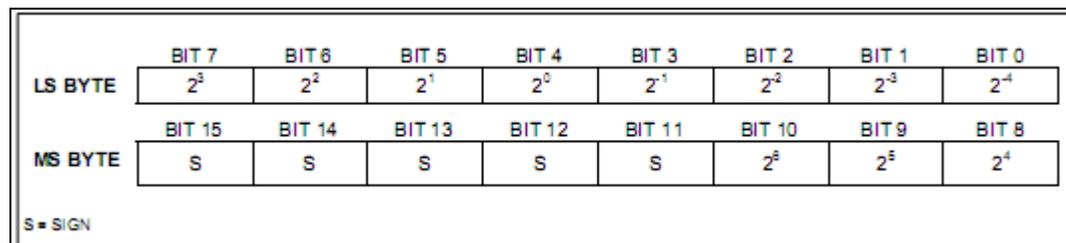


Fig.3.- Formato de los registro de temperatura

Como podemos ver en la figura 3 LS BYTE corresponde al byte menos significativo, es decir, al segundo byte de lectura, aquí encontramos que desde el BIT-0 al BIT-3 se encuentra nuestro valor decimal, desde el BIT-4 al BIT-7 encontramos la primera parte del valor absoluto de temperatura. MS BYTE corresponde al byte más significativo, es decir, es el primer byte de lectura, aquí encontramos que desde el BIT-8 al BIT-12 la segunda parte del valor absoluto de temperatura, y en este caso desde el BIT-11 al BIT-15 vemos que esta indicado por la letra "S", esto significa que corresponde al signo de la temperatura leída ("0" el valor es positivo y "1" el valor es negativo).

Organización de la memoria

La memoria del DS18B20 esta organizada de la manera en que se presenta en la figura 4.

Uno de los registros importantes de la memoria es el Byte 4 ya que aquí encontramos el registro de configuración, aquí el usuario puede configurar la resolución de conversión de la temperatura por medio de R0 y R1, cuando se enciende el sensor se encuentra por default, es decir, R0=1 y R1=1, lo cual indica que esta en una resolución de 12 BIT (observar figura 5).

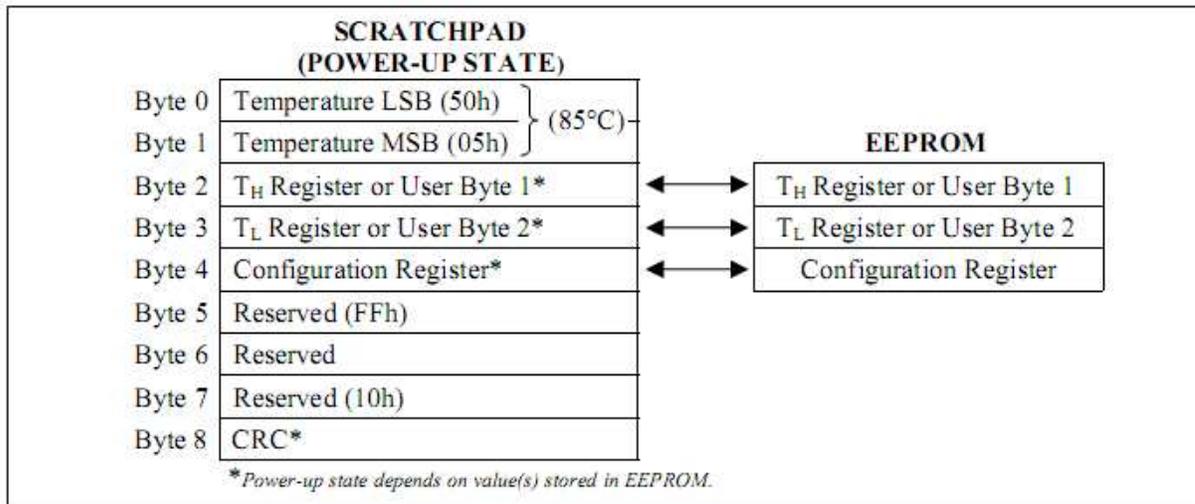


Fig.4.- Mapa de memoria

En el registro de configuración los BIT0 al BIT4 y BIT7 están reservados para el uso interno de este dispositivo.

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
0	R1	R0	1	1	1	1	1

Fig.5.- BYTE 4, Registro de configuración.

En la siguiente tabla 2 observamos el tiempo de conversión dependiendo de la resolución configurada en R0 y R1.

R1	R0	RESOLUTION (BITS)	MAX CONVERSION TIME	
0	0	9	93.75ms	($t_{CONV}/8$)
0	1	10	187.5ms	($t_{CONV}/4$)
1	0	11	375ms	($t_{CONV}/2$)
1	1	12	750ms	(t_{CONV})

Tabla.6.- Configuración de resolución de temperatura

LM35

El sensor LM35 del fabricante Nacional Semiconductor, este es uno de los sensores mas utilizados en el mundo, ya que por su facilidad para configurar y conversión de temperatura, lo hace uno de los dispositivos favoritos para muchos diseñadores de circuitos y aficionados a la electrónica.

El sensor LM35 es un circuito integrado de precisión de temperatura, su salida entrega un voltaje lineal proporcional a la temperatura en grados celcius. Este sensor tiene una precisión de 1°C y un rango que abarca desde -55°C a +155°C.

Este se presenta en diferentes encapsulados pero el mas común es el TO92 (ver figura 6), que es igual a la forma típica de un transistor con tres patillas, dos de ellas son para la alimentación y la tercera nos entrega el valor de tensión, que corresponderá a la temperatura.

Como ya lo habíamos dicho la salida es lineal y equivale a 10mV/°C por lo tanto:

- +1500 mV = 150°C
- +250 mV = 150°C
- -550 mV = 150°C

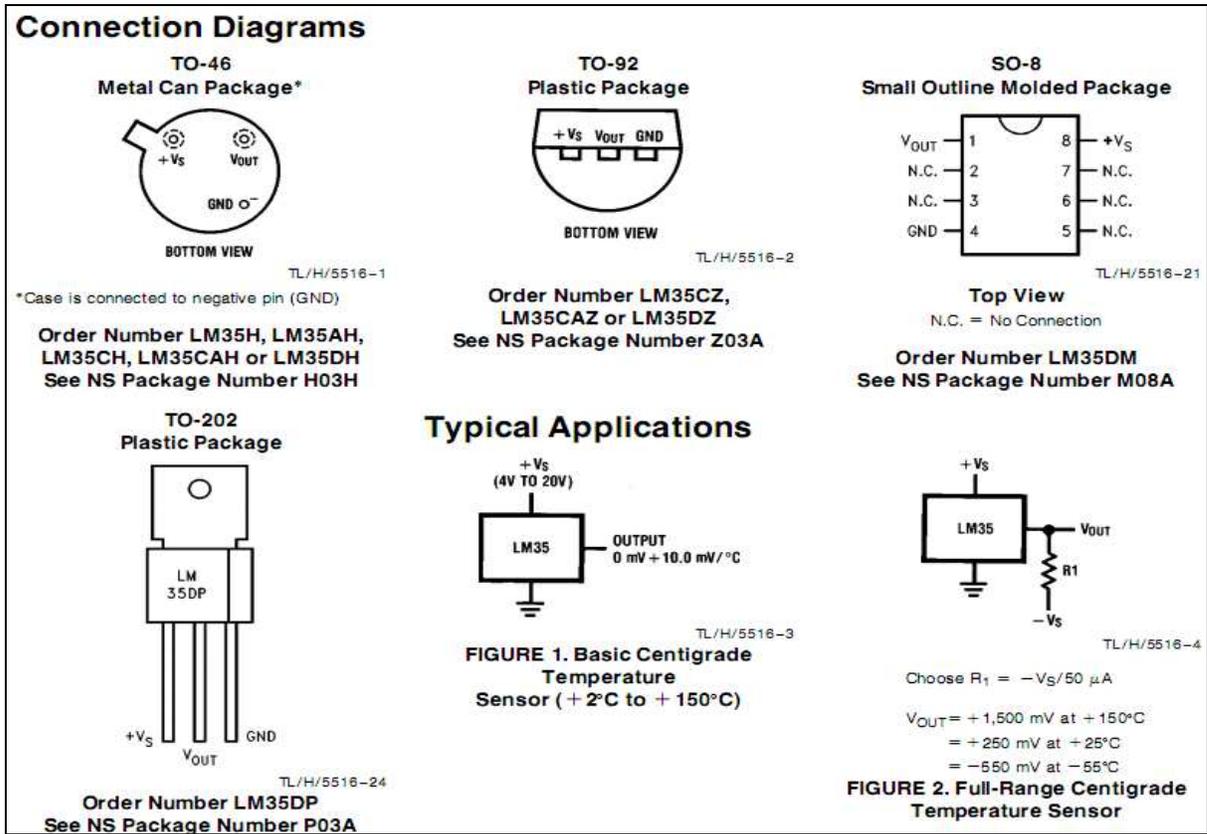


Fig.6.- Tipos de encapsulados LM35

Capítulo III: Microcontrolador PIC16F877A

3.1 Microcontrolador 16F877A

Un microcontrolador es un sistema cerrado, lo cual quiere decir que en un solo circuito integrado se encierra un sistema digital programable completo. Este dispositivo se destina a gobernar una sola tarea que no se puede modificar. Los microcontroladores disponen de los bloques esenciales: CPU, memorias de datos y programa, reloj, periféricos de entrada/salidas, etc.

3.1.1 Arquitectura De VON NEUMANN

La arquitectura tradicional de sistemas digitales programables de basa en el esquema propuesto por John Von Neumann. En este modelo la unidad central de proceso o CPU se conecta en una memoria única que contiene las instrucciones del programa y los datos.

El tamaño de la unidad de datos o instrucciones esta fijado por el ancho del bus de datos de la memoria exterior utilizada, que es de 8 bits. Un microprocesador con un bus de 8 bits que lo conecta con la memoria deberá manejar datos e instrucciones de una o más unidades de 8 bits de longitud. Cuando deba acceder a una instrucción o dato de más de un byte de longitud, deberá realizar más de un acceso a la memoria. Por otro lado este bus único limita la velocidad de operación del microprocesador, ya que no se puede buscar en la memoria una nueva instrucción antes de que finalicen las transferencias de datos que pudieran resultar de la instrucción anterior.

Resumiendo, las dos principales limitaciones de la arquitectura tradicional o de Von Neumann son:

- La longitud de las instrucciones esta limitada por la unidad de longitud de los datos, por lo tanto el microprocesador debe hacer varios accesos a memoria para buscar instrucciones completas.
- La velocidad de operación esta limitada por el efecto de cuello de botella que significa un único bus para datos e instrucciones, que impide superponer ambos tiempos de acceso.

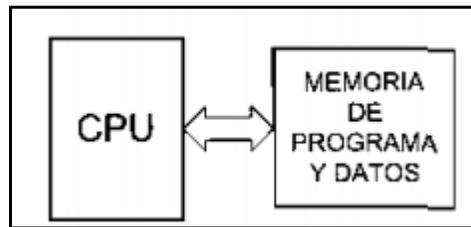


Fig.7.- Arquitectura Von Neumann

3.1.2 Arquitectura Harvard

Tradicionalmente los sistemas digitales programables se basaban en la arquitectura Von Neumann, caracterizada por disponer de una única memoria en la que se almacenan tanto los datos como las instrucciones. A esta memoria se accede a través de un sistema de buses únicos. La única ventaja que posee es que simplifica la lógica del microcontrolador.

Los microcontroladores PIC utilizan una arquitectura Harvard que dispone de dos memorias independientes a las que se conecta mediante dos grupos de buses separados

- Memoria de datos
- Memoria de programa

Ambos buses son totalmente independientes y pueden ser de distintos anchos, esto permite que la CPU pueda acceder de forma independiente y simultánea a la memoria de datos a la de instrucciones, consiguiendo que las instrucciones se ejecuten en menos ciclos de reloj.

Esta dualidad de la memoria de datos por un lado y por otro la memoria de programa, permite la adecuación del tamaño de las palabras y de los buses a los requerimientos específicos de las instrucciones y los datos.

Se puede concluir que las principales ventajas de la arquitectura Harvard son:

- El tamaño de las instrucciones no está relacionado con el de los datos y, por lo tanto, puede ser optimizado para que cualquier instrucción ocupe una sola posición de memoria programa. Así se logra una mayor velocidad y una menor longitud de programa.

- El tiempo de acceso a las instrucciones puede superponerse con el de los datos, logrando una mayor velocidad de operación.

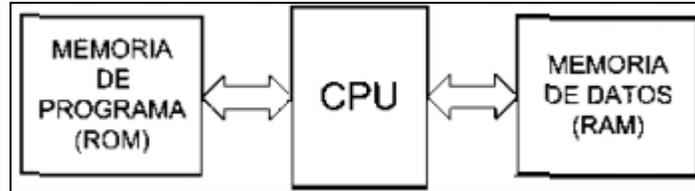


Fig.8.- Arquitectura Harvard

PALACIOS, Enrique. REMIRO, Fernando. LOPES, Lucas J. *Microcontrolador PIC16F84, Desarrollo de proyectos*, México DF. Alfaomega, 2004. ISBN9070-15-1033-X

3.2 Puertos

3.2.1 Puerto A (PORTA)

El puerto A posee 6 líneas bidireccionales. Los 3 registros asociados a este puerto son:

Registro PORTA (05H)

Registro de estado del Puerto A. Cada uno de los 6 bits menos significativos (RA5,..., RA0) de este registro están asociados a la línea física correspondiente del puerto. Al hacer una lectura este registro se lee el estado de todas los pines del puerto. Todas las escrituras al registro son operaciones del tipo “lee-modifica-escribe”, es decir, toda escritura al puerto implica que el estado de los pines es leído, luego es modificado y posteriormente se escribe al latch de datos del puerto.

05h	-	-	RA5	RA4	RA3	RA2	RA1	RA0
-----	---	---	-----	-----	-----	-----	-----	-----

Tabla.7.- Registro PORTA (05h)

Registro TRISA (85H)

Cada bit de este registro configura la dirección en que fluye la información del pin correspondiente del puerto A, así, para $k=0,1,\dots,5$ }

Bit k de TRISA = 1 configura la patita RA_k del puerto A como Entrada

Bit k de TRISA = 0 configura la patita RA_k del puerto A como Salida

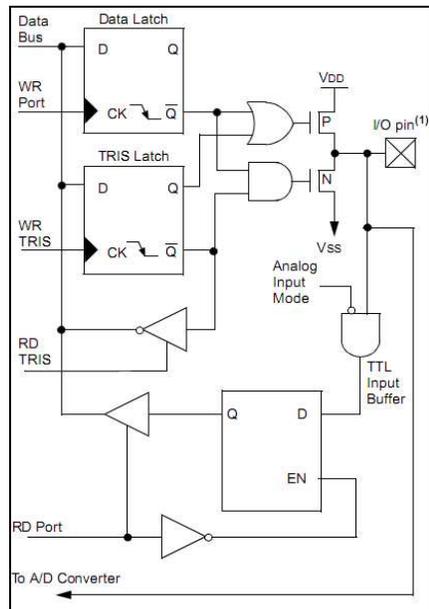
			RA5	RA4	RA3	RA2	RA1	RA0
85h			Registro de dirección de datos del puerto A					

Tabla 8.-

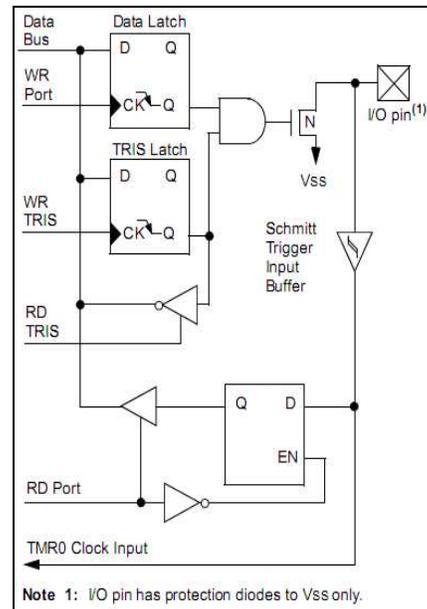
Todos los pines del puerto A poseen diodos de protección conectados a Vdd (contra altos voltajes de entrada) y a Vss (contra voltajes negativos) además, manejan niveles de entrada tipo TTL y como salidas se comportan como drivers tipo CMOS. Excepto el pin RA4, la cual como entrada posee un Disparador Schmitt y como salida es de drenaje abierto, además RA4 sólo posee un diodo de protección conectado a Vss.

3.2.2 El Registro ADCON1 (9FH)

Los pines RA0, RA1, RA2, RA3 y RA5 están multiplexados con las entradas analógicas AN0,..., AN4, de manera que antes de utilizarlas debemos configurar si serán usadas como entradas analógicas o como entradas / salidas digitales. Para seleccionar la segunda opción (entradas / salidas digitales) se debe colocar en la mitad menos significativa de este registro un 01102 (es decir, un 06h).



**Fig.9.- Diagrama en bloque
Pines RA3:RA0 y RA5**



**Fig.10.- Diagrama en Bloque
pin RA4/T0CKI**

3.2.3 Puerto B (PORTB)

El puerto B es un puerto digital de 8 bits, todas sus patitas son bidireccionales y trabaja en forma similar al puerto A. Tiene tres registros asociados: El registro de datos PORTB, el registro de dirección de los datos TRISB y el registro OPTION_REG.

Registro PORTB (06H, 106H)

Los ocho bits que contiene reflejan directamente el estado de los ocho pines del puerto B RB0,..., RB7.

Registro TRISB (86H, 186H)

En forma similar a TRISA, al poner un 0 en un bit de TRISB se configura el pin RB correspondiente como salida y al poner un 1 en un bit de TRISB se configura el pin RB correspondiente como entrada.

Registro OPTION_REG (81H, 181H)

El bit 7 de este registro, denominado RBPU es usado para conectar/desconectar una resistencia “pull-up” conectada a cada pin RB.

Poniendo un 0 en este bit todas las resistencias se conectan. Para desconectar las resistencias “pull-up” se debe poner este bit en 1, también se desconectan automáticamente cuando la patita correspondiente es configurada como salida. Un Reset desconecta todas las resistencias.

Pines RB4 hasta RB7

Estas cuatro patitas del puerto B tienen la capacidad de generar una solicitud de interrupción a la CPU cuando están configuradas como entradas. El estado de estas patitas es comparado con el último estado que tenían durante la última lectura a PORTB, guardado en un latch. Los bits que indican que hay una diferencia entre estos valores por cada patita están conectados a una puerta OR cuya salida activa el bit RBIF del registro INTCON solicitando con esto una interrupción. Esta interrupción es especialmente útil para despertar al dispositivo de su estado de SLEEP cuando alguna de las cuatro líneas es activada, por ejemplo, en respuesta a la presión de una tecla.

Esta característica de solicitud de interrupción cuando se detecta un cambio junto con las resistencias “pull-up” configurables para estas cuatro patitas, las hacen ideales para el manejo de teclados en dispositivos portátiles que requieren “dormirse” durante largos ratos para economizar baterías y “despertarse” cuando una tecla es presionada.

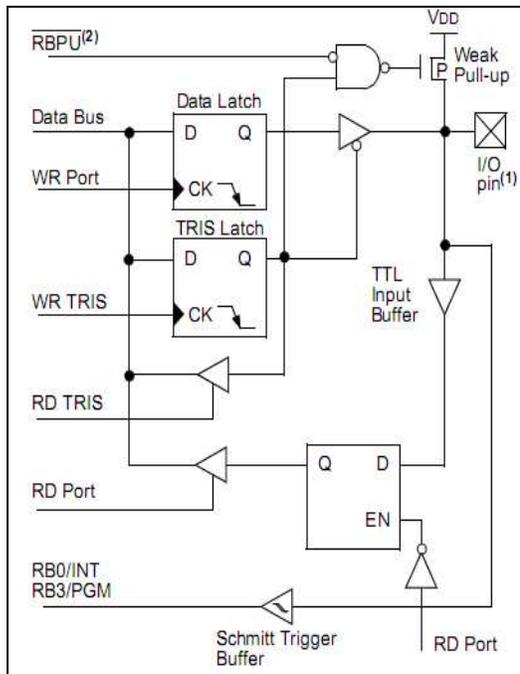


Fig.11.-

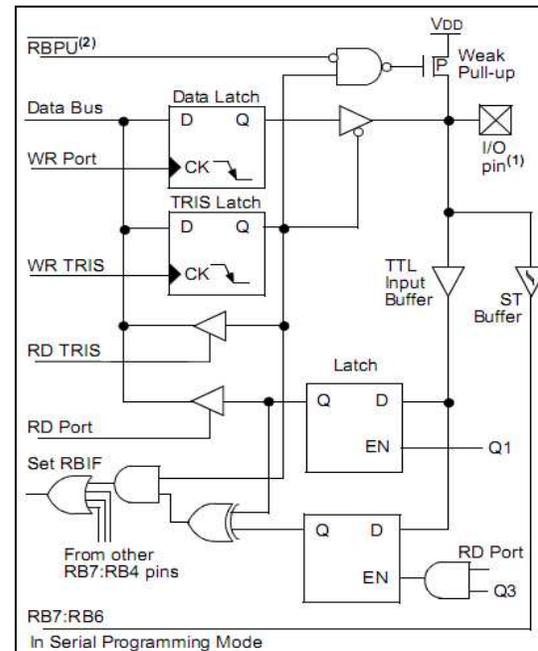


Fig.12.-

3.3 Conversor analógico digital (ADC)

El microcontrolador PIC16F877A de Microchip puede desempeñar muchas funciones y una de ellas es la de su conversor analógico-digital. Antes de meternos de pleno en su funcionamiento vamos a comentar los conceptos básicos de una conversión de una señal analógica a digital.

Tanto nuestra voz como muchas de las señales que se envían a través de un medio guiado como un cable o no guiado como es el aire son de tipo continuo y pueden tomar infinitos valores a lo largo del tiempo. Por ejemplo podemos decir que la señal eléctrica que se transmite de la tarjeta de sonido al altavoz es continua y puede tomar cualquier tensión entre los dos hilos.

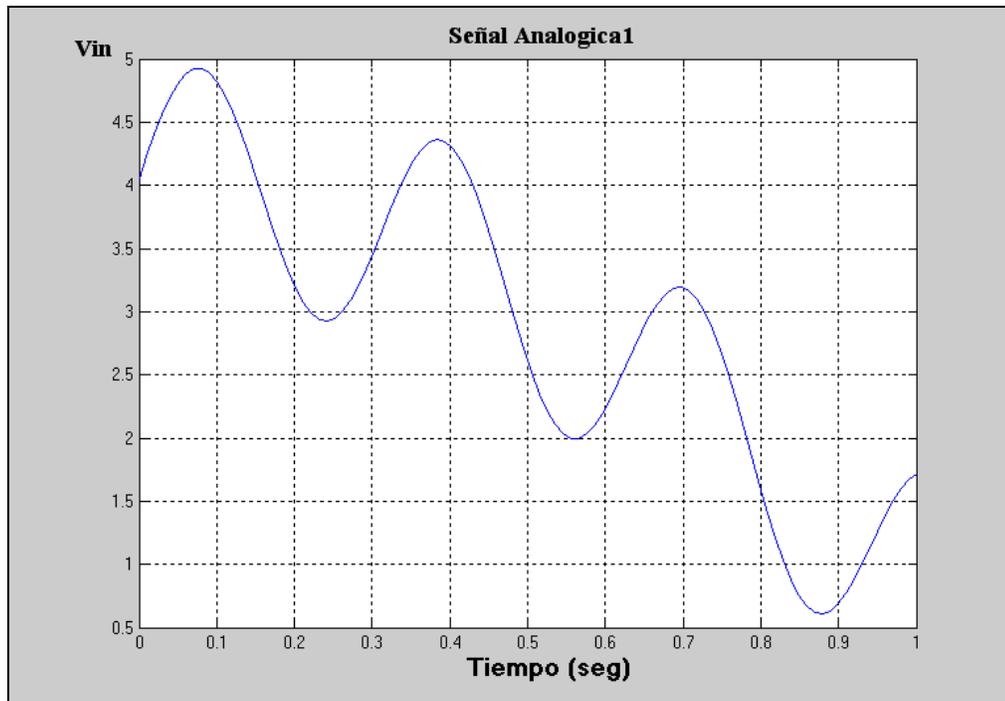


Fig.13.- Señal Analógica

Fig. Octave v.3.0.2, para Windows XP/Vista

El interés en digitalizar una señal puede surgir por varios motivos: el hecho de querer almacenarla en un soporte digital o transmitirla digitalmente para poder reconstruirla, poder tratar con programas los valores analógicos que dé un sensor, etc.

Acotando la señal en un intervalo de tiempo y unos valores mínimos y máximos de tensión tenemos que tener en cuenta dos factores fundamentales a la hora de almacenar dicha señal en un formato adecuado que pueda almacenarse digitalmente (con ceros y unos): se tiene que muestrear y cuantificar.

El muestreo implica que tenemos que coger una muestra de la señal cada T segundos ya que no hay memoria suficiente capaz de almacenar los infinitos puntos de una señal en un intervalo cualquiera de tiempo. En el ejemplo de las figuras se ha acotado un segundo de tiempo y 5 V de tensión de entrada analógica del PIC.

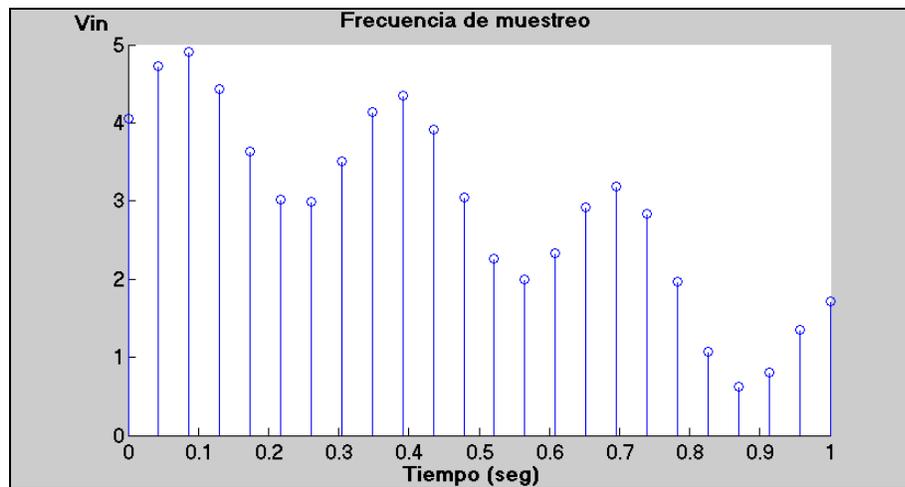


Fig.14.- Muestreo de Señal Analógica

Fig. Octave v.3.0.2, para Windows XP/Vista

En la figura 14 se han tomado 24 muestras en un segundo, esto es, la frecuencia de muestreo.

La cuantificación surge por el mismo motivo que el muestreo pero para el eje de ordenadas: una vez tenemos una muestra su amplitud puede tomar infinitos valores, debemos redondear entre unos valores fijos a lo largo de ese eje. Estos valores van a depender del número de bits que vayamos a almacenar para cada muestra, por ejemplo, en la imagen se cogen 4 bits y con ellos se pueden formar 16 combinaciones y por lo tanto 16 distintos niveles en los que se puede dividir el eje. El PIC cuantifica con 10 bits luego son 1023 niveles.

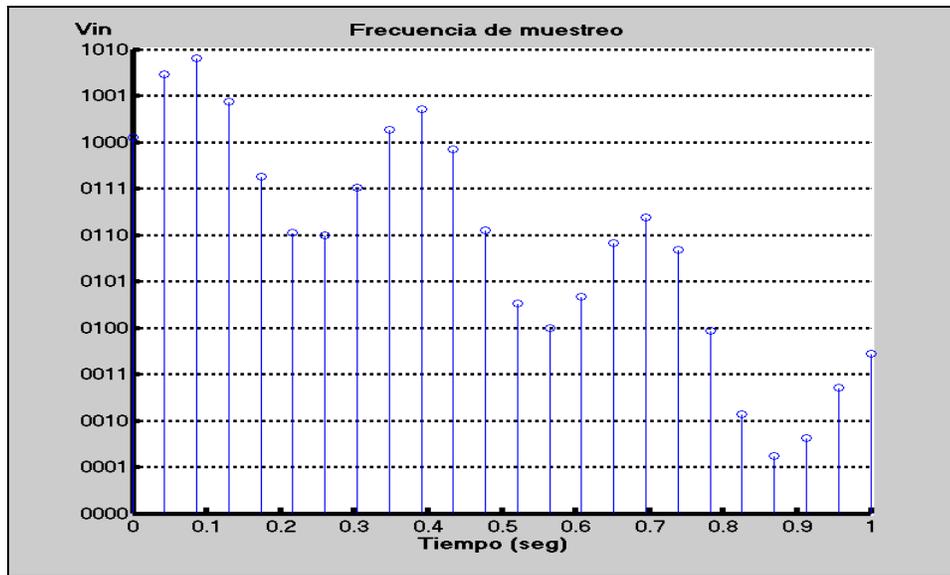


Fig.15.- Señal Cuantificada

Fig. Octave v.3.0.2, para Windows XP/Vista

Tanto en un proceso como en el otro, se va a provocar una pérdida de la forma de la señal original y eso implica que se va a cometer un error de aproximación cuya magnitud se puede calcular y se comenta con más detalle en el siguiente apartado.

3.3.1 Técnicas de cuantificación

Como información complementaria, decir que existen diferentes tipos de cuantificación, cada uno de las cuales se amoldara a los datos que estemos enviando.

- **Cuantificación uniforme:** la distancia entre los niveles de reconstrucción es siempre la misma. No hacen ninguna suposición acerca de la naturaleza de la señal a cuantificar, de ahí que no proporcionen los mejores resultados. Sin embargo, tienen como ventaja que son los más fáciles y menos costosos de implementar.
- **Cuantificación logarítmica:** incrementa la distancia entre los niveles de reconstrucción conforme aumenta la amplitud de la señal. Muy usada en señales de voz.

- **Cuantificación no uniforme:** si conocemos la función de la distribución de probabilidad, podemos ajustar los niveles de reconstrucción a la distribución de forma que se minimice el error cuadrático medio. Esto significa que la mayoría de los niveles de reconstrucción se den en la vecindad de las entradas más frecuentes y, consecuentemente, se minimice el error (ruido).
- **Cuantificación vectorial:** se basa en cuantificar según las muestras vecinas, resulta más eficiente cuantificar los datos en bloques de N muestras.
- **Cuantificadores de Vecino Más Cercano (Voronoi):** el proceso de codificación no necesita almacenar una descripción geométrica de las celdas, la codificación es mediante una comparación de distancias.

3.3.2 Módulo ADC del PIC 16F877A

El PIC16F877A posee un módulo ADC interno que les permite manejar 8 entradas analógicas. En la siguiente figura se muestra un diagrama de bloques del módulo ADC.

Se carga una muestra a través de la entrada analógica y se mantiene en un condensador. La salida de esta muestra coincide con la entrada del convertidor. El conversor entonces genera un resultado digital de este nivel analógico a través de sucesivas aproximaciones. La conversión A/D de la señal de entrada analógica resulta en un número digital de 10-bit. El modulo A/D tiene entradas de referencias de alto y bajo voltaje, estos son configurables por software combinando VDD, VSS, RA2 o RA3.

El modulo A/D tiene cuatro registros. Estos registros son:

- A/D Registro de Resultado Alto (ADRESH)
- A/D Registro de Resultado Bajo (ADRESL)
- A/D Registro de Control 0 (ADCON0)
- A/D Registro de Control 1 (ADCON1)

El registro ADCON0, el que es mostrado a continuación, controla la operación del modulo conversor A/D. El registro ADCON1 configura las funciones de los pines del puerto.

Registro ADCON0

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit 7						bit 0	

Tabla 9.-

➤ **bit 7-6 ADCS1:ADCS0:** bits de selección del reloj de conversión del A/D

00 = FOSC/2

01 = FOSC/8

10 = FOSC/32

11 = FRC (Oscilador interno RC)*

* El oscilador RC interno del convertidor tiene un TAD típico de 4 μ seg, sin embargo, puede variar entre 2 y 6 μ seg. Este reloj es recomendable para operación en modo SLEEP, ya que este modo desconecta la frecuencia del reloj externo.

* Precaución: El convertidor A/D no trabajará correctamente con un TAD menor que TAD (mínimo) = 1.6 μ seg. El usuario deberá cuidar la elección del reloj adecuado para no violar esta limitante.

➤ **bit 5-3 CHS2:CHS0:** bit de selección del canal analógico

000 = Canal 0, (RA0/AN0)

001 = Canal 1, (RA1/AN1)

010 = Canal 2, (RA2/AN2)

011 = Canal 3, (RA3/AN3)

100 = Canal 4, (RA5/AN4)

101 = Canal 5, (RE0/AN5)

110 = Canal 6, (RE1/AN6)

111 = Canal 7, (RE2/AN7)

- **bit 2 GO/DONE:** bit del estado de Conversión A/D

Si ADON=1

1=La conversión A/D esta en progreso

0=La conversión A/D no esta en progreso

- **bit1** no esta implementado. Se lee como '0'

- **bit0 ADON:** Encendido del convertidor

1=El convertidor se enciende

0=Se apaga y no consume corriente

Registro ADCON1

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

Tabla.10.-

- **bit7 ADFM:** bit de selección del formato del resultado A/D

1= Justificado a la derecha. Los 6 bits mas significativos de ADRESH son leídos como '0'

0=Justificado a la izquierda. Los 6 bits menos significativos de ADRESL son leídos como '0'.

- **bit 6-4** No están implementados

- **bit 3-0 PCFG3:PCFG0** Bits de configuración de las entradas del convertidor

PCFG3: PCFG0	AN7 ⁽¹⁾ RE2	AN6 ⁽¹⁾ RE1	AN5 ⁽¹⁾ RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	CHAN/ Refs ⁽²⁾
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	RA3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	RA3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	RA3	VSS	2/1
011x	D	D	D	D	D	D	D	D	VDD	VSS	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1/2

Tabla.11.-

A= Entrada analógica

D= Entrada y salida Digital

3.3.3 Configuración del conversor A/D

Para lograr la conversión con el modulo A/D se debe seguir los siguientes pasos. Estos pasos son entregados por el fabricante.

1. Configurar el modulo A/D:
 - configurar pines analógicos, referencias de voltaje y entradas y salidas digitales a través del registro ADCON1.
 - Seleccionar canales de entrada A/D en el registro ADCON0
 - Seleccionar el reloj de conversión en el registro ADCON0.
 - Encender el modulo A/D en el registro ADCON0.
2. Configurar interrupciones A/D (si se desea):
 - Limpiar bit ADIF

- Establecer bit ADIE.
 - Establecer bit PEIE
 - Establecer bit GIE
3. Esperar el tiempo de adquisición necesaria.
 4. Iniciar la conversión:
 - Establecer bit GO/DONE(ADCON0)
 5. Esperar que el tiempo de conversión este completada:
 - A través de la técnica de poleo chequear si bit GO/DONE esta limpio (con interrupciones habilitadas)
 - Esperar por la interrupción A/D
 6. Leer resultado A/D en el par de registro (ADRESH; ADRESL), limpiar el bit ADIF si es necesario.
 7. Para la siguiente conversión seguir en el paso 1 o el paso 2. Para la siguiente conversión, esperar al menos $2TAD$ (Donde TAD es el tiempo de conversión por bit).

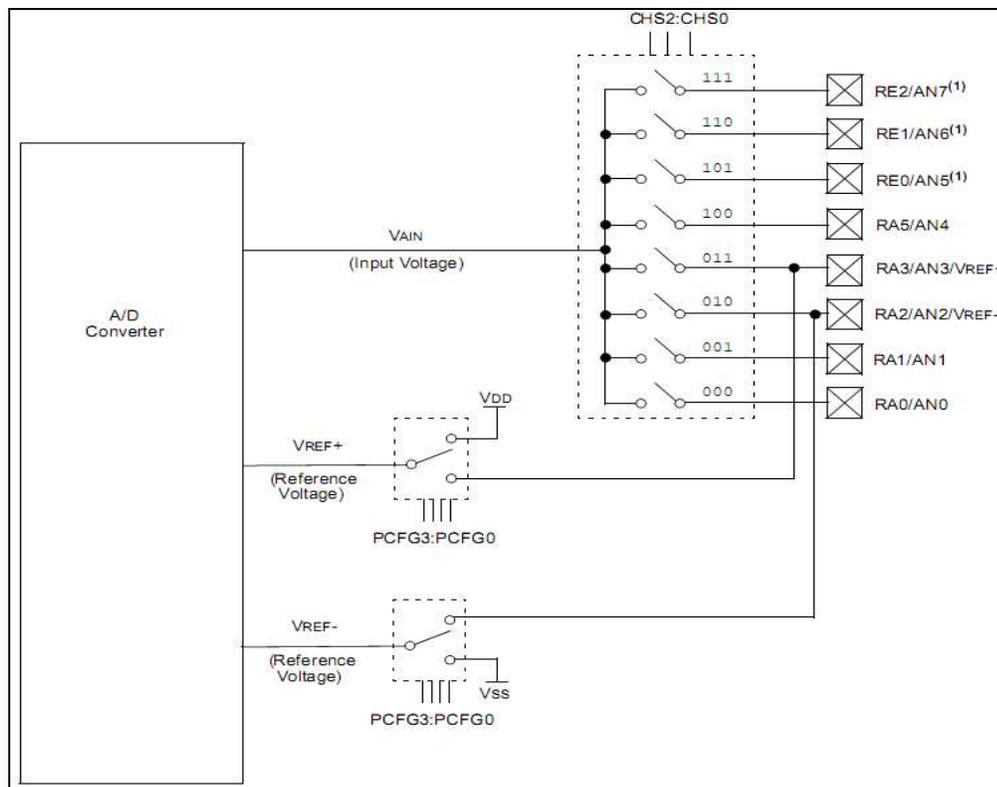


Fig.16.- Diagrama en bloques

3.3.4 Registros De Resultado A/D

El par de registros ADRESH: ADRESL (1Eh:9Eh) son cargados con el dato (D) resultante de una conversión analógico / digital al terminar ésta. Cada uno de estos registros es de 8 bits, por lo tanto, juntos pueden guardar hasta 16 bits. Sin embargo, como el resultado D es de 10 bits, el módulo de conversión A/D permite justificarlo (alinearlo) en la parte izquierda o derecha de los 16 bits disponibles, para elegir alguna de las dos opciones se usa el bit ADFM (ADCON1<7>) como se muestra en la figura 17.

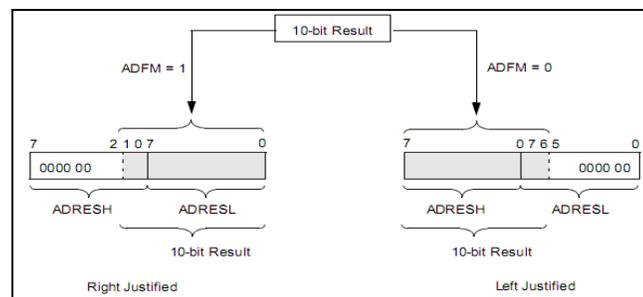


Fig.17.- Registros ADRESH: ADRESL

Características Eléctricas Del Convertidor

Característica	mínimo	típico	máximo
VREF+-VREF-	2v	-	VDD+0.3v
VREF+	VDD-2.5v	-	VDD+0.3v
VREF-	VSS-0.3v	-	VREF+-2v
Voltaje analógico VAIN	VSS-0.3v	-	VREF ++0.3v
Impedancia de la fuente de señal externa ZAIN	-	-	10 KW
Corriente promedio consumida por el convertidor IAD	Estándar	-	220µA
	Extendido	-	90µA

Tabla.12.-

3.4 Modulo de Entrada y Salida USART

La USART (Universal Synchronous Asynchronous Receiver Transmitter) es uno de los dos periféricos contenidos en el PIC que le permiten realizar comunicación en serie. El otro es el MSSP (Master Synchronous Serial Port), pero de este no se hablara nada debido a que no fue utilizado para este trabajo.

La USART, también conocida como SCI (Serial Communications Interface) puede configurarse como una unidad de comunicación en serie para la transmisión de datos asíncrona con dispositivos tales como terminales de computadora o computadoras personales, o bien para comunicación síncrona con dispositivos tales como convertidores A/D o D/A, circuitos integrados o memorias EEPROM con comunicación serie, etc.

La gran mayoría de los sistemas de comunicación de datos digitales actuales utilizan la comunicación en serie, debido a las grandes ventajas que representa esta manera de comunicar los datos:

- **Económica:** Utiliza pocas líneas de transmisión inclusive puede usar sólo una línea.
- **Confiable:** Los estándares actuales permiten transmitir datos con bits de paridad y a niveles de voltaje o corriente que los hacen poco sensibles a ruido externo. Además por tratarse de información digital, los cambios en amplitud de las señales (normalmente causados por ruido) afectan muy poco o nada a la información.
- **Versátil:** No está limitada a usar conductores eléctricos como medio de transmisión, pudiendo usarse también: fibra óptica, aire, vacío, etc. Además el tipo de energía utilizada puede ser diferente: luz visible, infrarroja, ultrasonido, pulsos eléctricos, radio frecuencia, microondas, etc.

Una gran cantidad de periféricos se comunican actualmente en serie con un micro computador: líneas telefónicas, terminales remotas, unidades de cassette magnético, el ratón, teclados, etc.

Comunicación en paralelo: En este caso se utiliza una línea física por cada bit del dato a comunicar además de posibles líneas para protocolo. Esquemáticamente en la siguiente figura se muestra como se transmitiría el dato de 8 bits 1001 0111= 97h. Este tipo de comunicación se puede realizar mediante el PIC usando el puerto D como puerto de datos y las líneas del puerto E como líneas de protocolo.

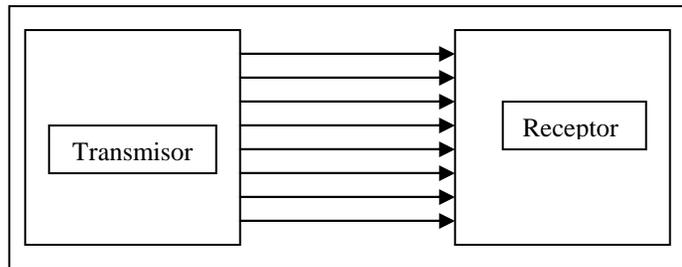


Fig.18.- Comunicación en Paralelo

La principal ventaja de la comunicación en paralelo es la alta velocidad de transmisión, ya que se envían simultáneamente todos los bits de un dato. No obstante, si la distancia entre el transmisor y el receptor es grande, puede ser que el costo de las líneas sea tan alto que se vuelva incosteable este método de comunicación.

Comunicación en Serie: En cambio, la comunicación en serie sólo utiliza una línea para la transmisión de datos, y opcionalmente alguna línea o líneas para protocolo. Por ejemplo, en la siguiente figura se muestra como se transmitiría en serie el mismo dato. La desventaja obvia de la comunicación serie es que los bits de un dato se envían de a uno por uno, de manera que mientras que la comunicación en paralelo envía en un ciclo un dato de 8 bits, a la comunicación serie le toma más de 8 ciclos (ya que además del dato en la comunicación serie se requiere agregar algunos bits de sincronización).

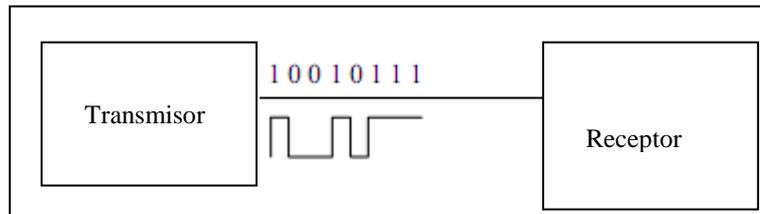


Fig.19.- Comunicación en Serie

Sin embargo, debido a que la comunicación serie requiere sólo una línea para la transmisión esto abarata los costos en líneas de transmisión y no sólo esto, ya que este hecho también hace posible que los datos puedan ser enviados no necesariamente por un conductor eléctrico, sino inclusive por aire o por el vacío si en lugar de pulsos eléctricos se usan impulsos electromagnéticos, tales como: ondas de radio, microondas, pulsos luminosos, infrarrojo, ultrasonido, láser (a través de fibra óptica), etc.

3.4.1 Protocolo De Comunicación Serie

A diferencia de la comunicación en paralelo, en la comunicación en serie se hace necesario establecer métodos de sincronización para evitar la interpretación errónea de los datos transmitidos. Para ilustrar esto consideremos la siguiente información en serie:

...0100110001001100100....

Esta información puede interpretarse de diversas maneras, (aún si se recibe a la velocidad adecuada) dependiendo del punto de inicio de separación de datos, por ejemplo, una posible interpretación sería como sigue:

...01	00110001	00110010	0.....
-------	----------	----------	--------

Tabla.13.-

Que interpretado como códigos ASCII corresponde a los caracteres '1' y '2'. Sin embargo, otra posible interpretación es:

...010	01100010	01100100
--------	----------	----------

Tabla.14.-

Que corresponde a los caracteres 'b' y 'd'.

Sincronización De Bit

Una manera de resolver el problema anterior es la sincronización de bits que puede realizarse por varios métodos:

1. Enviar por una línea adicional una señal de reloj que indique el centro de las celdas de bits en la línea de datos (datos no - auto reloj).
2. Enviar con cada bit y por la misma línea de datos información que permita extraer la señal de reloj (datos auto reloj).
3. Lograr mediante alguna estrategia que los relojes de transmisión y de recepción se mantengan en fase continuamente.

Codificación No Auto Reloj

En la figura 20 se muestran las tres codificaciones de una línea de datos:

- RZ.- Una celda de bit es 1 si contiene un impulso positivo y un 0 si no lo contiene.
- NRZ.- La celda contiene un 1 o 0 de acuerdo al nivel de la señal (constante) en la celda.
- NRZI.- La celda de bit contiene un 1 si hay una transición y un 0 si no la hay.

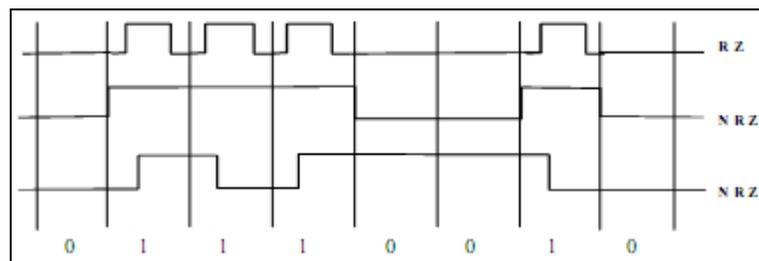


Fig.20.-

Como puede verse, en estos sistemas (RZ, NRZ y NRZI) las secuencias de ceros no contienen ninguna transición que permita ubicar la situación de las celdas de bit. de hecho, el formato NRZ no la contiene ni en los unos.

Codificación Auto Reloj

Algunos métodos que contienen en la misma línea de datos información adicional para determinar la velocidad del reloj a costa de disminuir la cantidad de información útil a la mitad que los métodos no-auto reloj. En la figura 21 se muestran los más utilizados, como son:

PE: Codificación de fase.

FSC: Codificación por cambio de frecuencia

FM: Modulación de frecuencia.

MFM y M2

FM: Modulación de frecuencia modificadas.

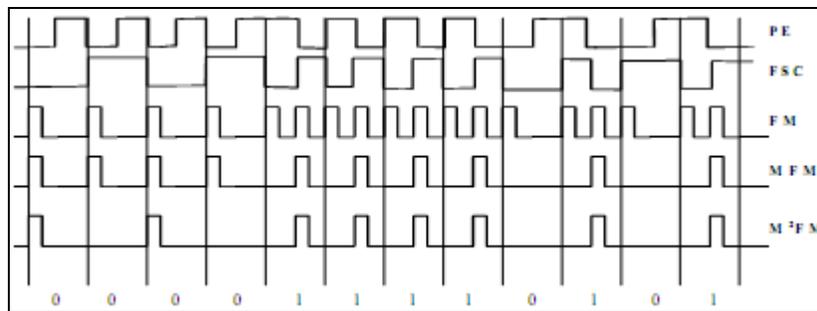


Fig.21.-

Los métodos autoreloj son muy útiles cuando la velocidad de transmisión no es constante, por ejemplo, cuando los datos han sido grabados en un medio magnético giratorio, por ejemplo discos, cintas magnéticas, etc.

Sincronización de carácter

Algunos sistemas utilizan líneas adicionales que envían impulsos para indicar el inicio de un bloque de caracteres. Otros sistemas que no requieren líneas adicionales a la línea de datos son:

Método Asíncrono

Cada carácter va señalizado mediante dos bits: un bit de inicio y un bit de paro, estos dos bits permiten al receptor reconocer el inicio y el final de cada carácter. La especificación RS404 de EIA (Electronic Industries Association) define las características del método asíncrono para transmisión en serie de acuerdo a las siguientes reglas:

1. Cuando no se envían datos la línea debe mantenerse en estado 1.
2. Cuando se va a mandar un carácter se envía primero un bit de inicio de valor 0.
3. A continuación se envían todos los bits del carácter a transmitir al ritmo marcado por el reloj de transmisión.
4. Después del último bit del carácter enviado se envía un bit de paro de valor 1.

Método Síncrono

Cada mensaje o bloque de transmisión va precedido de unos caracteres de sincronismo. Así, cuando el receptor identifica una configuración de bits igual a la de los caracteres de sincronismo da por detectado el inicio y el tamaño de los datos.

3.5 La USART del PIC 16f877A

La USART del PIC puede ser configurada para operar en tres modos:

Modo Asíncrono (full duplex (transmisión y recepción simultáneas)),

Modo Síncrono – Maestro (half duplex)

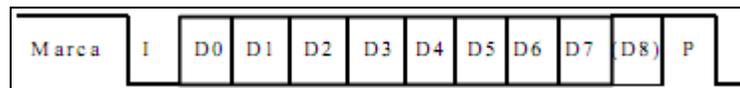
Modo Síncrono – Esclavo (half duplex)

En este trabajo se utilizó el modo asíncrono, el cual detallaremos a continuación.

3.5.1 Modo Asíncrono

En este modo la USART usa un formato estándar NRZ asíncrono, el cual para la sincronización usa: 1 bit de inicio (I), 8 o 9 bits de datos y 1 bit de paro (P). Mientras no se están transmitiendo datos la USART envía continuamente un bit de marca. El modo asíncrono se selecciona limpiando el bit SYNC del registro TXSTA (98H). El modo asíncrono es deshabilitado durante el modo SLEEP.

Cada dato es transmitido y recibido comenzando por el LSB. El hardware no maneja bit de Paridad, pero el noveno bit puede ser usado para este fin y manejado por software.



El módulo Asíncrono de la USART consta de 3 módulos fundamentales:

- El circuito de muestreo
- El generador de frecuencia de transmisión (Baud Rate)
- El transmisor asíncrono
- El receptor asíncrono.

3.5.2 El Circuito De Muestreo

El dato en el pin de recepción (RC7/RX/DT) es muestreado tres veces para poder decidir mediante un circuito de mayoría, si se trata de un nivel alto o un nivel bajo.

3.5.3 El Generador De Baud Rate (BRG)

Este generador sirve tanto para el modo síncrono como el asíncrono y consiste de un contador/divisor de frecuencia de 8 bits controlado por el registro SPBRG (99H). De tal manera que la frecuencia de transmisión se calcula de acuerdo a la siguiente tabla:

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = $F_{osc}/(64(X+1))$	Baud Rate = $F_{osc}/(16(X+1))$
1	(Synchronous) Baud Rate = $F_{osc}/(4(X+1))$	N/A

Tabla.15.- Formula Baud Rate

X= Valor en SPBRG (0 a 255)

BAUD RATE (K)	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-	-	-	-
1.2	1.221	1.75	255	1.202	0.17	207	1.202	0.17	129
2.4	2.404	0.17	129	2.404	0.17	103	2.404	0.17	64
9.6	9.766	1.73	31	9.615	0.16	25	9.766	1.73	15
19.2	19.531	1.72	15	19.231	0.16	12	19.531	1.72	7
28.8	31.250	8.51	9	27.778	3.55	8	31.250	8.51	4
33.6	34.722	3.34	8	35.714	6.29	6	31.250	6.99	4
57.6	62.500	8.51	4	62.500	8.51	3	52.083	9.58	2
HIGH	1.221	-	255	0.977	-	255	0.610	-	255
LOW	312.500	-	0	250.000	-	0	156.250	-	0

BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	0.300	0	207	0.3	0	191
1.2	1.202	0.17	51	1.2	0	47
2.4	2.404	0.17	25	2.4	0	23
9.6	8.929	6.99	6	9.6	0	5
19.2	20.833	8.51	2	19.2	0	2
28.8	31.250	8.51	1	28.8	0	1
33.6	-	-	-	-	-	-
57.6	62.500	8.51	0	57.6	0	0
HIGH	0.244	-	255	0.225	-	255
LOW	62.500	-	0	57.6	-	0

Tabla.16.- Baud Rate para modo asíncrono (BRGH=0)

BAUD RATE (K)	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-	-	-	-
1.2	-	-	-	-	-	-	-	-	-
2.4	-	-	-	-	-	-	2.441	1.71	255
9.6	9.615	0.16	129	9.615	0.16	103	9.615	0.16	64
19.2	19.231	0.16	64	19.231	0.16	51	19.531	1.72	31
28.8	29.070	0.94	42	29.412	2.13	33	28.409	1.36	21
33.6	33.784	0.55	36	33.333	0.79	29	32.895	2.10	18
57.6	59.524	3.34	20	58.824	2.13	16	56.818	1.36	10
HIGH	4.883	-	255	3.906	-	255	2.441	-	255
LOW	1250.000	-	0	1000.000	-	0	625.000	-	0

BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-
1.2	1.202	0.17	207	1.2	0	191
2.4	2.404	0.17	103	2.4	0	95
9.6	9.615	0.16	25	9.6	0	23
19.2	19.231	0.16	12	19.2	0	11
28.8	27.798	3.55	8	28.8	0	7
33.6	35.714	6.29	6	32.9	2.04	6
57.6	62.500	8.51	3	57.6	0	3
HIGH	0.977	-	255	0.9	-	255
LOW	250.000	-	0	230.4	-	0

Tabla.17.- Baud Rate Para Modo Asíncrono (BRGH=1)

Debido a que el divisor es de 8 bits, no se puede tener cualquier velocidad de transmisión deseada, ya que X se deberá redondear al entero más cercano. En las dos tablas anteriores se muestran algunos valores baud estándares, el divisor necesario ($X=SPBRG$) bajo diferentes frecuencias Fosc y el error producido en porcentaje.

3.5.4 El Transmisor Asíncrono

En la siguiente figura se muestra el diagrama de bloques del transmisor de la USART. El corazón de este módulo es el registro de corrimiento (transmit shift register, TSR). La única manera de acceder al registro TSR es a través del registro TXREG (19H).

Para transmitir un dato, el programa deberá ponerlo primero en el registro TXREG. En cuanto el TSR termina de enviar el dato que tenía (en cuanto transmite el bit de paro) lee el dato contenido en TXREG (si hay alguno) esto ocurre en un ciclo TCY. En cuanto el dato de TXREG es transferido al TSR el TXREG queda vacío esta condición es indicada mediante el bit bandera TXIF (que es el bit 4 del registro PIR1 (0Ch)), el cual se pone en alto. Este bit NO puede ser limpiado por software, sólo dura un instante en bajo cuando se escribe un nuevo dato a TXREG. Si se escribe un dato seguido de otro (back to back) a TXREG el primero se transfiere inmediatamente a TSR y el otro tiene que esperar hasta que el TSR termine de enviar el bit de Stop del primero. Durante esta espera TXIF permanece en bajo.

Existe otro bit, llamado TRMT (TXSTA<1>), el cual muestra el estado del TSR. TRMT se pone en alto cuando TSR está vacío, y en bajo cuando TSR está transmitiendo un dato. mientras que TXIF puede generar una interrupción TRMT no lo puede hacer, TRMT está pensado para ser consultado por “poleo” (sin usar interrupciones).

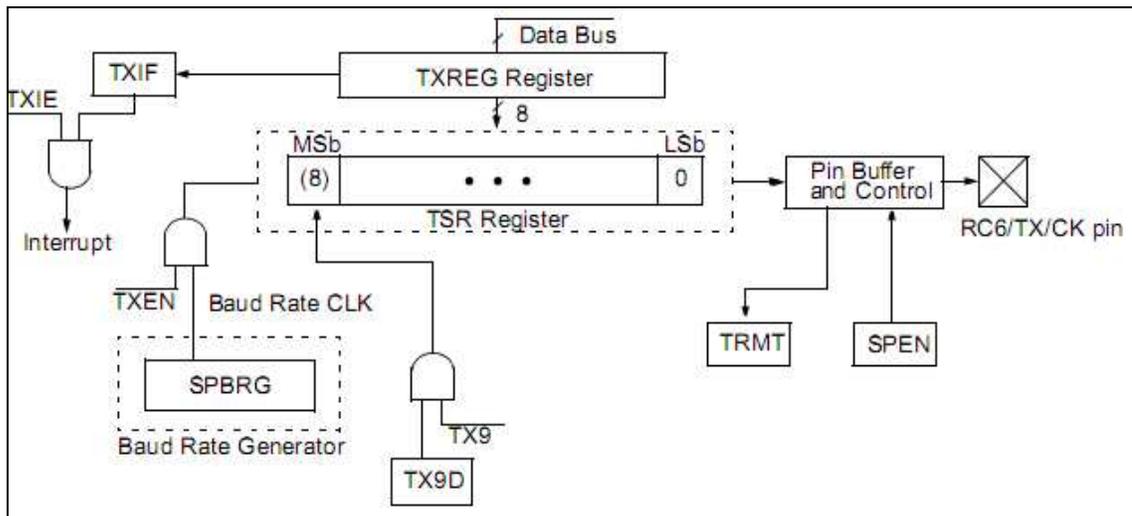


Fig.22.- Diagrama en bloque del transmisor USART

Para habilitar el módulo de transmisión es necesario poner en alto el bit TXEN (TXSTA<5>), mientras no se habilite el módulo, la patita de transmisión (RC6/TX/CK) se mantiene en alta impedancia. Si TXEN es deshabilitada a la mitad de una transmisión, esta será abortada y el transmisor será reseteado.

Si se está usando un noveno bit TX9 (TXSTA<6>), éste deberá ser escrito antes de escribir los 8 bits restantes a TXREG, ya que en cuanto se escribe un dato a este registro inmediatamente es transferido a TSR (si éste está vacío).

De acuerdo a lo anterior, la inicialización del módulo de transmisión consiste en los siguientes pasos:

1. Inicializar baud rate escribiendo al registro SPBRG el divisor adecuado y opcionalmente al bit BRGH.
2. Habilitar comunicación asíncrona limpiando el bit SYNC y poniendo el bit SPEN.
3. Si se van a usar interrupciones, poner el bit TXIE (PIE<4>).
4. Poner el bit TX9 si se desea transmitir datos de 9 bits
5. Habilitar transmisión poniendo el bit TXEN, lo cual pondrá el bit TXIF.
6. Colocar el noveno bit del dato en TX9D si se están usando datos de 9 bits.
7. Cargar el dato al registro TXREG (inicia la transmisión).

3.5.5 El Receptor Asíncrono

El módulo de recepción es similar al de transmisión, en la siguiente figura se muestran los bloques que lo constituyen. Una vez que se ha seleccionado el modo asíncrono, la recepción se habilita poniendo en alto el bit CREN (RCSTA<4>).

El dato es recibido mediante la línea RC7/RX/DT, la cual maneja un registro de corrimiento de alta velocidad (16 veces el Baud rate).

El corazón del receptor es el registro de corrimiento RSR. Este registro no es accesible por software, pero, cuando el dato recibido se ha completado (se ha recibido el bit de Stop) el dato de RSR es transferido automáticamente al registro RCREG (1Ah) si éste está vacío y al mismo tiempo es puesto en alto la bandera de recepción RCIF (PIR1<5>). La única manera de limpiar la bandera RCIF es leyendo el los datos del registro RCREG. El registro RCREG puede contener hasta dos datos, ya que es un buffer doble que funciona como una cola de dos posiciones.

Si las dos posiciones del registro RCREG están llenas (no han sido leídas) y se detecta el bit de Stop de un tercer dato de recepción, lo cual ocasiona un transferencia automática del dato recibido a RCREG, esto destruirá el primer dato recibido y activará el indicador de sobrescritura OERR (RCSTA<1>). Para evitar esto, se deberán leer los dos datos en RSREG haciendo dos lecturas consecutivas.

La única manera de limpiar el bit OERR una vez que ha sido activado es reseteando el módulo de recepción (limpiando CREN y volviéndolo a poner), si no se limpia OERR se bloquea la transferencia de datos de RSR a RCREG y no puede haber más recepción de datos.

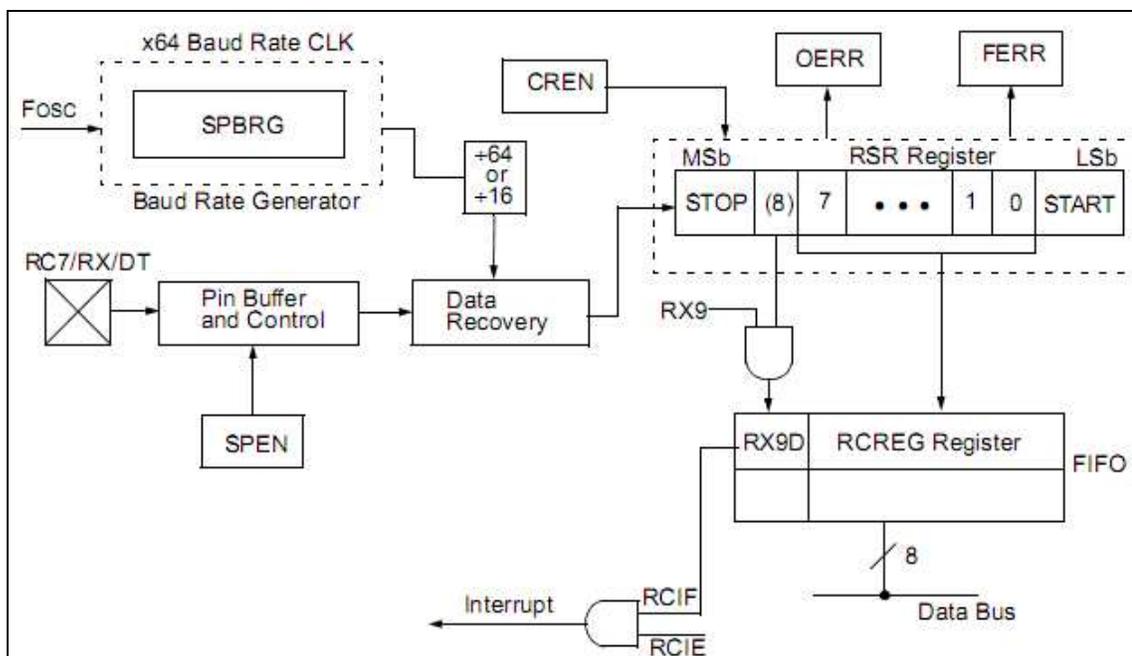


Fig.23.- Diagrama En Bloque Del Receptor USART

Si se detecta un bit nivel bajo en la posición del bit de stop se pone el indicador de error de encuadre (frame error) FERR RCSTA<2>. Tanto este indicador como el noveno bit RX9D de los datos están en una cola de dos posiciones al igual que los datos recibidos, de manera que al leer RCREG se actualizan FERR y RX9D con nuevos valores, por lo cual estos bits deberán ser leídos antes de leer RCREG para no perder su información.

De acuerdo a lo anterior, la inicialización del módulo de recepción es como sigue:

1. Inicializar el baud rate escribiendo al registro SPBRG el divisor adecuado y opcionalmente al bit BRGH.
2. Habilitar el puerto serie asíncrono limpiando el bit SYNC y poniendo el bit SPEN.
3. Si se van a usar interrupciones, poner el bit RCIE (PIE<5>)
4. Si se desea recepción de datos de 9 bits se deberá poner el bit RX9 (RCSTA<0>)
5. Habilitar la recepción poniendo el bit CREN (RCSTA<4>)
6. El bit RCIF se pondrá cuando la recepción de un dato se complete y se generará una interrupción si RCIE está puesto.
7. Leer el registro RCSTA para obtener el noveno bit (si se están recibiendo datos de 9 bits) o para determinar si ha ocurrido un error de recepción.
8. Leer los 8 bits del dato recibido leyendo el registro RCREG
9. Si ocurrió algún error este se limpia al limpiar el bit CREN, el cual deberá volver a ponerse si se desea continuar la recepción.

Capítulo IV: Software Utilizados

4.1 Microchip Mplab IDE



Fig. MPLAB v.8.0, para Windows 98/2000/XP

MPLAB-IDE es una Plataforma de Desarrollo Integrada bajo Windows, con múltiples prestaciones, que permite escribir el programa para los PIC en lenguaje ensamblador (assembler) o en C, crear proyectos, ensamblar o compilar, simular el programa y finalmente programar el componente, si se cuenta con el programador adecuado. MPLAB incorpora todas las utilidades necesarias para la realización de cualquier proyecto y, para los que no dispongan de un emulador, el programa permite editar el archivo fuente en lenguaje ensamblador de nuestro proyecto, además de ensamblarlo y simularlo en pantalla, pudiendo ejecutarlo posteriormente en modo paso a paso y ver como evolucionarían de forma real tanto sus registros internos, la memoria RAM y/o EEPROM de usuario como la memoria de programa, según se fueran ejecutando las instrucciones. Además el entorno que se utiliza es el mismo que si se estuviera utilizando un emulador.

4.1.1 Ficheros Resultantes Del Ensamblado

Tras el ensamblado del fichero fuente *.asm se producen varios ficheros, los más importantes son:

- Fichero ejecutable o hexadecimal. Es un fichero con datos numéricos codificados en hexadecimal. tiene la extensión *.hex. Contiene los códigos maquina del programa que servirán para grabar la memoria del programa del microcontrolador y ejecutarlo.
- Fichero de errores. es un fichero con extensión *.err. contiene los errores producidos durante el proceso de ensamblado.
- Fichero listable. Es un fichero de texto con la extensión *.lst que contiene toda la información del programa: código fuente, códigos de maquina, direcciones de cada instrucción, errores producidos, etc.

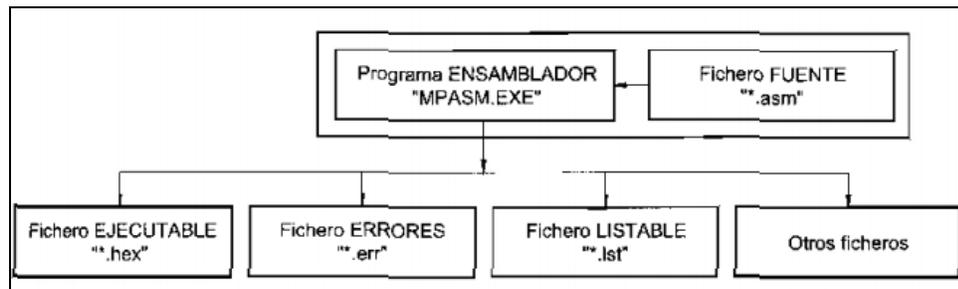
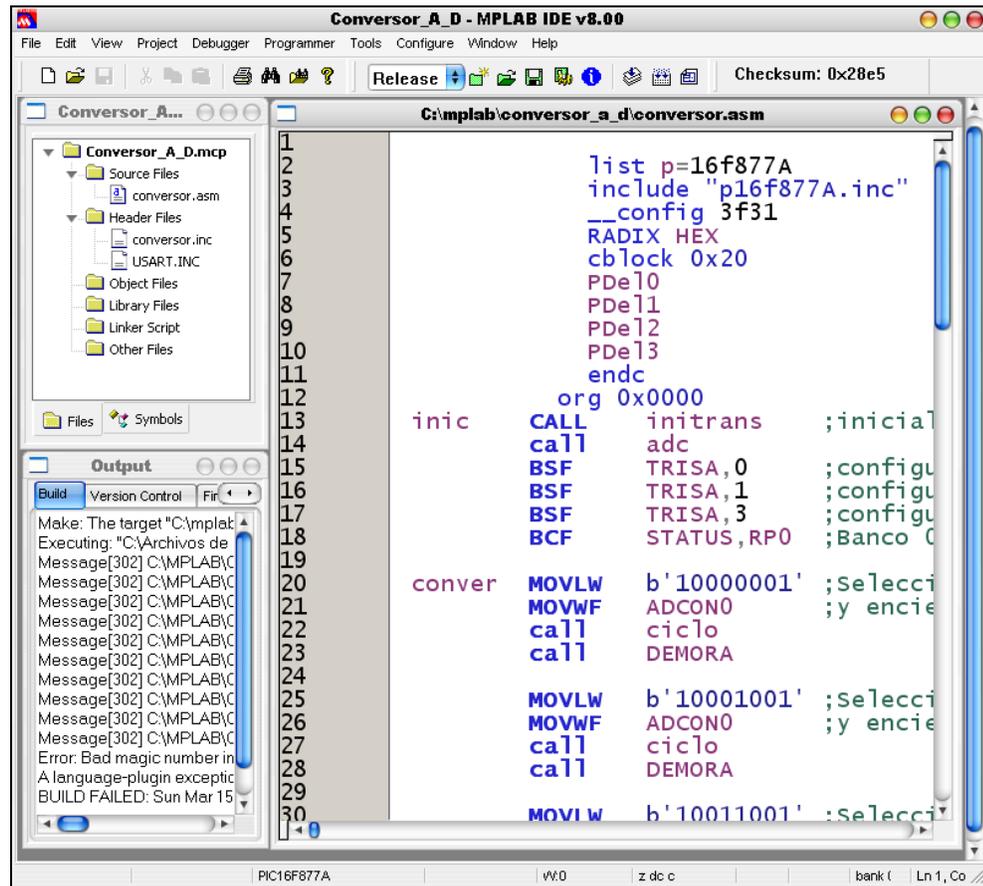


Fig.24.- Ficheros Resultantes

En esta imagen se puede apreciar una ventana común de MPLAB, el cual contiene:

- Una ventana de visualización de archivos. Aquí podemos acceder al programa principal reconocida por la extensión *.asm y las librerías con extensión *.inc o *.lbr que se utilizaran durante el proceso de programación.
- Ventana de programación. En esta ventana se escribe en código assembler el programa general que ejecutara posteriormente el microcontrolador.
- Ventana de información. Acá veremos los mensajes de advertencias y errores después de la compilación.



Img.3.- Entorno MPLAB

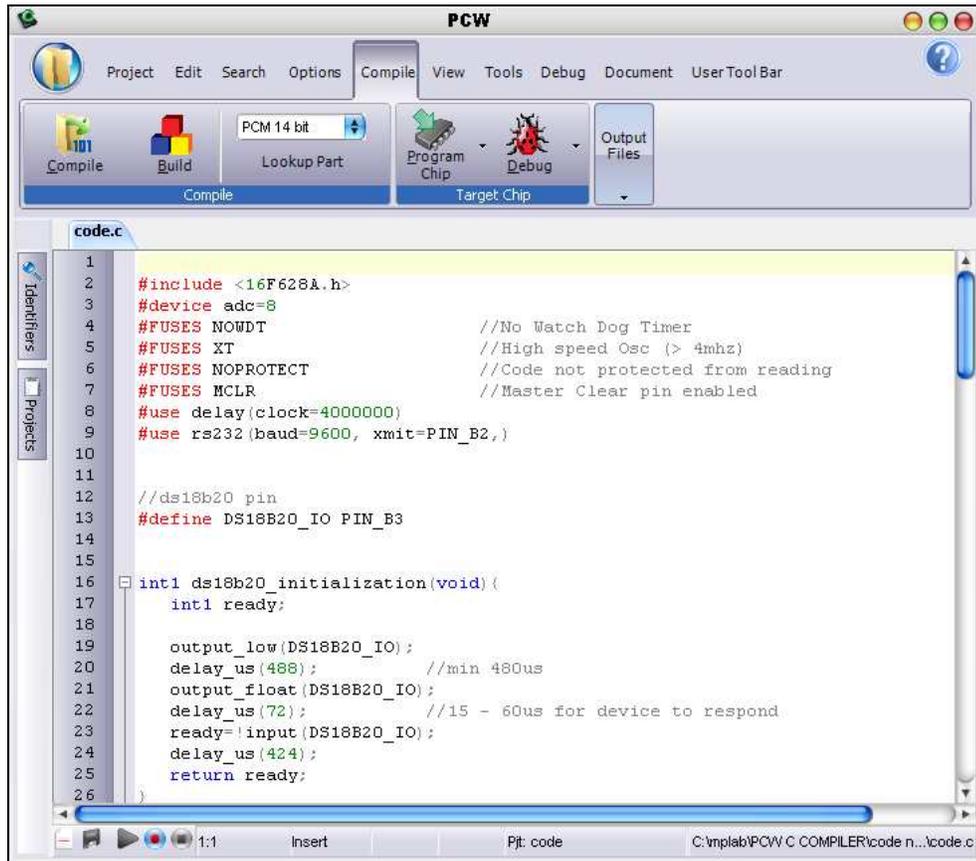
4.2 Compilador PCW de CCS

Para realizar la programación de los microcontroladores PIC en un lenguaje como el C, es preciso utilizar un compilador de C.

Dicho compilador nos genera ficheros en formato hexadecimal, que es el necesario para programar un microcontrolador de 6, 8, 18 ó 40 pines.

Uno de los compilador de C que actualmente es el PCW de la casa CCS Inc. A su vez, el compilador se integra en un entorno de desarrollo integrado (IDE) que permite desarrollar todas y cada una de las fases que se compone un proyecto, desde la edición hasta la compilación pasando por la depuración de errores. La última fase, a excepción de la depuración y retoques hardware finales, será programar el PIC.

Al igual que el compilador de Turbo C, éste "traduce" el código C del archivo fuente (*.c) a lenguaje máquina para los microcontroladores PIC, generando así un archivo en formato hexadecimal (*.hex). Además de éste, también genera otros seis ficheros.

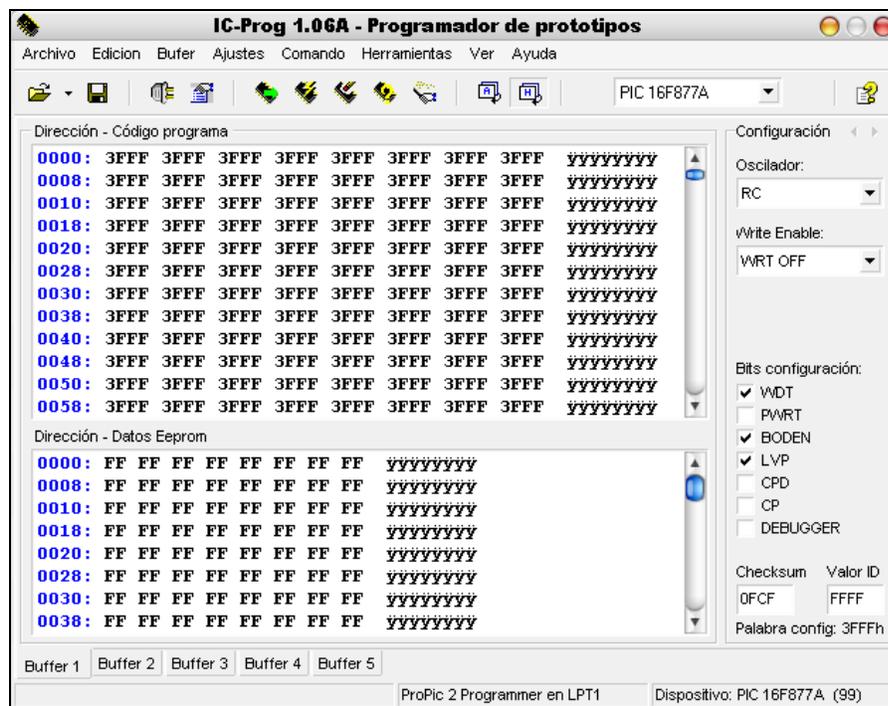


Img.4.- Entorno PCW-Compiler

4.3 ICprog

El IC-Prog es uno del software más popular para la grabación de microcontroladores PIC. Permite la programación de muchos dispositivos y esta probado con numerosos programadores, entre ellos ProPic 2 Programmer. Es de libre distribución en la pagina Web www.ic-prog.com se puede descargar y recoger toda la información de uso.

Una vez descargado, la instalación de este software es muy sencilla, basta con descomprimir el fichero icprog.zip y seguir el procedimiento usual en Windows. Este archivo consta del fichero icprog.exe, que contiene todo el código necesario para su funcionamiento para cualquier sistema operativo Windows. En caso de utilizar este software con Windows XP, 2000 o NT, es necesario descargar el archivo icprog.sys de la misma Web y situarlo en la misma carpeta, junto con el icprog.exe.



Img.5.- Entorno IC-Prog

4.4 Proteus

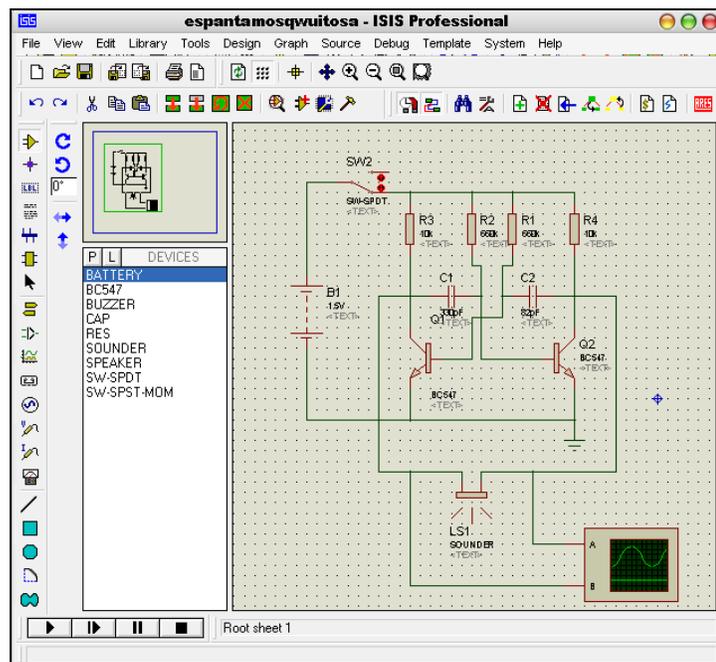
Es un entorno integrado diseñado para la realización completa de proyectos de construcción de equipos electrónicos en todas sus etapas: diseño, simulación, depuración y construcción. La suite se compone de cuatro elementos, perfectamente integrados entre sí:

4.4.1 ISIS

Es el corazón del entorno integrado Proteus. Es mucho más que un simple programa de dibujo de esquemas electrónicos. Combina un entorno de diseño de una potencia excepcional con una enorme capacidad de controlar la apariencia final de los dibujos.

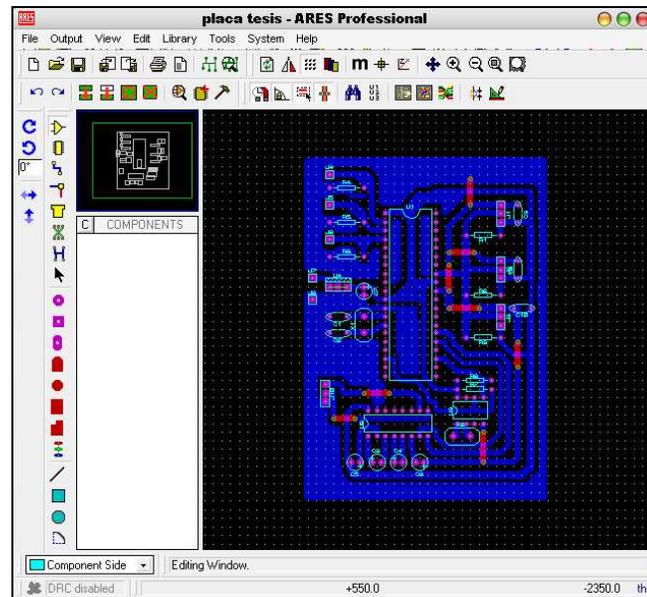
ISIS es la herramienta ideal para una rápida realización de complejos diseños de esquemas electrónicos destinados tanto a la construcción de equipos electrónicos como a la realización de tareas de simulación y prueba.

Además, ISIS es una herramienta excepcional para la realización de atractivos esquemas electrónicos destinados a su publicación en libros, manuales o documentos técnicos.



Img.6.- Entorno Proteus ISIS

4.4.2 ARES



Img.7.- Entorno Proteus ARES

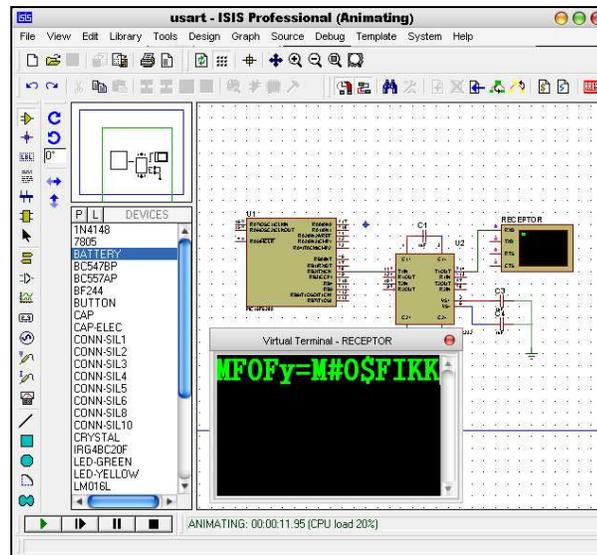
Es la herramienta de la suite Proteus dedicada al diseño de placas de circuito impreso (PCB). Está plenamente integrada con la herramienta ISIS.

Una vez diseñado en ISIS el esquema electrónico, se genera automáticamente la lista de redes (NETLIST). Una red es un grupo de pines interconectados entre sí y la lista de redes es una lista con todas las redes que forman nuestro diseño. ARES es capaz de recibir esta lista de redes para diseñar, a partir de ella, nuestra placa de circuito impreso. De esta forma nos aseguramos que nuestra placa tendrá unidos entre sí los pines de forma idéntica a como los hemos definido en nuestro esquema electrónico.

Cualquier modificación que realicemos en nuestro esquema, podrá ser reenviado desde ISIS a ARES donde aparecerán resaltados los cambios que se hayan producido. De esta forma la modificación y rediseño de nuestra placa se realizará de forma mucho más simple y segura.

Al igual que ISIS era capaz de enviar su listas de redes a programas de terceras partes, también ARES es capaz de importar listas de redes procedentes de otros programas que utilicen los formatos estándar Tango o Multiwire.

4.4.3 VSM

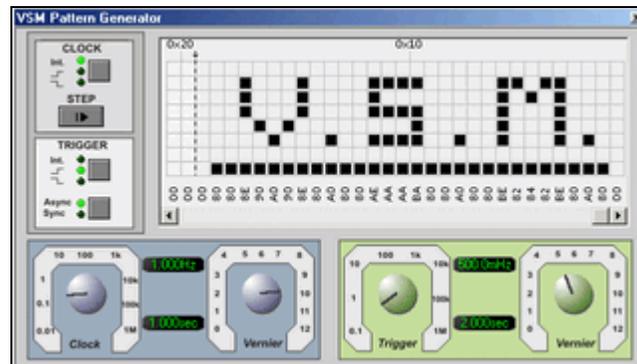


Img.8.- Entorno Proteus VSM

La suite **PROTEUS** incluye la herramienta **VSM** (Virtual System Modelling), un completo simulador para esquemas electrónicos que contienen microprocesador. **VSM** combina la simulación de circuitos electrónicos según el estándar **SPICE**, con modelos de animación de los componentes electrónicos y los microprocesadores que conforman el circuito.

De esta forma es posible, por primera vez, desarrollar y probar un diseño antes de que el prototipo físico sea construido.

4.4.4 ProSPICE



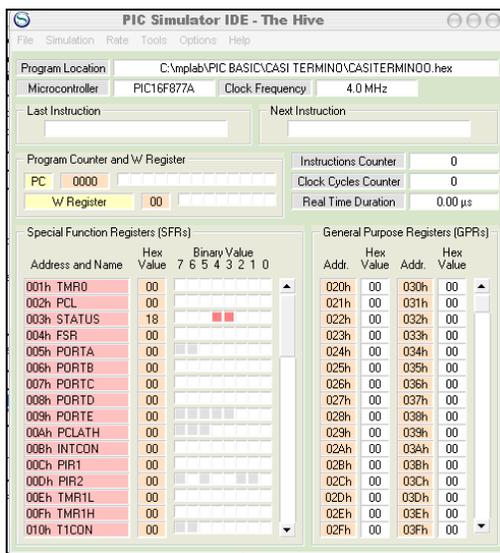
Img.9.- Entorno Proteus ProSPICE

A diferencia de los tradicionales simuladores de circuitos basados en gráficos, ProSPICE ofrece una completa animación interactiva de los circuitos. Es posible realizar pruebas del comportamiento de cada esquema electrónico de forma tan sencilla como hacer click con el ratón sobre componentes interactivos suministrados en librería (v.g. pulsadores, interruptores, potenciómetros, etc.). Y pueden observarse los resultados utilizando de forma completamente intuitiva los componentes interactivos suministrados también en librería (v.g. leds, displays de 7 segmentos, motores, multitesters, etc.).

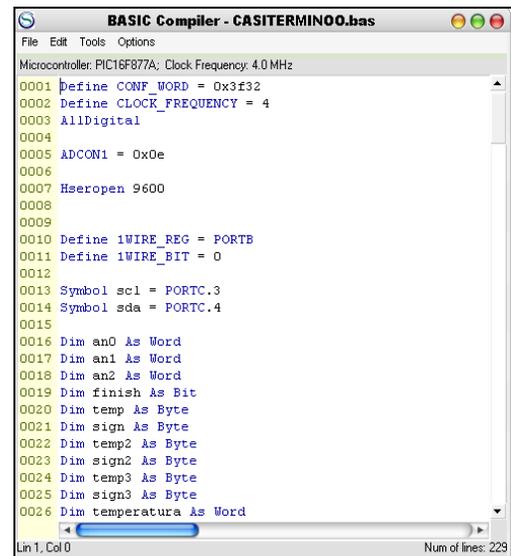
4.5 PIC Simulator IDE

Se trata de un entorno de programación integrado, que además de permitir la creación de programas en lenguaje assembler incorpora un muy completo BASIC, con soporte para dispositivos 1-WIRE, LCD, RS-232 y muchos más.

Otro aspecto destacable es la inclusión de una gran cantidad de módulos de E/S que simulan exactamente lo que esta pasando en nuestro programa, incluidos un modulo LCD, teclado, osciloscopio, displays de 7 segmentos entre otros. El autor del programa es Vladimir Soso, y la Web oficial del **PIC SIMULATOR IDE** es www.oshonsoft.com, desde donde puede bajarse una versión demo, mas que suficiente para realizar las practicas propuestas en este "curso". Además, el autor ha desarrollado entornos de programación similares para micros AVR, PIC18, PIC10F, Z80, etc.



Img.10.- Entorno De La Interfaz De Trabajo



Img.11.- Entorno De Programación

4.6 HyperTerminal



HyperTerminal es un programa que se puede utilizar para conectar con otros equipos, sitios Telnet, sistemas de boletines electrónicos (BBS), servicios en línea y equipos Host, mediante un módem, un cable de módem nulo o Ethernet.

Aunque utilizar HyperTerminal con un servicio de boletín electrónico para tener acceso a información de equipos remotos es una práctica que está dejando de ser habitual gracias al World Wide Web, HyperTerminal sigue siendo un medio útil para configurar y probar el módem o examinar la conexión con otros sitios.

HyperTerminal graba los mensajes enviados o recibidos por servicios o equipos situados al otro extremo de la conexión. Por esta razón, puede actuar como una valiosa herramienta para solucionar problemas de configuración y uso del módem. Para confirmar que el módem está bien conectado o ver su configuración, puede enviar comandos a través de HyperTerminal y ver los resultados. HyperTerminal ofrece la funcionalidad de desplazamiento, que le permite revisar el texto recibido que sobrepase el espacio de la pantalla.

HyperTerminal sirve también para transferir archivos grandes de un equipo a un equipo portátil a través del puerto serie, en lugar de realizar la configuración del portátil en una red.

Puede utilizar HyperTerminal para ayudar a depurar el código fuente desde un terminal remoto. También puede utilizar HyperTerminal para comunicarse con los equipos antiguos basados en caracteres.

HyperTerminal está diseñado para ser una herramienta fácil de utilizar y no viene a sustituir a otras herramientas principales disponibles en el mercado. HyperTerminal puede utilizarse para realizar las tareas específicas descritas, pero no debe intentar utilizarlo para necesidades de comunicación más complejas.

Capítulo V: Protocolos De Comunicación Analizados Y Aplicados

5.1 TIA/EIA RS-232 Recommended Standard 232C

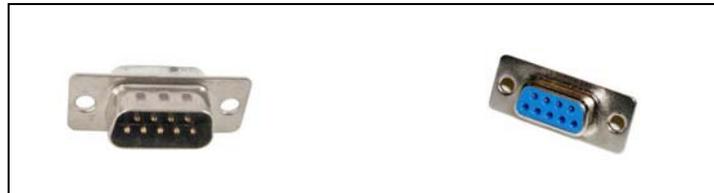


Fig.25.- DB9 Macho y Hembra

Es un estándar de comunicación para la transmisión de datos en serie entre equipos. La EIA (Electronics Industries Association) liberada de Data Terminal Equipment (DTE), por ejemplo el PC, a Data Communications Equipment (DCE), por ejemplo una impresora, para cubrir las conexiones eléctricas, mecánicas y funcionales entre terminales y equipos de comunicaciones. El estándar EIA RS-232 se introdujo en 1960. La transmisión de datos digital se hace en serie a través de una línea asimétrica, no terminada, entre dos equipos. La versión europea está bajo la especificación CCITT V.24. La distancia máxima de enlace está sobre los 15 metros y la velocidad de transmisión desde 20 kbps a 480 kbps.

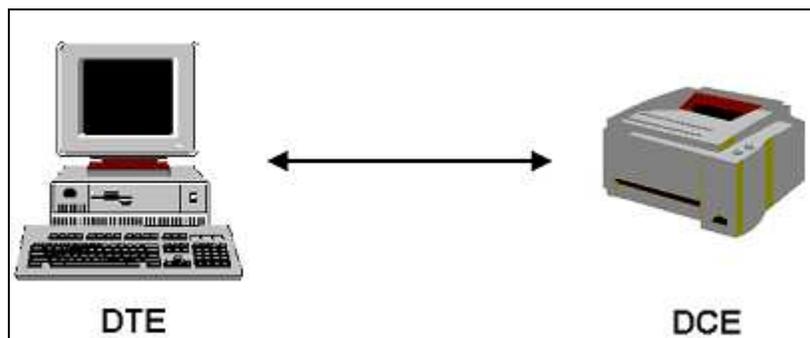


Fig.26.-

5.2 Protocolo One Wire

El fabricante americano Dallas Semiconductors ha diseñado una familia de periféricos muy atractiva para su uso con microcontroladores porque solo utiliza una línea para transferir datos denominada 1-Wire Bus o Bus de una línea.

Normalmente se conecta como en la figura:

- Un maestro que suele ser un microcontrolador que lleva el control del sistema.
- Uno o mas esclavos que proporcionan información y son gobernados por el Maestro.

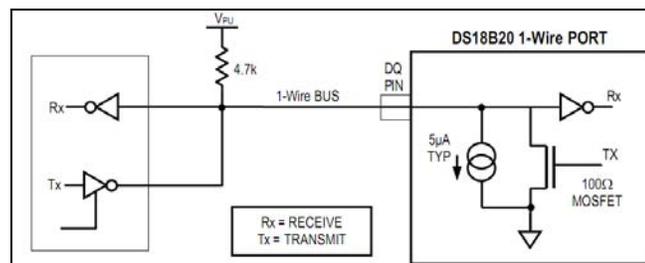


Fig.28.-

El bus de una línea requiere de una resistencia Pull-Up de 4,7 K Ω , de esta forma el estado de reposo del bus será el nivel alto.

Si el bus se mantiene a nivel bajo durante más de 480 μ s (microsegundos), se produce un reset de todos los dispositivos esclavos conectados a él.

5.2.1 Señales De Un Bus De Una Línea

Los dispositivos compatibles con 1-Wire Bus utilizan un estricto protocolo de comunicación con el fin de asegurar la integridad de los datos. Este protocolo define varias señales tipo:

- Pulso de Reset
- Pulso de Presence
- Escritura de nivel lógico '0'

- Escritura de nivel lógico '1'
- Lectura de nivel lógico '0'
- Lectura de nivel lógico '1'

Todas estas señales excepto el pulso de Presence, son inicializadas por el microcontrolador Maestro mediante un flanco de bajada y un nivel bajo de al menos 1µs.

Entre cada señal diferente debe transcurrir un tiempo mínimo de 1 µs. Durante el cual el bus debe mantenerse en alta impedancia, o que supone un nivel alto debido a la resistencia de Pull-Up.

Todos los datos y comandos son transmitidos con el bit de menor peso LSB (Bit menos Significativo).

5.2.2 Inicialización: Pulsos De Reset Y Presence

En el bus de una línea toda comunicación comienza con una secuencia de inicialización que consiste en un pulso de reset ejecutado por el microcontrolador, seguido por un pulso de Presence ejecutado por el dispositivo esclavo.

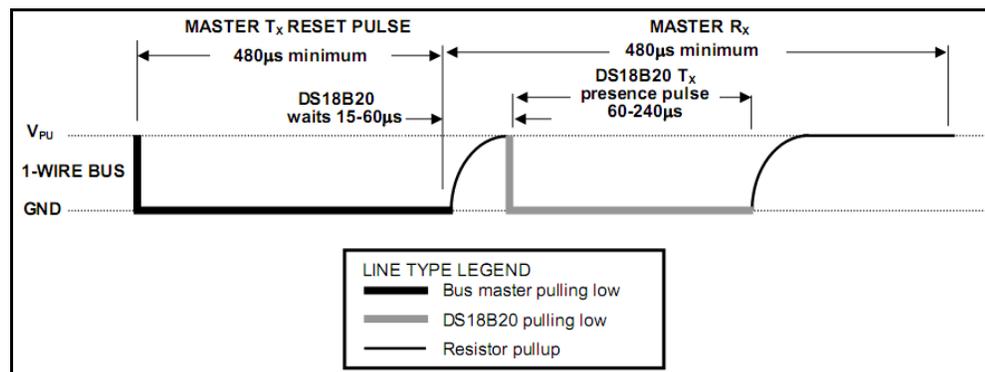


Fig.29.- Pulso De Reset Y Presence

Cuando un esclavo envía el pulso de presence en respuesta al del Reset le indica al microcontrolador que esta preparado para trabajar.

Durante la secuencia de iniciación, se produce el siguiente protocolo:

1. El microcontrolador maestro transmite un pulso de reset que mantiene un bajo en la línea durante al menos 480 μ s.
2. A continuación el maestro para a modo entrada y, por lo tanto, pone al bus en un nivel alto gracias a las resistencias de Pull-Up.
3. Cuando el esclavo detecta esto, espera 15 a 60 μ s entonces transmite el pulso de presence, que consiste en poner la línea a nivel bajo durante un tiempo de entre 60 a 240 μ s.
4. A continuación, el esclavo deja el bus en alta impedancia y espera comandos u ordenes del microcontrolador.

5.3 Protocolo I2C (Inter Integrated Circuit Bus)

El I2C es un bus serie desarrollado por Philips Semiconductors y que es ampliamente utilizado en la industria electrónica.

Este bus esta formado por dos hilos que puede conectar varios dispositivos mediante un hardware muy simple, como se muestra en la figura. Por esos dos hilos se produce la comunicación serie, bit a bit. Se transmite dos señales una por cada línea.

- **SCL**, (Serial Clock) es la señal de reloj que se utiliza para la sincronización de los datos.
- **SDA**, (Serial Data). Es la línea para la transferencia de los datos.

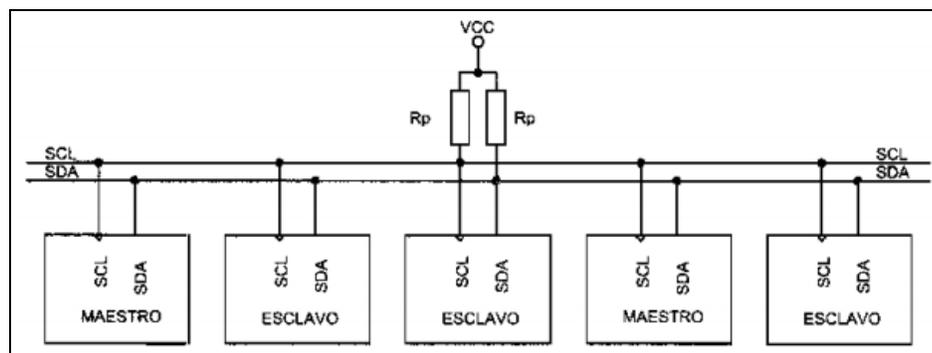


Fig.30.- Estructura De Bus I2C

5.3.1 Condiciones de START y STOP

Antes de que se establezca un intercambio de datos entre el circuito Master y los Esclavos, el Master debe informar el comienzo de la comunicación (condición de Start) la línea SDA cae a cero mientras SCL permanece en nivel alto. A partir de este momento comienza la transferencia de datos. Una vez finalizada la comunicación se debe informar de esta situación (condición de Stop). La línea SDA pasa a nivel alto mientras SCL permanece en estado alto.

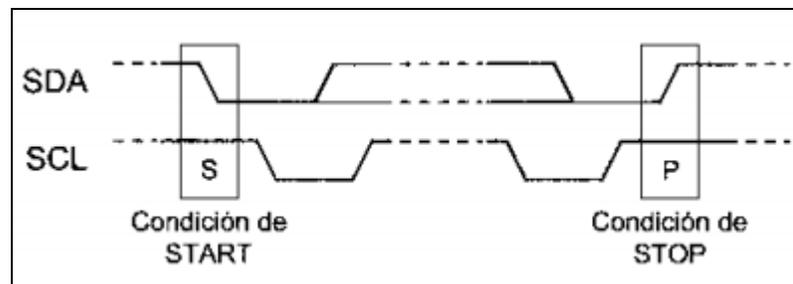


Fig.31.- Condiciones de START y STOP

Capítulo VI: Diseño, Construcción Y Pruebas Del Prototipo

Los requerimientos de este trabajo es medir temperatura y salinidad de la mezcla entre agua dulce y agua de mar, las cuales se deben realizar en tres estaciones específicas del equipo. Este equipo es un estanque semicilíndrico de fibra de vidrio en forma de canoa de 2 metros de largo por 0.3 metros de diámetro, este tamaño es justo y necesario para realizar las pruebas que se requieran basados en los parámetros que se han de medir.

Para el llenado de la canoa se tiene dos estanques de agua sobre este equipo, uno de ellos se mantiene con agua de mar y el otro se mantiene con agua dulce, siendo estos los que alimentan de líquido a cada lado respectivamente, para así formar la mezcla salina-dulce, simulando el encuentro entre el agua dulce del río y el agua salada del mar en el momento de desembocar en este último. Para el vaciado tiene salidas de agua en ambos costados de la canoa, creándose así una circulación permanente de mezcla líquida.

Este equipo está instalado en los laboratorios de Biología Marina en la localidad de Calfuco, estos laboratorios cuentan con un sistema de aguas circulantes, lo que ayuda a renovar constantemente el agua que egresa de los estanques, esto se aprovechará para el llenado y vaciado de las aguas.

6.1 Etapa de entrada

Para la toma de muestras de datos y a través de lo estudiado en los capítulos anteriores, y los requerimientos necesitados, como son la medición de temperatura y muestra de salinidad en un estanque, se considero utilizar el dispositivo DS18B20 para la medición de temperatura. Por otro lado, para obtener el valor salinidad se construyo sondas basadas en el principio de conductividad eléctrica.

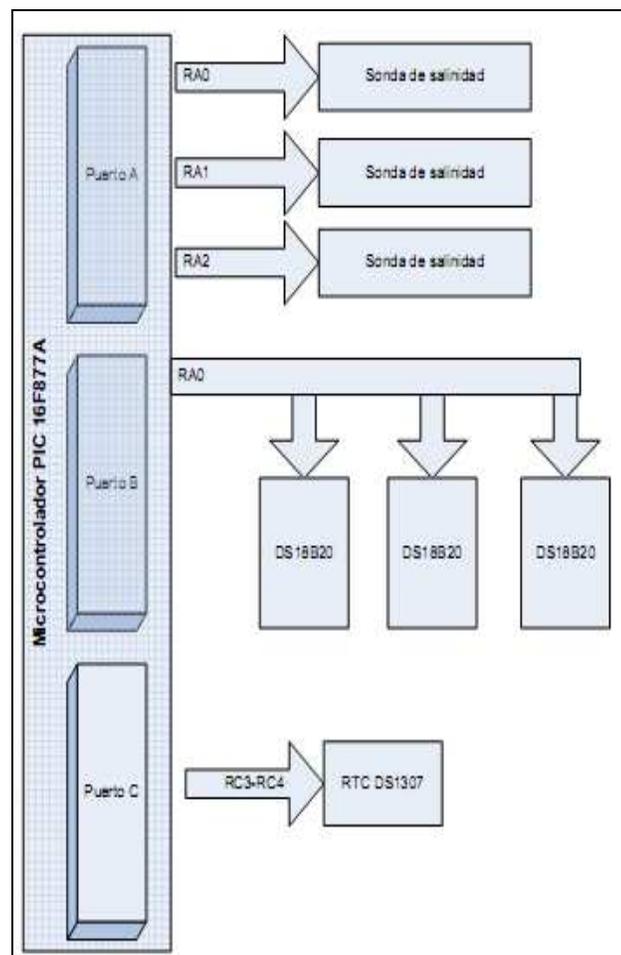


Fig.32.- Etapa de entrada

6.1.1 Sensor de temperatura DS18B20

Se ha escogido realizar las mediciones con este tipo de sensor ya que trabaja bajo el protocolo 1-wire. El que trabaje bajo este protocolo nos da la ventaja que podemos colocar los tres sensores en la misma línea ahorrándonos hardware. Además estos dispositivos cuentan con un número de serie único para cada uno, facilitando el proceso de adquisición de datos ya que a traves de ellos y comandos de programación se les indica lo que deben realizar, en un momento dado.

En la figura 33, que esta realizada con el Software Proteus en su apartado ISIS, observamos la configuración eléctrica de estos sensores dispuestos cada uno en la misma línea conectadas al pin RB0 del microcontrolador PIC 16F877A, también se observa una resistencia de 4.7 K Ω entre alimentación y DQ (pin donde se transmite el intercambio de información), esta resistencia es recomendada por el fabricante para estabilizar los valores de corriente en la escritura y lectura de datos, al momento de intercambiar información.

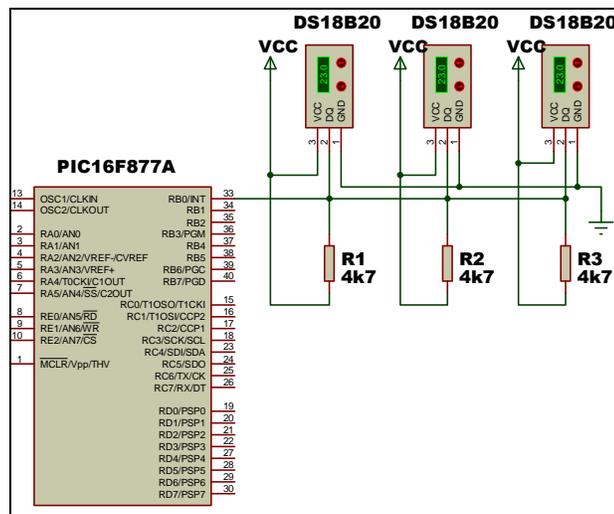


Fig.33.- Conexión eléctrica DS18B20

6.1.2 Sensor de Salinidad

A través del tiempo de estudio para este trabajo se observó que el costo de estos sensores eran algo altos, esta fue la razón por la cual se decidió diseñar y construir estas sondas. Bajo el principio de conductividad eléctrica, vista en el capítulo 2, se logró diseñar una sonda capaz de medir la resistividad eléctrica, ésta sonda está construida con dos tornillos de acero inoxidable y soldados a un cable RG-58, con una separación entre tornillos de 1 cm. aproximadamente y encapsulados dentro de un tubo de pvc. Para la conexión al equipo y posterior toma de dato se colocó un conector tipo "F" en cada una de las sondas.

En la figura 34 podemos ver la configuración eléctrica entre las sondas y el microcontrolador. Aquí podemos observar que cada sonda está conectada al puerto A en un pin independiente del microcontrolador PIC 16F877A, esto se debe a que los datos que tomaremos como salinidad corresponden a un valor analógico de tensión (diferencia de voltaje) el cual es cuantificado con el módulo analógico digital del microcontrolador y así luego obtendremos un valor correspondiente a la salinidad medida en cada estación.

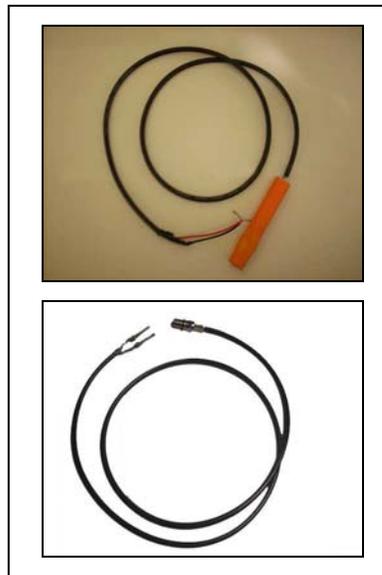


Fig.33.-Sensor de Salinidad

6.1.3 Reloj de Tiempo Real DS1307

Este dispositivo es un reloj calendario de la empresa Dallas Semiconductors muy útil y fácil de utilizar, ya que a través de su protocolo de comunicación I2C se puede conectar a cualquier tipo de microcontrolador.

Utilizamos el DS1307 para poder identificar en que lapso de tiempo fue realizada la toma de muestra tanto la de temperatura como la de salinidad. El DS1307 comienza desde cero segundos cada momento que se encienda el equipo, así mismo si se resetea el circuito el DS1307 volverá contabilizar desde cero.

En la figura 35 que vemos en este apartado, se encuentra el Pinout o patillaje del circuito integrado, y a continuación el conexionado eléctrico con el microcontrolador. En ambos se ven los pines X1 y X2 estos están destinados para colocar un cristal con una frecuencia de resonancia de 32.768 Khz, siendo este el que nos entrega un espacio de tiempo de un segundo.

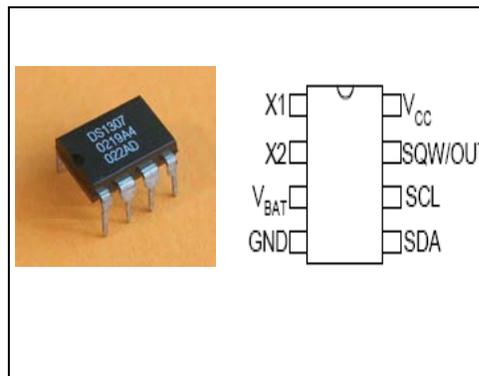


Fig.35.- Pinout DS1307

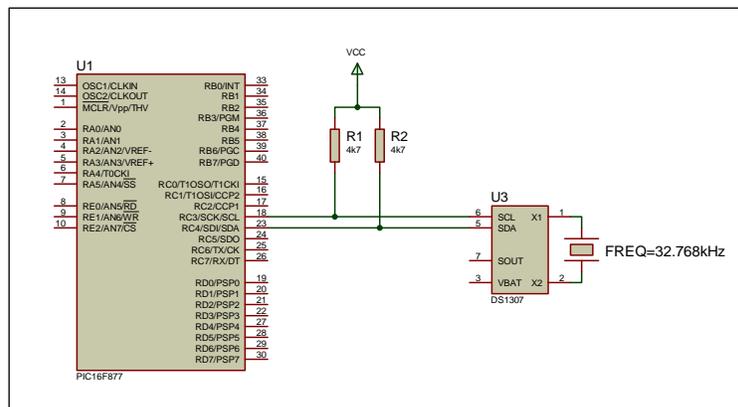


Fig.36.- Conexión DS1307

6.2 Etapa de proceso

En esta etapa se realizo en base al microprocesador PIC16F877A, con este potente circuito integrado nos da la facilidad de utilizar diferentes periféricos tanto analógicos como digitales, convirtiéndose en el cerebro de todo el equipo.

Comienza con la entrada de datos desde los sensores a cada unidad interna del microcontrolador. Este muestreo lo realiza en forma sistemática, es decir, primero toma los valores de temperatura consultando a cada DS18B20 por separado, a través del número de serie que tienen interno cada uno de estos sensores, luego consulta el valor independiente de cada uno de los conversores analógico-digital en los cuales tiene el valor correspondiente a la salinidad del agua.

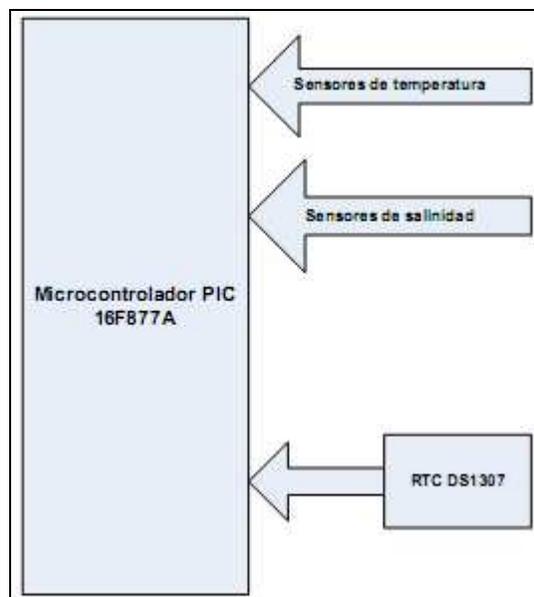


Fig.37.- Etapa de proceso

Para lograr realizar lo anteriormente expuesto, se debe crear un software específico, con los correspondientes requisitos.

6.2.1 Software

Para la realización del software, primero se debe señalar que los microcontroladores solo reconocen el lenguaje de maquina, es decir ceros y unos, lo que hace casi imposible poder crear un programa en esas condiciones.

Para lograr crear un programa que realizara la toma de muestras y su posterior ordenamiento se utilizo variados software de programación específicos para microcontroladores, es así como se ocupo lenguaje C a través del programa Compilador PCW, lenguaje Assembler con el compilador MpLab y finalmente lenguaje Basic con el Software PIC Simulator IDE, siendo este último el utilizado para compilar el programa completo para la toma de datos. Para cualquiera de los casos presentados anteriormente presentados, se crean ficheros con archivos específicos con el leguaje de maquina para el microcontrolador.

Durante el proceso primero se comenzó trabajando con Mplab, através de este compilador se desarrollo un programa capas de utilizar el modulo ADC (Conversor Análogo Digital) del microcontrolador PIC16F877A, esta parte es utilizada para las sondas de salinidad.

Para una mejor comprensión se muestra el programa realizado en lenguaje assembler.

```

list p=16f877A
include "p16f877A.inc"
__config 3f31
RADIX HEX
cblock 0x20
PDe10
PDe11
PDe12
PDe13
endc
org 0x0000
inic call    initrans ;inicializa el puerto serie para transmisión
call    adc
BSF    TRISA,0      ;configura como entrada el canal digital RA0
BSF    TRISA,1      ;configura como entrada el canal digital RA1
BSF    TRISA,3      ;configura como entrada el canal digital RA3
BCF    STATUS,RP0   ;Banco 0

```

```

conver      movlw  b'10000001'      ;Selecciona el canal AN0, reloj de conversión Fosc/2
              movwf  ADCON0          ;y enciende el convertidor
              call   ciclo
              call   DEMORA
              movlw  b'10001001'    ;Selecciona el canal AN1, reloj de conversión Fosc/2
              movwf  ADCON0          ;y enciende el convertidor
              call   ciclo
              call   DEMORA
              movlw  b'10011001'    ;Selecciona el canal AN3, reloj de conversión Fosc/2
              movwf  ADCON0          ;y enciende el convertidor
              call   ciclo
              call   DEMORA
              movlw  0x0D
              call   envia
              movlw  0x0a
              call   envia
              goto   conver

```

Este programa fue probado a través del software Proteus en su modulo ISIS, el cual se muestra a continuación.

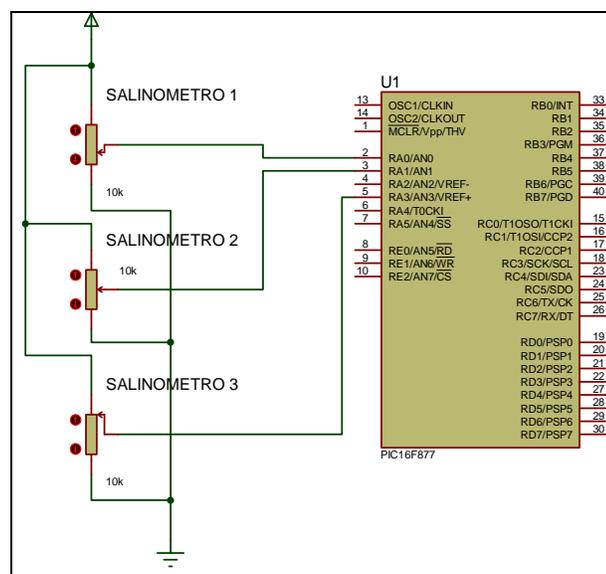


Fig.38.- Simulación sensores de salinidad

El compilador C es utilizado para este proyecto en forma específica, gracias a que éste software contiene librerías diseñadas especialmente para ahorrar tiempos de programación, y dentro de esas librerías se encuentra uno para dispositivos 1-wire, y con algunas modificaciones se logra obtener el siguiente código.

```

#include <16F628A.h>
#define adc=8
#define FUSES NOWDT
#define FUSES XT
#define FUSES NOPROTECT
#define FUSES MCLR
#define delay(clock=4000000)
#define rs232(baud=9600, xmit=PIN_B2,)
//ds18b20 pin
#define DS18B20_IO PIN_B3
int1 ds18b20_initialization(void){
    int1 ready;
    output_low(DS18B20_IO);
    delay_us(488);
    output_float(DS18B20_IO);
    delay_us(72);
    ready=input(DS18B20_IO);
    delay_us(424);
    return ready;
}
void ds18b20_write_bit(int8 b){
    output_low(DS18B20_IO);
    if(b == 1){
        delay_us(2);
        output_float(DS18B20_IO);
    }
    delay_us(125);
    output_float(DS18B20_IO);
}
void ds18b20_write_byte(int8 B){
    int8 i, aux;
    for(i=0; i<8; i++){
        aux = B >> i;
        aux &= 0x01;
        ds18b20_write_bit(aux);
    }
}
int1 ds18b20_read_bit(void){
    output_low(DS18B20_IO);
    delay_us(2);
    output_float(DS18B20_IO);
    delay_us(8);
    return(input(DS18B20_IO));
}
int8 ds18b20_read_byte(void){
    int8 i, result=0x00;
    for(i=0; i<8; i++){
        if(ds18b20_read_bit())
            result |= (0x01 << i);
        delay_us(125);
    }
    return result;
}
void main() {
    int8 i;
    int8 scratchpad_data[9];
    ds18b20_write_byte(0xCC);
    ds18b20_write_byte(0x4E);
    if(ds18b20_initialization()){
        ds18b20_write_byte(0x33);
        delay_ms(800);
        /*while (busy == 0){
            printf("waiting...\n");
            busy = ds18b20_read_byte();
        }
        busy=0;*/
        for(i=0; i<8; i++){
            scratchpad_data[i]=ds18b20_read_byte();
            printf("scratchpad_data[%d]=%X\r\n", i,
            scratchpad_data[i]);
            delay_ms(5000);
        }
    }
}

```

En la figura siguiente muestra el esquema circuital necesario para obtener el número de serie del sensor DS18B20, previamente cargando el código compilado a lenguaje de maquina.

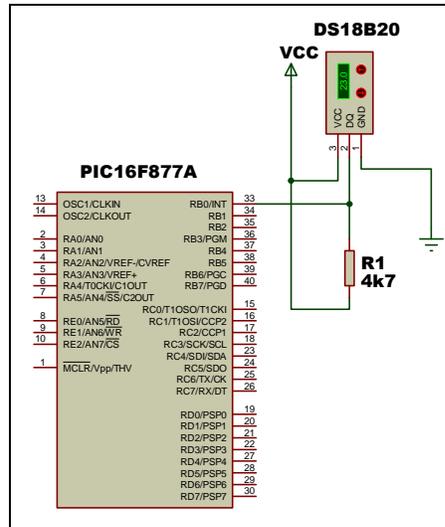


Fig.39.- Circuito obtención de N° de serie

El resultado de este trabajo previo nos otorga la siguiente información vital para la conformación final del programa. Estos valores son mostrados en código hexadecimal, en donde el primer valor corresponde a la familia de dispositivo, los siguientes seis es el nombre y el ultimo corresponde al CRC (código de redundancia cíclica).

- *Sensor DS18B20 numero de serie 1:* 28h, 74h, 4ch, 5Eh, 01h, 00h, 00h, F4h,
- *Sensor DS18B20 numero de serie 2:* 28h, 29h, 46h, 5Eh, 01h, 00h, 00h, 2dh,
- *Sensor DS18B20 numero de serie 3:* 28h, C2h, 4ah, 5Eh, 01h, 00h, 00h, DDh,

Finalmente para la conformación del programa final se utilizo el compilador PIC Simulator IDE. Este software trabaja en base a diferentes lenguajes de programación, eligiéndose el lenguaje BASIC, con esto se logra crear un programa sencillo y entendible.

PIC Simulator IDE contiene librerías especializadas para determinadas tareas, es así que encontramos librerías para el ADC del microcontrolador, otra para la comunicación 1-wire y otro para la comunicación I2C, facilitando el exceso de código y ahorro de memoria del PIC 16F877A.

Por último, el software diseñado es que a continuación se presenta,

```

Define CONF_WORD = 0x3F32
Define CLOCK_FREQUENCY = 4
AllDigital
ADCON1 = 0x0e
Hseropen 9600
Define 1WIRE_REG = PORTB
Define 1WIRE_BIT = 0
Symbol scl = PORTC.3
Symbol sda = PORTC.4
Dim an0 As Word
Dim an1 As Word
Dim an2 As Word
Dim finish As Bit
Dim temp As Byte
Dim sign As Byte
Dim temp2 As Byte
Dim sign2 As Byte
Dim temp3 As Byte
Dim sign3 As Byte
Dim temperatura As Word
Dim temperatura2 As Word
Dim temperatura3 As Word
Dim a As Word
Dim b As Word
Dim t As Word
Dim fraccion As Word
Dim fraccion2 As Word
Dim fraccion3 As Word
Dim addr As Byte
Dim sec As Byte
Dim min As Byte
Dim hour As Byte
Dim day As Byte
Dim date As Byte
Dim month As Byte
Dim year As Byte

Dim d_hi As Byte
Dim d_lo As Byte
Hserout "Toma de muestras", 0x0d, 0x0a
Hserout "DIA.; TIEMPO; SALINIDAD 1;
SALINIDAD 2; SALINIDAD 3;
TEMPERATURA 1; TEMPERATURA 2;
TEMPERATURA3", 0x0d, 0x0a
I2CWrite sda, scl, %11010000, 0x00, 00
I2CWrite sda, scl, %11010000, 0x01, 00
I2CWrite sda, scl, %11010000, 0x02, 00
I2CWrite sda, scl, %11010000, 0x03, 01
I2CWrite sda, scl, %11010000, 0x04, 01
I2CWrite sda, scl, %11010000, 0x05, 03
I2CWrite sda, scl, %11010000, 0x06, 09
loop:
I2CRead sda, scl, %11010000, 0x00, sec
I2CRead sda, scl, %11010000, 0x01, min
I2CRead sda, scl, %11010000, 0x02, hour
I2CRead sda, scl, %11010000, 0x03, day
I2CRead sda, scl, %11010000, 0x04, date
I2CRead sda, scl, %11010000, 0x05,
month
I2CRead sda, scl, %11010000, 0x06, year
Adcin 0, an0
Adcin 1, an1
Adcin 2, an2
WaitMs 250
1wireInit
1wireSendByte 0xcc, 0x44
WaitMs 1
loop2:
1wireGetBit finish
If finish = 0 Then Goto loop2
1wireInit
sensor1:
1wireInit

```

```

        1wireSendByte 0x55, 0x28, 0x74, 0x4c,
0x5e, 0x01, 0x00, 0x00, 0xf4, 0xbe
        Goto toma1
sensor2:
        1wireInit
        1wireSendByte 0x55, 0x28, 0x29, 0x46,
0x5e, 0x01, 0x00, 0x00, 0x2d, 0xbe
        Goto toma2
sensor3:
        1wireInit
        1wireSendByte 0x55, 0x28, 0xc2, 0x4a,
0x5e, 0x01, 0x00, 0x00, 0xdd, 0xbe
        Goto toma3
toma1:
        1wireGetByte temp, sign
        a.HB = sign
        a.LB = temp
        temperatura = ShiftRight(a, 4)
        temperatura = temperatura And 0x00ff
        If temp.0 = 1 Then
                fraccion = 625
        Else
                fraccion = 0
        Endif
        If temp.1 = 1 Then
                fraccion = fraccion + 1250
        Else
                fraccion = 0
        Endif
        If temp.2 = 1 Then
                fraccion = fraccion + 2500
        Else
                fraccion = 0
        Endif
        If temp.3 = 1 Then
                fraccion = fraccion + 5000
        Endif
        1wireSendByte 0x55, 0x28, 0x74, 0x4c,
0x5e, 0x01, 0x00, 0x00, 0xf4, 0xbe
        Goto toma1
        WaitMs 250
        Goto sensor2
toma2: 1wireGetByte temp2, sign2
        b.HB = sign2
        b.LB = temp2
        temperatura2 = ShiftRight(b, 4)
        temperatura2 = temperatura2 And 0x00ff
        If temp2.0 = 1 Then
                fraccion2 = 625
        Else
                fraccion2 = 0
        Endif
        If temp2.1 = 1 Then
                fraccion2 = fraccion2 + 1250
        Else
                fraccion2 = 0
        Endif
        If temp2.2 = 1 Then
                fraccion2 = fraccion2 + 2500
        Else
                fraccion2 = 0
        Endif
        If temp2.3 = 1 Then
                fraccion2 = fraccion2 + 5000
        Else
                fraccion2 = 0
        Endif
        WaitMs 250
        Goto sensor3
toma3: 1wireGetByte temp3, sign3
        t.HB = sign3
        t.LB = temp3
        temperatura3 = ShiftRight(t, 4)
        temperatura3 = temperatura3 And 0x00ff
        fraccion = 0
        Endif
        WaitMs 250
        Goto sensor2
toma2: 1wireGetByte temp2, sign2
        b.HB = sign2
        b.LB = temp2
        temperatura2 = ShiftRight(b, 4)
        temperatura2 = temperatura2 And 0x00ff
        If temp2.0 = 1 Then
                fraccion2 = 625
        Else
                fraccion2 = 0
        Endif
        If temp2.1 = 1 Then
                fraccion2 = fraccion2 + 1250
        Else
                fraccion2 = 0
        Endif
        If temp2.2 = 1 Then
                fraccion2 = fraccion2 + 2500
        Else
                fraccion2 = 0
        Endif
        If temp2.3 = 1 Then
                fraccion2 = fraccion2 + 5000
        Else
                fraccion2 = 0
        Endif
        WaitMs 250
        Goto sensor3
toma3: 1wireGetByte temp3, sign3
        t.HB = sign3
        t.LB = temp3
        temperatura3 = ShiftRight(t, 4)
        temperatura3 = temperatura3 And 0x00ff

```

Al analizar este código se puede dar cuenta de la estructura lógica que se a tomado para la obtención final de un software capaz de realizar el muestreo de seis sensores en forma correlativa, en donde primero se toman el tiempo desde el RTC (Reloj de tiempo Real), Luego se toman los valores de Salinidad a través del modulo ADC y finalmente Se consulta cada uno de los sensores DS18B20.

Todos estos datos son transmitidos posteriormente a través del modulo USART del microcontrolador PIC 16F877A.

A continuación podemos ver un diagrama en boques de la línea lógica que se ha tomado para la conformación del software.

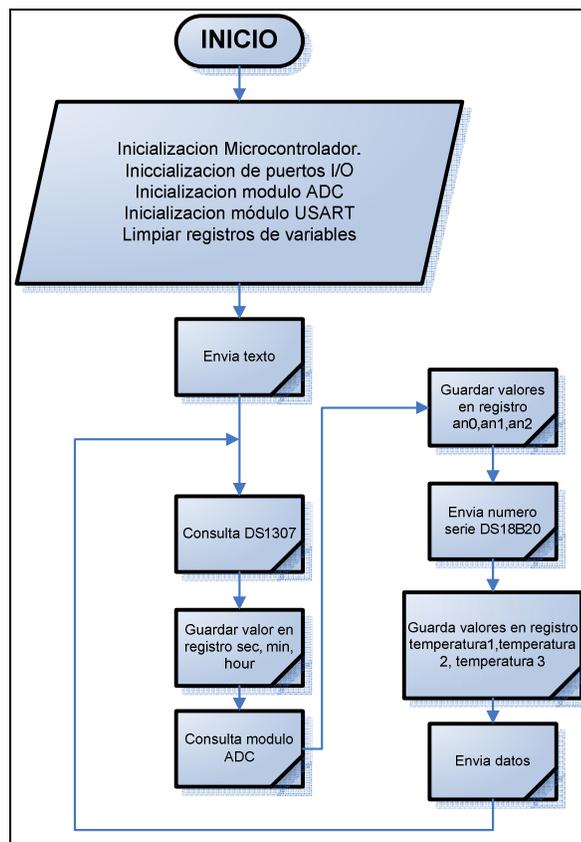


Fig.40.- Diagrama de flujo de software del Proyecto

6.3 Etapa de Salida

Una vez que los datos se encuentran en la memoria del microcontrolador, es necesario visualizarlos, para ellos se transmite esta información en forma serial asíncrona con el módulo USART, previamente configurado vía software.

La visualización final se realiza a través del puerto serial DB-9 de un computador personal, y se puede observar con el software Hipertérmino. Este software requiere de un proceso intermedio el cual se muestra a continuación, este proceso asegura la visualización de los datos.

Primero se debe abrir una nueva sesión a través de la ruta inicio/todos los programas/accesorios/comunicaciones/hipertérmino. Es así como veremos la siguiente imagen en donde ingresaremos un nombre, para nuestro caso "Monitoreo".



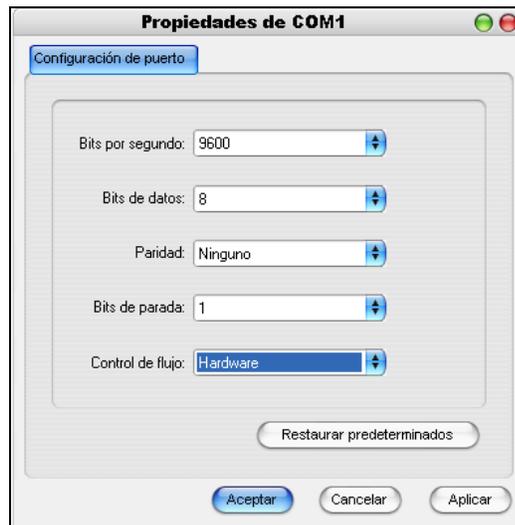
Img. 12

El siguiente paso es elegir que en que puerto se ha conectado el equipo, estos llevan el nombre de COM1, COM2, COM3, ETC.



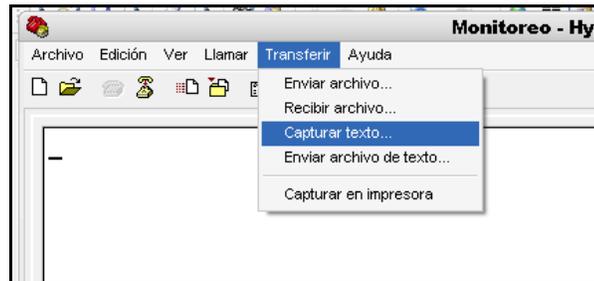
Img. 13

El paso siguiente es configurar los datos necesarios para establecer la comunicación estos se muestran en la imagen, esto nos asegura una correcta visualización de los datos.



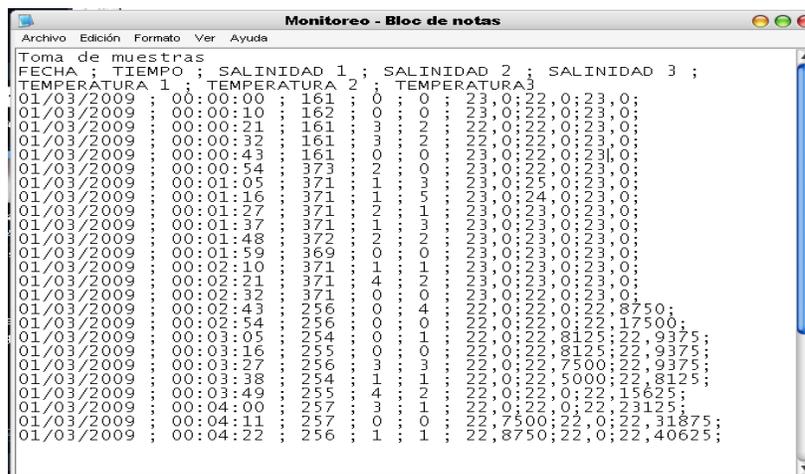
Img.14

Por ultimo activamos la opción de capturar texto, esta opción es muy importante debido a que se pueden guardar los valores sensados en un archivo de texto con extensión *.txt.



Img.15

Este archivo puede luego ser manejado a traves del software Microsoft Excel para el análisis estadístico y grafico del estado salínico y de temperatura de equipos construido.



Img.16

En la imagen anterior podemos ver el resultado de la captura de datos estructurado la información necesaria para poder realizar un último tratamiento en una planilla Excel. Como en el ejemplo que se muestra en la tabla siguiente con sus correspondientes gráficas.

FECHA	TIEMPO	SALINIDAD 1	SALINIDAD 2	SALINIDAD 3	TEMPERATURA 1	TEMPERATURA 2	TEMPERATURA3
01/03/2009	00:00:00	161	150	140	23	22	23
01/03/2009	00:00:10	162	150	140	23	22	23
01/03/2009	00:00:21	161	153	142	22	22	23
01/03/2009	00:00:32	161	153	132	22	22	23
01/03/2009	00:00:43	161	140	130	23	22	23
01/03/2009	00:00:54	373	142	130	23	22	23
01/03/2009	00:01:05	371	141	153	23	25	23
01/03/2009	00:01:16	371	141	155	23	24	23
01/03/2009	00:01:27	371	122	151	23	23	23
01/03/2009	00:01:37	371	121	163	23	23	23
01/03/2009	00:01:48	372	212	152	23	23	23
01/03/2009	00:01:59	369	210	200	23	23	23
01/03/2009	00:02:10	371	211	201	23	23	23
01/03/2009	00:02:21	371	224	202	23	23	23
01/03/2009	00:02:32	371	200	220	23	22	23
01/03/2009	00:02:43	256	200	224	22	22	22,875
01/03/2009	00:02:54	256	200	224	22	22	22,175
01/03/2009	00:03:05	254	200	211	22	22,8125	22,9375
01/03/2009	00:03:16	255	160	120	22	22,8125	22,9375
01/03/2009	00:03:27	256	163	123	22	22,75	22,9375
01/03/2009	00:03:38	254	161	110	22	22,5	22,8125
01/03/2009	00:03:49	255	164	112	22	22	22,15625
01/03/2009	00:04:00	257	163	100	22	22	22,23125
01/03/2009	00:04:11	257	150	100	22,75	22	22,31875
01/03/2009	00:04:22	256	151	101	22,875	22	22,40625

Tabla.18.- Distribución de datos en Plantillas Excel

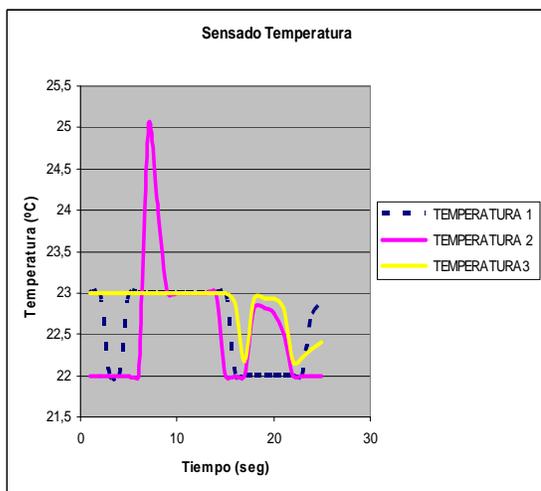


Fig.41.- Grafica final de temperatura

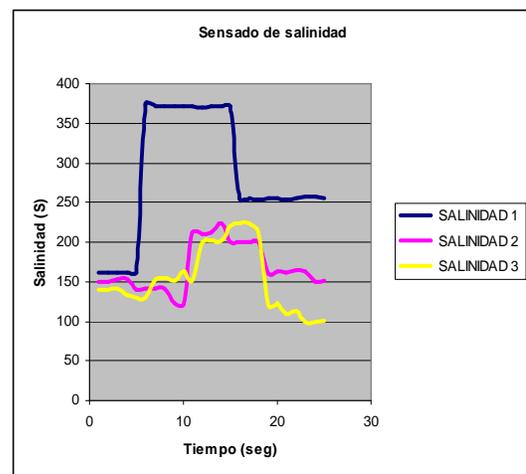


Fig.42.- Grafica final de salinidad

6.4 Proyección Y Mejoramiento

El mejoramiento progresivo de este proyecto también debe ser considerado, es así como a continuación se mostrara las posible complementación, haciendo de este un equipo mucho más completo y con mayor proyección .

➤ Integración Protocolo USB

Siguiendo la línea comercial, este equipo puede ser complementado con un puerto USB para facilitar la obtención de la información sensada, además, la información sería guardada en una memoria en un archivo con extensión *.TXT para facilitar el transporte en dispositivos de almacenamiento rápido. A través de esta utilidad sería posible con solo conectar un dispositivo de almacenamiento, como por ejemplo un Pendrive, y almacenar la información guardada.

➤ Transmisión y recepción inalámbrica

La aplicación de tarjetas inalámbricas es una muy buena solución si se quiere tener control a distancia, es así como este equipo puede ser ampliado a este recurso ya que estas tarjetas contienen un modulo especializado para la comunicación USART del microcontrolador y un modulador / demodulador RF para el enlace inalámbrico.

➤ Sensores adicionales

Realizar mediciones de otros parámetros físicos como Oxígeno disuelto en el agua y PH, sería de gran aporte para un mejor estudio de un ambiente real. Medir estas dos variables incluida la temperatura y salinidad, llevaría a este simulador a un nuevo nivel de experiencia para laboratorio, convirtiéndose en un equipo con utilidades muy variadas, con el que no solo se podría simular una formación estuarinas, sino que también otros ambientes marinos.

Conclusiones

Realizar una investigación previa es de mucha importancia para los diseñadores de equipos ya que esto prestara una visualización especifica a lo requerimientos que el usuario final necesita. A través de esta investigación, además, se alcanzan nuevos conocimientos, lo que es un aporte al crecimiento intelectual y profesional.

Realizar pruebas de distintos dispositivos, mejora la opinión a la hora de elegir un dispositivo u otro. Bajo esta idea se probaron diferentes tipos de microcontroladores PIC, los cuales aportaron a crear programas afines de cada situación que se presentaba durante el proceso de investigación.

Debido a los avances en la electrónica analógica y digital, el mercado surte con una gran variedad de dispositivos muy útiles que son capaces de dar solución inmediata a problemas tanto cotidianos como específicos, es así como un reloj calendario o un sensor de temperatura, están al alcance de cualquier diseñador aficionado o profesional en el desarrollo de equipos. También su bajo coste hace que sean utilizados en forma masiva.

Bibliografía

- ARCOS, Dagoberto. Proyecto FIP 2000-09. *Determinación de la capacidad de carga de las zonas estuarinas de los Ríos Valdivia y Bueno X Región (Río Valdivia)* Talcahuano, Chile. INPESCA, (Instituto de investigación pesquera). 2002.
- PINO, Mario. PERILLO, Gerardo. SANTAMARINA, Pablo. *Residual fluxes in a cross-section of the Valdivia river estuary*. Chile, Instituto de Geociencias de la Universidad Austral de Chile. 20 de abril de 1993.
- DYER, K.R. *Estuary: a physical introduction*. Wiley Inc. Second edition, New York. 1998
- KINNE O. *Physiological aspects of animal life in estuaries with special reference to salinity*. Netherlands J. Sea Res. 3:224- 244. 1966
- SAVANT, CJ. RODEN, Martin S. CORPENTEN, Gordon L. *Diseño electrónico Circuitos de sistema*. Prentice Hall
- REAMEN, Donal A. *Análisis y diseño de circuitos electrónicos*, tomo I y II. Mc. Graw Hill. 1999.
- HORENSTEIN, Mark N. *Microelectrónica: Circuito y Dispositivos*. Prentice Hall Hispanoamérica S. A.
- PALACIOS, Enrique. REMIRO, Fernando. LOPES, Lucas J. *Microcontrolador PIC16F84, Desarrollo de proyectos*, México DF. Alfaomega, 2004. ISBN9070-15-1033-X

Referencia Electrónica

- <http://www.maxim-ic.com/>
- <http://www.itu.int/>
- <http://www.koders.com/>
- <http://www.oshonsoft.com/>
- <http://micropic.wordpress.com/>
- <http://www.matpic.com/>
- <http://www.roso-control.com/>
- <http://www.electronicafacil.net/descargas/>
- <https://www.ccsinfo.com/>
- <http://www.microchip.com/>
- <http://www.arcelect.com/rs232.htm>
- <http://www.abcdatos.com/utiles/ascii.html>
- <http://www.olimex.cl/>
- <http://www.micro-examples.com/>
- <http://www.jvmbots.com/>
- <http://www.todopic.com.ar/>
- <http://www.x-robotics.com/>
- <http://www.psicofxp.com/>
- <http://pyspanishdoc.sourceforge.net/>
- <http://mdubuc.freeshell.org/>
- <http://www.piclist.com/>
- <http://pic-linux.foroactivo.net/>
- <http://www.yoreparo.com/>
- <http://www.acuarios-marinos.com/>
- <http://www.learobotics.com/>
- <http://www.ucontrol.com.ar/>
- <http://proyectosfie.com/>
- <http://www.unicrom.com/>
- <http://www.pic16f84a.com/>