

Transmisión de datos por la red de distribución eléctrica con módem de baja velocidad

Marcos López Cabeceira Roberto Zangróniz Cantabrana

diciembre de 2001

Índice General

1	Transmisión por la red eléctrica	21
1.1	Introducción	21
1.2	Redes de datos sobre la línea eléctrica	23
1.2.1	Características de la línea eléctrica	23
1.3	Clasificación de los módem de línea	24
1.3.1	Módem de baja velocidad	25
1.3.2	Utilidades de los módem de baja velocidad	30
1.4	Módem de alta velocidad	31
1.4.1	Utilidades de los módems de alta velocidad	35
2	Microcontroladores	37
2.1	Introducción	37
2.2	Definición de microcontrolador	37
2.3	Aplicaciones de los microcontroladores	39
2.4	Mercado de los microcontroladores	40
2.5	Tipos de memoria	42
2.5.1	EPROM	43
2.5.2	OTP (One Time Programmable)	43
2.5.3	E ² PROM	43

2.5.4	ROM	44
2.5.5	Flash EPROM	44
2.5.6	RAM	44
2.6	Periféricos internos	44
2.6.1	Convertidor A/D	45
2.6.2	Temporizador/contador	45
2.6.3	Temporizador con auto recarga	45
2.6.4	Temporizador watchdog	45
2.6.5	Puertos de E/S	46
2.6.6	Unidad de comparación y captura	46
2.6.7	Interface serie	46
2.7	Los microcontroladores más populares	47
2.8	Herramientas de desarrollo	50
2.8.1	Herramientas de software	50
2.8.2	Herramientas de hardware	51
3	La fuente de alimentación	53
3.1	Introducción	53
3.2	Diseño sin transformador	54
3.3	Problemas en el diseño	55
3.3.1	Potencia	55
3.3.2	Aislamiento	56
4	El emisor de señal	59
4.1	Introducción	59
4.2	El circuito integrado LM555	60
4.3	El diseño final	62

<i>ÍNDICE GENERAL</i>	5
5 El detector de señal	65
5.1 Introducción	65
5.2 El circuito integrado NE567	65
5.3 El diseño final	67
6 El controlador del sistema	71
6.1 Introducción	71
6.2 El microcontrolador PIC 16F84A	72
6.3 El diseño en la placa central	74
6.4 El diseño en la placa sensora	75
6.5 El diseño en la placa actuadora	76
7 La interfaz con el PC	79
7.1 Introducción	79
7.2 La comunicación	79
7.3 Aislamiento eléctrico: el optoacoplador CI 4N35	80
7.4 Lógica RS232 versus lógica TTL	81
7.5 El Diseño final	84
8 El sensor de temperatura	85
8.1 Introducción	85
8.2 El sensor LM35DZ	86
8.3 El CAD Sigma Delta	86
8.3.1 Cuantización	86
8.3.2 Conversión A/D paralelo (flash)	89
8.3.3 Conversión por ponderación	89
8.3.4 Conversión por conteo	90

8.3.5	Conversión de un bit	90
8.4	El diseño final	92
9	El regulador de luz	97
9.1	Introducción	97
9.2	Tiristores	98
9.3	Diac	99
9.4	SCR	99
9.4.1	Silicon Controlled Rectifier	99
9.4.2	Característica V/I	100
9.5	Triac	106
9.5.1	Circuito equivalente de un triac	107
9.6	Conexión con la red eléctrica	108
9.6.1	Introducción	108
9.6.2	Aplicaciones	109
9.6.3	Teoría de operación	109
9.7	El diseño final	111
10	La capa física	115
10.1	Introducción	115
10.2	Codificación	116
10.3	Todos escuchan a todos	119
10.4	Sincronización	121
10.5	Tramas de control	121
10.6	Corrección de errores	122
10.7	Distribución de la información	122
10.8	Un ejemplo de una trama	123

<i>ÍNDICE GENERAL</i>	7
11 Código de los μC	125
11.1 El PIC 16F84A de la placa central	125
11.2 El PIC 16F84A de la placa sensora	126
11.3 El PIC 16F84A de la placa actuadora	127
12 Dome	133
12.1 Introducción	133
12.2 Interconexión RS232	134
12.3 Comunicaciones por el puerto serie	136
12.4 Interfaz de comunicaciones	138
12.5 Interfaz del usuario	139
12.6 Enviar y recibir datos	141
13 Aspectos económicos	143
13.1 Introducción	143
13.2 La placa central	143
13.3 La placa sensora	143
13.4 La placa actuadora	144
13.5 Conclusiones	145
A Perspectivas futuras	147
A.1 Mejoras	147
A.2 Problemas	149
B Dome: Manual de usuario	151
B.1 Primeros pasos	151
B.2 Conexión con un puerto	153
B.3 Comunicación con la placa central	156

C	Esquemas	159
C.1	La placa central	159
C.2	La placa sensora	159
C.3	La placa actuadora	159
D	Prototipos	175
D.1	La placa central	175
D.2	La placa sensora	175
D.3	La placa actuadora	175
D.4	El programador del PIC 16F84A	175
D.5	Una visión general	175

Índice de Figuras

1.1	<i>Aplicaciones de una red de datos de baja velocidad.</i>	31
1.2	<i>Aplicaciones de una red de datos de alta velocidad.</i>	36
2.1	<i>Diagrama de bloques típico de un μC.</i>	38
2.2	<i>Mercado mundial de microcontroladores: a) en número de unidades, b) por volumen de negocio.</i>	41
3.1	<i>Esquema de la fuente de alimentación.</i>	55
3.2	<i>Transformadores de aislamiento de la red eléctrica.</i>	57
4.1	<i>Esquema de pines del CI LM555.</i>	60
4.2	<i>Esquema interno de un CI LM555.</i>	61
4.3	<i>Esquema básico de CI LM555 como multivibrador astable.</i>	62
4.4	<i>Esquema del emisor de señal.</i>	64
5.1	<i>Esquema de pines del CI NE567.</i>	66
5.2	<i>Esquema básico de un CI LM567.</i>	67
5.3	<i>Amplitud de la señal de entrada versus ancho de banda.</i>	68
5.4	<i>Esquema del detector de tono.</i>	70
6.1	<i>Diagrama de pines del PIC 16F84A.</i>	73

6.2	<i>Módulo controlador de la placa central.</i>	75
6.3	<i>Módulo controlador de la placa sensora.</i>	77
6.4	<i>Módulo controlador de la placa actuadora.</i>	78
7.1	<i>Esquema de pines del CI 4N35.</i>	81
7.2	<i>Esquema interno del optoacoplador CI 4N35.</i>	82
7.3	<i>Esquema de la interfaz con el PC.</i>	83
8.1	<i>Sensor de temperatura centígrado básico</i>	87
8.2	<i>Sensor de temperatura centígrado de amplio rango.</i>	88
8.3	<i>Señal analógica.</i>	89
8.4	<i>Muestreador más cuantificador.</i>	90
8.5	<i>Señal analógica muestreada.</i>	91
8.6	<i>Señal analógica cuantificada.</i>	92
8.7	<i>Señal digital.</i>	93
8.8	<i>Modulador de primer orden.</i>	94
8.9	<i>Esquema del sensor de temperatura.</i>	95
9.1	<i>Símbolo del tiristor.</i>	99
9.2	<i>Símbolo del diac.</i>	100
9.3	<i>Símbolo del SCR.</i>	101
9.4	<i>Gráfica de la característica V/I de un SCR.</i>	101
9.5	<i>Interrupción de corriente anódica.</i>	102
9.6	<i>Conmutación forzada.</i>	103
9.7	<i>Circuito de control de fase de media onda y resistencia variable disparado a 0°.</i>	104
9.8	<i>Circuito de control de fase de media onda y resistencia variable disparado a 180°.</i>	105

ÍNDICE DE FIGURAS	11
9.9 <i>Disparo en cada ciclo.</i>	105
9.10 <i>Símbolo del triac.</i>	106
9.11 <i>Modelo de triac de dos SCR.</i>	107
9.12 <i>Gráfica de la característica V/I de un triac.</i>	108
9.13 <i>Estructura de un pin de entrada/salida.</i>	110
9.14 <i>Conexión de la red eléctrica con un PIC.</i>	111
9.15 <i>Señal en el pin de entrada.</i>	112
9.16 <i>Valores típicos de la conexión con la red eléctrica.</i>	113
9.17 <i>Esquema del regulador de luz.</i>	114
10.1 <i>Los cuatro tipos de bits.</i>	118
10.2 <i>Un ejemplo de trama.</i>	118
10.3 <i>Grabación de la dirección en la E^2PROM.</i>	120
10.4 <i>La trama de sincronismo.</i>	122
10.5 <i>Un ejemplo de una trama completa.</i>	124
11.1 <i>Diagrama de flujo del microcontrolador de la placa central.</i>	129
11.2 <i>Diagrama de flujo del microcontrolador de la placa sensora.</i>	130
11.3 <i>Diagrama de flujo del microcontrolador de la placa actuadora.</i>	131
12.1 <i>Conexión de las diferentes señales en el protocolo RS232.</i>	134
12.2 <i>Diagrama de flujo del programa Dome.</i>	142
B.1 <i>Ventana principal de la aplicación Dome.</i>	152
B.2 <i>Cuadro de diálogo “Acerca de Dome...”.</i>	153

B.3	<i>Modo Release y modo Debug.</i>	153
B.4	<i>Habilitación de los controles en el cuadro de diálogo al abrir el puerto.</i>	154
B.5	<i>Cuadro de diálogo “Configuración”.</i>	155
B.6	<i>Cuadro de diálogo de información.</i>	156
B.7	<i>Petición de temperatura a la placa central.</i>	156
B.8	<i>Envío de una intensidad de luz a la placa central.</i>	157
B.9	<i>Envío de un comando de depuración a la placa central.</i>	157
C.1	<i>Esquema del emisor de señal.</i>	160
C.2	<i>Esquema de la fuente de alimentación.</i>	161
C.3	<i>Esquema del módulo controlador.</i>	162
C.4	<i>Esquema del detector de señal.</i>	163
C.5	<i>Esquema de la interface con el PC.</i>	164
C.6	<i>Esquema del emisor de señal.</i>	165
C.7	<i>Esquema de la fuente de alimentación.</i>	166
C.8	<i>Esquema del módulo controlador.</i>	167
C.9	<i>Esquema del detector de señal.</i>	168
C.10	<i>Esquema del sensor de temperatura.</i>	169
C.11	<i>Esquema del emisor de señal.</i>	170
C.12	<i>Esquema de la fuente de alimentación.</i>	171
C.13	<i>Esquema del módulo controlador.</i>	172
C.14	<i>Esquema del detector de señal.</i>	173
C.15	<i>Esquema del regulador de luz.</i>	174
D.1	<i>Vista superior de la placa central.</i>	176
D.2	<i>Vista inferior de la placa central.</i>	177

D.3	<i>Vista superior de la placa sensora.</i>	178
D.4	<i>Vista inferior de la placa sensora.</i>	179
D.5	<i>Vista superior de la placa actuadora.</i>	180
D.6	<i>Vista inferior de la placa actuadora.</i>	181
D.7	<i>Vista superior de programador del PIC 16F84A.</i>	182
D.8	<i>Una visión general de todo el sistema funcionando.</i>	183

Índice de Tablas

12.1	Función de las diferentes señales en un conector RS232.	135
13.1	<i>Coste de la placa central.</i>	144
13.2	<i>Coste de la placa sensora.</i>	144
13.3	<i>Coste de la placa actuadora.</i>	145
13.4	<i>Coste de la placa actuadora.</i>	145
B.1	<i>Valores de los parámetros de configuración.</i>	155

Prólogo

El proyecto consiste en el aprovechamiento de la red de distribución eléctrica presente en los hogares como canal físico para el intercambio de información. Esta nueva funcionalidad de la red aparece como consecuencia del interés creciente en controlar los diferentes electrodomésticos de una casa de forma automática. En este proyecto hemos abordado el paso inicial del sistema central y dos periféricos de prueba: un sensor de temperatura y un regulador de luz.

Capítulo 1 En este capítulo comentamos las propiedades de la red eléctrica así como una clasificación básica en dos tipos de módem que ayuda a comprender donde encuadrar nuestro proyecto y su objetivo.

Capítulo 2 Aquí realizamos un resumen de las diferentes aplicaciones de los microcontroladores, así como una rápida descripción de las diferentes partes que los componen. También hemos incluido un vistazo al mundo de los microcontroladores en general.

Capítulo 3 Explicamos el diseño de la fuente de alimentación así como los diferentes problemas que nos encontramos en su realización y sus limitaciones.

Capítulo 4 Aquí abordamos la codificación de la señal para transformar las señales digitales del microcontrolador en una señal susceptible de enviarse por la red eléctrica.

Capítulo 5 En este capítulo abordamos el paso contrario al capítulo anterior, ahora decodificamos la señal analógica procedente de la red eléctrica en una señal digital manejable por un microcontrolador.

Capítulo 6 Nos hemos detenido en explicar las funcionalidades del microcontrolador que hemos decidido usar. También hacemos un recorrido por el diseño que hemos tomado para el microcontrolador en cada una de las tres diferentes placas.

Capítulo 7 Explicamos la comunicación entre un ordenador personal PC por medio de su puerto serie y uno de nuestros dispositivos que va a hacer de placa central. Así conseguimos una interfaz para el usuario desde donde controlar el sistema.

Capítulo 8 Aquí explicamos el diseño del sensor de temperatura y como el microcontrolador puede a través de este sensor obtener el valor de la temperatura por medio de un conversor analógico/digital sigma-delta.

Capítulo 9 Exponemos el diseño de un regulador de luz controlado digitalmente desde un microcontrolador, gobernada su intensidad desde el microcontrolador principal del dispositivo actuador.

Capítulo 10 En este capítulo comentamos todo lo relativo a la codificación que realizamos para convertir nuestra señal digital del microcontrolador en una señal analógica y las diferentes partes de las que se compone.

Capítulo 11 Aquí comentamos los programas que están ejecutando los microcontroladores de cada uno de los tres dispositivos diseñados. Hacemos una descripción general de su funcionamiento y del ciclo principal de trabajo que realizan.

Capítulo 12 Aquí explicamos el desarrollo de la aplicación Dome que hace de interfaz con el usuario. En líneas generales explicamos su funcionamiento y las tareas que realiza.

Capítulo 13 Para terminar, nos ocupamos de los aspectos económicos constructivos y de diseño de nuestros prototipos de módem, así como del programador de los PIC y la aplicación Dome.

Capítulo 1

Transmisión por la red eléctrica

1.1 Introducción

En la actualidad, existen varias tecnologías completamente diferenciadas para la transmisión de datos en el hogar. Éstas son: pares trenzados, red eléctrica, hilos telefónicos y radiofrecuencia.

Pares trenzados El cable de pares trenzados no blindados (UTP Cat. 5) es la solución tradicional para redes de datos locales. Se utiliza en el estándar Ethernet, y permite velocidades de transferencia de 10 ó 100*Mbps*. Sin embargo, resulta inapropiado para un hogar por el coste que entraña el cableado. Esta solución dista mucho de ser idónea.

Red eléctrica La red eléctrica es accesible desde casi cualquier lugar en el hogar. La tecnología actual utiliza los cables de esta red para transmitir información, lo que elimina el coste de cablear una red paralela sólo para datos.

La utilización de esta red como soporte para la transmisión de datos se puede estudiar desde dos planteamientos muy diferentes. El primero consiste en la transmisión de señales de control y seguridad, y está encaminado a intercomunicar todos los electrodomésticos del hogar. El segundo es la transmisión de información de banda ancha (multimedia, Internet, etc.) entre los diversos ordenadores del hogar y ciertos electrodomésticos (equipos estéreo y televisores, fundamentalmente). En el primer caso, la velocidad de transmisión por la red eléctrica está por debajo de los 100kbps . En el segundo caso, la velocidad pronto alcanzará los 100Mbps .

Red telefónica El cable trenzado de la línea telefónica también puede utilizarse para transmitir datos. Por sus características, este cable resulta más apropiado para la transmisión de datos que la red eléctrica. Además, proporciona un acceso al exterior. Sin embargo, presenta el problema de que la línea telefónica no suele llegar a todas las habitaciones de la casa, por lo que su utilidad queda bastante limitada.

Los fabricantes de productos de red sobre la línea telefónica se han unido en la Home Phoneline Networking Alliance (conocida como HomePNA). La más veterana de estas empresas es Tut Systems, cuya tecnología a 1Mbps ha sido adoptada como primera versión de las especificaciones de la HomePNA. Por otro lado, Intel ha desarrollado un módem que funciona a 10Mbps .

Red inalámbrica La comunicación sin hilos a través de radiofrecuencia es otra de las soluciones propuestas. Los equipos de red que usan esta tecnología se comunican a través de la banda sin licencia de $2,4\text{GHz}$.

Existe una gran variedad de tecnologías de radiofrecuencia, pero la mayoría de ellas sigue el estándar IEEE 802.11 para redes inalám-

bricas. Este estándar define actualmente una velocidad de transferencia de $11Mbps$. El último de los estándares aparecido es Bluetooth, que permite una tasa de transferencia de $1Mbps$. Bluetooth está dedicado al mercado de consumo, por lo que prima su bajo coste.

Al igual que en el caso de la línea eléctrica, varios fabricantes se han unido en un consorcio llamado Home Radio Frequency Working Group, que ha propuesto un estándar para redes locales. Este estándar se llama SWAP (Shared Wireless Access Protocol), y ha sido diseñado para transmitir datos a $2Mbps$.

Intel comercializa diversos módem de radiofrecuencia que funcionan a una velocidad de $1,6Mbps$.

1.2 Redes de datos sobre la línea eléctrica

Todas las redes de datos se diseñan sobre una capa física de la que, en principio, son independientes. Tan sólo los elementos de esta última capa son específicos del medio físico por el que se transmite. Para poder transmitir datos digitales por una línea analógica, como la línea eléctrica, es necesario utilizar un módem que realice la traducción entre señales digitales y señales transmisibles por el medio.

1.2.1 Características de la línea eléctrica

Los cables de la red eléctrica están diseñados para transportar señales de potencia de muy baja frecuencia (50 ó $60Hz$). Por tanto, la respuesta en frecuencia de estos cables está muy atenuada para altas frecuencias. Esta atenuación se sitúa entre los -30 y los $-50dB$, lo que aún permite su utilización.

Sin embargo, aparecen dos problemas que complican el uso de la red eléctrica como canal de comunicación. El primero de ellos

es el efecto de todos los aparatos conectados a la red: los reguladores de luz y otros dispositivos controlados por triacs producen ruido armónico con la frecuencia de red ($50/60Hz$), las fuentes de alimentación conmutadas generan ruido en los armónicos de su frecuencia de conmutación; por último, los cepillos de los motores eléctricos producen ruido de banda ancha. El segundo de los problemas es que, aunque no haya ningún aparato conectado, se acopla una gran cantidad de ruido de RF en los cables de la red eléctrica, pues éstos no tienen ningún tipo de apantallamiento ni están trenzados.

Estos dos fenómenos provocan que la representación de la relación señal/ruido de la línea eléctrica frente a la frecuencia muestre unos picos y valles muy pronunciados y oscilantes, y que aparezcan depresiones profundas de más de $80dB$ a frecuencias variables.

A esto hay que unir el hecho de que la impedancia de la línea es cambiante en todo momento, ya que depende de los dispositivos conectados y no de la línea en sí. Además, la señal recorre múltiples caminos antes de llegar a su destino. Esto produce múltiples retardos de la señal, que si bien no afectan a una señal de $50Hz$, pueden producir interferencia entre símbolos (ISI) en la transmisión de datos. Estos hechos hacen que la red eléctrica se estudie como un caso muy parecido a la transmisión de señales de RF. De hecho, algunos fabricantes de módem de línea utilizan técnicas adoptadas de la RF para la transmisión por la red (Intellon con su técnica OFDM es uno de ellos).

1.3 Clasificación de los módem de línea

Los módem de línea se pueden separar en dos grupos bien diferenciados según su velocidad de transferencia:

Módem de baja velocidad Fueron los primeros en salir, y ya llevan varios años implantados en el mercado. Son módem

de muy baja velocidad (1200 a 9600*bps*) o de baja velocidad (100*kbps*) pero muy robustos, y por tanto son apropiados para tareas de control.

Módem de alta velocidad Se diferencian de los anteriores en velocidad, pues tienen tasas de transferencia superiores a 1*Mbps*, y en algunos casos llegan hasta 100*Mbps*. Aunque éstos, teóricamente, ya se encuentran desarrollados, no hay todavía ninguna empresa que los tenga disponibles para el mercado de consumo. Su utilidad se centra en la difusión de contenidos multimedia y el acceso a Internet de banda ancha.

1.3.1 Módem de baja velocidad

A continuación se describen diversos módem de línea, cuya velocidad de transferencia es baja, que actualmente se encuentran en el mercado.

ST7537 (SI Microelectronics)

Es un módem integrado en un sólo chip, con una velocidad máxima de 1200*bps* y semi dúplex; es decir, cada módem es capaz de transmitir y recibir pero no simultáneamente. Este circuito necesita una interfaz de línea con la red eléctrica que proporcione potencia para la transmisión y aisle el circuito de la red. Dispone de una salida de detección de portadora para evitar colisiones cuando se conectan varios módem en red. Por lo que respecta a su tecnología de modulación, señalar que este dispositivo transmite las señales mediante modulación FSK, con una frecuencia portadora única de 132,45*kHz*.

ST7536 (SI Microelectronics)

Es otro módem integrado en un sólo chip, sucesor del anterior, con una velocidad de transmisión seleccionable entre 600 ó 1200*bps*, y semi dúplex. Tiene un control automático de ganancia mediante una señal de realimentación que inyecta a la red para monitorizar la calidad de la señal.

Necesita únicamente, aparte del circuito de módem, unos pocos componentes (un cristal, cuatro resistencias y cinco condensadores) y un circuito de interfaz de línea (PLI, PowerLine Interface). El PLI consiste básicamente en una etapa de potencia para salida, un previo de entrada y un transformador para aislar e inyectar la señal a la red eléctrica. Funciona también mediante modulación FSK pero además permite seleccionar la frecuencia portadora de entre cuatro: 62, 72, 82 y 86*kHz*.

Pese a su baja velocidad de transferencia, ambos circuitos forman una alternativa sencilla y de muy bajo coste para ciertos tipos de red.

CEWay PL-One (Domosys)

Es un módem de línea que funciona bajo el estándar CEBus (EIA600). Este estándar define todas las capas de una red de datos a través de red eléctrica, con una velocidad de transferencia de unos 5*kbps*. El integrado contiene, además del circuito de módem, un núcleo de microcontrolador 8052, con 256 bytes de RAM, tres contadores, puerto serie semi dúplex, 15 terminales de E/S y posibilidad de direccionar 64*kB* de datos y 64*kB* de programa externos. La transmisión se hace mediante pulsos de barrido en frecuencia (“chirps”) de 100*μs* de duración. Se usan dos tipos de modulación de estos pulsos: en el preámbulo del paquete se modula en ASK, y los datos del paquete están modulados en PSK.

PLT-22 (Echelon)

Al igual que el anterior, este módem está concebido para funcionar dentro de un estándar de red, en este caso LonWorks. Este estándar contempla la implementación de una red de datos sobre varios medios físicos: par trenzado, línea eléctrica, radiofrecuencia, coaxial y fibra óptica. El módem PLT-22 realiza la interfaz entre el chip Neural (controlador de red) y el medio físico de la línea eléctrica. Tiene una tasa de transferencia de $5,4\text{kbps}$. El chip Neural está siendo fabricado por Toshiba. El módem contiene un DSP que realiza una modulación BPSK sobre dos frecuencias de portadora duales, entre las que se selecciona automáticamente la más adecuada en cada momento: 132 y 115kHz a 10MHz de reloj, y 75 y 86kHz a $6,5536\text{MHz}$.

Para funcionar necesita un chip Neural y un circuito de interfaz con la red.

SSC P485 (Intellon)

Utiliza la misma tecnología de modulación de banda ancha que recoge el estándar CEBus, pero permite la conexión a cualquier tipo de red, aunque en las hojas de características del fabricante no se reconoce. Se diferencia del CEWay PL-One en que no posee un núcleo microcontrolador interno, sino que debe conectarse a un microcontrolador externo. Utiliza la misma tecnología de portadora de banda ancha (SSC, spread spectrum carrier) que el estándar CEBus. Sin embargo, en las hojas de características anuncia una velocidad de transferencia de 9600bps en lugar de los cerca de 5kbps del estándar. Esto se debe a que consideran la velocidad de transferencia de los bits de cada carácter (raw data), sin considerar el preámbulo, los bits de parada y stop y el tiempo entre caracteres y entre mensajes.

SSC P200 y SSC P300 (Intellon)

Estos dos circuitos no son sólo módem, sino que integran toda la implementación de la capa física de una red de datos; es decir, realizan la interfaz con el medio físico y el control del flujo. Ambos utilizan el canal de la red eléctrica conforme al estándar EIA600 (CEBus). Sin embargo, el integrado P200 está pensado para ser utilizado con servicios no reconocidos por el estándar, mientras que el P300 está pensado para ser compatible con el resto de productos CEBus. Estos dos integrados utilizan la tecnología de banda ancha con señales “chirp” especificadas en el estándar EIA600 (CEBus).

PL-triple-C 250 (Polytrax)

Este sistema no es un circuito de módem, sino un módem completo integrado en una caja y listo para funcionar. Se comunica con un PC en serie a través de una línea RS232 y la velocidad de transmisión es superior a la de todos los anteriores: $25kbps$. Por lo que respecta a su tecnología de modulación, utiliza una combinación de modulación FSK y PSK entre 95 y $125kHz$.

Polytrax está preparando una segunda versión de su módem, que utilizará una modulación DMT con OFDM/QAM (Digital Multi Tone with Orthogonal Frequency Division Multiplex/Quadratic Amplitude Modulation), implementada en un DSP de Texas Instruments. Este módem funcionará a $155kbps$ utilizando el ancho de banda permitido en la normativa de la Unión Europea, y a $2,5Mbps$ en los casos de las normativas americana y asiática.

IT800 e IT5000 (Itran)

Estos dos circuitos integrados realizan todas las funciones de modulación y demodulación necesarias para la comunicación a través de la red eléctrica, y sólo requieren componentes pasivos para su

circuito de interfaz de la red. Ambos integrados son compatibles con el estándar CEBus (EIA600) pero también pueden funcionar a mayores velocidades de transmisión (*7kbps* para el IT800 y *50kbps* para el IT5000). Ambos integrados utilizan la modulación DCSK (Differential Code Shift Keying), que es una técnica de modulación de banda ancha compatible con el estándar CEBus.

AN192 (Adaptive Networks)

Consiste en un chipset de dos integrados: la interfaz de red PLC192 y el controlador DLP. Con ellos se obtiene una velocidad de transferencia de *19,2kbps*. El circuito PLC192 realiza la comunicación con la red eléctrica. El DLP, que contiene un núcleo de microprocesador 65C02, gobierna el integrado PLC192 y controla los protocolos de red. Utiliza una técnica de modulación de banda ancha con un sistema de sincronización adaptativa patentado.

AN1000 (Adaptive Networks)

Este módem es un chipset de tres integrados: los controladores de capa física PLC1000A y PLC1000D, y un procesador de aplicaciones e interfaz de capa de datos denominado DLP. Tiene una tasa de transferencia de *100kbps*.

Los modelos PLC1000A y PLC1000D realizan la función de transceptores con la línea eléctrica. El integrado DLP es el mismo utilizado en el módem anterior. Respecto a la tecnología de modulación, utiliza la misma técnica que el módem anterior.

AN48 (Adaptive Networks)

Este módem está formado por un chipset de dos integrados: la interfaz de red AN48 y el controlador DLP. Está diseñado para cumplir

la normativa europea CENELEC de ancho de banda, más restrictiva que la normativa americana, y por tanto sólo alcanza una velocidad de transferencia de $4,8\text{kbps}$.

Como en los dos casos anteriores, el circuito AN48 realiza la comunicación con la red eléctrica y el circuito DLP el control de los protocolos. Su tecnología de modulación es la misma que de los dos módem anteriores, pero utiliza un ancho de banda limitado entre 9 y 95kHz .

1.3.2 Utilidades de los módem de baja velocidad

Los módem de línea de baja velocidad se utilizan fundamentalmente para tareas de control. La aplicación más usual es el control remoto de luces: con un pequeño dispositivo colocado en el propio enchufe de la lámpara se puede controlar, desde cualquier otro enchufe de la casa, el encendido/apagado de la lámpara o su intensidad. Ver la figura 1.1 página 31.

Otras aplicaciones de control, no tan básicas, son el control remoto de electrodomésticos: encendido de la lavadora a la hora programada, o apagado automático de la plancha al salir de casa. Conectando los electrodomésticos al ordenador, y éste a Internet, se puede acceder desde el trabajo a los electrodomésticos. Esto sirve, por ejemplo, para programar la calefacción para que se encienda al salir hacia casa.

También se puede implementar un sistema de alarma sin necesidad de cableado y que, además, permita la supervisión del estado desde cualquier lugar a través de Internet.

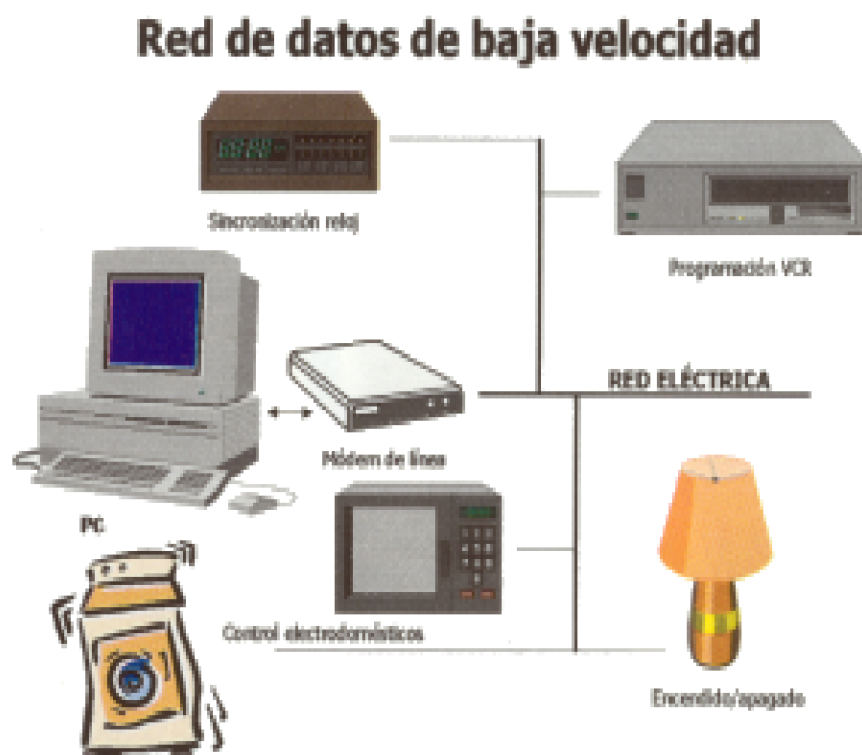


Figura 1.1: Aplicaciones de una red de datos de baja velocidad.

1.4 Módem de alta velocidad

Los módem de alta velocidad a través de la línea eléctrica (power-line modem) se encuentran todavía en fase de desarrollo. A continuación se presentan diferentes fabricantes que ofrecen estos productos, aunque las hojas de características publicadas todavía son preliminares y, en la mayoría de los casos, aún no están disponibles sus productos.

Intellon

Aparte de fabricar los módem de baja velocidad citados anteriormente, esta empresa ha desarrollado una tecnología de modulación

para módem de línea a alta velocidad denominada OFDM. Sin embargo todavía no ha presentado ningún circuito de módem de banda ancha con esta tecnología. La modulación OFDM (Orthogonal Frequency Division Multiplexing) consiste en dividir el ancho de banda disponible en una serie de subbandas portadoras. La información se envía modulada con cualquier técnica digital y repartida entre todas las subportadoras a la vez, con una capacidad de transferencia de datos baja para cada una. En total se mandan tantos bits como el número de subportadoras multiplicado por los bits codificados en cada banda (esto depende de la técnica digital utilizada: por ejemplo, en QPSK son *2bits*, y en 64-QAM son *6bits*).

Inari

Inari es el proveedor de silicio de Intelogis, una compañía que fabrica dispositivos de red sobre la línea eléctrica. Actualmente su único producto es un controlador de redes denominado IPL0201. Intelogis comercializa utilizando tecnología de Inari un módem de línea completo denominado PassPort. Este producto intercomunica PCs entre sí y PC con impresoras a través de un cable paralelo y la red eléctrica.

El IPL0201 es un controlador de red sobre la línea eléctrica que funciona con una velocidad máxima de transmisión de *2Mbps*. Integra en un solo chip los siguientes elementos: transceptor con la línea eléctrica, núcleo microcontrolador compatible 8051, interfaz con un controlador (comunicación USB y paralelo), circuito de detección de errores y criptografía, y protocolo de intercambio por la red eléctrica. Necesita un circuito externo que realice el filtrado de la señal de entrada en cuatro subbandas, la amplificación de la señal de salida y el transformador de aislamiento. El módem transmite la información mediante cuatro portadoras de distinta frecuencia moduladas en BPSK o QPSK.

Esta empresa ha llegado a un acuerdo recientemente con Texas Instruments para el desarrollo y fabricación de la segunda generación de sus módem de línea.

Design of Systems on Silicon (DS2)

Esta empresa española está desarrollando dos circuitos de módem de línea: un módem de banda estrecha y un módem de banda ancha que alcanzará los *75Mbps* (actualmente alcanza *30Mbps* en pruebas). En breve se lanzará el DSS4200, primer chip de la compañía.

Los módem de DS2 están diseñados para proporcionar un servicio de conexión a Internet a través del cable de la compañía de suministro eléctrico. El módem de banda ancha de DS2 se conecta en la subestación transformadora, y da servicio con un único módem a un elevado número de usuarios que disponen en su casa de un módem de banda estrecha. Un protocolo maestro/esclavo entre estos dos módem garantiza la mejor utilización del canal asignado a cada usuario. El módem de la subestación eléctrica se conecta a Internet mediante un router de Internet o un conmutador ATM. También provee servicios de videoconferencia, vídeo por demanda, voz sobre IP, etc. para un mismo usuario en cualquier punto de su instalación eléctrica.

Utiliza una técnica de banda ancha OFDM (Orthogonal Frequency Division Multiplexing), similar a la utilizada por las tecnologías XDSL. Los módem dividen el espectro en 1280 tonos OFDM, por los cuales se transmite la información digital en paralelo. La capacidad de transferencia por portadora es variable y adaptativa dependiendo de la SNR, con lo que se pueden evitar frecuencias problemáticas con el ruido y desacoplo de impedancia.

Con la tecnología de DS2, múltiples usuarios (hasta 50) se conectan a un solo módem de banda ancha en la subestación eléctrica. Por

contra, la tecnología XDSL precisa un módem en la central telefónica por cada usuario.

El módem de DS2 proporciona una calidad de servicio (QoS, Quality of Service) adaptativa, de manera que se podrá configurar para distintos anchos de banda según la velocidad contratada por el usuario doméstico.

Itran Communications

Aparte de los módem IT800 e IT5000, ya vistos, la empresa israelí Itran Communications ha desarrollado un módem de banda ancha a $2,5Mbps$, y tiene previsto sacar pronto un nuevo producto a $12Mbps$. Ambos están pensados para soluciones de bajo coste.

ITM1 Este circuito integra un módem de línea completo a $2,5Mbps$. Para funcionar sólo necesita unos cuantos componentes pasivos externos para su conexión a la red eléctrica y un microcontrolador, con el que se comunica a través de un bus SPI.

Itran utiliza en sus módem de banda ancha una técnica de modulación propia denominada ACSK (Adaptive Code Shift Keying), que es una variante no convencional de la técnica de banda ancha de secuencia directa.

ITM10 Este módem tendrá una velocidad de transferencia de $12Mbps$, de acuerdo con las previsiones actuales. Al igual que el modelo anterior, para tener un coste bajo no integra ningún microcontrolador, por lo que requiere uno externo para funcionar. Este circuito dispone una interfaz paralelo con el microcontrolador. Utilizará la misma técnica ACSK con algunas mejoras para aumentar la tasa de transferencia.

Adaptive Networks

Esta empresa dispone en la actualidad de los tres módem de baja velocidad, ya vistos, y está desarrollando un módem de alta velocidad.

El módem de alta velocidad HSP alcanzará una tasa de *20Mbps* (raw data). Está compuesto por un chipset de dos integrados: el Powerline Signal Processor (PSP) y el Analog Front-End (AFE). El primero de ellos es un controlador del protocolo de red, y el segundo es un transceptor con la red eléctrica. La interfaz con el controlador PSP es un puerto serie USB. Utiliza una técnica de modulación optimizada de banda ancha con sincronización y ecualización adaptativas. El fabricante asegura una tasa de errores (Bit Error Rate) menor de 10^{-9} , la misma tasa que proporciona un cable dedicado.

Esta técnica permite la transmisión de los datos en paquetes muy cortos gracias a la sincronización rápida, lo que disminuye la probabilidad de que se corrompan. Este enfoque es completamente opuesto a la técnica OFDM de Intellon, que basa su robustez en la transmisión de los datos en símbolos muy largos para que no sean degradados por espureos de la línea.

1.4.1 Utilidades de los módems de alta velocidad

Los módem de alta velocidad pueden transmitir los mismos contenidos que una red de datos convencional, como archivos compartidos entre computadoras y acceso a Internet de alta velocidad. Estos datos tienen como emisor y destinatario dos ordenadores.

Pero la utilización de la red eléctrica abre las puertas a un nuevo tipo de transmisión de datos multimedia entre un ordenador y electrodomésticos preparados para interpretarlos. Esto permite la difusión de música en formato MP3 a diversos reproductores conectados en cualquier lugar de la casa, o la transmisión de vídeo digital

en MPEG hacia los televisores del hogar. Ver la figura 1.2 página 36.



Figura 1.2: *Aplicaciones de una red de datos de alta velocidad.*

Capítulo 2

Microcontroladores

2.1 Introducción

En la actualidad los microcontroladores (μC) se pueden encontrar en cualquier parte del campo de las aplicaciones electrónicas. Por ejemplo, los μC son utilizados en el control de la caja de cambios en un automóvil, una cámara digital, una lavadora, etc. A pesar de que los μC habían sido utilizados en el pasado principalmente en el ámbito de la electrónica industrial, se hicieron populares y su uso se extendió a otros campos gracias al abaratamiento del silicio, a la posibilidad de una mayor integración de componentes, tales como el convertidor A/D y memoria externa, y a la capacidad de una mayor densidad de integración de circuitos. De aquí la existencia de dispositivos cada vez más pequeños y potentes que son capaces de llevar a cabo una gran cantidad de tareas en un espacio muy reducido.

2.2 Definición de microcontrolador

Un μC es un circuito integrado que incluye, en un único chip, todos o la mayoría de componentes de un controlador, es decir, un

dispositivo que se utiliza para controlar objetos, procesos o eventos. La figura 2.1 página 38 muestra el diagrama de bloques típico. Típicamente, un μC incluye todos aquellos elementos propios de un sistema procesador:

- CPU (Central Processing Unit)
- Memoria RAM
- Memoria ROM/PROM/
- EPROM/E²PROM
- Puertos de E/S serie y paralelo
- Temporizadores/Contadores
- Convertidor A/D
- Controlador de interrupciones



Figura 2.1: Diagrama de bloques típico de un μC .

Puesto que la cantidad de memoria y el número de periféricos internos del μC es limitada, microcontroladores tienden a ser utilizados en pequeños sistemas que requieren poco más que el μC y unos cuantos componentes de soporte específicos a la tarea de control. Como resultado, el precio de estos sistemas de control suele ser de relativamente bajo coste y su tamaño de reducidas dimensiones.

2.3 Aplicaciones de los microcontroladores

El campo de aplicación de los μC ha ido extendiéndose paulatinamente desde su aparición. Así, hoy en día cualquier sistema que mida, almacene, controle, calcule o presente información es un candidato a llevar un μC en su interior. Sólo unos cuantos ejemplos los encontraríamos en electrodomésticos (hornos microondas, frigoríficos, televisores, cámaras de vídeo, equipos de música), equipos informáticos (impresoras, módems, disqueteras), automóviles (control de inyección, diagnóstico, climatización), control medioambiental (invernaderos, fábricas, viviendas), instrumentación, sistemas aeroespaciales y otros muchos más. También, en muchos de los ejemplos anteriores sería posible encontrar más de un μC compartiendo e intercambiando información entre ellos y gobernados por un procesador central más potente.

Mientras que en los microprocesadores el objetivo está fijado en alcanzar la máxima capacidad de procesamiento de información, en los μC el objetivo consiste en implementar un conjunto de funciones de control de la forma más efectiva económicamente. Así pues, las aplicaciones más apropiadas para el uso son aquellas en las que la necesidad de procesamiento de información no es muy alta. Además, al ser pequeño el tamaño de los μC y consumir generalmente muy poco esto los hace ideales para sistemas portátiles y autónomos. Una aplicación especial de los μC es el registro de datos como puedan ser en una estación meteorológica la temperatura, la humedad, la

lluvia, la velocidad del viento, etc. En una aplicación típica, el micro tiene que gestionar varias tareas según la prioridad de cada una o la aparición de sucesos externos: una nueva orden entrada desde el teclado, subir la temperatura externa, etc.

2.4 Mercado de los microcontroladores

Cada fabricante de ofrece en sus distintas familias una gran variedad de versiones de un mismo μC , con distinta capacidad y tipo de memoria, encapsulado y un número y tipo distinto de periféricos incluidos en el chip. Sin embargo, es posible clasificar a todos ellos atendiendo al número de bits de sus registros internos, lo que también se conoce como ancho de palabra del dispositivo. Según este concepto, el mercado de los μC podría clasificarse en μC de 4, 8, 16 y 32 bit.

En la gran mayoría de las aplicaciones un circuito de 4 u 8 bit puede ser más que suficiente, mientras que los de 16 y 32 bit se reservan para aquellas aplicaciones que requieran una alta capacidad de procesamiento y un nivel de integración elevado. En la figura 2.2 página 41 se puede apreciar que más de la mitad del mercado está dominado por los μC de 8 bit, y la tendencia es que siga creciendo. También se observa que a pesar de su sencillez los de 4 bit conservan su sector de mercado gracias al elevado número de aplicaciones sencillas existentes y a la tendencia de incorporar μC cada vez más en los pequeños sistemas de control.

El sector del automóvil es el principal responsable del crecimiento del mercado de los μC . Varias de las familias de μC fueron especialmente diseñadas para aplicaciones dentro del sector del automóvil. Éste es, además, un sector muy exigente. La electrónica debe operar bajo temperaturas extremas y ser capaz de soportar vibraciones, golpes e interferencias electromagnéticas. El hecho de que el μC in-

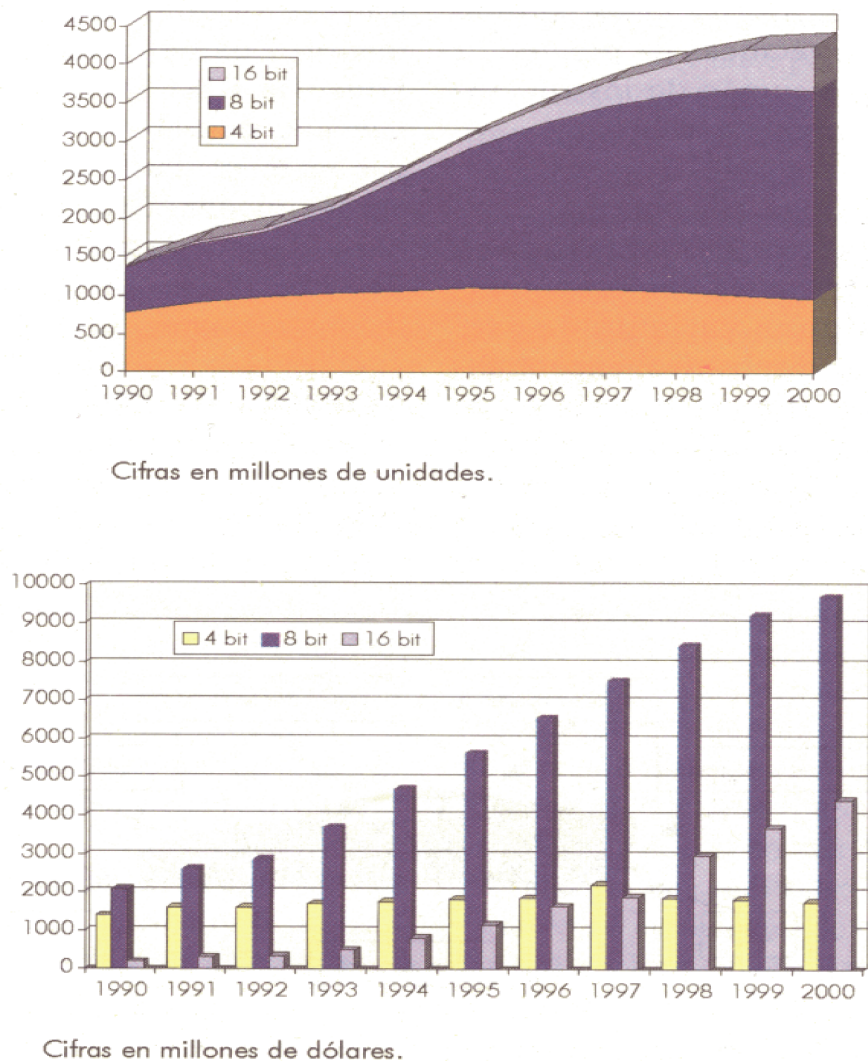


Figura 2.2: Mercado mundial de microcontroladores: a) en número de unidades, b) por volumen de negocio.

tegre en su interior varios periféricos hace que la electrónica de todo el sistema sea más fiable al estar formada por un menor número de componentes discretos, reduciendo así, o eliminando, la mayor parte de los problemas anteriores. El mercado del automóvil no es el único que está creciendo. Se estima que actualmente en cada casa de los

países desarrollados hay del orden de 50 μC y que para finales del año 2005 este número se acerque a los 250.

Además, no sería de extrañar que las previsiones de crecimiento anteriores se quedaran cortas puesto que desde hace relativamente pocos años una serie de empresas, desde fabricantes de chips hasta proveedores de servicios, agrupadas en consorcio bajo las siglas ETI (Extend The Internet) están colaborando para promover la tecnología EMIT de la compañía emWare como estándar industrial para conectar dispositivos a la red. La tecnología EMIT proporciona los elementos necesarios para conectar cualquier dispositivo de forma rápida y económica, y funciona con la mayoría de μC de 8 bit en adelante haciendo que se comporten como microservidores de la red, facilitando así un acceso remoto a la aplicación desde cualquier punto de la red. Todo ello constituye un aliciente que puede animar a la inclusión de μC en sistemas en los que todavía él no está presente.

2.5 Tipos de memoria

Una de las consideraciones en la elección del dispositivo más adecuado para una aplicación en particular es la cantidad de memoria necesaria para almacenar el programa de control de la aplicación y aquellos datos necesarios de la aplicación y su control. A diferencia de los grandes sistemas basados en microprocesador, la gran mayoría de los μC se caracterizan por incluir en el propio chip una cierta cantidad de memoria cuyo tamaño y tipo suele variar entre los distintos miembros de una misma familia. No obstante, también algunos de ellos permiten el conexionado de memoria externa como un componente a parte.

La memoria EPROM ha sido por mucho tiempo el método más popular para almacenar el código del programa de la aplicación. Además de la EPROM, existen otros tipos de memoria que junto

con ésta pasamos a describir a continuación.

El tipo de memoria a utilizar está normalmente ligado al número de unidades que se van a comercializar. Así, de menor a mayor número se utilizaría una memoria EPROM, OTP o ROM. Por otro lado, el uso de la memoria EEPROM queda normalmente restringida a la fase de diseño de los primeros prototipos de la aplicación. Debido a que ésta puede ser grabada y borrada con facilidad se van a poder realizar todas aquellas modificaciones en el programa de la aplicación hasta que se cumpla con las especificaciones de diseño.

2.5.1 EPROM (Erasable Programmable Read Only Memory)

Se trata de una memoria no volátil. Durante el proceso de grabación, los datos son guardados en las direcciones deseadas, mediante la aplicación señales de control y tensiones especiales de programación de valor más elevado al normal. Su contenido puede ser borrado mediante la exposición a rayos ultravioleta de la ventana que posee el chip de memoria. Típicamente estas memorias pueden ser grabadas y borradas al menos unas 100 veces y a menudo muchas más.

2.5.2 OTP (One Time Programmable)

Similar a la memoria EPROM pero su encapsulado no dispone de ventana por lo que no es posible borrar su contenido. Esto la hace más barata que la EPROM y se trata de una buena opción para almacenar el programa de la aplicación una vez ya finalizado.

2.5.3 E²EPROM (Electrically Erasable Programmable Read Only Memory)

También conocida como E²EPROM. Memoria similar a la EPROM excepto que puede ser borrada eléctricamente. Limitaciones de este

tipo de memoria incluyen baja velocidad, coste elevado y un número limitado de veces que puede ser grabada y borrada, típicamente entre 10.000 y 100.000 veces.

2.5.4 ROM

Este tipo de memoria es rentable cuando el número de unidades a producir es elevado. Esta memoria posee, sin embargo, un inconveniente ya que el proceso de grabación se realiza en fábrica y una vez grabada no es posible modificar su contenido. En consecuencia, si se descubre un error en el programa habrá que desechar todos los μC que ya hayan sido grabados.

2.5.5 Flash EPROM

Se trata de una memoria que puede ser borrada eléctricamente y que está desplazando a las EEPROM. Proporciona una mejor solución que las EEPROM especialmente cuando se trata de grandes cantidades de memoria pues su borrado es más rápido y puede hacerse por grandes bloques o bien borrando directamente todo su contenido.

2.5.6 RAM

Se trata de memoria volátil. A diferencia de los otros tipos de memoria, su contenido desaparece cuando se quita la alimentación a menos que se conecte a una batería. Este tipo de memoria se utiliza para almacenar datos de forma temporal y no para guardar el programa de la aplicación como ocurre en los sistemas microprocesador.

2.6 Periféricos internos

Los μC suelen incluir en su interior periféricos especiales que facilitan la tarea de control de la aplicación. El tipo y número de ellos

varía según las necesidades de la aplicación para la cual acostumbran a estar pensados. A continuación, y sin pretender hacer una lista exhaustiva, se describen los más comunes:

2.6.1 Convertidor A/D

Convierte una señal analógica externa, típicamente una tensión, en una representación digital. Los μC que poseen este periférico pueden ser utilizados en instrumentación, registro de datos externos o cualquier aplicación que implique el contacto con el mundo analógico.

2.6.2 Temporizador/contador

Este periférico hace posible la medida del intervalo de tiempo entre dos sucesos, activar/desactivar señales al cabo de un cierto tiempo o bien contar el número de veces que se produce un suceso, que normalmente se trata de un flanco de subida/bajada ocurrido en una determinada señal de control. Sus prestaciones suelen variar bastante de un μC a otro.

2.6.3 Temporizador con auto recarga

Comparado con un temporizador clásico, este temporizador se recarga automáticamente a su valor inicial cuando se alcanza el final de la cuenta, liberando así a la CPU de esta tarea.

2.6.4 Temporizador watchdog

Se trata de un temporizador con funcionamiento totalmente en paralelo con el μC que lo reinicializa si el programa no lo refresca a tiempo. Proporciona un método de recuperación del control del programa de la aplicación en el caso que se produzca un fallo de

funcionamiento. Este periférico es especialmente útil en aquellos entornos con fuertes interferencias electromagnéticas o bien en sistemas de control autónomos que no están sometidos a una vigilancia continuada.

2.6.5 Puertos de E/S

La mayoría de μC poseen varios puertos de E/S digital que permiten gestionar, por ejemplo, LED o la entrada desde un teclado. Generalmente, un puerto consiste de 8 o menos bit que pueden ser programados como líneas de entrada o salida.

2.6.6 Unidad de comparación y captura

Periférico utilizado en cualquier clase de generación de señales digitales y captura de sucesos como puedan ser la generación de pulsos, modulación de anchura de pulsos, medida de anchura de pulsos, etc. Su uso está muy extendido en las aplicaciones de control en el sector del automóvil (control de inyección, ABS, etc.) así como en aplicaciones industriales (control de motores paso a paso, generación de frecuencias, conversión digital-analógica, etc.).

2.6.7 Interface serie

El interface serie es utilizado para intercambiar datos con el exterior. Muchos μC poseen tanto periféricos de comunicaciones asíncronos (SCI o UART) como síncronos (SPI). Además, la fuerte implantación de los μC en el sector del automóvil y en general en el mercado industrial hace que muchos de ellos incorporen buses específicos de comunicaciones como el I² o el CAN.

2.7 Los microcontroladores más populares

A continuación se realiza una breve descripción de los μC que con el tiempo se han convertido en los más populares hasta la fecha. La pregunta que uno se hace en el momento de decidir trabajar con un μC es: ¿qué dispositivo debería utilizar para...? La respuesta no es fácil y a veces la decisión no está fundamentada exclusivamente en las necesidades de la aplicación a diseñar sino que en ella intervienen otros factores. El mejor consejo sería escoger aquel μC que tuviera un conjunto completo de herramientas de desarrollo a un buen precio y una buena documentación: manuales, notas de aplicación, etc. Las herramientas de desarrollo pueden ser decisivas en la elección, ya que pueden suponer una ayuda magnífica en el desarrollo de la aplicación. También se deberá tener en cuenta el soporte técnico que se pueda recibir del fabricante así como la disponibilidad del circuito en sus distintas versiones (OTP, ventana y máscara), o el hecho de que se disponga de módulos de programas ya realizados con otros μC y que se puedan aprovechar. Asimismo, hay que tener presente el coste de todos los componentes y que a veces el uso de un μC más barato puede encarecer los demás componentes de la aplicación.

No obstante, las características técnicas del componente escogido sí que pueden ser determinadas atendiendo a los requisitos de la aplicación. Así, el número de bits del μC vendrá fijado por la precisión de los datos a manejar. Sin embargo, si existen partes críticas de la aplicación que deban ejecutarse en un tiempo limitado es posible tener que acudir a un μC de mayor número de bits, 16 o 32, que normalmente son mucho más potentes. La cantidad y tipo de líneas o puertos de E/S del μC vendrán fijadas por la cantidad y la naturaleza de las señales a controlar involucradas en la aplicación. Para su determinación lo mejor es realizar un diagrama de bloques detallado en donde aparezcan todas estas señales. Cuando la apli-

cación vaya destinada a un sistema portátil y alimentado a baterías será muy importante que el μC escogido posea modos de operación de bajo consumo y posibilidad de operar con tensiones inferiores a los tradicionales 5V. El tamaño de la memoria se puede estimar basándose en el número y extensión de funciones, procedimientos y módulos distintos que contendrá el programa de la aplicación, distinguiendo entre memoria RAM y no volátil. Recuérdese que el tipo de esta última está generalmente ligada al número de unidades que se van a comercializar.

Casi todos los μC existentes en el mercado están basados en el concepto CISC. Un micro CISC típico puede llegar a sobrepasar las 80 instrucciones, muchas de ellas muy potentes y especializadas para la realización de tareas de control específicas. La ventaja de la arquitectura CISC es que las instrucciones son tratadas como macros permitiendo al programador utilizar una instrucción en lugar de otras muchas pero más simples.

Sin embargo, la tendencia del mercado es hacia los dispositivo tipo RISC. Estos poseen un reducido juego de instrucciones lo que hace que su unidad de control interna utilizada en la decodificación de instrucciones sea más simple, menor tamaño de silicio pudiendo dedicar a integrar un mayor número de periféricos. Esta sencillez en la unidad de control hace que el μC pueda trabajar a frecuencias mayores que los de tipo CISC. Los beneficios de la arquitectura RISC se traducen en un menor tamaño del chip, mayor velocidad (la ortogonalidad del juego de instrucciones permite adoptar técnicas de pipeline en la ejecución de instrucciones) y menor consumo.

8051

Originariamente de Intel, en la actualidad muchos son los fabricantes que proporcionan alguna variante de este popular dispositivo, encontrándose en el mercado más de un centenar de variantes diferentes,

prácticamente cada aplicación tiene el μC que mejor se ajusta a sus necesidades. Se trata de un elemento fácil de programar y potente que dispone de una gran cantidad de software tanto comercial como de libre distribución y de herramientas de desarrollo.

68HC11

Se trata de otro μC popular y potente con 8 bits de datos y 16 bits de direcciones fabricado por Motorola y Toshiba. Es posible encontrarlo en una gran variedad de versiones con distintos tipos de memoria y número de periféricos internos.

PIC

Los PIC de Microchip fueron los primeros basados en la filosofía RISC. Poseen un reducido número de instrucciones, del orden de 30 frente a las más de 80 en otros μC tipo CISC. Sus buses separados para instrucciones y datos, arquitectura Harvard, permiten un acceso simultáneo a instrucciones de programa y datos, y un solapamiento en la ejecución de instrucciones gracias a su pipeline de dos etapas, lo que propicia una velocidad de proceso mayor. Los beneficios de su filosofía RISC incluyen frecuencias de trabajo más elevadas, un menor tamaño de encapsulado, menor número de patas, sólo 8 patas en la familia PIC 12C5XX, así como un menor consumo.

COP400 y COP800

National Semiconductor ofrece estas dos familias de populares μC . La COP400 es una familia de 4 bit. Lejos de ser considerada una vieja tecnología, cada vez son más las aplicaciones existentes en el mercado destinadas a este tipo de μC . La familia COP400 está formada por más de 60 versiones distintas y disponibles a partir de 0,5 dólares. La familia COP800 provee soluciones de bajo coste

en aquellas aplicaciones que requieran chips de 8 bit con un amplio abanico de características. Posee un juego de instrucciones poderoso y eficiente. La mayoría de las instrucciones constan de un sólo byte y requieren un sólo ciclo máquina para su ejecución. Esta familia incorpora una amplia variedad de configuraciones de memoria y de periféricos.

ST6

Se trata de un μ C de 8 bit fabricado por STMicroelectronics orientado a aplicaciones de baja/media complejidad que se encuentra disponible en versiones EPROM, E²PROM y OTP. Existen varias versiones de este micro que incluyen temporizadores, convertidores A/D y líneas de puertos de E/S con capacidad de absorber hasta $20mA$, ideales para poder conectar LED sin prácticamente circuitería adicional. Poseen además un gran número de herramientas de desarrollo.

2.8 Herramientas de desarrollo

Una causa fundamental en el éxito de un μ C ha sido y es la disponibilidad de herramientas de desarrollo que permitan completar con éxito el diseño tanto software como hardware de la aplicación.

2.8.1 Herramientas de software

Ensamblador

Realiza la traducción de los programas escritos en lenguaje ensamblador a código máquina. Suele soportar directivas que facilitan el desarrollo del programa y permite definir constantes, variables, direcciones, directivas de ensamblaje, etc.

Linker (enlazador)

Compone los diversos módulos objeto generados por el ensamblador en un fichero ejecutable directamente por el μ C de destino.

Simulador

Permite comprobar el funcionamiento correcto del programa sin tenerlo que ejecutar directamente sobre el μ C de destino. Normalmente posee facilidades para el control de ejecución (paso a paso, breakpoints, etc.) y permite la visualización on-line de los registros internos del micro, contenidos de posiciones de memoria, etc.

Compilador

La programación del sistema puede hacerse utilizando un lenguaje de alto nivel. Uno de los más populares es el lenguaje C, que permite un acceso más simple a los recursos hardware.

2.8.2 Herramientas de hardware**Emulador**

Se trata de una herramienta de desarrollo potente que simula físicamente el comportamiento del μ C, de forma que se puede probar el programa de la aplicación sobre el hardware desarrollado sin tener que conectar el μ C. El emulador reproduce todas las señales eléctricas que produciría el chip si estuviera conectado. Esta herramienta es normalmente bastante cara.

Placas de evaluación

Se trata normalmente de pequeñas placas en las que se encuentra montado un μ C junto con un reducido hardware auxiliar como

pueden ser LED, pulsadores, puertos de comunicaciones, etc. y en el que además se tiene un fácil acceso a cada una de las patas o patillas del μC . El programa de la aplicación se carga desde un PC y se ejecuta sobre el micro de la placa. Generalmente estas placas permiten también realizar un depurado de la aplicación o simplemente comprobar si el μC escogido tiene las suficientes prestaciones para ejecutar la aplicación en particular que se esté diseñando.

Analizador lógico

Es el equivalente digital al osciloscopio. Un número de canales, normalmente elevado permite monitorizar diversas señales del sistema. La visualización se puede hacer en forma de cronograma o bien en representación numérica. La potencia del analizador lógico viene dada por la capacidad de almacenar acontecimientos previos o posteriores a una condición de disparo que establece el usuario, lo cual facilita mucho la tarea de comprobación y la detección de fallos bien sea en el hardware o en el software.

Capítulo 3

La fuente de alimentación

3.1 Introducción

El diseño de la fuente de alimentación constituye el primer paso de la construcción del sistema global. Partimos de la señal de la red eléctrica como fuente de energía para nuestros dispositivos, esta es, una señal alterna y de relativa alta tensión comparada con los niveles de tensión continua que queremos obtener. Por tanto necesitamos realizar dos pasos fundamentales en el diseño de la fuente de alimentación: como primer paso debemos reducir la tensión de la señal alterna a un valor útil para nuestros dispositivos, y como segundo paso debemos rectificar la señal y disminuir su rizado para obtener una señal de tensión continua.

Una forma muy común de realizar una fuente es usar básicamente un transformador y un puente de diodos con lo que además conseguimos aislamiento eléctrico entre la red de distribución eléctrica y el resto del sistema, un punto que aunque no parece primordial a priori, veremos resaltada su importancia a lo largo de este proyecto.

Ahora bien, nosotros nos hemos decantado por otro tipo de fuente más reducida aunque a costa de sufrir serias limitaciones.

3.2 Diseño sin transformador

Nuestra premisa fundamental ha sido integrar todo el sistema en un reducido espacio. Bien es cierto, que en los prototipos realizados no nos hemos preocupado del tamaño final, solamente de su funcionalidad y versatilidad. Con esta idea en la mente hemos desarrollado una fuente de alimentación sin transformador a pesar de todos los problemas que supone a priori, y los que nos encontramos durante su diseño.

La fuente de alimentación dispone de una resistencia de potencia y un condensador de alta tensión y capacidad colocados a la entrada. Ver la figura 3.1 página 55. A continuación un diodo rectifica y un zener reduce la tensión alterna hasta un valor determinado, necesario para alimentar a los diferentes circuitos integrados del sistema. Además, en paralelo con el diodo zener, hemos colocado un condensador de alta capacidad que filtra el rizado que existe sobre la señal de continua. De esta forma la tensión de red es reducida, rectificada y estabilizada a una tensión continua determinada por el diodo zener.

Para emitir por la red eléctrica usamos el CI LM555, para recibir usamos un decodificador de tono CI NE567; y para llevar el control de los distintos dispositivos tenemos un microcontrolador PIC 16F84A.

El primero lo alimentamos a $16V$ y los otros dos los alimentamos a $5V$. Así nos encontramos que necesitamos dos tensiones distintas de alimentación, y con el fin de usar una única fuente de alimentación, hemos colocado como diodo zener uno de $16V$ y a continuación añadimos un divisor de tensión realizado en su camino a tierra con otro diodo zener de $5V$ y por supuesto en paralelo a ambos zener colocamos dos condensadores de muy alta capacidad ($1mF$) para eliminar el rizado, en la medida de lo posible. De esta forma entre extremos de cada uno de los dos diodos zener obtene-

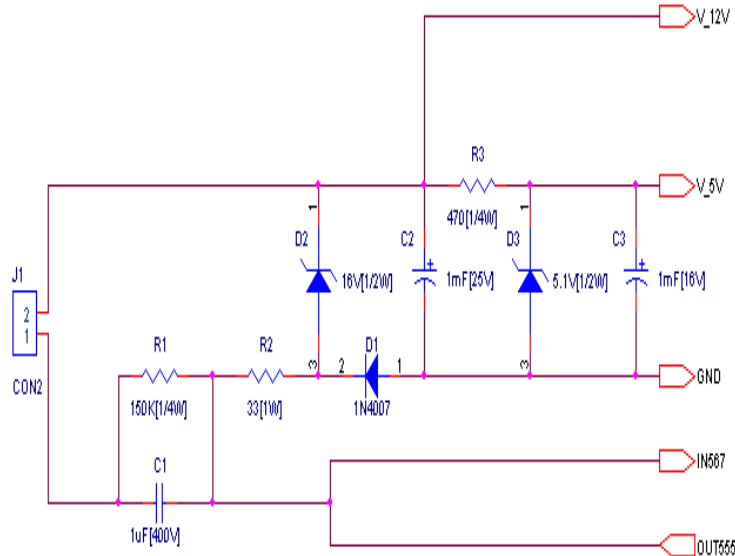


Figura 3.1: *Esquema de la fuente de alimentación.*

mos las tensiones de alimentación continua que necesitamos para alimentar a nuestros circuitos integrados consiguiendo además una cierta independencia de la carga en la corriente suministrada.

3.3 Problemas en el diseño

3.3.1 Potencia

El circuito de alimentación sin transformador supone un gran ahorro de espacio, pero resulta desventajoso desde el punto de vista de la potencia que suministra. Esta se ve limitada por la corriente que es capaz de atravesar el diodo zener. El mayor problema está en dar una buena corriente de alimentación al CI LM555 para que pueda

introducir una señal de buena calidad en la red eléctrica. Esto es algo fundamental debido a la gran cantidad de ruido que nos encontramos en la red eléctrica. Por ello el diodo zener que pusimos en la fuente es un diodo de potencia con el fin de obtener la máxima corriente de alimentación.

3.3.2 Aislamiento

Otro problema básico que nos produce el uso de una fuente de alimentación sin transformador es la falta de aislamiento eléctrico entre las fases de la red eléctrica y nuestro circuito electrónico. Por ejemplo, para poder testear nuestros diseños necesitamos y utilizamos un osciloscopio, aquí sale el problema a relucir, puesto que la tierra de la sonda se corresponde con la tierra del propio enchufe macho del osciloscopio a la red eléctrica y ante la ausencia de transformador, según ponemos la sonda sobre un punto del circuito estamos cerrando un bucle que produce un cortocircuito. Por ello nos hemos visto obligados a usar un transformador entre la sonda y nuestros circuitos para evitar este efecto y poder trabajar con el osciloscopio. Ver figura 3.2 página 57, este transformador consiste en un par de bobinados de cable sin núcleo, uno de salida y otro de entrada a la red eléctrica, con una relación de transformación de $1 : 1$ y a los que hemos añadido un filtro pasa alta que nos elimine la componente de $50Hz$ de la red. Cabe destacar que, la introducción de este circuito como interfaz de medida, introduce numerosos errores en las medidas que hemos tenido que tener muy presentes al observar los resultados obtenidos.

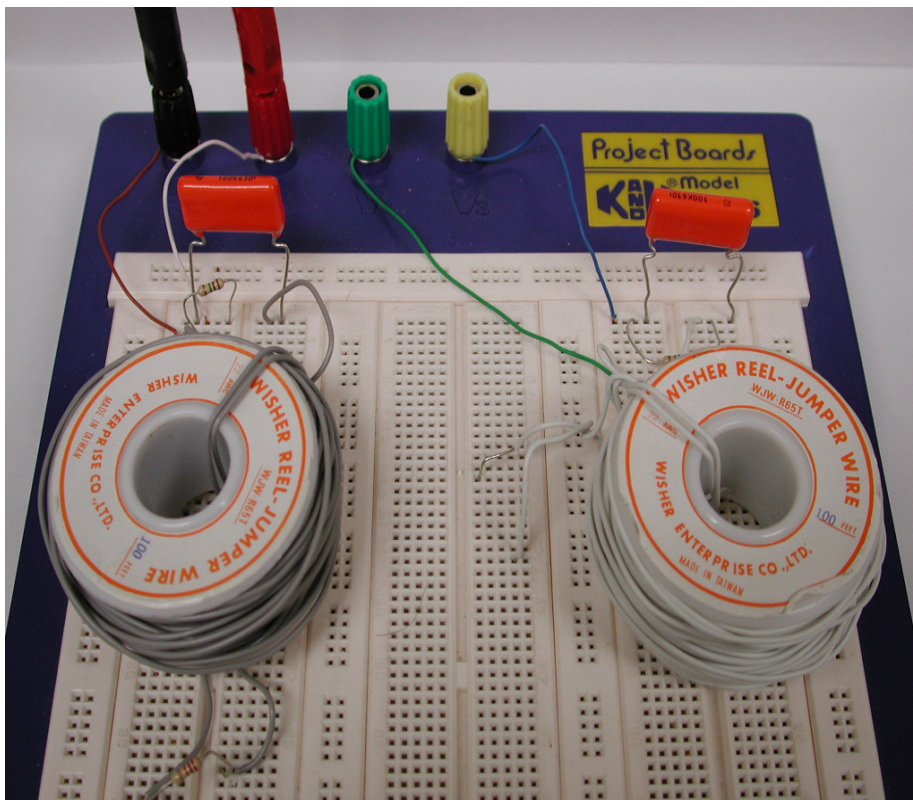


Figura 3.2: Transformadores de aislamiento de la red eléctrica.

Capítulo 4

El emisor de señal

4.1 Introducción

Nuestro sistema consiste en una serie de dispositivos colocados en diferentes enchufes de una casa que intercambian información usando como base el cableado de la red eléctrica. El emisor es el módulo que se encarga de emitir la señal que se transmitirá por la red de distribución eléctrica y que supone la base de las comunicaciones entre los diferentes dispositivos de nuestro sistema.

Teniendo en cuenta que la red eléctrica transporta una señal de $220V$ sobre una señal alterna de $50Hz$, hemos tomado un tono puro de frecuencia $36KHz$, muy lejos de la señal de red y sus armónicos principales, como base de nuestras comunicaciones. Para generar esta señal hemos optado por trabajar con el CI LM555, un temporizador extensamente utilizado que además nos permite controlarlo directamente desde el PIC, pudiendo generar a nuestro antojo la codificación sobre la señal a enviar.

4.2 El circuito integrado LM555

EL CI LM555 es un temporizador de alta estabilidad capaz de producir retrasos y oscilaciones. Ver la figura 4.1 página 60. En el modo de funcionamiento de temporizador o monoestable, el tiempo es controlado por una resistencia y un condensador externo. Para un funcionamiento como oscilador, la frecuencia y el ciclo de trabajo son determinados por dos resistencias externas y una capacidad. Existen varias versiones en el mercado de este temporizador. EL CI LM555 puede alcanzar una corriente máxima de salida de $200mA$ y admite una alimentación máxima de $18V$.

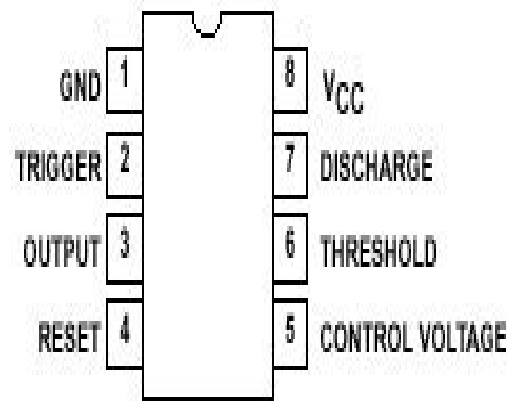


Figura 4.1: Esquema de pines del CI LM555.

En la figura 4.2 pagina 61 podemos ver un esquema básico del interior de un circuito integrado LM555. Como parte básica, apreciamos una red de tres resistencias iguales que fijan en sus puntos intermedios los valores umbrales, de $2/3V_{cc}$ y $1/3V_{cc}$, para dos amplificadores operacionales que están funcionando como comparadores. Las salidas de estos atacan a las patillas de RESET y SET de un flip-flop del que se aprovecha su salida complementada. De ahí ya obtenemos la salida por medio de un inversor, y el ataque a un tran-

sistor en colector abierto que jugará el papel del camino de descarga para el condensador.

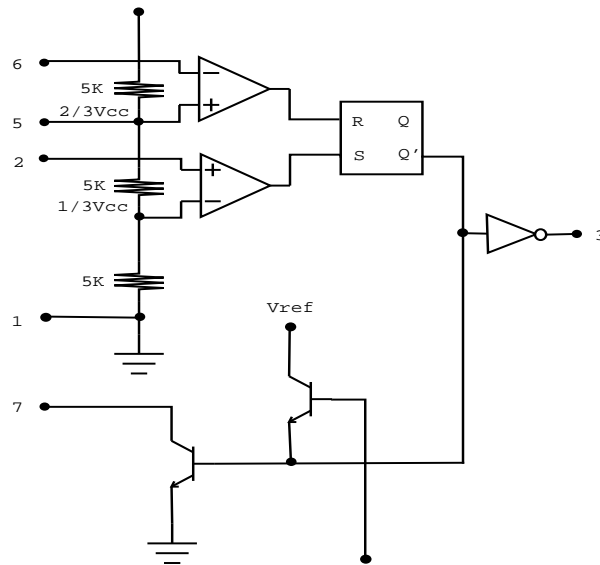


Figura 4.2: Esquema interno de un CI LM555.

Para entender el funcionamiento del CI LM555 podemos establecer unas reglas fundamentales. Si la tensión en la patilla 2 desciende por debajo de $1/3V_{cc}$, la S se pone a “1” (V_{cc}), luego la Q se pondrá a “1”, la Q’ a “0” (0V), entonces en la patilla 3 se pondrá a “1” y el transistor de la patilla 7 a “1”. Al ascender la tensión en la patilla 6 por encima de $2/3V_{cc}$, R se pone a “1”, Q a “0”, Q’ a “1”, entonces la patilla 3 vale “0”, y el transistor de la patilla 7 entra en saturación. Si la tensión en la patilla 4 desciende por debajo de 1V, la salida en la patilla 3 se pone a “0” y el transistor de la patilla 7 pasa a saturación.

4.3 El diseño final

Para nuestro sistema lo que necesitamos es un oscilador que genere la señal de 36KHz deseada, esto lo conseguimos haciendo funcionar al circuito integrado LM555 como *multivibrador astable*. En la figura 4.3 pagina 62 podemos ver el esquema básico de un LM555 trabajando como multivibrador astable.

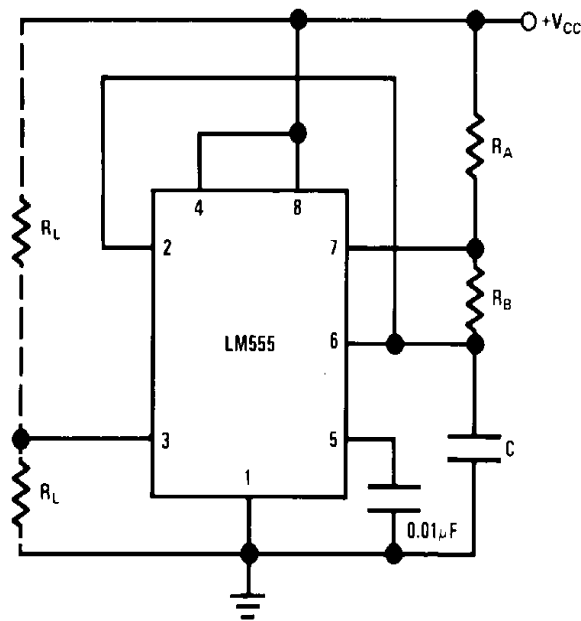


Figura 4.3: Esquema básico de CI LM555 como multivibrador astable.

En el momento de conectar la alimentación, el condensador C está a “0” (descargado), por tanto la patilla 2 está a “0” con lo que la patilla 3 está a V_{cc} y el transistor de la patilla 7 en corte (no conduce). Entonces el condensador se carga a través de dos resistencias y cuando la tensión en la patilla 6 alcanza $2/3V_{cc}$, la patilla 3 pasa a “0” y el transistor a saturación, descargando el condensador a través de la resistencia que tiene conectada, y del transistor. En el momento en que la patilla 2 descienda por debajo de $1/3V_{cc}$, la patilla 3

pasa a V_{cc} y el transistor a corte, empezando el otro estado inestable. La frecuencia de la señal cuadrada obtenida viene dada por:

$$f = \frac{1.44}{(R_A + 2R_B)C}$$

Falta señalar que dado que la capacidad externa se carga a través de $R_A + R_B$ y se descarga a través de R_B , el ciclo de trabajo puede precisarse en proporción a la relación entre estas dos resistencias.

Pero esto no es suficiente para lo que necesitamos puesto que debemos poder controlar el oscilador desde el microcontrolador. Bien, esto lo conseguimos conectando un pin de salida del PIC 16F84A directamente al pin 5 del CI LM555. De esta manera, cuando la patilla 5 esté puesta en alta por medio del microcontrolador, el CI LM555 trabajará como multivibrador astable aunque los valores umbrales para los comparadores ya no serán $2/3V_{cc}$ ni $1/3V_{cc}$. Y con la patilla 5 puesta en baja por el microcontrolador, el condensador C permanece descargado continuamente y con salida en la patilla 3 a "0". Ver figura 4.4 página 64, así ya podemos controlar la generación de una señal con ausencia y presencia de señal de $36KHz$ a nuestro antojo, en función de la palabra binaria a transmitir, lo que da lugar a una codificación FSK.

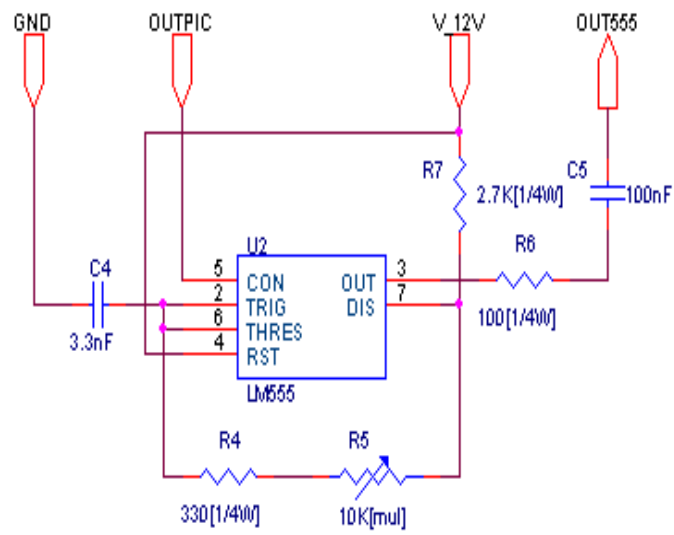


Figura 4.4: *Esquema del emisor de señal.*

Capítulo 5

El detector de señal

5.1 Introducción

El receptor es el módulo que se encarga de recibir la señal que se transmite por la red de distribución eléctrica proveniente de los diferentes dispositivos del sistema. Así establecemos la conexión entre la placa central y una placa sensora/actuadora utilizando como enlace físico el cableado de la red eléctrica. Sabemos que el módulo emisor de nuestro sistema emite a una frecuencia de $36KHz$ y que además tenemos la señal de $50Hz$ de red, por tanto debemos, de alguna manera, discernir entre las dos frecuencias y entre la existencia o ausencia de la señal de $36KHz$.

Esto lo conseguimos con un detector de tono por medio del CI LM567 que nos proporciona una salida en baja, exclusivamente, ante la presencia a su entrada de la señal de $36KHz$ de nuestro interés.

5.2 El circuito integrado NE567

El decodificador de tono CI NE567 es un lazo enganchado en fase PLL (Phase Locked Loop) de gran estabilidad. Su principal función es conducir una carga en cualquier momento en el que una frecuencia

concreta aparezca a su entrada. El ancho de banda de la frecuencia central y el retraso de salida son ajustables independientemente por cuatro componentes externos lo que le confiere una gran versatilidad.

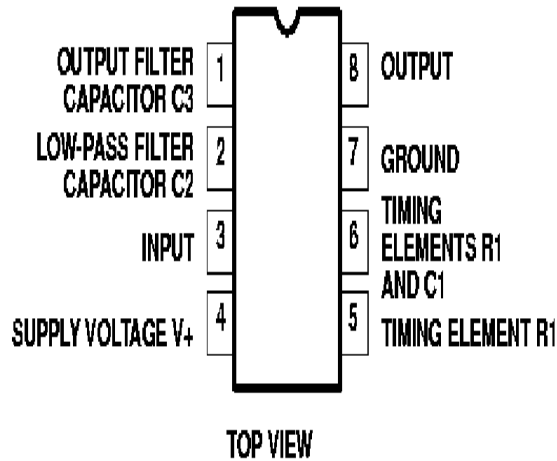


Figura 5.1: Esquema de pines del CI NE567.

Para nuestra aplicación usamos el circuito integrado NE567 como detector de tono, dado que necesitamos detectar la ausencia o presencia de la señal de 36KHz que hemos introducido en la red por medio del CI LM555. De este modo conseguimos decodificar nuestra señal FSK y reproducir la información recibida. Dado que, como hemos dicho, trabajamos a una frecuencia de 36KHz nos interesa entonces que la frecuencia central f_0 esté en los 36KHz . Por otro lado el ancho de banda de detección, BW, que se corresponde con el rango de captura, nos interesa que sea pequeño evitando la entrada de señales espúreas. Este ancho de banda, centrado en f_0 , indica el rango de frecuencias dentro del cual una señal por encima de un umbral de 20mV_{rms} produce un cero lógico a la salida del CI NE567.

5.3 El diseño final

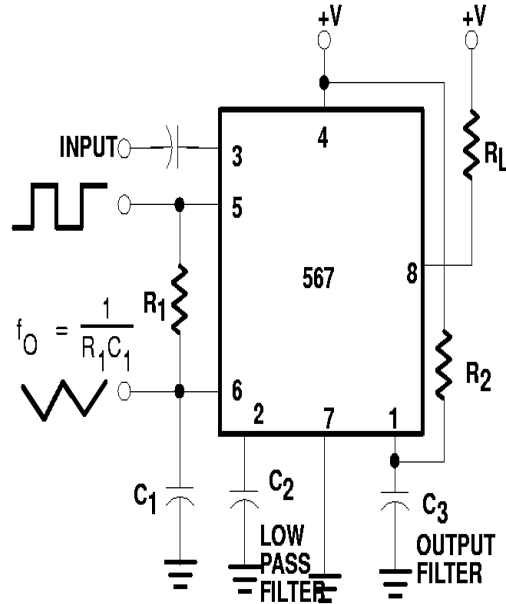


Figura 5.2: Esquema básico de un CI LM567.

En la figura 5.2 página 67 podemos ver una conexión típica de un circuito integrado NE567. Con tres pasos sencillos podemos diseñar nuestro detector de tono para los requerimientos básicos que nuestro sistema necesita.

Primeramente seleccionamos R_1 y C_1 para fijar nuestra frecuencia central. Para una mayor estabilidad en temperatura, R_1 debe tener un valor entre $2K\Omega$ y $20K\Omega$, y el coeficiente de temperatura del producto R_1C_1 debe ser lo suficientemente estable en el rango de temperaturas en que vamos a trabajar. Sabemos que la frecuencia central f_0 viene dada por:

$$f_0 = \frac{1}{1.1R_1C_1}]$$

entonces como nos interesa $f_0 = 36KHz$ y si tomamos por ejemplo $C_1 = 4nF$, obtenemos $R_1 = 6.3K\Omega$. Por supuesto no vamos a

poder fijar el valor de R_1 de una forma exacta por esto lo que hacemos es poner en serie una resistencia de $4.7K\Omega$ y un potenciómetro de forma que podemos ajustarnos a la resistencia requerida y con ello a la frecuencia central que andamos buscando.

Seguidamente debemos fijar el valor de C_2 para determinar así el ancho de banda de detección, BW. El cálculo lo realizamos por medio de la gráfica 5.3 pagina 68, donde observamos la amplitud de la señal de entrada versus ancho de banda, parametrizada por el producto $f_0 C_2$.

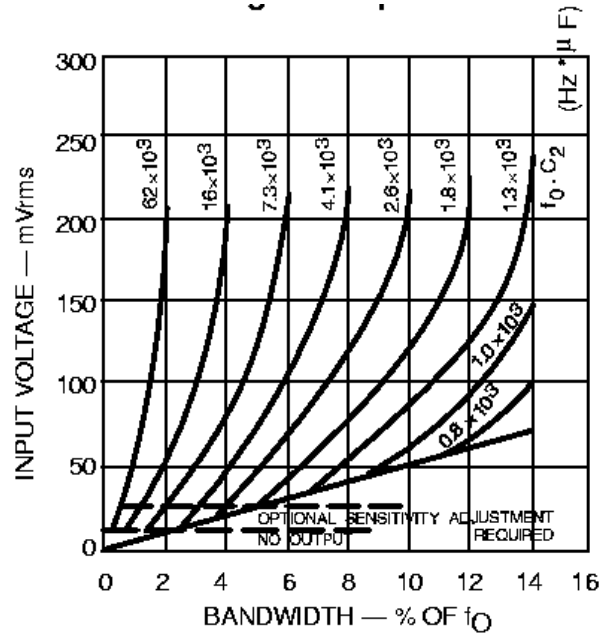


Figura 5.3: Amplitud de la señal de entrada versus ancho de banda.

Observamos como para poder obtener un ancho de banda constante necesitamos una amplitud de entrada por encima de los $200mV_{rms}$, y como entonces el ancho de banda queda fijado por el producto $f_0 C_2$. Para nuestro sistema el ancho de banda dependerá de la distancia que exista entre nuestros equipos puesto que la amplitud de entrada disminuye con la distancia.

Como último paso está la elección de C_3 , que determina la ban-

da de actuación en frecuencia de un filtro pasa baja que atenúa las frecuencias fuera de la banda de detección para eliminar salidas espúreas. Si C_3 es demasiado pequeño, las frecuencias cercanas al ancho de banda de detección producirán transiciones erróneas en la salida del CI NE567. Si C_3 es demasiado alto, aumentará el retraso entre la detección de una frecuencia y su repercusión en la salida del circuito integrado porque esta no se producirá hasta que el voltaje en C_3 pase el voltaje umbral. Un valor mínimo para C_3 es, por ejemplo, dos veces C_2 , nosotros hemos tomado cuatro veces C_2 .

Como comentario final , decir que en nuestro diseño final hemos añadido a la salida una etapa inversora con el fin de obtener una salida controlada y activa a nivel alto cuando se presenta en la entrada la frecuencia de $36KHz$, y en nivel bajo ante la ausencia de dicha frecuencia. Ver la figura 5.4 página 70.

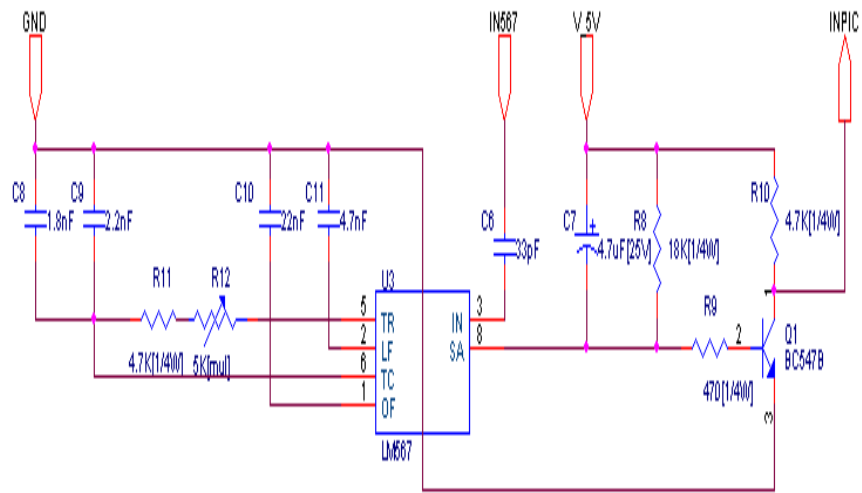


Figura 5.4: *Esquema del detector de tono.*

Capítulo 6

El controlador del sistema

6.1 Introducción

El sistema de forma global consiste en la comunicación entre dispositivos electrónicos, tanto entre un ordenador personal PC y la placa central como entre la placa central y la placa sensora/actuadora. Vemos por tanto que tenemos dos tipos de comunicaciones conectadas por medios físicos bastante diferentes, con lo que el hardware que necesitamos también será diferente. Pero estas diferencias son a nivel de la interfaz. Veamos, si nos fijamos en la placa central veremos que el conjunto del módulo del CI LM555 y el módulo del CI NE567 forman la interfaz de conexión con la red eléctrica que constituye la primera de las interfaces. Los dos optoacopladores junto con el CI 741 constituyen la interfaz de conexión con el puerto serie del PC formando la segunda de las interfaces. Por tanto podríamos resumir esquemáticamente la placa central en un módulo de alimentación siempre necesario, un módulo de interfaz con la red eléctrica y un módulo de interfaz con el PC. Si ahora nos fijamos en una de las otras dos placas, tanto la placa sensora como en la placa actuadora, nos encontramos de nuevo con el módulo de alimentación, también tenemos el módulo de interfaz con la red eléctrica y añadimos el módulo correspondiente al sensor o al actuador.

De todo esto se deduce que en cada placa falta el corazón del sistema, un circuito integrado que controle esos dos módulos básicos que tiene cada placa y desarrolle la funcionalidad asignada a cada uno de los dispositivos. Nosotros hemos escogido el microcontrolador PIC 16F84A que nos ofrece las prestaciones que necesitamos en un reducido espacio.

6.2 El microcontrolador PIC 16F84A

EL PIC 16F84A es un microcontrolador de 8 bit con 18 pines como podemos ver en la figura 6.1 página 73. Dispone de 13 pines de entrada/salida dispuestos en dos puertos, uno de 8 pines RB0-RB7 y otro de 5 pines RA0-RA4. Además uno de los pines de entrada/salida RB0 es a su vez una entrada de interrupción externa al PIC, INT, y otro pin de entrada salida RA4 se puede configurar como reloj de entrada a un temporizador interno al integrado TOCKI. Otros dos pines OSC1-OSC2 se usan para conectar un reloj externo al PIC, que se pueden configurar en modo RC para introducir la señal de reloj por el pin CLKIN y obtener por el pin CLKOUT otra señal de la cuarta parte de la frecuencia. Además dispone de un pin de reset hardware /MCLR y dos pines de alimentación: VDD masa y VSS a una tensión de 5V.

Este microcontrolador dispone de las siguientes características:

- Un juego de 35 instrucciones para programación.
- Una velocidad de operación de hasta $4MHz$.
- 1024 palabras de memoria de programa.
- 68 bytes de memoria de datos RAM.
- 64 bytes de memoria.
- Palabras de instrucción de 14 bits.

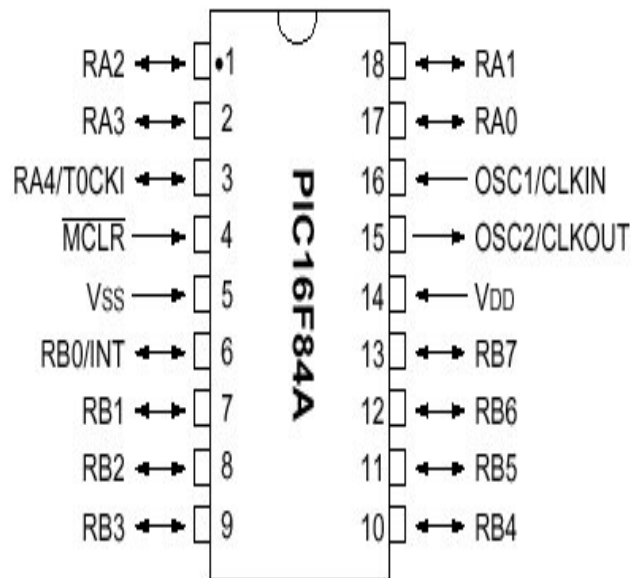


Figura 6.1: *Diagrama de pines del PIC 16F84A.*

- Anchura del bus de datos de 8 bits.
- 15 registros hardware con funciones especiales.
- Pila hardware de 8 niveles.
- Modos de direccionamiento directos, indirectos y relativos.
- Cuatro fuentes de interrupción:
 - Pin externo RB0/INT.
 - Final de cuenta del temporizador TMR0.
 - Cambio en los pines RB7-RB4.
 - Escritura de datos en la EEPROM completada.
- Temporizador de 8 bits con prescaler de 8 bits.
- Watchdog timer.
- Protección de código.

- Soporta 1000 ciclos de lectura/escritura en memoria de programa Flash.

6.3 El diseño en la placa central

Una parte común a todos nuestros diseños es el uso de un oscilador de cristal de $4MHz$ junto con dos condensadores de $33pF$ a tierra. Dado que una instrucción se ejecuta en cuatro ciclos de reloj, tenemos un tiempo de $1\mu seg$ por instrucción ejecutada (excepto aquellas que suponen saltos en el contador del programa que duran el tiempo de dos instrucciones).

Otra parte siempre común es la deshabilitación del reset hardware del PIC, de esta forma conectamos una resistencia de $15K\Omega$ desde alimentación al pin /MCLR dejando este pin siempre en alta y por tanto deshabilitado.

Además como algo específico de la placa central conectamos en el pin RB3 la línea que irá después de pasar por el optoacoplador al DB9.Rx, y conectamos el pin RB4 con el optoacoplador que va a la línea DB9.Tx. Así, funcionando los optoacopladores de interfaz transparente, establecemos comunicación entre el PIC y el puerto serie del ordenador. Ahora bien este microcontrolador no dispone de puerto serie hardware por tanto hemos desarrollado un puerto serie emulado por software que interacciona con el puerto COM del PC por medio de estas dos líneas y una más de retorno de señal.

El PIC también necesita comunicarse con las demás placas por medio de la red eléctrica por tanto utiliza otros dos pines para comunicarse con la interfaz a la red. Así, el pin RB7 es un pin de salida que actúa sobre el circuito integrado LM555 habilitando y deshabilitando la emisión de señal, y el pin RB0/INT funcionando como entrada de interrupción sirve para recibir los datos enviados por el circuito integrado NE567 al detectar la señal de $36KHz$ en

la red.

En la figura 6.2 página 75 podemos ver el diseño final del módulo del PIC con todos sus componentes.

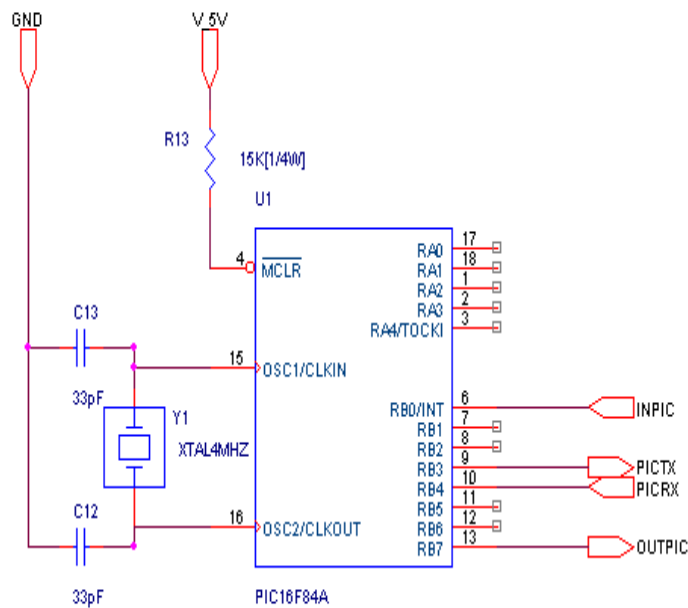


Figura 6.2: Módulo controlador de la placa central.

6.4 El diseño en la placa sensora

Lleva la parte común a todos nuestros diseños del oscilador de cristal de $4MHz$ junto con dos condensadores de $33pF$ a tierra. Dado que una instrucción se ejecuta en cuatro ciclos de reloj, tenemos un tiempo de $1\mu seg$ por instrucción ejecutada (excepto aquellas que suponen saltos en el contador del programa que duran el tiempo de dos instrucciones). Otra parte es la deshabilitación del reset

hardware del PIC dejando este pin siempre desactivado.

En la placa sensora el PIC también necesita comunicarse con las demás placas por medio de la red eléctrica por tanto utiliza otro dos pines para comunicarse con la interfaz a la red. Así, el pin RB7 es un pin de salida que actúa sobre el circuito integrado LM555 habilitando y deshabilitando la emisión de señal, y el pin RB0/INT funcionando como entrada de interrupción sirve para recibir los datos enviados por el circuito integrado NE567 al detectar la señal de $36KHz$ en la red.

En esta placa, tenemos como novedad el módulo del sensor de temperatura donde el PIC hace la labor de comparador y de realimentación del conversor analógico digital implementado: el conversor sigma-delta. Así el pin RA0 funciona de realimentación y el pin RA1 funciona de comparador del conversor.

En la figura 6.3 página 77 podemos ver el diseño final del módulo del PIC con todos sus componentes.

6.5 El diseño en la placa actuadora

Lleva la parte común a todos nuestros diseños del oscilador de cristal de $4MHz$ junto con dos condensadores de $33pF$ a tierra. Dado que una instrucción se ejecuta en cuatro ciclos de reloj, tenemos un tiempo de $1\mu seg$ por instrucción ejecutada (excepto aquellas que suponen saltos en el contador del programa que duran el tiempo de dos instrucciones). Otra parte es la deshabilitación del reset hardware del PIC dejando este pin siempre desactivado.

En la placa actuadora el PIC también necesita comunicarse con las demás placas por medio de la red eléctrica por tanto utiliza otro dos pines para comunicarse con la interfaz a la red. Así, el pin RB7 es un pin de salida que actúa sobre el circuito integrado LM555 habilitando y deshabilitando la emisión de señal, y el

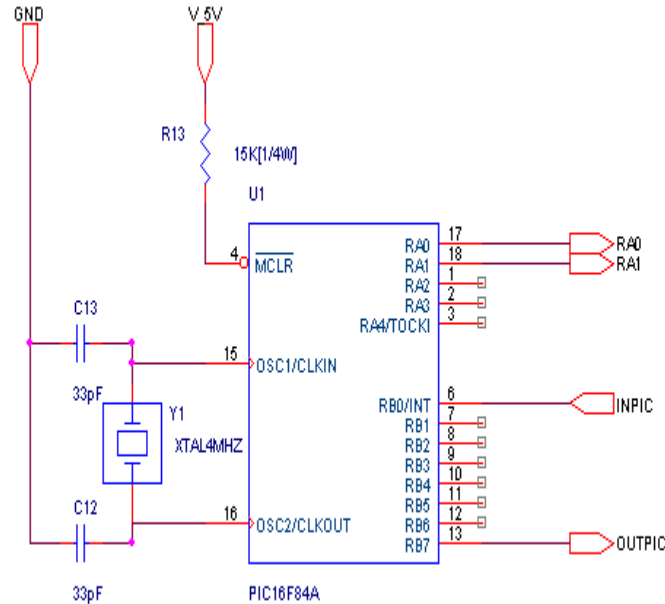


Figura 6.3: Módulo controlador de la placa sensora.

pin RB0/INT funcionando como entrada de interrupción sirve para recibir los datos enviados por el circuito integrado NE567 al detectar la señal de $36KHz$ en la red.

En esta placa, tenemos como novedad el módulo del actuador de luz donde el PIC 16F84A envía al PIC 12C508A una palabra indicando el tiempo que éste debe tardar en disparar el triac desde que detecta el último cruce por cero. Así es el PIC 16F84A el que controla la potencia que se le pasa al enchufe. Por tanto conectamos los pines RB0-RB3 del PIC 16F84A a los pines GP0-GP3 para intercambiar la palabra.

En la figura 6.4 página 78 podemos ver el diseño final del módulo del pic con todos sus componentes.

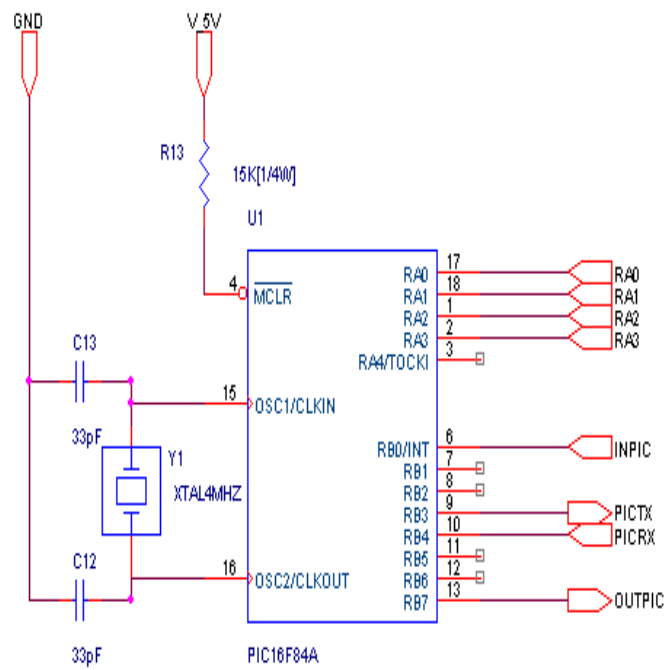


Figura 6.4: Módulo controlador de la placa actuadora.

Capítulo 7

La interfaz con el PC

7.1 Introducción

Nuestro sistema se basa en dos placas , una sensora y otra actuadora, que hacen de periféricos y otra placa, la central, que controla todo el sistema. Ahora bien, debemos añadir un terminal manejable por el usuario para interaccionar con el sistema, concretamente con la placa central que es la que maneja toda la información. Nosotros hemos utilizado como terminal un ordenador personal, PC, existente en la mayoría de las casas, donde nuestro sistema es susceptible de usarse.

Por tanto, necesitábamos diseñar una interfaz de conexión entre la placa central y el PC. Tomamos como punto de partida el puerto serie del PC para el intercambio de información con nuestro microcontrolador de la placa central.

7.2 La comunicación

La comunicación entre nuestro microcontrolador PIC 16F84A y el puerto serie del PC se realiza mediante un protocolo a tres hilos: Rx (pin 2 del DB9), Tx (pin 3 del DB9) y GND (pin 5 del DB9).

La línea Rx es la línea de recepción de datos del puerto serie, y por tanto es la línea de transmisión de datos para el microcontrolador. La línea Tx, al contrario, es la línea de transmisión de datos del puerto serie y por tanto es la línea de recepción de datos para el microcontrolador. En cuanto a la línea GND es la línea de masa común para el puerto serie y para el PIC.

Pero a la hora de interconectar por medio de estas tres líneas el PIC y el puerto serie, surgen dos problemas importantes: uno es el aislamiento eléctrico entre el circuito y la red eléctrica, y otro problema es la adaptación entre la lógica TTL del microcontrolador y la lógica RS232 del puerto serie.

7.3 Aislamiento eléctrico: el optoacoplador CI 4N35

Ya hemos comentado en un capítulo anterior, hablando de la fuente de alimentación de nuestros equipos, que estas fuentes carecen de transformadores, por tanto carecemos de aislamiento eléctrico con la red de distribución eléctrica. Esto no supone un problema grave mientras trabajemos únicamente con nuestra placa y sin conectar a ella ningún otro tipo de dispositivo que también se alimente de la red; pero es aquí donde aparecen los problemas dado que vamos a conectar a la placa, el puerto serie de un PC que a su vez se alimenta de la red eléctrica. Así, sin aislamiento eléctrico estamos cerrando un lazo de masa y podemos provocar un cortocircuito.

Para evitarlo nos hemos decantado por usar optoacopladores que nos proporcionan ese aislamiento eléctrico salvando el mayor tamaño de los transformadores. Usamos entonces dos optoacopladores CI 4N35, uno para la línea Rx y otro para la línea Tx. La idea en todo momento es tener el puerto serie a un lado, en el otro extremo el PIC y en el medio los dos optoacopladores aislando ambos circuitos.

El optoacoplador CI 4N35 En la figura 7.1 pagina 81 podemos ver el pillaje básico del optoacoplador CI 4N35 y en la figura 7.2 pagina 82 vemos el esquema interno del optoacoplador. El optoacoplador CI 4N35 está compuesto por un diodo infrarrojo de Arseniuro de Galio que hace de fuente, y que está ópticamente acoplado a un fototransistor NPN de Silicio. Proporciona un aislamiento eléctrico de alto voltaje que soporta hasta $3,5KV$. Lo que nos interesa es ese acoplamiento optico entre el diodo fuente y el transistor que rompe la continuidad eléctrica.

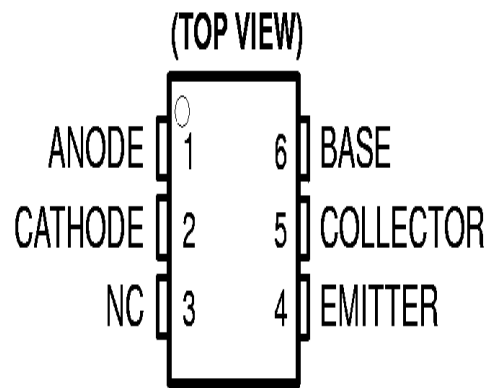


Figura 7.1: Esquema de pines del CI 4N35.

7.4 Lógica RS232 versus lógica TTL

La lógica TTL es la lógica digital con la que trabaja el microcontrolador PIC 16F84A: el uno lógico se representa por un valor de tensión entre 2 y 5V, y representa un cero lógico por un valor de tensión entre 0 y 0,8V, esto como entrada. Como salida la lógica TTL representa un uno lógico por un valor de tensión entre 2,4 y 5V y representa un cero lógico por un valor de tensión entre 0 y 0,4V. Por otro lado tenemos la lógica RS232, esta es la lógica digital con

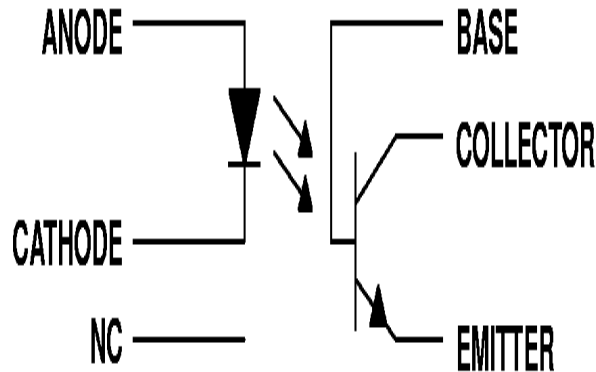


Figura 7.2: Esquema interno del optoacoplador CI 4N35.

la que trabaja el puerto serie de un PC que cambia frente a la lógica TTL. La lógica RS232 representa un uno lógico por un valor de tensión entre -3 y $-25V$, y representa un cero lógico por un valor de tensión entre 3 y $25V$, esto para recepción. Para transmisión la lógica RS232 representa un uno lógico por un valor de tensión entre -5 y $-15V$ y representa un cero lógico por un valor de tensión entre 5 y $15V$.

Por tanto necesitamos hacer una conversión de tensiones entre lógicas. Con este objetivo existen unos circuitos integrados ya especializados, los CI MAX232, que realizan directamente la conversión. Pero nosotros hemos intentado aprovechar los optoacopladores para no necesitar usar un CI MAX232. Así, para transmitir del puerto serie al PIC, conseguimos adaptar los niveles de tensión sin más que usar un optoacoplador y aprovechar que el transistor nos hace la función de inversor. Para recibir desde el PIC hacia el puerto serie tenemos que amplificar en tensión (en módulo), y esta subida la realizamos ayudados de un amplificador operacional CI LM741 en configuración de comparador. Ver la figura 7.3 página 83. Usamos dos patillas extras del DB9, la patilla DB9.7 para obtener

10V y la patilla DB9.4 para obtener $-10V$, esto lo conseguimos por medio del software *Dome*, instalado en el PC, que coloca los valores lógicos necesarios en el puerto para que dichas patillas tengan los valores de tensión anteriores (que son los valores de tensión que dá generalmente el puerto serie de un PC). Así alimentamos al amplificador operacional con $\pm 10V$ y dada la ausencia de realimentación, la salida del CI LM741 satura en valores de tensión cercanos a la alimentación y que son compatibles con la lógica RS232 del puerto serie.

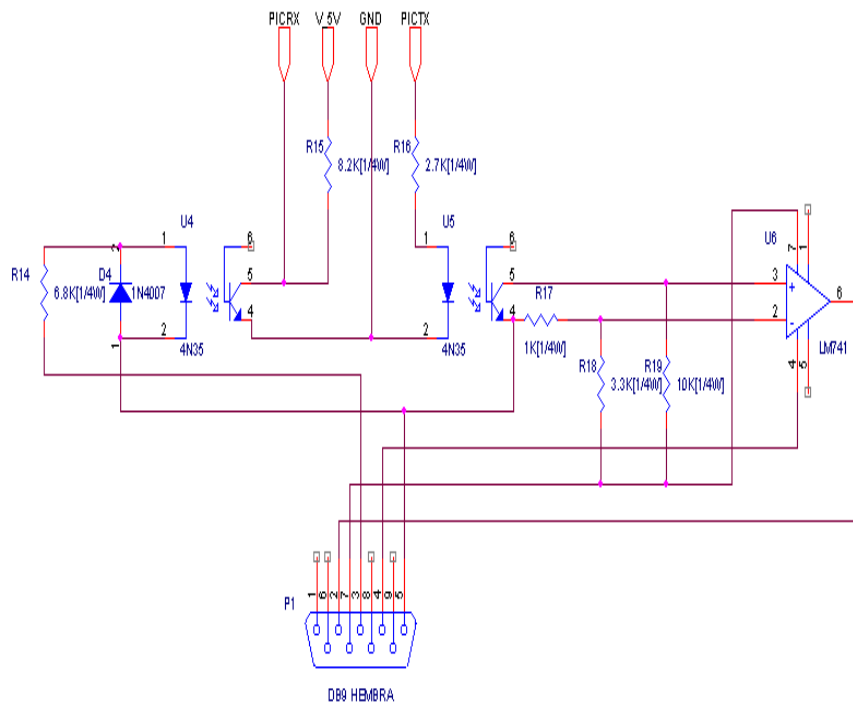


Figura 7.3: *Esquema de la interfaz con el PC.*

7.5 El Diseño final

Por tanto, hemos conseguido una interfaz entre el PC y nuestra placa central, aislada eléctricamente. Como ya hemos comentado anteriormente, se han usando dos optoacopladores CI 4N35 y un amplificador CI LM741 a cambio de olvidarnos del típico CI MAX232. Al final esta interfaz es transparente al PC y al PIC, para ellos es como si fuese una comunicación directa, siempre bajo el protocolo del puerto serie que hemos programado en el PIC de la placa central, dado que el PIC 16F84A carece de una UART que nos hubiese facilitado las cosas.

Capítulo 8

El sensor de temperatura

8.1 Introducción

El desarrollo de un sensor de temperatura es algo casi imprescindible en cualquier aplicación de control. Ya podemos estar hablando de una industria, una casa o un coche, para todo se necesita controlar la temperatura por medio de sensores. Esto hace necesario que diseñemos, como primer paso de aplicación para nuestro sistema, un sensor de temperatura.

Hay sensores de temperatura de todo tipo, pero todos buscan una cosa común: conseguir una magnitud de salida proporcional a la temperatura, y lo más lineal posible. Para nuestra aplicación hemos escogido el sensor LM35DZ que da una salida en tensión continua proporcional a la temperatura ambiente. Ahora bien, el siguiente paso es medir de alguna manera esa tensión con nuestro microcontrolador que carece de conversor analógico-digital. Por eso hemos implementado un conversor sigma-delta de un solo bit que nos permite obtener el valor de la tensión continua usando simplemente tres resistencias, un condensador y tres pines del PIC 16F84A.

8.2 El sensor LM35DZ

En este apartado hemos decidido usar el circuito integrado LM35DZ de National Semiconductor como sensor de temperatura. La serie LM35 son circuitos integrados sensores de temperatura de precisión, cuyo voltaje de salida es linealmente proporcional a la temperatura Celsius (escala centígrada). Así la serie LM35 ofrece una ventaja sobre los sensores de temperatura lineales calibrados en Kelvin porque no se necesita restar una constante para obtener la temperatura en escala centígrada. El CI LM35 no requiere ninguna calibración externa y ofrece exactitudes de $\pm 1/4^{\circ}C$ a temperaturas cercanas a la temperatura ambiente, $\pm 3/4^{\circ}C$ para un rango de $-55^{\circ}C$ a $+150^{\circ}C$.

Su baja impedancia de salida, su salida lineal y calibración inherente lo hacen especialmente útil para su uso en dispositivos de control. Además consume solamente $60\mu A$ de forma que su calentamiento interno es muy bajo, menos de $0,1^{\circ}C$ en aire sin movimiento.

Su uso resulta realmente sencillo pudiendo adaptarlo a cualquier tipo de necesidades. Podemos observar en la figura 8.1 página 87 y en la figura 8.2 página 88 dos formas prácticas de usarlo para dos rangos diferentes de temperatura.

8.3 El CAD Sigma Delta

8.3.1 Cuantización

El procedimiento de conversión de una señal analógica a una señal discreta incluye varios pasos. Primero se realiza un paso de discretización del momento de muestreo. El muestreo convierte la variable independiente, por ejemplo tiempo, en valores discretos. En circuitos electrónicos se refiere a la funcionalidad del Sample&Hold

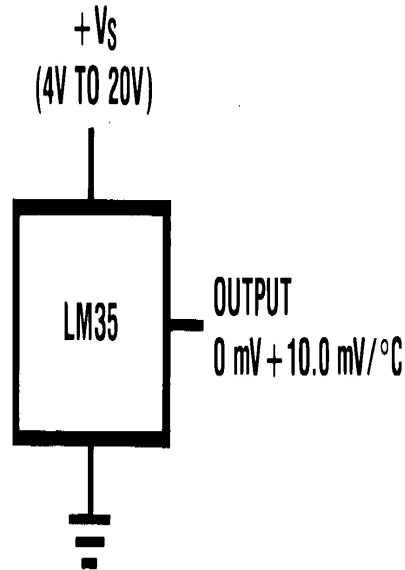
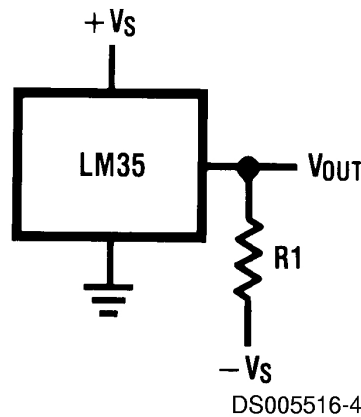


Figura 8.1: *Sensor de temperatura centígrado básico (+2°C a +150°C).*

(S&H), de tomar una muestra del voltaje y mantenerlo mientras el digitalizador trabaja sobre el valor.

Mientras el circuito de S&H mantiene en su salida el valor muestreado, la entrada al S&H puede variar sin afectar la salida. Así, el conversor A/D no ve la señal analógica directamente, solo ve un voltaje constante durante el período de mantenimiento del circuito S&H.

La conversión A/D convierte la variable dependiente en un valor discreto mediante la cuantización. El efecto de la cuantización puede significar que una señal inicial con un valor por ejemplo de 3,2938V es convertido en 3,294V por los dígitos de precisión del conversor A/D. Esto es el error de cuantización. En las figuras 8.6 página 92 y 8.7 página 93 se pueden notar leves diferencias entre la traza de la señal discretizada por el S&H y la señal digitalizada por el A/D. Esto es el error de cuantización.

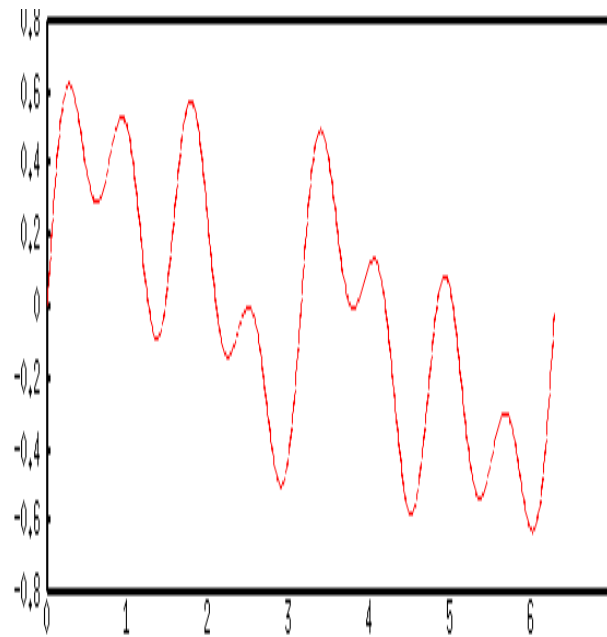


Choose $R_1 = -V_S/50 \mu A$
 $V_{OUT} = +1,500 \text{ mV at } +150^\circ\text{C}$
 $= +250 \text{ mV at } +25^\circ\text{C}$
 $= -550 \text{ mV at } -55^\circ\text{C}$

Figura 8.2: *Sensor de temperatura centígrado de amplio rango.*

El efecto de la cuantización puede calcularse sabiendo el número de bits del conversor A/D. Con un conversor de 8 bits, o sea 256 pasos, y un rango dinámico de 5V entre señal baja y señal alta, el cambio del bit menos significativo representa aproximadamente 0,01953V, digamos 0,02V. Entonces nuestro voltaje de 3,2938V va a resultar en un valor de 3,28V o 3,30V. Si el conversor fuera de 16 bits entonces tiene 65536 pasos y cada paso correspondería a 0,00007629V y reflejaría 3,2938V con cuatro dígitos de precisión.

Una muestra en el señal de salida puede tener un error de cuantización equivalente a $\pm 1/2LSB$. Se determina el error de cuantización (pero no de discretización de muestreo) restando la señal digitalizada de la señal muestreada. O desde otro punto de vista, la señal digital es igual a la señal muestreada más un error. Este error tiene el aspecto de ruido aleatorio distribuido uniformemente entre $\pm 1/2LSB$, con media cero y desviación estándar $\sim 0,29LSB$.

Figura 8.3: *Señal analógica.*

8.3.2 Conversión A/D paralelo (flash)

Una manera de convertir una señal analógica a un número digital utilizando N bits de precisión, es tener $2N$ voltajes de referencia. Se compara el voltaje de entrada con los voltajes de referencia simultáneamente. La salida es determinada inmediatamente. La ventaja de este método es que es muy rápido, el inconveniente es que requiere mucho circuito.

8.3.3 Conversión por ponderación

Otra manera de digitalizar un voltaje de entrada es comparar la entrada con un voltaje de referencia correspondiente al mayor bit de referencia. Por ejemplo, si el rango total de conversión es de 0 - 5V, entonces el voltaje del primer bit sería de 2,5V. Si la entrada es mayor que la referencia, el bit toma un "1", y la referencia es restada del voltaje de entrada, y así sucesivamente. Es parecido

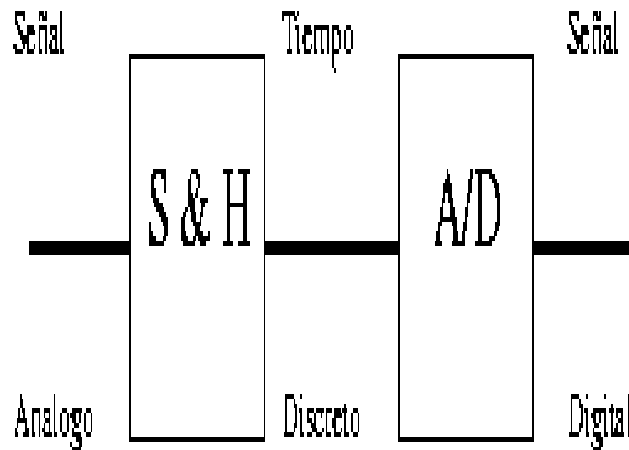


Figura 8.4: *Muestreador más cuantificador.*

a una búsqueda binaria en una tabla o archivo. El método es más lento que el primero pero el circuito es mucho más simple.

8.3.4 Conversión por conteo

La manera más simple de determinar el voltaje de entrada es contar cuantas veces hay que restar un voltaje pequeño a la entrada para anularla. El resultado del conteo es el valor digital del voltaje de entrada.

8.3.5 Conversión de un bit

Un esquema popular en telecomunicaciones y en reproducción de música de alta fidelidad es la conversión A/D y D/A de un solo bit. El concepto básico es de la *modulación delta*, donde un circuito tiene una entrada analógica, y una salida digital. A la entrada del

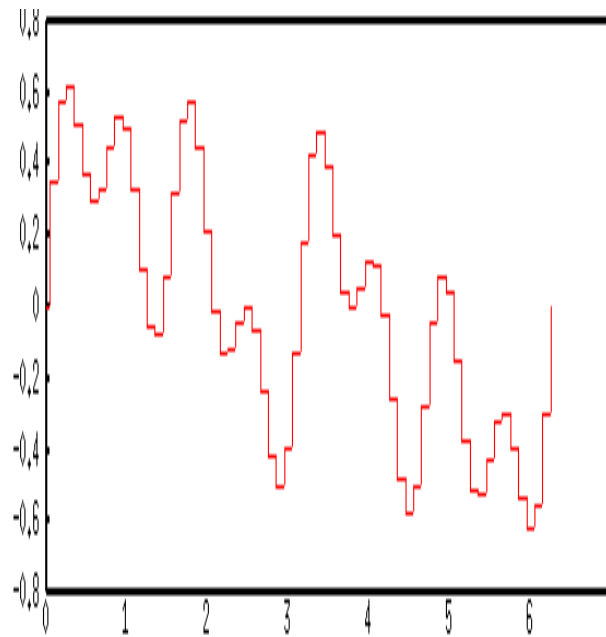
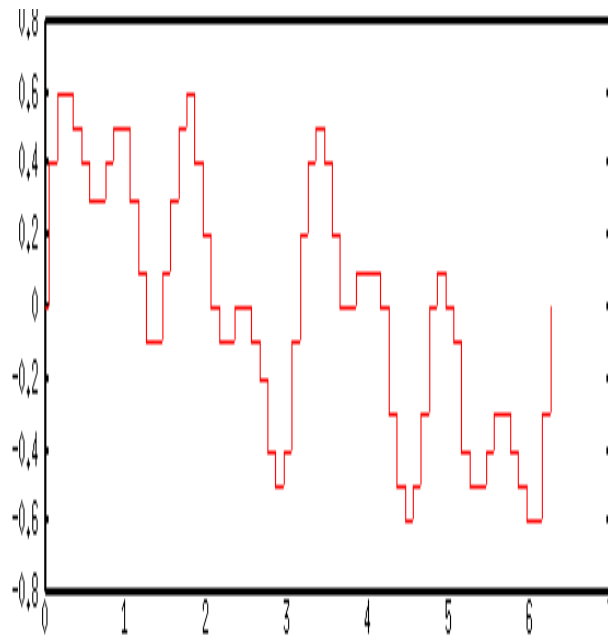


Figura 8.5: *Señal analógica muestreada.*

circuito un comparador determina si el voltaje entrante es mayor o menor o una referencia interna. Si es mayor, el circuito emite un “1” a su salida, y a la vez aumenta la referencia interna en un voltaje pequeño. Si fuera menor, se emite un “0” y se reduce la referencia interna por el valor precisado, por ejemplo, un $1mV$ por ciclo del reloj. Entonces la salida del circuito es una secuencia de “1’s” y “0’s” proporcional a la derivada del voltaje entrante.

Un circuito denominado *sigma-delta* utiliza modulación delta para generar una secuencia de dígitos proporcional a la derivada de la entrada. El circuito entonces muestrea la secuencia en grupos de bits, por ejemplo grupos de 1024 bits. Si todos los bits son unos, entonces la entrada corresponde al voltaje máximo permitido. Si todos son ceros, el voltaje corresponde al mínimo permitido. De ahí viene el nombre, modulación delta seguida por sumación (sigma).

Figura 8.6: *Señal analógica cuantificada.*

8.4 El diseño final

Existen en el mercado multitud de microcontroladores con convertidores analógicos/digitales en el interior del propio circuito integrado, si bien estos resultan más caros en precio y en tamaño del integrado. Así pues, y dada la existencia de los convertidores analógicos/digitales sigma-delta, nos inclinamos por elegir un microcontrolador sin convertidor analógico/digital. De esta manera hemos adaptado el diseño de un convertidor sigma-delta a su realización mediante un microcontrolador.

De la figura 8.8 página 94 observamos que necesitamos un integrador, un comparador y una realimentación. Todo esto resulta sencillo de realizar con, el microcontrolador, un poco de código y algún elemento pasivo. El integrador no es más que un filtro pasabaja que podemos realizar con una resistencia y un condensador, justo a la entrada de la tensión a medir, como comparador usamos un pin

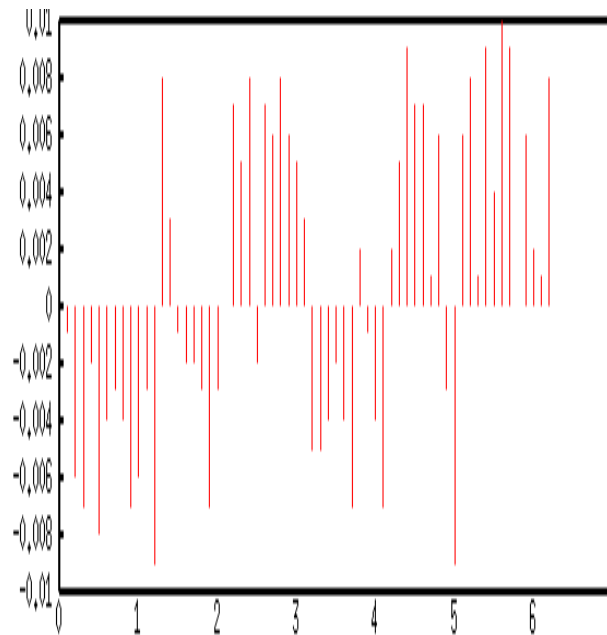


Figura 8.7: *Señal digital.*

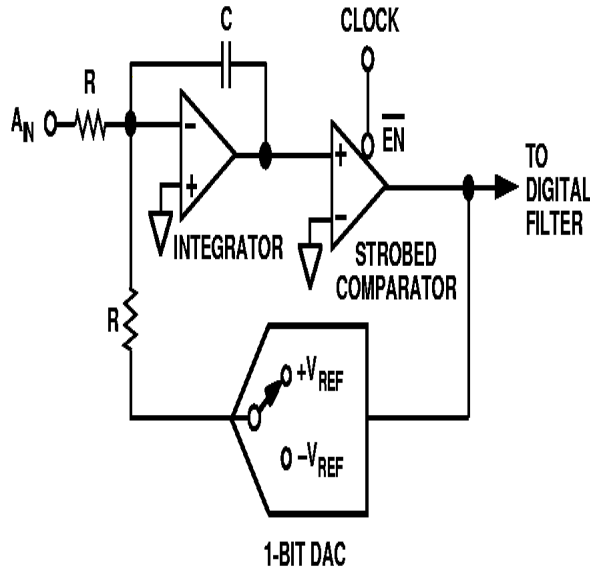
de entrada del microcontrolador sabiendo que compara con $\sim 1.3V$ (tensión crítica de distinción entre el uno lógico y el cero lógico), y como realimentación un pin de salida del microcontrolador que impondrá una tensión de $5V$ (uno lógico) o de $0V$ (cero lógico). Ver figura 8.9 página 95.

Para conectar los distintos bloques tenemos un pequeño código que ejecuta el microcontrolador y que se puede entender fácilmente con el siguiente pseudocódigo:

```

Asignar registro LowCounter
LowCounter = 0
Poner Out en baja hasta que In = alta
Poner Out en alta hasta que Out = baja
For N cuentas
    Poner Out complementario a In
    Si In = baja incrementar LowCounter

```

Figura 8.8: *Modulador de primer orden.*

Next N

De esta manera en nuestra aplicación en la que la tensión a convertir es la salida del sensor LM35 obtenemos directamente en la LowCounter el valor de la temperatura en grados centígrados, lista para mostrarla en cualquier aplicación sin ninguna cuenta añadida de ajuste de escala o de linealización. Si que existe un detalle, el sensor LM35 da una tensión proporcional a la temperatura con una precisión de $10mV/^{\circ}C$ y dado que en nuestro programa obtenemos el valor de la temperatura en una variable de un byte, es obvio que de $0^{\circ}C$ a $25,6^{\circ}C$ la variable pasa de valer 00h a valer FFh, de forma que dentro de ese rango si obtenemos directamente la temperatura en grados centígrados. Pero si nos vamos al siguiente rango de $25,6^{\circ}C$ a $51,2^{\circ}C$ la variable de nuevo pasa de valer de 00h a FFh ahora para obtener la temperatura real debemos sumar a la variable el valor de 256 que es lo que supone haber sobrepasado una vez el contador. De esta manera sacamos el valor de la temperatura en

Capítulo 9

El regulador de luz

9.1 Introducción

Un regulador no resulta algo desconocido, existen en cualquier tienda de accesorios electrónicos. Se colocan en el lugar de la vieja llave de luz y, normalmente, disponen de un dial que podemos girar obteniendo una intensidad de luz a nuestro antojo. Ahora bien, estos reguladores funcionan por medio de un potenciómetro, así según giramos el dial aumentamos o disminuimos la intensidad de una luz. Eso está muy bien, pero nosotros lo que estamos buscando es algo con otra filosofía distinta, no buscamos un regulador de luz *manual* sino un regulador de luz que podamos controlar desde un PC. En definitiva necesitamos un regulador de luz controlado a través de lógica digital y nada de manipular directamente en él. Con esta nueva filosofía hemos diseñado un regulador de luz controlado desde el microcontrolador PIC 16F84A dispuesto en la placa actuadora. La idea consiste en detectar el cruce por cero de la señal de red de $50Hz$, y así poder impedir el paso de corriente durante un tiempo concreto a intervalos regulares, siguiendo los semiciclos de dicha señal de red. Esto provoca una disminución de la potencia de señal eléctrica que dejamos pasar, si además añadimos que ese tiempo de ausencia de señal de red lo podemos variar a nuestro antojo des-

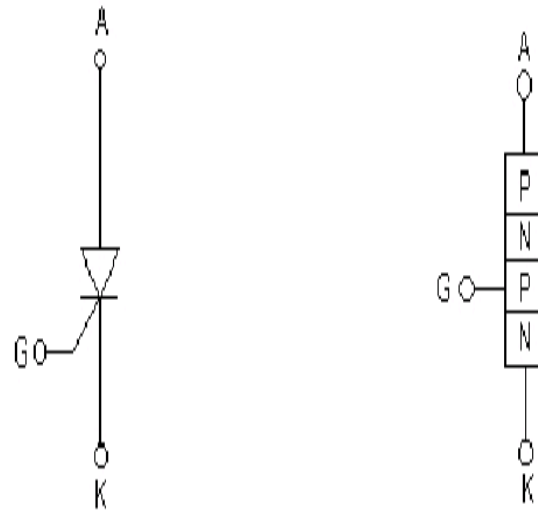
de el microcontrolador PIC 16F84A obtenemos un dispositivo que controla la potencia de señal que se deja pasar por un enchufe externo, tenemos por tanto un regualdor de luz controlado digitalmente. Además resulta sencillo enviar ese dato del tiempo de limitación desde el PC al pic central y desde allí al PIC 16F84A de la placa actuadora para adaptar la luz a los requerimientos que el usuario a pedido desde el PC.

Como parte imprescindible en nuestro diseño está el triac, un dispositivo que impide el paso de corriente por una línea hasta que es disparado externamente, a continuación vuelve a su estado de reposo una vez atravesado el cruce por cero. Controlando entonces el triac tenemos ya resuelto el regualdor de luz. Dada su importancia nos vamos a detener en su descripción de una manera más concisa.

9.2 Tiristores

Un tiristor es uno de los tipos más importantes de los dispositivos semiconductores de potencia. Los tiristores se utilizan en forma extensa en los circuitos electrónicos de potencia. Se operan como conmutadores biestables, pasando de un estado no conductor a un estado conductor. Para muchas aplicaciones se puede suponer que los tiristores son interruptores o conmutadores ideales, aunque los tiristores prácticos exhiben ciertas características y limitaciones.

Un tiristor es un dispositivo semiconductor de cuatro capas de estructura PNPN, con tres uniones PN y tres terminales: ánodo, cátodo y compuerta. La figura 9.1 página 99 muestra el símbolo del tiristor y una sección recta de tres uniones PN.

Figura 9.1: *Símbolo del tiristor.*

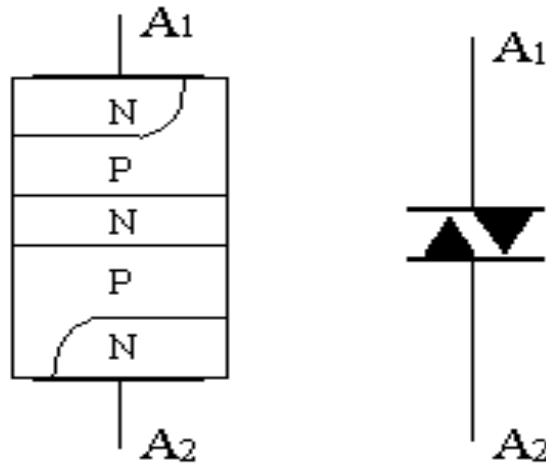
9.3 Diac

El Diac es un dispositivo semiconductor de dos terminales de estructura similar a la del transistor que presenta cierto tipo de conductividad biestable en ambos sentidos. Cuando las tensiones presentes en sus terminales son suficientemente altas se utiliza principalmente junto a los triacs para el control en fase de los circuitos. Es un tipo de tiristor que puede conducir en los dos sentidos. En la figura 9.2 pagina 100 se aprecia la construcción básica y el símbolo del Diac.

9.4 SCR

9.4.1 Silicon Controlled Rectifier

Los SCR de puerta sensible se caracterizan por su facilidad de disparo, gracias a su bajo nivel de corriente de puerta. Esta alta sen-

Figura 9.2: *Símbolo del diac.*

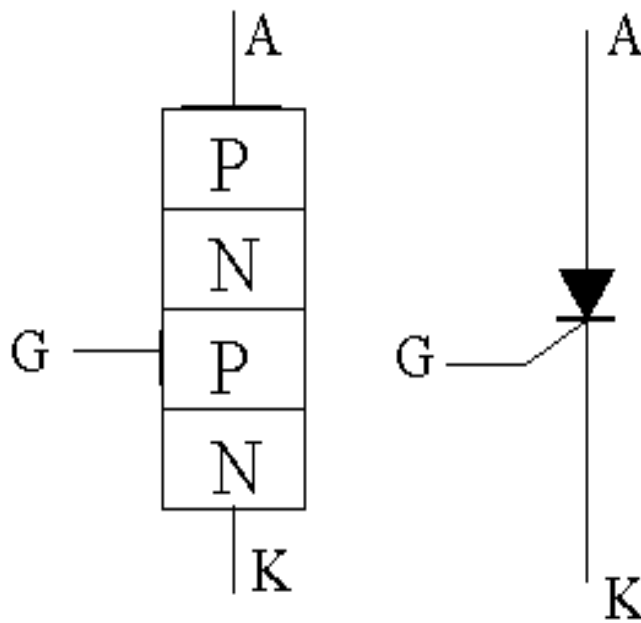
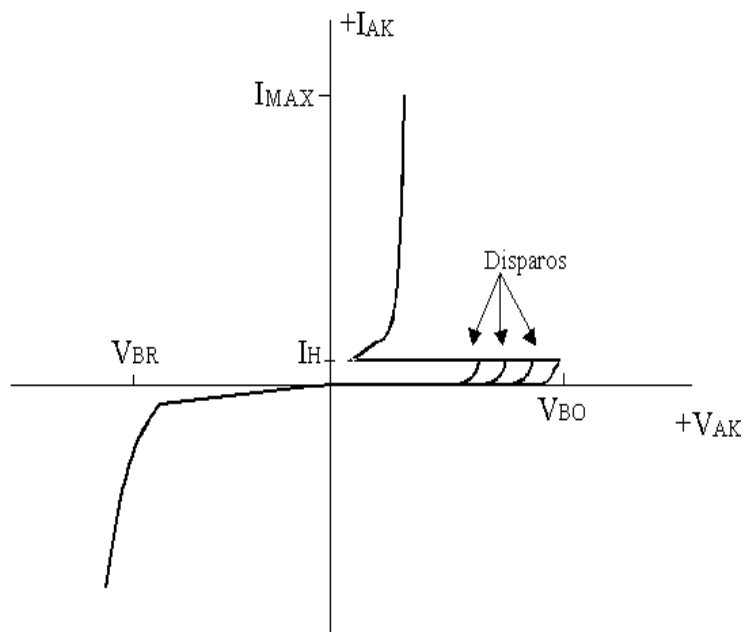
sibilidad facilita la posibilidad de disparo por medio de señales de baja intensidad provenientes de microcontroladores o circuitos integrados, simplificando el diseño de la circuitería de disparo y reduciendo el número de componentes y por consiguiente el coste del circuito. Son ideales para aplicaciones de baja potencia en las que se requiere un bajo consumo.

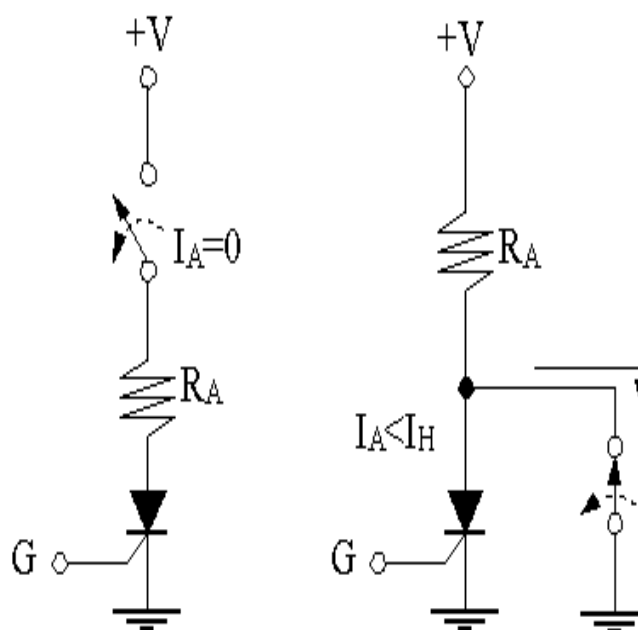
El SCR es un dispositivo de cuatro capas que posee tres terminales: ánodo, cátodo y puerta (gate). Presenta dos estados de operación: abierto y cerrado, como si se tratase de un interruptor. En la figura 9.3 página 101 podemos apreciar la construcción básica y el símbolo del SCR.

9.4.2 Característica V/I

Tal y como se aprecia en la figura 9.4 pagina 101, la parte de polarización inversa de la curva es análoga a la del diodo.

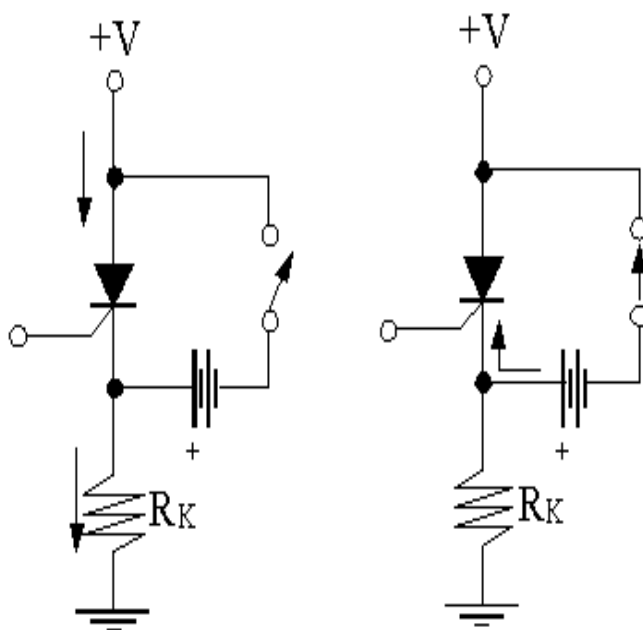
En cuanto a la parte de polarización positiva, el diodo no conduce hasta que se recibe un pulso de tensión en el terminal de puerta (gate). Una vez recibido, la tensión entre ánodo y cátodo cae hasta

Figura 9.3: *Símbolo del SCR.*Figura 9.4: *Gráfica de la característica V/I de un SCR.*

Figura 9.5: *Interrupción de corriente anódica.*

ser menor que un voltio y la corriente aumenta rápidamente, quedando limitada en la práctica por componentes externos. Podemos ver en la curva cuatro valores importantes. Dos de ellos provocarán la destrucción del SCR si se superan: V_{RB} e I_{MAX} . V_{RB} (Reverse Breakdown Voltage) es la tensión a partir de la cual se produce el fenómeno de avalancha. I_{MAX} es la corriente máxima que puede soportar el SCR sin sufrir daño. Los otros dos valores importantes son la tensión de cebado V_{BO} (Forward Breakover Voltage) y la corriente de mantenimiento I_H . Para que el dispositivo interrumpa la conducción de la corriente que circula a través del mismo, ésta debe disminuir por debajo del valor I_H (corriente de mantenimiento). Hay dos métodos básicos para provocar la apertura del dispositivo: interrupción de corriente anódica y conmutación forzada. Ambos métodos se presentan en las figuras 9.5 página 102 y 9.6 pagina 103.

En la figura 9.5 página 102 se observa cómo la corriente anódi-

Figura 9.6: *Conmutación forzada.*

ca puede ser cortada mediante un interruptor bien en serie (figura izquierda), o bien en paralelo (figura derecha). El interruptor en serie simplemente reduce la corriente a cero y hace que el SCR deje de conducir. El interruptor en paralelo desvía parte de la corriente del SCR, reduciéndola a un valor menor que I_H . En el método de conmutación forzada, que aparece en la figura 9.6 página 103, se introduce una corriente opuesta a la conducción en el SCR. Esto se realiza cerrando un interruptor que conecta una batería en paralelo al SRC.

Una aplicación muy frecuente de los SCR es el control de potencia en alterna en reguladores (dimmer) de lámparas, calentadores eléctricos y motores eléctricos. En la figura 9.7 página 104 se muestra un circuito de control de fase de media onda y resistencia variable. Entre los terminales se aplican $120V_{ac}$. R_L representa la resistencia de la carga (por ejemplo un elemento calefactor o el filamento de una lámpara). R_1 es una resistencia limitadora de la corriente y

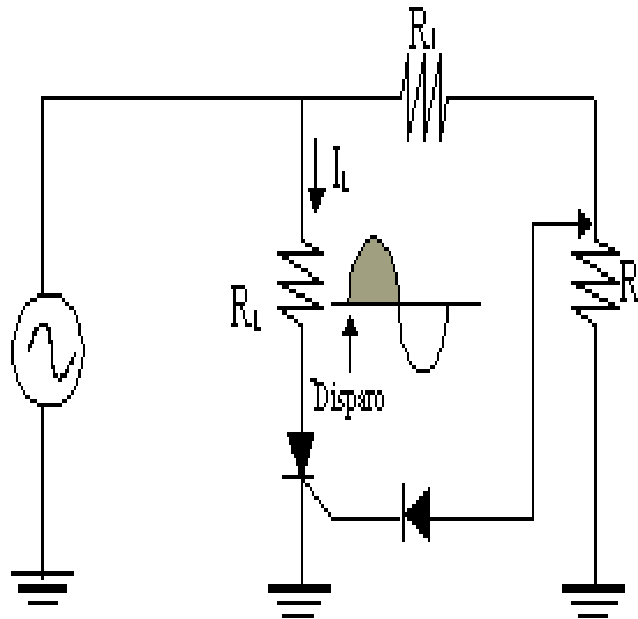


Figura 9.7: *Circuito de control de fase de media onda y resistencia variable disparado a 0° .*

R_2 es un potenciómetro que ajusta el nivel de disparo para el SCR. Mediante el ajuste del mismo, el SCR se puede disparar en cualquier punto del ciclo positivo de la onda en alterna entre 0 y 180° , como se aprecia en la figura 9.8 página 105

Cuando el SCR se dispara cerca del principio del ciclo (aproximadamente a 0°), como en la figura 9.7 página 104, conduce durante aproximadamente 180° y se transmite máxima potencia a la carga. Cuando se dispara cerca del pico positivo de la onda, como en la figura 9.8 página 105, el SCR conduce durante aproximadamente 90° y se transmite menos potencia a la carga. Mediante el ajuste de R_X , el disparo puede retardarse, transmitiendo así una cantidad variable de potencia a la carga. Cuando la entrada en alterna es negativa, el SCR se apaga y no conduce otra vez hasta el siguiente disparo durante el ciclo positivo. Es necesario repetir el disparo en cada ciclo como se ilustra en la figura 9.9 página 105. El diodo se

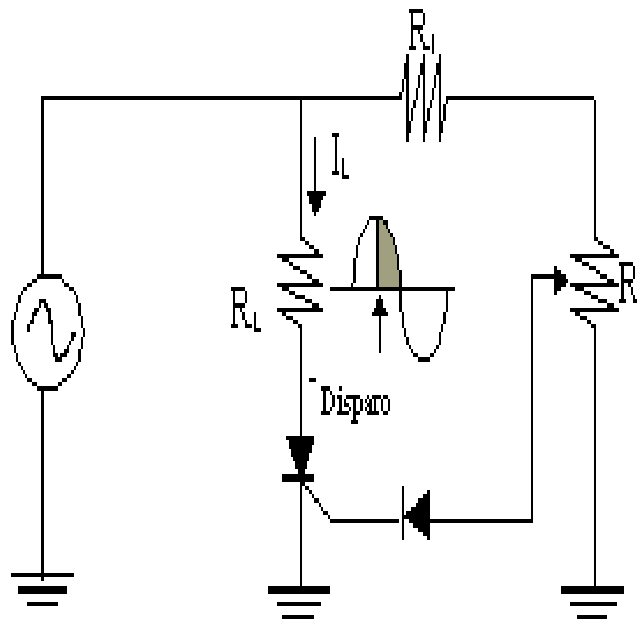


Figura 9.8: *Circuito de control de fase de media onda y resistencia variable disparado a 180° .*

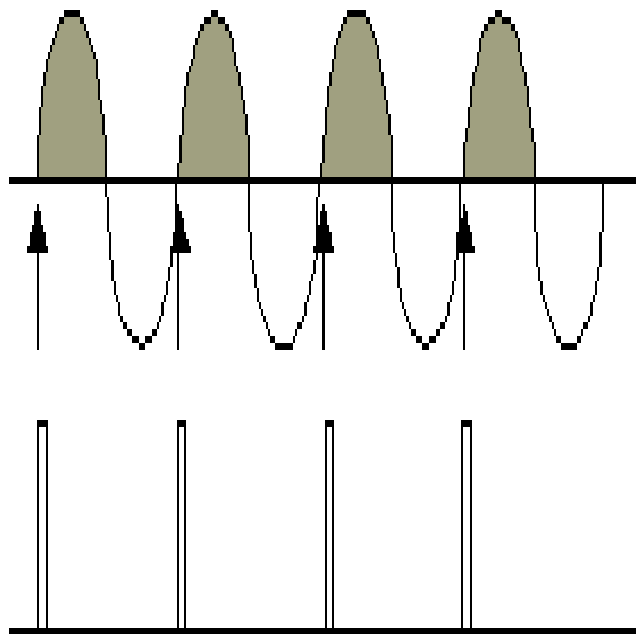


Figura 9.9: *Disparo en cada ciclo.*

coloca para evitar que voltaje negativo en alterna sea aplicado a la gate del SCR.

9.5 Triac

El triac (triode AC conductor) es un semiconductor capaz de bloquear tensión y conducir corriente en ambos sentidos entre los terminales principales T_1 y T_2 . Su estructura básica y símbolo aparecen en la figura 9.10 página 106. Es un componente simétrico en cuanto a conducción y estado de bloqueo se refiere, pues la característica en el cuadrante I de la curva V_{T2-T1}/I_{T2} es igual a la del cuadrante III como puede verse en el figura 9.12 página 108. Tiene una caída de tensión en conducción prácticamente iguales a las de un tiristor y el hecho de que entre en conducción, si se supera la tensión de ruptura en cualquier sentido, lo hace inmune a destrucción por sobretensión.

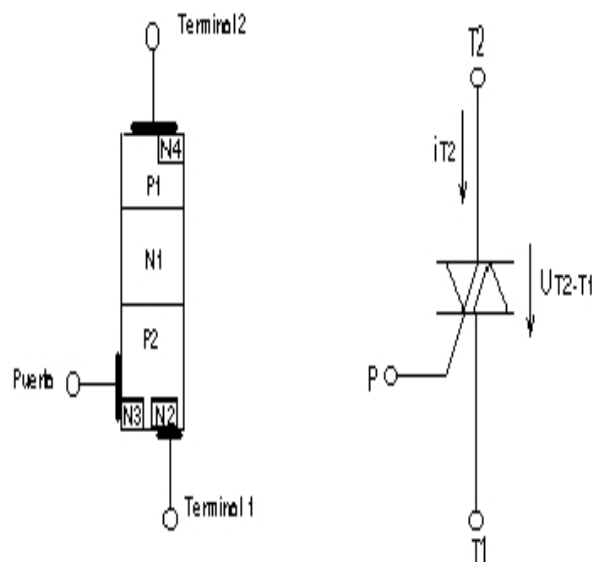


Figura 9.10: *Símbolo del triac.*

9.5.1 Circuito equivalente de un triac

Se puede considerar a un triac como si fueran dos SCR conectados en antiparalelo, con una conexión de compuerta común, como se muestra en la figura 9.11 página 107. Dado que el triac es un dispositivo bidireccional, no es posible identificar sus terminales como ánodo y cátodo. Si el terminal MT_2 es positivo con respecto a el terminal MT_1 , se activará al aplicar una señal negativa a la puerta, entre la puerta y el terminal MT_1 . No es necesario que estén presentes ambas polaridades en las señales de la puerta y un triac puede ser activado con una sola señal positiva o negativa de puerta. En la práctica, la sensibilidad variará de un cuadrante a otro, el triac normalmente opera en el cuadrante I (voltaje y corriente de puerta positivos) o en el cuadrante III (voltaje y corriente de puerta negativos).

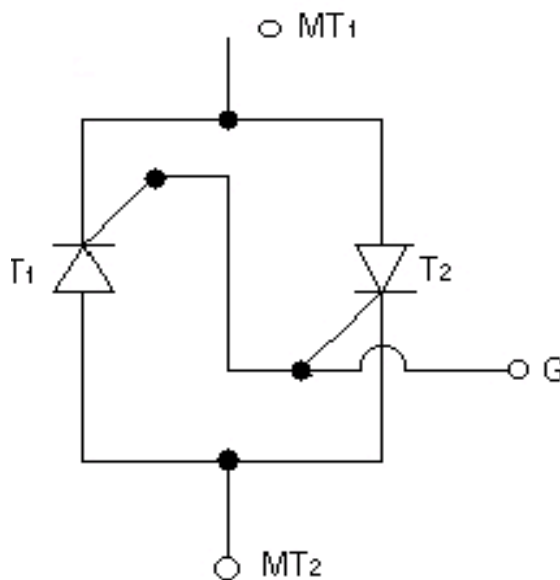


Figura 9.11: *Modelo de triac de dos SCR.*

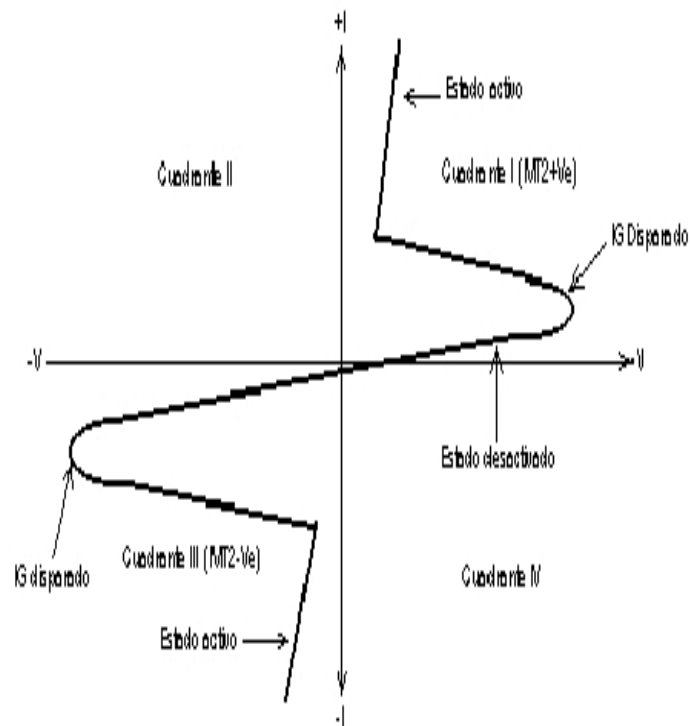


Figura 9.12: Gráfica de la característica V/I de un triac.

9.6 Conexión con la red eléctrica

9.6.1 Introducción

Este apartado describe un método simple para medir parámetros de la red eléctrica de las líneas de potencia alterna. Parámetros tales como cruce por cero y frecuencia. Este método es útil para medidas en sistemas de potencia de 50, 60 y 400 Hz con voltajes por encima de varios cientos de voltios. El método requiere solamente un componente externo: una resistencia.

9.6.2 Aplicaciones

Este método de medida puede usarse en cualquier aplicación donde los parámetros de la línea de potencia se usan para sistemas de medida y control. Aplicaciones típicas son temporizadores, corrección del factor de potencia, medida de potencia y motorización de la línea de potencia. Una aplicación adicional es generar una señal de reloj dada la relativa estabilidad de la frecuencia de la línea de potencia.

9.6.3 Teoría de operación

El método consiste en sacar partido de la protección estática de entrada que existe en los pines de entrada/salida del puerto GPIO del PIC 12C508A. Estos circuitos de protección están diseñados para cortocircuitar los pines de entrada a la fuente de alimentación cuando estos reciben un gran voltaje, de esta forma se protege al PIC de picos debidos a electricidad estática. En el PIC 12C508A este circuito de protección consiste en dos diodos PN colocados en cada entrada como se ilustra en la figura 9.13 página 110. Estos diodos cortocircuitarán cualquier voltaje mayor que V_{DD} a la fuente de V_{DD} y voltaje menor que V_{SS} a la fuente de V_{SS} . Pueden manejar varios miliamperios sin causar daño al chip. Se pueden aplicar grandes voltajes directamente al chip mientras estén limitados en corriente.

El método más barato de limitar en corriente consiste en usar una resistencia de gran valor. Esto se muestra esquemáticamente en la figura 9.14 página 111. R_9 está conectada a la línea de potencia y al pin GP4. Los diodos de protección ESD del puerto GPIO nos permite, como ya hemos señalado, realizar esta conexión sin daño para el chip.

Cuando el voltaje en la línea de potencia es positivo, el diodo de protección del pin a V_{DD} está en directa, así el buffer de entrada verá

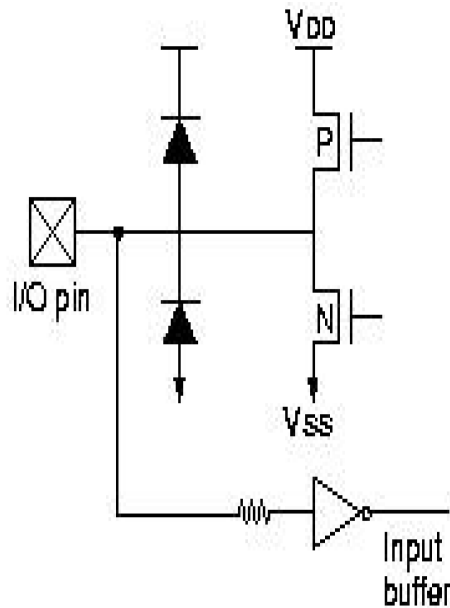


Figura 9.13: Estructura de un pin de entrada/salida.

aproximadamente $V_{DD} + 0.7$ voltios y el software leerá el pin como en nivel alto. Cuando el voltaje en la línea de potencia es negativo, el diodo de protección del pin a V_{SS} está en directa, así el buffer de entrada verá aproximadamente $V_{SS} - 0.7$ voltios y el software leerá el pin como en nivel bajo. Por tanto, el software puede detectar el cruce por cero sin más que testear un cambio de estado en GP4.

En la figura 9.15 página 112 se muestra la forma típica de la señal que entra al PIC. Para una señal senoidal de $115V_{ac}$ y 60 ciclos tarda $32\mu\text{seg}$ en pasar de 0 a $2V$, por tanto el umbral típico de $2V$ en los pines de entrada del PIC 12C508A permiten la detección del cruce por cero en $30\mu\text{seg}$.

Si la capacidad típica en un pin de entrada/salida es $5pF$, entonces R debería valer ($T = RC$) $6M\Omega$ o menos para optimizar el cruce por cero. Una resistencia de $5M\Omega$ con una señal aplicada de $155V_{ac}$ limitará la corriente a $32\mu A$, un valor que está dentro del

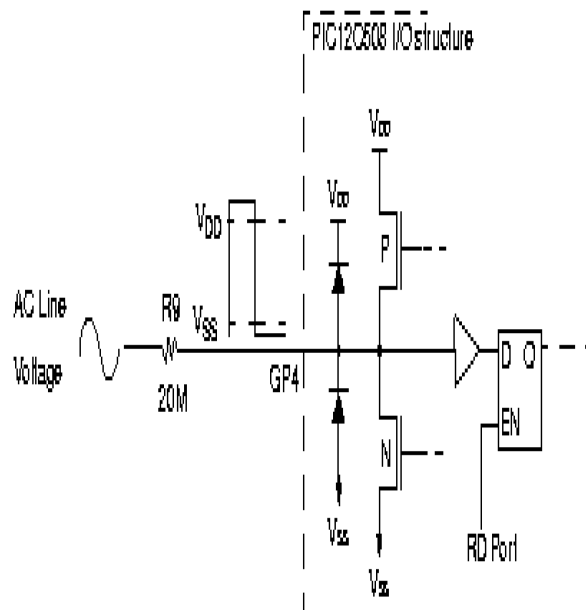


Figura 9.14: Conexión de la red eléctrica con un PIC.

margen de seguridad del PIC 12C508A. Por tanto debemos utilizar una resistencia a la entrada del pin con valores entre 5 y $6M\Omega$.

9.7 El diseño final

Ahora que ya conocemos el funcionamiento de un triac y que ya comprendemos como conectarnos con la red de distribución eléctrica, es el momento de abordar el diseño final adoptado para la parte del regulador de luz. La red ataca al PIC 12C508A por GP4, este como ya se ha explicado, detecta sin mayor problema el paso por cero de la señal. A su vez, el PIC 16F84A ataca con RB3 a RB0 a los pines GP3 a GP0 del GPIO dándole al PIC 12C508A el valor de intensidad de luz que se requiere, numerado del 0 al 15.

Internamente, el PIC 12C508A divide el semiciclo de la señal de red ($10mseg$) en 16 partes iguales, durando cada una $0,62mseg$, de

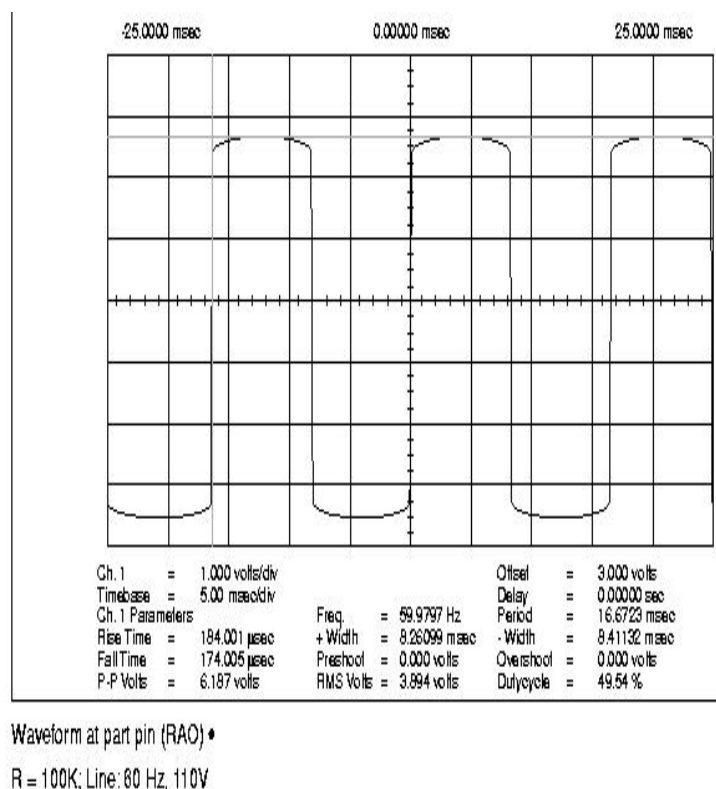


Figura 9.15: Señal en el pin de entrada.

esta forma el PIC 12C508A activa el pin GP5 al cabo de $N \cdot 0,62 \text{ msec}$ de la detección del paso por cero, siendo N el número transferido por el PIC 16F84A al PIC 12C508A.

Hay que hacer la observación de que realmente para evitar meternos en el semiciclo siguiente al recibir un cero, se han tomado $0,58 \text{ msec}$ para cada tramo.

De la explicación del funcionamiento de un triac se puede deducir fácilmente que no puede ser disparado por las tensiones TTL que usan nuestros microcontroladores, así pues, hemos usado un optotriac como interfaz para disparar el triac. De esta manera el PIC 12C508A por medio de GP5 dispara el optotriac, el que a su vez dispara al triac usando su pín de puerta, produciéndose un control

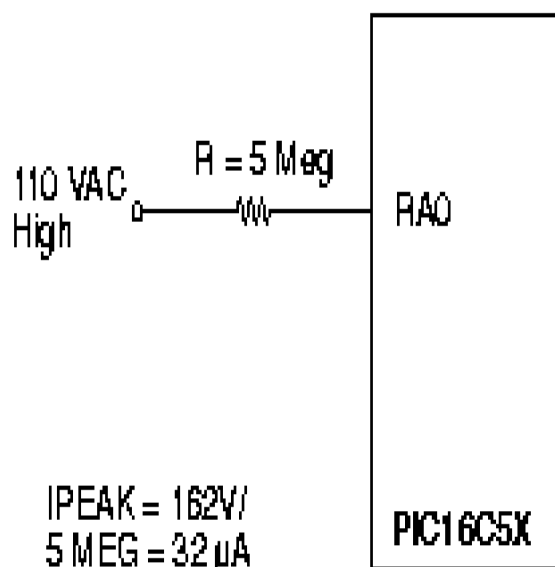


Figura 9.16: *Valores típicos de la conexión con la red eléctrica.*

sobre la parte de señal que se deja pasar, en definitiva un control sobre la potencia de señal alterna que dejamos pasar por el triac. Ver la figura 9.17 página 114. Ahora ya sin más que poner un enchufe en serie con el triac obtenemos un control de potencia sobre cualquier elemento susceptible de necesitar un enchufe. También queremos hacer notar la existencia de los llamados triac lógico que si pueden ser disparados con niveles TTL, pero en nuestro caso nos vimos obligados a usar aquello de lo que hemos podido disponer.

Como último comentario no debemos olvidar que la fuente de alimentación implementada para este diseño utiliza solamente componentes discretos y no tiene transformador o regulador de voltaje. La gran ventaja es el bajo coste económico y físico que supone. La desventaja es que esta fuente de alimentación no puede suministrar grandes corrientes, de forma que se debe tener cuidado de no sobrecargarla aumentando en excesivamente el número de componentes.

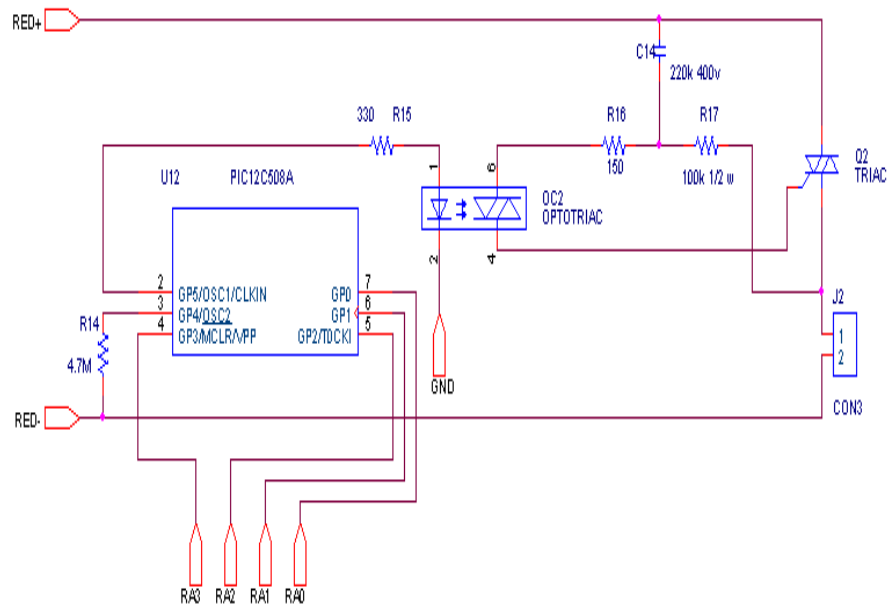


Figura 9.17: Esquema del regulador de luz.

Capítulo 10

El protocolo de comunicación: La capa física

10.1 Introducción

En este capítulo vamos a hablar sobre el protocolo de comunicación que hemos ideado para una correcta interpretación de la información entre los diferentes dispositivos que se pueden enchufar a la red eléctrica. Engloba diferentes partes puesto que se pensó ya con visión de futuro en algo más complejo que enviar un simple byte cada vez que necesitásemos un intercambio de información. Además había que establecer la posibilidad de un gran número de sensores enchufados al sistema, así como la gran cantidad de ruido que conlleva el uso de la red eléctrica para el transporte de información.

Para centrar ideas vamos a definir de una manera rápida las distintas partes del sistema. Una parte es la *placa central* que es la que comunica el PC con la red eléctrica, otra parte es la *placa sensora* que comunica un sensor de temperatura con la red eléctrica y por último la *placa actuadora* que comunica un regulador de luz con la red eléctrica.

10.2 Codificación

Empecemos por aclarar que es lo que tenemos y donde estamos, es decir, la parte clave es el PIC 16F84A que trabaja con lógica TTL: $0V$ es un cero lógico y $5V$ es un uno lógico, pero la comunicación física entre los distintos enchufes de la red se realiza obviamente por la red eléctrica. Luego tenemos que codificar esos unos y ceros lógicos del microcontrolador a una señal o una combinación de señales susceptibles de enviarse por la red eléctrica. Para esta codificación contamos con el módulo del CI LM555 que genera una señal de una frecuencia concreta bajo mandato del PIC. Debe quedar claro entonces que la comunicación es bit a bit dado nuestro diseño.

A primera vista la solución más factible es la realización de una modulación FSK simple, así usar por ejemplo la ausencia de nuestra señal monofrecuencia como codificación del cero lógico, y tomar como codificación del uno lógico la presencia en la red de la señal. De esta manera, y dado que el CI LM555 emite cuando recibe un uno lógico y que el módulo del CI NE567 envía un uno lógico al recibir dicha señal a su entrada, la comunicación sería muy sencilla para los PIC puesto que sería totalmente transparente para ellos, como si estuviesen directamente conectados.

Esta primera idea es interesante por su sencillez pero resulta muy poco efectiva contra el ruido, ante cualquier tipo de ruido que incluya pulsos en la red puede dar lugar a la confusión del receptor creyendo que recibe un uno lógico cuando en realidad lo que se envió fue un cero lógico. A partir de aquí las posibilidades son múltiples y variadas, podemos pensar en hacer más largos los unos que los ceros o viceversa, o realizar algún tipo de codificación intermedia entre el nivel lógico a transmitir al CI LM555 y lo que realmente se le transmite o incluso usar una modulación FSK con dos frecuencias una para el cero lógico y otra para el uno lógico.

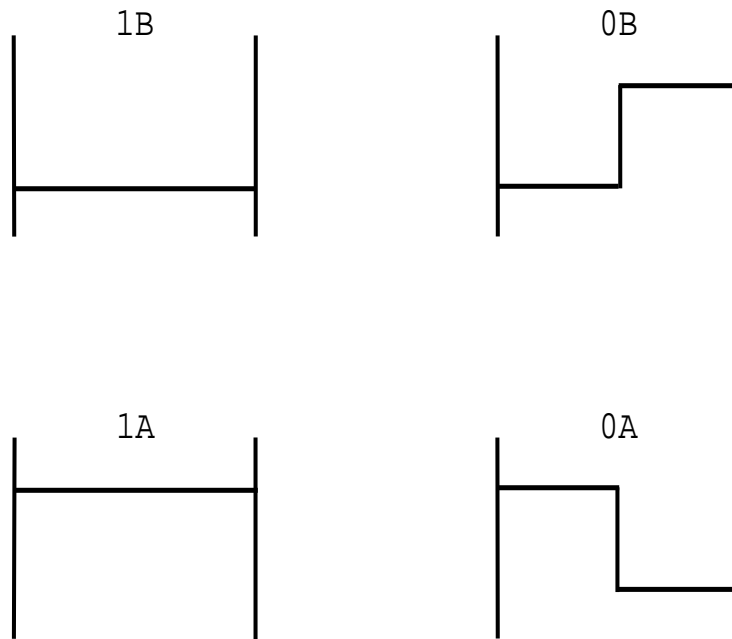
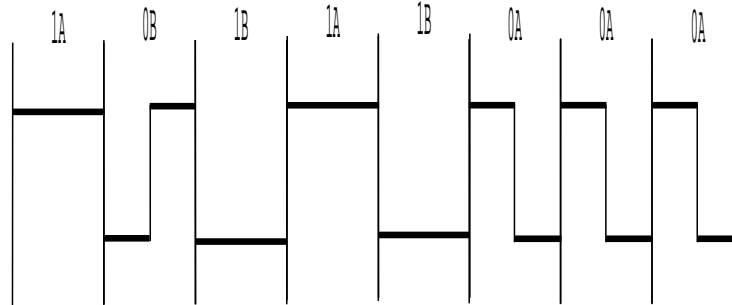
Nosotros hemos descartado de principio la inclusión de dos fre-

cuencias para no incluir más hardware a nuestros diseños con el fin de hacer algo funcional pero también pequeño.

Hemos decidido realizar una codificación intermedia que sigue unas reglas sencillas que un microcontrolador puede implementar sin grandes problemas, ver figura 10.1 página 118:

- Hay dos tipos de unos lógicos, 1A y 1B, y hay dos tipos de ceros lógicos, 0A y 0B, para una duración T de bit dada.
- El uno lógico tipo 1A vale $5V$ durante un tiempo igual a T .
- El uno lógico tipo 1B vale $0V$ durante un tiempo igual a T .
- El cero lógico tipo 0A vale $5V$ durante un tiempo igual a $T/2$ y vale $0V$ en el siguiente intervalo $T/2$.
- El cero lógico tipo 0B vale $0V$ durante un tiempo igual a $T/2$ y vale $5V$ en el siguiente intervalo $T/2$.
- A la hora de formar cada byte se empieza siempre por un uno tipo 1A si lo que hay que mandar es un “1” o se empieza por un cero tipo 0A si lo que hay que mandar es un “0”, nunca se empezará un byte por un uno tipo 1B o por un cero tipo 0B.
- A la hora de ir intercalando los bits dentro de un byte se sucederán, siguiendo la regla anterior, y teniendo en cuenta que siempre se produzcan un número máximo de transiciones.

Por ejemplo, si tenemos un byte que empieza por un bit “1” y luego un “0”, la trama se codificará con un 1A seguido de un 0B para que exista transición de $5V$ a $0V$ entre el 1A y el 0B. Otro ejemplo, el siguiente byte (msb) 00011101 (lsb) si se empezase la transmisión por el menos significativo lo que se transmitirá realmente al CI LM555 después de esta codificación previa es algo como lo siguiente (lsb) 1A0B1B1A1B0A0A0A (msb). Ver figura 10.2 página 118.

Figura 10.1: *Los cuatro tipos de bits.*Figura 10.2: *Un ejemplo de trama.*

Resumiendo, desde el PIC hasta la red eléctrica hay dos pasos fundamentales en la codificación de la información. El primero, siguiendo esta serie de reglas, se realiza en el interior del PIC, es decir, lo que sale del PIC, aprovechando el ejemplo anterior es esa trama de 1A0B1B..., sus tensiones correspondientes: 5V durante el tiempo T, 0V durante un tiempo T/2 seguido de 1V durante otros T/2,... Y esto es lo que realmente ataca a la patilla 5 del CI

LM555. Así el segundo paso de la codificación es la conversión, que realiza ya el módulo del CI LM555, de esa trama de 1A0B1B... en ausencia o presencia de la señal monofrecuencia emitida por el propio CI LM555.

Todo esto no ha sido realizado de forma gratuita sino porque en el medio de transmisión que usamos existe mucho ruido. Ruido procedente de todos los aparatos conectados a la red eléctrica y por las señales que se acoplan de radiofrecuencia. Pues bien estas reglas se han hecho con el fin de obtener un código más resistente al ruido, es análogo al uso de códigos ya existentes como puede ser el código Manchester.

Ya como comentario final adelantar que en nuestras transmisiones enviamos un sincronismo seguido de cuatro bytes y que las reglas expuestas se aplican a cada byte independientemente de los otros tres.

10.3 Todos escuchan a todos

En la introducción a este capítulo hemos comentado que el sistema se puede dividir en tres partes fundamentales en cuanto a dispositivos conectados directamente a la red eléctrica. Bien, estas tres partes a su vez las podemos englobar en dos fundamentales formando un sistema maestro-esclavo: la placa central que hace las funciones de maestro y la placa sensora y la placa actuadora que son ambas esclavas. Las comunicaciones del maestro con los esclavos se realizan por el cable de la red eléctrica, entonces sólo tenemos un único medio físico para el intercambio de información. Por tanto el maestro envía información y se queda esperando una respuesta de la placa esclava correspondiente. Pues bien para diferenciar el destino de la información entre las dos placas esclavas hemos establecido un número diferente para cada placa que sirve de dirección. Esta

dirección se envía en los primeros seis bits de la trama completa de cuatro bytes, así lo primero que hace el destino al recibir datos por la red eléctrica es testear que la dirección recibida es la suya, sino es así, se pone otra vez a la escucha en el caso de una placa esclava o se pone a escuchar el puerto serie en el caso de la placa central.

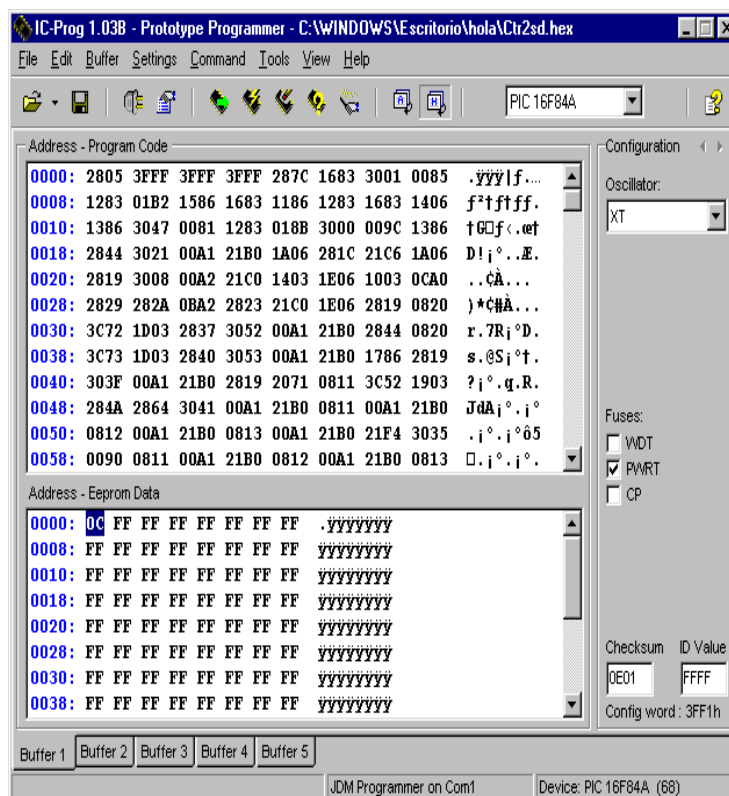


Figura 10.3: Grabación de la dirección en la E^2 PROM.

Ahora bien, para establecer esas direcciones las decidimos en tiempo de grabación del programa en el PIC, es decir en tiempo de diseño como se puede ver en la figura 10.3 página 120. El grabador nos permite grabar en la posición que nos interese de la ROM, hemos elegido la posición cero de la ROM. Por tanto todos los PIC buscarán en su dirección cero de la ROM para saber su dirección. Este es un sistema poco dinámico y versátil pero es lo más sencillo, si bien una posible mejora consistiría en que las direcciones en la ROM

se grabaran en tiempo de uso del sistema, es decir que al enchufar en la red eléctrica la placa de un sensor, éste recibiese una trama que le asignase una dirección para grabarla en su ROM y trabajar a partir de ahí.

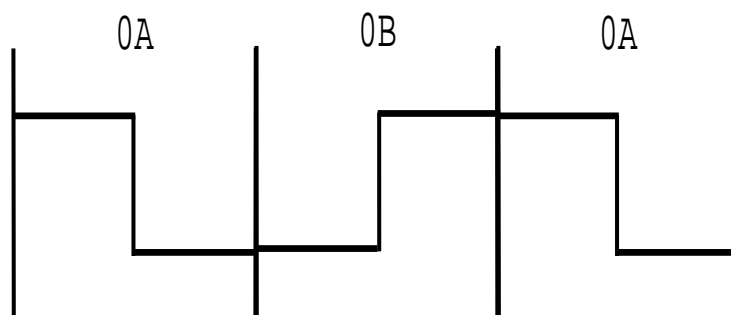
10.4 Sincronización

Hasta ahora hemos hablado de intercambio de información y de su codificación pero no hemos comentado que para que el microcontrolador, de la placa que actúa en un determinado momento como receptor, reciba datos correctos este deberá testear la salida de su módulo del CI NE567 cada $T/4$ siendo T el tiempo de duración de un bit. Para ello debemos sincronizarnos al principio de la trama y trabajar a partir de ahí con un contador que nos diga cada cuanto tenemos que mirar la salida del CI NE567. El contador que usamos es el temporizador interno del propio PIC junto con el prescaler que se le puede asignar.

Por tanto en nuestras transmisiones incluimos una trama de sincronismo compuesta por tres bits, pero para evitar que se pueda repetir en el interior de los bytes de información nos saltamos una de las reglas de codificación: vamos a permitir que entre estos tres bits no haya transiciones entre bit y bit, así la trama está compuesta invariablemente por (mbs) 0A0B0A (lsb) donde empezamos a transmitir por el bit menos significativo. Ver figura 10.4 página 122.

10.5 Tramas de control

En este apartado vamos a analizar algo que no está implementado pero que hemos dejado sitio en la trama para usarlo con una idea muy concreta: para diferenciar que tipo de trama se está recibiendo. Es decir, el receptor podría recibir diferentes ordenes como petición

Figura 10.4: *La trama de sincronismo.*

de temperatura, conexión a la red (obtención de una dirección) o desconexión. Son dos bits enviados justo después de la dirección y que complementan juntos un byte. Estos dos bits nos dan lugar a cuatro posibles tramas de control con las que poder trabajar, está en manos de un futuro trabajo desarrollar esta idea.

10.6 Corrección de errores

Además de todo lo comentado hemos añadido al final de la trama un byte exclusivamente para desarrollar algún tipo de corrección de errores o simplemente repetir información ya mandada aunque sea redundante para tratar siempre de contrarrestar los errores producidos por lo ruidosa de la red eléctrica. Nosotros efectivamente hemos utilizado este byte de forma redundante, queda para trabajos futuros desarrollar un buen algoritmo de corrección contra errores.

10.7 Distribución de la información

Veamos, hasta ahora hemos hablado de seis bits de dirección y de dos bits de control, todos juntos forman un byte, el primero que será transmitido y que denominaremos DIR. A continuación se envían dos campos de información de un byte cada uno que portan datos

útiles para el sistema como puede ser la temperatura o la intensidad de luz, los llamaremos a partir de ahora INF1 e INF2. Y como final de la trama un byte para corrección de errores que llamaremos ERR.

Por tanto cada vez que cualquiera de las placas envía información lo hace siguiendo siempre un mismo patrón: *sincronismo* + *DIR* + *INF1* + *INF2* + *ERR* dando lugar a tramas invariablemente de tres bits de sincronismo más cuatro bytes donde ahora sólo sacamos verdadero partido de los dos primeros bytes.

10.8 Un ejemplo de una trama

En la figura 10.5 página 124 podemos ver un ejemplo de trama completa con los siguientes campos:

- Sincronismo.
- DIR (msb) 00011101 (lsb).
- INF1 (msb) 00000011 (lsb).
- INF2 (msb) 11110001 (lsb).
- ERR (msb) 11111000 (lsb).

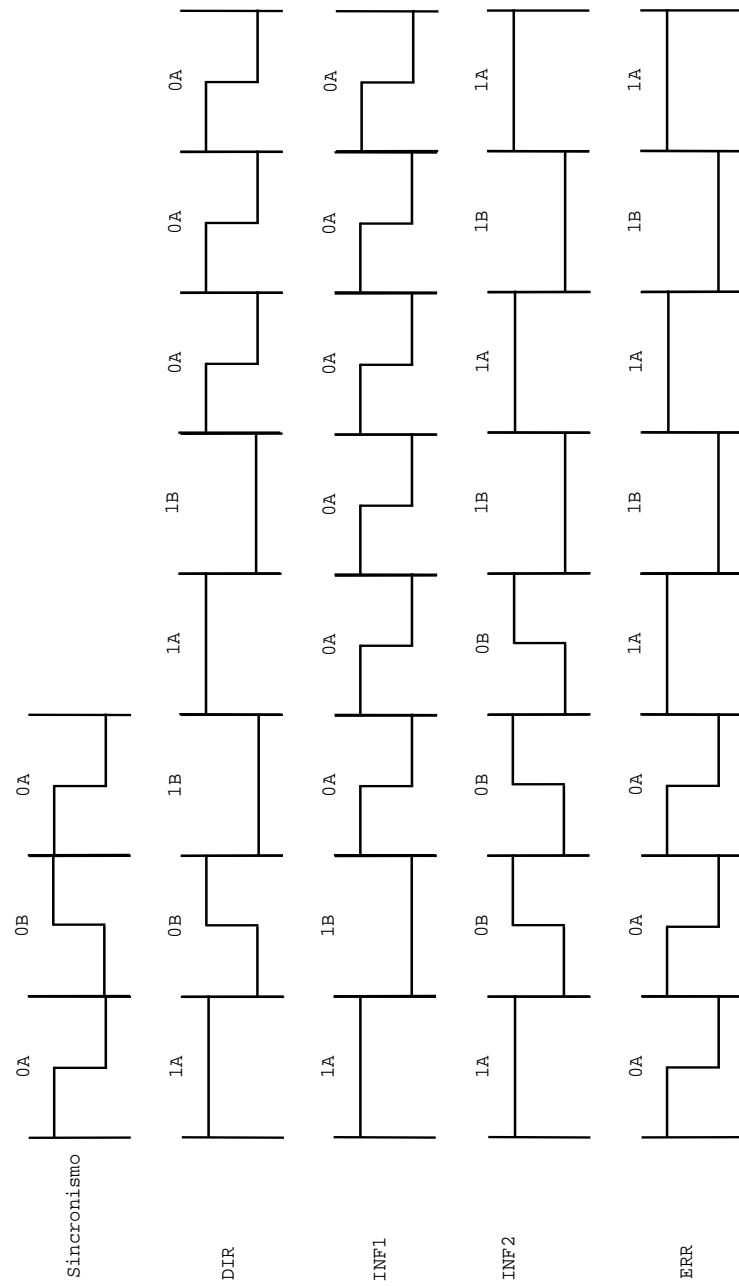


Figura 10.5: *Un ejemplo de una trama completa.*

Capítulo 11

Código de los μC

11.1 El PIC 16F84A de la placa central

Sabemos que la placa central es la que hace de interfaz entre nuestra aplicación Dome del PC y la red eléctrica, por tanto el microcontrolador de esta placa debe atender al puerto serie del PC y a la línea de red. El sistema completo está controlado por Dome, es decir, es en la aplicación Dome donde se toman las decisiones de lo que se quiere hacer, y la placa central simplemente obedece y adapta esa información a la red eléctrica. Así en el inicio de una comunicación el microcontrolador está escuchando únicamente el puerto serie a la espera de recibir un comando. Todo esto lo podemos ver en el diagrama de flujo de la figura 11.1 página 129.

El primer paso siempre es la configuración del microcontrolador, debemos reservar memoria, definir la configuración de pines, asignar el temporizador y declarar las interrupciones. Una vez realizado este primer paso, el microcontrolador se queda en un bucle esperando recibir un comando válido por el puerto serie del PC. Además de los comandos de depuración, admite dos comandos fundamentales: una *t* indicando petición de temperatura o un *número* indicando proporcionalmente la intensidad de luz requerida en la placa actu-

adora. Ante cualquiera de estos dos comandos el microcontrolador pasa a enviar la información por la red eléctrica, si se trata de la *t* envía una petición de temperatura y si se trata de un *número* lo envía tal cual. Una vez realizado el envío el microcontrolador arranca una temporización dentro de la cual tendrá que producirse la recepción de la contestación, en caso contrario se terminará la temporización y abandonará la recepción por la red volviendo a esperar recibir un comando del puerto serie, evitando colgar el sistema si por alguna causa el microcontrolador no consigue detectar una contestación correcta, ya sea debido a la aparición de ruido o porque la trama recibida no coincide con su dirección. Pero si todo ha ido bien recibirá una contestación correcta dentro de la temporización, en este caso enviará al PC la temperatura si es lo que se había pedido o enviará al PC la confirmación de la placa actuadora. Acto seguido el microcontrolador vuelve a su estado inicial de esperar un comando enviado desde Dome.

Hemos hecho una mera descripción del programa ejecutado en el microcontrolador de la placa central, pero el código entero está disponible en el CDROM que acompaña a esta memoria.

11.2 El PIC 16F84A de la placa sensora

Sabemos que la placa sensora es la que tiene el sensor de temperatura y una interfaz para comunicación por la red eléctrica. Además también sabemos que es Dome quien controla el sistema, es decir, la petición de la temperatura empezará en Dome pasando a la placa central y desde allí a través de la red eléctrica llega a la placa sensora que después de medir la temperatura enviará su valor de vuelta por la red. Por tanto el microcontrolador de la placa sensora está escuchando únicamente la red eléctrica a la espera de una recepción válida. Todo esto lo podemos ver en el diagrama de flujo de la figura 11.2 página 130.

El primer paso siempre es la configuración del microcontrolador, debemos reservar memoria, definir la configuración de pines, asignar el temporizador y declarar las interrupciones. Una vez realizado esto, el microcontrolador se queda en un bucle esperando una recepción válida por la red eléctrica. Para que dicha recepción sea válida debe ocurrir que obtengamos el sincronismo sin errores y además que la dirección que recibimos en la trama coincida con la dirección que el microcontrolador tiene almacenada en su *E²PROM*: 0C. Si todo va bien, se recibe una trama correcta ante la que el microcontrolador pasa a testear si se trata de una trama de acción, sino fuera así nos volveríamos a esperar una nueva recepción por el puerto. Esta comprobación del tipo de trama ilustra la posibilidad de añadir diferentes tipos de tramas permitiendo ampliar la funcionalidad al sistema en futuros trabajos. Si se trata efectivamente de una trama de acción el microcontrolador pasa a realizar la conversión sigma-delta tras la cual obtiene el valor de la temperatura y la envía de vuelta por la red eléctrica. Tras este paso, el microcontrolador de nuevo se queda esperando una nueva recepción procedente de la red eléctrica.

Hemos hecho una mera descripción del programa ejecutado en el microcontrolador de la placa sensora, pero el código entero está disponible en el CDROM que acompaña a esta memoria.

11.3 El PIC 16F84A de la placa actuadora

Sabemos que la placa actuadora es la que tiene el regulador de luz y una interfaz para comunicación por la red eléctrica. Además también sabemos que es Dome quien controla el sistema, es decir, es Dome quien ordena y manda que intensidad de luz debe haber en la placa actuadora. Desde Dome se envía el nuevo valor de intensidad pasando a la placa central y desde allí a través de la red eléctrica llega a la placa actuadora que modifica la intensidad de luz de acuerdo

con dicho valor, a continuación manda una confirmación de vuelta por la red. Por tanto el microcontrolador de la placa actuadora está escuchando únicamente la red eléctrica a la espera de una recepción válida. Todo esto lo podemos ver en el diagrama de flujo de la figura 11.3 página 131.

El primer paso siempre es la configuración del microcontrolador, debemos reservar memoria, definir la configuración de pines, asignar el temporizador y declarar las interrupciones. Una vez realizado esto, el microcontrolador se queda en un bucle esperando una recepción válida por la red eléctrica. Para que dicha recepción sea válida debe ocurrir que obtengamos el sincronismo sin errores y además que la dirección que recibimos en la trama coincida con la dirección que el microcontrolador tiene almacenada en su *E²PROM*: 0E. Si todo va bien, se recibe una trama correcta ante la que el microcontrolador pasa a testear si se trata de una trama de acción, sino fuera así nos volveríamos a esperar una nueva recepción por el puerto. Esta comprobación del tipo de trama ilustra de nuevo la posibilidad de añadir diferentes tipos de tramas permitiendo ampliar la funcionalidad del sistema en futuros trabajos. Si se trata efectivamente de una trama de acción el microcontrolador pasa el valor recibido de intensidad al microcontrolador PIC 12C508A que reajusta con el nuevo dato los tiempos de disparo del triac. Tras esto el microcontrolador principal envía una confirmación por la red eléctrica y de nuevo se queda esperando una nueva recepción procedente de la red eléctrica.

Hemos hecho una mera descripción del programa ejecutado en el microcontrolador de la placa actuadora, pero el código entero está disponible en el CDROM que acompaña a esta memoria.

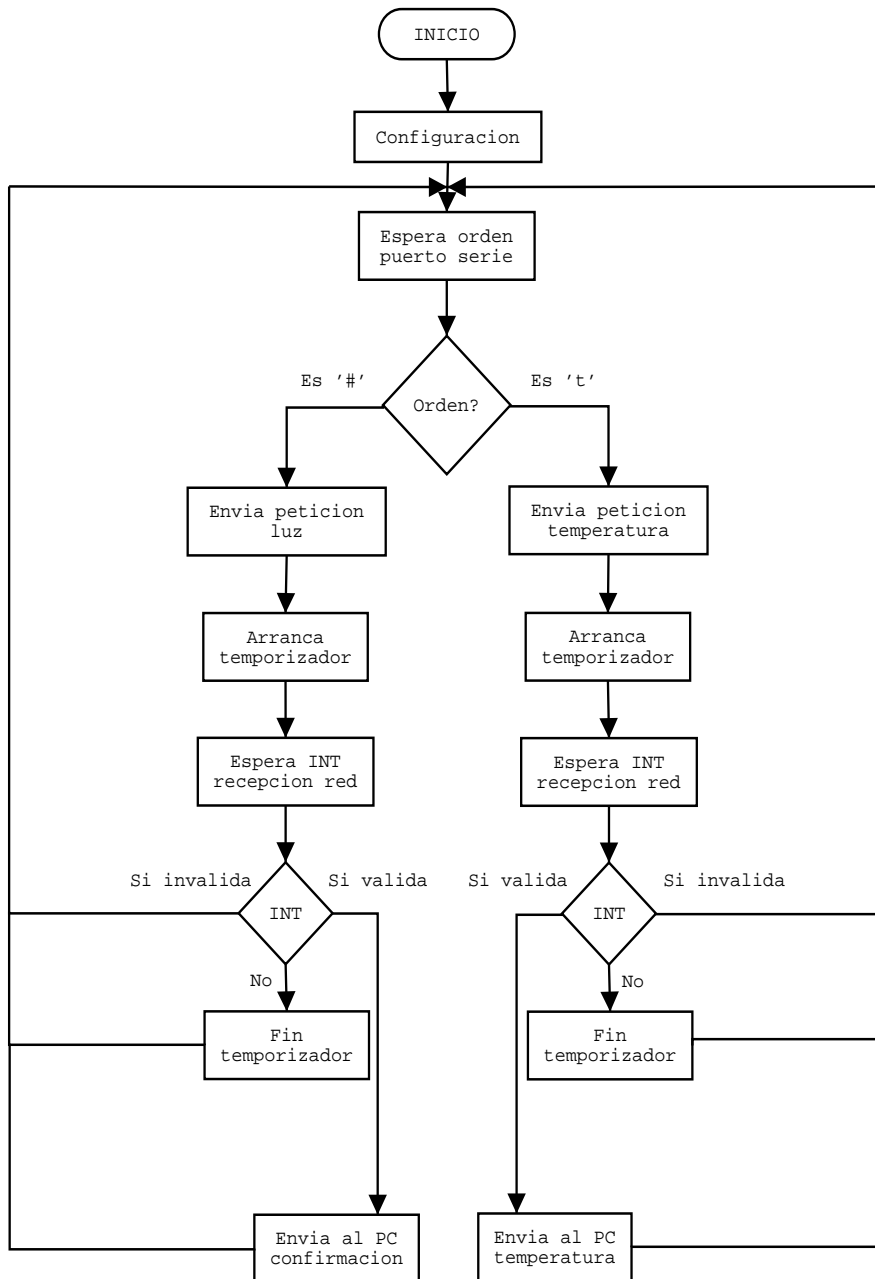


Figura 11.1: Diagrama de flujo del microcontrolador de la placa central.

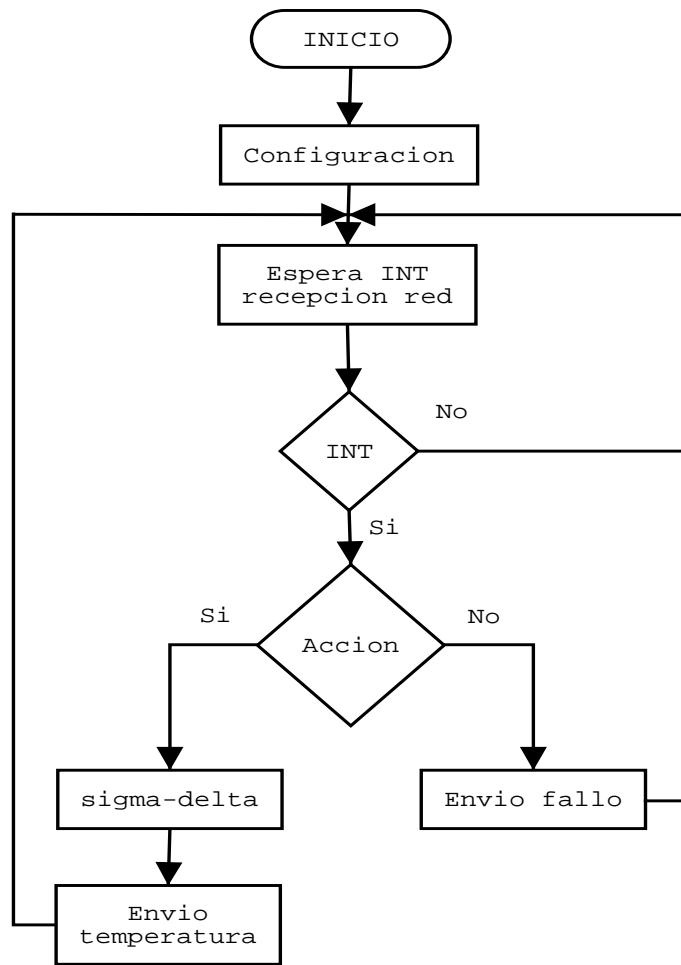


Figura 11.2: *Diagrama de flujo del microcontrolador de la placa sensora.*

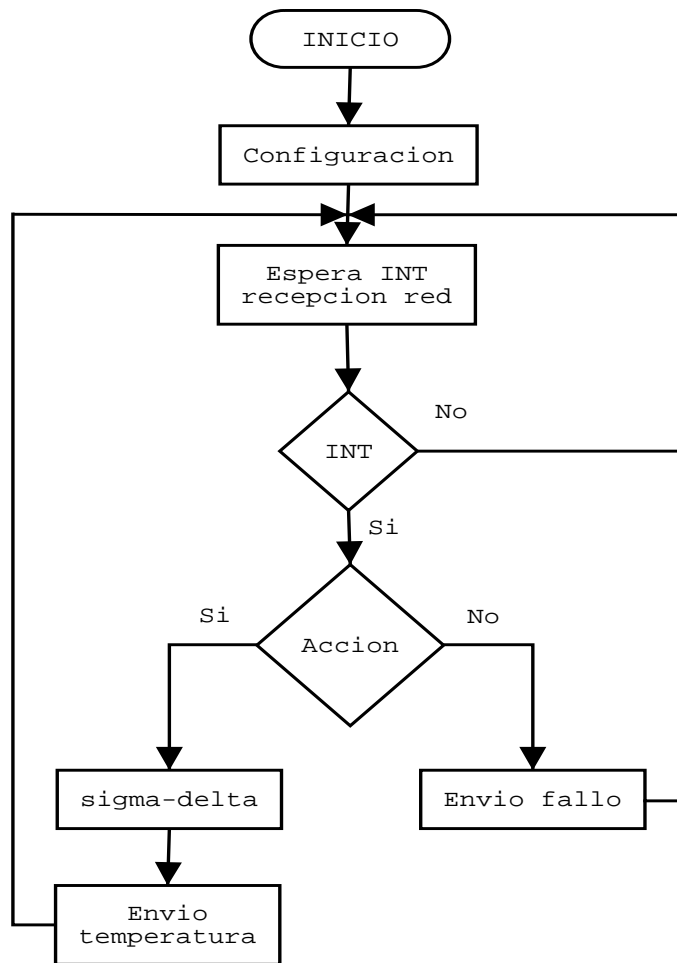


Figura 11.3: Diagrama de flujo del microcontrolador de la placa actuadora.

Capítulo 12

Dome

12.1 Introducción

Dome es la aplicación que corre sobre el PC y que implementa la comunicación con la placa central mediante una interconexión RS232. Su función es la de proporcionar un entorno amigable para el usuario además de mostrar las prestaciones de las tres palcas, a saber, placa central, placa sensora y placa actuadora. Realiza el papel de maestro en un esquema maestro/esclavo con la placa central, la cual a su vez es el maestro en el esquema maestro/esclavo que forma por su parte con las placas sensora y actuadora.

Es una aplicación basada en la API Win32 y que debería funcionar correctamente en Windows 9X y Windows ME. Tiene un esquema de cuadro de diálogo en el que se han embebido un menú de aplicación y los diferentes controles para el envío y la recepción de datos hacia y desde la placa sensora y la actuadora a través de la placa central. Está programada y compilada usando el paquete de desarrollo Visual C++ 6.0 y las MFC.

12.2 Interconexión RS232

Las señales disponibles en un conector RS232 están pensadas únicamente para asegurar la correcta transmisión y recepción de datos desde un equipo denominado DTE (Data Terminal Equipment) a un DCE (Data Communication Equipment). Un DTE es generalmente un ordenador y un DCE un módem. En la figura 12.1 página 134 vemos el enlace estándar entre un DTE y un DCE, mientras que la función de cada una de las señales la vemos en la tabla 12.1 página 135.

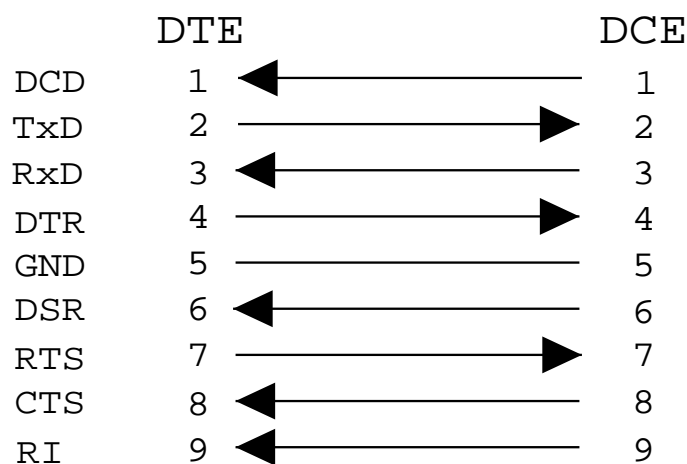


Figura 12.1: Conexión de las diferentes señales en el protocolo RS232.

TxD se encarga de transportar los datos serie hasta el módem. Para ello, han tenido que activarse RTS, CTS, DTS y DTR.

RxD es la recepción de datos, no dependiendo de ninguna otra señal RS232.

RTS tiene como misión conmutar un módem semiduplex entre modos de recepción y transmisión. Cuando DTE quiere transmitir, informa al módem de su deseo activando esta señal. Cuando el módem conmuta para transmisión, se lo informa al DTE activando

Señal	Nombre	Función
TxD	Transmitted Data	Salida de datos DTE
RxD	Received Data	Entrada de datos DTE
RTS	Request To Send	DTE desea cambiar a transmisión
CTS	Clear To Send	DCE listo para transmitir
DSR	Data Set Ready	DCE listo para comunicar con DTE
GND	Signal Common	Línea común del circuito (tierra)
DCD	Data Carrier Detect	Detectar si está conectado
DTR	Data Terminal Ready	Pone a trabajar al módem
RI	Ring Indicator	Anuncia una llamada

Tabla 12.1: Función de las diferentes señales en un conector RS232.

CTS, indicando que ya puede enviar los datos.

El módem origen no transmite ni activa su DSR hasta recibir el tono de respuesta del módem remoto.

DCD se activa cuando el módem recibe una portadora remota. En un módem semi-duplex, DCD se activa únicamente en el módem receptor.

Una vez que el módem esté conectado a la línea, DTR deberá permanecer activa mientras dure la conexión, si se inhibe, se produce la desconexión, interrumpiendo bruscamente el enlace.

Además del enlace estándar, existen otros, como la conexión denominada módem nulo (cable se seis hilos), utilizada generalmente para transferir ficheros entre dos ordenadores. Esta conexión, como su nombre indica, no es en absoluto un módem, sino una conexión directa entre dos ordenadores (DTE) para comunicarse siguiendo las reglas lógicas del protocolo RS232. Otra solución para la comunicación, más sencilla todavía, es la conexión de tres hilos (TxD, RxD y GND).

De lo anterior, puede deducirse que para que exista una comuni-

cación entre dos equipos tiene que haber un acoplamiento entre ellos, y dicho acoplamiento puede realizarse por software o por hardware.

El acoplamiento hardware sólo puede realizarse si ambos equipos están físicamente conectados mediante un cable. Se suele realizar mediante señales DTR/DSR o bien utilizando las señales secundarias RTS/CTS.

El acoplamiento software no siempre es posible, ya que para que pueda darse, los equipos deben reconocer caracteres de control. En un acoplamiento software es el receptor el que controla el acoplamiento. Lo hace de la forma siguiente; cuando su cola de entrada está llena, envía un caracter de desconexión (normalmente el 0x13, en ASCII el XOFF), cuando el transmisor recibe este carácter se detiene. Cuando la cola de entrada del receptor puede recibir más caracteres, envía un carácter de conexión (normalmente el 0x11, en ASCII el XON), cuando el transmisor recibe este carácter reinicia el envío de caracteres.

12.3 Comunicaciones por el puerto serie

La gestión los puertos de comunicación no nos ha resultado una tarea fácil. Lo primero que hay que pensar es que los datos llegan a los puertos de manera asíncrona. Esto sugiere que el dato que llega tiene que procesarse inmediatamente, puesto que pueden llegar otros datos. de esta tarea se encarga el hardware del PC, de forma que cuando detecta la llegada de un dato, interrumpe el flujo normal del proceso para ceder el control a la rutina de proceso de comunicaciones. Esta rutina tiene que ser una rutina de Windows en lugar de una rutina de la aplicación, por dos razones básicamente:

1. Windows debe mantener el control de la multitarea. En efecto, si la llegada de un dato hiciera que se transfiriera el control

del procesador a nuestra aplicación, Windows perdería su capacidad para gestionar la multitarea. Esto quiere decir que Windows tiene que estar entre la aplicación y el hardware.

2. Windows no puede dirigir el dato recibido directamente a la aplicación. La razón es que todos los datos que recibe en el puerto de comunicaciones no llegan con la identidad de la aplicación que los tiene que recibir. Por lo tanto, Windows tiene que guardar en un buffer los datos que llegan para una aplicación. Para qué aplicación, debe determinarse por adelantado, esto es, nuestra aplicación debe haberle pedido a Windows la propiedad del puerto.

Cuando una aplicación solicita a Windows la propiedad de un puerto, Windows sólo se lo dará si ninguna otra aplicación lo tiene. Por el mismo motivo, mientras nuestra aplicación tiene el control del puerto, Windows se lo prohíbe a las demás aplicaciones que lo soliciten. Cuando la aplicación termine la operación de E/S con el puerto, debe dejar el control del mismo para que otras aplicaciones puedan utilizarlo.

Existen dos formas de abrir un puerto de comunicaciones:

Solapada (overlapped) Un puerto abierto para operaciones solapadas permite múltiples hilos realizando operaciones de E/S simultáneas, así como realizar otra tarea mientras las operaciones estén pendientes.

No solapadas (nonoverlapped) Si el puerto se abre para operaciones no solapadas, el hilo queda bloqueado mientras la operación de E/S solicitada no esté completada. En este caso si un hilo queda bloqueado mientras la operación de E/S finalice, cualquier otro hilo que requiera una operación de E/S quedará bloqueado.

12.4 Interfaz de comunicaciones

La interfaz de comunicaciones ha sido integrada en la clase vista de nuestra aplicación mediante una serie de funciones miembro que describimos a continuación.

La función *Iniciar* obtiene del registro de Windows la configuración por omisión del dispositivo de comunicaciones, mientras que la función *Terminar* será invocada cuando se corta la comunicación para guardar en el registro de Windows la configuración actual del dispositivo de comunicaciones.

La función *EstablecerConexión* permite abrir el puerto de comunicaciones especificado por la variable miembro *m_indPuerto*, con las propiedades del resto de las variables índice *m_ind...* de forma solapada. También invoca a la función **SetCommMask** para especificar los eventos que serán atendidos, a la función **SetupComm** para especificar el tamaño de las colas de recepción y de transmisión, a **PurgeComm** para finalizar las operaciones de E/S pendientes, construye una estructura DCB e invoca a la función **SetCommState** para configurar el dispositivo de comunicaciones con los valores almacenados en dicha estructura, invoca a **SetCommTimeouts** para establecer los tiempos límites para operaciones de recepción y transmisión, lanza un hilo dedicado a controlar cada evento de interés que se produzca en el puerto de comunicaciones y invoca a la función **EscapeCommFunction** para activar la señal DTR mientras dure la conexión. La estructura DCB está definida en *windows.h* y recoge todos los parámetros relacionados con la configuración de un puerto.

La función *MensajeDeError* convierte un código de error en el correspondiente mensaje obtenido del sistema, y visualiza un diálogo con dicho mensaje.

La función *ConfigurarDisCom* se usa para construir la estructura

DCB y configurar el dispositivo de comunicaciones así como para establecer los tiempos límites para las operaciones de recepción y de transmisión, *EstablecerConexion* invoca a *ConfigurarDisCom*.

La función *EstablecerConexion*, después de abrir el puerto de comunicaciones, lanza un hilo *ControlarEventos*. Cuando ocurre un evento sobre el dispositivo de comunicaciones, este hilo secundario se lo notifica al hilo principal (a la aplicación) enviándole un mensaje, la respuesta a este mensaje es la función *OnEventoCom* que realiza un proceso u otro en función del evento ocurrido. Por ejemplo si el evento es que se recibió información en la cola de recepción del dispositivo de comunicaciones, esta función leerá dicha información y la procesará; en nuestro caso la visualizará en la caja de “Dato recibido” y si se trata de información procedente de la placa sensora la mostrará en los “displays” de temperatura.

La función *LeerCaracteresPuerto* lee un bloque del puerto y lo almacena en un array, leerá como máximo BLOQUEMAX caracteres mediante la función **ReadFile**. Por el contrario la función *EscribirCarsPuerto* escribe un bloque en el puerto procedente de un array pasado como parámetro.

La función *CortarConexion* cierra el puerto de comunicaciones. Para ello, pone la variable *m_ConexionEstablecida* a valor FALSE para que el hilo *ControlarEventos* pueda finalizar, inhabilita los eventos, desactiva la línea DTR, termina las operaciones de E/S, limpia las colas de recepción y transmisión y cierra el puerto de comunicaciones.

12.5 Interfaz del usuario

Cuando se ejecuta la aplicación una de las primeras tareas que hay que realizar es leer la configuración almacenada en el registro de Windows. Este proceso lo realizaremos desde la función **OnIni-**

tialUpdate, que también se encarga de inhabilitar los controles que hacen uso de la comunicación con el puerto (situación que persiste hasta que se abre un puerto de comunicaciones).

Cuando se establece una conexión entre dos equipos, previamente hay que especificar los siguientes parámetros: puerto, baudios, paridad, bits de datos, bits de parada y control de flujo.

Para establecer los parámetros anteriores el usuario dispone de una caja de diálogo que le permitirá establecer los parámetros con los que se realizarán las comunicaciones. La respuesta a esta acción del usuario es la función *OnConfigParam*, que invocará a la función *EstablecerConexión*, si el usuario cerró el cuadro de diálogo mediante el botón “Aceptar”, para abrir el puerto de comunicaciones. Si el proceso se ejecuta satisfactoriamente, ponemos la variable *m_ConexiónEstablecida* al valor TRUE, variable que utilizaremos para identificar si el puerto está o no abierto. Si el usuario cerró el cuadro de diálogo con el botón “Cancelar” no se realiza ninguna acción, en cambio si el usuario pulsa el botón “Restaurar” se establecerán como valores por omisión los últimos valores que fueron almacenados en el registro de Windows.

Existen dos ordenes en el menú de la aplicación; una para abrir el puerto de comunicaciones con los valores de los parámetros actualmente seleccionados, correspondiente a la función *OnConexion-Establecer*, y otra para cerrar el puerto de comunicaciones, vinculada a la función *OnConexionCortar* que guarda la configuración actual en el registro de Windows e invoca a la función *CortarConexión*. Además existen funciones auxiliares que impiden abrir un puerto de comunicaciones cuando ya está abierto o cerrarlo cuando aún no está abierto.

12.6 Enviar y recibir datos

Para enviar los datos el usuario arrancará la aplicación, establecerá las comunicaciones, escribirá el texto a enviar en la caja de “Dato a transmitir” y pulsará el botón “Enviar”, o pulsará directamente el botón “Temperatura” o uno de los botones mutuamente excluyentes de luz que escriben el texto correspondiente en la caja de “Dato a transmitir”. De cualquier forma se invocará a la función *OnEnviar* que enviará la información a la cola de salida invocando a la función *EscribirCarsPuerto* y después limpia la caja de “Dato a transmitir”.

Cuando los caracteres enviados desde la placa central se reciben en la cola de recepción se produce un evento que es capturado por el hilo *ControlarEventos* por medio de la función **WaitCommEvent**. Entonces el hilo envía un mensaje al hilo principal (a la aplicación) notificándole el evento ocurrido en el puerto de comunicaciones. Como respuesta a este mensaje se ejecuta la función *OnEventoCom* que en este caso invoca a la función *LeerCaracteresPuerto* para obtener los datos del puerto, y después a la función *OnVisualizarCars* para visualizarlos en la caja de “Dato recibido” y actualizar el valor de los “displays” de temperatura en el caso de tratarse de un dato de la placa sensora.

En la figura 12.2 página 142 podemos observar el diagrama de flujo general del programa, el código fuente se encuentra en el CDROM que acompaña a esta memoria.

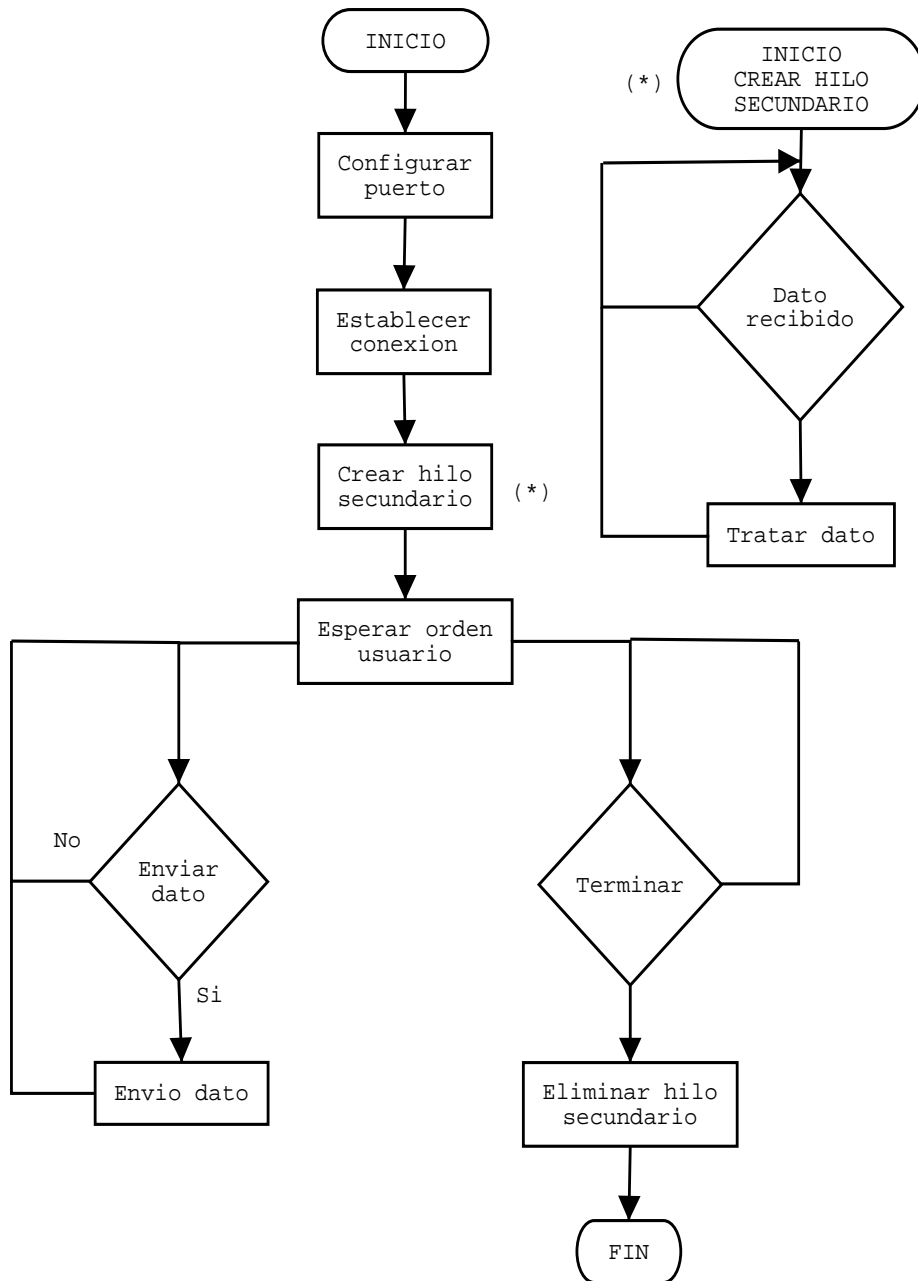


Figura 12.2: Diagrama de flujo del programa Dome.

Capítulo 13

Aspectos económicos

13.1 Introducción

Vamos a tratar los aspectos económicos de la realización de los diferentes prototipos de módem que hemos diseñado. Realizamos el estudio teniendo en cuenta primeramente cada una de las placas prototipo, para después obtener el presupuesto total junto en el programador de los PIC y el programa Dome.

13.2 La placa central

En la tabla 13.1 página 144 podemos observar el coste de cada módulo básico en la placa central, así como su presupuesto total.

13.3 La placa sensora

En la tabla 13.2 página 144 podemos observar el coste de cada módulo básico en la placa sensora, así como su presupuesto total.

	Coste (ptas.)	Coste (euros)
Fuente de alimentación	426	2,56
Emisor de señal	342	2,06
Detector de señal	395	2,37
Microcontrolador	1138	6,84
Interfaz con el PC	127	0,76
Cables, conectores y placa	1352	8,13
Mano de obra	60000	360,61
Tiempo de diseño	680000	4086,88
TOTAL	743780	4470,21

Tabla 13.1: *Coste de la placa central.*

	Coste (ptas.)	Coste (euros)
Fuente de alimentación	426	2,56
Emisor de señal	342	2,06
Detector de señal	395	2,37
Microcontrolador	1138	6,84
Sensor de temperatura	178	1,70
Cables, conectores y placa	1052	6,32
Mano de obra	50000	300,51
Tiempo de diseño	590000	3545,97
TOTAL	643531	3867,70

Tabla 13.2: *Coste de la placa sensora.*

13.4 La placa actuadora

En la tabla 13.3 página 145 podemos observar el coste de cada módulo básico en la placa actuadora, así como su presupuesto total.

	Coste (ptas.)	Coste (euros)
Fuente de alimentación	426	2,56
Emisor de señal	342	2,06
Detector de señal	395	2,37
Microcontrolador	1138	6,84
Regulador de luz	1183	7,11
Cables, conectores y placa	1410	8,47
Mano de obra	55000	330,56
Tiempo de diseño	650000	3906,58
TOTAL	709894	4266,55

Tabla 13.3: *Coste de la placa actuadora.*

13.5 Conclusiones

El coste total del proyecto incluyendo cada placa así como el programador de los PIC y el desarrollo de la aplicación Dome se puede ver en la tabla 13.4 página 145.

	Coste (ptas.)	Coste (euros)
Placa central	743780	4470,21
Placa sensora	643531	3867,70
Placa actuadora	709894	4266,55
Programador	21360	128,38
Dome	380000	2283,85
TOTAL	2498565	15016,68

Tabla 13.4: *Coste de la placa actuadora.*

Apéndice A

Perspectivas futuras

A.1 Mejoras

- En la interfaz entre la placa central y el PC existe una limitación en la velocidad de transmisión de datos, el optoacoplador que hemos utilizado no nos da una buena respuesta por encima de tasas de transmisión de $1200k\text{bps}$. Por tanto se conseguiría aumentar la tasa de transmisión sin más que buscar un optoacoplador con mejores prestaciones.
- Una clara e inevitable mejora del sistema, si se quiere desarrollar más funcionalidad y velocidad, es la aplicación de algún tipo de algoritmo de corrección de errores para reducir en lo posible los efectos del ruido presente en la red eléctrica. Ya pensando en esta necesidad hemos dejado con toda la intención el último byte de las tramas enviadas para usarlo como corrección de errores o incluso si se quiere se puede mandar directamente como información redundante.
- En el capítulo de codificación de las señales hemos comentado que en la comunicación por la red eléctrica todos escuchan a todos. Dada esta situación hemos tenido que asignar una dirección única a cada uno de los dispositivos conectados a

la red. Esta dirección se asigna en tiempo de grabación del microcontrolador por medio del software de grabación. Ahora bien, no debemos olvidar que el microcontrolador puede escribir en su memoria E^2PROM en tiempo de ejecución lo que permitiría olvidarnos de la asignación previa de direcciones y podríamos asignarlas, por ejemplo, desde el PC usando Dome.

- Aprovechando la idea anterior se pueden dar unos pasos más y tratar de obtener un sistema automático de asignación de direcciones. La idea podría consistir en lo siguiente: primero todos los dispositivos tienen su dirección con valor cero, desde Dome se asigna una dirección a la placa central que la almacena en su E^2PROM , a partir de aquí cada vez que se enchufa un dispositivo nuevo en la red envía una trama avisando al PIC de la placa central y después al PC, y desde Dome se asigna una dirección nueva al dispositivo y se envía para que la almacene en su E^2PROM . De esta manera se puede conseguir un sistema parecido al plug & pray de Windows.
- Pero no debemos olvidar nunca una cuestión, si ampliamos las funcionalidades del sistema de forma que las placas esclavas puedan enviar algún tipo de información sin que la placa central se lo haya pedido entonces debemos desarrollar algún tipo de detección de portadora para evitar colisiones. En definitiva añadir un testeo de la línea durante un tiempo no menor a un bit antes de enviar nada por la red.
- Otra posible mejora es la utilización de un temporizador Watchdog. Se trata de un temporizador con funcionamiento totalmente en paralelo con el μC que lo reinicializa si el programa no lo refresca a tiempo. De esta forma se dota al sistema de un método de recuperación del control del programa en caso de que se produzca un fallo de funcionamiento.

A.2 Problemas

- Debido a usar una fuente de alimentación sin transformador no tenemos aislamiento eléctrico entre la red eléctrica y nuestro circuito electrónico. Este es un grave problema que nos obliga a realizar este aislamiento ante el uso de cualquier dispositivo externo a nuestras placas que esté enchufado a la red eléctrica. Así para usar un osciloscopio nos vimos obligados a aplicar la sonda por medio de unos transformadores. El mismo problema tuvimos al enchufar nuestra placa central con el puerto serie del ordenador por lo que tuvimos que usar unos optoacopladores para solucionar el problema. También hemos usado multímetros para testear nuestros circuitos pero estos no suponen un problema porque o bien están internamente aislados o son de batería.
- El cambio de temperatura afecta directamente a nuestros circuitos de una manera dramática, se produce un desplazamiento de la frecuencia emitida por el CI LM555 así como un desplazamiento en el rango de frecuencias que detecta el CI NE567. Esto puede producir que el sistema pueda dejar de funcionar ante un foco de calor cercano intenso como una lámpara.
- Cuidado con los potenciómetros de ajuste porque aunque los módulos de transmisión y recepción con la red son análogos en cada una de las placas, no lo son el resto del circuito que sí afecta al funcionamiento de los circuitos integrados CI LM555 y NE567.
- Como toda la potencia que suministra nuestra fuente de alimentación la obtenemos de dos zener, la potencia que va a ser capaz de suministrar la fuente va a estar limitada por la corriente máxima que soportan dichos componentes.

Apéndice B

Dome: Manual de usuario

B.1 Primeros pasos

El presente manual describe de manera somera las prestaciones de la aplicación Dome, que como sabemos implementa la comunicación con la placa central mediante una interconexión RS232. Es una aplicación basada en la API Win32 y debería funcionar correctamente en Windows 9X y Windows ME.

Una vez iniciada la aplicación, se cargarán los valores de los parámetros de configuración almacenados en el registro de Windows y aparecerá un cuadro de diálogo como el que podemos observar en la figura B.1 página 152. En él disponemos de los menús y submenús siguientes:

- Conexión:
 - Establecer: Inicia una conexión con un puerto usando los parámetros por defecto.
 - Cortar: Termina una conexión previamente inicializada, liberando el puerto.
 - Salir: Finaliza la conexión actual, si existe, almacena los parámetros actuales de configuración en el registro de

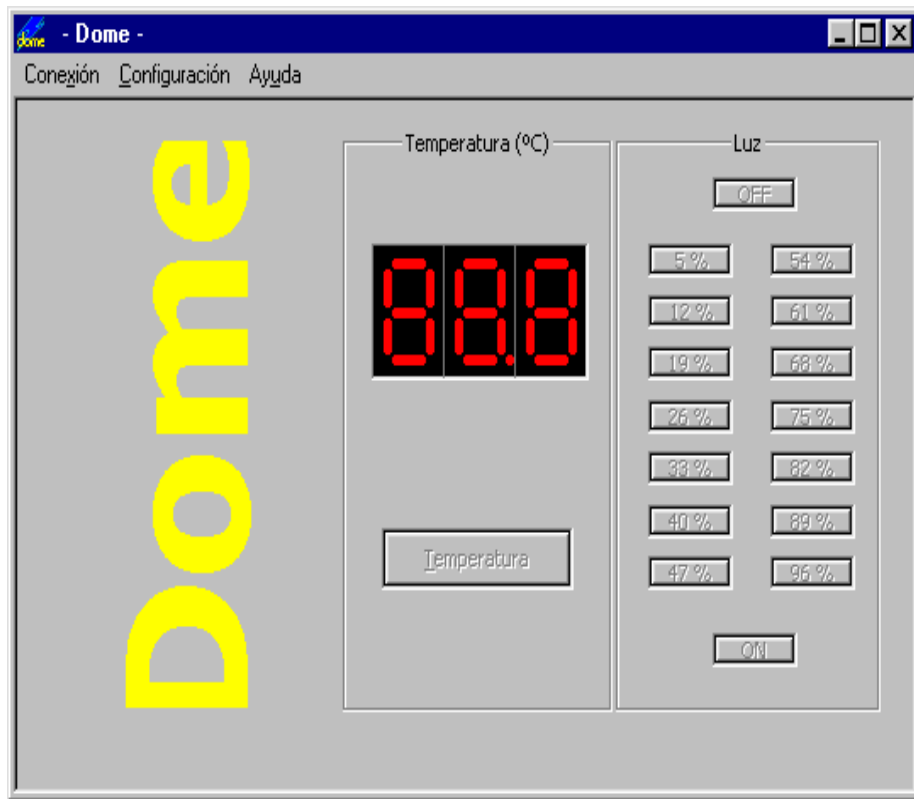


Figura B.1: *Ventana principal de la aplicación Dome.*

Windows y termina el programa.

- Configuración:
 - Parámetros COM: Muestra un cuadro de dialogo donde se pueden cambiar los parámetros de configuración e iniciar una conexión con el puerto, así como recuperar los últimos parámetros que se almacenaron en el registro de Windows e iniciar una conexión con el puerto.
- Ayuda:
 - Acerca de Dome: Muestra un cuadro de diálogo con información acerca de los autores de Dome, como podemos ver en la figura B.2 página 153.

- Debug/Release: Permite intercambiar entre los dos modos de funcionamiento básicos del programa; el modo Debug para depuración y el modo Release para el funcionamiento normal. Ver la figura B.3 página 153.

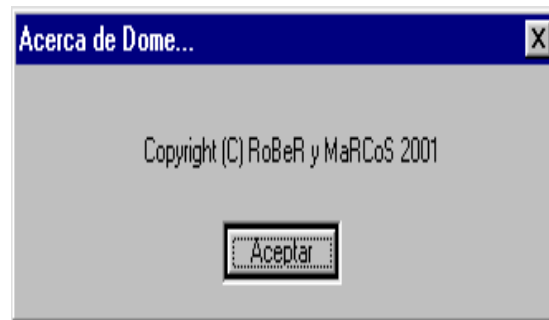


Figura B.2: *Cuadro de diálogo “Acerca de Dome...”*.

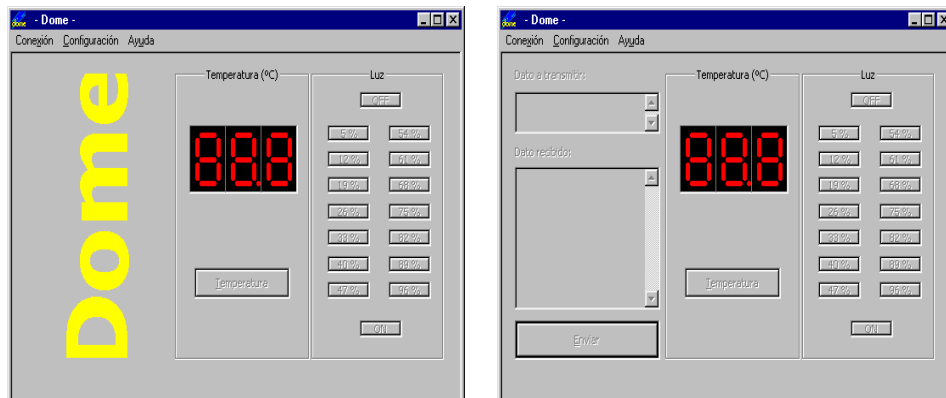


Figura B.3: *Modo Release y modo Debug*.

B.2 Conexión con un puerto

Existen dos formas distintas de conectar con un puerto:

1. Mediante el menú “Establecer” dentro del menú “Conexión” iniciamos la conexión con un puerto usando los parámetros

por defecto. Si la conexión tiene éxito se habilitarán los controles en el cuadro de diálogo, ver la figura B.4 página 154, si por el contrario no es posible realizar la conexión por algún motivo, por ejemplo porque el puerto está siendo usado por otra aplicación, nos informa del error ocurrido.

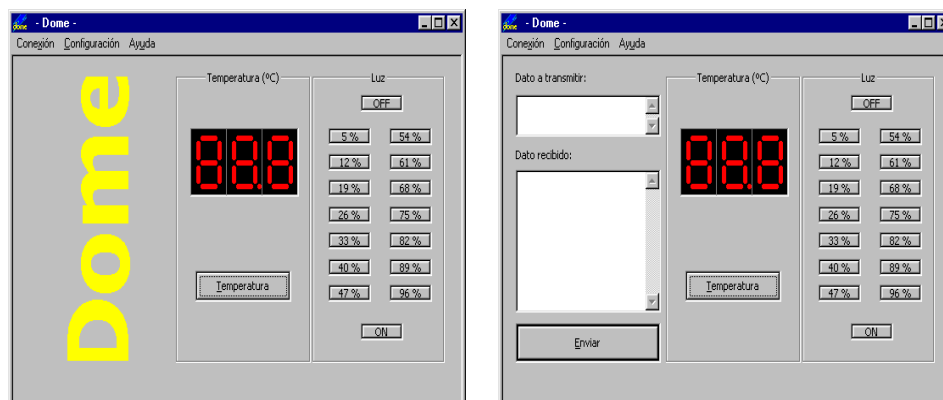


Figura B.4: *Habilitación de los controles en el cuadro de diálogo al abrir el puerto.*

2. Mediante el cuadro de diálogo “Configuración” que aparece en el figura B.5 página 155, donde podemos especificar los siguientes parámetros: puerto, baudios, paridad, bits de datos, bits de parada y control de flujo, dentro de los valores que se pueden ver en la tabla B.1 página 155. Este cuadro de dialogo posee tres botones:

- Aceptar: Cierra el cuadro de diálogo e inicia la conexión con un puerto usando los parámetros especificados. Si la operación tiene éxito se habilitarán los controles en el cuadro de diálogo, y se informa mediante el cuadro de diálogo que podemos ver en la figura B.6 página 156, si por el contrario no es posible realizar la conexión por algún motivo, nos informa del error ocurrido.
- Restaurar: Recupera los últimos parámetros que se al-

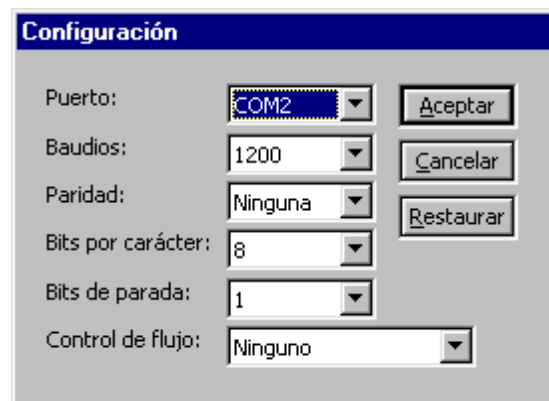


Figura B.5: Cuadro de diálogo “Configuración”.

macenaron en el registro de Windows.

- Cancelar: Cierra el cuadro de diálogo sin realizar ninguna acción.

Puerto	Baudios	Paridad	Datos	Parada	Control flujo
COM1	110	Ninguna	4	1	Ninguno
COM2	300	Par	5	1, 5	Xon/Xoff
COM3	600	Impar	6	2	Hardware(DTR/DSR)
COM4	1200	Marca	7		Hardware(RTS/CTS)
	2400	Espacio	8		
	4800				
	9600				
	14400				
	19200				
	38400				
	56000				
	128000				
	256000				

Tabla B.1: Valores de los parámetros de configuración.



Figura B.6: Cuadro de diálogo de información.

B.3 Comunicación con la placa central

Disponemos de tres posibles tipos de acciones:

- Petición de temperatura: mediante el botón “Temperatura” solicitamos el valor de la temperatura actual a la placa central, la cual reenvía la petición a la placa sensora y nos remite la contestación de esta, que aparecerá reflejada en los “displays”. Ver la figura B.7 página 156.

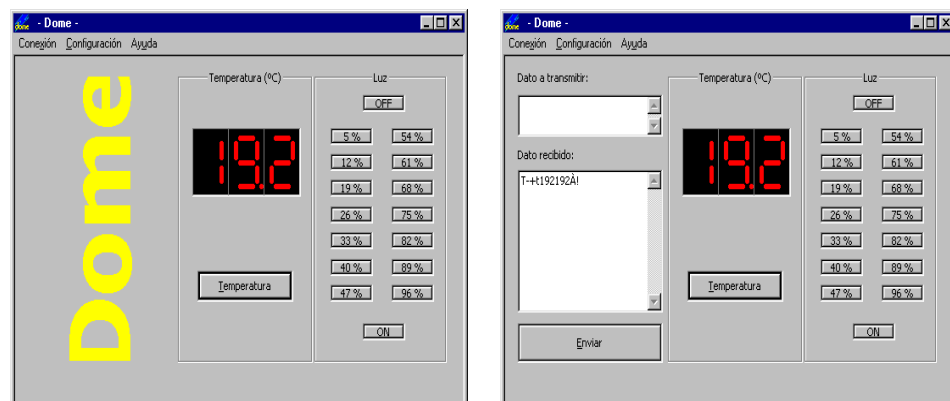


Figura B.7: Petición de temperatura a la placa central.

- Envío de una intensidad de luz: mediante alguno de los botones de luz enviamos el valor de la intensidad de luz a la placa cen-

tral, la cual se lo reenvía a la placa actuadora y nos remite la contestación de esta. Ver la figura B.8 página 157.

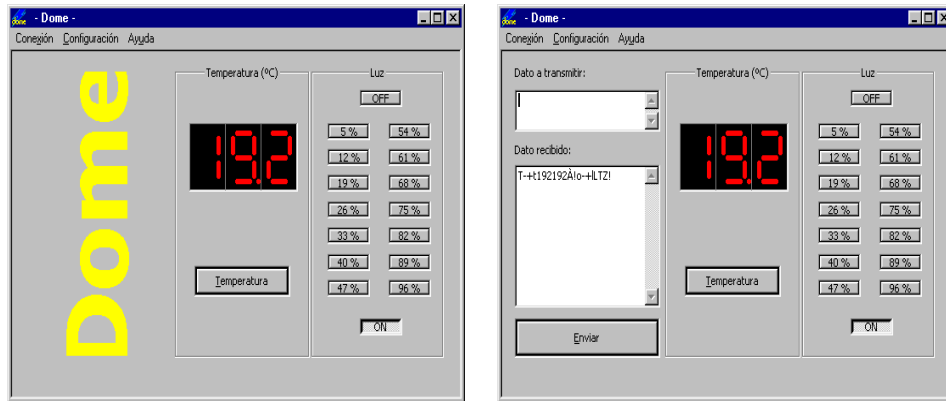


Figura B.8: *Envío de una intensidad de luz a la placa central.*

- Envío de un comando de depuración: mediante el botón “Enviar” transmitimos el dato introducido previamente en la caja de “Dato a transmitir” a la placa central, es evidente, que esta acción únicamente se puede realizar en modo Debug. Ver la figura B.9 página 157.

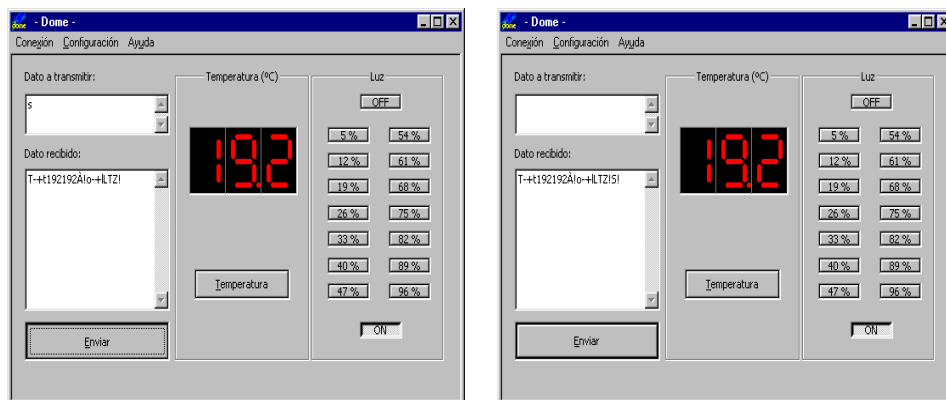


Figura B.9: *Envío de un comando de depuración a la placa central.*

Apéndice C

Esquemas

C.1 La placa central

C.2 La placa sensora

C.3 La placa actuadora

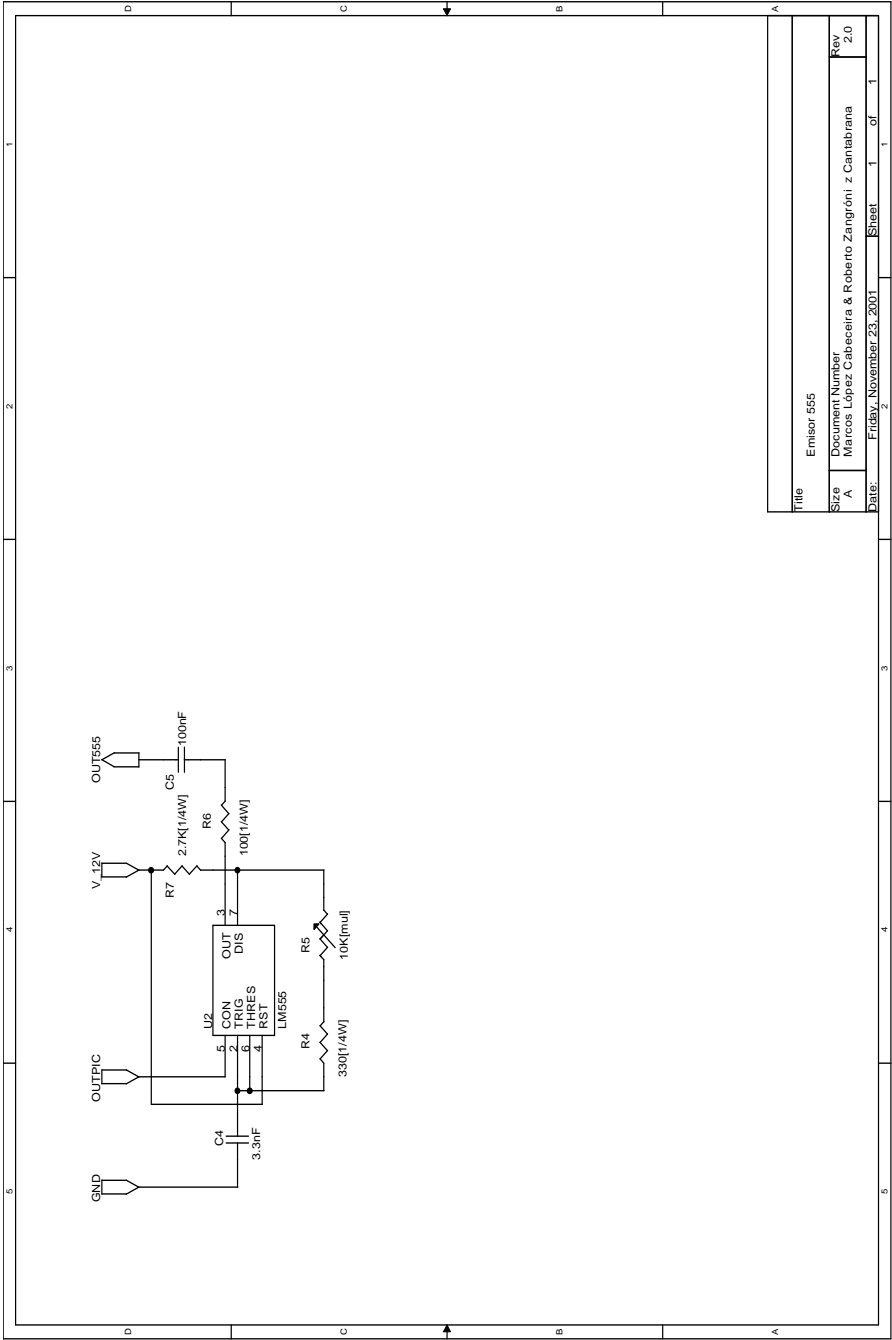


Figura C.1: Esquema del emisor de señal.

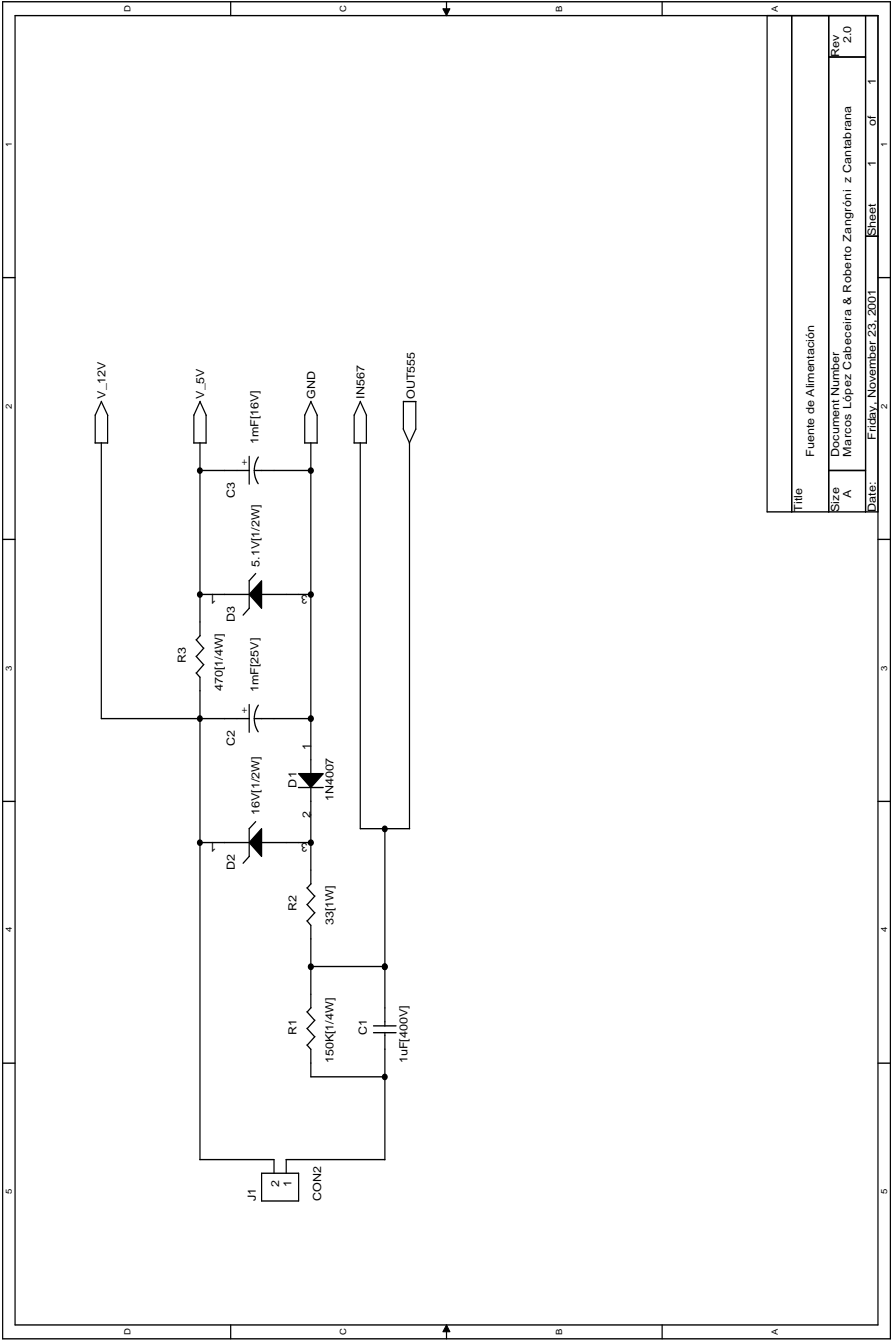


Figura C.2: Esquema de la fuente de alimentación.

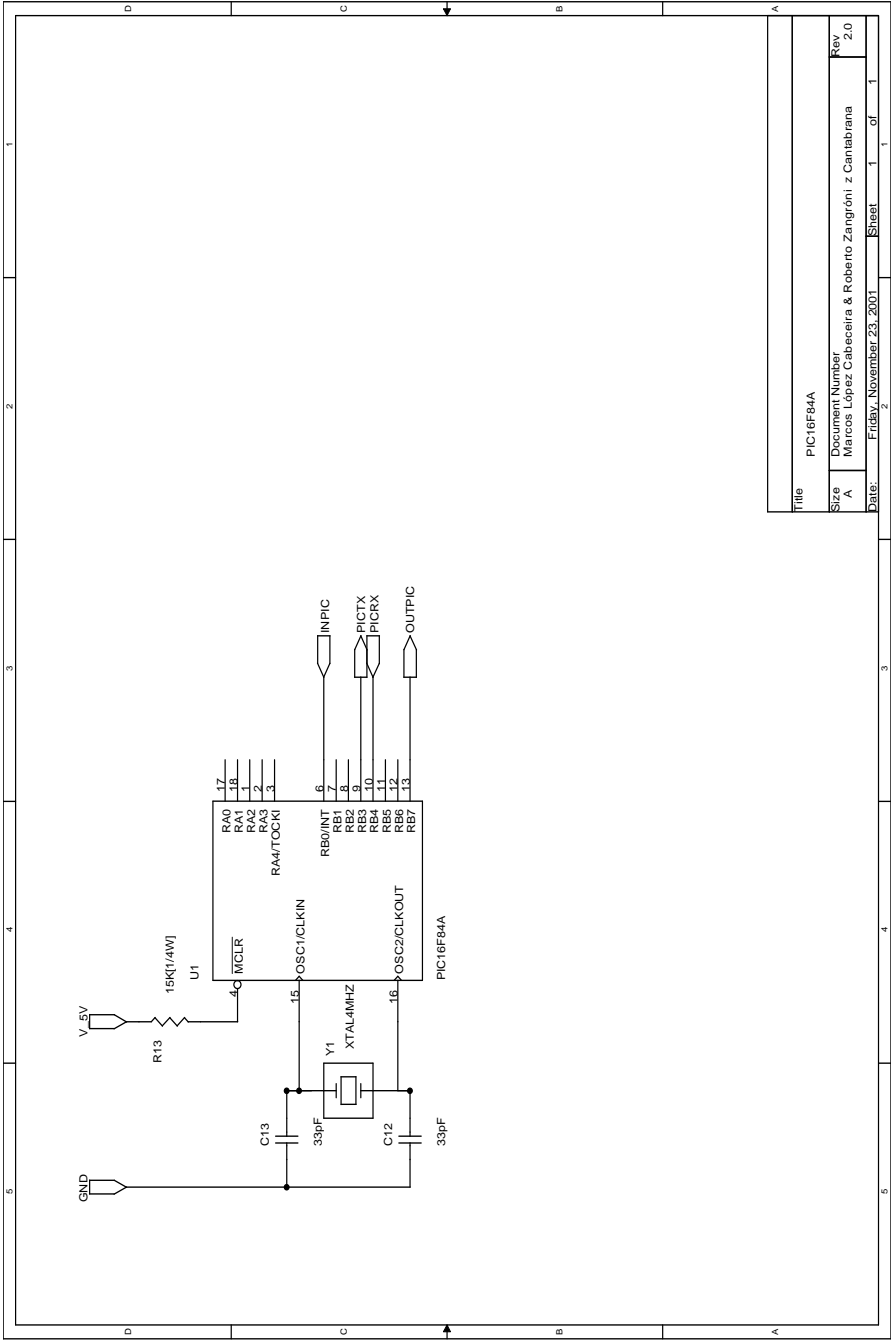


Figura C.3: Esquema del módulo controlador.

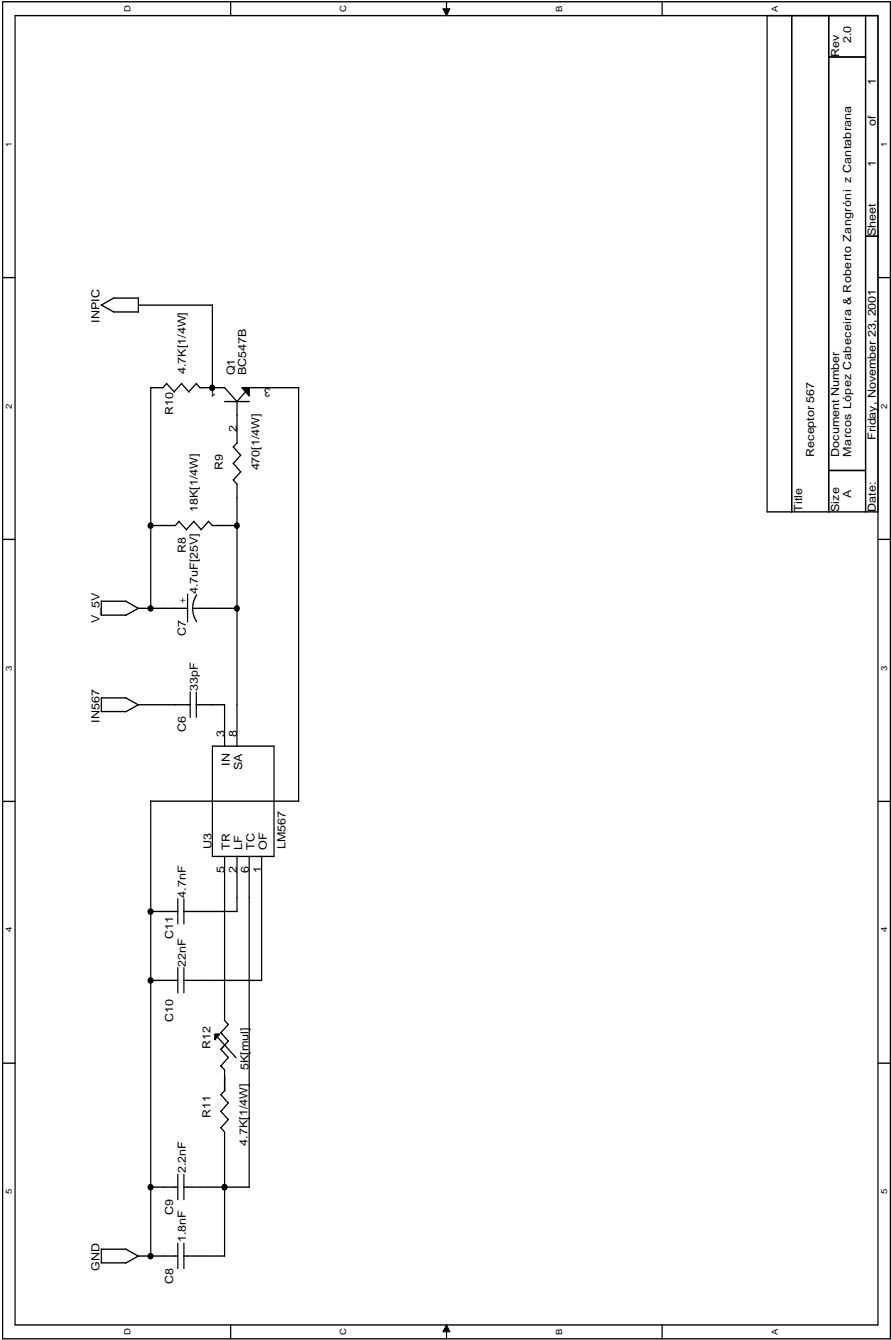


Figura C.4: Esquema del detector de señal.

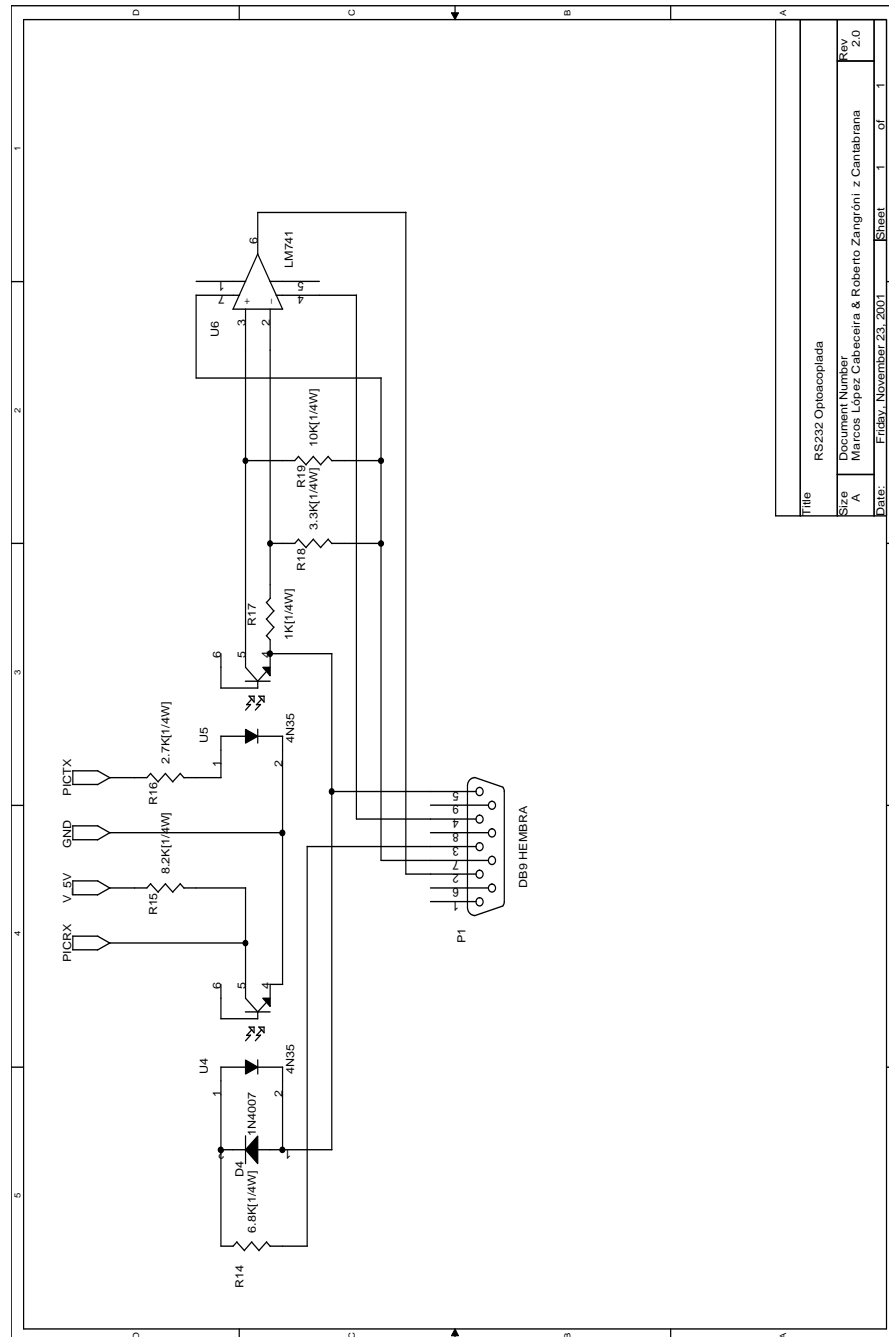


Figura C.5: *Esquema de la interface con el PC.*

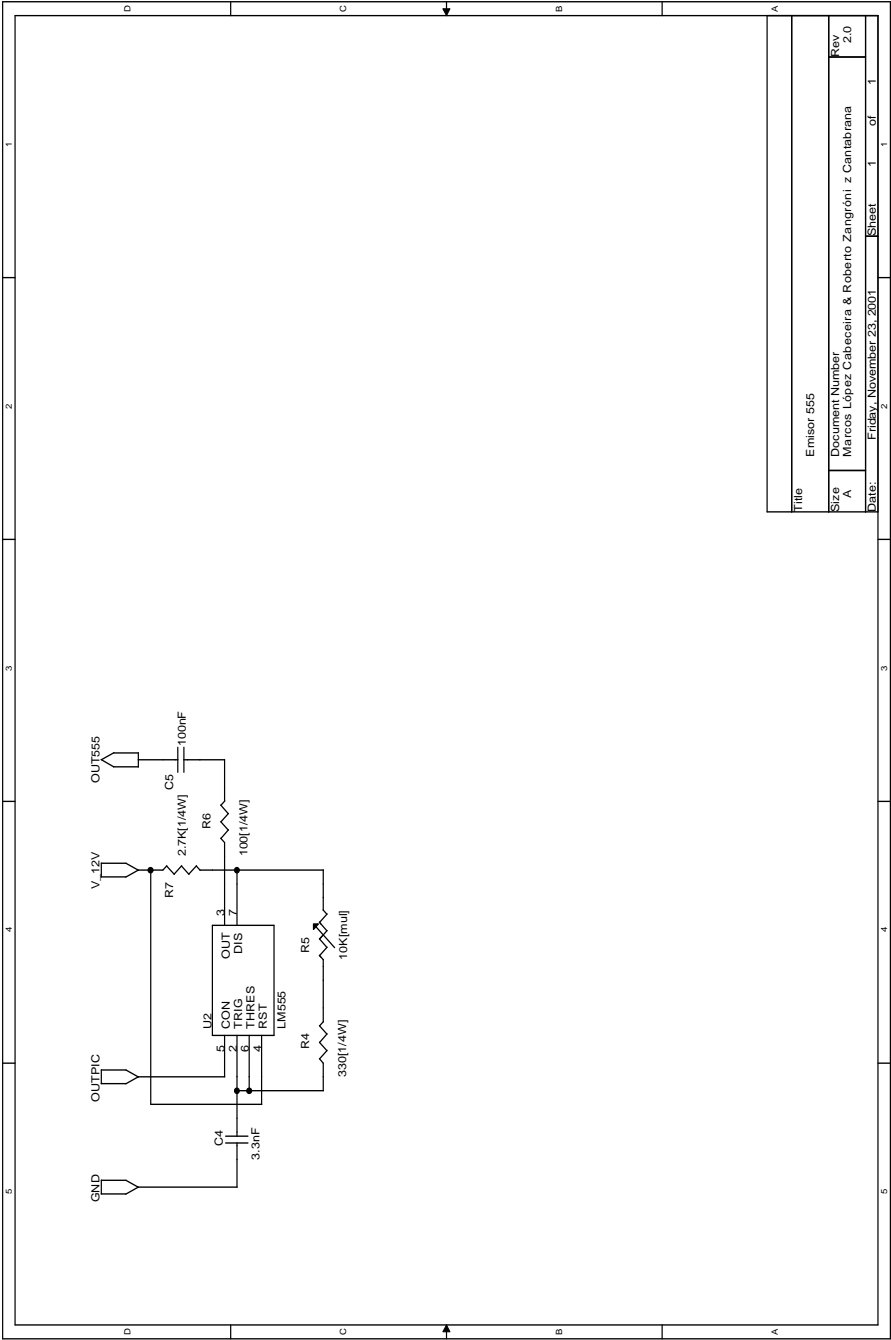


Figura C.6: Esquema del emisor de señal.

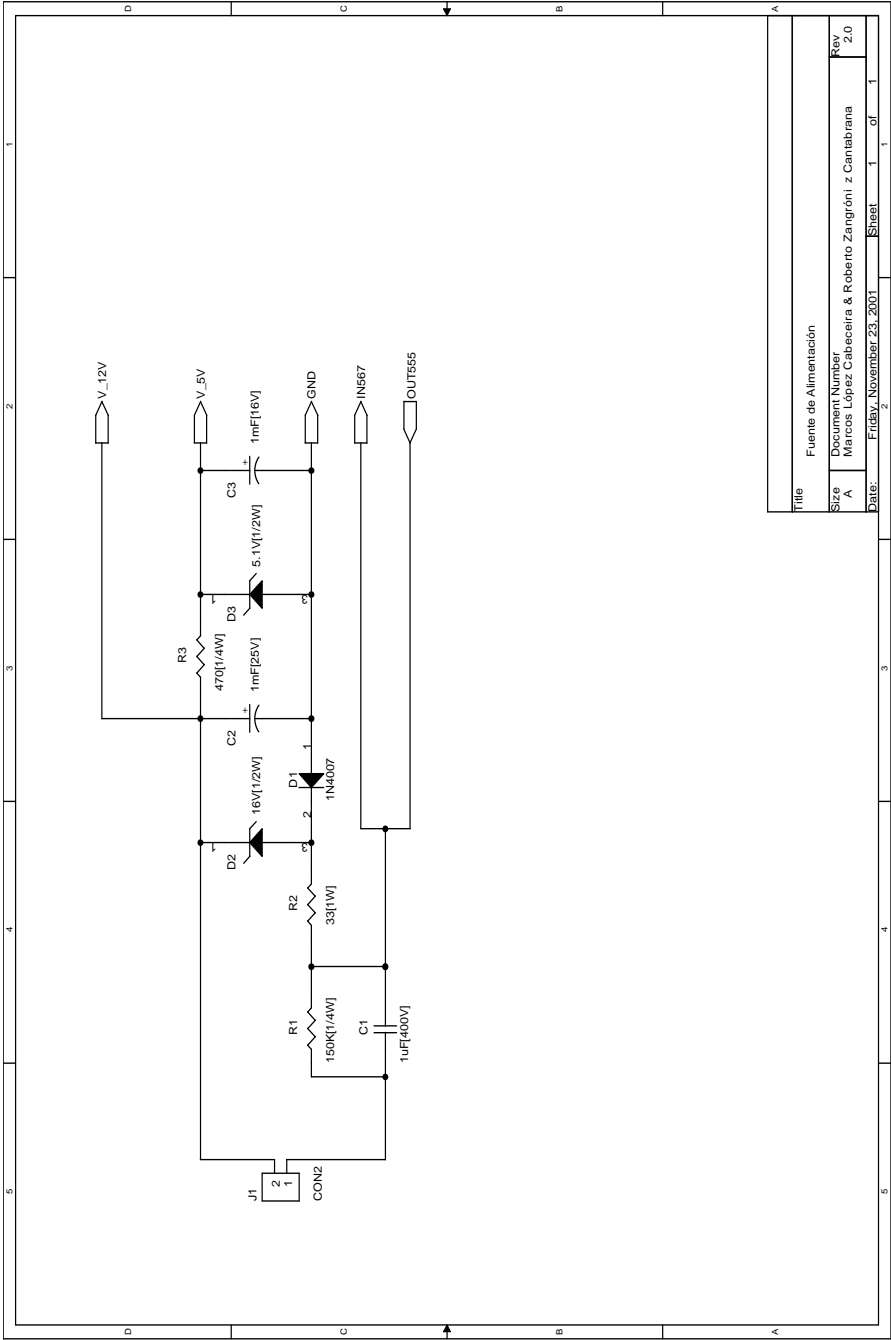


Figura C.7: Esquema de la fuente de alimentación.

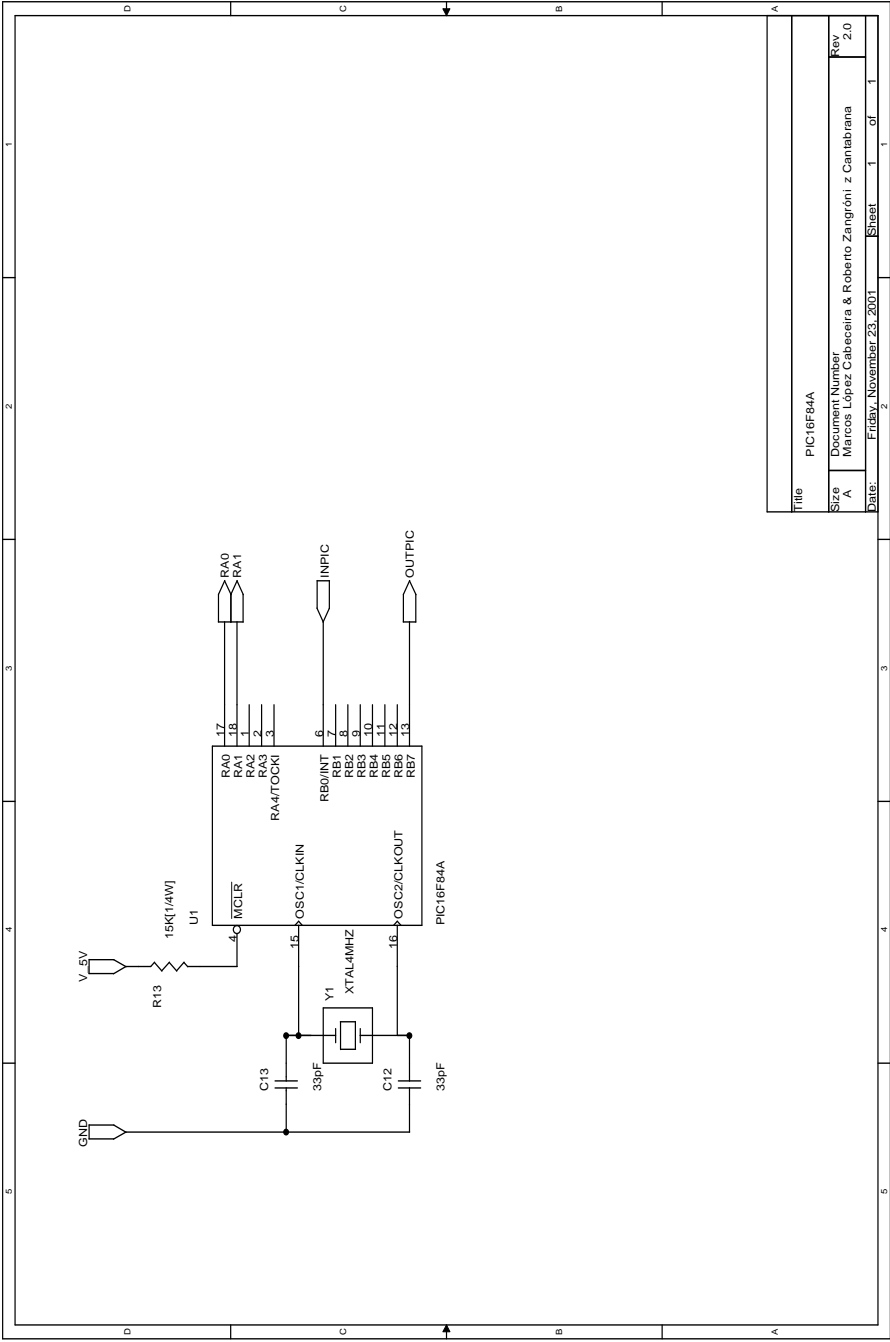


Figura C.8: Esquema del módulo controlador.

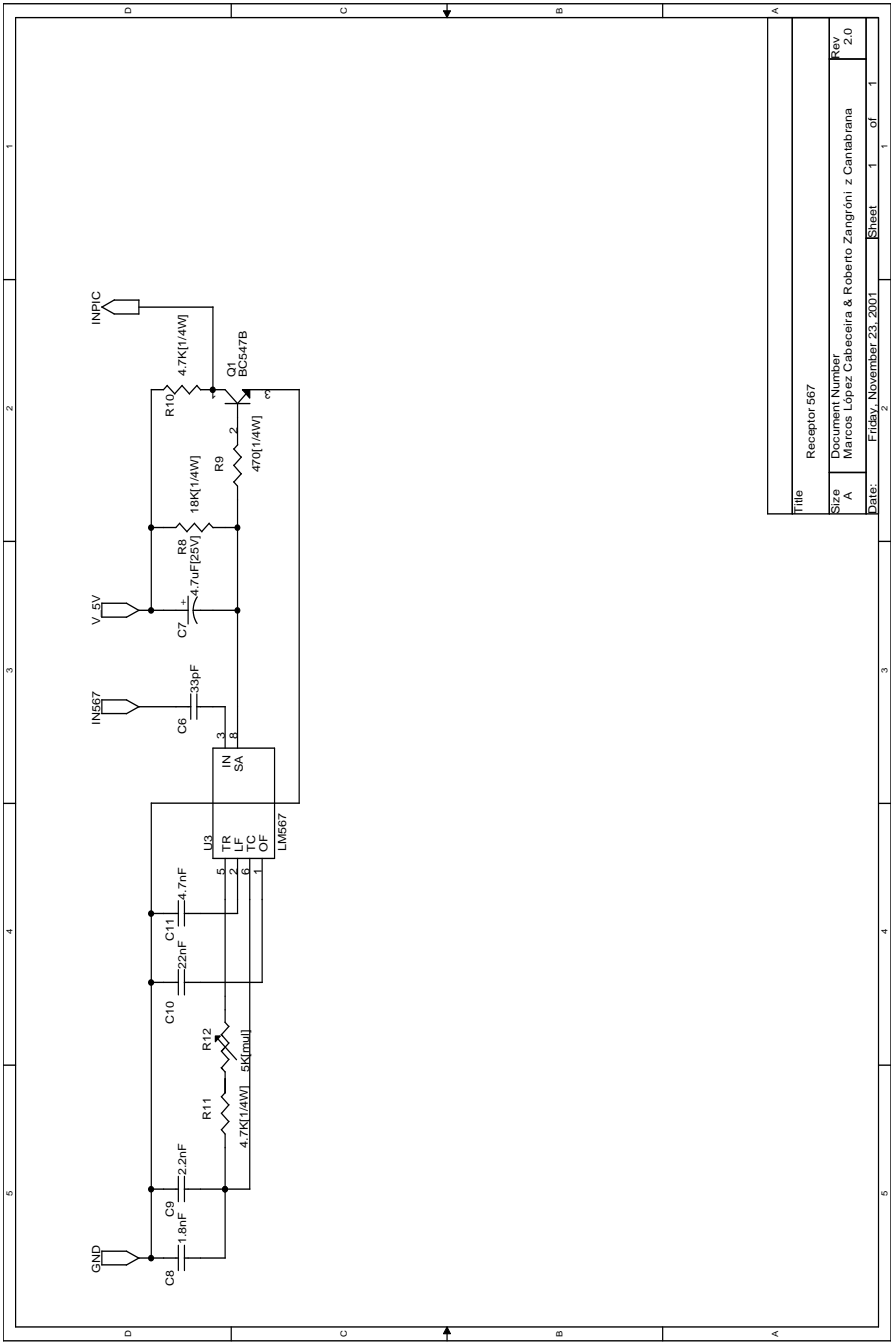


Figura C.9: Esquema del detector de señal.

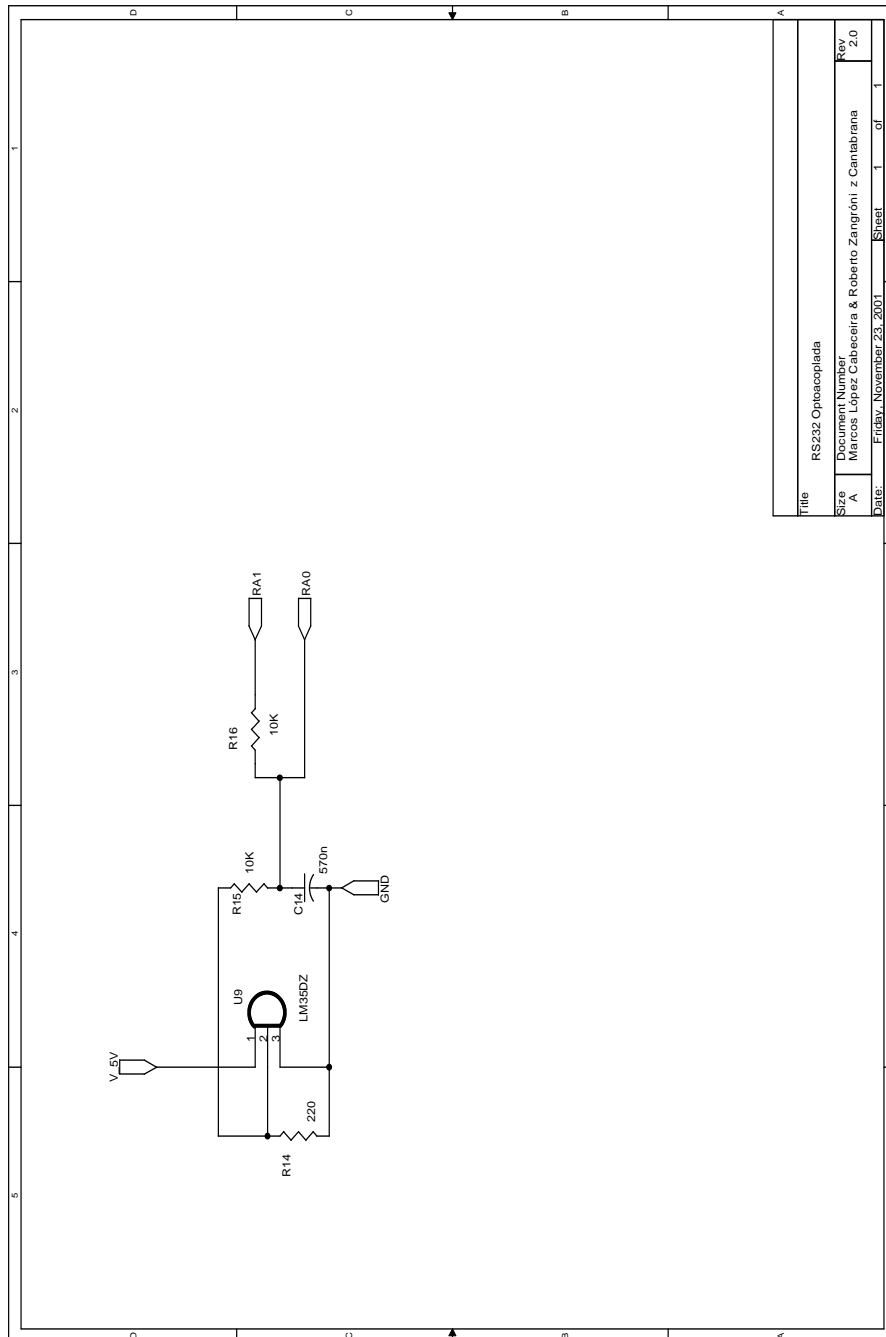


Figura C.10: *Esquema del sensor de temperatura.*

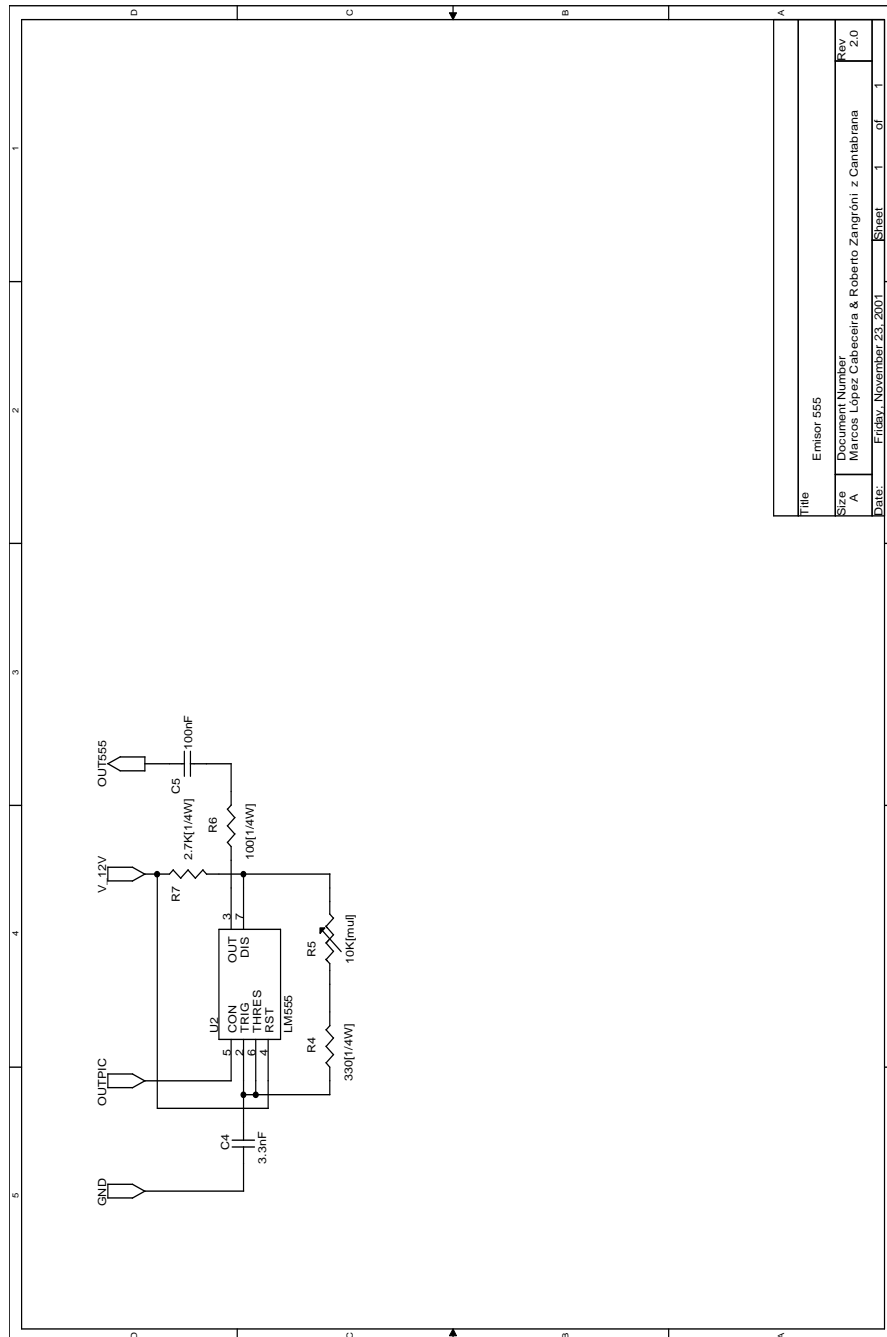


Figura C.11: *Esquema del emisor de señal.*

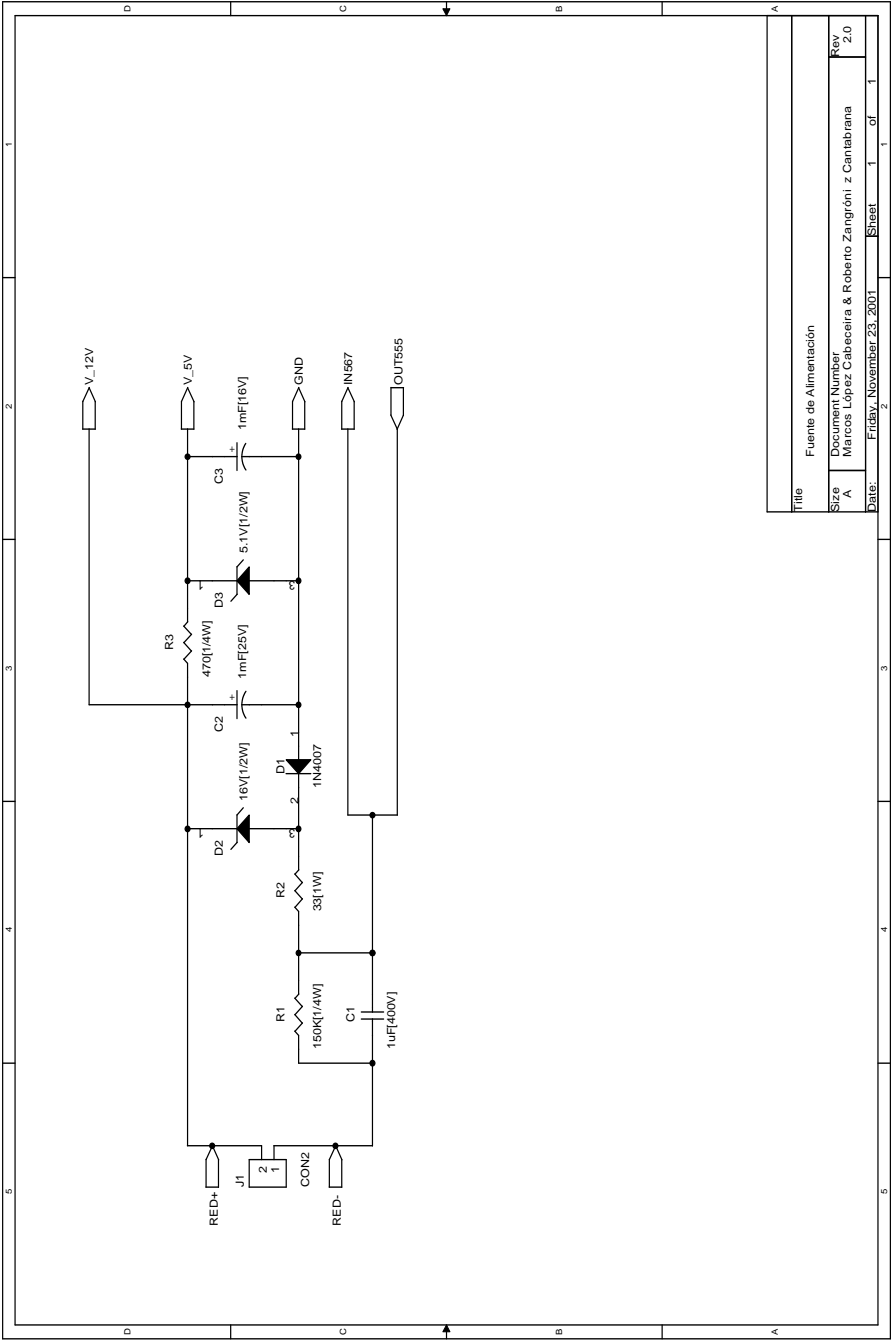


Figura C.12: Esquema de la fuente de alimentación.

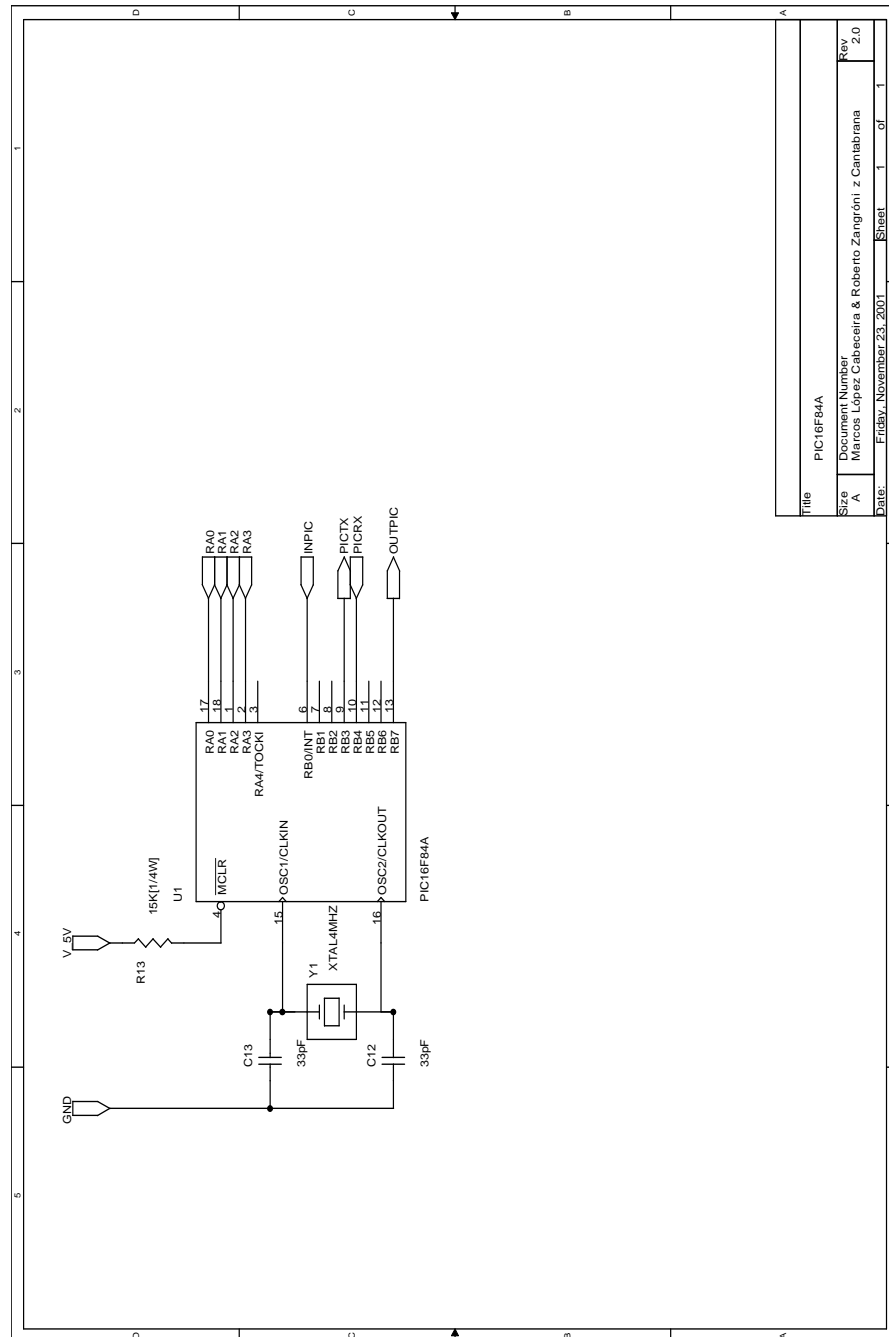


Figura C.13: *Esquema del módulo controlador.*

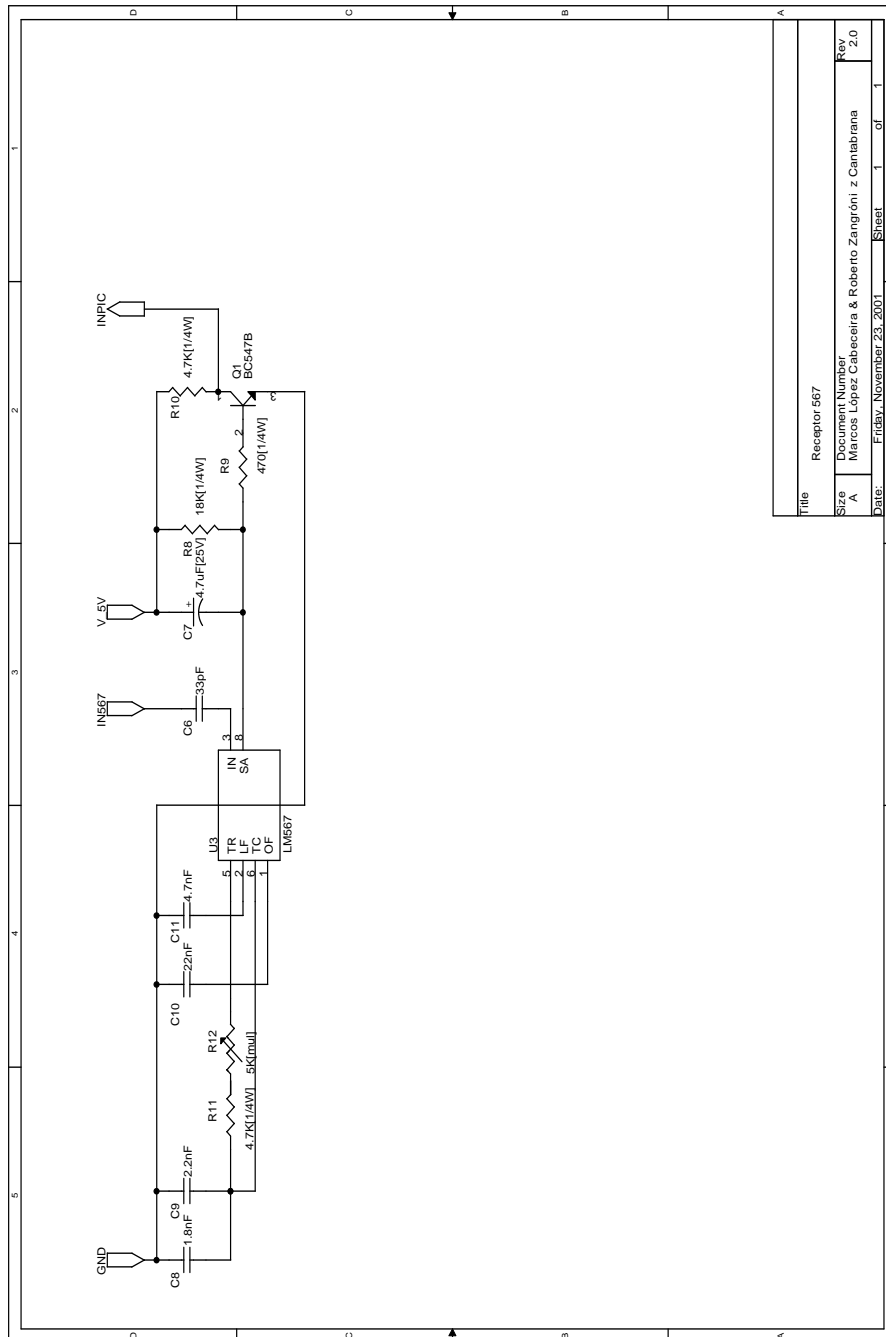


Figura C.14: *Esquema del detector de señal.*

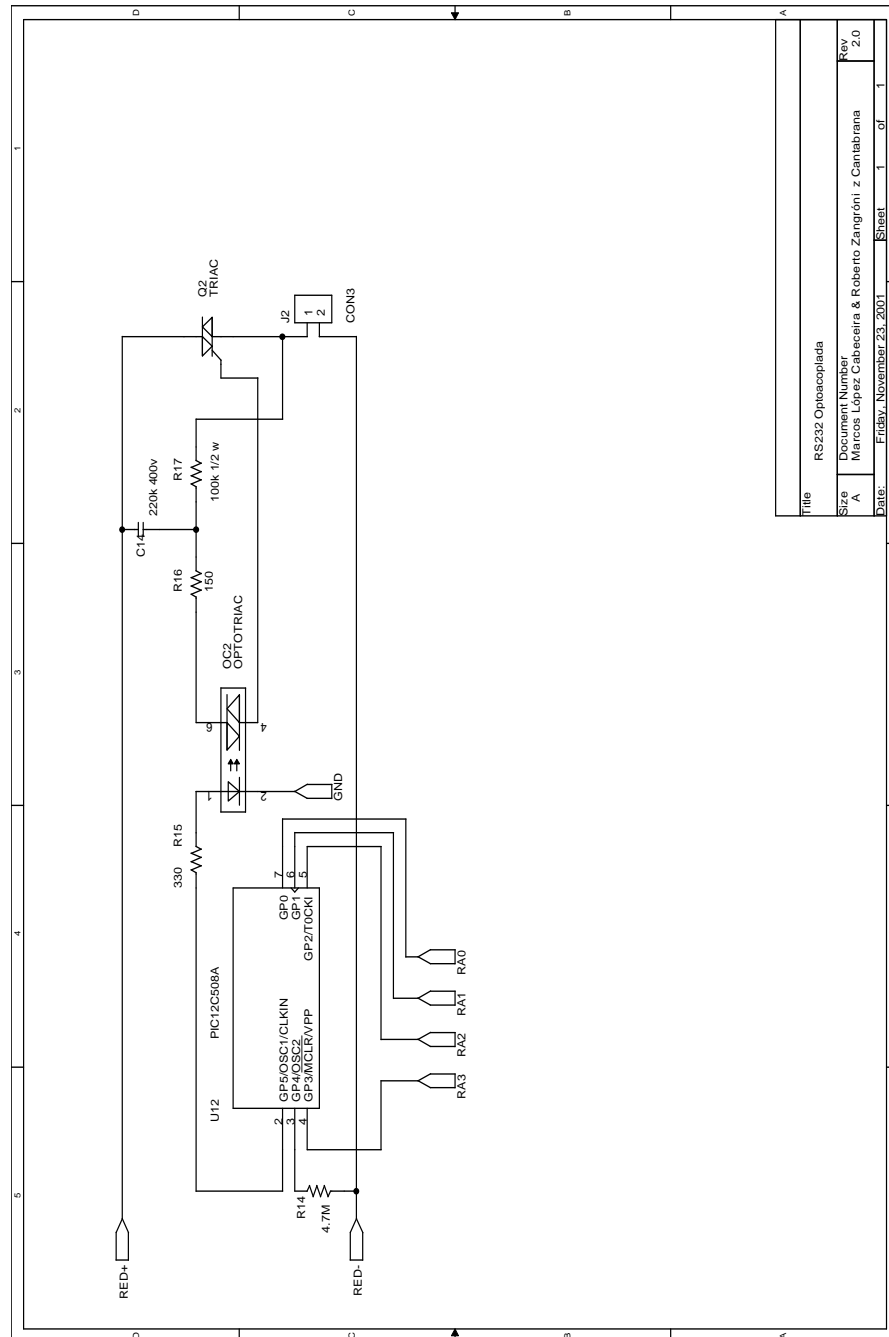


Figura C.15: *Esquema del regulador de luz.*

Apéndice D

Prototipos

D.1 La placa central

D.2 La placa sensora

D.3 La placa actuadora

D.4 El programador del PIC 16F84A

D.5 Una visión general

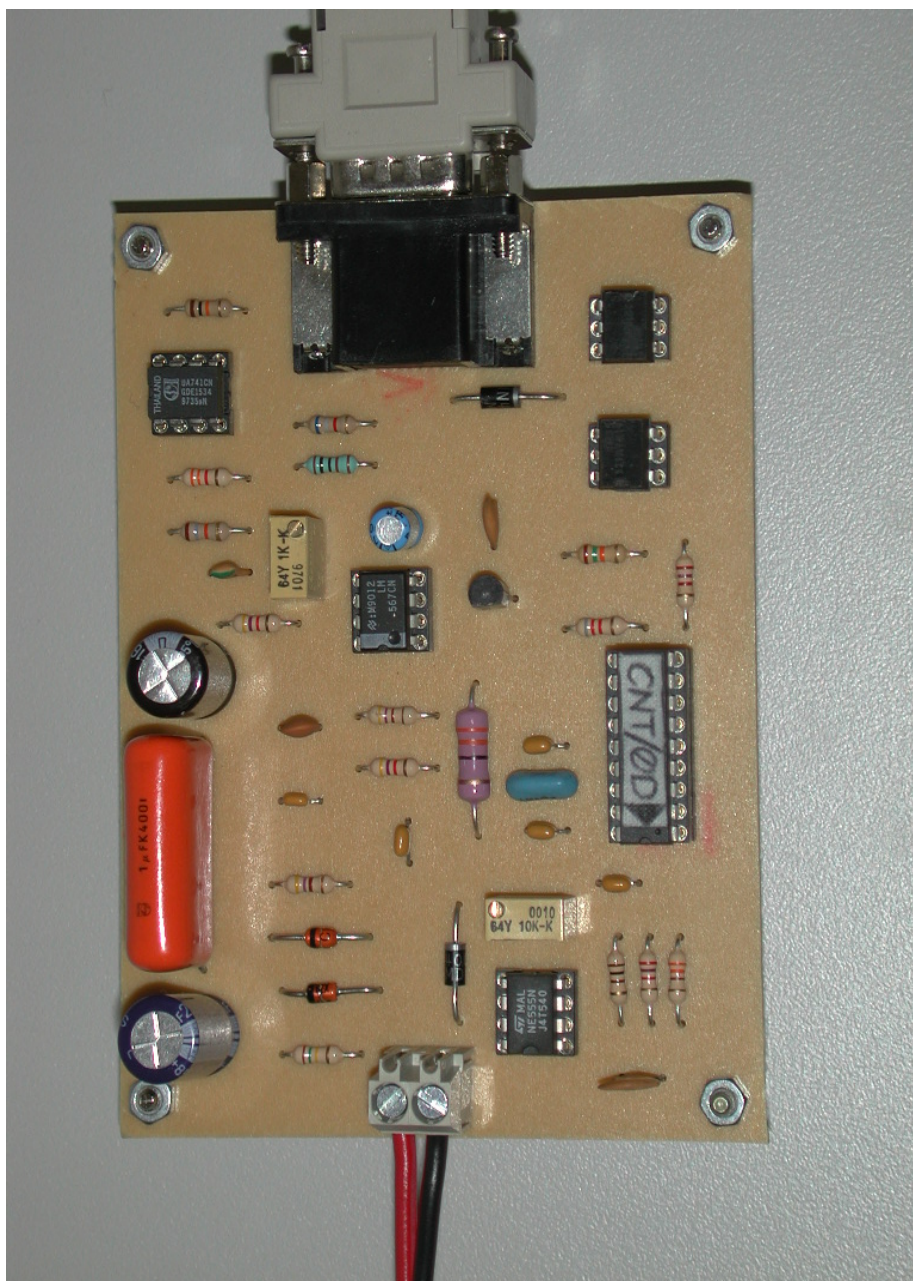


Figura D.1: *Vista superior de la placa central.*

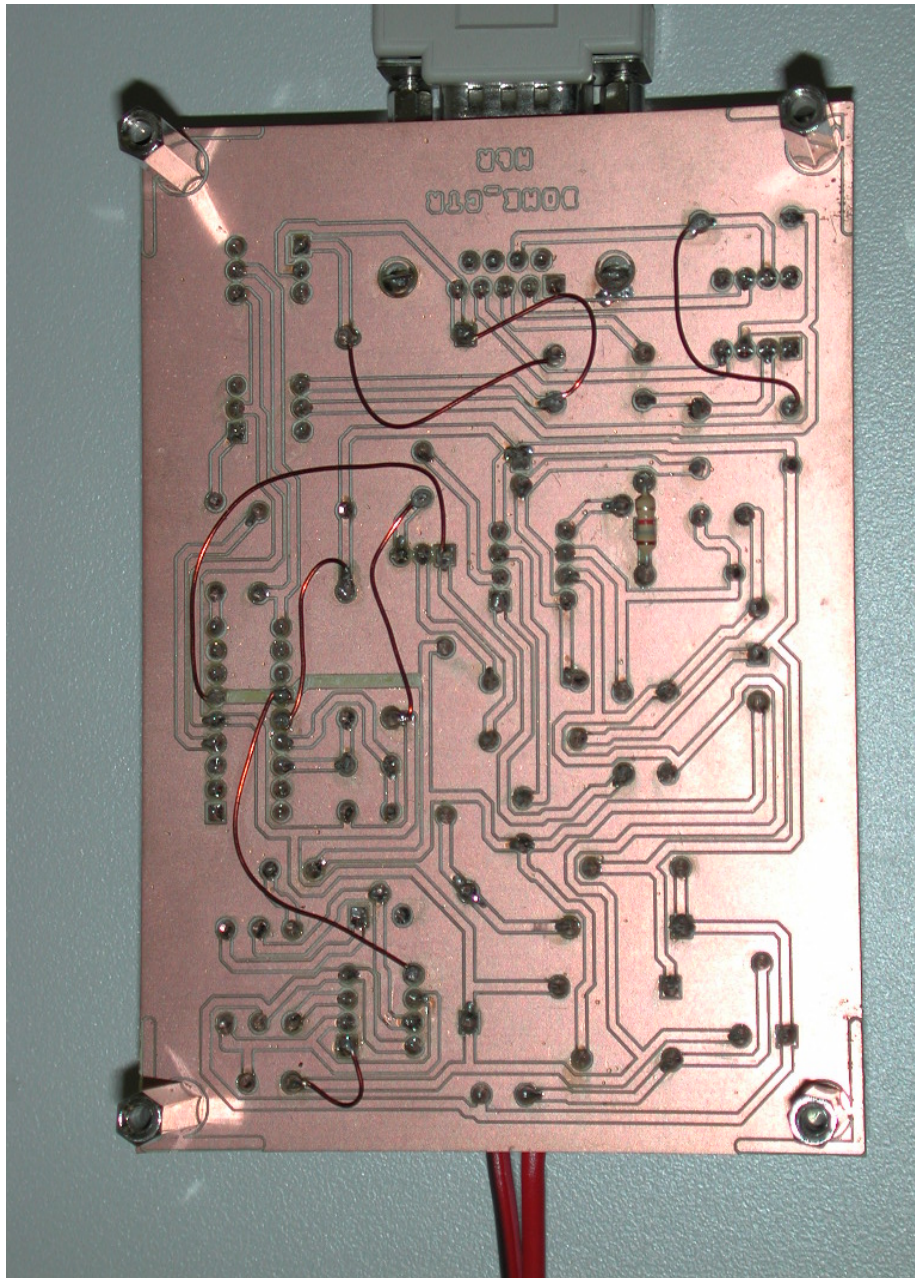


Figura D.2: *Vista inferior de la placa central.*



Figura D.3: Vista superior de la placa sensora.

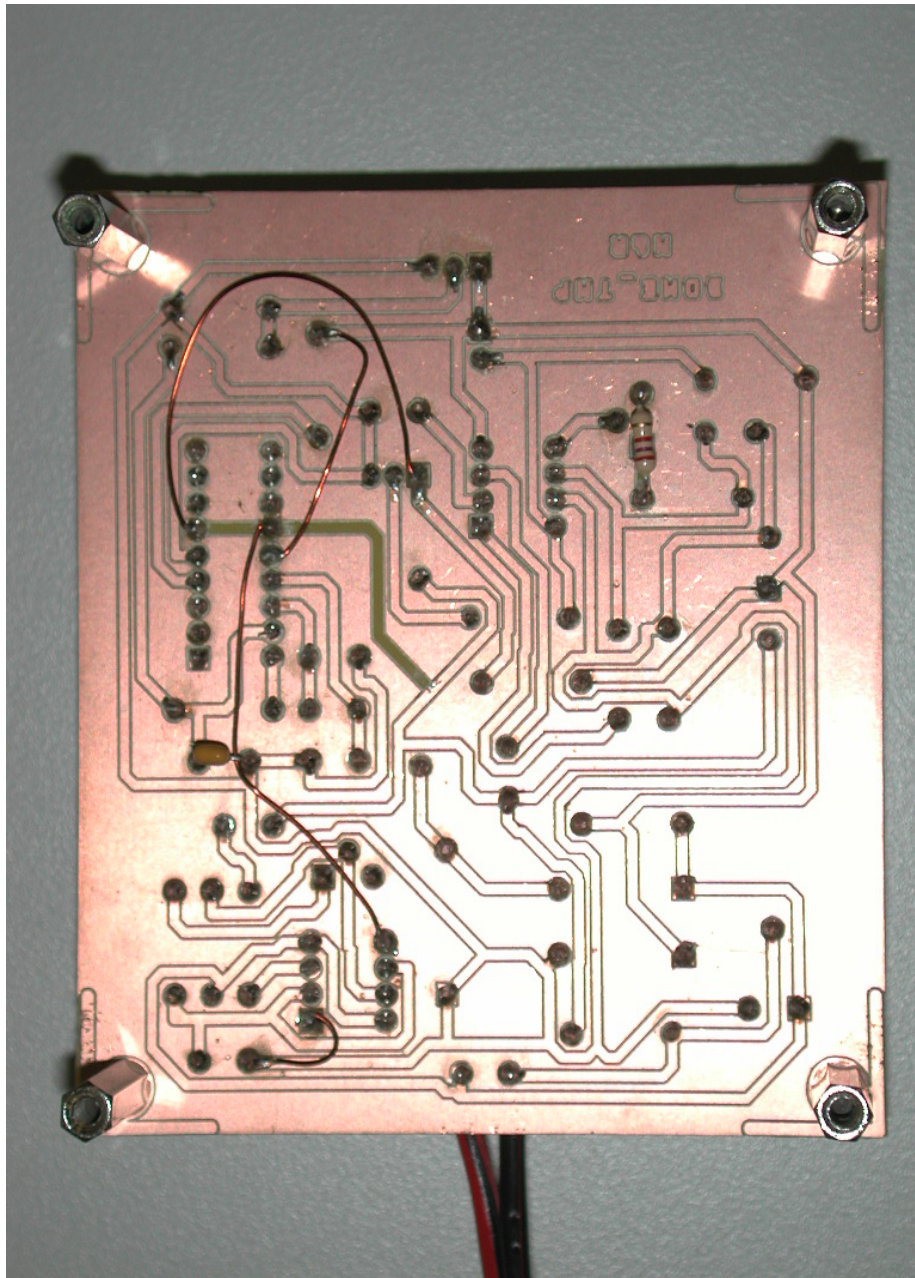


Figura D.4: *Vista inferior de la placa sensora.*

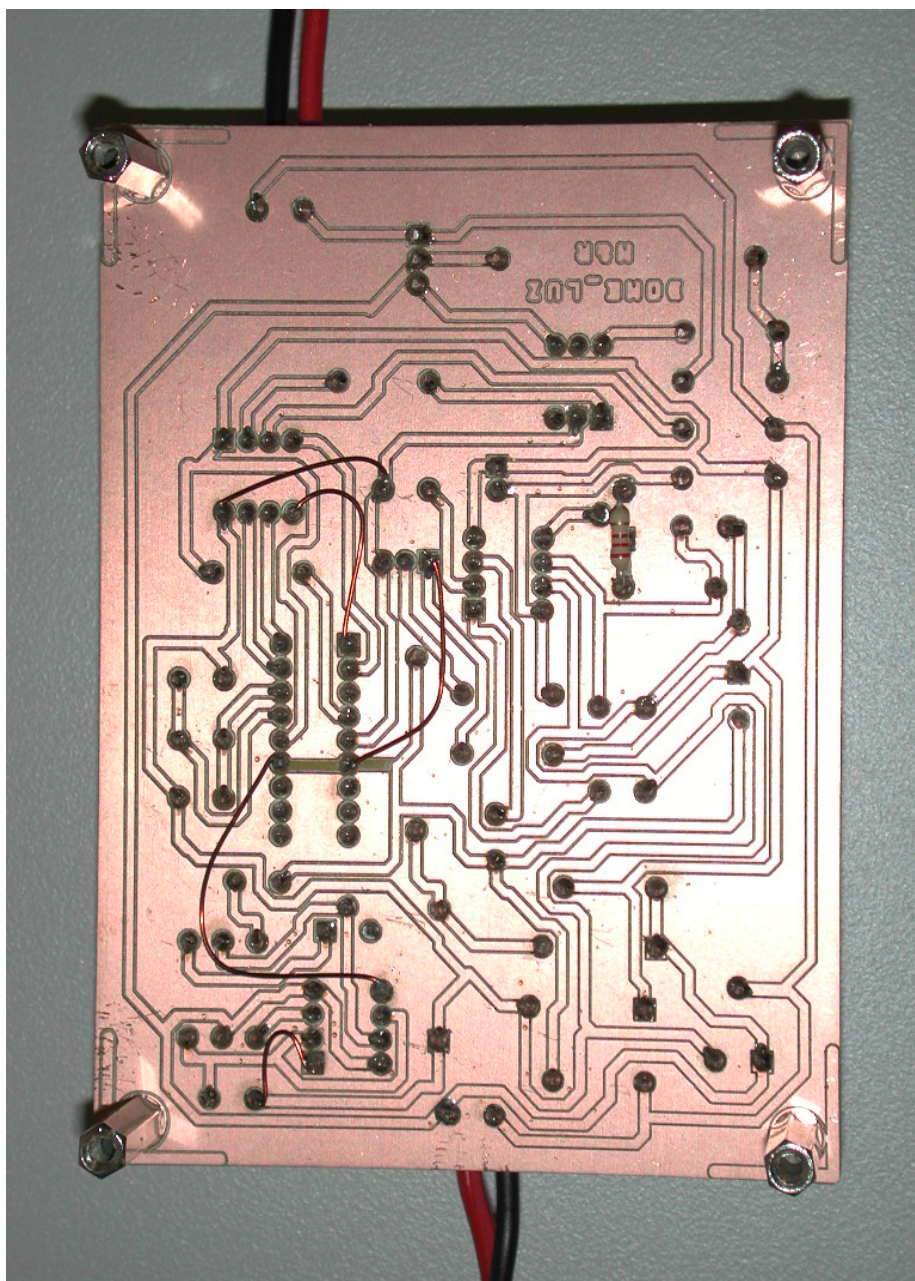


Figura D.6: *Vista inferior de la placa actuadora.*



Figura D.7: *Vista superior de programador del PIC 16F84A.*

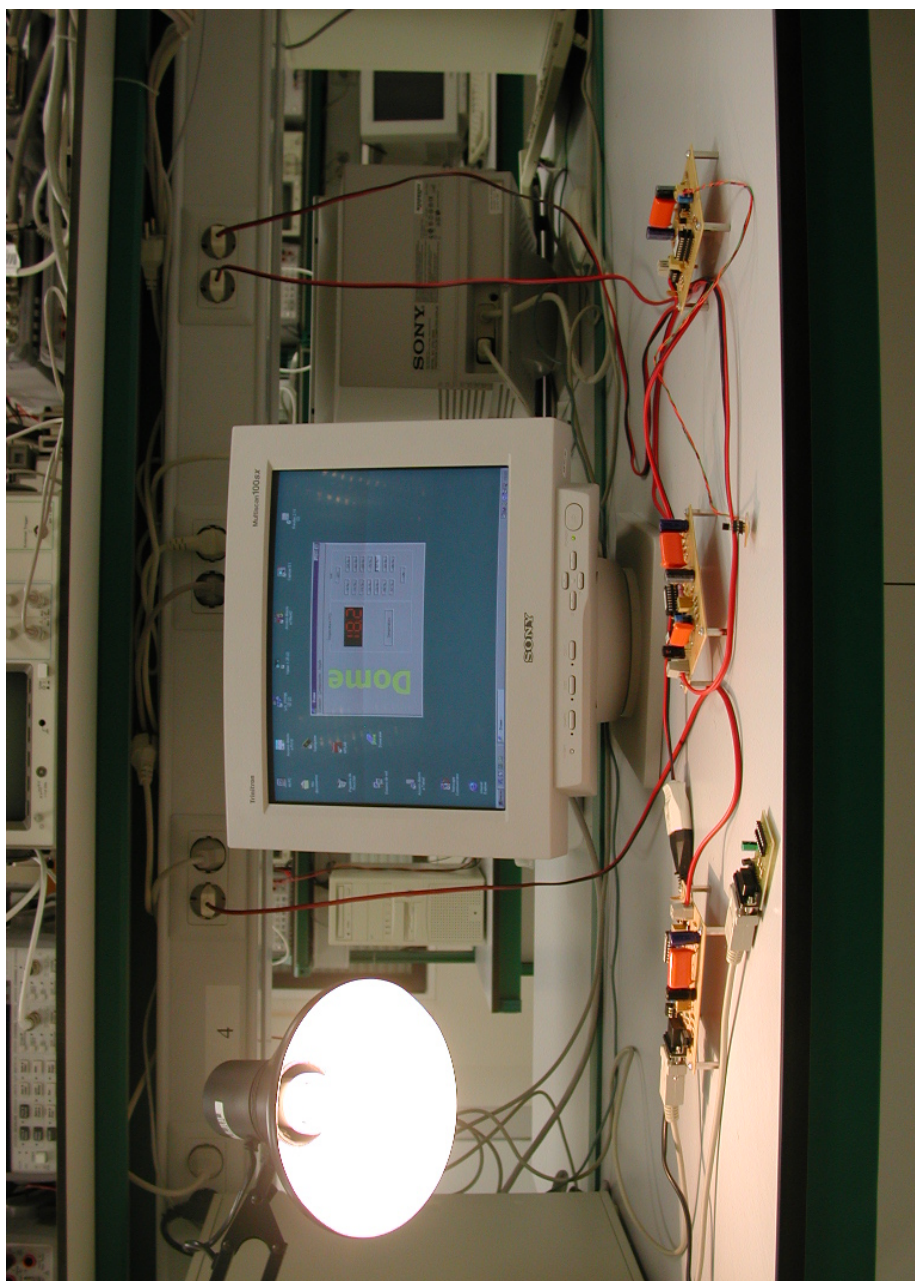


Figura D.8: *Una visión general de todo el sistema funcionando.*