



INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

Curso Académico 2011/2012

Proyecto de Fin de Carrera

Interfaz de Comunicación

**Transmisor Thomcast de OC &
Sintetizador Continental PTP040**

Autor: Óscar Román Fernández

Tutor: Dr. Norberto Malpica González de Vega

AGRADECIMIENTOS

Me gustaría que estas líneas sirvieran para expresar mi más profundo y sincero agradecimiento a todas aquellas personas que con su ayuda han colaborado en la realización del presente proyecto, en especial al Dr. Norberto Malpica, por la orientación, el seguimiento y la supervisión continúa del mismo, pero sobre todo por la motivación y el apoyo recibido.

Especial reconocimiento merece el interés mostrado por mi trabajo y las sugerencias recibidas de la profesora y amiga Susana Borromeo López, con la que me encuentro en deuda por el ánimo infundido y la confianza en mí depositada.

También me gustaría agradecer la ayuda recibida del director del Dpto. Juan A. Hernández Tamames y a Joaquín Vaquero López, por su evaluación en la viabilidad del proyecto.

Sin olvidar a los técnicos de laboratorio Alfonso de Lara Rubio y Keny Rafael Oncebay por sus aportaciones y ayudas a la hora del diseño de los PCB's.

Con mucho cariño quiero mostrar mi sincero agradecimiento a Emy Gómez secretaria de ETSII y amiga, por que se presta siempre en ayudar a un alumno y además lo hace de corazón, con una sonrisa, no cambies nunca.

A mis mentores en el campo de la física, José Luis Trueba y Ángel Luis Álvarez, profesores y amigos, cuya labor no fue solo la de impartir y enseñar la materia, sino también la de hacerme ver cuando hay "aire en la velas" para remar con más fuerza.

Quisiera hacer extensiva mi gratitud a mis compañeros de RTVE y, especialmente al equipo técnico de RNE, por su amistad y colaboración, en especial a Jefe de Centro Emisor Francisco Gonzalo por el suministro de la documentación y de los datos necesarios para la realización del presente proyecto, además de por ser un buen compañero y amigo.

También quiero dar las gracias a la Dirección de RRHH, en especial a D. José Luis Blanco Heras, por su amistad, al Director de RNE Benigno Moreno, así como a toda Dirección Técnica RNE, Miguel Ángel Coletto, José Luis Hernández, García Mediano y a mi jefe y amigo Luis Melgar por su ayuda prestada.

Un agradecimiento muy especial y cariñosísimo, por la comprensión, paciencia y el ánimo recibidos por parte de mis padres, Clemente y Encarnación, y mi hermano Javier, gracias de corazón por estar siempre a mi lado.

Sin olvidar a mis familiares y amigos, en especial a John Omonte y Laura Quintans por estar siempre en los momentos importantes de mi vida.

A todos vosotros, muchas gracias.

Óscar Román Fernández.

RESUMEN DEL PROYECTO

El Centro Emisor de Onda Corta de Radio Exterior de España (REE-RNE-RTVE) necesita poder conectar a un transmisor de onda corta, un sintetizador distinto del proporcionado por el fabricante.

El nuevo sintetizador tiene una conexión física y lógica distinta a la del sintetizador original, por tanto no son compatibles entre sí.

El proyecto resuelve, mediante el diseño de un interfaz, la incompatibilidad existente, tanto física como lógica, para el uso de este sintetizador en el transmisor de onda corta.

Como ya se ha mencionado, tanto las conexiones físicas como la lógica de control son distintas, teniendo protocolos de comunicación propietarios.

El proyecto de Interfaz de Comunicación ha sido desarrollado siguiendo las siguientes etapas:

1. **Análisis del problema:** En esta etapa evaluamos a priori, la posible solución y si es factible la resolución del mismo, desde el punto de vista de un proyecto de ingeniería.
2. **Tecnologías:** En este punto se estudian las tecnologías empleadas.
3. **Diseño:** En esta etapa se abordan las posibles soluciones y se elige la que mejor se adapta a la resolución de dicho problema.
4. **Evaluación:** En este punto realizamos una valoración del sistema desde el punto de vista de su viabilidad.
5. **Desarrollo del sistema:** Debido a la incompatibilidad tanto física como lógica tenemos que dividir este punto en dos:

5.1 **Hardware**, para resolver las incompatibilidades físicas.

5.2 **Software**, para resolver las incompatibilidades lógicas.

Por un lado, indicar, que tenemos la marca comercial del transmisor de onda corta, Thomcast TSW 2250 Transmisor de OC de 250 kW, que en adelante denominaremos “Transmisor”.

Por el otro lado tenemos el sintetizador, que queremos conectar al transmisor, de la marca comercial, Continental Electronics PTP 040S4030, en adelante denominaremos “Sintetizador”.

RTVE, RNE, REE, Continental Electronics y Thomcast, son marcas registradas propiedad de sus respectivos propietarios.

ÍNDICE

1. INTRODUCCIÓN	7
2. OBJETIVOS	10
2.1 ANÁLISIS	10
2.2 DISEÑO INICIAL	11
2.3 EVALUACIÓN	12
3. INTERFAZ.....	13
3.1 INTEFAZ HARDWARE	13
3.2 INTERFAZ SOFTWARE:	16
4. TECNOLOGÍAS	17
4.1 STANDARD RS232.....	17
4.2 TECNOLOGÍA TTL	19
4.3 INTERFAZ TTL-RS232.....	20
5. DESCRIPCIÓN INFORMÁTICA	22
5.1 SOLUCIÓN HARDWARE	22
5.1.1 DESCRIPCIÓN HARDWARE	23
5.1.2 DISEÑO DEL HARDWARE.....	23
5.1.3 ESQUEMA GENERAL DEL INTERFAZ DE COMUNICACIÓN	51
5.1.4 ESQUEMAS DE LOS CIRCUITOS	52
5.2 SOLUCIÓN SOFTWARE	58
5.2.1 INTRODUCCIÓN	58
5.2.2 DESCRIPCIÓN SOFTWARE.....	61
5.2.3 PROTOCOLO DE COMUNICACIÓN	62
5.2.4 DIAGRAMAS DE FLUJO DE CONTROL	63
5.2.5 IMPLEMENTACIÓN.....	70
5.2.6 CARGA DEL PROGRAMA	70
5.3 MODO USUARIO.....	71
5.3.1 DETECCIÓN DE ANOMALIAS	72
5.3.2 MODO TEST DISPLAY FRECUENCÍMETRO	73
5.3.3 RESET DEL PIC MANUAL.....	73
6. CONCLUSIONES.....	74
7. BIBLIOGRAFÍA	75
8. GLOSARIO DE TÉRMINOS.....	76
9. ANEXOS	81
9.1 DISEÑO DE LOS PCB'S	81
9.2 ALGORITMO DE REALIZACIÓN DE PROYECTOS CON MICRO- CONTROLADORES PARA INGENIEROS INFORMÁTICOS	81

ÍNDICE DE IMÁGENES

1. Figura 1-1. Bloques básicos de un Emisor de OC.	7
2. Figura 1.1-2. Bloques básicos de un Emisor de OC con detalle del bloque Transmisor OC.	8
3. Figura 2.2-1. Esquema de conexiones entre sintetizador, Interfaz de Comunicación y transmisor.	12
4. Figura 3.1-1. Detalle de los conectores de conexión del sintetizador.	13
5. Figura 3.1-2. Tabla de descripción de los pines para programación de las frecuencias en remoto	15
6. Figura 3.2-1. Tabla del Protocolo TRC.	16
7. Figura 4.1-1. Descripción de pines en RS-232.	18
8. Figura 5.1-1. Bloques del sistema Interfaz de Comunicación.	22
9. Figura 5.1.2-1. Tabla de características básicas entre fuentes rectificadas y fuentes conmutadas.	24
10. Figura 5.1.2-2. Etapas básicas de una fuente de alimentación rectificada y sus señales de salida.	25
11. Figura 5.1.2-3. Fuente de alimentación rectificada de 5V.	25
12. Figura 5.1.2-4. Esquema de conexiones del MAX232 al conector RS232 y adaptación de niveles.	27
13. Figura 5.1.2-5. Selección de registro de memoria tipo D.	28
14. Figura 5.1.2-6. Detalle del chip PIC16F886.	29
15. Figura 5.1.2-7. Detalle de los pines del chip PIC16F886.	31
16. Figura 5.1.2-8. Tabla de las funciones principales de los pines del PIC.	32
17. Figura 5.1.2-9. Continuación tabla de las funciones principales de los pines del PIC.	33
18. Figura 5.1.2-10. Módulos principales que acompañan al microcontrolador PIC.	34
19. Figura 5.1.2-11. Diagramas de bloques del PIC.	35
20. Figura 5.1.2-12. Detalle de funcionamiento del registro TRIS.	36
21. Figura 5.1.2-13. Detalle del oscilador interno, formado por dos osciladores internos separados.	38
22. Figura 5.1.2-14. Esquema de oscilador externo.	39
23. Figura 5.1.2-15. Tabla de valores recomendados para la configuración del oscilador.	40
24. Figura 5.1.2-16. Detalle del circuito temporizador interno del microcontrolador.	41
25. Figura 5.1.2-17. Uso en un programa del temporizador watchdog.	42
26. Figura 5.1.2-18. Circuito Reset.	43
27. Figura 5.1.2-19. Detalle del efecto rebote producido por un interruptor mecánico.	44
28. Figura 5.1.2-20. Tiempo de espera de inicio, Reset T.	45
29. Figura 5.1.2-21. Esquema de conexiones del microcontrolador.	46
30. Figura 5.1.2-22. Display 7 segmentos	47
31. Figura 5.1.2-23. Esquema del display frecuencímetro.	48
32. Figura 5.1.2-24. Conexión Amphenol 57-40500.	49
33. Figura 5.1.3-1. Esquema general del Interfaz de Comunicación.	51
34. Figura: 5.2.1-1. Programación de un microcontrolador.	59
35. Figura: 5.2.1-2. Secuencia de programación de un microcontrolador.	59

36. Figura: 5.2.1-3. Compilación del código C.	60
37. Figura: 5.2.2-1. Esquema básico de conexiones de datos.....	61
38. Figura 5.2.4-1. Diagrama del algoritmo principal.	64
39. Figura 5.2.4-2. Diagrama de flujo del tratamiento de una interrupción.	66
40. Figura 5.2.4-3. Diagrama de flujo del procedimiento Interfaz.	68
41. Figura 5.2.4-4. Diagrama de bloques del procedimiento Cambio de Frecuencia.	69
42. Figura 5.2.6-1. Programador de microcontroladores.....	71
43. Figura 5.3-1. Detalle del conector Amphenol.....	71
44. Figura 5.3-2. Detalle del conector RS232.....	73
45. Figura:9.2-1. Algoritmo de realización de proyectos con microcontroladores para Ingenieros Informáticos.....	81

1. INTRODUCCIÓN

Este proyecto nace de una necesidad real creada en un entorno de trabajo, como es la corporación RTVE. Durante el trabajo en un Centro Emisor de Radio Exterior de España, de RNE, se observa la necesidad de poder conectar a un transmisor de Onda Corta un sintetizador de distinta marca comercial que el diseñado por el fabricante.

Previamente vamos a explicar las partes principales del funcionamiento de un emisor de OC con modulación AM.

En la Figura 1-1, podemos observar los bloques básicos que lo forman. A continuación daremos una breve descripción de su funcionamiento.

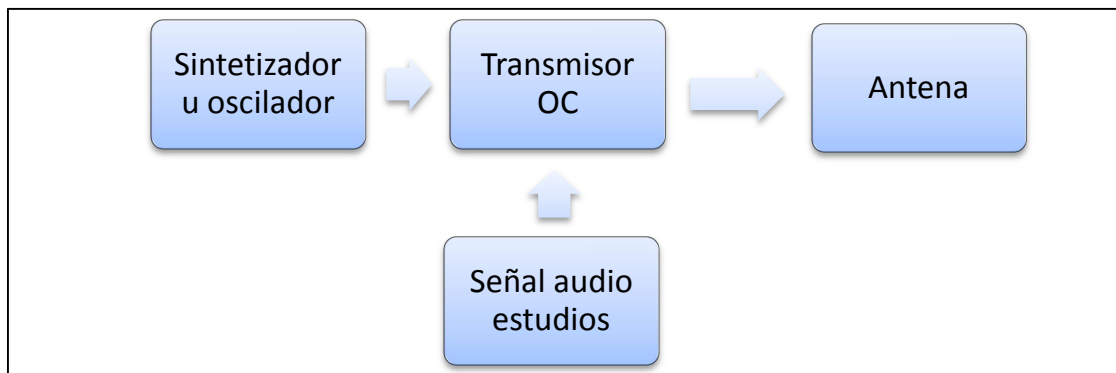


Figura 1-1. Bloques básicos de un Emisor de OC.

BLOQUES BÁSICOS DE UN EMISOR DE OC

1. Sintetizador de frecuencias u oscilador, genera una señal sinusoidal de bajo nivel. Esta señal será la portadora de RF.

2. Transmisor OC, será el encargado de modular la señal de audio junto con la portadora, dando a su salida una señal AM.

Está formado por tres bloques (figura 1.1-2):

2.1 Amplificador RF, la señal del Sintetizador de frecuencias u oscilador entra en el Transmisor de OC, se amplifica por medio del amplificador RF, pasando al Modulador.

2.2 Modulador, multiplica la señal en un mezclador balanceado, por un lado entra la señal de audio de los estudios y por el otro la portadora, como resultado, se obtiene la señal AM a un bajo nivel de señal.

2.3 Amplificador final RF, la señal AM se inyecta a un amplificador final de potencia.

3. Antena, se trata de un dispositivo pasivo (conductor metálico) diseñado con el objetivo de emitir hacia el espacio libre. Una antena transmisora transforma voltajes en ondas electromagnéticas.

4. Señal audio estudios, recibida a través de un enlace microondas, es la señal que modula de la portadora, en una emisión AM.

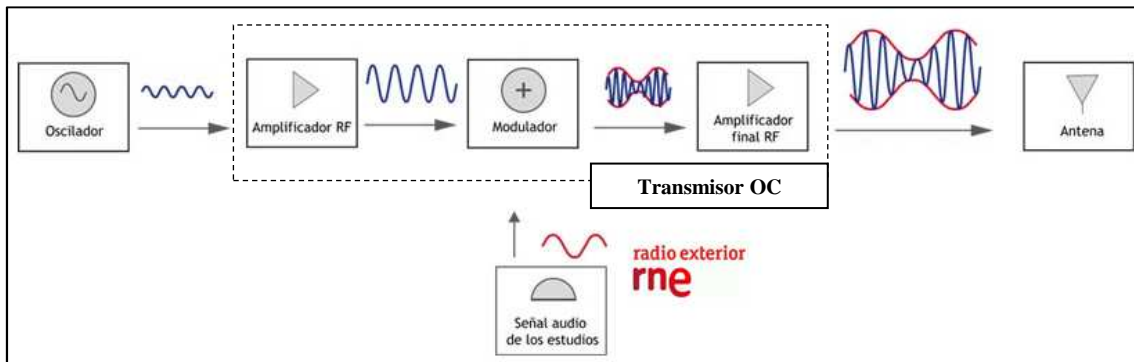


Figura 1.1-2. Bloques básicos de un emisor de OC con detalle del bloque Transmisor OC.

Ante esta necesidad, surge la idea de poder realizar un Interfaz de Comunicación, el cual aporta la solución tanto de las conexiones físicas, actualmente incompatibles por el tipo de conector, como a nivel lógico, adaptando el protocolo de comunicaciones.

Para la resolución del mismo se requiere, por tanto, de un proyecto técnico de ingeniería, en este caso se aborda como PFC de la titulación de Ingeniería Técnica de Informática de Sistemas, cuyo proyecto requiere de las tareas de análisis, diseño, evaluación y desarrollo.

Vamos a necesitar por tanto el estudio de la documentación técnica de la que se dispone, por parte del fabricante de ambos sistemas transmisor y sintetizador.

Las conclusiones a las que llegamos una vez revisada la documentación son las siguientes:

- **El sintetizador** u oscilador, su función principal es entregar al transmisor una frecuencia (función sinusoidal) fijada por un valor f que nos solicita el transmisor.
- **El transmisor**, su función principal es la de modular en amplitud una señal de audio, para entregar a su salida, una señal de OC con modulación AM.

Para su funcionamiento necesita de un sintetizador que entregue una señal portadora, el valor de esa frecuencia portadora será solicitada por el transmisor al sintetizador, mediante un protocolo de comunicación.

Con esta idea inicial del funcionamiento de los sistemas, el estudio de los manuales y documentación aportada por los fabricantes, podemos inferir que es posible la compatibilidad de dichos equipos, siempre que incluyamos entre ambos un Interfaz de Comunicación.

Dicho Interfaz va a necesitar tanto de una parte física, en la cual adaptaremos las señales y conectores eléctricos, como una parte lógica que se encargue de interpretar las distintas señales y eventos para el correcto funcionamiento del interfaz.

2. OBJETIVOS

2.1 ANÁLISIS

Una vez visto el esquema general de un transmisor de OC con modulación AM, nos vamos a centrar en los dos elementos a conectar entre si, el sintetizador y el transmisor.

Este análisis lo vamos a dividir en dos partes, haciendo una descripción física y una descripción lógica, tanto del transmisor como del sintetizador.

1. Descripción física:

1.1 Sintetizador

- Conector BNC, salida analógica de la frecuencia solicitada por el transmisor.
- Amphenol 57-40500, entrada/salida digital para control remoto de las funciones del sintetizador.

1.2 Transmisor

- Conector BNC, entrada de la señal analógica procedente del sintetizador, portadora de RF.
- Conector RS232, puerto de comunicaciones de entrada/salida para controlar las funciones del sintetizador.

2. Descripción lógica:

2.1 Sintetizador

- Recepción en paralelo de la frecuencia en modo control remoto.
- No tiene transmisión de asentimiento cuando recibe una nueva frecuencia.

Por tanto podemos decir que este equipo solo esta a la “escucha” de recibir órdenes, no genera ningún mensaje de confirmación.

2.2 Transmisor

- Trasmisión en serie de los comandos para el control remoto del sintetizador.

- Recepción serie de asentimiento de un correcto funcionamiento por parte del sintetizador.

Tras esta breve introducción podemos decir, que el transmisor envía las órdenes y necesita la recepción de un mensaje confirmado que la operación ha sido correcta.

De este análisis sacamos la conclusión que el conector BNC de salida de la señal analógica desde el sintetizador y la entrada analógica con conector BNC del transmisor son compatibles entre si, y por tanto se puede conectar con un cable coaxial con la impedancia característica y los conectores necesarios.

Los conectores de entrada/salida de datos digitales, tanto del sintetizador como del transmisor son totalmente distintos, y la información que manejan también (protocolos de comunicación distintos).

Una primera fase de nuestro proyecto será crear el interfaz de conexiones necesario para interconectar sendos equipos, a esta fase le vamos a denominar solución hardware.

La segunda fase será denominada solución software, y su función será la de decodificar y codificar los mensajes que ambos equipos han de enviarse para un correcto funcionamiento del sistema.

Se podría resumir que nuestro Interfaz de Comunicación hace de “intérprete” entre ambos equipos.

2.2 DISEÑO INICIAL

Tenemos por tanto que realizar un diseño inicial tanto a nivel físico como lógico.

Una primera aproximación del diseño físico podría ser el siguiente esquema de conexiones de nuestro Interfaz de Comunicación con el sintetizador y con el transmisor, figura 2.2-1.

A nivel lógico o de protocolo de comunicaciones, tenemos que proceder a interpretar los mensajes recibidos a través del conector RS232, identificaremos el mensaje mediante el software de nuestro Interfaz de Comunicación, procesaremos la información recibida y una vez resuelto el mensaje enviaremos por el conector Amphenol 57-40500 la información en paralelo, codificada con el protocolo adecuado.

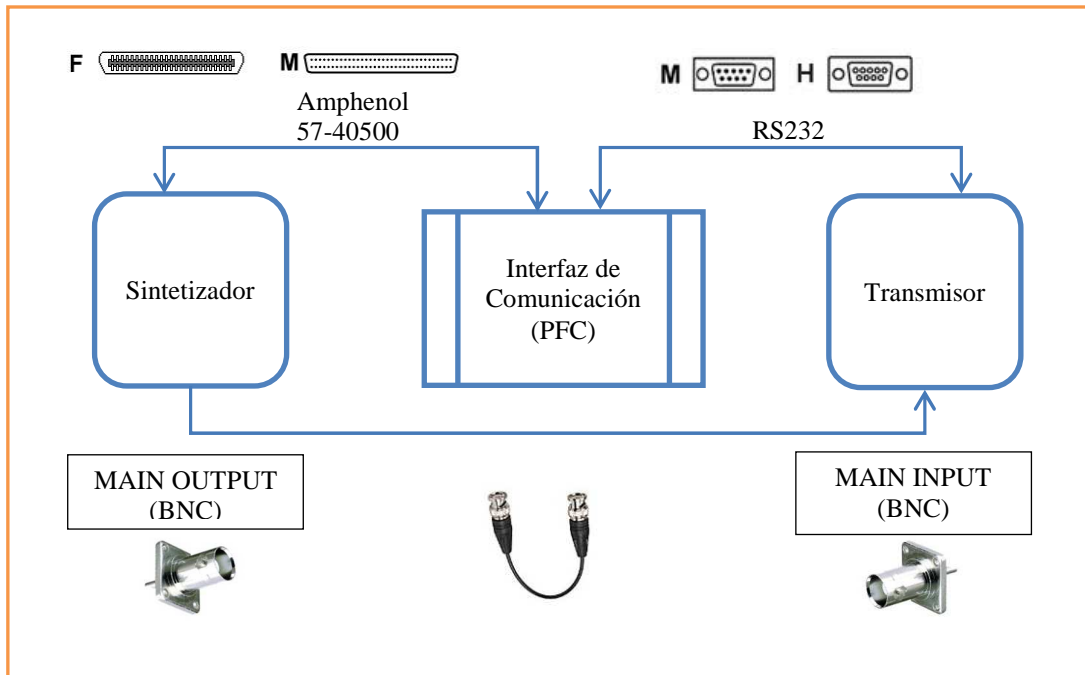


Figura 2.2-1. Esquema de conexiones entre sintetizador, Interfaz de Comunicación y transmisor.

Para concluir generamos los comandos de asentimientos necesarios para el transmisor y que no son generados por el sintetizador, por no estar incluidos estos en su protocolo de comunicación.

2.3 Evaluación

Con toda la información recopilada hasta el momento, tenemos una idea de la magnitud del proyecto al que nos enfrentamos.

Por un lado la parte física de conexión y por el otro la parte lógica o de interpretación de los comandos.

En la etapa de desarrollo se procederá a elegir un componente u otro dependiendo de las tecnologías y necesidades prácticas, pero por el momento una primera aproximación de las necesidades del sistema sería las siguientes:

1. Por un lado necesitamos un hardware de conexión apropiado a las señales que enviamos y recibimos. Haciéndolas compatibles entre si por medio de nuestro Interfaz de Comunicación.
2. Todo este hardware necesitará a su vez para un correcto funcionamiento de una parte lógica o software, que será la encargada de hacer las “traducciones” de los códigos enviados entre unos y otros elementos del sistema.

3. INTERFAZ

3.1 INTERFAZ HARDWARE

Vamos a detallar el interfaz hardware de cada equipo, para ello mostraremos sus características técnicas.

Interfaz transmisor:

Según documentación proporcionada por el fabricante, tenemos dos conexiones, necesarias para el funcionamiento del sistema:

1. RF IN, entrada de la señal analógica procedente del sintetizador, esta señal es la portadora de la señal AM.

2. RS232, su interfaz es un conector de entrada/salida de datos RS232, nos proporciona un enlace de comunicación con el transmisor.

Descripción de los pines:

Pin 2: TXD (datos transmisión salida)

Pin 3: RXD (datos recepción entrada)

Pin 5: GND (masa)

Pin 7,8: unidos entre si.

Pin 1, 4, 6: unidos entre si.

Pin 9: no conectado.

Interfaz del Sintetizador:

Según documentación proporcionada por el fabricante, tenemos dos conexiones, necesarias para el funcionamiento del sistema, ver figura 3.1-1:

1. Main output, su interfaz es un conector BNC. A su salida tenemos la señal analógica que necesita el transmisor como portadora de la señal AM.

2. Remote control, su interfaz es un conector de entrada/salida de datos RS232, nos proporciona un enlace de comunicación y control con el transmisor.

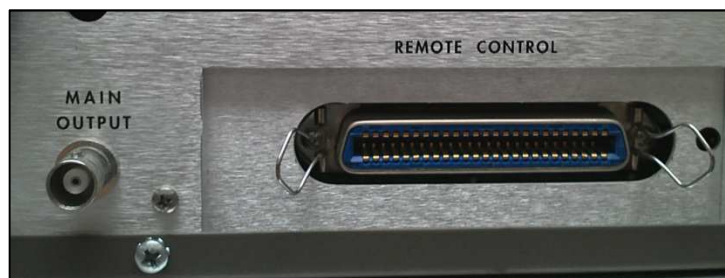



Figura 3.1-1. Detalle de los conectores de conexión del sintetizador.

Características técnicas del interfaz Remote Control:

- Programación en remoto, por niveles TTL, en paralelo y codificado en BCD con lógica negativa.
- El uso de lógica negativa hace que todas las funciones están por defecto Off considerando un estado alto o falso estado.
- Tecnología: compatible con 74-Type IC's.
- Conector de programación remota, este conector suministra en remoto la frecuencia y el modo de programación (remota o local).
- Conector Amphenol 57-40500 (HEMBRA) F 

Modo de programación de Frecuencia por control remoto:

Para utilizar el modo de programación en remoto, tenemos que configurar los siguientes pines.

- Remote enable: Pin 42, este pin con un nivel alto lógico, activa la programación de frecuencia por control remoto.

Por defecto esta conectado a una resistencia de pull-up a nivel high. Para activar el modo Remoto tenemos que poner el pin 42 a un nivel Low.

- Vcc +5 V: Pin 48 (100 mA máx.), por este pin tenemos el valor de tensión indicado, es un pin de salida.
- Ground: Pin 50, utilizaremos esta masa para aislar el conductor mediante una malla, que evitará interferencias en las señales recibidas.

El resto de los pines del conector son las entradas en BDC del valor de la frecuencia que queremos programar en el sintetizador.

Código BCD

El código BCD (Binary Coded Decimal - Código Binario Decimal) es un código binario utilizado para representar números decimales. Consiste en números binarios de 4 dígitos que representan los primeros diez dígitos (0, 1, 2, 3...8, 9).

Aunque cuatro dígitos binarios pueden crear 16 combinaciones, el código BCD utiliza solamente las primeras diez combinaciones.

En la Figura 3.1-2, tenemos una tabla donde se describen los pines para programación de las frecuencias en remoto:

Modo de programación de la frecuencia en Remoto						
Dígito	BCD Pesos	1	2	4	8	Latch Enable
		Número de Pin				
10 Mhz	0 - 9 BCD	15	16	---	---	23
1 Mhz	0 - 9 BCD	17	18	19	20	24
100 Khz	0 - 9 BCD	1	2	26	27	24
10 Khz	0 - 9 BCD	3	4	28	29	25
1 Khz	0 - 9 BCD	5	6	30	31	25
100 Hz	0 - 9 BCD	7	8	32	33	46
10 Hz	0 - 9 BCD	9	10	34	35	46
1 Hz	0 - 9 BCD	11	12	36	37	47
0.1 Hz	0 - 9 BCD	13	14	38	39	47

Figura 3.1-2. Tabla de descripción de los pines para programación de las frecuencias en remoto.

La columna Latch Enable indica el número del pin que hay que activar una vez tengamos la frecuencia en BCD y lógica negativa en el conector, para validar los datos y proceder a la carga de los mismos.

Indicar que todos los pines se activan con niveles altos lógicos. Este sistema, esta implementado en lógica negativa, por tanto un valor alto lógico equivale a un valor eléctrico bajo de los niveles TTL normalizados.

Interfaz de Comunicación:

Nuestro Interfaz de Comunicación, eléctricamente tendrá que cumplir con las características técnicas compatibles con las dos tecnologías que tenemos presente en el sistema.

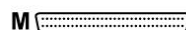
Por un lado, una conexión hacia el transmisor mediante un Conector RS232



Descripción de los pines:

- Pin 2: RXD (datos recepción entrada)
- Pin 3: TXD (datos transmisión salida)
- Pin 5: GND (masa)
- Pin 7, 8: unidos entre si.
- Pin 1, 4, 6: unidos entre si.
- Pin 9: no conectado.

Por el otro lado, una conexión hacia el sintetizador mediante el Conector Amphenol 57-30500 (MACHO)



Niveles TTL:

Low: +0.7 V máx.

High: +2.0 V min.

3.2 INTERFAZ SOFTWARE

Para el desarrollo software nos fijaremos en el protocolo de comunicación:

El Transmisor tiene un sistema de Control Remoto que utiliza el protocolo TRC.

Protocolo TRC

Lo detallamos en la Figura 3.2-1, cada comando del Transmisor al Sintetizador consiste en el envío de caracteres ASCII, así como la recepción de respuesta del Sintetizador.

Comandos del Transmisor al Sintetizador	Respuesta del Sintetizador al Transmisor	Descripción
F<Frecuency> (EOI)	F<Frecuency> (EOI)	Envía la frecuencia al sintetizador.

Figura 3.2-1. Tabla del Protocolo TRC.

La configuración de comunicación vía RS232 sigue unos parámetros para el control remoto.

- Baud Rate : 9600
- Data Bits: 8
- Stop Bits: None
- Parity: None
- XOR/XOFF: OFF
- Handshake: None
- Transition: 0->1 EOI line

4. TECNOLOGÍAS

4.1 STANDARD RS232

RS232 (Recommended Standard 232, también conocido como Electronic Industries Alliance RS-232C) es una norma para el intercambio de una serie de datos binarios entre un DTE (Data Terminal Equipment, Equipo terminal de datos) y un DCE (Data Communication Equipment, Equipo de Comunicación de datos).

Conector RS-232, conector de interconexión entre los mismos, se requerirá la conexión de un DTE (Data Terminal Equipment) con otro DTE.

El RS-232 consiste en un conector tipo DB-25 (de 25 pines), aunque es normal encontrar la versión de 9 pines (DE-9, o popularmente también denominados DB-9) este es el que encontramos en el transmisor, el cual nos va a servir para el envío y recepción de las instrucciones para comandar el sintetizador.

Las señales con las que trabaja este puerto serie son digitales, de +12 V (0 lógico) y -12 V (1 lógico), para la entrada y salida de datos, y a la inversa en las señales de control. El estado de reposo en la entrada y salida de datos es -12 V. Según los ensayos realizados, para el control del puerto RS232 de nuestro sistema se puede utilizar niveles de +- 5 V, sin que se aprecie ninguna anomalía ni pérdida de datos.

El RS-232 puede transmitir los datos en grupos de 5, 6, 7 u 8 bits, a unas velocidades determinadas (normalmente, 9600 bits por segundo o más). A los datos transmitidos, les sigue un bit opcional de paridad (indica si el número de bits transmitidos es par o impar, para detectar fallos), y después 1 o 2 bits de Stop. Normalmente, el protocolo utilizado es 8N1 (que significa, 8 bits de datos, sin paridad y con 1 bit de Stop).

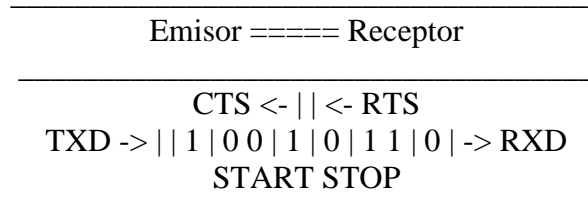
Una vez que ha comenzado la transmisión de un dato, los bits tienen que llegar uno tras otro a una velocidad constante y en determinados instantes de tiempo. Por eso se dice que el RS-232 es asíncrono por carácter y síncrono por bit. Los pines que portan los datos son RXD y TXD. Las demás se encargan de otros trabajos: DTR indica que el sistema está encendido, DSR que el aparato conectado a dicho puerto está encendido, RTS que el sistema puede recibir datos (porque no está ocupado), CTS que el aparato conectado puede recibir datos, y DCD detecta que existe una comunicación, presencia de datos.

Tanto el interfaz desarrollado como el transmisor tienen que usar el mismo protocolo serie para comunicarse entre sí. Puesto que el estándar RS-232 no permite indicar en qué modo se está trabajando, es el usuario quien tiene que decidirlo y configurar ambas partes. Como ya se ha visto, los parámetros que hay que configurar son: protocolo serie (8N1), velocidad del puerto serie, y protocolo de control de flujo.

Este último puede ser por hardware (handshaking RTS/CTS) o bien por software (XON/XOFF). La velocidad del puerto serie no tiene por qué ser la misma que la de transmisión de los datos, de hecho debe ser superior. Por ejemplo, para transmisiones de

1200 baudios es recomendable usar 9600, y para 9600 baudios se pueden usar 38400 (o 19200).

Este es el diagrama de transmisión de un dato con formato 8N1. El receptor indica al emisor que puede enviarle datos activando la salida RTS. El emisor envía un bit de START (nivel alto) antes de los datos, y un bit de STOP (nivel bajo) al final de estos.



En la Figura 4.1-1. Descripción de pines en RS-232, se puede observar una imagen del conector RS232 y de la asignación de los pines con su función.

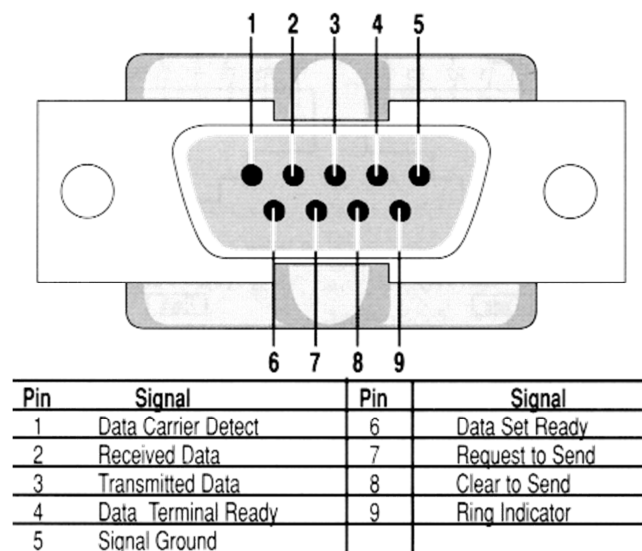


Figura 4.1-1. Descripción de pines en RS-232.

- TXD (Transmit Data, transmisión de datos, salida, pat. 2): Señales de datos que se transmiten del DTE al DCE. En principio, los datos no se pueden transmitir si alguno de los terminales RTS, CTS, DSR ó DTR está desactivado.
- RXD (Receive Data, recepción de datos, entrada, pat. 3): Señales de datos transmitidos desde el DCE al DTE.
- DTR (Data Terminal Ready, terminal de datos preparado, salida, pat. 20): Señal del DTE que indica que está conectado, generalmente en “0” indica que el DTE está listo para transmitir o recibir.

- DSR (Data Set Ready, dispositivo preparado, entrada, pat. 6): Señal del DCE que indica que el dispositivo está en modo de transmisión de datos.
- RTS (Request To Send, petición de envío, salida, pat. 4): Señal del DTE al DCE, notifica al DCE que el DTE dispone de datos para enviar. Se emplea en líneas semiduplex para controlar la dirección de transmisión.
- DCE que tome las medidas necesarias para prepararse para la transmisión.
- CTS (Clear To Send, preparado para transmitir, entrada, pat. 5): Señal del DCE al DTE indicando que puede transmitirle datos.
- CD (Carrier Detect, detección de portadora, entrada, pat. 8): Señal del DCE que ha detectado la señal portadora enviado por un modem remoto o que la línea telefónica está abierta.
- RI (Ring Indicator, timbre o indicador de llamada entrante, entrada, pat. 22): Señal del DCE indicando que está recibiendo una llamada por un canal conmutado.
- SG (GND) (System Ground ó Signal Ground, masa de señal, pat. 7): Masa común para todas las líneas.
- FG (GND) (Shield ó Protective Ground, tierra de protección, pat. 1): El conductor esta eléctricamente conectado al equipo.

Una secuencia normal, a través de la RS232, es la siguiente:

Ambos dispositivos son alimentados, indicando encendido (si ha sido establecido en el equipo). El DTE activa el terminal DTR y el DCE activa el terminal DSR. Una interfaz RS-232 bien diseñada no comunicará hasta que estos dos terminales estén activos. El DTE esperará la activación del terminal DSR y el DTE la activación del terminal DTR. Aunque DTR y DSR algunas veces pueden ser utilizados para el control del flujo, estos terminales solo indican que los dispositivos están conectados.

El DTE pregunta al DCE si este está listo. El DTE activa la línea RTS. El DCE si está listo, responde activando la línea CTS. Puestos de acuerdo ambos equipos, se puede entrar a comunicar.

Los datos son transferidos en ambos sentidos. El DTE envía información al DCE a través del terminal TXD. El DCE envía información al DTE a través del terminal RXD.

4.2 TECNOLOGÍA TTL

La entrada de datos de nuestro sintetizador se realiza a través del Conector Amphenol 57-30500, ya hemos detallado la disposición de la información en los pines, en el punto anterior, esa sería la información o parte lógica. Por otro lado nos ha

indicado el fabricante que utiliza tecnología TTL, por tanto procedemos al estudio de dicha tecnología.

TTL son las siglas en inglés de transistor-transistor logic, es decir, "lógica transistor a transistor". Es una familia lógica o lo que es lo mismo, una tecnología de construcción de circuitos electrónicos digitales. En los componentes fabricados con tecnología TTL los elementos de entrada y salida del dispositivo son transistores bipolares.

La tecnología TTL se caracteriza por sus etapas:

- Etapa de entrada por emisor.
- Separador de fase. Es un transistor conectado en emisor común que produce en su colector y emisor señales en contrafase.

- Driver. Está formada por varios transistores, separados en dos grupos. El primero va conectado al emisor del separador de fase y drenan la corriente para producir el nivel bajo a la salida. El segundo grupo va conectado al colector del divisor de fase y produce el nivel alto.

Su tensión de alimentación característica está comprendida entre los 4,75 V y los 5,25 V (como se ve un rango muy estrecho).

Los niveles lógicos vienen definidos por el rango de tensión comprendida entre 0,0V y 0,08V para el estado L (bajo) y los 2,4 V y V_{cc} para el estado H (alto).

La velocidad de transmisión entre los estados lógicos es la principal ventaja de TTL, si bien esta característica le hace aumentar su consumo siendo esta su principal desventaja. Motivo por el cual han aparecido diferentes versiones de TTL como FAST, LS, S, etc. y últimamente los CMOS: HC, HCT y HCTLS. En algunos casos puede alcanzar poco más de los 250 Mhz.

Las señales de salida TTL se degradan rápidamente si no se transmiten a través de circuitos adicionales de transmisión (no pueden viajar más de 2 m por cable sin graves pérdidas).

4.3 INTERFAZ TTL-RS232

Para una comunicación full duplex desde la UART de un microcontrolador (PIC) deben conectarse un mínimo número de señales, concretamente TXD y RXD así como la masa (GND, SG o Signal Ground).

Las UART (Transmisor y Receptor Síncrono Asíncrono Universal) están diseñadas para convertir las señales que maneja la CPU y transmitir las al exterior.

Las UART deben resolver problemas tales como la conversión de voltajes internos del DCE con respecto al DTE, gobernar las señales de control, y realizar la transformación desde el bus de datos de señales en paralelo a serie y viceversa.

Son robustas, toleran circuitos abiertos, cortocircuitos y escritura simultánea sobre un mismo pin, entre otras consideraciones. Es en la UART en donde se implementa la interfaz.

Queremos conectar nuestro microcontrolador (con señales típicamente entre 3.3 y 5 V) con un puerto RS-232 estándar, para ello utilizaremos un driver de línea, típicamente un MAX232 o compatible, el cual mediante dobladores de voltaje positivos y negativos permite obtener la señal bipolar (típicamente alrededor de +/- 6 V) requerida por el estándar.

El MAX232 es un conversor de nivel TTL/RS232. Sólo es necesario este circuito integrado y 4 condensadores. En la siguiente etapa procederemos al estudio de dicho sistema para poder integrarlo en nuestro proyecto. (MAX232) Se puede consultar las hojas de características técnicas de este componente en el CD adjunto.

Sobre los circuitos, todos los voltajes están con respecto a la señal de tierra, las convenciones que se usan son las siguientes:

Voltaje	Señal	Nivel Lógico	Control
+3 a +15	Espacio	0	On
-3 a -15	Marca	1	Off

Los valores de voltaje se invierten con respecto a los valores lógicos. Por ejemplo, el valor lógico positivo corresponde al voltaje negativo. También un 0 lógico corresponde a la señal de valor verdadero ó activado. Por ejemplo si la línea DTR está al valor 0 lógico, se encuentra en la gama de voltaje que va desde +3 a +15 V, entonces DTR está listo (ready).

Otras maneras para convertir TTL a RS-232 es usando circuitos típicos de transistores y diodos discretos o los circuitos integrados MC1488 y MC1489, los cuales implementan dos drivers y receptores TTL, (4 por chip), para una RS-232 de forma compatible con las reglas anteriores.

5. DESCRIPCIÓN INFORMÁTICA

Con todos los elementos anteriores, ya tenemos una idea de las necesidades de nuestro interfaz.

En función de las necesidades descritas anteriormente configuraremos nuestro sistema adaptándolo a un proyecto de ingeniería.

Hemos decidido atacar la resolución del problema dividiéndolo en varias etapas, (*Divide et vinces*, "Divide y vencerás", Julio César), según la funcionalidad de las mismas.

5.1 SOLUCIÓN HARDWARE

Se ha optado por realizar una división funcional de los distintos bloques, así cada uno de ellos realizará una función específica dentro del desarrollo del Interfaz de Comunicación.

Podemos ver esos bloques en la figura 5.1-1, donde se muestra de manera simplificado el Interfaz de Comunicación con los diferentes módulos que lo componen.

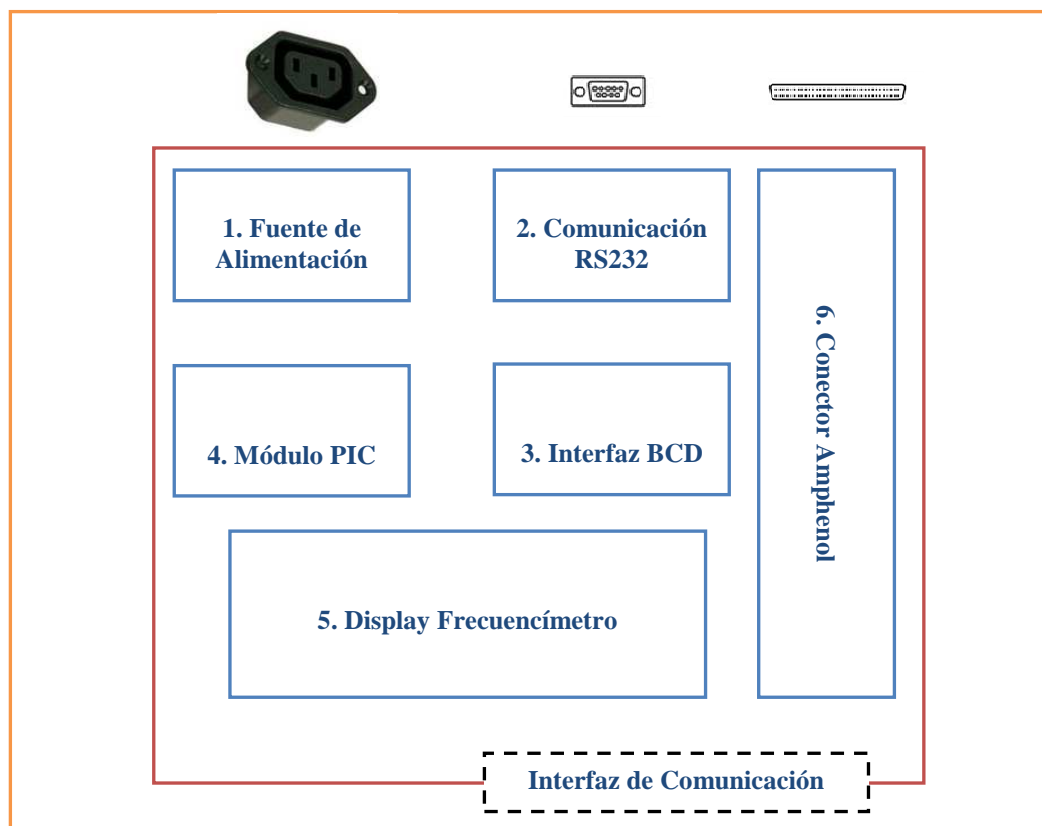


Figura 5.1-1. Bloques del sistema Interfaz de Comunicación.

5.1.1 DESCRIPCIÓN HARDWARE

Como ya hemos visto, hemos dividido el hardware de nuestro proyecto según su funcionalidad en grupos, ahora vamos a detallar brevemente cada uno de ellos:

1. **Fuente de alimentación**, este grupo lo formaran los componentes encargados de alimentar y filtrar las tensiones requeridas para el sistema, teniendo en cuenta consumos y potencia requerida.
2. **Comunicación Rs232**, será el grupo de componentes encargados en el Interfaz de Comunicación con el puerto serie RS232, su función será la adaptación de señales TTL, entregadas por nuestro PIC y entregará una señal adaptada al estándar RS232.
3. **Interfaz BCD**, estará formado por los componentes necesarios para adaptar la información entregada por el puerto de comunicaciones de nuestro PIC al interfaz de salida BCD. Las características lógicas de dicho interfaz serán explicadas más adelante en el proyecto por no cumplir un estándar y ser, como ya se ha visto, una comunicación propietaria del sintetizador PTS040.
4. **Módulo PIC**, esta formado por el PIC 16F887 y los componentes discretos asociados, se trata del núcleo o parte central del proyecto, será el encargado de realizar todas las operaciones lógicas y de control para el desarrollo correcto de las funcionalidades del Interfaz de Comunicación.
5. **Display Frecuencímetro**, se trata de un módulo formado por un dispositivo visualizador y los componentes asociados al funcionamiento del mismo. En el podemos observar el valor de la frecuencia del sintetizador. Este módulo tiene una función solo informativa y es prescindible para el funcionamiento del sistema.
6. **Conector Amphenol**, en esta placa estará el conector Amphenol 57-30500 para el interfaz con el sintetizador y un LED indicativo que mostrará si el sintetizador esta listo para recibir un mensaje.

5.1.2 DISEÑO DEL HARDWARE

En este apartado vamos a implementar los distintos diseños de los grupos funcionales de la lista anterior. Para ello atenderemos las necesidades del proyecto, tanto a niveles eléctricos, lógicos, como en los estándares que tienen que cumplir.

1. Fuente de alimentación

Para el diseño de este módulo, consultamos las características técnicas necesarias y las diversas opciones de diseño.

Dentro de las distintas fuentes de alimentación existen dos grandes grupos:

- Fuentes rectificadas.
- Fuentes conmutadas.

La figura 5.1.2-1, es una tabla donde mostramos una pequeña lista de características básicas de estos dos tipos de fuentes.

Fuentes rectificadas	Fuentes conmutadas
Sencillas de diseñar.	Complejas en su diseño.
Consumos bajos.	Consumos altos.
Una única tensión de salida.	Múltiples tensiones de salida.
Aumento de peso y volumen proporcional a la potencia a entregar	En potencias altas tienen menor peso y volumen que las rectificadas
En consumos bajos son baratas	En consumos bajos son caras por complejidad y número de componentes

Figura 5.1.2-1. Tabla de características básicas entre fuentes rectificadas y fuentes conmutadas.

A la vista de estas características, la elección para el diseño del módulo, es la de una fuente de alimentación rectificada.

Los consumos que vamos a necesitar son muy bajos, del orden de 600mA, por tanto su peso y tamaño serán reducidos.

Una única tensión de salida para todo el circuito.

Por su sencillez a la hora de diseñarse y precio este tipo de fuente está justificada como la mejor opción para el diseño de este módulo del Interfaz de Comunicación.

Diseñar otro tipo de fuente no está aconsejado para las necesidades reales de este proyecto.

Elegida el tipo de fuente a diseñar, pasaremos a explicar las partes más importantes que la forman.

Podemos ver en la Figura 5.1.2-2, las distintas etapas básicas y sus señales de salida.

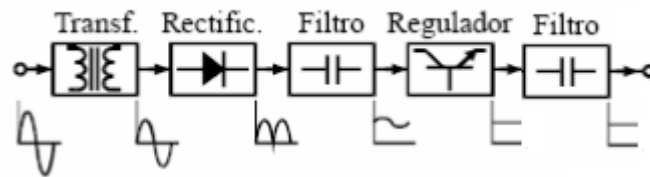


Figura 5.1.2-2. Etapas básicas de una fuente de alimentación rectificada y sus señales de salida.

A continuación, en la Figura 5.1.2-3, mostraremos el esquema del diseño de nuestra fuente de alimentación.

Antes atenderemos a explicar las partes más importantes de esta y la justificación de la elección de los componentes a la hora del diseño.

La fuente diseñada, para nuestra aplicación, atiende a unos consumos bajos, del orden de 600 mA. Por ello, en el conector J1 necesitaremos conectar una fuente AC o DC de 9 V sin regular de al menos 1 A.

También tenemos la opción de conectar una pila de 9 V en el conector J2, esta batería tendría una doble función, por un lado puede hacer el sistema portable, o podría ser una batería de respaldo en caso de corte en el suministro eléctrico o bajada de tensión en el mismo.

Si miramos el esquema de la figura 5.1.2-3, este atiende a una fuente de alimentación básica, donde el núcleo de dicha fuente es un viejo conocido en el mundo de la electrónica, LM 7805. Se puede consultar las hojas de características técnicas de este componente en el CD adjunto.

La principal función es la de regular una tensión de salida fijada en 5 V, con un margen de tensiones de entrada. De esta manera vamos a asegurar una tensión correcta para todos los circuitos del sistema Interfaz de Comunicación.

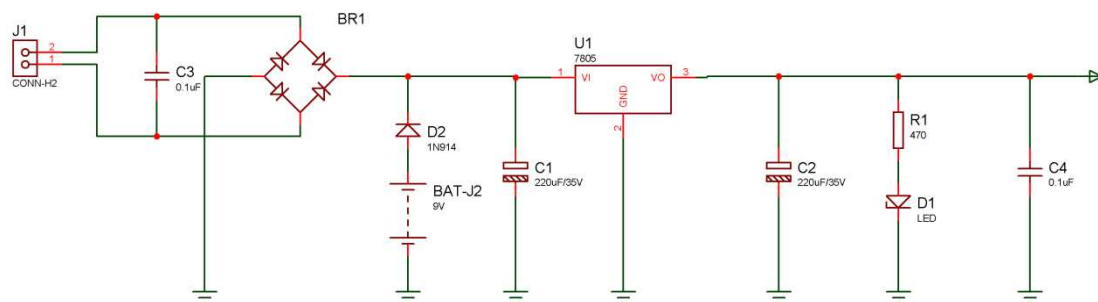


Figura 5.1.2-3. Fuente de alimentación rectificada de 5 V.

Los demás componentes discretos son los mínimos necesarios para el diseño de nuestra fuente de alimentación.

Un condensador de entrada para filtrar la señal, un puente de diodos para filtrar las componentes alternas, y condensadores de entrada y salida del regulador de tensión para corregir el nivel de rizado.

El diodo LED D1, es un indicativo visual del correcto funcionamiento de este módulo.

En el esquema, módulo fuente de alimentación, no se ha incluido para el mismo el conector J2, por ser una función opcional.

2. Comunicaciones Rs232

Para el diseño de este módulo se ha elegido la utilización de un clásico en la conversión de tensiones, el MAX232, recordemos que el funcionamiento de este módulo consiste en la adaptación de señales TTL entregadas por el PIC en sus puertos de entrada y de salida (RX, TX) y la adaptación de dichas señales para el uso en un puerto de comunicaciones RS232.

Este tipo de conexión es asíncrona, lo que significa que no se utiliza una línea especial para transmitir la señal de reloj. Puesto que se utiliza sólo una línea de comunicación, tanto el receptor como el transmisor reciben y envían los datos a misma velocidad que ha sido predefinida para mantener la sincronización necesaria.

El corazón de dicho módulo, al que ya hemos hecho referencia, es el MAX232, un conversor de tensiones muy utilizado en electrónica y en especial en el manejo del puerto RS232 y PIC's.

Podemos ver el esquema de conexiones en la siguiente figura 5.1.2-4.

Toda la electrónica discreta que lo acompaña, sirve para adaptar tensiones para el correcto funcionamiento del MAX232, tal y como figura en la hoja de características del fabricante.

Se puede consultar las hojas de características técnicas de este componente en el CD adjunto.

- Un multiplexor (3 a 8), para poder seleccionar uno de los 5 registros, salidas Y0 a la Y4. (74LS138) Se puede consultar las hojas de características técnicas de este componente en el CD adjunto.
- Cinco registros tipo D de cuatro entradas, (d0, d1, d2, d3), donde almacenaremos el dato enviado por el PIC, en formato BCD. (74LS75) Se puede consultar las hojas de características técnicas de este componente en el CD adjunto.
- Un inversor, ya que la salida del multiplexor esta negada. (74LS04) Se puede consultar las hojas de características técnicas de este componente en el CD adjunto.

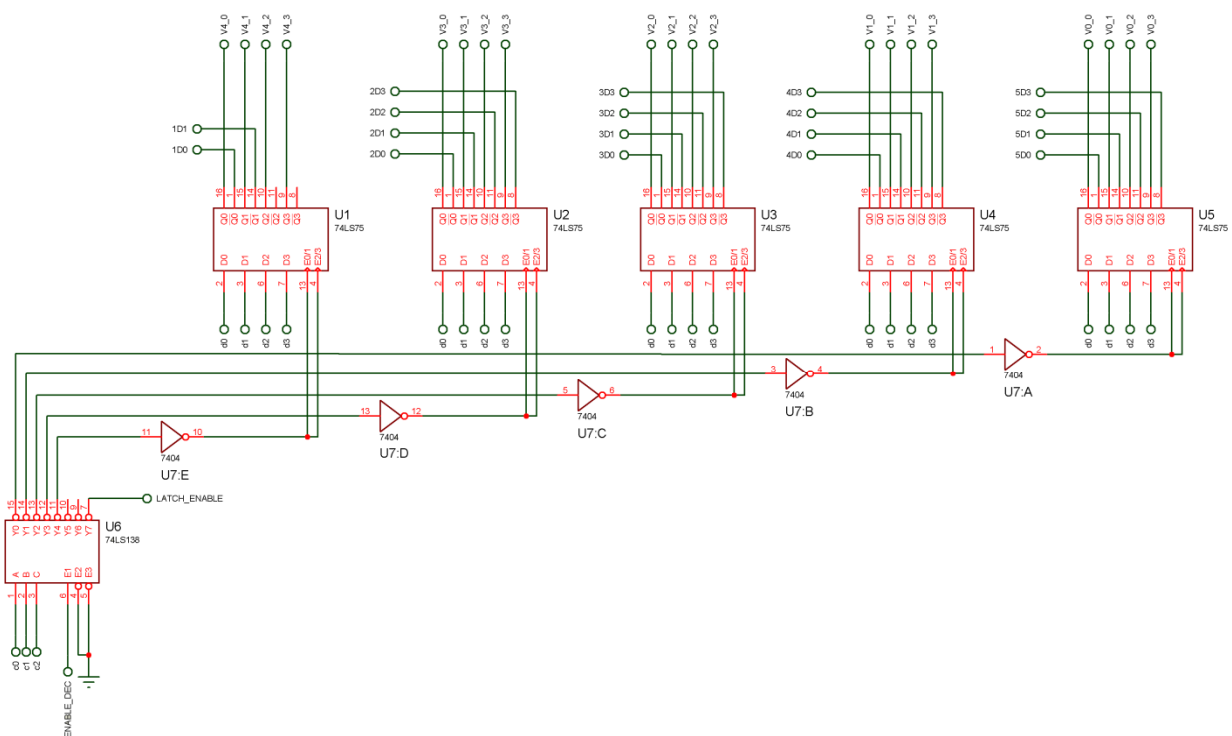


Figura 5.1.2-5. Selector de registro de memoria tipo D.

La salida Y7 se utilizara para enviar una señal de LATCH_ENABLE al sintetizador indicando que ya puede leer los datos del conector de entrada.

La entrada ENABLE_DEC estará conectada al PIC, quien será el que activará la señal para hacer funcionar el multiplexor únicamente cuando tengamos el dato a la salida del puerto del PIC. En ese instante activamos esta señal para cargar dicho dato en el registro adecuado, quedando la información almacenada, para posteriormente mostrarla en el display y enviarla al sintetizador.

El funcionamiento de dicho módulo es el siguiente:

Por un lado tenemos la entrada de datos (d0-d1-d2-d3). Esta información llega directamente de uno de los puertos del PIC (B0, B1, B2, B3).

Como podemos observar, llega a todos los registros a la vez a través de un bus de datos. La selección se realiza mediante la señal E0/1 y E2/3, vamos a poder activarlo cuando queramos que carguen los datos en el registro correspondiente o los ignore.

El multiplexor (3 a 8) recibirá de la salida del puerto (B4, B5, B6) del PIC un código (0, 1, 2, 3, 4) en binario para indicar cuál de los 5 biestables tiene que aceptar los datos que tiene a su entrada.

La salida (7) del multiplexor, LATCH_ENABLE, se envía al conector Amphenol, para indicar al sintetizador que los datos que tiene a su entrada son correctos.

La entrada ENABLE_COD conectada al bus del PIC (B7), nos servirá para poner primero la dirección del registro donde queramos escribir y luego activar dicha señal sin producir errores en la carga de los datos.

4. Módulo PIC

Se trata del corazón del sistema, se va a encargar de interpretar todas las señales recibidas, y dar las órdenes adecuadas.

Este módulo llevará a demás del Microcontrolador PIC16F886, los componentes discretos asociados a la configuración según el fabricante. (16F886) Se puede consultar las hojas de características técnicas de este componente en el CD adjunto.



Figura 5.1.2-6. Detalle del chip PIC16F886.

El PIC16F886 es un producto de la compañía Microchip®. Figura 5.1.2-6.

Incorpora todos los componentes disponibles en la mayoría de los microcontroladores modernos.

Por su bajo precio, un rango amplio de aplicaciones, alta calidad y disponibilidad, es una solución perfecta para controlar diferentes procesos en la

industria, dispositivos de control, medición de variables de procesos, etc. Por estas singularidades es el microcontrolador apropiado para el Interfaz de Comunicación.

Algunas de sus características principales se enumeran a continuación:

- **Arquitectura RISC**
 - o El microcontrolador cuenta con solo 35 instrucciones diferentes
 - o Todas las instrucciones son uni-ciclo excepto las de ramificación
- **Frecuencia de operación 0-20 Mhz**
- **Oscilador interno de alta precisión**
 - o Calibrado de fábrica
 - o Rango de frecuencia de 8 Mhz a 31 Khz seleccionado por software
- **Voltaje de la fuente de alimentación de 2.0 V a 5.5 V**
 - o Consumo: 220 uA (2.0 V, 4 Mhz), 11 uA (2.0 V, 32 Khz), 50 nA (en modo de espera)
- **Ahorro de energía en el Modo de reposo**
- **Brown-out Reset (BOR) con opción para controlar por software**
- **35 pines de entrada/salida**
 - o Alta corriente de fuente y de drenador para manejo de LED
 - o Resistencias pull-up programables individualmente por software
 - o Interrupción al cambiar el estado del pin
- **Memoria ROM de 8K con tecnología FLASH**
 - o El chip se puede re-programar hasta 100.000 veces
- **Opción de programación serial en el circuito**
 - o El chip se puede programar incluso incorporado en el dispositivo destino
- **256 bytes de memoria EEPROM**
 - o Los datos se pueden grabar más de 1.000.000 veces
- **368 bytes de memoria RAM**
- **Convertidor A/D:**
 - o 14 canales
 - o Resolución de 10 bits
- **3 temporizadores/contadores independientes**
- **Temporizador perro guardián**
- **Módulo comparador analógico con:**
 - o Dos comparadores analógicos
 - o Referencia de voltaje fija (0.6 V)

- Referencia de voltaje programable en el chip
- **Módulo PWM incorporado**
- **Módulo USART mejorado**
 - Soporta las comunicaciones seriales RS-485, RS-232 y LIN2.0
 - Auto detección de baudios
- **Puerto Serie Síncrono Maestro (MSSP)**
 - Soporta los modos SPI e I2C

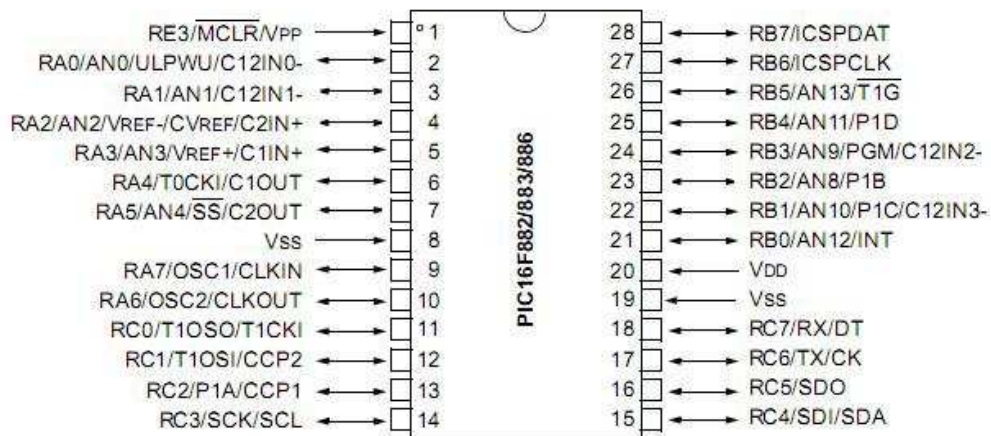


Figura 5.1.2-7. Detalle de los pines del chip PIC16F886.

En la Figura 5.1.2-7, podemos ver la disposición de los pines de nuestro PIC, a continuación vamos a proceder a la descripción de pines más importantes.

La mayoría de los pines del microcontrolador PIC16F886 son multipropósito como se muestra en la figura anterior. Por ejemplo, la asignación RA3/AN3/Vref+/C1IN+ para el quinto pin del microcontrolador indica que éste dispone de las siguientes funciones:

RA3 Tercera entrada/salida digital del puerto A

AN3 Tercera entrada analógica

Vref+ Referencia positiva de voltaje

C1IN+ Entrada positiva del comparador C1

La funcionalidad de los pines presentados anteriormente es muy útil puesto que permite un mejor aprovechamiento de los recursos del microcontrolador sin afectar a su funcionamiento. Estas funciones de los pines no se pueden utilizar simultáneamente, sin embargo se pueden cambiar en cualquier instante durante el funcionamiento.

En las siguientes tablas (figuras 5.1.2-8 y 5.1.2-9) podemos observar una relación con el nombre, función y descripción de los principales pines de PIC.

Nombre	Función	Descripción
RE3/MCLR/Vpp	RE3	Entrada de propósito general en el puerto PORTE
	MCLR	Pin de reinicio. El nivel lógico bajo en este pin reinicia al microcontrolador
	Vpp	Voltaje de programación
RA0/AN0/ULPWU/C12IN0-	RA0	E/S de propósito general en el puerto PORTA
	AN0	Entrada del canal 0 del convertidor A/D
	ULPWU	Entrada de desactivar el modo de espera
	C12IN0-	Entrada negativa del comparador C1 o C2
RA1/AN1/C12IN1-	RA1	E/S de propósito general en el puerto A
	AN1	Canal 1 del convertidor A/D
	C12IN1-	Entrada negativa del comparador C1 o C2
RA2/AN2/Vref-/CVref/C2IN+	RA2	E/S de propósito general en el puerto PORTA
	AN2	Canal 2 del convertidor A/D
	Vref-	Entrada de referencia negativa de voltaje del convertidor A/D
	CVref	Salida de referencia de voltaje del comparador
	C2IN+	Entrada positiva del comparador C2
RA3/AN3/Vref+/C1IN+	RA3	E/S de propósito general en el puerto PORTA
	AN3	Canal 3 del convertidor A/D
	Vref+	Entrada de referencia positiva de voltaje del convertidor A/D
	C1IN+	Entrada positiva del comparador C1
RA4/T0CKI/C1OUT	RA4	E/S de propósito general en el puerto PORTA
	T0CKI	Entrada de reloj del temporizador T0
	C1OUT	Salida del comparador C1
RA5/AN4/SS/C2OUT	RA5	E/S de propósito general en el puerto PORTA
	AN4	Canal 4 del convertidor A/D
	SS	Entrada del módulo SPI (<i>Selección del esclavo</i>)
	C2OUT	Salida del comparador C2
RE0/AN5	RE0	E/S de propósito general en el puerto PORTE
	AN5	Canal 5 del convertidor A/D
RE1/AN6	RE1	E/S de propósito general en el puerto PORTE
	AN6	Canal 6 del convertidor A/D
RE2/AN7	RE2	E/S de propósito general en el puerto PORTE
	AN7	Canal 7 del convertidor A/D
Vdd	+	Suministro de voltaje positivo
Vss	-	Tierra (ground - GND)

Figura 5.1.2-8. Tabla de las funciones principales de los pines del PIC.

Nombre	Función	Descripción
RA7/OSC1/CLKIN	RA7	E/S de propósito general en el puerto PORTA
	OSC1	Entrada del oscilador de cristal
	CLKIN	Entrada del reloj externo
RA6/OSC2/CLKOUT	OSC2	Salida del oscilador del cristal
	CLKO	Salida en la que se presenta la señal Fosc/4
	RA6	E/S de propósito general en el puerto PORTA
RC0/T1OSO/T1CKI	RC0	E/S de propósito general en el puerto PORTC
	T1OSO	Salida del oscilador del temporizador 1
	T1CKI	Entrada de reloj del temporizador 1
RC1/T1OSO/T1CKI	RC1	E/S de propósito general en el puerto PORTC
	T1OSI	Entrada del oscilador del temporizador 1
	CCP2	E/S de los módulos CCP1 y PWM1
RC2/P1A/CCP1	RC2	E/S de propósito general en el puerto PORTC
	P1A	Salida del módulo PWM
	CCP1	E/S de los módulos CCP1 y PWM1
RC3/SCK/SCL	RC3	E/S de propósito general en el puerto PORTC
	SCK	E/S de reloj del módulo MSSP en el modo SPI
	SCL	E/S de reloj del módulo MSSP en el modo I ² C
RD0	RD0	E/S de propósito general en el puerto PORTD
RD1	RD1	E/S de propósito general en el puerto PORTD
RD2	RD2	E/S de propósito general en el puerto PORTD
RD3	RD3	E/S de propósito general en el puerto PORTD
RC4/SDI/SDA	RC4	E/S de propósito general en el puerto PORTC
	SDI	Entrada <i>Data</i> del módulo MSSP en el modo SPI
	SDA	E/S <i>Data</i> del módulo MSSP en el modo I ² C
RC5/SDO	RC5	E/S de propósito general en el puerto PORTC
	SDO	Salida <i>Data</i> del módulo MSSP en el modo SPI
RC6/TX/CK	RC6	E/S de propósito general en el puerto PORTC
	TX	Salida asíncrona del módulo USART
	CK	Reloj síncrono del módulo USART
RC7/RX/DT	RC7	E/S de propósito general en el puerto PORTC
	RX	Entrada asíncrona del módulo USART
	DT	Datos del módulo USART en modo síncrono

Figura 5.1.2-9. Continuación tabla de las funciones principales de los pines del PIC.

Ya que el PIC es el centro de control de nuestro proyecto, vamos a proceder a explicar de forma clara y concisa la funcionalidad del mismo.

Para ello, mostraremos un diagrama de bloques, ver figura 5.1.2-11, con las partes más importantes que lo componen, así como los recursos de los que vamos a disponer, sin profundizar más que lo imprescindible en el funcionamiento de la CPU.

Los módulos principales que acompañan a la CPU, son memoria ROM, memoria RAM o registros del sistema, Oscilador, Temporizador y Puertos E/S, ver Figura 5.1.2-10.

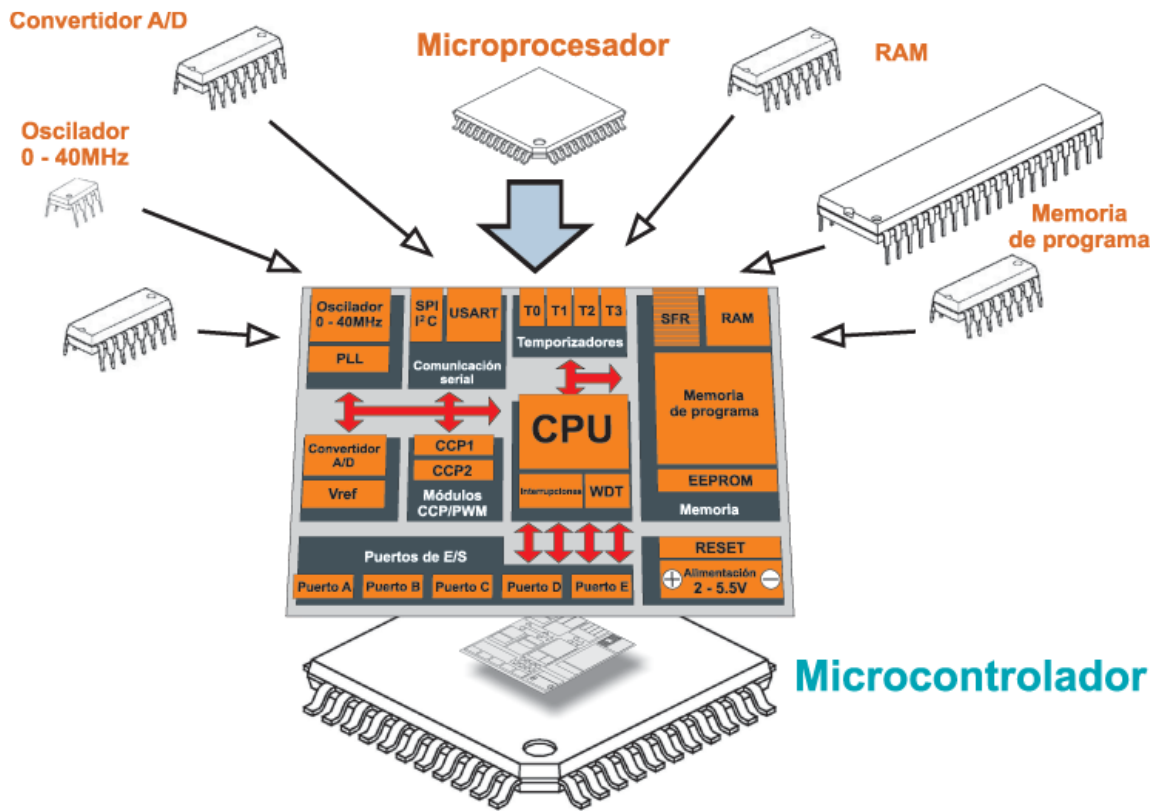


Figura 5.1.2-10. Módulos principales que acompañan al microcontrolador PIC.

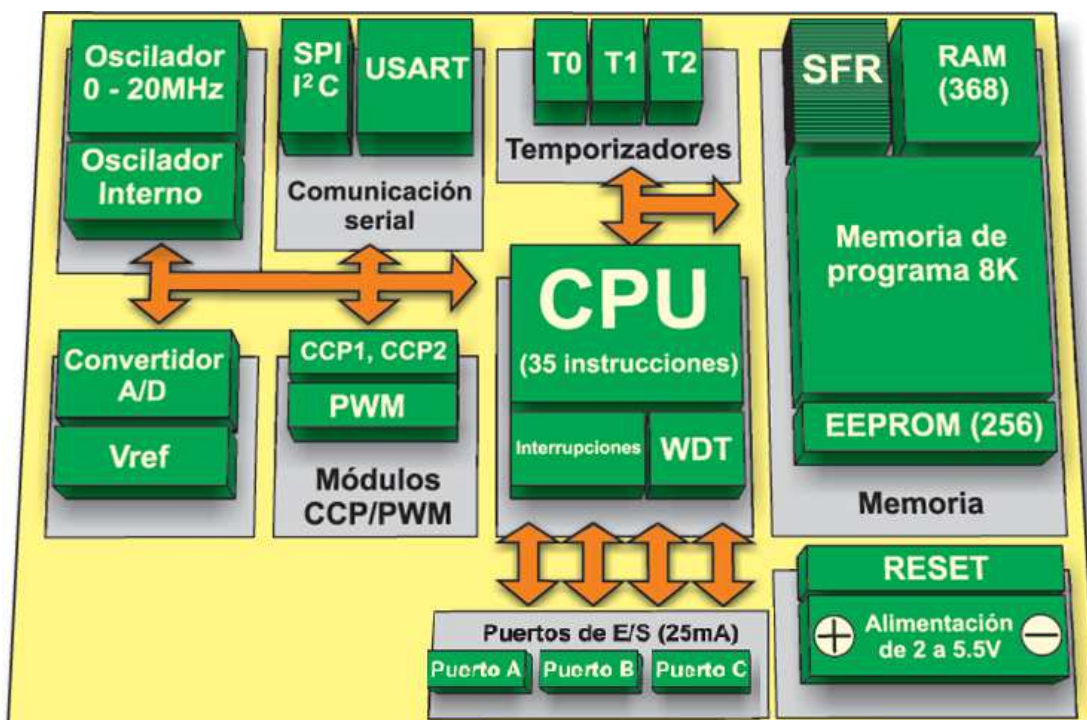
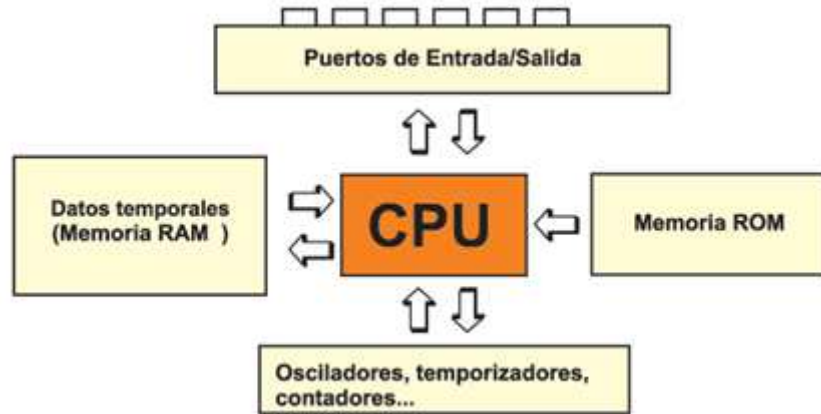


Figura 5.1.2-11. Diagramas de bloques del PIC.

Para el desarrollo del proyecto vamos a necesitar los siguientes recursos de nuestro microcontrolador:

4.1. Puertos de entrada/salida (E/S)

Para hacer útil un microcontrolador, hay que conectarlo a un dispositivo externo, o sea, a un periférico. En nuestro caso, estará conectado por un lado a un registro de biestables, con un lógica de control, y por el otro lado por la conexión serie a nuestro adaptador MAX232. Cada microcontrolador tiene uno o más registros (denominados puertos) conectados a los pines en el microcontrolador. Se denominan puertos de

entrada/salida porque se puede cambiar la función de cada pin en cualquier momento mediante software.

Una de las características más importantes del microcontrolador es el número de los pines de entrada/salida, que permite conectarlo con los periféricos.

Con el propósito de sincronizar el funcionamiento de los puertos de E/S con la organización interna del microcontrolador de 8 bits, ellos se agrupan, de manera similar a los registros, en tres puertos denotados con A, B, C.

Todos ellos tienen las siguientes características en común:

- Pines de E/S multifuncionales. Si un pin realiza una de estas funciones, puede ser utilizado como pin de E/S de propósito general.
- Cada puerto tiene su propio registro de control de flujo, o sea el registro TRIS correspondiente: TRISA, TRISB, TRISC etc. lo que determina el comportamiento de bits del puerto, pero no determina su contenido. Al poner a cero un bit del registro TRIS (bit=0), el pin correspondiente del puerto se configurará como una salida. De manera similar, al poner a uno un bit del registro TRIS (bit=1), el pin correspondiente del puerto se configurará como una entrada, ver detalle de funcionamiento en la figura 5.1.2-12.

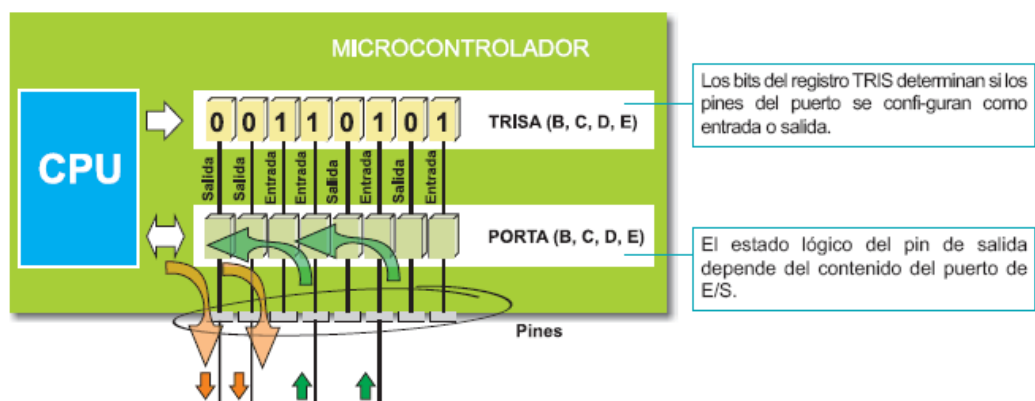


Figura 5.1.2-12. Detalle de funcionamiento del registro TRIS.

Gracias a la tecnología del registro TRIS, a la hora de programar el microcontrolador, podremos definir la función de dichos pines, configurando si son de entrada o salida de datos.

- Una de las características más importantes de los pines de entrada/salida es la corriente máxima que pueden entregar/recibir. En la mayoría de los microcontroladores la corriente obtenida de un pin es suficiente para activar un LED u otro dispositivo de baja corriente (10 - 20 mA).

En nuestro caso son valores suficientes para alimentar una lógica cableada de la familia TTL con la que esta diseñada el proyecto.

- Otra característica importante de los pines es que pueden disponer de los resistores pull-up. Estos resistores conectan los pines al polo positivo del voltaje de la fuente de alimentación y su efecto se puede ver al configurar el pin como una entrada conectada a un interruptor mecánico o a un botón de presión. Actualmente los microcontroladores tienen las resistencias pull-up configurables por software. Esta característica esta presente en el microcontrolador utilizado en nuestro proyecto, es una característica a tener en cuenta en caso de cambiar la tecnología empleada de TTL a CMOS, a la hora de implementar el desarrollo hardware.
- Cada puerto de E/S normalmente está bajo el control de un registro SFR especializado, lo que significa que cada bit de ese registro determina el estado del pin correspondiente en el microcontrolador. Por ejemplo, al escribir un uno lógico (1) en un bit del registro de control (SFR), el pin apropiado del puerto se configura automáticamente como salida. Eso significa que el voltaje llevado a ese pin se puede leer como 0 o 1 lógico. En caso contrario, al escribir 0 al registro SFR, el pin apropiado del puerto se configura como salida. Su voltaje (0 V o 5 V) corresponde al estado del bit apropiado del registro del puerto (0 o 1 lógico, respectivamente).

Para el desarrollo de nuestro Interfaz de Comunicación utilizaremos uno de los puertos E/S digital TTL de los que dispone el PIC, en este caso el RB. Toda la lógica de control y datos de nuestro módulo Interfaz BCD será controlado con este registro.

El puerto RC tiene una función especial en sus pines multipropósito, concretamente en el RC6, RC7, se trata de una conexión serie para envío y recepción de datos, el chip tiene un puerto de comunicaciones serie TX y RX, para enviar y recibir datos, en nuestro caso lo utilizaremos para conectarlo al módulo comunicación RS232.

4.2. Oscilador interno

Para sincronizar todos los procesos que se llevan a cabo dentro del microcontrolador, se debe utilizar una señal de reloj.

Para generar una señal de reloj, se debe utilizar un oscilador. El microcontrolador dispone de varios osciladores capaces de funcionar en modos diferentes.

Como se muestra en la figura 5.1.2-13, la señal de reloj se genera por uno de los dos osciladores integrados.

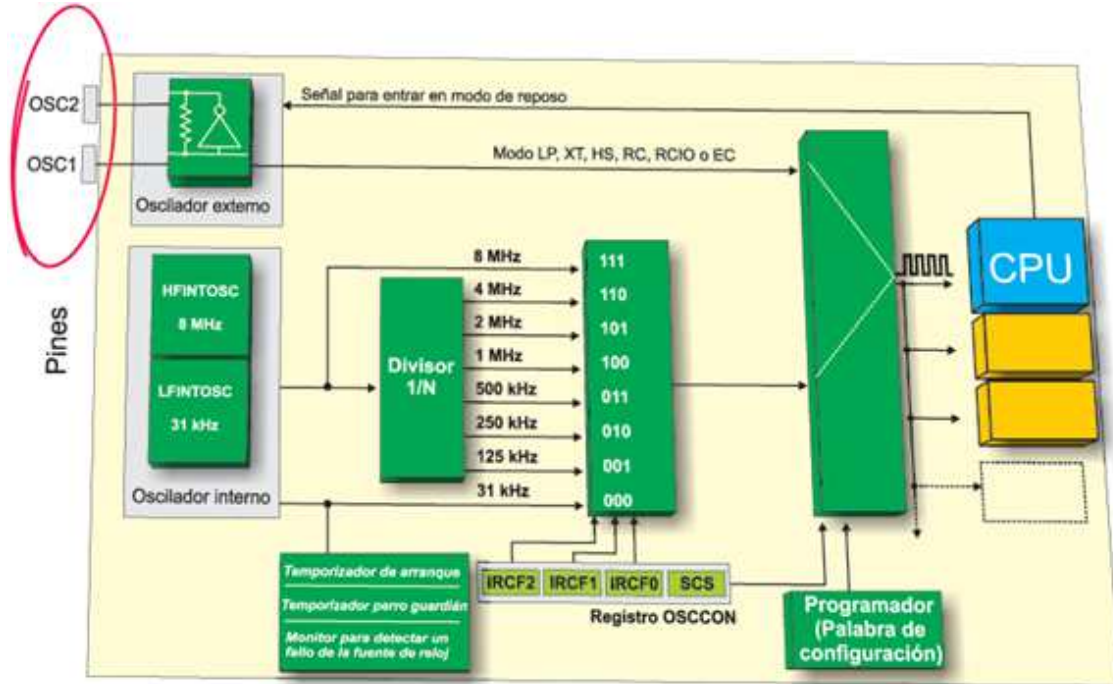


Figura 5.1.2-13. Detalle del oscilador interno, formado por dos osciladores internos separados.

Como se aprecia en el detalle de la figura anterior el oscilador interno esta formado por otros dos osciladores separados:

- **El HFINTOSC**, es un oscilador interno de alta frecuencia calibrado a 8 Mhz. El microcontrolador puede utilizar una señal de reloj generada a esta frecuencia o después de haber sido dividida en el pre-escalador.
- **El LFINTOSC**, es un oscilador interno de baja frecuencia calibrado a 31 Khz. Sus pulsos de reloj se utilizan para funcionamiento de los temporizadores de encendido y perro guardián, asimismo puede utilizarse como fuente de señal de reloj para el funcionamiento de todo el microcontrolador.

4.3. Oscilador externo

Es el encargado de generar los pulsos que permiten el funcionamiento armónico y síncrono de todos los circuitos del microcontrolador. El oscilador se configura normalmente de tal manera que utilice un cristal de cuarzo o resonador cerámico para estabilización de frecuencia, Oscilador Pierce. Además, puede funcionar como un circuito autónomo (como oscilador RC). Es importante decir que las instrucciones del programa no se ejecutan a la velocidad impuesta por el mismo oscilador sino varias veces más despacio. Eso ocurre porque cada instrucción se ejecuta en varios ciclos del oscilador. En algunos microcontroladores se necesita el mismo número de ciclos para

ejecutar todas las instrucciones, mientras que en otros, el tiempo de ejecución no es el mismo para todas las instrucciones.

En nuestro caso, tenemos un microcontrolador con un juego de instrucciones RISC, que es un acrónimo derivado del inglés Reduced Instruction Set Computer, lo que proporciona al PIC16F886 dos grandes ventajas:

- La CPU cuenta con sólo 35 instrucciones simples.
- El tiempo de ejecución es igual para casi todas las instrucciones y tarda 4 ciclos de reloj. La frecuencia del oscilador se estabiliza con un cristal de cuarzo. Las instrucciones de salto y de ramificación tardan ocho ciclos de reloj en ejecutarse. Esto significa que si la velocidad de operación del microcontrolador es 20 Mhz, el tiempo de ejecución de cada instrucción será 200 nS.

En la figura 5.1.2-14, se muestra el diseño del circuito del oscilador externo diseñado en este proyecto, por si a la hora de poner el sistema en el centro emisor, la radiación electromagnética, provocase interferencias en el reloj interno del chip.

La configuración hardware se realiza por medio de unos *jumpers*, JP1 y JP2, podemos utilizar el oscilador externo o dejar actuar al oscilador del propio chip.

Se recomienda no montar este oscilador hasta no haber comprobado un funcionamiento errático con el oscilador interno del chip.

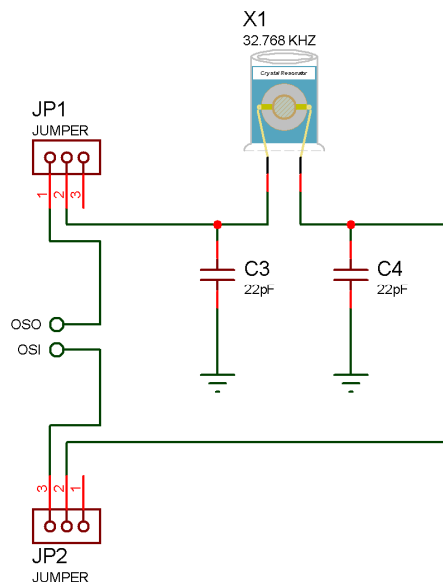


Figura 5.1.2-14. Esquema de oscilador externo.

Los pines RC0/T1OSO y RC1/T1OSI se utilizan para registrar los pulsos que vienen de los dispositivos periféricos, pero también tienen una función adicional.

Como se puede ver en la figura anterior, se configuran simultáneamente como entrada (pin RC1) y salida (pin RC0) del oscilador de cuarzo LP (Low Power - de bajo consumo) adicional.

Este circuito está principalmente diseñado para funcionar a bajas frecuencias (hasta 200 Khz.), exactamente para el uso de cristal de cuarzo de 32.768 Khz. Este cristal se utiliza en los relojes de cristal puesto que es fácil de obtener un pulso de duración de un segundo al dividir esta frecuencia.

Como el oscilador no depende del reloj interno, puede funcionar incluso en el modo de reposo. Se habilita al poner a uno el bit de control T1OSCEN del registro T1CON.

El usuario debe proporcionar tiempo muerto por medio de software (unos pocos milisegundos) para habilitar que el oscilador se inicie apropiadamente.

La siguiente figura 5.1.2-15, podemos ver una tabla que muestra los valores recomendados de los capacitores convenientes con el oscilador de cuarzo. No es necesario que estos valores sean exactos.

De todas formas, la regla general es:

Cuánto más alta sea la capacidad, más alta será la estabilidad, lo que a la vez prolonga el tiempo necesario para la estabilización del oscilador.

OSCILADOR	FRECUENCIA	C 1	C 2
LP	32 Khz	33 pF	33 pF
	100 Khz	15 pF	15 pF
	200 khz	15 pF	15 pF

Figura 5.1.2-15. Tabla de valores recomendados para la configuración del oscilador.

El consumo de corriente del microcontrolador se reduce a nivel más bajo en el modo de reposo ya que el consumidor de corriente principal - el oscilador - no funciona. Es fácil de poner al microcontrolador en este modo - al ejecutar la instrucción SLEEP. El problema es cómo despertar al microcontrolador, o sea cómo generar una interrupción para “despertarlo”. Como el microcontrolador “duerme”, se debe usar una interrupción causada por dispositivos periféricos para despertarlo.

Para resolver el problema, un oscilador de cuarzo LP (de bajo consumo de corriente) completamente independiente, capaz de funcionar en el modo de

reposito, está integrado en el microcontrolador PIC16F886. Simplemente, un circuito anteriormente separado ahora está integrado en el microcontrolador y asignado al temporizador Timer1. En la figura 5.1.2-16, se puede observar que el oscilador está habilitado al poner a 1 el bit T1OSCEN del registro T1CON. El bit TMR1CS del mismo registro se utiliza para habilitar que el temporizador Timer1 utilice secuencias de pulsos de ese oscilador.

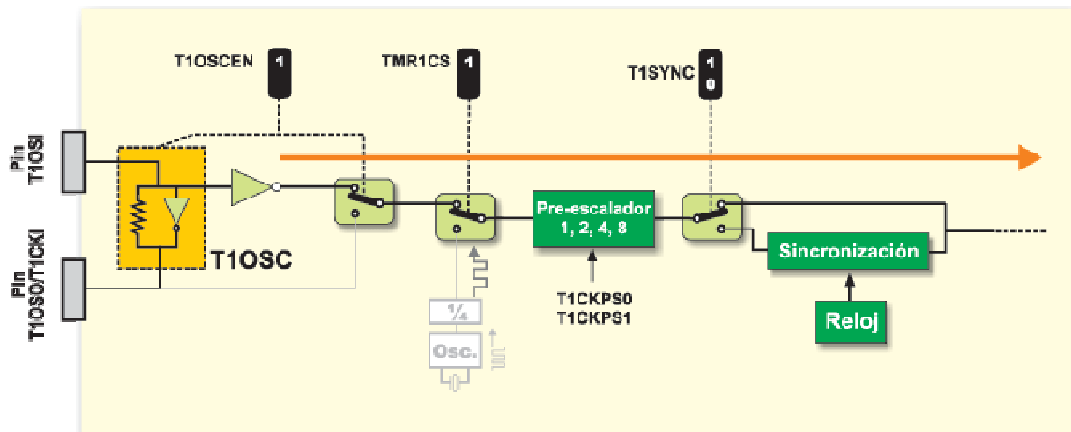


Figura 5.1.2-16. Detalle del circuito temporizador interno del microcontrolador.

Una señal generada por este oscilador de cuarzo está sincronizada con el reloj del microcontrolador al poner a 0 el bit T1SYNC. En este caso, el temporizador no puede funcionar en modo de reposo, porque el circuito para sincronización utiliza el reloj del microcontrolador.

La interrupción por desbordamiento en el registro del temporizador Timer1 puede estar habilitada. Si el bit T1SYNC se pone a 1, tales interrupciones se producirán en el modo de reposo también.

Este sistema para poner el microcontrolador en modo SLEEP, nos va a resultar muy práctico en los tiempos en los que el transmisor no realice ninguna petición de cambio de frecuencia a nuestro interfaz y utilizaremos en la parte Software del sistema Interfaz de Comunicación.

4.4. Temporizador perro guardián (Watchdog)

El perro guardián es un temporizador conectado a un oscilador RC completamente independiente dentro del microcontrolador.

Si el perro guardián está habilitado, cada vez que cuenta hasta el máximo valor en el que ocurre el desbordamiento del registro se genera una señal de reinicio del microcontrolador y la ejecución de programa inicia en la primera

instrucción. Lo importante es evitar que eso ocurra al utilizar el comando adecuado.

La idea se basa en el hecho de que cada programa se ejecuta en varios bucles, más largos o cortos. Si las instrucciones que reinician el temporizador perro guardián se colocan en lugares estratégicos del programa, aparte de los comandos que se ejecutan regularmente, el funcionamiento del perro guardián no afectará a la ejecución del programa. Ver ejemplo de programa en la figura 5.1.2-17. Si por cualquier razón, con frecuencia por los ruidos eléctricos, el contador de programa “se queda atrapado” dentro de un bucle infinito, el valor del registro continuará aumentando, por el temporizador perro guardián, alcanzando el máximo valor, el registro se desbordará ocurriendo el reinicio, el programa será ejecutado desde el principio.

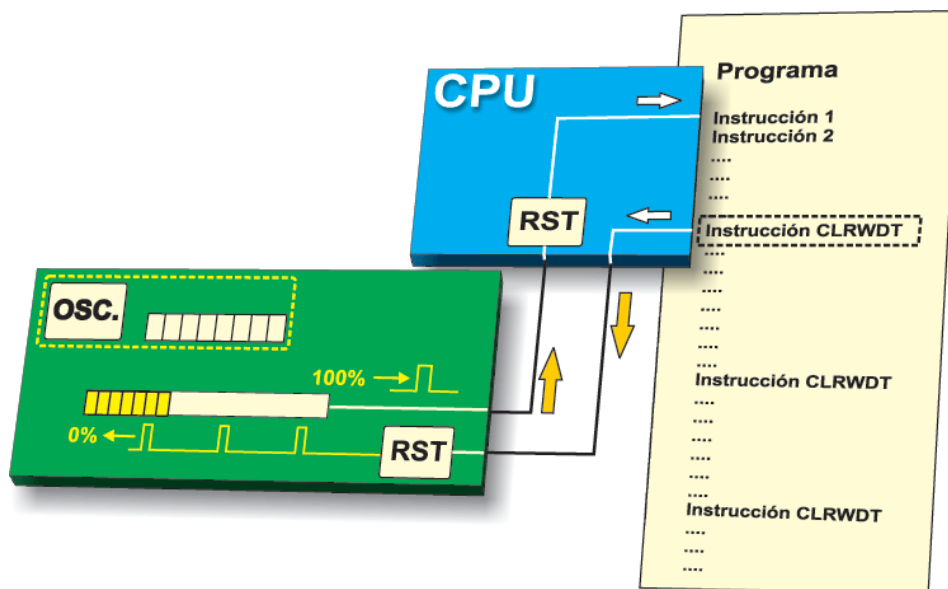


Figura 5.1.2-17. Uso en un programa del temporizador watchdog.

Utilizaremos esta función del Perro guardián o Watchdog, que nos permitirá subsanar cualquier “cuelgue” del PIC y reinicio del sistema, para asegurar un funcionamiento sin supervisión.

4.5. EUSART, adaptador de comunicación serie síncrona y asíncrona

El módulo Transmisor/Receptor Universal Síncrono/Asíncrono mejorado (Enhanced Universal Synchronous Asynchronous Receiver Transmitter - EUSART) es un periférico de comunicación serie de entrada/salida. Asimismo es conocido como interfaz de comunicación serie (Serial Communications Interface - SCI). Contiene todos los generadores de señales de reloj, registros de

desplazamiento y buffer de datos necesarios para realizar transmisiones de datos serie de entrada/salida, independientemente de la ejecución de programa del dispositivo.

El EUSART integrado en el PIC16F886 posee las siguientes características:

- Transmisión y recepción asíncrona en modo Full-duplex
- Caracteres de anchura de 8 – 9 bits programables
- Detección de dirección en modo de 9 bits
- Detección de errores por saturación del búfer de entrada
- Comunicación Half Duplex en modo síncrono

Esta función del microcontrolador, será imprescindible para la comunicación con el módulo Comunicación RS232, su conexión se hace a través del puerto C, en los pines RX y TX.

4.6. Circuito de Reset del PIC

El pin de Reset o reinicio, marcado frecuentemente con MCLR (Master Clear Reset), sirve para el reinicio externo del microcontrolador.

Al aplicar un cero (0) lógico en la entrada MCLR del microcontrolador, reiniciamos el sistema.

En la figura 5.1.2-18, tenemos el diseño del circuito Reset que hemos incorporado al módulo PIC.

El circuito Reset se conecta al pin MCLR del microcontrolador.

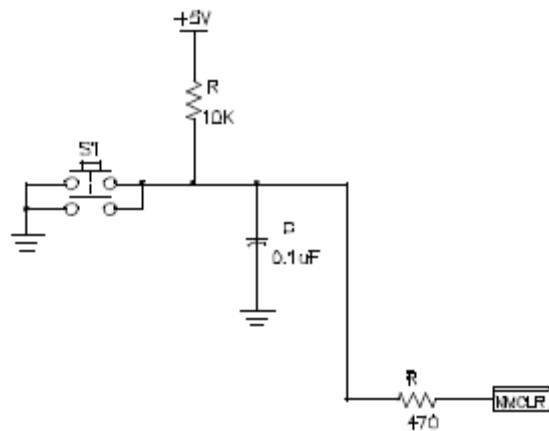


Figura 5.1.2-18. Circuito Reset.

esta a nivel alto el microcontrolador se pone a ejecutar la primera instrucción en el programa.

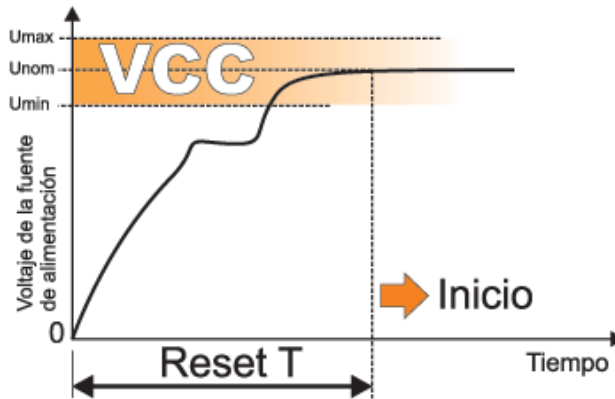


Figura 5.1.2-20. Tiempo de espera de inicio, Reset T.

4.7. El Brown out

Se trata de un estado potencialmente peligroso que ocurre al apagar el microcontrolador o en caso de que el voltaje de la fuente de alimentación se salga de unos márgenes debido al ruido eléctrico.

El microcontrolador dispone de varios circuitos que funcionan a niveles de voltaje diferentes, ese estado puede causar un comportamiento descontrolado.

Para evitarlo, el microcontrolador normalmente tiene un circuito incorporado para el brown out reset.

El circuito reinicia inmediatamente el microcontrolador si el voltaje de alimentación cae por debajo del límite.

Para finalizar con el diseño de este módulo, mostramos a continuación el esquema de conexiones del PIC, con el resto de módulos.

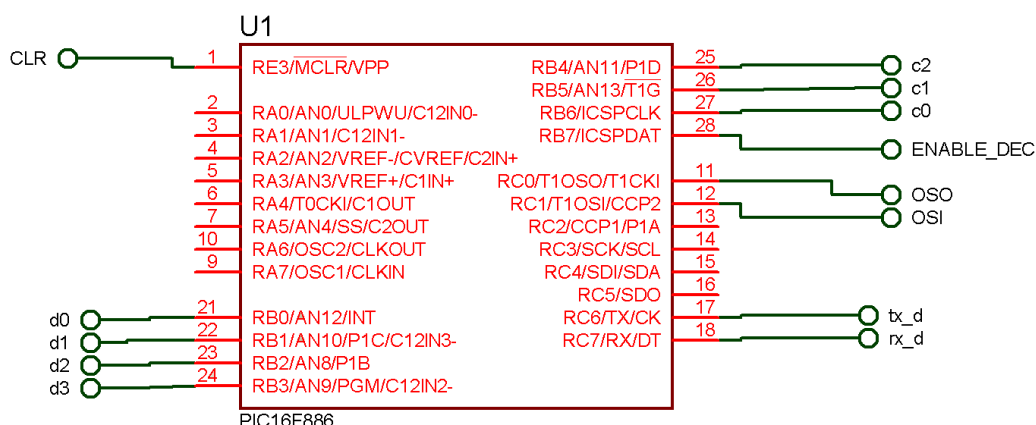


Figura 5.1.2-21. Esquema de conexiones del microcontrolador.

Las aplicaciones prácticas de los recursos del microcontrolador, quedan de manifiesto en el esquema de conexiones, figura 5.1.2-21., donde podemos observar:

- Pin 1: conexión del Circuito Reset. Figura 5.1.2-18.
- Pines 21 al 24: Bus de Datos, conexión con el módulo Interfaz BCD. Configuración del puerto digital E/S mediante el registro TRIS de la figura 5.1.2-12TRIS.
- Pines 25 al 27: Bus de Dirección, conexión con el módulo Interfaz BCD. Configuración del puerto digital E/S mediante el registro TRIS de la figura 5.1.2-12TRIS.
- Pin 28: Señal de Enable_Dec, conexión con el módulo Interfaz BCD. Configuración del puerto digital E/S mediante el registro TRIS de la figura 5.1.2-12TRIS.
- Pines 11 y 12: RC0/T1OSO y RC1/T1OSI, conexión con el oscilador externo figura 5.1.2-14.
- Pines 17 y 18: TX y RX, comunicación serie, conexión con el módulo Comunicación RS232. Figura 5.1.2-4

5. Display Frecuencímetro

Este módulo, se ha incluido para poder mostrar de una manera visual al operador técnico el correcto funcionamiento del sistema.

5.1 Visualizador LED

Un visualizador LED no es nada más que varios diodos LED moldeados en la misma caja plástica.

El visualizador utilizado en este proyecto, es el visualizador de 7 segmentos, ver figura 5.1.2-22.

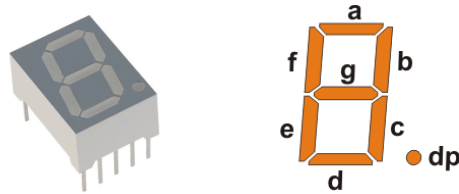


Figura 5.1.2-22. Display 7 segmentos.

Está compuesto por 8 LED's. Los siete segmentos de un dígito están organizados en forma de un rectángulo para visualizar los símbolos, mientras que el segmento adicional se utiliza para el propósito de visualizar los puntos decimales.

Para simplificar la conexión, los ánodos y los cátodos de todos los diodos se conectan al pin común, así que tenemos visualizadores de ánodo común y visualizadores de cátodo común respectivamente.

Los segmentos están etiquetados con letras de a... g y dp, como se muestra en la figura 5.1.2-22. Al conectarlos, cada diodo LED se trata por separado, lo que significa que cada uno necesita su propia resistencia para limitar la corriente.

En la figura 5.1.2-23. Podemos ver el esquema del display frecuencímetro. Esta formado por un conjunto de cinco visualizadores LED de 7 segmentos de cátodo común conectados a un decodificador BCD a 7 segmentos, (74LS48, se puede consultar las hojas de características técnicas de este componente en el CD adjunto), hemos decidido usar cátodo común para así poder aprovechar la salida Q de los registros de la etapa Interfaz BCD, esta salida Q quedaba sin usar ya que el interfaz del sintetizador utiliza lógica negativa usando la salida Q (negado) de los registros, ver Figura 5.1.2-5.

Cambiando la posición del jumper se puede hacer un test a los displays, y así poder saber si funcionan correctamente.

Para visualizar un valor codificado con 4 bits se utilizan displays de 7 segmentos formados por diodos LED's y un circuito decodificador BCD (4 bits) a 7 segmentos (circuito integrado 74LS48).

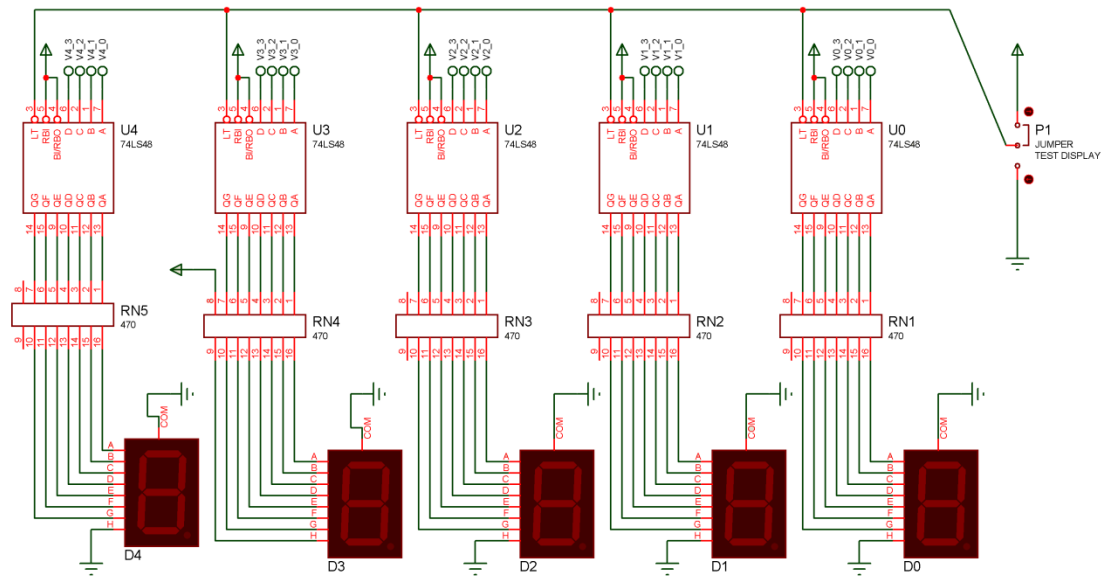


Figura 5.1.2-23. Esquema del display frecuencímetro.

La tecnología empleada es TTL, por lo que se deberá alimentar a +5 V en el terminal marcado como Vcc y a masa en el terminal marcado con GND. La lógica es positiva (+5 V = 1 lógico; 0 V = 0 lógico). Las 7 salidas del decodificador se conectan al segmento del display correspondiente.

El encendido o no de cada uno de los segmentos del display depende de la tabla de verdad del 74LS48, que se puede consultar en las hojas de características técnicas de este componente en el CD adjunto.

La entrada H permite el control del encendido o no del punto decimal del display, en nuestro caso se fija en el segundo display para frecuencias de xx.xxx Khz.

En nuestro caso solo usaremos del 0 al 9 para la representación BCD.

6. Conector Amphenol

Detalle de conexión del Conector Amphenol 57-40500 interfaz con el sintetizador, figura 5.1.2-24.

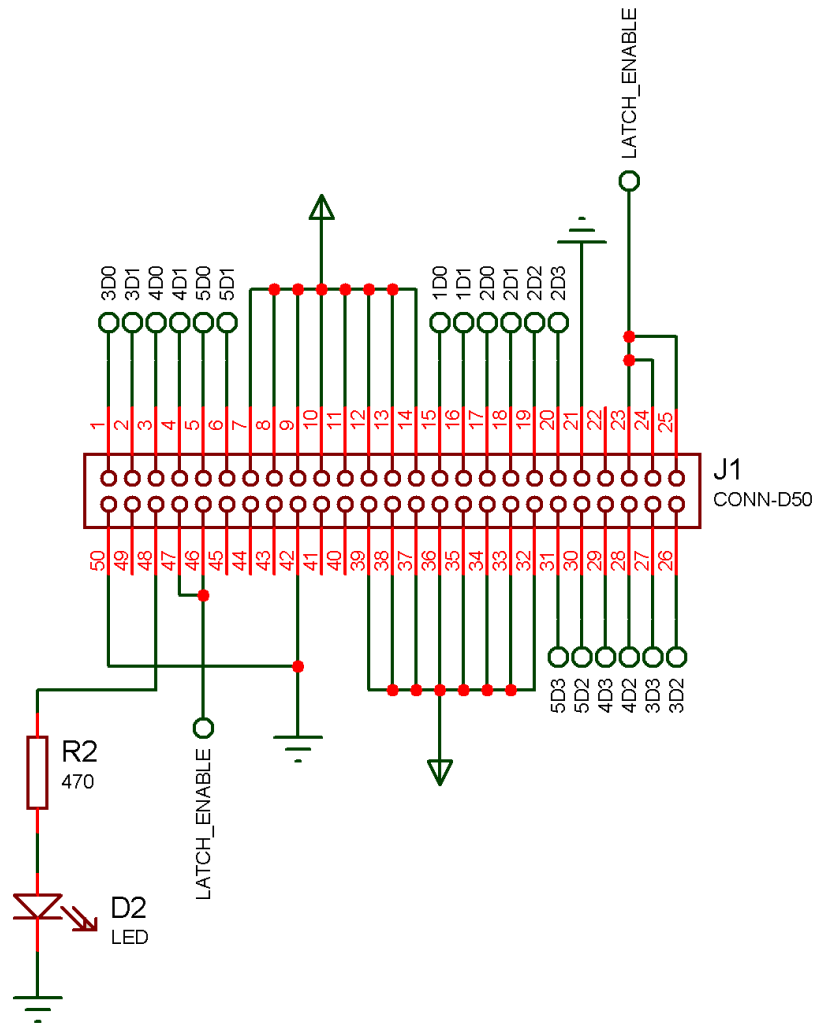


Figura 5.1.2-24. Conexión Amphenol 57-40500.

El diodo LED D1, se ha dispuesto para tener una señal visual de confirmación de funcionamiento de este módulo.

A este conector llegan las siguientes señales:

Salida de datos:

- Señales de datos en BCD:
 - o 1D0, 1D1 (2 bit´s valores 0, 1, 2)
 - o 2D0, 2D1, 2D2, 2D3 (4 bit´s valores 0 al 9)
 - o 3D0, 3D1, 3D2, 3D3 (4 bit´s valores 0 al 9)
 - o 4D0, 4D1, 4D2, 4D3 (4 bit´s valores 0 al 9)
 - o 5D0, 5D1, 5D2, 5D3 (4 bit´s valores 0 al 9)
- Señal de aceptación de datos:
 - o Latch enable (1 bit)

Entrada de datos:

- Pin 48, entrada Vcc. para el control de conexión correcta

El resto de conexiones corresponden a alimentaciones y masas según los datos proporcionados por el fabricante.

5.1.3 ESQUEMA GENERAL DEL INTERFAZ DE COMUNICACIÓN

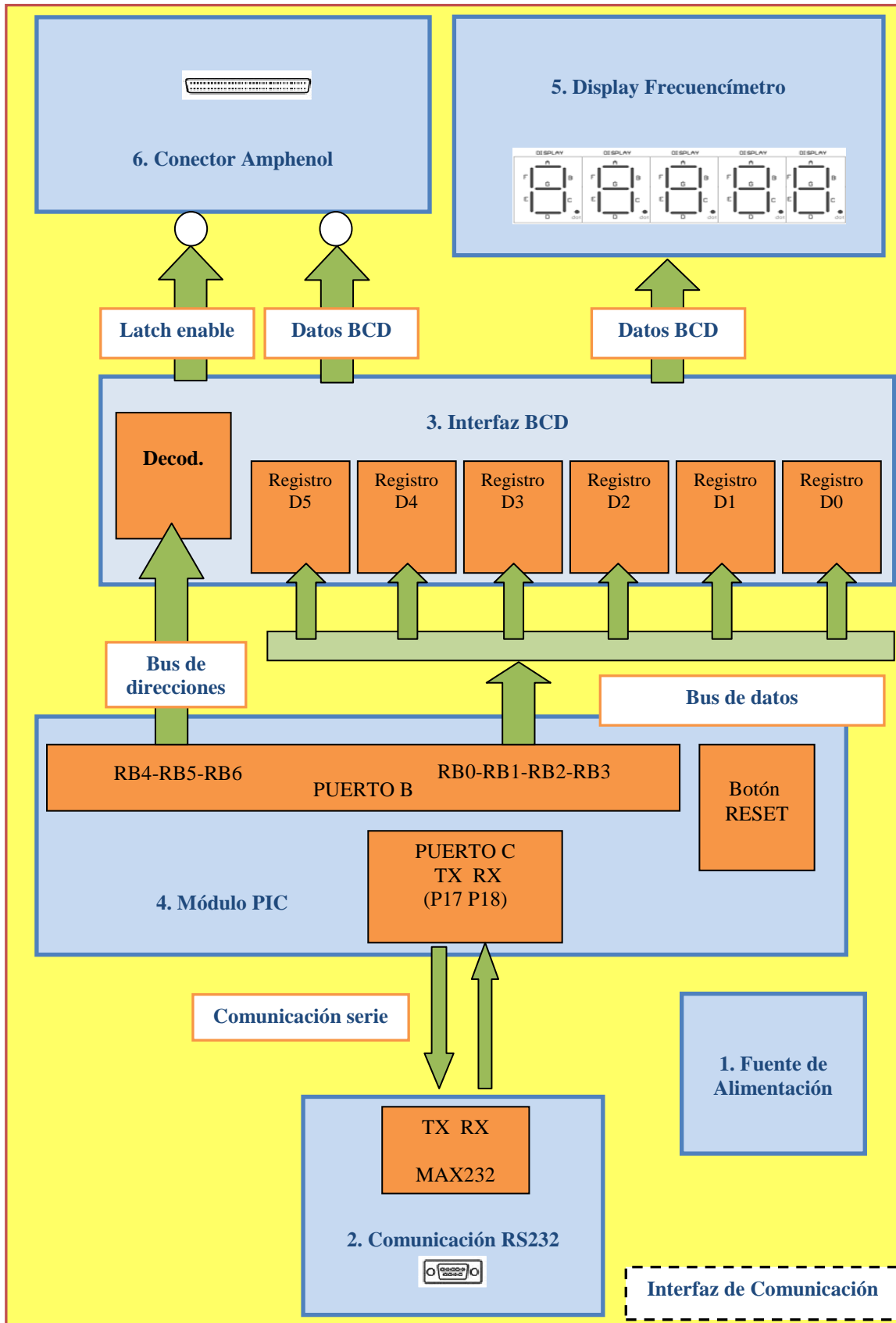
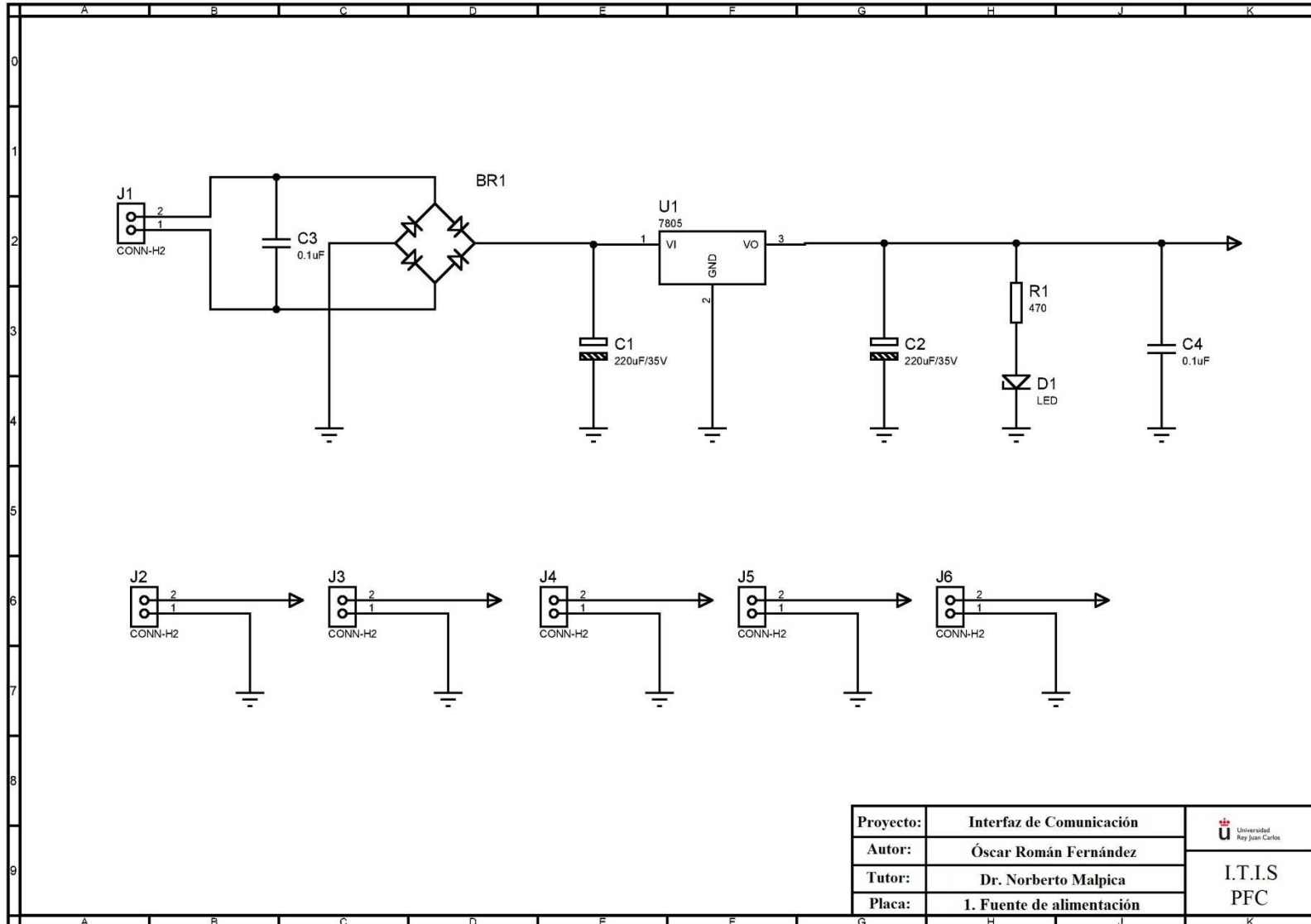
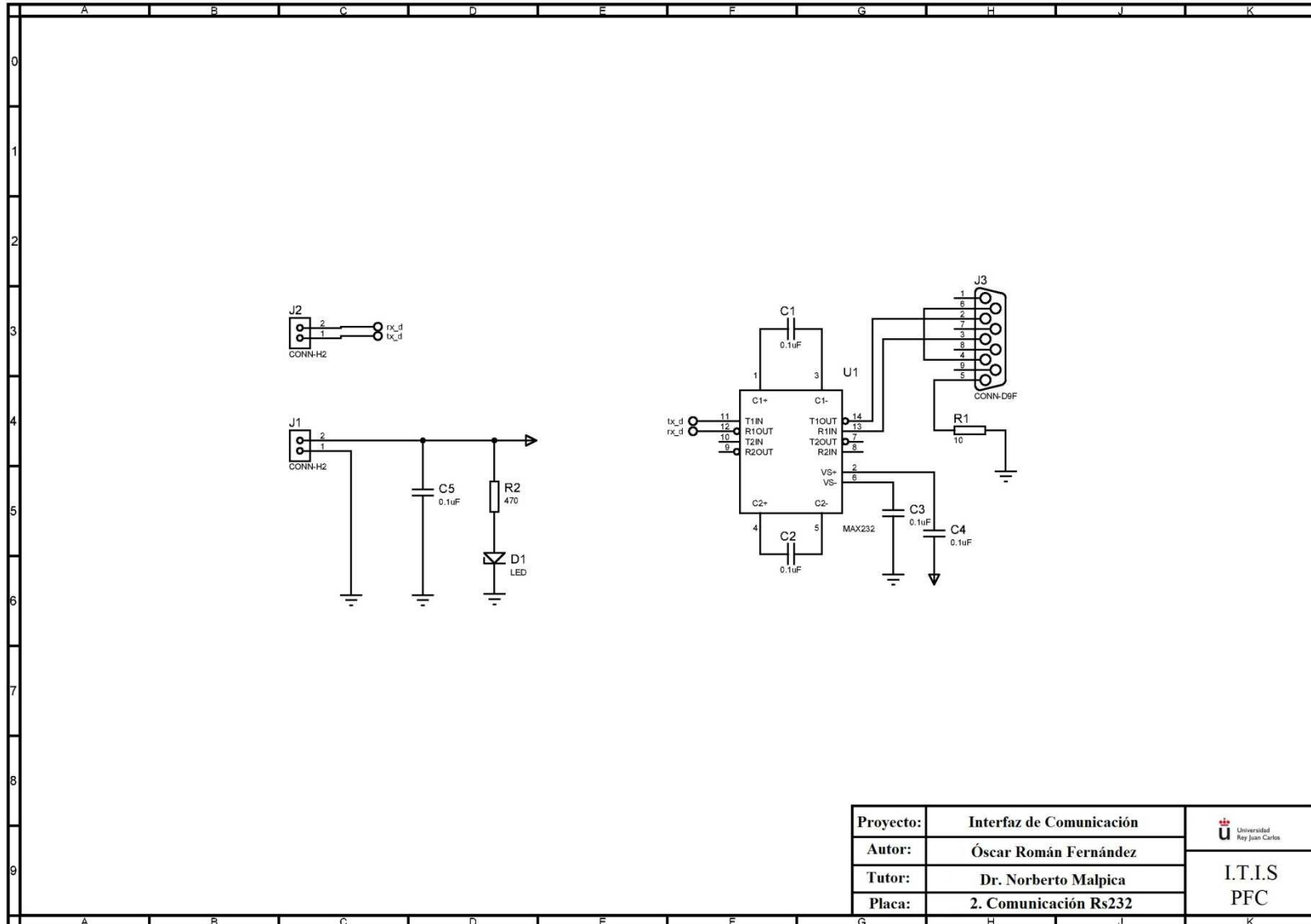


Figura 5.1.3-1. Esquema general del Interfaz de Comunicación.


5.1.4 ESQUEMAS DE LOS CIRCUITOS



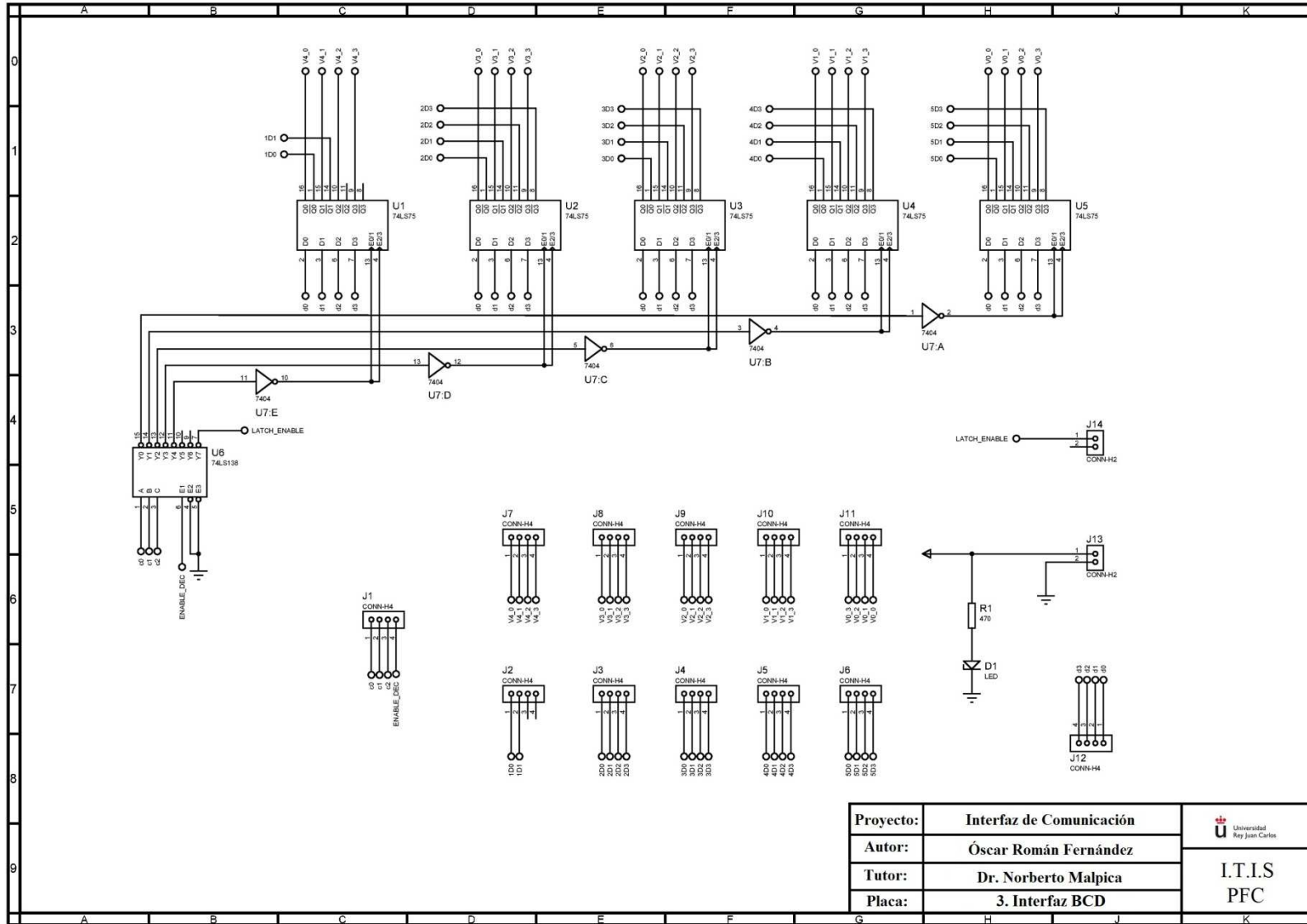
5.1.4 ESQUEMAS DE LOS CIRCUITOS



Proyecto:	Interfaz de Comunicación
Autor:	Óscar Román Fernández
Tutor:	Dr. Norberto Malpica
Placa:	2. Comunicación Rs232

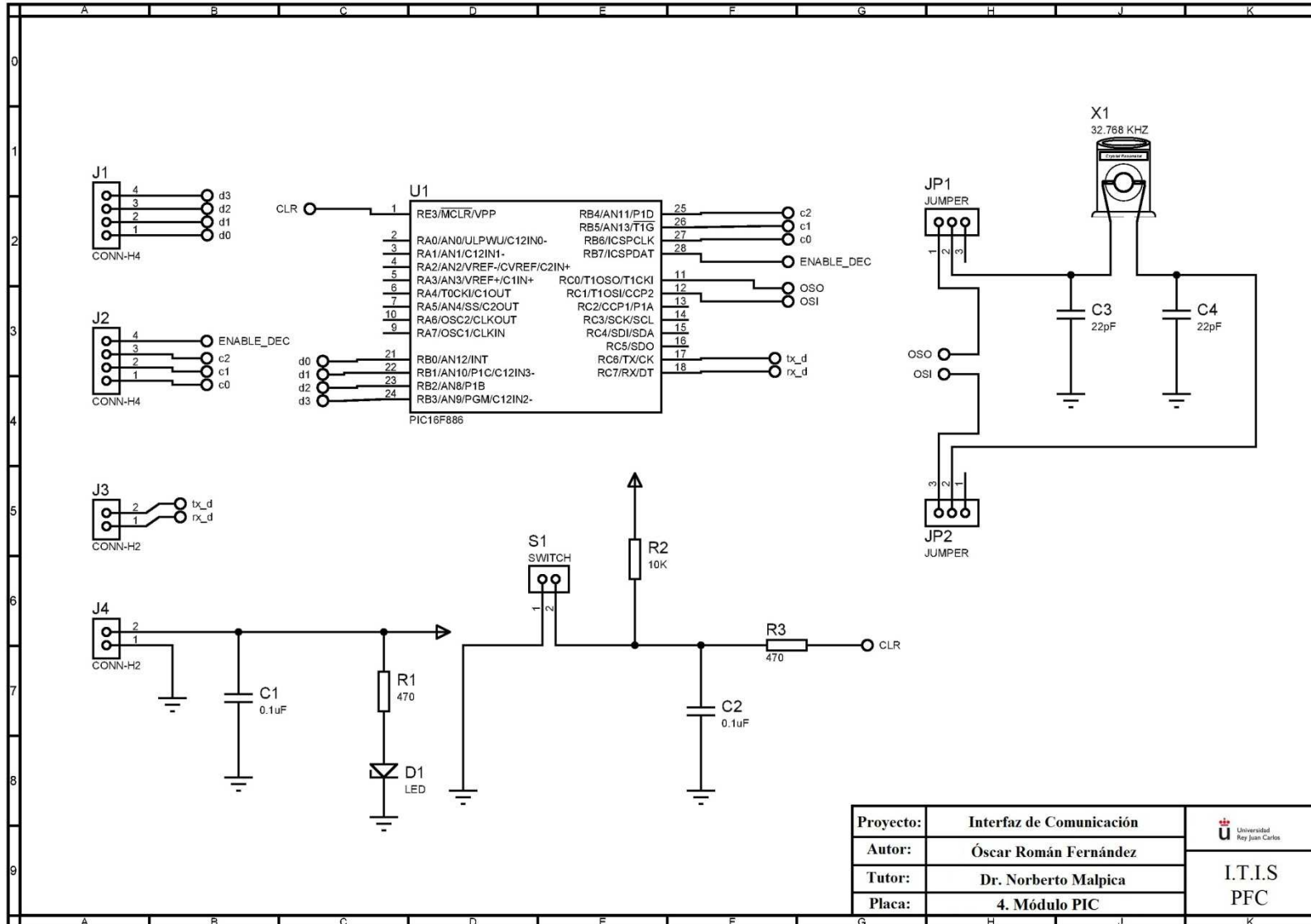

 Universidad Rey Juan Carlos
I.T.I.S
PFC

5.1.4 ESQUEMAS DE LOS CIRCUITOS



Proyecto:	Interfaz de Comunicación	 I.T.I.S PFC
Autor:	Óscar Román Fernández	
Tutor:	Dr. Norberto Malpica	
Placa:	3. Interfaz BCD	

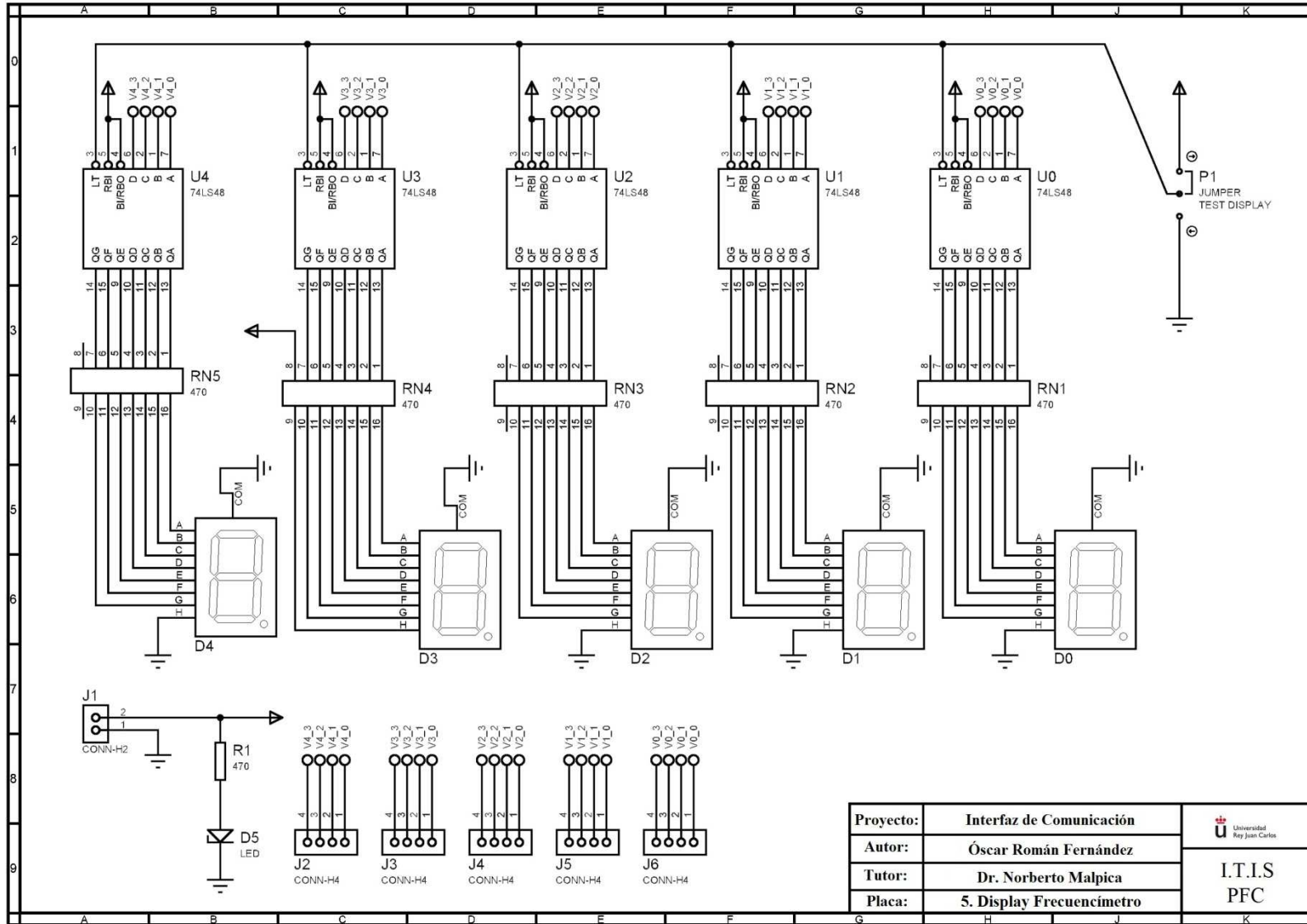
5.1.4 ESQUEMAS DE LOS CIRCUITOS



Proyecto:	Interfaz de Comunicación
Autor:	Óscar Román Fernández
Tutor:	Dr. Norberto Malpica
Placa:	4. Módulo PIC

I.T.I.S PFC

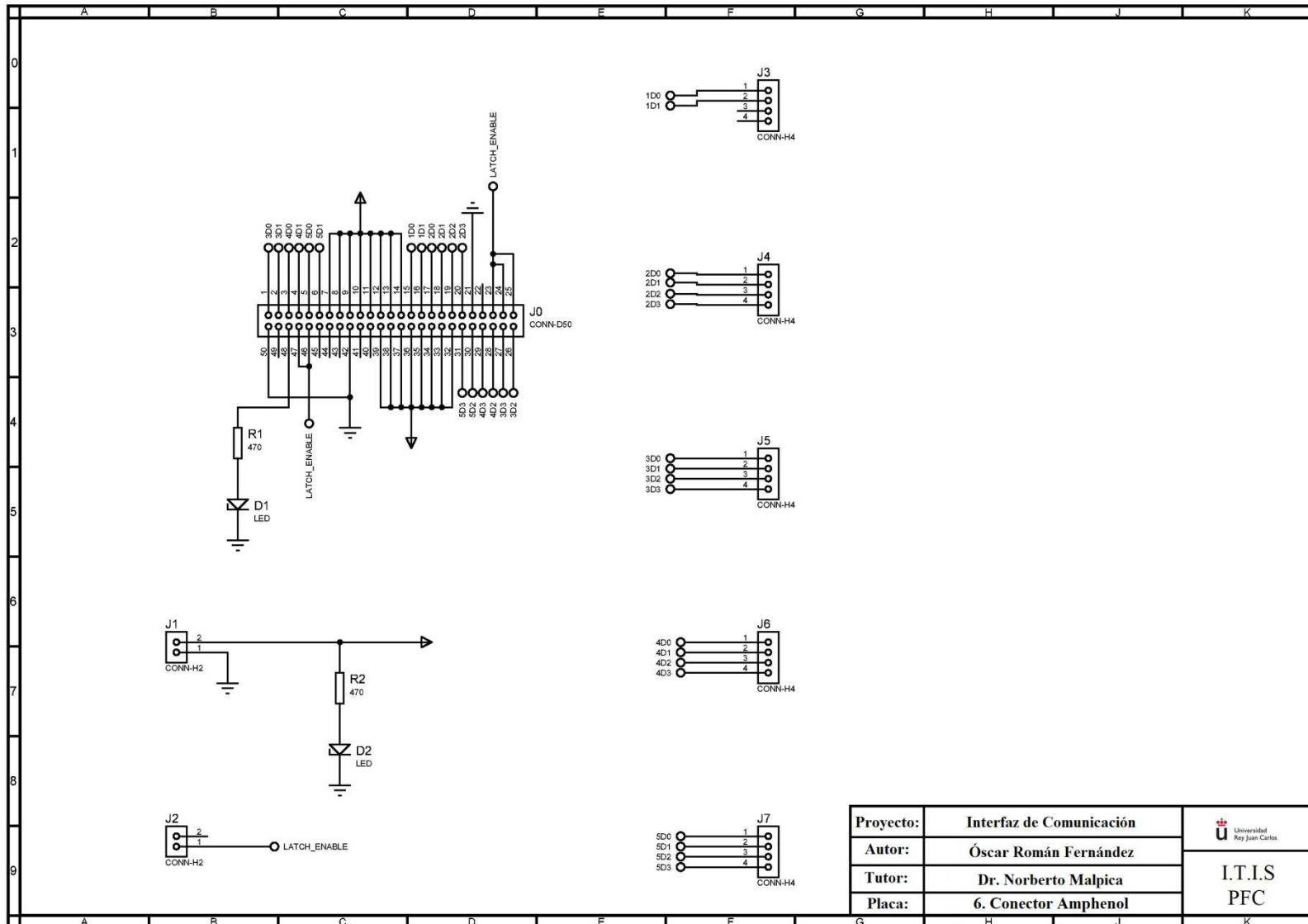
5.1.4 ESQUEMAS DE LOS CIRCUITOS




Proyecto:	Interfaz de Comunicación
Autor:	Óscar Román Fernández
Tutor:	Dr. Norberto Malpica
Placa:	5. Display Frecuencimetro


 Universidad Rey Juan Carlos
I.T.I.S
PFC

5.1.4 ESQUEMAS DE LOS CIRCUITOS



Proyecto:	Interfaz de Comunicación
Autor:	Óscar Román Fernández
Tutor:	Dr. Norberto Malpica
Placa:	6. Conector Amphenol


 Universidad
 Rey Juan Carlos
I.T.I.S
PFC

5.2 SOLUCIÓN SOFTWARE

5.2.1 INTRODUCCIÓN

Una vez terminada la parte del diseño hardware pasaremos a una nueva etapa dentro del diseño final de la solución, el desarrollo software.

El microcontrolador ejecuta el programa cargado en la memoria Flash. El código ejecutable está compuesto por una serie de ceros y unos, aparentemente sin significado. Cada palabra es interpretada por la CPU como una instrucción a ser ejecutada durante el funcionamiento del microcontrolador. Como es más fácil trabajar con el sistema de numeración hexadecimal, el código ejecutable se representa con frecuencia como una serie de los números hexadecimales denominada código hex.

A todas las instrucciones que el microcontrolador puede reconocer y ejecutar se le denominan colectivamente Juego de instrucciones. En nuestro microcontrolador PIC el conjunto de instrucciones tiene 35 diferentes.

En este apartado, nos centraremos en la programación del PIC.

Hay diversas formas de programar el PIC, usando lenguajes de alto nivel tales como BASIC, C, o usando lenguajes de bajo nivel como el Lenguaje Ensamblador.

En este proyecto, se ha optado por usar un lenguaje de programación de alto nivel como es C, ya que nos va a permitir el acceso de las funciones del PIC de una manera más natural.

Entre las características software de nuestro controlador, la mas importante a destacar es la arquitectura RISC (Reduced Instruction Set Computer, computadora con juego de instrucciones reducidas), por lo tanto carece de algunas instrucciones básicas.

Con esta arquitectura el microcontrolador reconoce y ejecuta sólo operaciones básicas (sumar, restar, copiar etc...) Las operaciones más complicadas se realizan al combinar éstas.

Por ejemplo, no hay instrucción apropiada para multiplicar dos números. Por supuesto, este problema se puede resolver gracias a la aritmética que permite realizar las operaciones complejas al descomponerlas en un gran número de operaciones más simples. En este caso, la multiplicación se puede sustituir con facilidad por adición sucesiva ($a \times b = a + a + a + \dots + a$, b veces).

Al utilizar un lenguaje de programación de alto nivel como es C, no hay que preocuparse, ya que el compilador encontrará automáticamente la solución a éste problema y otros similares. Para multiplicar los números a y b, basta con escribir $a*b$.

Si queremos realizar la programación de los microcontroladores PIC en el lenguaje C, es preciso utilizar un compilador de C. Figura: 5.2.1-1.



Figura: 5.2.1-1. Programación de un microcontrolador.

Dicho compilador nos genera ficheros en formato Intel-hexadecimal, archivos hex, que son los necesarios para programar un microcontrolador. Ver figura 5.2.1-2.

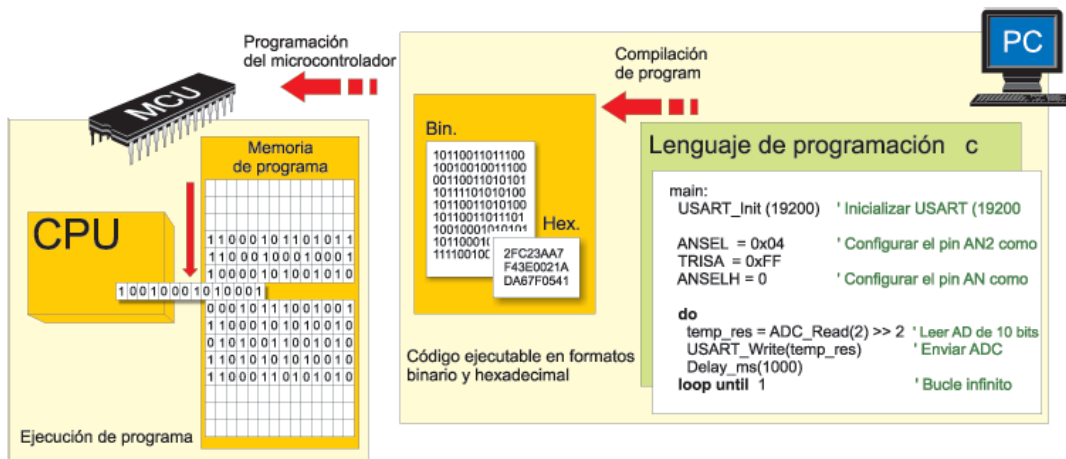


Figura: 5.2.1-2. Secuencia de programación de un microcontrolador.

El compilador de C que vamos a utilizar es el PCW de la casa CCS Inc.

A su vez, el compilador lo integraremos en un entorno de desarrollo integrado (en nuestro caso hemos usado MPLAB IDE) que nos va a permitir 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.

MPLAB-IDE es una Plataforma de Desarrollo Integrada bajo Windows, con múltiples prestaciones, que permite escribir el programa para los PIC en lenguaje de C,

crear proyectos, ensamblar y compilar, simular el programa y finalmente programar el componente.

MPLAB incorpora todas las utilidades necesarias para la realización de cualquier proyecto y, dispone 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.

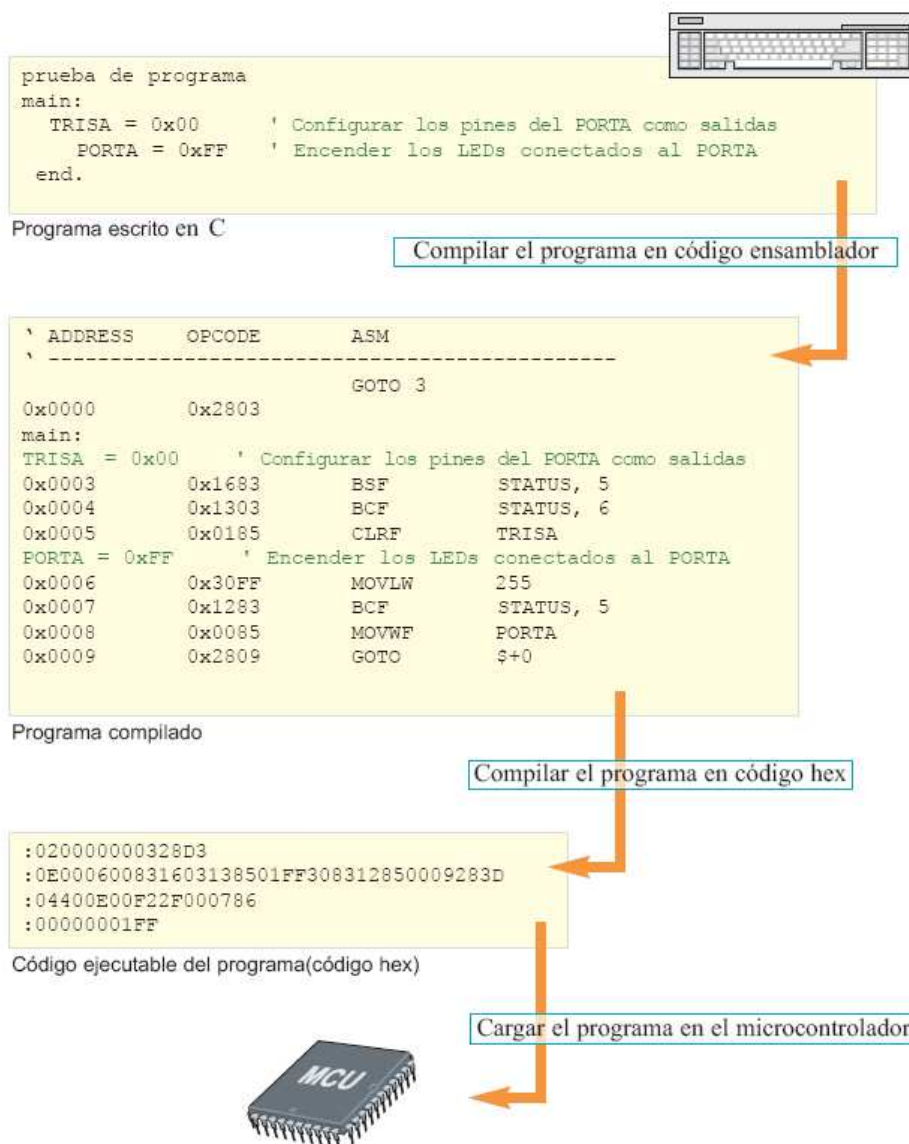


Figura: 5.2.1-3. Compilación del código C.

En la figura 5.2.1-3, puede ver el proceso por el cual el compilador "traduce" el programa escrito en C, código fuente (.C), generando el programa compilado en

lenguaje ensamblador, en la segunda vuelta del compilador, se obtiene el código linker, y de este último finalmente obtendremos el código ejecutable o lenguaje máquina de los microcontroladores, el formato de salida es un archivo hexadecimal (.hex).

5.2.2 DESCRIPCIÓN SOFTWARE

Para poder realizar la comunicación de los dos equipos vamos a proceder a explicar de una manera razonada el protocolo de comunicación y los mensajes que han de enviarse los equipos en cada una de las fases.

En la figura 5.2.2-1, podemos ver el esquema básico de conexiones de datos.

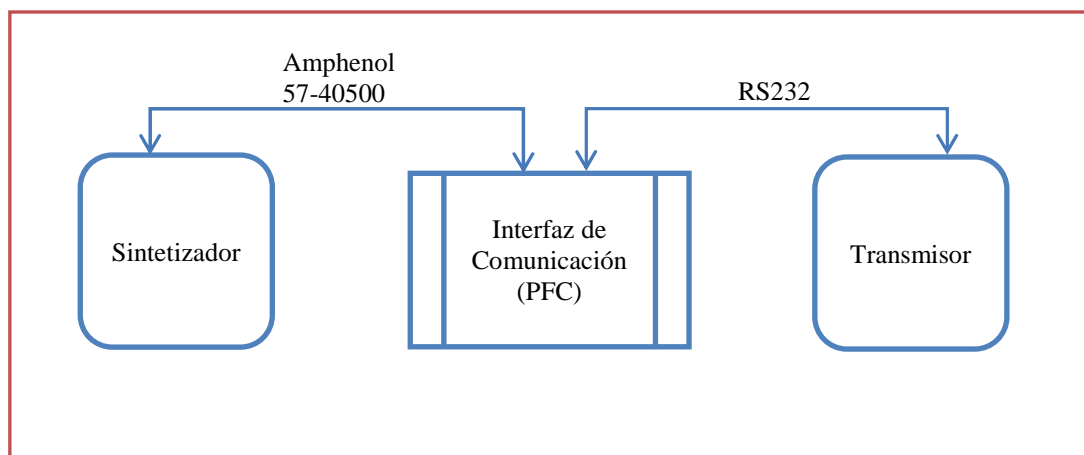


Figura: 5.2.2-1. Esquema básico de conexiones de datos.

Los elementos implicados en dicha comunicación son:

- **Transmisor**, se trata del equipo maestro, envía las órdenes y espera respuesta de confirmación.
- **Interfaz de Comunicación**, se encarga de traducir las instrucciones dadas por el Transmisor y de entregar los mensajes de asentimiento cuando la instrucción ha sido ejecutada.
- **Sintetizador**, se trata del equipo pasivo o esclavo que recibe instrucciones y las ejecuta. Su comunicación es únicamente en un único sentido, entrada de datos.

5.2.3 PROTOCOLO DE COMUNICACIÓN

1. Inicialización o encendido de los sistemas

Se trata de la etapa donde los sistemas se encienden para empezar a estar operativos, vamos a verlo en los tres equipos.

- **Transmisor:** inicializa los procesos de comunicación de sus sistemas, en este primer momento no tenemos intercambio de mensajes por el puerto RS232.

Terminado este proceso queda a la espera de que un operador introduzca una frecuencia en el sistema.

- **Interfaz de Comunicación:** inicialización de los puertos de comunicaciones RS232 en modo I/O, y del puerto de datos TTL en modo Salida.

No existe intercambio de mensajes ni por el puerto RS232 ni por el puerto Amphenol 57-40500.

Una vez inicializado los puertos se inicia también el Watchdog, que estará presente en todo momento para asegurar un buen funcionamiento del sistema y evitar cuelgues debido a fallos inducidos por el entorno de trabajo con alto nivel de radiación electromagnética.

Terminado este proceso queda en modo escucha por el puerto RS232 esperando instrucciones del transmisor.

- **Sintetizador:** Este equipo está desde el inicio en una fase de escucha continuada, preparado para la recepción de órdenes.

2. Modo operación

En la etapa anterior los sistemas quedaron inicializados y a la espera de recibir datos para procesarlos.

- **Transmisor:** Un operador introduce un nuevo canal o frecuencia de emisión, en el panel de operación o a través de un equipo en remoto. En ese momento el transmisor envía a través del puerto de comunicaciones RS232 a nuestro Interfaz de Comunicación el código con la frecuencia asignada. En el formato F<Frecuencia> (EOI).

Quedando a la espera de recibir por parte de nuestro Interfaz de Comunicación la confirmación de mensaje recibido y procesado correctamente.

Una vez recibido el asentimiento por parte del Interfaz de Comunicación, por el módulo RS232, la emisora estará lista para nuevas instrucciones del operador.

- **Interfaz de Comunicación:** en la etapa anterior permanece esperando una interrupción en el puerto de comunicaciones RS232. Recibe la frecuencia del transmisor y despierta con una interrupción, ejecutando la rutina donde procesara dicho código, si la información es correcta, derivará en enviar al Sintetizador el código con la frecuencia y el mensaje de asentimiento al transmisor, que lo dejamos a la espera de recibir la confirmación de que la orden ha sido interpretada y ejecutada correctamente.

Quedando nuevamente a la escucha de nuevas instrucciones por el módulo de comunicación RS232.

- **Sintetizador:** Este equipo está en fase de escucha por su conector Amphenol 57-40500, y una vez recibe el código de nueva frecuencia, aplica la nueva configuración según la información de sus pines y entrega por su salida analógica la frecuencia al transmisor.

3.2.4 DIAGRAMAS DE FLUJO DE CONTROL

El siguiente paso en el desarrollo del software mostraremos los diagramas de control de flujo de los algoritmos principales del sistema.

Se ha decidido hacer varios diagramas, del más general, para tener una visión completa, a los más particulares, de los cuales podremos obtener las operaciones concretas del PIC, según las variables que se presentan.

1. Diagrama de flujo del algoritmo principal

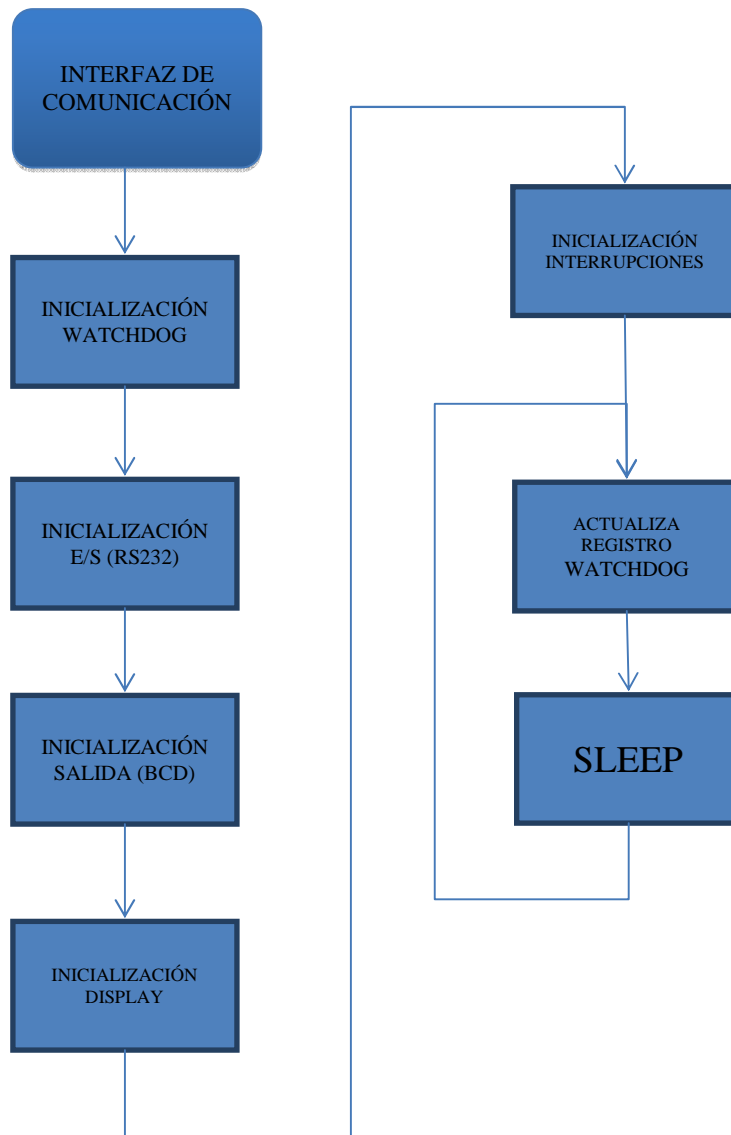


Figura 5.2.4-1. Diagrama del algoritmo principal.

En la figura 5.2.4-1. Podemos ver el diagrama del algoritmo principal del programa.

En él se puede observar una parte importante de inicializaciones, vamos a proceder a explicar cada una de ellas.

1. Inicializar el Watchdog, que va a estar presente en todas las fases de ejecución de nuestro programa, de esta manera tendremos un sistema capaz de reiniciarse en caso de cuelgue del sistema.

2. Inicializar los puertos de comunicaciones TX y RX, así como los valores de dicha comunicación, con este proceso tendremos el puerto de comunicaciones RS232 operativo, pero aun no estará a la escucha de ningún mensaje, para eso tendremos que esperar a activar las interrupciones que son las que nos van a despertar del proceso para avisarnos que tenemos un mensaje de entrada por el puerto de comunicaciones RS232.

3. Inicializar el puerto de salida que comunica con el módulo Interfaz BCD, de esta manera lo dejaremos preparado para transmitir las órdenes al sintetizador cuando sea necesario.

4. Inicializar el Display, dejándolo a un valor de 00.000, con esto y con los LED de las placas de cada módulo, el personal técnico validara que el sistema esta listo para recibir órdenes del transmisor.

5. Activar las interrupciones del PIC, que van a ser las encargadas de despertar a nuestro micro cuando se produzca la llegada de un mensaje por parte del transmisor, esta interrupción hace saltar el contador del programa a una subrutina de interrupción donde vamos a procesarla, una vez terminada la subrutina de interrupción el contador del programa retorna a este al punto donde inicialmente estaba al producirse la interrupción.

Al hablar de interrupciones indicar que solo vamos a activar aquella que necesitamos, en este caso interrupción recepción serie USART. Esta Interrupción es la que maneja los pines RX del PIC, recepción de información a través del puerto de comunicaciones.

6. Actualizar el valor del Watchdog, para evitar que el sistema se reinicie.

7. Sleep, dormimos al PIC a la espera de que una interrupción nos despierte.

2. Diagrama de flujo del tratamiento de una interrupción



Figura 5.2.4-2. Diagrama de flujo del tratamiento de una interrupción.

En la figura 5.2.4-2, podemos ver el diagrama de flujo del tratamiento de una interrupción, esta dividido en las siguientes etapas:

1. Desactivar las interrupciones, de esta manera cancelamos la opción de ser avisados en caso de la llegada de otra interrupción. Con lo que evitaremos entrar en otra interrupción sin haber terminado de atender la que tenemos pendiente.

2. Llamar al procedimiento Interfaz, este será el encargado de leer el comando recibido por el puerto RX y ejecutarlo.

3. Activar las interrupciones, una vez termine el procedimiento Interfaz, la interrupción a sido atendida y antes de volver al programa principal activaremos nuevamente las interrupciones.

3. Diagrama de flujo del procedimiento Interfaz

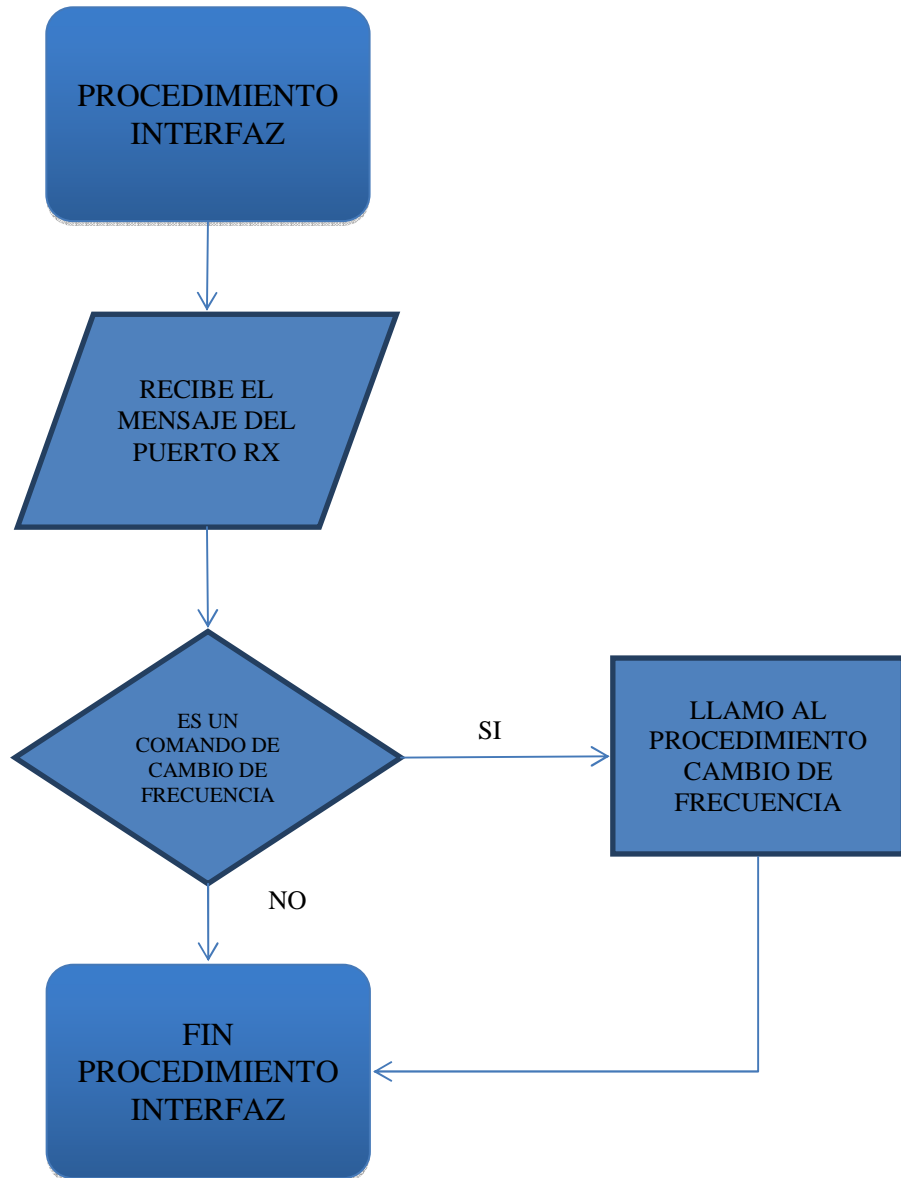


Figura 5.2.4-3. Diagrama de flujo del procedimiento Interfaz.

El procedimiento Interfaz, figura 5.2.4-3, lo primero que hace es recibir el mensaje por el puerto RX, y va a analizar si es un comando conocido, si es así llamara al procedimiento cambio de frecuencia, en caso contrario terminara el procedimiento.

4. Diagrama de bloques del procedimiento Cambio de Frecuencia

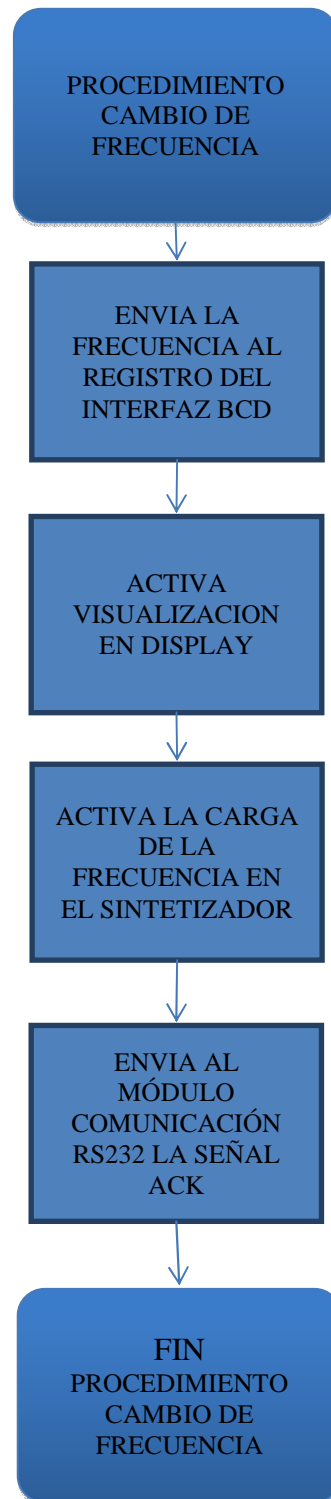


Figura 5.2.4-4. Diagrama de bloques del procedimiento Cambio de Frecuencia.

El procedimiento Cambio de Frecuencia, figura 5.2.4-4, va a realizar los siguientes pasos: Primero se envía al registro del módulo BCD la frecuencia

recibida, esta vez transformada al formato BCD, allí guardaremos los valores de los 5 dígitos, en los registros, una vez tenemos los datos guardados en los registros, procedemos a activar la carga de esos datos en el display y a su vez a enviar la señal de carga de datos, nueva frecuencia, en el sintetizador. Terminado esta etapa enviamos al módulo RS232 a través del puerto TX, la confirmación al transmisor de que la operación requerida esta correctamente terminada y ya esta lista la frecuencia en el sintetizador.

5.2.5 IMPLEMENTACIÓN

La implementación en C del código del PIC, se ha adjuntado en el CD que acompaña el PFC.

5.2.6 CARGA DEL PROGRAMA

Para la carga del programa dentro del PIC, lo mejor es utilizar un programador de dispositivos, de esta manera aislaremos el chip y evitaremos problemas a la hora de grabar el contenido de nuestro código, ver detalle en la figura 5.2.6-1.

Para ello es necesario explicar el funcionamiento de dos patillas del PIC, que se utilizan para tal propósito, nos referimos a los pines RB6 y RB7.

El PIC16F886 no dispone de ningún pin especial para la programación. Esta función se les asigna a los pines de E/S. Más concretamente son los pines del puerto PORTB utilizados para la transmisión de señal de reloj (RB6) y de datos (RB7) durante la programación. Además, es necesario suministrar el voltaje de alimentación Vdd (5 V) así como el voltaje apropiado Vpp (12 - 14 V) para la programación de memoria FLASH. El pin MCLR se utiliza para este propósito.

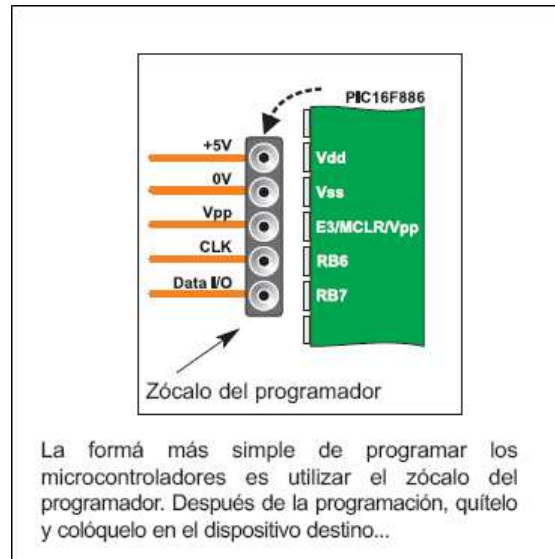


Figura 5.2.6-1. Programador de microcontroladores.

Como hemos mencionado, para habilitar la carga de un código hex en el microcontrolador es necesario tener un dispositivo especial, denominado el programador, con software apropiado, en nuestro caso hemos utilizado el programador que viene con el entrenador MPLAB.

Una vez programado, sólo queda instalar el microcontrolador programado en el dispositivo Interfaz de Comunicación.

Si es necesario hacer algunos cambios en el programa, el procedimiento anterior se puede repetir un número ilimitado de veces.

5.3 MODO USUARIO

El Interfaz de Comunicación diseñado no requiere de ningún tipo de supervisión por parte del usuario final o técnico, una vez realizadas las siguientes conexiones, estará listo para funcionar.

1. Conectar el Interfaz de Comunicación con el sintetizador utilizando un cable con conectores Amphenol 57-40500 macho/hembra en sus extremos. Ver figura 5.3-1.

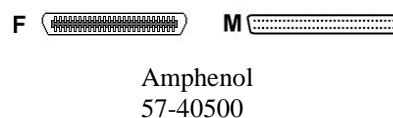


Figura 5.3-1. Detalle del conector Amphenol.

2. Conectar el Interfaz de Comunicación con el transmisor utilizando un cable con conectores RS232 macho/hembra en sus extremos. Ver figura 5.3-2.



Figura 5.3-2. Detalle del conector RS232.

3. Conectar la alimentación, entre 7 V y 12 V en alterna, procedente de un transformador externo.

A partir de ese momento el sistema ya estará listo para funcionar y realizar las operaciones de Interfaz de Comunicación.

5.3.1 DETECCIÓN DE ANOMALÍAS

El operador puede ver el correcto funcionamiento, una vez haya realizado la operación de cambio de frecuencia en el transmisor, se reflejara en el display visualizador el valor de la frecuencia actual. Inicialmente ha de presentar un valor de 00.000 Khz. en el display.

Se han dispuesto en todos los módulos unos diodos LED para detectar anomalías en la alimentación del sistema. Una vez conectado el sistema, todos los LED tendrán que estar luciendo, indicando un correcto funcionamiento.

En el módulo Conector Amphenol, tenemos dos diodos LED, a diferencia que en los demás módulos que solo hay un diodo LED de indicación de alimentación correcta.

Esos dos LED muestran la siguiente información.

D1: LED que señala una buena conexión con el sintetizador.

D2: LED de alimentación correcta.

En el caso de no lucir D1, nos indica un fallo de comunicación con el sintetizador. Este puede ser causa de un fallo por una mala conexión entre el sintetizador y el conector Amphenol.

Cuando el sintetizador no esta encendido el diodo D1 no lucirá. (Fallo de comunicación con el sintetizador)

5.3.2 MODO TEST DISPLAY FRECUENCÍMETRO

El Interfaz de Comunicación dispone en el módulo Display Frecuencímetro, un jumper con dos posiciones para realizar un test de funcionamiento correcto.

Una posición del jumper es la de modo normal de funcionamiento y la otra cambia a modo test el módulo. Con este modo el usuario puede realizar un test donde todos los LED del display de 7 segmentos quedan activos para observar si existe alguno defectuoso.

5.3.3. RESET DEL PIC MANUAL

El operador puede realizar un Reset del sistema solo con pulsar el botón Reset del módulo PIC, con ello el microcontrolador iniciará la rutina de reinicio.

6. CONCLUSIONES

Se ha diseñado un sistema autónomo, capaz de hacer de interfaz entre los dos equipos para los que ha sido proyectado.

Para llevarlo acabo hemos tenido que realizar tanto el diseño hardware como software del sistema, dando con ello solución al problema inicial.

Por un lado tenemos la opción de conectar al transmisor cualquiera de los dos modelos de sintetizadores de los que se dispone, el original del fabricante y el sintetizador Continental Electronics PTP040, para el que hemos desarrollado el proyecto.

Siempre se podrá conectar de manera directa y sin el uso de nuestro interfaz el sintetizador original del fabricante. Y con el empleo de nuestro Interfaz de Comunicación se podrá conectar el sintetizador para el cual hemos creado nuestro proyecto.

Los posibles trabajos futuros serán la adecuación del interfaz a los posibles y nuevos modelos de sintetizadores que se incorporen en el mercado y así de esta manera poder hacer compatible cualquier modelo de sintetizador con el transmisor.

Este proyecto se ha realizado para compatibilizar el transmisor Thomcast a cualquiera de los dos sintetizadores existentes, pero de la misma manera y como opciones de desarrollo futuras se podría diseñar un nuevo interfaz para compatibilizar los transmisores Continental Electronics para el uso de sintetizadores Thomcast u otras combinaciones.

Para ello tomaríamos todas las notas y documentación contenida en dicho proyecto que nos servirán de base para el desarrollo de la nueva Interfaz de Comunicación.

7. BIBLIOGRAFÍA

Aplicaciones de los microcontroladores PIC de Microchip.

J. Ma. Angulo Usategui, E. Martín Cuenca y J. Angulo Martínez.

Editorial McGraw Hill, 1998.

Arquitectura de Equipos y Sistemas Informáticos.

Carlos Valdivia Miranda.

Editorial Paraninfo.

Arquitectura de Equipos y Sistemas Informáticos.

Carlos Valdivia Miranda.

Editorial Paraninfo.

Microchip PIC Microcontrollers.

Data Book, Microchip Technology Inc.

Microcontroladores PIC: prácticas de programación.

Lehmann, Stefan.

Editorial Marcombo, 2008.

Microcontroladores PIC, la solución en un chip.

J. Ma. Angulo Usategui, E. Martín Cuenca y J. Angulo Martínez.

Editorial Paraninfo, 1997.

Microcontroladores PIC. Diseño práctico de aplicaciones.

José María Angulo Usategui e Ignacio Angulo Martínez.

Editorial Mc GrawHill.

Microcontroladores PIC. Diseño práctico de aplicaciones. José María Angulo Usategui e Ignacio Angulo Martínez.

Editorial Mc GrawHill.

Programación de microcontroladores PIC: desarrollo de 30 proyectos con PICBASIC Y PICBASIC Profesional.

Dogan, Ibrahim.

Editorial Marcombo, D.L. 2007.

8. GLOSARIO DE TÉRMINOS

AC - Se denomina corriente alterna (abreviada CA en español y AC en inglés, de alternating current) a la corriente eléctrica en la que la magnitud y el sentido varían cíclicamente. La forma de oscilación de la corriente alterna más comúnmente utilizada es la de una oscilación sinusoidal, puesto que se consigue una transmisión más eficiente de la energía.

AM - Amplitud modulada o modulación de amplitud es un tipo de modulación lineal que consiste en hacer variar la amplitud de la señal portadora de forma que esta cambie de acuerdo con las variaciones de nivel de la señal que contiene la información que se desea transmitir, llamada señal moduladora o modulante.

Archivo hex - Representación ASCII del código de máquina. Un archivo .hex está compuesto de registros que le especifican al microcontrolador datos o instrucciones que serán ubicados en un dispositivo de memoria programable.

BCD - (Binary-Coded Decimal) o Decimal codificado en binario es un estándar para representar números decimales en el sistema binario, en donde cada dígito decimal es codificado con una secuencia de 4 bits.

CMOS o Complementary metal-oxide-semiconductor - Es una de las familias lógicas empleadas en la fabricación de circuitos integrados. Su principal característica consiste en la utilización conjunta de transistores de tipo pMOS y tipo nMOS configurados de tal forma que, en estado de reposo, el consumo de energía es únicamente el debido a las corrientes parásitas.

Código de Objeto - Código intermedio producido por un lenguaje ensamblador o compilador. Este código no contiene direcciones para la memoria del programa o las variables del RAM incluidas en el mismo, pero contiene demarcadores para las direcciones que el linker debe determinar cuando coloca el código en la memoria del programa.

Código fuente - Archivo de texto que es procesado por un lenguaje ensamblador o un compilador para producir un archivo de objeto intermedio, o código de máquina que pueda ejecutarse en un microcontrolador.

Compilador - Programa que convierte las proposiciones de un lenguaje de alto nivel en un código de máquina que puede ejecutarse en un microcontrolador. Una proposición de un lenguaje de alto nivel usualmente genera una cantidad de instrucciones de código de máquina.

Conector BNC - (Del inglés Bayonet Neill-Concelman) es un tipo de conector para uso con cable coaxial. Inicialmente diseñado como una versión en miniatura del Conector Tipo C. BNC es un tipo de conector usado con cables coaxiales como RG-58 y RG-59 en aplicaciones de RF que precisaban de un conector rápido, apto para UHF y de impedancia constante a lo largo de un amplio espectro.

DC - La corriente continua o corriente directa (CC en español, en inglés DC, de Direct Current) es el flujo continuo de electrones a través de un conductor entre dos puntos de distinto potencial. A diferencia de la corriente alterna (CA en español, AC en inglés), en la corriente continua las cargas eléctricas circulan siempre en la misma dirección (es decir, los terminales de mayor y de menor potencial son siempre los mismos). Aunque comúnmente se identifica la corriente continua con la corriente constante (por ejemplo la suministrada por una batería), es continua toda corriente que mantenga siempre la misma polaridad. También cuando los electrones se mueven siempre en el mismo sentido, el flujo se denomina corriente continua y va del polo positivo al negativo.

E/S o I/O, entrada/salida o input/output - Es la colección de interfaces que usan las distintas unidades funcionales (subsistemas) de un sistema de procesamiento de información para comunicarse unas con otras, o las señales (información) enviadas a través de esas interfaces. Las entradas son las señales recibidas por la unidad, mientras que las salidas son las señales enviadas por ésta.

EEPROM - Son las siglas de Electrically Erasable Programmable Read-Only Memory (ROM programable y borrable eléctricamente). Es un tipo de memoria ROM que puede ser programada, borrada y reprogramada eléctricamente. Son memorias no volátiles.

Emulador - Instrumento de hardware usado en lugar del microcontrolador en una aplicación. El emulador permite que el código sea rápidamente descargado, ejecutado y probado, tal como será ejecutado en la aplicación final.

Fuente de Alimentación - Es un dispositivo que convierte la tensión alterna de la red de suministro, en una o varias tensiones, prácticamente continuas, que alimentan los distintos circuitos del aparato electrónico al que se conecta.

Hardware - Corresponde a todas las partes tangibles de un sistema informático. Sus componentes son: eléctricos, electrónicos, electromecánicos y mecánicos.

Interfaz - Conexión física y funcional entre dos aparatos o sistemas independientes.

Interrupción - (También conocida como interrupción de hardware o petición de interrupción) Es una señal recibida por el procesador de un ordenador, indicando que debe "interrumpir" el curso de ejecución actual y pasar a ejecutar código específico para tratar esta situación.

Lenguaje ensamblador - Programa que convierte instrucciones mnemotécnicas en un código de máquina que pueda ejecutarse en un microcontrolador. Hay una correspondencia uno-a-uno entre la mnemotécnica y el código de máquina.

Linker - Programa que convierte el código de un objeto en un código de máquina ejecutable. El linker determina dónde se ubicará el código en la memoria y cómo será usada la RAM por las variables.

Memoria de Programa - La memoria de un microcontrolador que contiene una serie de instrucciones para ejecutar una aplicación.

Memoria FLASH - Es una tecnología de almacenamiento (derivada de la memoria EEPROM) que permite la lecto-escritura de múltiples posiciones de memoria en la misma operación. Permite velocidades de funcionamiento muy superiores frente a la tecnología EEPROM primigenia, que sólo permitía actuar sobre una única celda de memoria en cada operación de programación.

Microcontrolador - (Abreviado μ C, UC o MCU) es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica. Un microcontrolador incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida.

Modulación - Engloba el conjunto de técnicas que se usan para transportar información sobre una onda portadora, típicamente una onda sinusoidal. Estas técnicas permiten un mejor aprovechamiento del canal de comunicación lo que posibilita transmitir más información en forma simultánea además de mejorar la resistencia contra posibles ruidos e interferencias. Básicamente, la modulación consiste en hacer que un parámetro de la onda portadora cambie de valor de acuerdo con las variaciones de la señal moduladora, que es la información que queremos transmitir.

OC u Onda Corta - La que tiene una longitud comprendida entre 10 y 50 m. La Onda Corta, también conocida como SW (del inglés shortwave) o HF (high frequency) es una banda de radiofrecuencias comprendidas entre los 2300 y los 29.999 khz en la que transmiten (entre otras) las emisoras de radio internacionales para transmitir su programación al mundo y las estaciones de radioaficionados. En estas frecuencias las ondas electromagnéticas, que se propagan en línea recta, rebotan a distintas alturas (cuanta más alta la frecuencia a mayor altura) de la ionosfera (con variaciones según la estación del año y la hora del día), lo que permite que las señales alcancen puntos lejanos e incluso den la vuelta al planeta.

Onda Portadora - Es una forma de onda, generalmente sinusoidal, que es modulada por una señal que se quiere transmitir.

Oscilador Pierce - Es un oscilador en el cual el circuito resonante LC es remplazado por un cristal de cuarzo que tiene un factor Q (factor de mérito o factor de calidad) sumamente elevado. Con él se pueden obtener frecuencias estables del orden de las partes por millón. Sus principales ventajas radican en su capacidad de trabajo en altas frecuencias y en mantener estable su frecuencia de trabajo.

Programador de dispositivo - Instrumento de hardware que toma el código de máquina desde un archivo y lo programa en un microcontrolador programable o un chip de memoria programable.

Protocolo de Comunicaciones - Es el conjunto de reglas normalizadas para la representación, señalización, autenticación y detección de errores necesario para enviar información a través de un canal de comunicación.

Proyecto - Grupo de archivos usados para crear una aplicación, junto con instrucciones a lenguajes ensambladores, compiladores y linkers para convertir los archivos en un código de máquina ejecutable. Los archivos pueden ser archivos fuente de ensamble,

archivos fuente del compilador, bibliotecas, archivos de objeto pre-compilados, y archivos de instrucciones llamados escrituras del linker.

Puerto Paralelo - Es una interfaz entre una computadora y un periférico, cuya principal característica es que los bits de datos viajan juntos, enviando un paquete de byte a la vez. Es decir, se implementa un cable o una vía física para cada bit de datos formando un bus. En un puerto paralelo habrá una serie de bits de control en vías aparte que irán en ambos sentidos por caminos distintos.

Puerto Serie o puerto serial - Es una interfaz de comunicaciones de datos digitales, frecuentemente utilizado por computadoras y periféricos, donde la información es transmitida bit a bit enviando un solo bit a la vez.

Registros de archivo - El área RAM interna de micro PIC usada para almacenaje de variables.

Registros de Función Especial - Ubicaciones internas de un microPIC que pueden ser usadas para afectar la operación interna del controlador o un periférico. Entre los ejemplos se pueden incluir los registros de control de interrupción, los timers y los registros I/O.

Resistencia pull-up o resistencia de polarización - es aquella resistencia que posee uno de sus terminales al positivo de la fuente de alimentación. En electrónica digital es utilizada para combinar tecnología TTL y CMOS.

RF o radiofrecuencia - Cada una de las frecuencias de las ondas electromagnéticas empleadas en la radiocomunicación. Las ondas electromagnéticas de esta región del espectro, se pueden transmitir aplicando la corriente alterna originada en un generador a una antena.

RNE - Radio Nacional de España - Es una sociedad mercantil estatal, financiada en su totalidad por los Presupuestos Generales del Estado, que se encarga de la gestión directa del servicio público de radiodifusión en España. Desde 1973 está integrada en el Ente Público Radio Televisión Española (RTVE). Desde 2007 pertenece a la Corporación RTVE, regulada en la Ley 17/2006, de 5 de junio, de la radio y la televisión de titularidad estatal.

Señal Analógica - Es un tipo de señal generada por algún tipo de fenómeno electromagnético y que es representable por una función matemática continua en la que es variable su amplitud y periodo (representando un dato de información) en función del tiempo. Algunas magnitudes físicas comúnmente portadoras de una señal de este tipo son eléctricas como la intensidad, la tensión y la potencia.

Señal Digital - Es un tipo de señal generada por algún tipo de fenómeno electromagnético en que cada signo que codifica el contenido de la misma puede ser analizado en término de algunas magnitudes que representan valores discretos, en lugar de valores dentro de un cierto rango.

Sistema Digital - Es un conjunto de dispositivos destinados a la generación, transmisión, procesamiento o almacenamiento de señales digitales. También un sistema

digital es una combinación de dispositivos diseñados para manipular cantidades físicas o información que estén representadas en forma digital, es decir, que sólo puedan tomar valores discretos.

Tabla de Verdad o tabla de valores de verdad - Es una tabla que muestra el valor de verdad de una proposición compuesta, para cada combinación de valores de verdad que se pueda asignar a sus componentes.

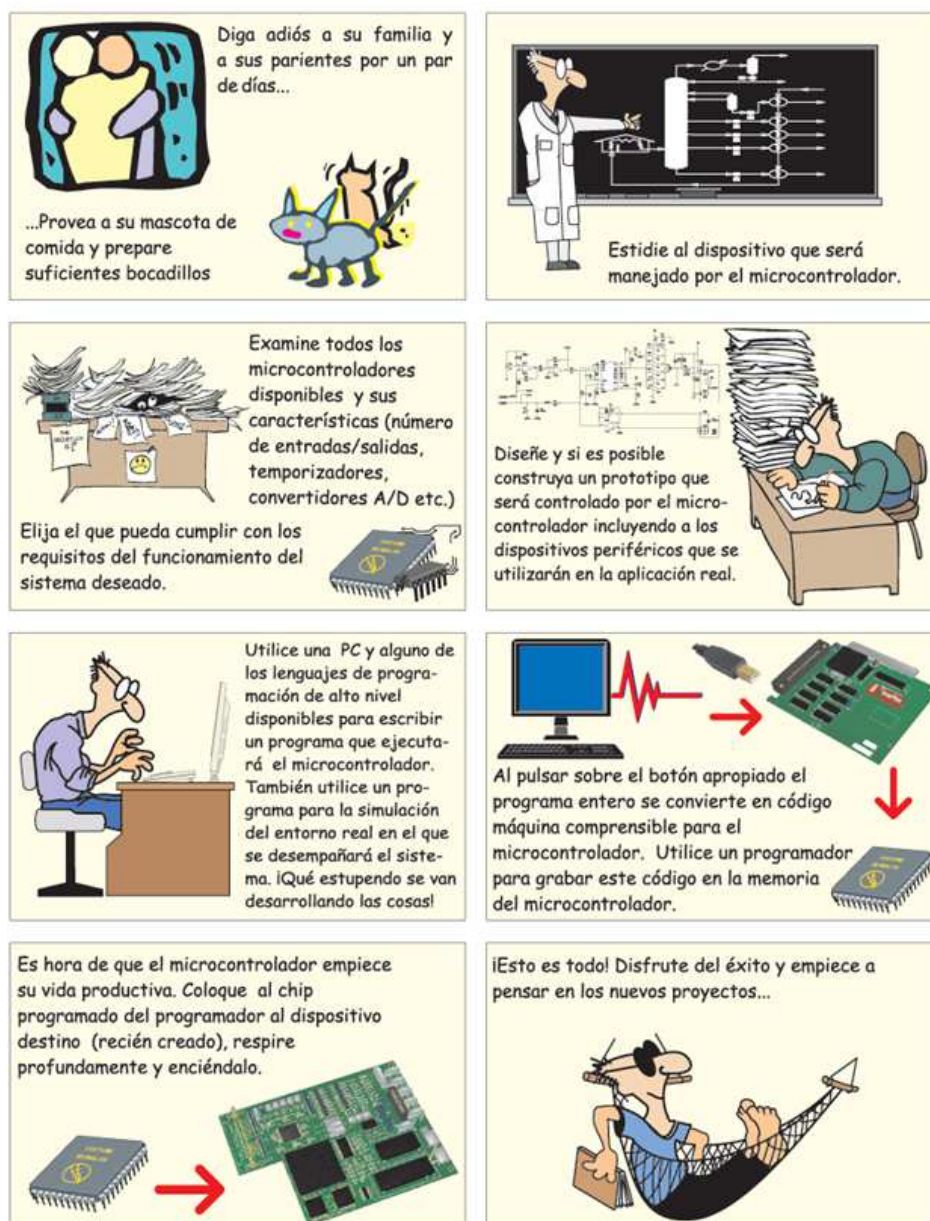
Velocidad de Transmisión de Datos – La medida de tiempo desde el instante en que se pone el primer bit en la línea hasta el último bit del paquete a transmitir. La unidad de medida en el Sistema Internacional (de estar contemplado en el mismo) sería en bits/segundo (b/s o también bps), o expresado en octetos o bytes (B/s).

9. ANEXOS

9.1 DISEÑO DE LOS PCB'S

El documento con la implementación del diseño de los PCB's y el listado de componentes, se ha adjuntado en el CD que acompaña el PFC.

9.2 ALGORITMO DE REALIZACIÓN DE PROYECTOS CON MICROCONTROLADORES PARA INGENIEROS INFORMÁTICOS



FUENTE: <http://www.mikroe.com>

Figura 9.2-1. Algoritmo de realización de proyectos con microcontroladores para Ingenieros Informáticos.