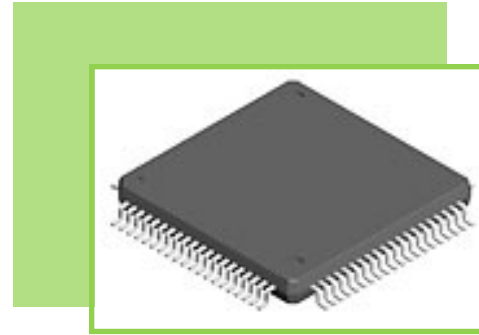


Capítulo 3

Desarrollo del hardware



- 3.1 **Descripción general del sistema**
- 3.2 **Microcontrolador MC9S12NE64**
 - 3.2.1 Aspectos generales
 - 3.2.2 Alimentación del microcontrolador
 - 3.2.3 Generador de reset y reloj
 - 3.2.4 Controlador de red
 - 3.2.4.1 Módulo EMAC
 - 3.2.4.2 Módulo EPHY
 - 3.2.4.3 Inicialización y configuración del controlador Ethernet
 - 3.2.5 Conexiones básicas del microcontrolador
- 3.3 **Puertos de comunicación**
 - 3.3.1 Interfaz de comunicación serie a USB
 - 3.3.2 Interfaz de comunicación Ethernet
 - 3.3.3 Interfaz de programación y depuración BDM
- 3.4 **Dispositivos entrada/salida e indicadores**
 - 3.4.1 Botones
 - 3.4.2 Leds
 - 3.4.3 Buzzer
 - 3.4.4 Teclado
 - 3.4.5 Display
- 3.5 **Dispositivos de monitoreo y control**
 - 3.5.1 Sensores
 - 3.5.2 Relevadores
 - 3.5.3 Entradas optoacopladas
- 3.6 **Diseño esquemático de la tarjeta de desarrollo**
 - 3.6.1 Circuito mínimo del microcontrolador
 - 3.6.2 Circuito de alimentación
 - 3.6.3 Interfaz de comunicación USB
 - 3.6.4 Dispositivos de entrada/ salida
 - 3.6.5 Dispositivos de monitoreo y control
- 3.7 **Circuito impreso de la tarjeta de desarrollo**
 - 3.7.1 Consumo de la tarjeta de desarrollo
 - 3.7.2 Implementación de la tarjeta de desarrollo

En el presente capítulo se describe el diseño e implementación de la parte física o hardware del sistema de desarrollo de propósito general que se mencionó como solución al problema planteado en el capítulo 1. Esta tarjeta de desarrollo integra medios para monitoreo y control de procesos analógicos y será el sistema en que se programará la pila de protocolos TCP/IP, permitiendo así la comunicación de manera local y a distancia con otros dispositivos que utilicen los protocolos TCP/IP para comunicarse.

3.1 Descripción general del sistema

El diseño del sistema de desarrollo se basa en un *microcontrolador* como dispositivo de procesamiento ya que su alto grado de integración lo convierte en la mejor opción en comparación con otros circuitos digitales o microprocesadores, lo que se refleja en la disminución de los componentes y el volumen del producto final, la simplificación del diseño y principalmente la reducción del costo total.

El sistema de desarrollo que se plantea integra:

- Medios para monitoreo y control de procesos en general
- Comunicación local y a distancia con otros dispositivos
- Dispositivos de entrada y salida
- Indicadores luminosos y auditivos para denotar acciones específicas que ocurren en el sistema

Para tal efecto serán necesarios:

- Entradas optoacopladas
- Sensores
- Relevadores
- Puerto de comunicación USB
- Puerto de comunicación Ethernet
- Puerto de programación y depuración (PYDBDM)
- Display
- Teclado
- Leds
- Interruptores momentáneos de presión (*push button*)
- Indicador audible o buzzer

Selección del microcontrolador

El modelo de microcontrolador que se elija debe contar al menos con los siguientes componentes que son útiles para aplicaciones de monitoreo, control y comunicación:

- Módulo convertidor analógico–digital
- Temporizadores
- Fuentes de interrupción externa
- Terminales de entradas y salidas digitales de propósito general
- Módulo de comunicación serie
- Módulo de comunicación Ethernet

El módulo de comunicación Ethernet es el principal criterio de selección del microcontrolador. Los microcontroladores que cuentan con un módulo Ethernet tienen recursos adecuados para el manejo de aplicaciones que requieren el uso de este componente, por ejemplo, la memoria de programa, la memoria temporal y la velocidad de operación. Es decir que aquellos microcontroladores que integran un módulo Ethernet disponen de suficiente memoria para programar una aplicación de comunicación por red y su velocidad de operación es tal que dicha comunicación puede ser implementada. Por tanto es suficiente un microcontrolador que trabaje a la frecuencia mínima necesaria y no es indispensable el uso de cantidades grandes de memoria, aunque existen algunos modelos con mayor capacidad de estos recursos.

En la Tabla 3.1 se resumen las características de algunos microcontroladores disponibles en el mercado que cuentan con un módulo Ethernet. En la tabla se pueden apreciar las diferencias en los periféricos de interés.

Algunos de los modelos incluyen componentes adicionales que no se muestran porque no se utilizan y por lo tanto no son de interés.

Tabla 3.1 Comparación de características entre familias de microcontroladores

Fabricante	Familia o modelo	Longitud de palabra	Velocidad (MHz)	Flash KB	RAM KB	Canales A/D	Máxima resolución A/D (bits)	Timers	Pines E/S	Puertos serie	Ethernet	No de pines	más barato / más caro	precio promedio
Microchip	PIC32MX6xx y PIC32MX7xx	32	80	256 ó 512	64 ó 128	16	10	5 x 16 bits 2 x 32 bits	53 u 85	6 SCI de 3 a 4 SPI de 4 a 5 I2C	10/100 Ethernet MAC	64 ó 100	\$ \$\$	\$
	Si2NE	16	25	64	8	8	10	4 x 16 bits	59 ó 91	2 SCI 1 SPI 1 I2C	10/100 Base-TX (MAC-PHY)	80 ó 112	\$\$	\$\$
Freescale	MCF5223x	32	60	128 ó 256	32	8	12	4 x 32 bits 4 x 16 bits	de 56 a 80	3 SCI 1 I2C 1 QSPI	10/100 Base-TX (MAC-PHY)	80, 112 ó 121	\$ \$\$	\$
	MCF528X	32	80	256 ó 512	64	8	10	8 x 16 bits 4 x 32 bits	150	3 SCI 1 I2C 1 QSPI	10/100 Ethernet MAC	256	\$\$\$\$ \$\$\$\$\$	\$\$\$\$\$
Atmel	AT32UC3A0xxx y AT32UC3A1xxx	32	66	128, 256 ó 512	32 ó 64	8	10	3 x de 16 bits	69 ó 109	4 USART 2 SPI	10/100 Ethernet MAC	100 ó 144	\$	\$
TEXAS INSTRUMENTS	LM3S6xxx y LM3S8xxx	32	25 y 50	de 64 a 256	de 16 a 64	de 0 a 8	10	5 ó 6	de 30 a 46	de 1 a 3 SCI 1 ó 2 SPI de 0 a 2 I2C	10/100 Base-TX (MAC-PHY)	100	\$ \$\$	\$\$
	LM3S9xxx	32	80 y 100	128 ó 256	64 ó 96	16	10	1 ó 7	60 ó 85	3 SCI 2 SPI 2 I2C	10/100 Base-TX (MAC-PHY)	100	\$ \$\$	\$\$
Dallas Semiconductor	DS80C4xxx	8	75	64 (ROM)	1 ó 64	0	0	4 x 16 bits	64	3 puertos serie	10/100 Ethernet MAC	100	\$\$ \$\$\$	\$\$\$
NXP (antes Philips)	LPC17xxx	32	100	de 128 a 512	32 ó 64	6	12	4	52 ó 70	4 SCI 1 SPI 3 I2C	10/100 Ethernet MAC	80 ó 100	\$ \$\$	\$
	LPC23xxx y LPC24xxx	32	72	de 128 a 512	de 34 a 38	6 u 8	10	4	de 70 a 160	4 SCI 1 SPI 3 I2C	10/100 Ethernet MAC	de 100 a 208	\$ \$\$\$	\$\$
Renesas (antes Mitsubishi y Hitachi)	Rx62N	32	100	384 ó 512	64 ó 96	8	10 ó 12	4 x 8 bits 4 x 16 bits	103 ó 126	6 SCI 2 SPI 2 I2C	10/100 Ethernet MAC	144, 145 ó 176	\$\$	\$\$
	H8S/24xxx	16	34	512 (ROM)	40	8	10	4 x 8 bits 4 x 16 bits	de 85 a 110	3 puertos serie	10/100 Ethernet MAC	144 ó 176	\$\$ \$\$\$	\$\$\$
Zilog	eZ80F91xx	8	50	256	16	0	0	4 x 16 bits	32	2 SCI 1 SPI 1 I2C	10/100 Ethernet MAC	144	\$\$ \$\$\$\$\$	\$\$

El módulo Ethernet que incluyen algunas de las familias de microcontroladores consta sólo de un controlador de acceso al medio (MAC), en la tabla se indica como *10/100 Ethernet MAC*. Este módulo implementa las funciones de la capa de enlace, pero carece de la funcionalidad de la capa física. La capa física se implementa con un dispositivo denominado transceptor (por transmisor y receptor) físico, comúnmente abreviado como PHY (*PHYSical Transceiver*). Los microcontroladores que no cuentan con este dispositivo integrado necesitan conectarse a uno externo. No obstante, algunos modelos de microcontroladores incluyen también el transceptor físico dentro de su módulo Ethernet, en la Tabla 3.1 se puede identificar este tipo de módulo Ethernet como *10/100 Base-TX (MAC+PHY)*.

El uso de un microcontrolador con módulo Ethernet conformado por MAC y PHY conlleva a un diseño más reducido y simplificado. Estos microcontroladores tienen terminales dedicadas para las señales de recepción y transmisión que se conectan directamente a un transformador de aislamiento magnético y éste a su vez con un conector RJ45. Dichas señales de recepción y transmisión son codificadas y decodificadas dentro del microcontrolador de acuerdo al estándar Ethernet. En la Figura 3.1 se muestra la diferencia entre los dos tipos de módulos Ethernet mencionados y la conexión necesaria del sistema.

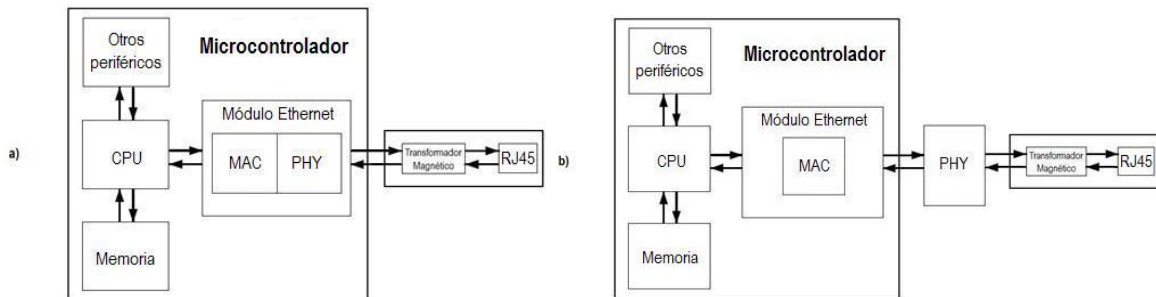


Figura 3.1 Módulos Ethernet en microcontroladores a) Módulo Ethernet con PHY y MAC. b) Módulo Ethernet con MAC

De los modelos de microcontroladores mostrados en la Tabla 3.1 podemos observar que son pocos los que integran un módulo Ethernet capaz de ejecutar la funcionalidad de la capa de enlace más la de la capa física. Dichos modelos corresponden a microcontroladores de los fabricantes *Freescale Semiconductor* y *Texas Instruments*. Las características de este conjunto de microcontroladores son similares, incluyendo el precio, aunque algunos tienen un poco más de capacidad en algunos recursos que otros.

Es importante considerar las herramientas para la programación de cada modelo que se contemple para la implementación del sistema, por ejemplo, la adquisición de un programador para el microcontrolador seleccionado. Los microcontroladores S12NE de *Freescale* (de la familia 9S12) tienen un procesador compatible con el de los antiguos microcontroladores M68HC12 y HCS12 de Motorola y por lo tanto, son compatibles con las herramientas de desarrollo disponibles para dichos microcontroladores. En contraste, la familia MCF5223, también de *Freescale*, tiene un núcleo denominado *ColdFire* y aunque esta familia es más reciente y un poco más potente que la anterior, las herramientas de desarrollo son más costosas; un programador para estos microcontroladores cuesta alrededor de 2.5 veces más que uno para la familia 9S12. En cuanto a los microcontroladores de *Texas Instruments*, a pesar de que tienen un precio similar a los S12NE y cuentan con más recursos, la adquisición de un programador representa una inversión de más de 10 veces lo que cuesta adquirir un programador para los 9S12¹¹.

Existen dos presentaciones del modelo S12NE64: una cuenta con un total de 80 terminales mientras que la otra dispone de 112, esta última cuenta con 70 terminales de entrada/salida y 10 sólo para entradas, lo cual es adecuado para conectar los dispositivos necesarios para nuestro sistema de desarrollo como: el display, el teclado, los leds y demás componentes antes mencionados. Este último microcontrolador es el que se ha seleccionado para la implementación del sistema de desarrollo, su estructura y funcionamiento se describe en el tema 3.2.

¹¹ De acuerdo a precios consultados en las páginas de los fabricantes y proveedores en 2010

3.2 Microcontrolador MC9S12NE64

El microcontrolador seleccionado para este desarrollo es el MC9S12NE64 (ó NE64), de la compañía *Freescale Semiconductor* y perteneciente a la familia de microcontroladores 9S12. Posee un núcleo HCS12, compatible con el set de instrucciones de los microcontroladores M68HC11 y con modelo de programación idéntico al de éstos. Este modelo de microcontrolador incorpora en el mismo chip el módulo controlador de red conformado por MAC y PHY.

3.2.1 Aspectos Generales

La nomenclatura con la que se designa a cada modelo de microcontrolador en particular especifica diferentes características con las que cuenta. Para el caso del MC9S12NE64:

- MC** → Estatus del producto, MC indica que es un producto completamente validado y que se encuentra en plena producción
- 9** → Tipo de memoria, el tipo 9 corresponde a memoria FLASH
- S12** → Tipo de núcleo
- NE** → Familia
- 64** → Cantidad de memoria interna flash (64 KB)

El microcontrolador MC9S12NE64 cuenta con:

- Unidad central de procesamiento de 16 bits (CPU HCS12).
- Memoria FLASH EEPROM de 64 Kbytes.
- Memoria RAM de 8 Kbytes.
- Controlador de acceso al medio Ethernet (EMAC) compatible con la norma IEEE 802.3, con transceptor físico Ethernet de 10/100 Mbps integrado (EPHY).
- Dos módulos de comunicación serie asincrónica (SCI).
- Una interfaz periférica serie síncrona (SPI)
- Una interfaz I²C (inter IC ó IIC).
- Un módulo temporizador (*timer*) de 4 canales de 16 bits (TIM) con captura de entradas.
- Salidas de comparación y acumulador de pulsos.
- Un convertidor analógico–digital de 10 bits y 8 canales (ATD).
- Más de 21 terminales disponibles como entradas para despertar (keypad wakeup inputs KWU).
- Dos entradas para interrupciones asíncronas externas.
- Circuito PLL.
- Regulador de voltaje interno (VREG_PHY) que genera un voltaje interno de 2.5 V (VDD) a partir de una fuente de alimentación externa de 3.15 a 3.45 V. La frecuencia de operación es de 50 MHz, lo que corresponde a una velocidad de bus de 25 MHz.

En la Figura 3.2 se muestra el diagrama de bloques del MC9S12NE64. Cada puerto del microcontrolador está asociado a una función que se puede multiplexar con la función de entrada y salida. En el diseño del sistema de desarrollo algunos puertos se usan como entradas y salidas para la conexión de componentes y otros se usan de acuerdo a la función a la cual están asociados. La Tabla 3.2 muestra la disposición y uso de los puertos del microcontrolador.

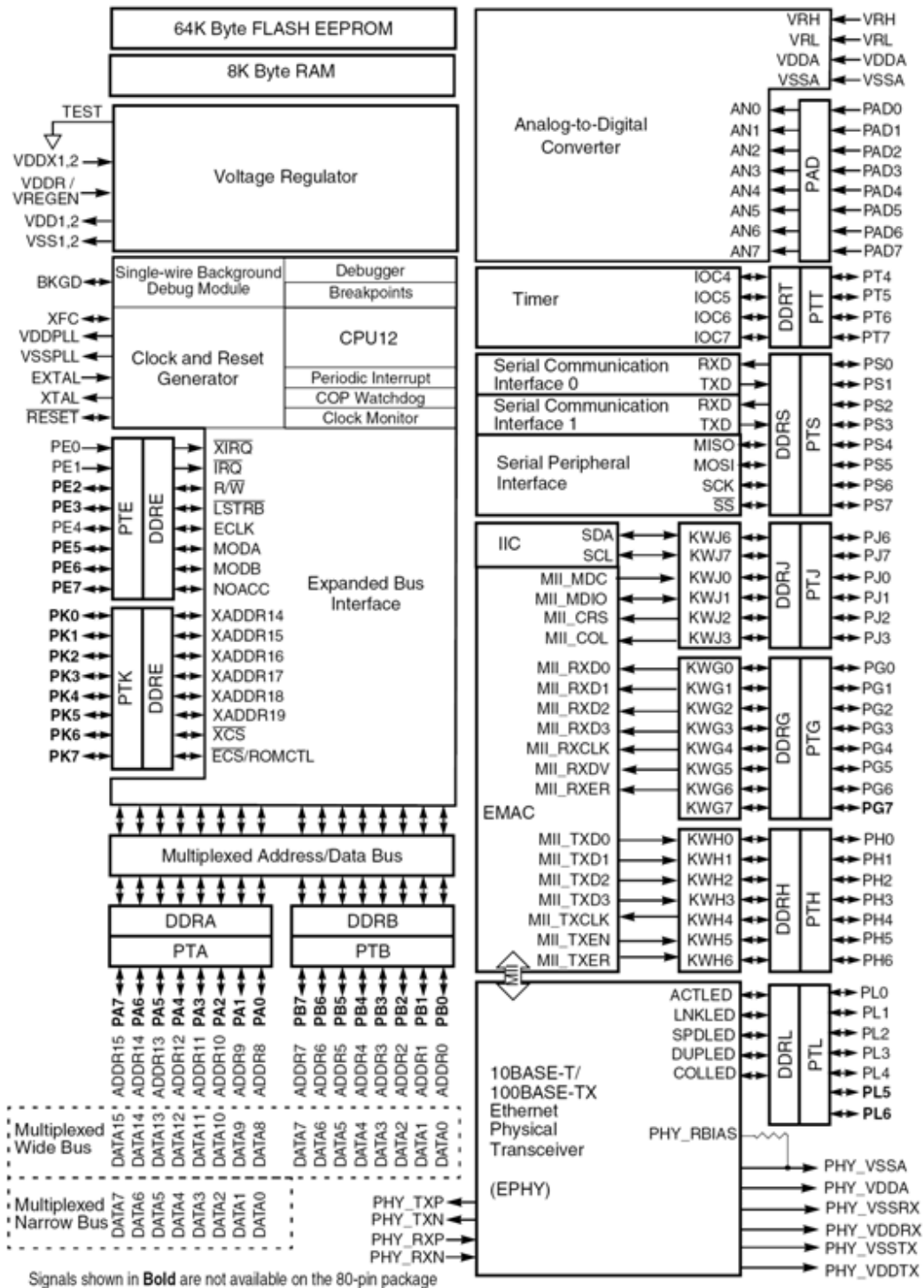


Figura 3.2 Diagrama de Bloques del microcontrolador MC9S12NE64

Tabla 3.2 Uso de los puertos del microcontrolador

Puerto	Función Asociada	No. de Terminales	Terminales utilizadas	Uso en el sistema
A	Terminales multiplexadas para direcciones y datos de la interfaz del bus de expansión	8	*	Libre para ser usado por las aplicaciones
B	Terminales multiplexadas para direcciones y datos de la interfaz del bus de expansión	8	*	Libre para ser usado por las aplicaciones
AD	Convertidor Analógico-Digital	8	AD0, AD1, AD2	Se utilizan tres terminales para conectar sensores y las demás se dejan libres para aplicaciones
E	Interfaz de expansión de bus y entradas de interrupción externa	8	*	Los dos bits menos significativos sirven para generar interrupciones por eventos externos, todos los bits del puerto se dejan libres para las aplicaciones
G	Interfaz Independiente del Medio Externa y entradas para despertar	8	G0-G7	Conexión del teclado matricial
H	Interfaz Independiente del Medio Externa y entradas para despertar	7	H0-H6	Conexión del display
J	Interfaz Independiente del Medio Externa, entradas para despertar e interfaz I ² C	6	J0-J5	Conexión de optoacopladores (2), relevadores (2) y buzzer (indicador auditivo)
K	Interfaz del bus de expansión	8	K0-K7	Conexión de botones y leds de propósito general
L	Leds de estatus de la conexión de red	7	L0-L6	Se utilizan cinco terminales (L0-L4) para indicar el estado de la conexión física de la red
S	Puerto de comunicación serie	8	S0-S3	Se utiliza su función asociada a la comunicación serie asíncrona (SCIO)
T	Timers	4	*	Libre para uso de las aplicaciones

3.2.2 Alimentación del Microcontrolador

El NE64 se alimenta con una fuente de voltaje directo de 3.3 V \pm 5%. Cuenta con varias terminales donde se debe conectar la entrada de voltaje. Algunos módulos necesitan una alimentación de 2.5 V, éstos pueden ser alimentados directamente por una fuente externa de 2.5 V o internamente con la señal generada por el regulador de voltaje interno, el cual se alimenta con 3.3 V. Este regulador interno se puede deshabilitar, pero esto implicaría que el microcontrolador se alimentara con dos fuentes externas.

En la Figura 3.3 se muestra la distribución de terminales del microcontrolador, en donde se han resaltado las terminales que deben ir conectadas a las fuentes de alimentación y en la Tabla 3.3 se resumen las características de alimentación de los periféricos del microcontrolador.

La terminal 82 (V_{DDR}) es la alimentación del regulador de voltaje interno, se alimenta con 3.3 V y genera una salida de 2.5 V que sirve a otros periféricos como alimentación. Si V_{DDR} se conecta a tierra, el regulador de voltaje interno se apaga y todos los periféricos que requieran la alimentación de 2.5 V deben tomarla de una fuente externa conectada a sus terminales de alimentación.

Las terminales 94 (V_{RH}) y 95 (V_{RL}) son las referencias de voltaje alto y bajo para el convertidor Analógico-Digital, éstas se pueden conectar a cualquier valor de voltaje de tal manera que definan el rango de operación del convertidor (dentro de los rangos de operación del microcontrolador), en este caso las referencias se han conectado a tierra y 3.3 V.

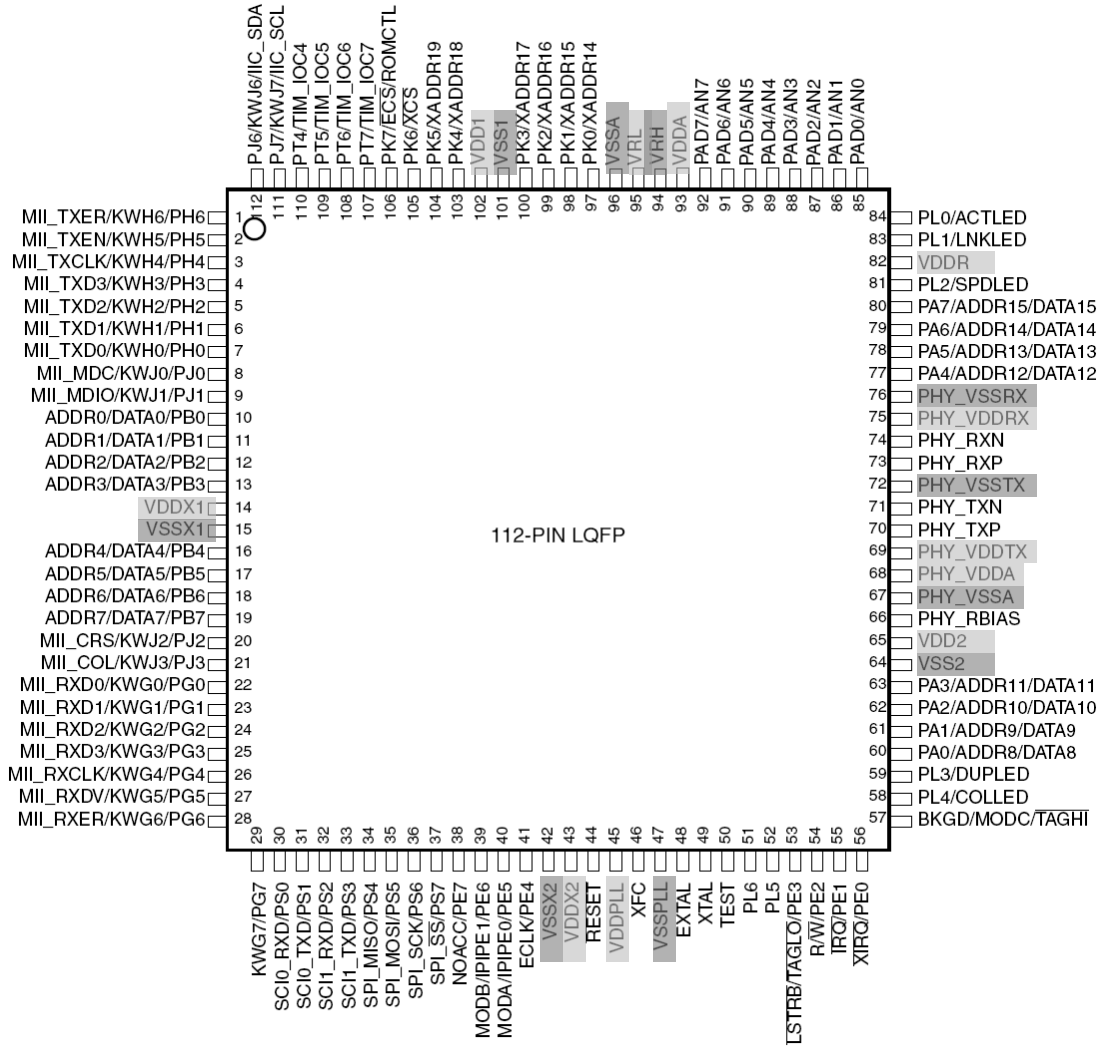


Figura 3.3 Encapsulado del microcontrolador MC9S12NE64

Tabla 3.3 Alimentación del microcontrolador MC9S12NE64

Número de Terminal	Nombre	Descripción	Valor nominal
14	V _{DDX1}	Señales externas de tierra y voltaje para entradas y salidas de puertos	3.3 V
43	V _{DDX2}		0 V
15	V _{SSX1}	Señales de voltaje y tierra para el módulo EPHY, pueden ser suministradas por el regulador de voltaje interno o conectadas externamente mediante las terminales correspondientes.	3.3 V
42	V _{SSX2}		0 V
82	V _{DDR}	Alimentación del regulador interno	3.3 V
93	V _{DDA}	Señales externas de voltaje y tierra para convertidor Analógico/Digital	3.3 V
96	V _{SSA}	Referencia positiva del convertidor ATD	0 V
94	V _{RH}		3.3 V
95	V _{RL}	Referencia negativa del convertidor ATD	0 V
69	PHY_V _{DDTX}	Señales de voltaje y tierra para la alimentación del núcleo,	2.5 V
75	PHY_V _{DDRX}		
68	PHY_V _{DDA}		
72	PHY_V _{SSTX}		
76	PHY_V _{SSRX}	Señales de voltaje y tierra para la alimentación del núcleo,	2.5 V
67	PHY_V _{SSA}		
102	V _{DD1}		

65	V _{DD2}	generadas por el regulador interno o suministradas por medio de las terminales externas cuando el regulador de voltaje se apague.	
101	V _{SS1}		0 V
64	V _{SS2}		
45	V _{DDPLL}	Alimentación de voltaje y tierra del módulo <i>Phase-Lock Loop</i> (PLL), la alimentación es generada por el regulador interno o suministrada externamente.	2.5 V
47	V _{SSPLL}		0 V

3.2.3 Generador de Reset y Reloj

El MC9S12NE64 cuenta con un módulo llamado Generador de Reset y Reloj, o CRG (Clock and Reset Generator), que se encarga de generar y distribuir las señales de reloj necesarias para cada uno de los periféricos y el núcleo. La señal de entrada proviene de un cristal conectado a dos terminales del microcontrolador (EXTAL y XTAL, terminales 48 y 49) y alimenta al módulo CRG, éste genera tres señales de reloj diferentes: el reloj de núcleo, el reloj de bus y el reloj oscilador. Cada uno de los módulos se alimenta con alguna de las señales generadas por el CRG, como se muestra en la Figura 3.4.

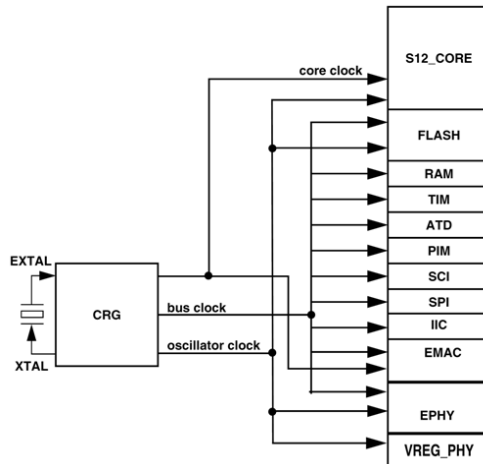


Figura 3.4 Señales de reloj del módulo CRG para los periféricos y el núcleo del microcontrolador

Las terminales externas de este módulo son:

- EXTAL. Es la entrada de reloj del sistema.
- XTAL. Es una salida en donde se puede leer la señal de reloj del microcontrolador (la frecuencia de bus, es usada por los periféricos).
- XFC. Es una entrada en donde se debe conectar un circuito RC de segundo orden para disminuir el voltaje de rizo en la estabilización de la señal de reloj generada por el módulo PLL, la velocidad con la que se estabilice el voltaje depende de los valores del arreglo RC y de la frecuencia de entrada. Si el módulo PLL no se va a utilizar la terminal XFC se debe conectar a la tierra del PLL, V_{SSPLL}. El arreglo que debe ser conectado a esta terminal se muestra en la Figura 3.5.

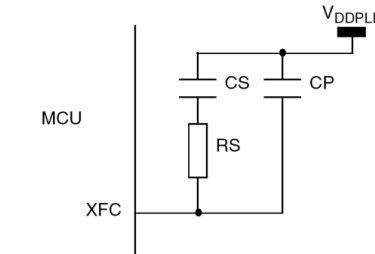


Figura 3.5 Arreglo RC conectado a la terminal XFC del módulo CRG

La terminal de RESET es bidireccional, como entrada recibe un valor bajo de voltaje para generar una inicialización de todo el microcontrolador a un estado conocido. Como salida envía una señal que indica que ocurrió una inicialización interna del sistema. Las causas que pueden generar un reset son:

- Encendido del sistema.
- Caída del voltaje de alimentación.
- Reset externo.
- Reset generado por el monitor de reloj, el cual supervisa la calidad de la señal de reloj y si ésta se pierde o deteriora genera un reset.
- Reset generado por *watchdog* es un mecanismo que monitorea el funcionamiento del microcontrolador y al encontrar una falla genera un reset, a este mecanismo se le conoce como *Computer Operating Properly (COP)*.

Las dos últimas condiciones de reset pueden ser deshabilitadas mientras que las otras tres no hay manera de impedir las.

El diagrama de bloques del módulo generador de reset y reloj se muestra en la Figura 3.6. Las entradas de este módulo corresponden a la señal de reset externa, la entrada de reloj, la señal del filtro RC y las entradas de alimentación. Estas entradas se muestran en la parte izquierda de la figura. Las señales de salida que genera el módulo son para uso interno de los periféricos del microcontrolador, mostradas en la parte derecha del diagrama, y corresponden a la señal de reset, las señales de reloj generadas para los periféricos y núcleo y las señales de interrupciones.

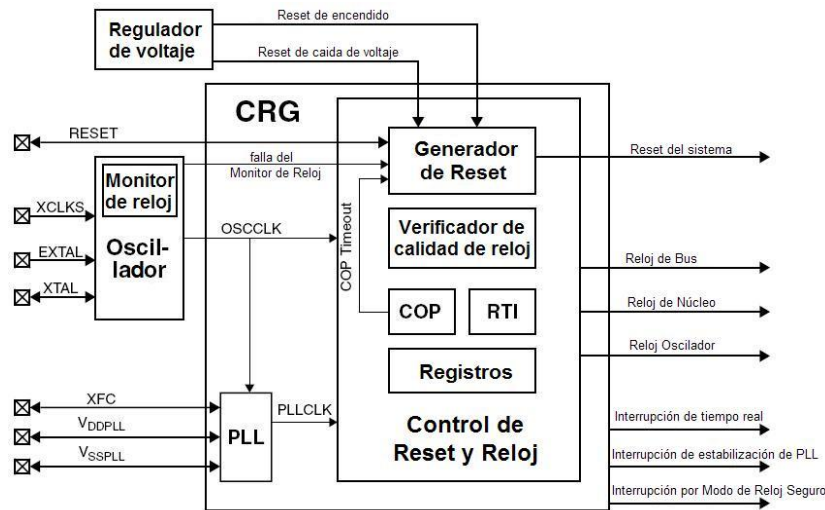


Figura 3.6 Diagrama de bloques del módulo Generador de Reset y Reloj (CRG)

Las señales de reloj necesarias para el funcionamiento del núcleo y de los periféricos pueden ser calculadas a partir de la señal OSCCLK, que proviene de la entrada de reloj conectada externamente, o a partir de la señal PLLCLK, que es generada por el módulo PLL cuando está habilitado.

El bloque PLL genera una base de tiempo distinta a la de la señal de entrada, la frecuencia que genera proviene de multiplicar y dividir el valor de la frecuencia de entrada (OSCCLK) utilizando las cantidades contenidas en los registros SYNRR y REFDRV respectivamente, de tal manera que la frecuencia generada por el módulo PLL es:

$$PLLCLK = 2 OSCCLK \frac{[SYNRR + 1]}{[REFDRV + 1]}$$

El valor que multiplica a la frecuencia de oscilación base se lee del registro *Synthesizer Register (SYNRR)* y es un valor de 6 bits. El valor que divide a la frecuencia base se lee del registro *Reference Divider Register (FERDIV)* y es un valor de 4 bits. En la Figura 3.7 se muestran ambos registros.

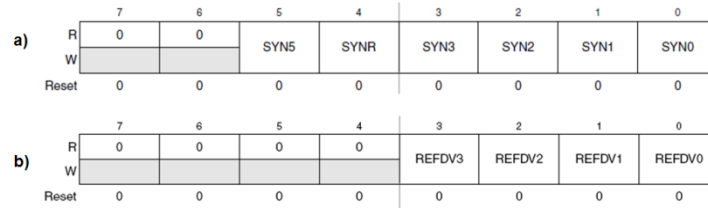


Figura 3.7 a) Registro Sintetizador SYN b) Registro Divisor de Referencia REF

Para seleccionar entre la señal OSCCLK o la señal PLLCLK como señal de reloj del sistema (SYSCLK) se utiliza el bit PLLSEL del registro CLKSEL. Si PLLSEL = 0, los relojes del sistema se calculan a partir de OSCCLK = SYSCLK. Si PLLSEL = 1 los relojes del sistema se calculan a partir de PLLCLK = SYSCLK, en este caso el módulo PLL debe encenderse escribiendo un uno al bit PLLON en el registro PLLCTL.

La frecuencia de bus, o *Bus Clock*, y la frecuencia del núcleo, o *Core Clock*, quedan determinadas por la frecuencia del sistema, SYSCLK, de la siguiente manera:

$$\text{Bus Clock} = \frac{\text{SYSCLK}}{2} \quad \text{Core Clock} = \text{SYSCLK}$$

Los periféricos y los bloques de memoria del microcontrolador se alimentan de la señal de reloj de bus, algunos periféricos también usan la frecuencia del Oscilador, *Oscillator Clock*. La señal *Core Clock* es la señal de reloj para el CPU.

El reloj oscilador es una señal que se genera cuando el reloj de sistema se pierde o debilita y alimenta a algunos periféricos a una frecuencia mínima f_{SCM} para impedir que dejen de funcionar mientras el reloj de sistema se recupera. A este mecanismo se le llama *modo de reloj seguro* y puede ser habilitado o deshabilitado.

El módulo oscilador del CRG cuenta con un monitor de reloj (*Clock Monitor*) que supervisa la señal del cristal de entrada, si no detecta señal dentro de un tiempo determinado, se genera un evento que consiste en la inicialización del sistema, o bien, entrar al modo de reloj seguro, dependiendo las configuraciones seleccionadas. Si se deshabilita el monitor de reloj no se detecta la pérdida o condición de baja calidad de la señal de reloj y en caso de que ocurra el funcionamiento del microcontrolador es impredecible.

3.2.4 Controlador de red del microcontrolador MC9S12NE64

El MC9S12NE64 es una solución para conectividad Ethernet completa y muy adecuada ya que cuenta con un controlador Ethernet implementado en los módulos EMAC (*Controlador de Acceso al Medio Ethernet*) y EPHY (*Transceptor Físico Ethernet*) y no necesita componentes activos extras para su funcionamiento. Se requieren pocos componentes para habilitar el controlador de red del microcontrolador:

- Cristal de 25 MHz
- Fuente de voltaje de 3.3 V
- Transformador de aislamiento magnético de alta velocidad (útil para suprimir el ruido eléctrico y acoplar señales)
- Conector RJ45 hembra
- Capacitores y resistencias
- 5 leds de estatus PHY

Contexto del controlador ethernet dentro de las redes de comunicaciones

El controlador de red del NE64 implementa las capas más bajas del modelo de red; el módulo EPHY implementa la capa física y el EMAC la capa de enlace de datos. En la Figura 3.8 se muestra en qué parte del modelo se implementan las funciones del controlador de red.

La funcionalidad, compatibilidad e interoperabilidad de la implementación Ethernet del NE64 con otros dispositivos basados en esta norma fue probada y completamente validada por el *Consortio Fast Ethernet* en el *Laboratorio de Interoperabilidad de la Universidad de New Hampshire*, donde fueron examinados los

módulos EMAC y EPHY del microcontrolador MC9S12NE64 y su operación fue comparada contra los criterios de la norma IEEE 802.3 para asegurar que el diseño Ethernet es correcto y robusto. Los resultados de estas pruebas indican que el NE64 es completamente funcional y que se interconecta de manera adecuada con dispositivos de diferentes fabricantes [16e].

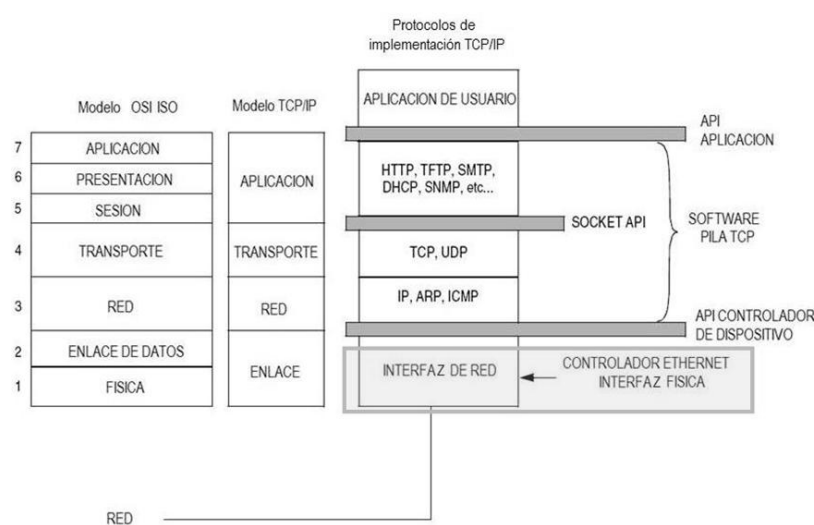


Figura 3.8 Relación entre el modelo de referencia OSI, el modelo TCP/IP y los protocolos que implementan el modelo TCP/IP

La funcionalidad, compatibilidad e interoperabilidad de la implementación Ethernet del NE64 con otros dispositivos basados en esta norma fue probada y completamente validada por el *Consortio Fast Ethernet* en el *Laboratorio de InterOperabilidad de la Universidad de New Hampshire*, donde fueron examinados los módulos EMAC y EPHY del microcontrolador MC9S12NE64 y su operación fue comparada contra los criterios de la norma IEEE 802.3 para asegurar que el diseño Ethernet es correcto y robusto. Los resultados de estas pruebas indican que el NE64 es completamente funcional y que se interconecta de manera adecuada con dispositivos de diferentes fabricantes [16e].

3.2.4.1 Módulo EMAC – Controlador de acceso al medio Ethernet

El módulo EMAC del MC9S12NE64 cuenta con las siguientes características:

- Filtro de reconocimiento de direcciones
- Modo de promiscuidad (en este modo se permite la recepción de todo el tráfico de red sin la aplicación de filtros [16e])
- Filtro ethernet
- Modo de PHY externo
- Control de flujo con recepción y detección de envío de tramas de pausa
- Dos buffers de recepción con detección de recepción de tramas válido
- Un buffer de transmisión con detección de transmisión de trama completa
- Detección de desbordamiento de buffers
- Detección de error por sobrepaso de tamaño máximo de buffer
- Detección de error de longitud de trama y error de código de redundancia
- Detección de transferencias en la interfaz MII
- Detección de colisiones retardadas y exceso de colisiones
- Modo Loop-back
- Modo Full-Duplex o Half-Duplex

En la Figura 3.9 se muestra el diagrama de bloques del módulo EMAC

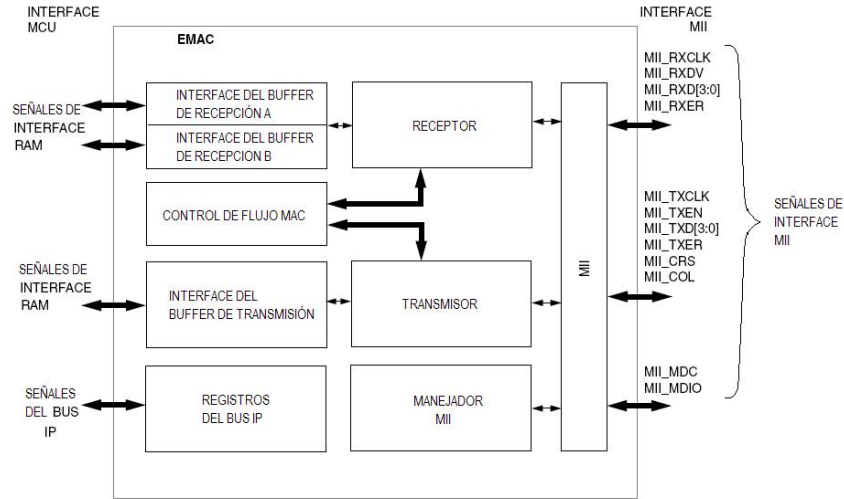


Figura 3.9 Diagrama de bloques del módulo EMAC

Los principales componentes del módulo incluyen el receptor, el transmisor, el control de flujo, el manejador MII y las interfaces de los buffers de recepción y transmisión.

El módulo EMAC del NE64 provee los mecanismos necesarios para transferir datos de la capa de enlace de un dispositivo Ethernet a otro. Está diseñado específicamente para procesadores embebidos de 8 ó 16 bits y cumple los criterios de la norma IEEE 802.3 (*Ethernet*) y algunas variantes, 802.3u (*fast Ethernet*), 802.3x (*Full-Duplex OPS*), por lo que soporta operación a 100 ó 10 Mbps, control de flujo (en modo Full-Duplex) y modo de auto negociación de habilidades. En la Figura 3.10 se muestran las características de las variantes mencionadas. El módulo EMAC Cuenta con una interfaz independiente del medio (MII) para comunicarse con el módulo EPHY.

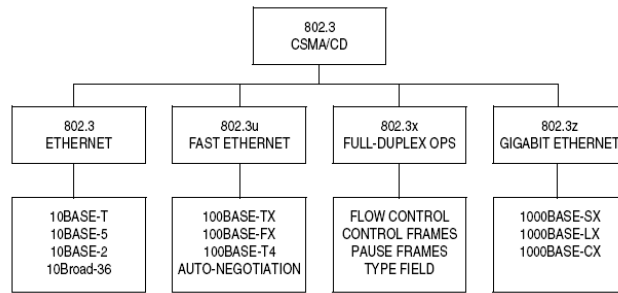


Figura 3.10 Variantes de la norma IEEE 802.3

En el registro de *Control de Red*, o NETCTL, se habilita el módulo EMAC y se configura el modo en Full-Duplex o Half-Duplex.

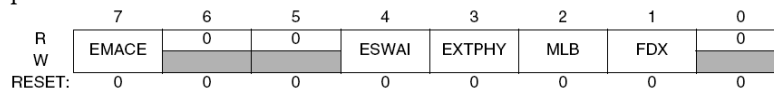


Figura 3.11 Registro de Control de Red NETCTL

El bit EMACE habilita el módulo escribiéndole un 1. Una vez habilitado es posible la transmisión y recepción de datos. Al deshabilitarlo (escribiendo un 0 en el bit EMACE) el transmisor y receptor EMAC quedan deshabilitados y cualquier transmisión que se encuentre en progreso se trunca.

Cuando se le escribe un 1 al bit MLB (modo de *loopback*) todo el tráfico del transmisor se dirige al receptor del mismo microcontrolador y se deshabilita la interfaz MII. Esto permite probar el funcionamiento del controlador Ethernet sin conectarlo a una red real.

El bit FDX (Full-Duplex) selecciona el modo de operación del controlador: FDX=1 habilita el modo Full-Duplex y FDX=0 habilita el modo Half-Duplex. Aquí se configura la dirección del flujo manualmente en

caso de no usar modo de auto negociación de habilidades. Si se usa auto negociación, esta configuración se determina en el momento de la conexión y posterior a esto se debe ajustar el bit FDX de a cuerdo al modo acordado.

Modos de operación: full duplex, half duplex, Velocidad y control de flujo

Los modos posibles de operación del controlador Ethernet incluyen: Half-Duplex a 10 Mbps, Full-Duplex a 10 Mbps, Half-Duplex a 100 Mbps y Full-Duplex a 100 Mbps.

Cuando se configura en modo Half-Duplex, dos o más dispositivos comparten el medio de transmisión, de manera que no es posible que más de un dispositivo utilice el medio para enviar información al mismo tiempo ni que un nodo reciba y transmita simultáneamente. En esta configuración el controlador de red utiliza el protocolo CSMA/CD para asignar el acceso al medio.

En el modo de operación Full-Duplex se tienen dos canales de comunicación independientes, lo que permite que dos equipos conectados punto a punto puedan utilizar el medio de manera simultánea, evitando colisiones. En este modo de operación el controlador de red no requiere implementar el protocolo CSMA/CD. Cuando se selecciona este modo se debe utilizar un mecanismo de control de flujo para evitar que el emisor saturé al receptor. Este mecanismo corresponde al envío de tramas de pausa y es configurable mediante los registros de control del módulo EMAC.

El EMAC es compatible con la característica de control de flujo definida por la IEEE 802.3x. Se puede habilitar siempre y cuando el controlador Ethernet esté configurado en modo Full-Duplex. El control de flujo se habilita o deshabilita en el registro de *Control y Estatus del Receptor*, o RXCTS, con el bit RFCE (*Recepción de Control de Flujo Habilitada*). La Figura 3.12 muestra el registro RXCTS.

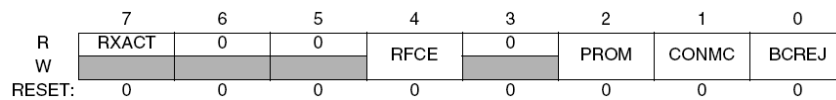


Figura 3.12 Registro de Control y Estatus del Receptor RXCTS

Cuando el bit RFCE tiene el valor de 1, el receptor EMAC detecta tramas de pausa y al recibirlas el transmisor detiene su operación por el tiempo solicitado. Cuando se tiene un 0 en el bit RFCE, las tramas de pausa recibidas son ignoradas.

El envío de tramas se controla en el registro de *Control y Estatus del Transmisor*, o TXCTS, el cual se ilustra en la Figura 3.13. Los bits del campo TCMD (*Comando de Transmisión*) le indican al transmisor la operación que debe realizar. Para enviar una trama de pausa el comando para el transmisor es 10 binario (2 en decimal).

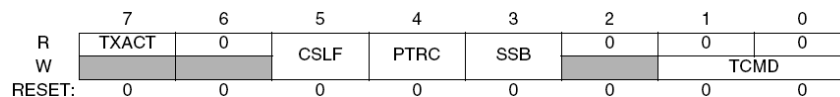


Figura 3.13 Registro de Control y Estatus del Transmisor TXCTS

Al escribir el valor binario 10 en el campo TCMD del registro TXCTS el transmisor construye la trama de pausa mostrada en la Figura 3.14 y la transmite mientras no haya una transmisión activa. La dirección de destino la ajusta el transmisor al valor 01-80-C2-00-00-01.

Preámbulo	SFD	Dirección Destino	Dirección Origen	Tipo/Longitud	Código de Operación MAC	Parámetros de Control MAC	Reservado	FCS
7 bytes	1 byte	6 bytes = (01-80-C2-00-00-01) o dirección de destino	6 bytes	2 bytes = Control MAC (88-08)	2 bytes = (00-01)	2 bytes = (00-00 a FF-FF)	42 bytes = todos ceros	4 bytes

Figura 3.14 Formato de una trama de pausa

Modo de auto negociación

Para determinar el modo de operación del controlador de red se puede utilizar la auto negociación o ajustarse los valores manualmente en los registros correspondientes. El modo de auto negociación se habilita escribiendo un 0 al bit ANDIS (*Deshabilitar Auto negociación*) del *registro 0 de Control del Transceptor Físico Ethernet* o EPHYCTL0, la Figura 3.15 ilustra el contenido del registro.

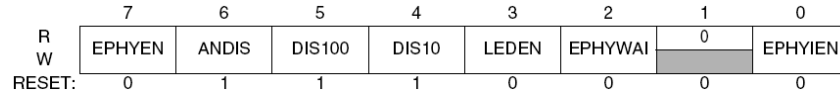


Figura 3.15 Registro 0 de Control del Transceptor Físico Ethernet EPHYCTL0

En modo de auto negociación el modo Full-Duplex y la velocidad de transmisión se determinan en el momento que se establece una conexión con otro dispositivo. Los dispositivos involucrados negocian las características de la comunicación y, si llegan a un acuerdo, ambos dispositivos se configuran a sí mismos para operar en dicho modo, a partir de entonces la comunicación se realiza atendiendo a la configuración seleccionada. Si no se requiere utilizar el modo de auto negociación se debe deshabilitar y el modo de operación se debe ajustar manualmente en el momento de la inicialización del dispositivo.

Filtro de reconocimiento de direcciones MAC

El receptor Ethernet del NE64 proporciona control de filtrado de paquetes entrantes basado en la dirección MAC de destino. Si se aplican filtros a las direcciones MAC, cada paquete recibido es inspeccionado para verificar su dirección de destino, si no cumple con los criterios seleccionados se desecha. Si el filtro de direcciones se deshabilita los paquetes recibidos son aceptados sin revisar su dirección de destino. El modo de promiscuidad permite al receptor aceptar cualquier paquete sin importar su dirección de destino. Esta opción sobrescribe cualquier otra configuración de filtro de direcciones.

El reconocimiento de direcciones MAC se configura en el registro de *Control y Estatus del Receptor*, (RXCTS), ilustrado en la Figura 3.12. El bit CONMC (Multicast condicionado), configura el filtrado de direcciones multicast. Si se le escribe un 1 se rechazan aquellos paquetes recibidos con dirección multicast que no se encuentre en la tabla de direcciones. En caso de que el bit CONMC tenga un 0 los paquetes con direcciones multicast son aceptados.

El bit BCREJ (Rechazar broadcast), aplica filtro a la dirección broadcast. Si BCREJ=1 las tramas recibidas con dirección broadcast se rechazan. Si BCREJ=0 se aceptan tramas con dirección broadcast.

Si se escribe 1 en el bit PROM se selecciona el modo de promiscuidad, con esto se deshabilita el filtro de reconocimiento de direcciones y no se toma en cuenta el valor de los bits CONMC y BCREJ.

Para aplicar el filtro por dirección multicast los grupos permitidos deben estar especificados en una tabla de direcciones. Para este fin existe un registro de 64 bits en donde se mapean las direcciones multicast aplicando un algoritmo de codificación sobre los 48 bits de la dirección MAC, del resultado obtenido se seleccionan los 6 bits más significativos. El valor resultante es un número entre el 0 y el 63. En la posición del registro MCHASH (*Tabla Multicast*, o *tabla HASH*) correspondiente al número calculado se escribe el valor de 1 para indicar que esa dirección está permitida.

Filtro ETHERTYPE

El filtrado por tipo de paquete se hace basado en el contenido del campo *tipo/longitud* de los paquetes recibidos. Dicho campo puede contener la longitud del paquete o un código que indica de qué tipo de paquete se trata. Los tipos, o *ethertype*, indican qué protocolo de la capa superior va a procesar el paquete. Existen diversos códigos para los diferentes protocolos que trabajan sobre la capa de enlace, pero los protocolos que nos interesan son el Protocolo de Internet versión 4, cuyo código corresponde a 0x0800, y el Protocolo de Resolución de Direcciones, cuyo código ethertype es 0x0806.

El registro en donde se configuran los valores ethertype permitidos se llama *Registro de Control Ethertype* (ETCTL) y se ilustra en la Figura 3.16. Al escribir un 1 en cualquiera de sus bits se le permite al receptor aceptar paquetes con el código indicado.

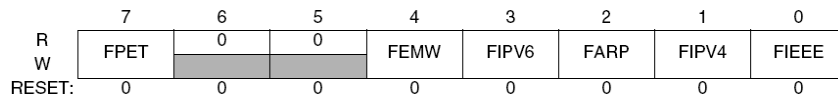


Figura 3.16 Registro de control Ethernethype ETCTL

Se permite la recepción de algún otro ethertype no indicado en el registro, escribiendo un 1 en el bit FPET (*Filtro de Ethernethype programable*). El valor del código correspondiente se escribe en el registro *Ethernethype Programable* (ETYP), que es un registro de 16 bits.

También es posible filtrar la recepción de paquetes que incluyan en el campo *tipo/longitud* la longitud del campo de datos en lugar del código ethertype. Para esto se utiliza el bit FIEEE, si se le escribe un 1 los paquetes con valor entre 0x0000 y 0x05DC en el campo *tipo/longitud* son aceptados. Si se escribe un 0 en dicho bit, estos paquetes son rechazados.

Buffers de recepción y transmisión

Para almacenar las tramas recibidas y las que se van a transmitir se utilizan buffers (registros temporales) de recepción y transmisión. El módulo EMAC cuenta con dos buffers de recepción (*Buffer de Recepción A* y *Buffer de Recepción B*) y uno de transmisión, localizados dentro de la memoria RAM.

El tamaño de los buffers de recepción y transmisión es configurable y el valor máximo corresponde a la longitud máxima de una trama Ethernet. La elección del tamaño determina también la cantidad de memoria RAM disponible para las aplicaciones de usuario. Por ejemplo, si se escoge el tamaño de 512 bytes, la cantidad de RAM utilizada por los tres buffers abarca un total de 1.5 kbytes, por lo que restan 6.5 Kbytes de RAM (ya que el microcontrolador cuenta con 8 Kbytes de memoria RAM). La selección del tamaño de los buffers debe considerar las necesidades de la aplicación. Una configuración de tamaño máximo sólo deja disponibles 3.5 Kbytes de RAM para las aplicaciones, mientras que una configuración menor ignora las tramas que no quepan en el buffer. El tamaño de los buffers se ajusta en el *Registro de Configuración de Buffers Ethernet* (BUFCFG) que se ilustra en la Figura 3.17.

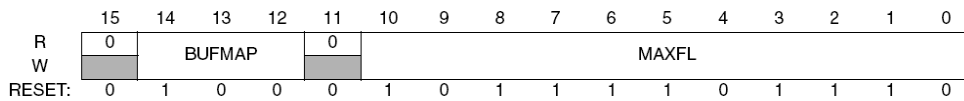


Figura 3.17 Registro de Configuración de Buffers Ethernet

En la Tabla 3.4 se muestran los valores que configuran el tamaño de los buffers, el cual se indica con los bits del campo BUFMAP, su localización dentro de la memoria RAM y la cantidad de RAM disponible para cada configuración.

Cuando una trama es recibida se almacena en el buffer de recepción que tenga espacio disponible y se genera una interrupción que le avisa al procesador que han llegado nuevos datos. Si ambos buffers están llenos el paquete se rechaza. Una vez que el paquete se almacena o se procesa, se debe limpiar la bandera de interrupción para que el buffer pueda recibir un nuevo paquete.

Tabla 3.4 Configuración del espacio de Buffers y memoria RAM

BUFMAP	Tamaño de los Buffers (Bytes)	Espacio en memoria del Buffer RXA	Espacio en memoria del Buffer RXB	Espacio en memoria del Buffer TX	Cantidad de memoria RAM restante
0	128	0x0000 – 0x007F	0x0080 – 0x00FF	0x0100 – 0x017F	7.625 K
1	256	0x0000 – 0x00FF	0x0100 – 0x01FF	0x0200 – 0x02FF	7.25 K
2	512	0x0000 – 0x01FF	0x0200 – 0x03FF	0x0400 – 0x05FF	6.5 K
3	1K	0x0000 – 0x03FF	0x0400 – 0x07FF	0x0800 – 0x0BFF	5 K
4	1.5K	0x0000 – 0x05FF	0x0600 – 0x0BFF	0x0C00 – 0x11FF	3.5 K
5-7		Reservado			

Receptor y transmisor EMAC

El receptor EMAC trabaja con poca intervención del CPU, una vez que el módulo EMAC se habilita el receptor comienza a procesar las tramas entrantes. Lo primero que hace al recibir un nuevo paquete es verificar la secuencia de bits del preámbulo y la secuencia del delimitador de inicio. Si los bits de estos campos no son correctos el receptor rechaza la trama y deja de recibir datos, en caso contrario continúa con la recepción hasta que la dirección de destino se almacena en el buffer. Al tener la dirección de destino aplica los filtros de reconocimiento de dirección que hayan sido configurados. Si la dirección no cumple alguna condición, la trama se rechaza y se interrumpe la recepción de datos. Si la trama es aceptada se verifica el contenido del campo *tipo/longitud*, si el valor no cumple con alguno de los requisitos configurados previamente se rechaza.

Cuando existe actividad en el receptor EMAC el bit de *Estatus Activo*, o RXACT, del registro RXCTS ilustrado en la Figura 3.12, permanece en 1. Al terminar el procesamiento de la trama recibida el bit se pone a 0, lo cual indica que el receptor se encuentra en un estado de inactividad.

Para transmitir una trama se coloca en el *Buffer de Transmisión* la dirección de destino, la de origen, el tipo o longitud de la trama y los datos. El transmisor genera los bits del preámbulo y el delimitador de inicio, calcula la secuencia de verificación y escribe los ceros de relleno en el campo de datos si la longitud no cubre el valor mínimo. Para iniciar la transmisión se debe escribir el comando START en el campo TCMD del registro TXCTS ilustrado en la Figura 3.13.

El bit TXACT del registro TXCTS indica que hay actividad en el transmisor, se pone a 1 después de escribir un comando de transmisión válido en el campo TCMD y se pone a 0 cuando se termina de enviar la trama, indicando que el transmisor se encuentra inactivo. No se permite iniciar una transmisión mientras en bit TXACT tenga el valor de 1.

Los comandos que maneja el transmisor Ethernet son valores de dos bits que se escriben en el campo TCMD. Hay tres comandos de transmisión: START, PAUSE y ABORT. El comando START inicia la transmisión de la trama almacenada en el *Buffer de Transmisión*. El comando PAUSE inicia la transmisión de una trama de pausa y el comando ABORT detiene una transmisión en progreso. En la Tabla 3.5 se muestra el código de cada comando del transmisor.

Tabla 3.5 Comandos del Transmisor Ethernet

TCMD	Comando	Descripción
0	Reservado	Ignorado
1	START	Transmite la trama en el buffer
2	PAUSE	Transmite una trama de pausa (en modo Full-Duplex)
3	ABORT	Detiene la transmisión

Interfaz independiente del medio (MII) y manejador MII

El manejador MII (*Medium Independent Interface*) sirve para comunicar al módulo EMAC con el módulo EPHY a través de la MII. El módulo EPHY es el encargado de la conversión de los datos en señales físicas y viceversa utilizando *codificación Manchester* y de su envío y recepción a través del medio físico. El EMAC envía información para su transmisión hacia el EPHY y recibe los datos que llegan de la red a través de él. El EMAC y el EPHY también intercambian información para leer y escribir datos de los registros internos del módulo EPHY.

La *Interfaz Independiente del Medio* hace uso de 18 señales. Las funciones de recepción y transmisión requieren 7 señales cada una (4 señales de datos, un delimitador, una señal de error y una señal de reloj). Para conocer el estado del medio físico se utilizan dos señales, una indica la presencia de una portadora (la portadora indica que el medio de transmisión está libre y puede ser usado) y la otra indica cuando ocurre una colisión. El manejador MII utiliza las dos señales restantes. La señal MII_MDIO se usa para intercambiar datos en forma serial sincrona entre los módulos EPHY y EMAC, utilizando la señal MII_MDC como reloj. Estas señales también están disponibles en modo externo.

MDC es el reloj del manejador y su frecuencia se ajusta con los bits MDCSEL del registro de *Estatus y Comandos del Manejador MII*, ilustrado en la Figura 3.18, antes de realizar cualquier comunicación entre

el EMAC y el EPHY. El valor de la frecuencia se obtiene a partir del reloj de bus de acuerdo a la siguiente expresión:

$$\text{Frecuencia MDC} = \frac{\text{Frecuencia de bus}}{2 \text{ MDCSEL}}$$

La frecuencia del reloj MDC debe ser menor o igual a 2.5 MHz para cumplir con las especificaciones de la interfaz MII de la IEEE 802.3. Recordando que la frecuencia de bus necesaria para la operación del controlador Ethernet se debe ajustar a 25 MHz, el valor necesario para MDCSEL corresponde a 5.

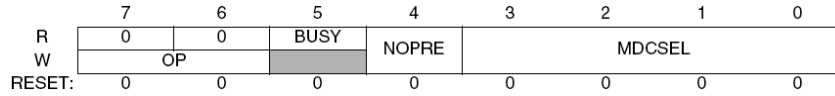


Figura 3.18 Registro de Estatus y Comandos del Manejador MII, MCMST

Se definen dos funciones para comunicarse a través del manejador MII, la operación de lectura y la operación de escritura. Para realizar una función de lectura se escribe el comando 10 (binario) en el campo OP (código de operación) del registro MCMST. El resultado de la lectura es un valor de 16 bits que se almacena en el registro *Datos Leídos del Manejador MII*, (MRDATA). La función de escritura se realiza escribiendo el comando 01 (binario) en el campo OP del registro MCMST. El dato que se desea escribir debe almacenarse previamente en el registro *Dato de Escritura del Manejador MII* (MWDATA), que es un registro de 16 bits.

El bit BUSY del registro MCMST indica actividad en el manejador MII, mientras este bit esté a 1 no se permite realizar funciones de lectura o escritura. Cuando el manejador completa la transferencia de datos entre el EMAC y el EPHY, el bit BUSY se pone a 0 indicando que puede ser solicitada otra operación.

3.2.4.2 Módulo EPHY – Transceptor físico ETHERNET

El *transceptor físico Ethernet* (EPHY) del MC9S12NE64 funciona de acuerdo a los criterios de la norma IEEE 802.3 para operación a 10 y 100 Mbps (*10BASE-T/100BASE-TX*). Requiere un cristal de 25 MHz para su operación.

El EPHY provee codificación y decodificación digital/analógica, lo cual permite la conexión del sistema digital formado por el microcontrolador, con el cable UTP que maneja señales analógicas. El diagrama de bloques del módulo se observa en la Figura 3.19 y Figura 3.20.

El EMAC se puede conectar a un dispositivo PHY externo a través de uno de los puertos del microcontrolador o al módulo EPHY interno. En el *registro de Control de Red*, ilustrado en la Figura 3.11, se configura el EMAC en modo de PHY externo escribiendo un 1 al bit EXTPHY. El módulo EMAC soporta la conexión a 32 dispositivos PHY, para lo cual cada dispositivo se identifica con una dirección de cinco bits. Si se usa el PHY interno su dirección se configura en el registro *1 de Control del Transceptor Físico Ethernet*.

La mayoría de las configuraciones del EPHY se hacen a través de sus registros internos, a los cuales se accede por medio de la interfaz MII (*Medium Independent Interface*). Los registros internos del EPHY no son accesibles directamente ni son visibles desde el mapa de registros del microcontrolador. Los registros del módulo EPHY que sí forman parte del mapa de registros del microcontrolador son los siguientes:

- Registro 0 de Control del Transceptor Físico Ethernet (EPHYCTL0)
- Registro 1 de Control del Transceptor Físico Ethernet (EPHYCTL1)
- Registro de Estatus del Transceptor Físico Ethernet (EPHYSR)

En la Figura 3.21 se muestran los registros EPHYCTL0 y EPHYCTL1 que se usan para configurar algunas opciones del módulo y para habilitarlo. Las configuraciones que se pueden ajustar desde los registros del microcontrolador son: la dirección del EPHY, habilitar los relojes para operaciones de 10 y 100 Mbps, auto negociación y habilitar interrupciones PHY (EPHYIEN).

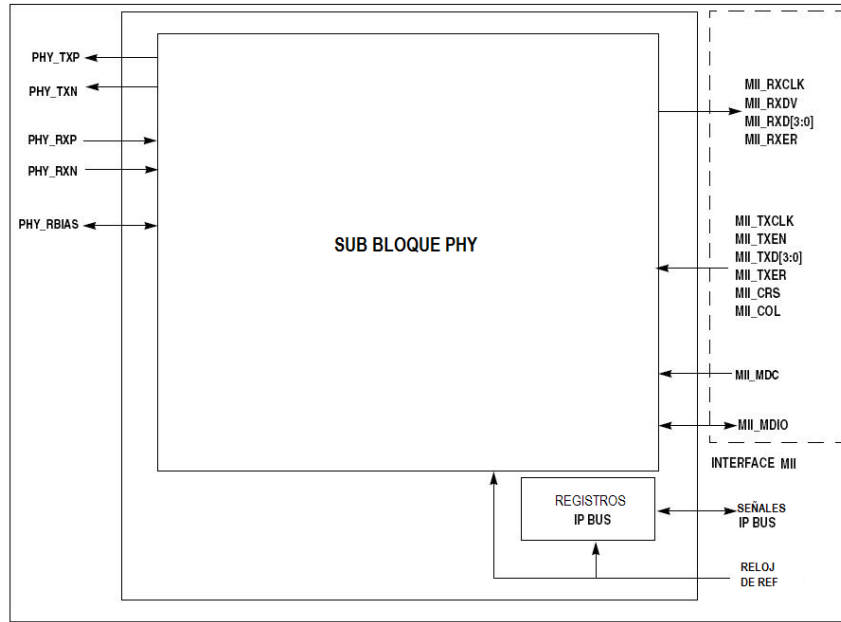


Figura 3.19 Diagrama de bloques del transceptor físico Ethernet

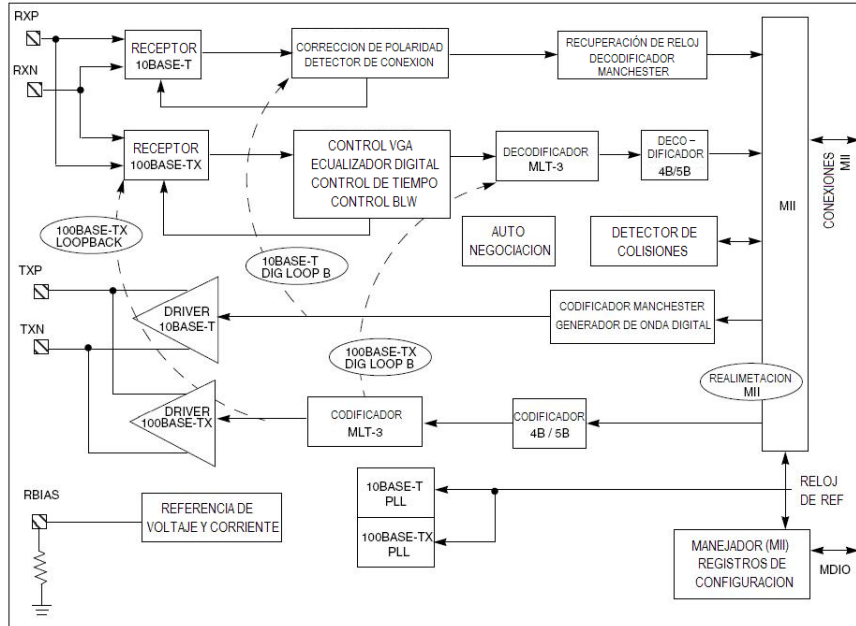


Figura 3.20 Sub bloque del transceptor físico Ethernet

Nombre del Registro	Bit 7	6	5	4	3	2	1	Bit 0
Ethernet Physical Transceiver Control Register 0 (EPHYCTL0)	R W	EPHYEN	ANDIS	DIS100	DIS10	LEDEN	EPHYWAI	0 W
Reset:		0	1	1	1	0	0	0
Ethernet Physical Transceiver Control Register 1 (EPHYCTL1)	R W	0	0	0	PHYADD 4	PHYADD 3	PHYADD 2	PHYADD 1 0
Reset:		0	0	0	0	0	0	0

Figura 3.21 Registros de control EPHY

	7	6	5	4	3	2	1	0
R	0	0	100DIS	10DIS	0	0	0	EPHYIF
W								
RESET:	0	0	1	1	0	0	0	0

Figura 3.22 Registro de Estatus EPHY, EPHYSR

En el registro EPHYSR se lee el estado del módulo EPHY. Se puede leer la velocidad a la que está operando el controlador, con los bits 100DIS y 10DIS, y la bandera de interrupción del módulo. En la Figura 3.22 se ilustra el registro EPHYSR.

Interrupciones

Tanto el módulo EMAC como el módulo EPHY cuentan con servicios de interrupción para diferentes eventos que se puedan presentar. Algunos eventos que pueden generar interrupciones en el módulo EMAC son:

- Recepción de trama de pausa
- Error en recepción de tramas
- Recepción de tramas válidas
- Fin de transmisión de trama

Algunos eventos que generan interrupción en el módulo EPHY son los siguientes:

- Recepción de un bit de confirmación
- Cambio en el estado de la conexión
- Cambio en parámetros de autonegociación
- Falla remota

3.2.4.3 Inicialización y configuración del controlador ETHERNET

El controlador Ethernet del microcontrolador se configura para seleccionar el modo de operación y se debe inicializar para poderse conectar a una red. Si se usa el transceptor físico interno ambos módulos deben inicializarse juntos. Se debe empezar por configurar el EPHY desde los registros accesibles desde el mapa de registros del microcontrolador. Luego se inicializa y configuran las opciones del módulo EMAC y se inicializa la comunicación MII. Una vez hecho esto se termina de configurar el módulo EPHY usando sus registros internos, a través de las operaciones de lectura y escritura MII desde el módulo EMAC.

La secuencia de inicialización para habilitar el controlador de red se enumera a continuación [16e].

1. Inicializar el módulo CRG para generar una frecuencia de bus de 25 MHz.
2. Deshabilitar los relojes de 10 y 100 Mbps del módulo EPHY, escribiendo 1 en los bits DIS10 y DIS100 (los relojes se habilitan hasta que el EMAC y el EPHY se configuren completamente).
3. Configurar la dirección del PHY interno en el registro EPHYCTL1.
4. Configurar modo de auto negociación (si no se requiere, se escribe 1 en el bit ANDIS en el registro EPHYCTLO, si se usa se escribe 0 a ese bit).
5. Habilitar las interrupciones PHY con el bit EPHYIEN del registro EPHYCTLO.
6. Habilitar módulo EPHY con el bit EPHYEN del registro EPHYCTLO.
7. Esperar el retardo $t_{\text{start-up}}$ (Figura 3.23).
8. Configurar el valor MDCSEL para generar la frecuencia del reloj MDC, en el registro MCMST.
9. Configurar el espacio de los buffers de recepción y transmisión usando el valor BUFMAP del registro BUF CFG.
10. Configurar la longitud máxima de las tramas de recepción, con los bits MAXFL del registro BUF CFG.
11. Configurar la dirección física MAC en el registro MACAD.
12. Configurar el filtro ephertype usando los registros ETYPE y ETCTL.
13. Configurar el filtro de direcciones MAC en el registro RXCTS.
14. Configurar el modo de control de flujo de recepción, con el bit RFCE del registro RXCTS.
15. Configurar modo de *loopback* y PHY externo, si son requeridos, en el registro de *control de red*, NETCTL.
16. Si no se usa auto negociación, configurar el modo Full-Duplex con el bit FDX del registro NETCTL. Si se usa auto negociación, al completarse se debe revisar la configuración acordada (en los registros internos PHY) y ajustar el valor correspondiente para el bit FDX.

17. Habilitar el módulo EMAC con el bit EMACE del registro NETCTL.
18. Configurar y habilitar las interrupciones EMAC usando el registro IMASK.
19. Inicializar y transmitir una trama de pausa.
20. Habilitar las interrupciones del sistema.
21. Configurar módulo EPHY a través del manejador MII (configurar velocidad, modo dúplex, control de flujo y anuncio de habilidades de auto negociación).
22. Configurar las interrupciones EPHY en el registro interno de control de interrupciones.
23. Iniciar los generadores de reloj EPHY
 - Si se usa auto negociación, encender los generadores de reloj EPHY escribiendo ceros en los bits DIS10 y DIS100 del registro EPHYCTL0.
 - Si no se usa auto negociación: configurar la velocidad y el modo dúplex con los bits DPLX y DATARATE del registro interno de control EPHY. Encender los generadores de reloj EPHY, escribiendo ceros en los bits DIS10 y DIS100 del registro EPHYCTL0.
24. Si se usa auto negociación, en cuanto se establece una conexión y se completa la auto negociación se leen de los registros internos las configuraciones establecidas y se actualiza el módulo EMAC ajustando el valor de los bits RFCE y FDX.

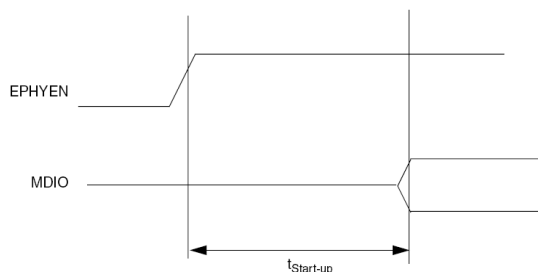


Figura 3.23 Tiempo de retardo de inicialización PHY

3.2.5 Conexiones básicas del microcontrolador

El circuito mínimo para el funcionamiento del microcontrolador incluye: fuente de alimentación, circuito de reset, cristal, capacitores de filtrado para terminales de alimentación, circuito PLL y resistencias de configuración. Adicionalmente se conectan leds de estatus para las funciones de comunicación por red y para la alimentación del sistema.

Sistema de alimentación

El sistema de desarrollo utiliza dos circuitos de alimentación, uno de 3.3 volts para el microcontrolador y uno de 5 volts que alimenta al convertidor USB y otros dispositivos. El circuito convertidor serie a USB está configurado en modo *bus-power*, es decir, que la alimentación necesaria para su funcionamiento la toma de la computadora a la que se conecta a través de un puerto USB. Se pueden aprovechar los 5 volts que provee la computadora para alimentar al circuito del microcontrolador. Por otro lado, cuando no es necesaria la comunicación del sistema con una computadora, éste debe ser capaz de trabajar a partir de una fuente de alimentación externa, que debe suministrar la alimentación al circuito del microcontrolador y a otros dispositivos que se le pudieran conectar (sin alimentar al circuito convertidor USB). En la Figura 3.24 se muestra el circuito que provee la alimentación necesaria para el sistema de desarrollo.

La fuente de alimentación externa se conecta en la terminal 5 de un *header*, mientras que la fuente de 5 volts que entrega la computadora se conecta en la terminal 1. Para seleccionar la fuente de alimentación del sistema se utiliza un jumper que conecta tres de las cinco terminales del *header*, ya sea las terminales 1 2 y 3 para seleccionar la fuente de la computadora, o las terminales 3 4 y 5 para seleccionar la fuente externa.

Cuando el sistema se alimenta con la fuente externa, ésta se conecta a un regulador de 3.3 volts y otro de 5 volts para generar la alimentación del circuito del microcontrolador y la alimentación de 5 volts para otros dispositivos. Si se requiere comunicar al sistema con una computadora, se conecta el cable USB sin que éste interfiera con la alimentación de la fuente externa, sólo alimenta al convertidor USB.

Cuando se requiera que todo el sistema trabaje a partir del voltaje entregado por la computadora el jumper se pasa a la posición de los pines 1 2 y 3, lo que provoca que la fuente externa quede desconectada del sistema, el voltaje del microcontrolador se obtiene a partir del regulador de 3.3 volts alimentado por el

voltaje del puerto USB de la computadora, y la entrada del regulador de 5 volts queda desconectada, obteniendo los 5 volts para otros circuitos de la alimentación de la computadora.

Si el jumper se coloca en las terminales 2 3 y 4, ambas fuentes de alimentación quedan desconectadas y el sistema queda apagado.

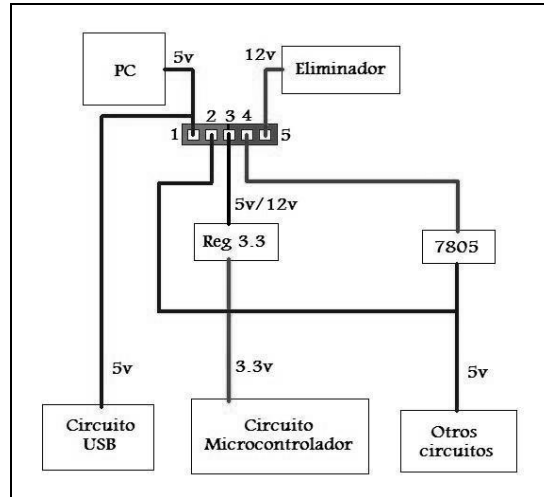


Figura 3.24 Circuito de alimentación para el microcontrolador

Circuito de reset

El circuito de reset genera un reinicio del estado del microcontrolador por hardware cuando se introduce un valor bajo en la terminal de reset del MC9S12NE64. Para esto se conecta un *push button* en un circuito como el que se muestra en Figura 3.25. Cuando el sistema se energiza el interruptor de RESET se encuentra abierto y la corriente circula a través de R_4 y C_3 , cargando el capacitor C_3 desde 0 hasta 3.3 volts con una constante de tiempo igual a $\tau = R_4 * C_3 = 27000 * 0.22 \times 10^{-6} = 5.94$ ms. Cuando se presiona el botón de reset el capacitor se descarga a través de R_2 con una constante de tiempo igual a $\tau = (R_2 * R_4 * C_3) / (R_2 + R_4) = (47 * 27000 * 0.22 \times 10^{-6}) / (47 + 27000) = 10.3 \mu s$, tiempo en el cual el capacitor alcanza un valor de voltaje que se puede considerar como cero lógico para generar el reset del microcontrolador y con un poco más de tiempo alcanza el valor de cero. Los valores de los elementos del circuito los propone el fabricante del microcontrolador para que cumplan con las especificaciones de la señal de reset que necesita el microcontrolador.

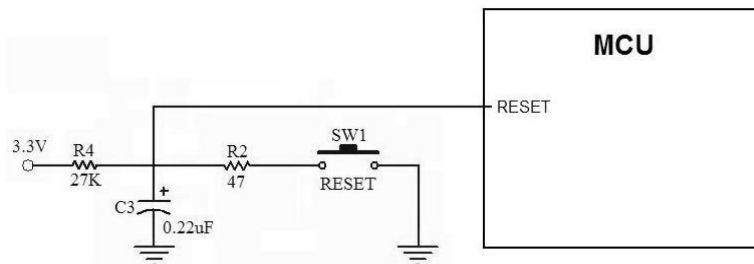


Figura 3.25 Circuito de reset para el microcontrolador

Cristal

El cristal para el microcontrolador es de 25 MHz con una tolerancia de 25 ppm (partes por millón) como especifica la IEEE 802.3 y se conecta entre las terminales XTAL y EXTAL con capacitores de 15 a 30 pF y una resistencia de 10MΩ como se muestra en la Figura 3.26, para generar una oscilación estable.

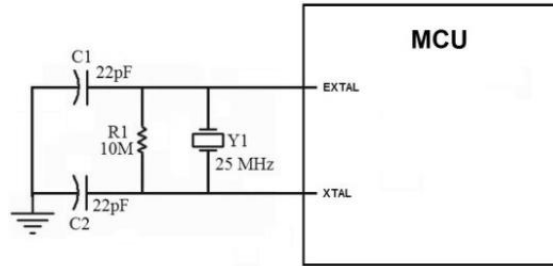


Figura 3.26 Conexión del cristal del microcontrolador

Leds indicadores de estado de la red

El puerto L del MC9S12NE64 tiene dos funciones: 1) como puerto de entrada y salida de datos y 2) como manejo de señales del estado de la conexión. A estas terminales se pueden conectar leds indicadores de estado. El módulo EPHY se puede configurar para que maneje de manera automática los leds de estado, escribiendo un 1 al bit LEDEN del registro EPHYCTL0. Las señales de conexión que maneja el microcontrolador son las que se muestran en la Tabla 3.6.

Tabla 3.6 Señales del estado de la conexión

Nombre de la Terminal	Señal	Descripción
COLLED	Colisión	Destella cuando ocurre una colisión (en modo Half-Duplex)
DUPLED	Duplex	Indica si la conexión es Full-Duplex (led encendido) o Half-Duplex (led apagado)
SPDLED	Velocidad	Indica la velocidad de la conexión, 10 Mbps (led apagado) ó 100 Mbps (led prendido)
LINKLED	Conexión	Indica si se ha establecido una conexión con otro dispositivo
ACTLED	Actividad	Destella cuando el controlador está recibiendo datos

La conexión de estos leds se muestra en la Figura 3.27:

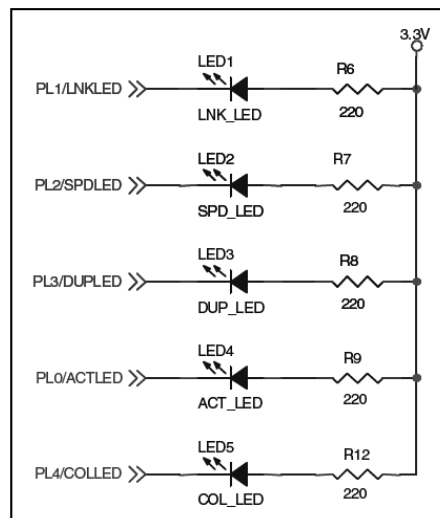


Figura 3.27 Conexión de los leds indicadores de estado

3.3 Puertos de comunicación

Los diferentes puertos de comunicación con los que cuenta el sistema de desarrollo se describen a continuación.

3.3.1 Interfaz de comunicación serie a USB

El sistema de desarrollo cuenta con una interfaz USB para comunicarse con una computadora. El uso del *Bus Universal Serial* (USB) se ha extendido desde su aparición por lo que resulta práctica la utilización de esta interfaz en nuestro sistema. Para implementar la comunicación vía USB se utiliza el convertidor serie a USB, FT232 del fabricante *FTDI Chip*, conectado al módulo de comunicación serial asincrónica SCIO del MC9S12NE64. El otro módulo serie se deja libre para uso de las aplicaciones. En la Figura 3.28 se muestra un diagrama de bloques de la interfaz y en la Figura 3.29 el diagrama de la distribución de terminales y un diagrama esquemático del circuito integrado.

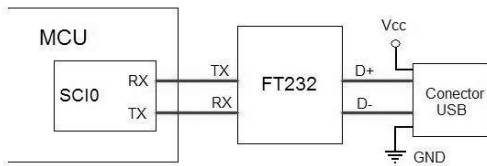


Figura 3.28 Conexión de la interfaz serie-USB

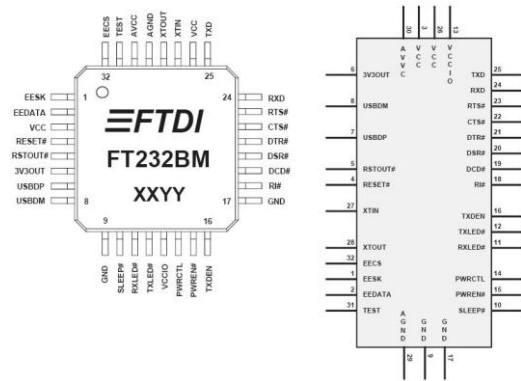


Figura 3.29 Encapsulado y esquemático del convertidor FT232

En la Tabla 3.7 se muestra la descripción de las señales del convertidor que son de interés para este sistema de desarrollo.

Tabla 3.7 Descripción de las señales del FT232

No. De Terminal	Señal	Tipo	Descripción
SEÑALES DE INTERFAZ UART			
25	TXD	Salida	Salida de transmisión asíncrona de datos
24	RXD	Entrada	Entrada de recepción asíncrona de datos
23	RTS#	Salida	Solicitud de envío / señal de handshake
22	CTS#	Entrada	Libre para envío / señal de handshake
21	DTR#	Salida	Terminal de datos listo / señal de handshake
20	DSR#	Entrada	Equipo de datos listo / señal de handshake
19	DCD#	Entrada	Detección de portadora
18	RI#	Entrada	Indicador de tono
SEÑALES DE INTERFAZ USB			
7	USBDP	E/S	Señal de datos positiva (D+)
8	USBDM	E/S	Señal de datos negativa (D-)
14	PWRCTL	Entrada	Modo bus power conectar a tierra, modo self power conectar a Vcc
OTRAS SEÑALES			
4	RESET	Entrada	Entrada para generar un reset, si no se requiere se conecta a Vcc
5	RSTOUT	Salida	Salida del generador de reset interno
27	XTIN	Entrada	Entrada del cristal Oscilador de 6 MHz

28	XTOUT	Salida	Salida del cristal Oscilador de 6 MHz
31	TEST	Entrada	Pone al dispositivo en modo de prueba, si se conecta a tierra opera en modo normal
SEÑALES DE ALIMENTACIÓN Y TIERRA			
6	3V3OUT	Salida	Salida de 3.3 V del regulador integrado
3, 26	VCC	PWR	Alimentación del núcleo, de +4,4 a +5.25 V
13	VCCIO	PWR	Alimentación para las terminales de la interfaz UART, de +3.0 a +5.25
9,17	GND	PWR	Tierra
30	AVCC	PWR	Alimentación analógica para el multiplicador de reloj interno
29	AGND	PWR	Tierra analógica para el multiplicador de reloj interno

Las señales de la interfaz UART utilizadas para implementar el protocolo de comunicación RS232 no se implementan en este sistema de desarrollo, simplemente se utilizan las señales de recepción y transmisión de datos, correspondientes a las terminales 24 (RXD) y 25 (TXD), se conectan a las correspondientes señales de transmisión y recepción, respectivamente, del módulo de comunicación serial asíncrona SCIO del MC9S12NE64. Estas dos señales son la única conexión requerida entre el convertidor y el microcontrolador.

El convertidor serie a USB puede tomar la alimentación Vcc del puerto USB de la computadora, a lo cual se denomina modo *Bus Power*. Si el circuito cuenta con su propia fuente de alimentación se llama modo *Self Power*. Con la terminal 14 (PWRCTL) se selecciona el modo de alimentación del circuito USB. Si se conecta a tierra el modo de alimentación es *Bus Power*, si se conecta a Vcc el modo es *Self Power*. Para el caso de este sistema, el circuito del convertidor serie a USB funciona en modo *Bus Power*, por lo tanto la terminal 14 se conecta a tierra.

El circuito de interfaz USB se muestra en la Figura 3.30, la configuración se tomó de la hoja de datos técnicos del convertidor FT232BM USB UART (obtenida de <http://www.ftdichip.com>). El FT232 requiere un cristal de 6 MHz conectado entre las terminales 27 y 28, con capacitores de 27 pF en cada terminal. Se requiere un inductor de ferrita en serie con la alimentación USB para prevenir ruido e interferencias electromagnéticas desde el dispositivo hacia la computadora. En la terminal 13 (VCCIO), se conecta el nivel deseado para las señales serie (RXD y TXD) que genera el circuito; Estas señales son las que se conectan con el puerto serie del microcontrolador. Se pueden generar señales de 5 ó 3.3 V, debido a que el microcontrolador maneja voltajes de 3.3, la terminal 13 se conecta a la fuente de 3.3V.

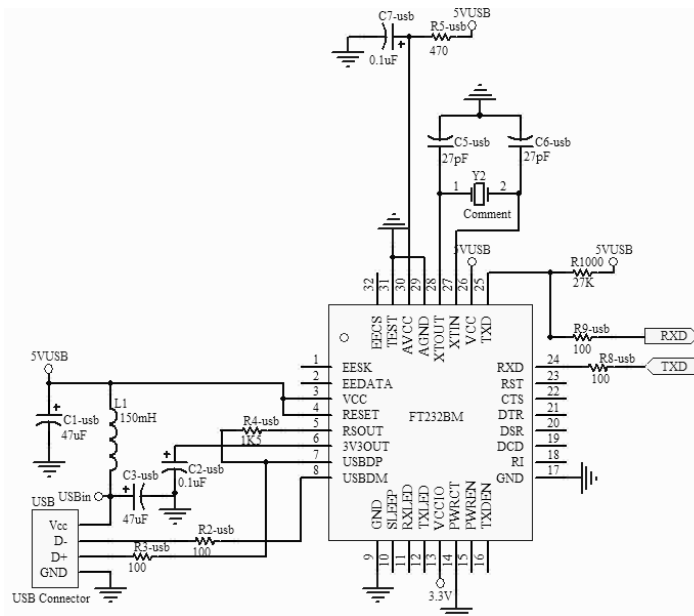


Figura 3.30 Circuito del convertidor serie a USB

Para que el convertidor FT232BM pueda comunicarse con una computadora es necesario instalar el programa controlador o *driver*. El controlador para el FT232BM se puede obtener de manera gratuita en la página del fabricante (<http://www.ftdichip.com>) y se encuentra disponible para sistemas operativos a partir de Windows XP (En este momento se utiliza el software CDM20416). Al instalar el software controlador en la computadora el dispositivo USB conectado a ella se identifica como un puerto.

3.3.2 Interfaz de comunicación Ethernet

Como se mencionó en el tema 3.2.4 la tecnología de comunicación Ethernet ha tenido variantes a lo largo de su evolución, como los modos de operación 10BaseT y 100BaseT que integra el módulo de comunicación Ethernet del MC9S12NE64, para dichos modos la norma IEEE 802.3 especifica el uso de cable de par trenzado (UTP, ver tema 2.6).

El cable de par trenzado consiste en pares de conductores de alambre de cobre, aislados entre sí y entrelazados, con la finalidad de disminuir la interferencia electromagnética de fuentes externas. Las señales de cada par son iguales y opuestas en polaridad, por lo tanto para transmitir datos se utiliza un par de conductores que guían las señales Tx+ y Tx- y otro par para las señales de recepción de datos Rx+ y Rx-. El cable de par trenzado se conecta al sistema de desarrollo utilizando un conector RJ45.

El módulo EPHY del microcontrolador es el encargado de enviar y recibir las señales hacia y desde el medio físico, constituido por el cable de par trenzado. Para esto el microcontrolador cuenta con las terminales PHY_RXN, PHY_RXP, PHY_TXN y PHY_TXP que manejan las señales Rx-, Rx+, Tx- y Tx+, respectivamente.

Los pares de señales de recepción y transmisión del microcontrolador deben estar aislados del medio físico mediante un transformador de aislamiento magnético de alta velocidad para redes LAN, como se aprecia en el diagrama de la Figura 3.31. El transformador magnético puede estar integrado con el conector RJ45 o ser independiente. La relación de transformación tanto para las señales RX como para las TX debe ser 1:1.

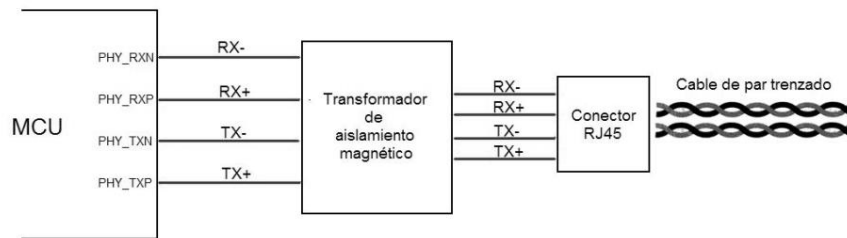


Figura 3.31 Conexión del transformador de aislamiento y el conector RJ45

El circuito de interfaz Ethernet que utiliza este sistema de desarrollo corresponde a un conector RJ45 con transformador de aislamiento magnético integrado, de la empresa Midcom (<http://www.midcom-inc.com>) con clave 74990111215. Se elige este conector ya que es económico, de fácil adquisición y cumple con los parámetros especificados por el fabricante del microcontrolador, mismos que se muestran en la Tabla 3.8.

Tabla 3.8 Especificaciones de Transformador de aislamiento magnético

Parámetro	Valor	Unidades	Condiciones de prueba
Tx/Rx turns ratio	1:1 CT / 1:1	—	—
Inductance	350	mH (min)	—
Insertion loss	1.1	dB (max)	1 to 100 MHz
Return loss	-18	(min)dB	1 to 30 MHz
	-14	(min)dB	30 to 60 MHz
	-12	(min)dB	60 to 80 MHz
Differential to common mode rejection	-40	dB (min)	1 to 60 MHz
	-30	dB (min)	60 to 100 MHz
Transformer	1500	V	—

La conexión del conector RJ45 corresponde al diagrama de la Figura 3.32. El módulo Ethernet del microcontrolador MC9S12NE64 tiene una terminal para conectar una resistencia de polarización bias, llamada PHY_RBIAS. Esta terminal necesita la conexión de una resistencia de precisión con un valor de 12.4K entre la terminal PHY_RBIAS y tierra, lo más cerca posible del microcontrolador.

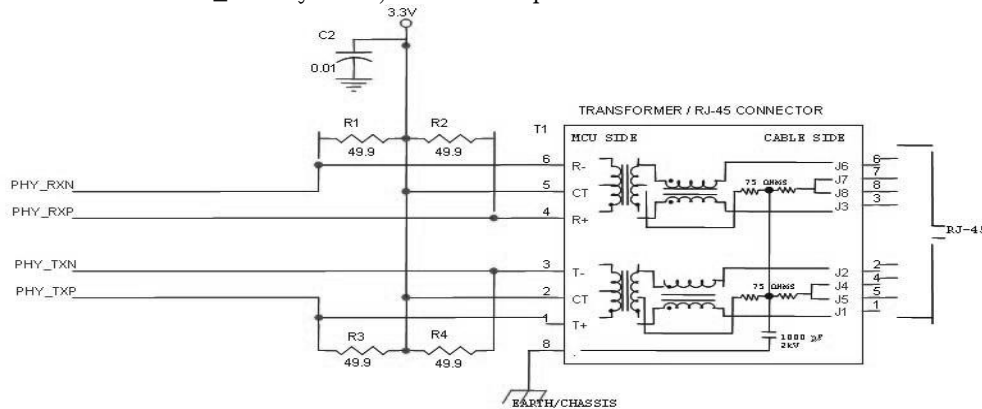


Figura 3.32 Conexión del transformador de aislamiento con el conector RJ45

3.3.3 Interfaz de programación y depuración BDM (PYDBDM)

El sistema debe poseer algún medio para realizar la programación del microcontrolador. Para este fin el microcontrolador MC9S12NE64, al igual que otros productos de *Freescale Semiconductors*, como es el caso del las familias HCS08, HC(S) 12(X) y RS08, cuenta con un módulo de depuración en segundo plano o BDM (*Background Debug Module*).

El módulo BDM permite la depuración y programación de un microcontrolador que se encuentra dentro de una tarjeta de aplicación, reemplazando a un emulador en circuito por hardware. El emulador en circuito utiliza componentes externos para duplicar la operación del microcontrolador, pero el uso de estos componentes externos puede representar una fuente de posibles errores, como fallos de conexión o compatibilidad de hardware, por tal motivo resulta conveniente el uso de un depurador lógico, como el que se genera con el módulo DBM.

El depurador lógico está integrado en un módulo del microcontrolador localizado fuera del CPU, lo cual permite que se lleven a cabo operaciones de lectura y escritura de localidades de memoria de manera autónoma, mientras que el CPU ejecuta código de usuario, de esta manera no se afecta el tiempo de operación del código en ejecución.

Cuando el BDM se activa toma control sobre el microprocesador permitiendo su depuración. Puede estar habilitado pero no activado, como es el caso en el que puede leer y escribir en localidades de memoria sin interrumpir la ejecución de algún código de usuario.

El módulo BDM se manipula a través de una computadora, desde la cual recibe y ejecuta comandos. Existen dos tipos de comandos BDM: comandos hardware y comandos firmware. Con comandos Hardware el BDM se habilita pero no se activa y el código de usuario puede permanecer en ejecución, permitiéndose la accesibilidad a todas las localidades de memoria incluidas las operaciones entrada/salida y el control de registros en memoria. Con comandos Firmware el modulo BDM se habilita y se activa. En dicho estado el BDM toma control del CPU, permitiendo la lectura y escritura de recursos del CPU como son el acumulador, registro X e Y, *stack pointer* y contador de programa.

Para poder comunicar al microcontrolador con la computadora a través del módulo BDM se requiere:

- Una computadora con una versión del software Code Warrior pertinente para el microcontrolador utilizado (para el MC9S12NE64 la versión 5.9 o superior). Este es el software desde donde se envían los comandos hacia el microcontrolador.
- Tarjeta de desarrollo con microcontrolador MC9S12NE64 y puerto BDM.
- Programador y depurador BDM (PYDBDM). Dispositivo que sirve de interfaz entre el microcontrolador y la computadora. En uno de sus extremos se conecta al módulo BDM del

microcontrolador a través de un conector de seis terminales y en el otro extremo se conecta a una computadora a través de un puerto USB.

El PYDBDM que se emplea en este desarrollo es el *USB Multilink Interface* de la compañía *P&E Microcomputer Systems*. En la Figura 3.33 se ilustra dicho dispositivo y en la Figura 3.34 se muestra la conexión entre el microcontrolador y la computadora a través del PYDBDM.



Figura 3.33 Interfaz Programador Y Depurador BDM

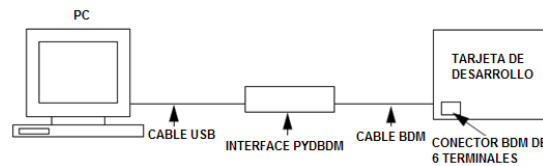


Figura 3.34 Conexión del sistema BDM

El conector BDM que debe tener la tarjeta de desarrollo donde reside el microcontrolador consta de seis terminales, de las cuales sólo cuatro son utilizadas. Estas seis terminales deben corresponder en la tarjeta y en el conector del PYDBDM. La distribución de las terminales se muestra en la Figura 3.35.

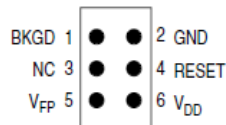


Figura 3.35 Distribución de terminales del BDM

Las terminales BKGD y GND del conector de la tarjeta de desarrollo se utilizan para que la computadora se comuniquen con el módulo BDM. La terminal BKGD permite el envío y la recepción de datos. Para llevar a cabo estas funciones un protocolo de comunicación re-sincroniza el inicio de transferencia de cada bit. Todos los bits comienzan con una señal con flanco de bajada inicializada por la computadora. El microcontrolador espera nueve ciclos de reloj E-clock tras detectar este flanco de bajada y luego muestrea el nivel en la terminal BKGD. Los datos se transfieren iniciando por el bit más significativo a una velocidad de 16 ciclos de E-clock por bit. E-clock se define como SYSCLK/2.

La conexión que se debe implementar en el sistema de desarrollo para que estas terminales cumplan su propósito de envío y recepción de comandos hardware o firmware es el que se muestra en la Figura 3.36.

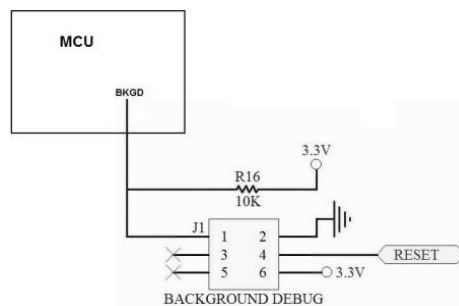


Figura 3.36 Conexión del conector BDM en la tarjeta de desarrollo

3.4 Dispositivos entrada/salida e indicadores

Los dispositivos de entrada y salida que se incorporan a la tarjeta de desarrollo son botones, leds, un indicador auditivo (buzzer), conector de teclado y display externo. La conexión de estos elementos se distribuye en algunos de los puertos de entradas y salidas del microcontrolador de acuerdo lo establecido en la Tabla 3.2.

3.4.1 Botones

El sistema cuenta con cuatro botones de propósito general conectados en la parte baja del puerto K. cada botón se conecta de acuerdo al arreglo mostrado en la Figura 3.37. Cuando el botón no está presionado la terminal correspondiente del puerto recibe un nivel bajo. Al momento de presionar el botón la resistencia se conecta a un nivel alto, mismo que entra a la terminal del puerto. Para poder leer el valor de los botones conectados a este puerto es necesario configurar la parte baja del puerto K como entradas.

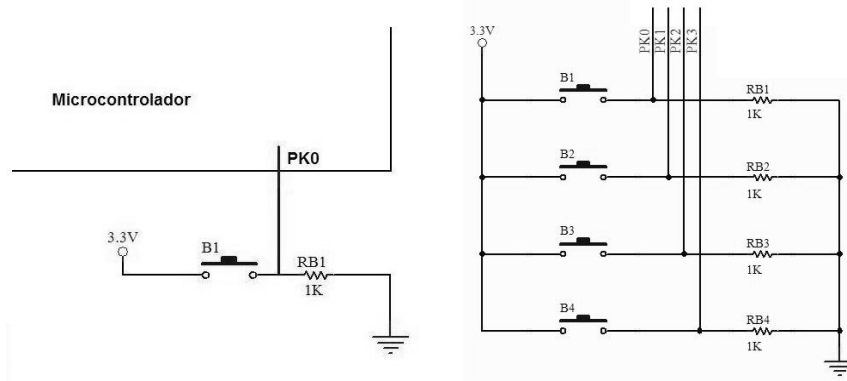


Figura 3.37 Conexión de los botones al microcontrolador

3.4.2 Leds

La parte alta del puerto K se utiliza para conectar cuatro leds de propósito general. Estas terminales se deben configurar como salidas para manipular el estado de los leds. Cuando se envía una señal alta a cualquiera de las cuatro terminales el led correspondiente se enciende, en caso contrario permanecerá apagado. El valor de la resistencia que acompaña a cada led responde a $R_{led} = \frac{V_{puerto} - V_{led}}{I_{led}}$, considerando los valores necesarios para encender un led como $V_{led}=1.2v$ y $I_{led}=10mA$ aproximadamente, el valor de la resistencia es $R_{led} = \frac{3.3v - 1.2v}{10mA} = 210\Omega$, buscando un valor comercial se considera 220Ω . En la Figura 3.38 se muestra la conexión.

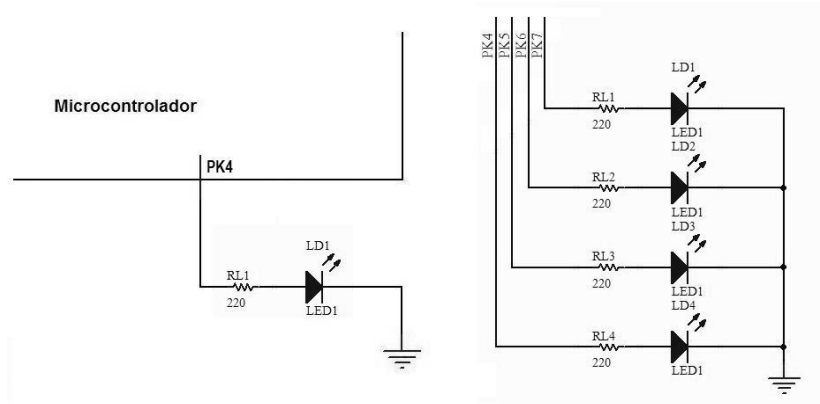


Figura 3.38 Conexión de los leds al microcontrolador

3.4.3 Buzzer

El *buzzer* (o *zumbador*) es un dispositivo audible que genera un sonido en diferentes tonos dependiendo de la frecuencia de la señal con la que se alimenta. La frecuencia de resonancia que manejan estos dispositivos se encuentra alrededor de los 3.5 KHz. Cuenta con dos terminales que sirven para conectar la señal de entrada.

Para manipular un dispositivo audible con el sistema de desarrollo se utiliza la terminal J1 del microcontrolador por donde se le envía una tren de pulsos del cual se puede variar la frecuencia para cambiar el tono. La otra terminal del *buzzer* se conecta a tierra.

3.4.4 Teclado

El sistema de desarrollo cuenta con un conector para teclado matricial de 16 teclas en el puerto G. Se utiliza este puerto ya que se requieren ocho líneas del microcontrolador para poder leer cualquiera de las 16 teclas. Las ocho terminales del teclado se dividen en cuatro renglones y cuatro columnas como se ilustra en la Figura 3.39, de ahí el término “matricial”.

Las columnas del teclado se conectan a la parte baja del puerto G, estas terminales se configuran como entradas, mientras que los renglones se conectan a la parte alta del puerto y dichas terminales se configuran como salidas. Adicionalmente se conecta una resistencia a cada columna del teclado poniendo el otro extremo de cada una a tierra, asegurando así un valor de cero lógico cuando ninguna de las teclas se encuentra presionada. Al momento de presionar una tecla se conecta una salida del puerto G con una entrada del mismo.

Para identificar cuál de las teclas fue presionada se envía un 1 lógico por una de las terminales de salida del puerto, mismo que se tendrá que ver reflejado en una de las cuatro entradas, mientras las demás permanecen en 0. La salida de 1 lógico se debe de enviar a cada uno de los renglones por separado y por cada 1 que se envíe se debe hacer una lectura de las entradas, de manera que se recorran todas las teclas leyendo un renglón a la vez.

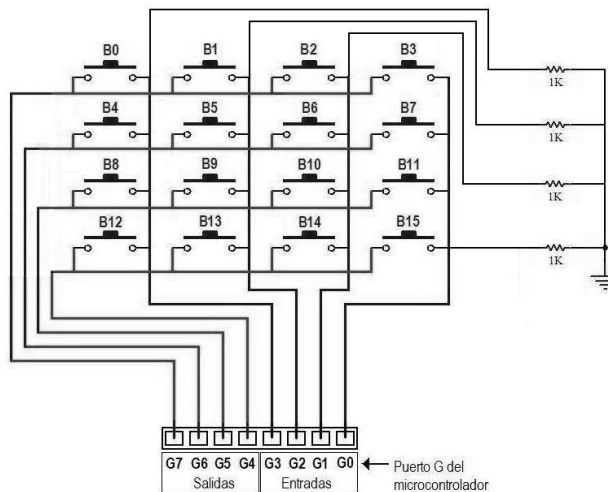


Figura 3.39 Conexión del teclado matricial al microcontrolador

Para facilitar la comprensión del funcionamiento del teclado imaginemos que se encuentra presionada la tecla B5. Primero se envía un 1 a la salida G7 y ceros a las demás salidas. Las teclas B0, B1, B2 y B3 están conectadas a un nivel alto en su parte izquierda, pero como ninguna de ellas está presionada todas las líneas de entrada tienen un valor de 0, por lo tanto la lectura de la parte baja del puerto será 0000. Después de esto se envía un 1 lógico a la salida G6 y ceros a las demás salidas. Ahora las teclas B4, B5, B6 y B7 están conectadas a un nivel alto en su parte izquierda. Como la tecla B5 está presionada, el 1 de la salida G6 se transfiere a la entrada G2 y en las demás entradas se tiene un valor de 0. La entrada que se leería en el caso anterior sería 0100. Si en lugar de tener presionada la tecla B5 fuera la tecla B7, el 1 se encontraría en la

entrada G0 y el valor correspondería a 0001. En la Tabla 3. se muestran las salidas que debe enviar el microcontrolador y las entradas que lee para cada una de de las teclas presionadas.

Tabla 3.9 Identificación de teclas presionadas

SALIDAS	ENTRADAS			
	1000	0100	0010	0001
1000	B0	B1	B2	B3
0100	B4	B5	B6	B7
0010	B8	B9	B10	B11
0001	B12	B13	B14	B15

3.4.5 Display

El sistema de desarrollo cuenta con un conector de 10 terminales para un display alfanumérico de cristal líquido (LCD) de 2 líneas con 16 caracteres cada una (16x2). Este display incluye un controlador Hitachi 44780, que es uno de los controladores más populares para este tipo de displays.

El display tiene 16 terminales:

- | | |
|--------------------|---|
| 1. GND | Tierra |
| 2. V _{DD} | Alimentación de 5 volts |
| 3. V ₀ | Nivel de ajuste de contraste |
| 4. RS | Línea de control de selección de registro |
| 5. R/W | Línea de control de lectura y escritura |
| 6. E | Línea de control para habilitar |
| 7. DB0 | Dato 0 |
| 8. DB1 | Dato 1 |
| 9. DB2 | Dato 2 |
| 10. DB3 | Dato 3 |
| 11. DB4 | Dato 4 |
| 12. DB5 | Dato 5 |
| 13. DB6 | Dato 6 |
| 14. DB7 | Dato 7 |
| 15. LED | Ánodo del led de luz de fondo |
| 16. LED | Cátodo del led de luz de fondo |

En la terminal 3 se conecta un potenciómetro que regula el nivel de contraste para la pantalla. Otra opción sería conectar la terminal 3 a un valor de voltaje fijo, como consecuencia se el contraste no es ajustable. En este proyecto se eligió la primera opción.

Las terminales 4, 5 y 6 son líneas de control que permiten manipular la operación del display. El display recibe órdenes denominas comandos desde el dispositivo que lo controla (en este caso un microcontrolador) indicándole que ejecute acciones como limpiar la pantalla o posicionar el cursor en un lugar determinado. Para diferenciar entre la recepción de un dato y la recepción de un comando, el display cuenta con un registro independiente para cada uno. Con la línea de control de selección de registro (RS) el microcontrolador le indica al display qué tipo de información está enviando. Si se envía un valor de 0 por la línea RS, indica que el microcontrolador enviará datos. Con un 1 en la línea RS se informa que a continuación se enviará un comando.

La señal R/W indica si la operación que está realizando el microcontrolador es de lectura o de escritura. Cuando el microcontrolador desea escribir un valor en el registro de datos o de comandos del display, debe poner la línea R/W a 0. Si por el contrario lo que se requiere es leer datos desde alguno de los registros del display, la señal R/W debe ser igual a 1 lógico.

La señal de *enable* (habilitar) es la que permite indicar el momento en que se realiza una operación de lectura o escritura. Cuando se manda un pulso alto desde el microcontrolador hacia el display a través de la línea E, se está indicando que se ejecute alguna operación.

Las siguientes ocho líneas corresponden a los datos. A través de estas líneas se puede enviar un carácter o un comando de ocho bits, pero también existe la posibilidad de enviar dichos datos en dos partes utilizando solamente las líneas de datos D4–D7; se realiza primero el envío de la parte baja del dato y después los cuatro bits de la parte alta. Esta última configuración es la que hemos seleccionado ya que nos permite ahorrar líneas de comunicación entre el microcontrolador y el display y por lo tanto usar una menor cantidad de líneas de entrada/salida del microcontrolador.

Las dos últimas terminales del display corresponden a la polarización de un led que ilumina la pantalla, pero no todos los displays cuentan con esta opción, en este caso dicha opción no es implementada.

Para el manejo del display se selecciona el puerto H del microcontrolador, el cual cuenta con 7 terminales que se distribuyen de la manera que se muestra en la Tabla 3.9:

Tabla 3.9 Conexión del puerto H a las señales del display

Puerto H	Terminal del LCD
H0	D4
H1	D5
H2	D6
H3	D7
H4	E
H5	RW
H6	RS

En la Figura 3.40 se muestra un diagrama de bloques de la conexión del display.

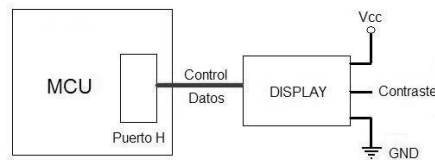


Figura 3.40 Conexión del display

3.5 Dispositivos de monitoreo y control

A continuación se describen los dispositivos incorporados al sistema de desarrollo que sirven para aplicaciones de monitoreo y control de procesos.

3.5.1 Sensores

La tarjeta de desarrollo cuenta con un conector para conectar sensores en el puerto AD, donde se encuentra el convertidor Analógico/Digital. Este puerto es de ocho terminales, de las cuales se emplean tres para conectar sensores de propósito general y se dejan las demás libres para que puedan ser conectados otros sensores según las necesidades de las aplicaciones de usuario.

Los sensores no se incluyen dentro del circuito de la tarjeta de desarrollo, con la idea de poder colocarlos en el lugar donde se requiera sensor variables. En vez de esto se incorporan en un módulo separado que se conecta a la tarjeta a través de cables.

El primer valor que se mide es el voltaje en un potenciómetro que se conecta en la terminal 1 del puerto AD. Se conecta un extremo del potenciómetro a la alimentación de 3.3 v, el otro extremo a la tierra y la terminal variable hacia la terminal de entrada del microcontrolador. De este modo el rango de operación del potenciómetro será de 0 a 3.3 volts, lo cual corresponde con el rango de operación del convertidor analógico digital, de manera que se aprovecha la escala completa de conversión.

El segundo sensor que se incluye, en la terminal 1 del puerto AD, es una fotorresistencia, o resistencia sensible a la luz. Con este elemento es posible sensor el cambio de intensidad luminosa en el lugar donde se coloca el sensor. Estos dispositivos se conocen comúnmente con el nombre de LDR (resistor dependiente de

luz) y tienen la propiedad física de variar su valor de resistencia eléctrica de acuerdo a las condiciones de iluminación que perciben. Cuando la intensidad luminosa es poca la resistencia del LDR es grande (en el orden de los Mega Ohms) y cuando se someten a altas intensidades luminosas su resistencia baja hasta unos cuantos Ohms, típicamente entre 50 Ohms y 1 kΩ, dependiendo el dispositivo en particular.

La conexión de la fotorresistencia se realiza formando un divisor de voltaje y conectando el punto de voltaje variable entre ambas resistencias a una de las entradas del convertidor analógico digital. Para calcular el valor de la resistencia fija del arreglo se considera un valor mínimo de resistencia del LDR de 500 Ω y un valor máximo de 2 MΩ. En el divisor de voltaje se distribuye el voltaje de la fuente entre las resistencias que forman el arreglo de manera proporcional a su valor. El cálculo de la resistencia fija se realiza considerando que el voltaje introducido al convertidor analógico digital cubra la mayor parte de su rango de operación, tratando de acercar la lectura a cero cuando el sensor detecte obscuridad y procurando que la lectura se encuentre cercana a los 3.3 volts en presencia de una alta intensidad de luz.

En la Figura 3.41 se muestra el arreglo del divisor de voltaje. Cuando la intensidad luminosa detectada por el sensor es baja, la resistencia del LDR es grande y el voltaje de la fuente se utilizar en la fotorresistencia, leyendo un valor cercano a cero en R₁. En caso de aumentar la intensidad luminosa, disminuye la resistencia del LDR, provocando que la mayor cantidad de voltaje se utilice en R₁ y la lectura en el convertidor analógico digital se encuentre cercana a los 3.3 V.

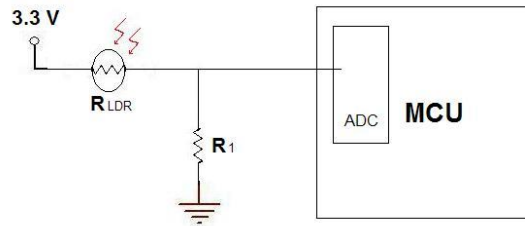


Figura 3.41 Conexión de la fotorresistencia

Las expresiones para calcular los voltajes de las resistencias en el divisor de voltajes son las siguientes:

$$V_{LDR} = \frac{R_{LDR}}{R_{LDR} + R_1} V_{in}$$

$$V_1 = \frac{R_1}{R_{LDR} + R_1} V_{in}$$

Para calcular un valor apropiado de R₁ consideremos la ecuación del voltaje V₁ y los casos de luminosidad máxima y mínima. En ausencia de luz la lectura deseable es cero, como este valor no puede ser posible consideremos un valor de 1% del rango completo, es decir $V_1 = 0.01 V_{in}$.

$$0.01 V_{in} = \frac{R_1}{R_{LDR} + R_1} V_{in}$$

$$\frac{R_1}{R_{LDR} + R_1} = 0.01$$

$$R_1 = 0.01 R_{LDR} + 0.01 R_1$$

$$0.99 R_1 = 0.01 R_{LDR}$$

$$R_1 = \frac{0.01}{0.99} R_{LDR} = \frac{0.01(2 \times 10^6)}{0.99} = 20.202 \text{ k}\Omega$$

Hacemos lo mismo para buscar un valor de R₁ que haga que el valor de voltaje leído sea cercano a 3.3v cuando el nivel de iluminación sea máximo, es decir, la resistencia del LDR sea igual a su valor mínimo. Consideramos como un valor cercano a 3.3 V el 99% del rango completo.

$$0.99V_{in} = \frac{R_1}{R_{LDR} + R_1} V_{in}$$

$$\frac{R_1}{R_{LDR} + R_1} = 0.99$$

$$R_1 = 0.99R_{LDR} + 0.99R_1$$

$$0.01R_1 = 0.99R_{LDR}$$

$$R_1 = \frac{0.99}{0.01} R_{LDR} = \frac{0.99(500)}{0.01} = 49.5 \text{ K}\Omega$$

El valor seleccionado para R_1 es el promedio de los dos valores calculados, es decir $34.8 \text{ K}\Omega$, pero tomando el valor comercial más cercano se selecciona $33 \text{ K}\Omega$. Con este valor el voltaje más bajo y el más alto, registrados por el convertidor analógico digital se muestran en la Tabla 3..

Tabla 3.11 Rango de operación del voltaje de la fotorresistencia

	RLDR	$V_1 = \frac{R_1}{R_{LDR} + R_1} V_{in}$	% de la escala completa	Rango de operación (volts)
Mínima luminosidad	2 MΩ	0.05 V	1.6%	0.05–3.25
Máxima luminosidad	500 Ω	3.25 V	98.5%	

Por último el tercer sensor que se incluye en la tarjeta de desarrollo, conectado en la terminal 2 del puerto AD, es un sensor de temperatura, el LM35. La salida de este integrado es proporcional a la variación de temperatura en escala Celsius y se comporta de manera lineal, proporcionando 10mV a la salida por cada grado de variación en la temperatura. Tiene dos configuraciones dependiendo el rango de temperatura que se desee sensar. La configuración básica abarca un rango de $+2^\circ\text{C}$ a $+150^\circ\text{C}$. La segunda configuración abarca la escala completa que maneja el sensor de -155°C a $+150^\circ\text{C}$. En la Figura 3.42 se muestran ambas configuraciones. La configuración seleccionada en este sistema de desarrollo es la escala reducida, ya que no se considera necesario realizar el sensado de temperaturas muy elevadas ni de temperaturas demasiado bajas.

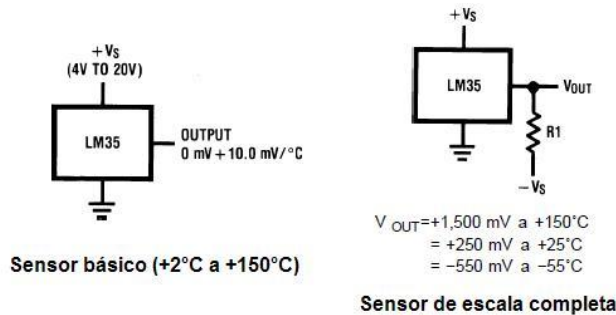


Figura 3.42 Configuraciones del sensor de temperatura LM35

El LM35 se fabrica en varios tipos de encapsulados pero en todos los casos cuenta con tres terminales (en el encapsulado tipo SO-8 sólo se utilizan tres de las ocho terminales con las que cuenta). Dichas terminales corresponden a la alimentación de voltaje, la tierra y la señal de salida. Su voltaje de alimentación puede ser desde los 4 hasta los 30 volts.

A este circuito se le agrega una etapa de acoplamiento antes de leer la señal de salida desde el microcontrolador. Dicha etapa consta de un amplificador debido a que la salida del sensor se mide en *milivolts*. El amplificador seleccionado es un LM124 que cuenta con la posibilidad de ser alimentado con fuentes de voltaje simples desde los 3 volts y fuentes de alimentación dual desde los ± 1.5 volts, razón por la

cual fue seleccionado. El voltaje de alimentación del amplificador para este caso es de 5 volts, con el que podemos alimentar al sensor y al amplificador con la misma fuente. El CI tiene cuatro amplificadores en su interior. En la Figura 3.43 podemos ver un diagrama del mismo.

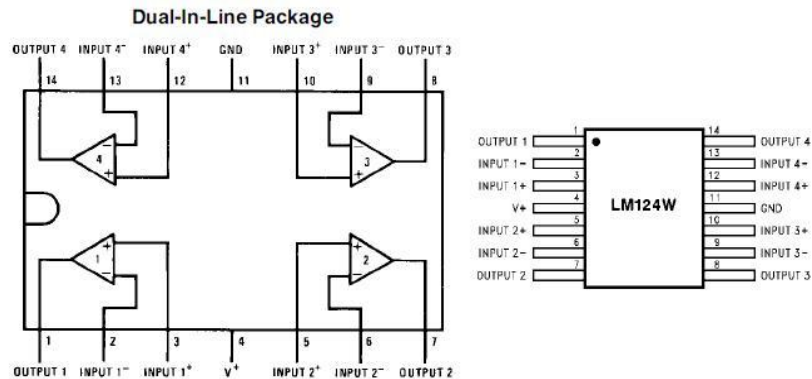


Figura 3.43 Diagrama del amplificador LM124

La configuración con la que se conecta el amplificador es una configuración no inversora y el valor de la ganancia se selecciona escogiendo un rango de operación del sensor, es decir, como no se pretende usar el sensor para leer temperaturas más altas de los 50°C, se escoge un rango de los 2°C a los 50°C, con lo cual a una temperatura de 2°C le corresponde una salida de 0 volts y a una temperatura de 50°C le corresponde un voltaje de 3.3 en la lectura del convertidor analógico digital. En la Tabla 3.10 se muestra la relación que permite obtener el valor de la ganancia del amplificador.

Tabla 3.10 Ganancia del amplificador

Temperatura	Salida del sensor (V)	Salida del amplificador (V)
2°C	0	0
3°C	0.010	0.01*G
x°C	(x-2)*0.010	(x-2)*0.010*G
50°C	0.48	3.3 $G = \frac{3.3}{0.48} = 6.8$

En la Figura 3.44 se muestra la conexión del sensor de temperatura, el amplificador y el microcontrolador. Los valores de las resistencias R₁ y R₂ se seleccionan de manera que el amplificador tenga una ganancia de 6.8. La ecuación que modela el amplificador se obtiene calculando la corriente en ambas resistencias e igualándolas, considerando el amplificador ideal, es decir, que la diferencia de potencial entre su entrada inversora y no inversora es cero y que la impedancia de entrada en ambas entradas es infinita, por lo tanto ninguna corriente fluye hacia el interior del amplificador por ninguna de sus dos entradas. El voltaje de la entrada inversora y la no inversora corresponde a V_{in}, siendo éste el voltaje de salida del sensor.

$$\frac{V_{in}}{R_1} = \frac{V_{out} - V_{in}}{R_2}$$

$$\frac{V_{out}}{R_2} = \frac{V_{in}}{R_1} + \frac{V_{in}}{R_2}$$

$$V_{out} = \left(\frac{R_2}{R_1} + 1\right)V_{in}$$

De la ecuación anterior obtenemos la expresión para la ganancia del amplificador (R₂/R₁+1). Si R₁=1.2 KΩ:

$$\frac{R_2}{R_1} + 1 = 6.8$$

$$R_2 = 5.8R_1 = 5.8(1.2K\Omega) = 6.96K\Omega$$

Para controlar el relevador se hace fluir corriente a través de su bobina por medio de una salida del microcontrolador. Para esto se configura una terminal del microcontrolador como salida digital, pero dicha salida no es capaz de manejar la corriente nominal de la bobina del relevador, la cual corresponde a 72 mA aproximadamente. Para solucionar este problema se emplea un transistor de unión bipolar en su configuración de emisor común, con la idea de transmitir una corriente a su base a través del microcontrolador y verla aumentada a través de su unión colector-emisor. De esta manera podemos conectar el relevador como carga del transistor y hacer circular la corriente necesaria para hacerlo conmutar, se coloca adicionalmente un diodo “volante” que protege al transistor de sobrevoltajes. La configuración mencionada se muestra en la Figura 3.46.

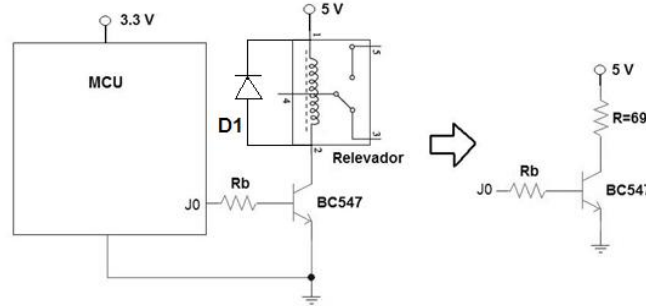


Figura 3.46 Conexión de relevadores

El transistor que se utiliza es el BC547 que es un TBJ tipo NPN. Este transistor es adecuado ya que maneja una corriente de colector máxima de 100 mA. Los voltajes máximos en sus uniones colector-emisor, colector-base y emisor-base son 45, 50 y 6 V respectivamente. La ganancia de corriente en DC varía según la clasificación del transistor (A, B o C) y la corriente de operación del colector. Para el caso en el que el transistor opera cerca de su punto de saturación la ganancia para los dispositivos con terminación A es de 120, para los que terminan en B es de 180 y aquellos que tienen una terminación C les corresponden una ganancia de 300. Para el diseño se usa un transistor clasificación B y se asume una ganancia de corriente de 180.

La resistencia en la base del transistor limita la corriente de entrada y absorbe el voltaje de entrada. Para calcular su valor se considera la malla formada por la salida del microcontrolador, la resistencia de la base, la unión base-emisor y tierra. Considerando que la corriente de emisor es prácticamente igual a la corriente de colector y que la corriente de colector es igual a la corriente de la base multiplicada por la ganancia se obtiene lo siguiente:

$$V_{in} = \frac{i_c}{\beta} R_b + V_{be}$$

Considerando que el voltaje de la unión base-emisor corresponde al voltaje de un diodo polarizado en directa, es decir 0.7 V, y que la corriente de colector corresponde a la corriente de activación del relevador (72 mA), despejamos de la ecuación el valor de la resistencia de base:

$$R_b = \frac{\beta(V_{in} - V_{be})}{i_c} = \frac{(180)(3.3 - 0.7)}{0.072} = 6.5 K\Omega$$

El voltaje colector-emisor se calcula con la malla que forman la fuente de 5 volts, la resistencia del relevador, la unión colector-emisor y la tierra, de la siguiente manera:

$$V_{IN} = R_c i_c + V_{ce}$$

$$V_{ce} = V_{IN} - R_c i_c = 5 - (69)(0.072) = 32mV$$

De lo anterior determinamos que el transistor opera en modo de saturación. En la Figura 3.47 se muestra la recta de carga del transistor.

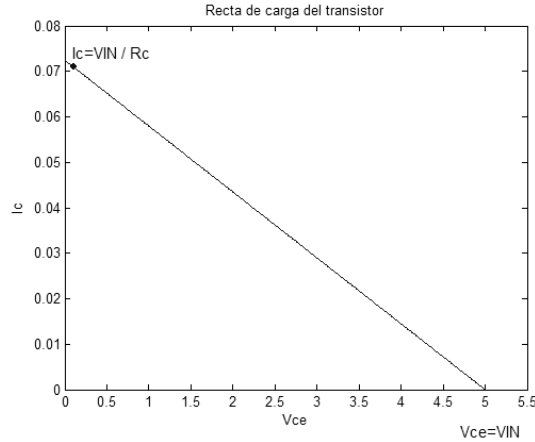


Figura 3.47 Recta de carga del transistor

3.5.3 Entradas optoacopladas

Las entradas optoacopladas se incluyen en el sistema de desarrollo debido a que resultan muy prácticas cuando se requiere que el microcontrolador procese entradas provenientes de dispositivos que operan a niveles más altos de voltaje y corriente. Los dispositivos optoacopladores permiten aislar eléctricamente dos circuitos, cambiar niveles de voltaje y evitan que el ruido de uno de los circuitos se transfiera al otro.

El optoacoplador cuenta con un LED infrarrojo (o IRED) a su entrada y un fotodetector a su salida que puede ser de diferentes tipos, para el caso de esta aplicación es un fototransistor NPN. Los dos elementos se encuentran acoplados ópticamente en el interior del circuito integrado. Cuando fluye una corriente a través del LED de la entrada éste se enciende y emite radiación infrarroja, la cual activa al fotodetector y provoca que conmute. Como los dos componentes del optoacoplador no se encuentran en contacto eléctrico, es posible que cada uno trabaje en un rango distinto de voltaje y corriente, incluso que manejen referencias de voltaje diferentes (cada circuito puede tener su propia tierra).

El dispositivo que se utiliza para conectar entradas optoacopladas al sistema es el circuito integrado 4N25. En la Figura 3.48 se muestra un diagrama esquemático de este optoacoplador. Las terminales 1 y 2 son las entradas de alimentación del LED. La base del transistor se encuentra disponible en la terminal 6, pero como no se utiliza se puede dejar sin conectar. Las terminales 4 y 5 corresponden al emisor y el colector del transistor respectivamente. En Tabla 3.12 se presentan las características eléctricas del 4N25.

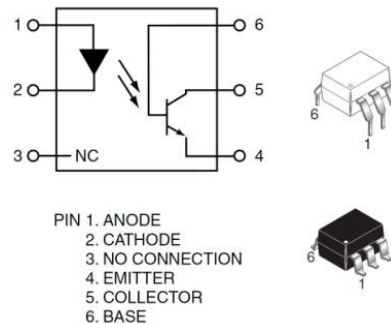


Figura 3.48 Diagrama esquemático del optoacoplador y ejemplos de encapsulados

Tabla 3.12 Parámetros eléctricos del optoacoplador

PARÁMETROS ACOPLADOS			EMISOR		RECEPTOR				
CTR (%)	$V_{CE(SAT)}$ (V) $I_F=50mA, I_C=2mA$		I_F máx (mA)	V_F máx (V) $I_F=10mA$		I_C máx (mA)	$V_{(BR)CB}$ (V)	$V_{(BR)CE}$ (V)	$V_{(BR)EC}$ (V)
	Típico	Máximo		Tip.	Máx.				
20	0.1	0.5	80	1.2	1.5	100	70	30	7

En la Figura 3.49 se muestra el circuito para conectar el optoacoplador al microcontrolador. A la entrada del optoacoplador se conecta una resistencia que limita la corriente del LED y disminuye el voltaje de la entrada para proporcionar el voltaje necesario para activar el LED. El diodo D₁ se agrega para permitir que señales de corriente alterna puedan ser conectadas al sistema y para proteger al LED en caso de polarizaciones inversas. A la salida del optoacoplador se coloca una resistencia de *pull down* que alimenta la entrada del microcontrolador con un nivel bajo mientras el transistor se encuentre en estado “apagado”. Al recibir una entrada el transistor pasa a su estado “encendido” y fluye una corriente del colector al emisor, provocando que la resistencia R₂ quede conectada a la fuente de voltaje y alimente la entrada del microcontrolador con un nivel alto. El diseño de este sistema se realiza considerando una entrada de voltaje máximo de 24 V_{DC}.

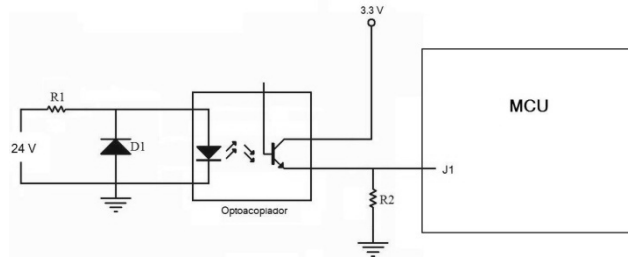


Figura 3.49 Circuito del optoacoplador

El valor de la resistencia R₂ debe ser tal que el transistor opere en estado de saturación para que el voltaje colector-emisor sea mínimo y se aplique la mayor cantidad de voltaje posible a la resistencia R₂. De los parámetros del optoacoplador mostrados en la Tabla 3.12 podemos observar que el voltaje colector-emisor de saturación corresponde a 0.1 V cuando la corriente de colector es 2 mA. Con lo anterior tenemos que cuando el transistor está conduciendo:

$$V_{R2} = V_{cc} - V_{CE(SAT)} = R_2 I_c$$

$$R_2 = \frac{V_{cc} - V_{CE(SAT)}}{I_c} = \frac{3.3 - 0.1}{2 \times 10^{-3}} = 1.6 \text{ K}\Omega$$

Para obtener el valor de la resistencia R₁ se ocupan los parámetros del emisor del optoacoplador. La corriente de activación del LED se obtiene a partir de la Relación de Transferencia de Corriente (CTR), correspondiente al 20%, lo cual indica que la corriente de salida del optoacoplador es un 20% de su corriente de entrada, por lo tanto la corriente de entrada corresponde a 10 mA. En la Figura 3.50 se muestra una gráfica del voltaje colector-emisor de saturación del transistor respecto a la corriente de activación del LED para diferentes valores de corriente de colector. Observando esta gráfica se confirma que el voltaje colector-emisor de saturación tiene un valor cercano a cero para una corriente de colector de 2 mA y una corriente de encendido del LED de 10 mA. En la Tabla 3.12 se indica que el voltaje de activación del LED tiene un valor típico de 1.2 volts, con lo cual obtenemos lo siguiente:

$$V_{R1} = V_{in} - V_F = R_1 I_F$$

$$R_1 = \frac{V_{in} - V_F}{I_F} = \frac{24 - 1.2}{10 \times 10^{-3}} = 2.28 \text{ K}\Omega$$

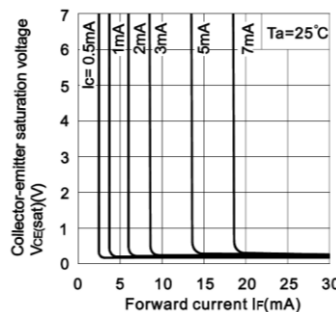


Figura 3.50 Corriente de activación del LED Vs Voltaje Colector-Emisor de saturación

3.6 Diseño esquemático de la tarjeta de desarrollo

Luego de la descripción de cada uno de los elementos que componen el sistema de desarrollo en la que se programará la pila de protocolos TCP/IP, a continuación se muestran los circuitos esquemáticos de cada uno de ellos, mismos que concuerdan con los cálculos realizados en las descripciones anteriores y que son la base para la generación del circuito impreso bajo el cual se maquilará la tarjeta de desarrollo. Estos diagramas se crean con ayuda Protel 2004, un software de diseño de circuitos electrónicos, donde se pueden crear diagramas esquemáticos y de circuitos impresos.

3.6.1 Circuito mínimo del microcontrolador

El circuito mínimo para el funcionamiento del microcontrolador incluye: fuente de alimentación, circuito de reset, cristal, capacitores de filtrado para terminales de alimentación y resistencias de configuración. Adicionalmente se colocan: leds de estatus para las funciones de comunicación por red y alimentación del sistema, así como las conexiones para el conector RJ45. El diagrama esquemático de dicho circuito se muestra en la Figura 3.51. (En el apéndice D se muestra el diagrama a mayor escala).

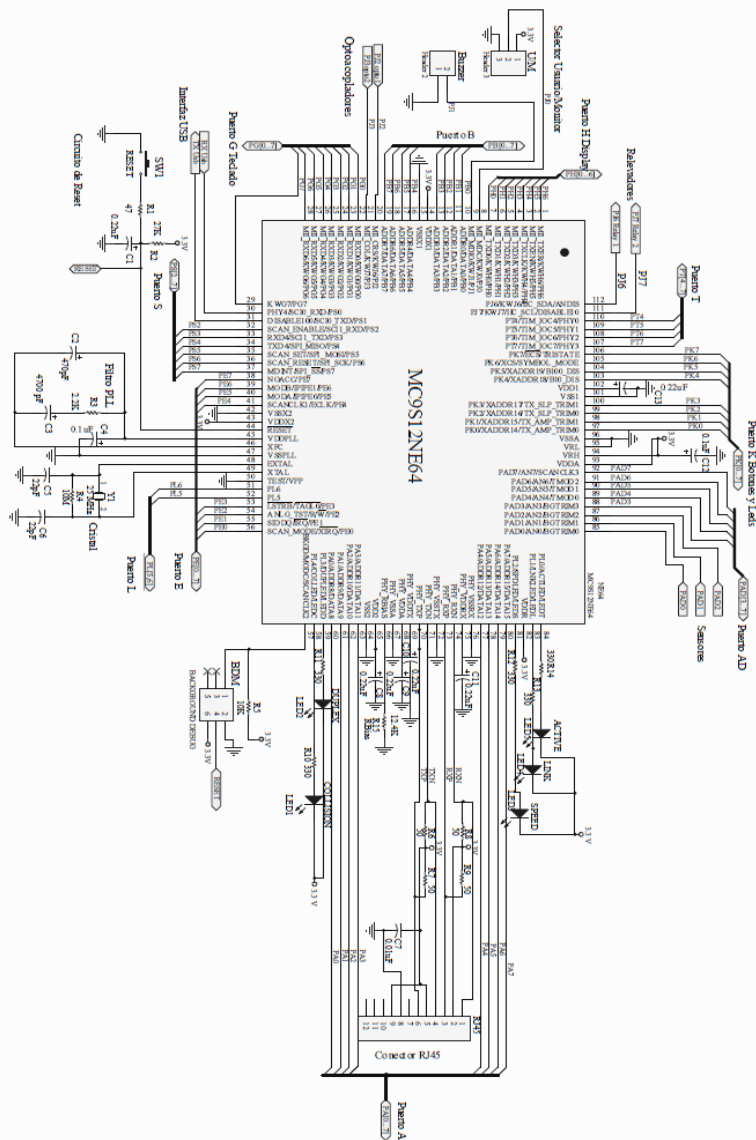


Figura 3.51 Diagrama esquemático circuito mínimo del microcontrolador

3.6.2 Circuito de alimentación

El circuito de alimentación consta de dos circuitos: uno de 3.3 volts para el microcontrolador y uno de 5 volts que alimenta al convertidor USB y otros dispositivos como se muestra en la Figura 3.52.

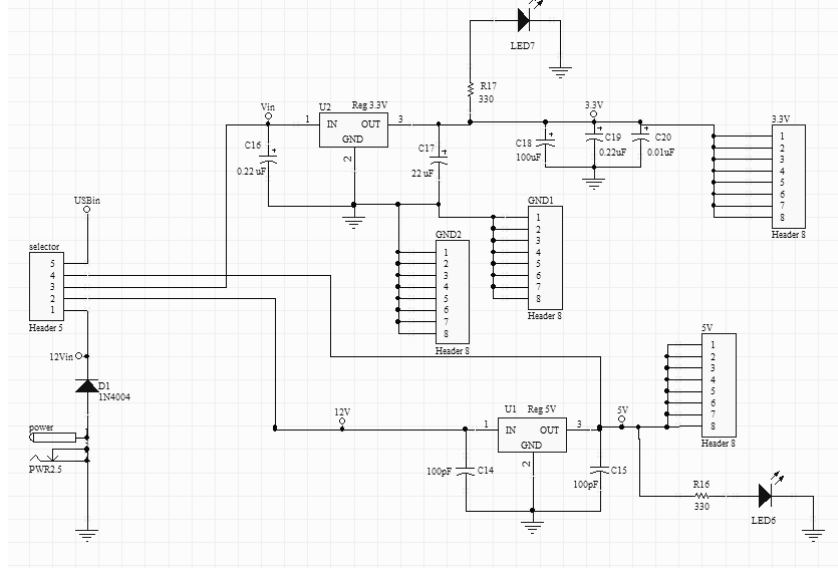


Figura 3.52 Diagrama esquemático circuito de alimentación

3.6.3 Interfaz de comunicación USB

La interfaz USB se compone del C.I. FT232BM y sus componentes de configuración según las especificaciones descritas anteriormente y mostradas en la Figura 3.53

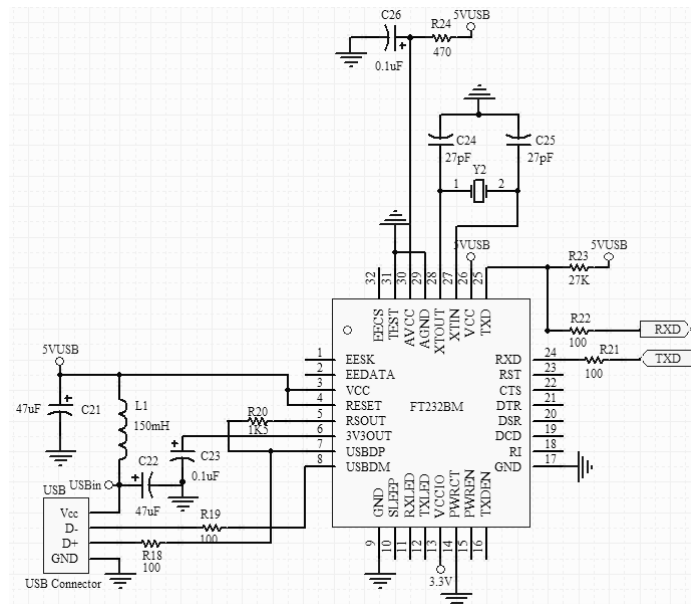


Figura 3.53 Diagrama esquemático Interfaz de comunicación USB

3.6.4 Dispositivos de entrada/ salida

El diagrama de los dispositivos de entrada y salida está compuesto por: los botones, leds, buzzer, conector de teclado y de display, como se muestra en la Figura 3.54. La tarjeta externa para demostrar el correcto funcionamiento del teclado y display está descrita en el Apéndice D.

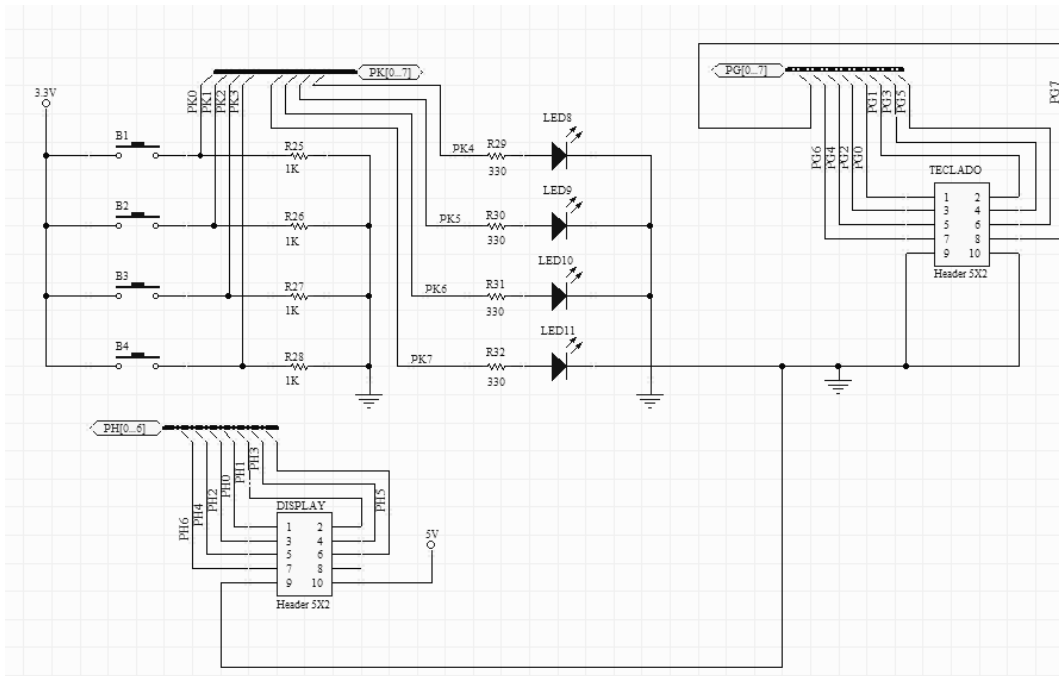


Figura 3.54 Diagrama esquemático dispositivo de entrada y salida

3.6.5 Dispositivos de monitoreo y control

Los dispositivos considerados de monitoreo y control en la tarjeta de desarrollo son: 2 relevadores, 2 entradas optoacopladas y una conexión dedicada para los sensores: Potenciómetro, LDR y Sensor de temperatura LM35, como se ilustra en la Figura 3.55. En busca de versatilidad en la tarjeta de desarrollo los sensores se colocan en una tarjeta externa misma que se describe en el Apéndice D, complemento de capítulo 3.

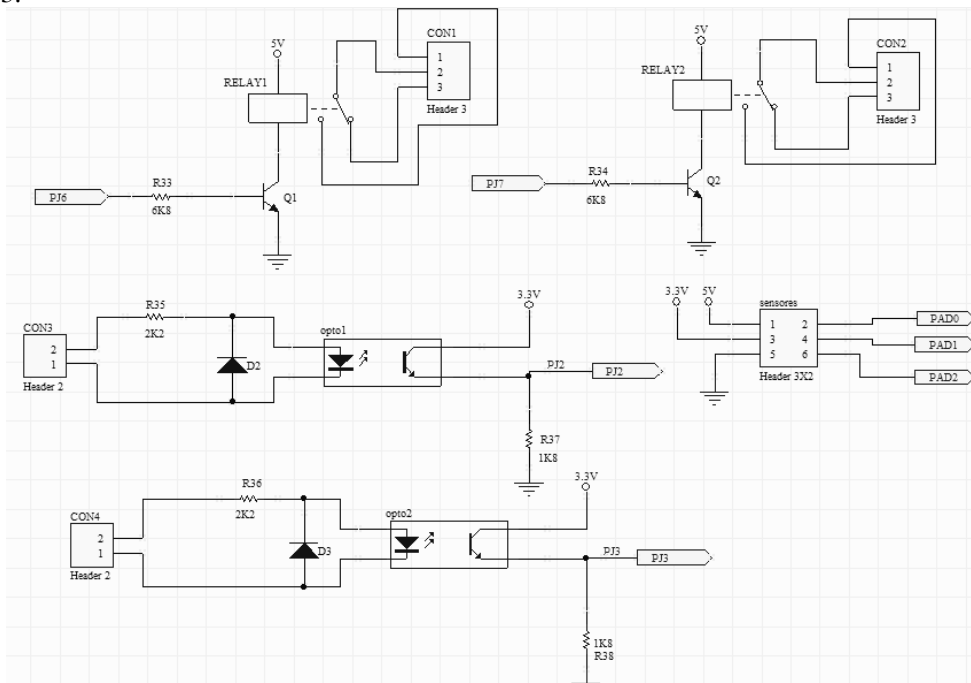


Figura 3.55 Diagrama esquemático dispositivos de monitoreo y control

3.7 Circuito impreso de la tarjeta de desarrollo

Basado en los diseños esquemáticos anteriores, se diseña el circuito impreso de la tarjeta de desarrollo mostrado en la Figura 3.56, en el cual se muestra un circuito de una sola capa, con barrenos no metalizados conformando un diseño sencillo y de bajo costo.

Los elementos eléctricos utilizados son del tipo *through hole*, debido a que estos son más comerciales, de bajo costo y fácil soldado, que los de tecnología de montaje superficial. La distribución espacial de los componentes en la tarjeta es acorde al acomodo de los puertos del microcontrolador, respetando las distancias recomendadas en la hoja de datos técnicos del MC9S12NE64 para algunos dispositivos como el conector RJ45 y el oscilador de 25 Mhz.

3.7.1 Consumo de la tarjeta de desarrollo

A continuación se muestra la suma del consumo de corriente eléctrica de la tarjeta de desarrollo considerando los valores máximos de corriente de cada uno de los elementos mostrados en la Tabla 3.13. Este análisis se realiza considerando el caso en que todos los elementos se encuentran activados, sin considerar elementos externos.

Tabla 3.13 Suma del consumo de corriente

Componente	Máxima Corriente consumida	Componente	Máxima Corriente consumida
MC9S12NE64	285 mA	Buzzer	5 mA
FT232BM	25 mA	Relevador (2)	72 mA c/u -> 144 mA
Regulador 5 V (7805)	10 mA	Optoacoplador	10 mA-> 20 mA
Regulador 3.3 V (LF33)	10 mA		
Leds (11)	10 mA c/u->110mA	TOTAL Imax	609 mA

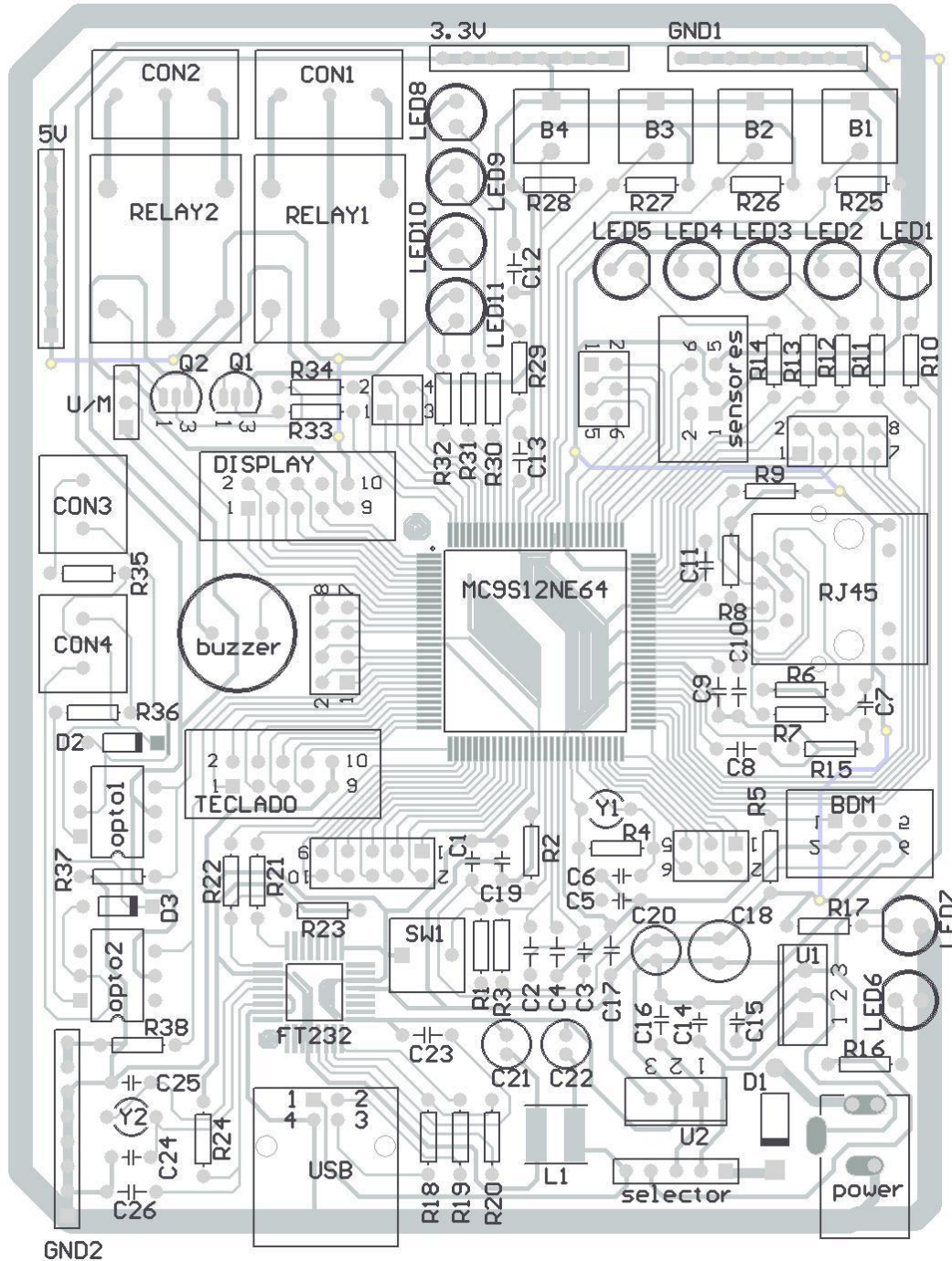


Figura 3.56 Circuito Impreso Tarjeta de desarrollo, vista por debajo

En el apéndice D, junto a los diseños de las tarjetas externas para los sensores, display y teclado se encuentra el listado de componentes que conforman la tarjeta de desarrollo, así como su respectivo diseño esquemático y de circuito impreso.

3.5.1 Implementación de la tarjeta de desarrollo

Con base en el circuito impreso diseñado, a partir del análisis de los componentes de la tarjeta de desarrollo que se llevó a cabo a lo largo del capítulo, finalmente contamos con el hardware o parte física de nuestro

sistema de desarrollo, en el cual se implementará la pila de protocolos TCP/IP y se muestra en la Figura 3.57.

En los siguientes capítulos se describirán las bases e implementación de la pila de protocolos TCP/IP y en el capítulo 6 se presentan las pruebas de funcionalidad de la tarjeta de desarrollo construida durante este capítulo.

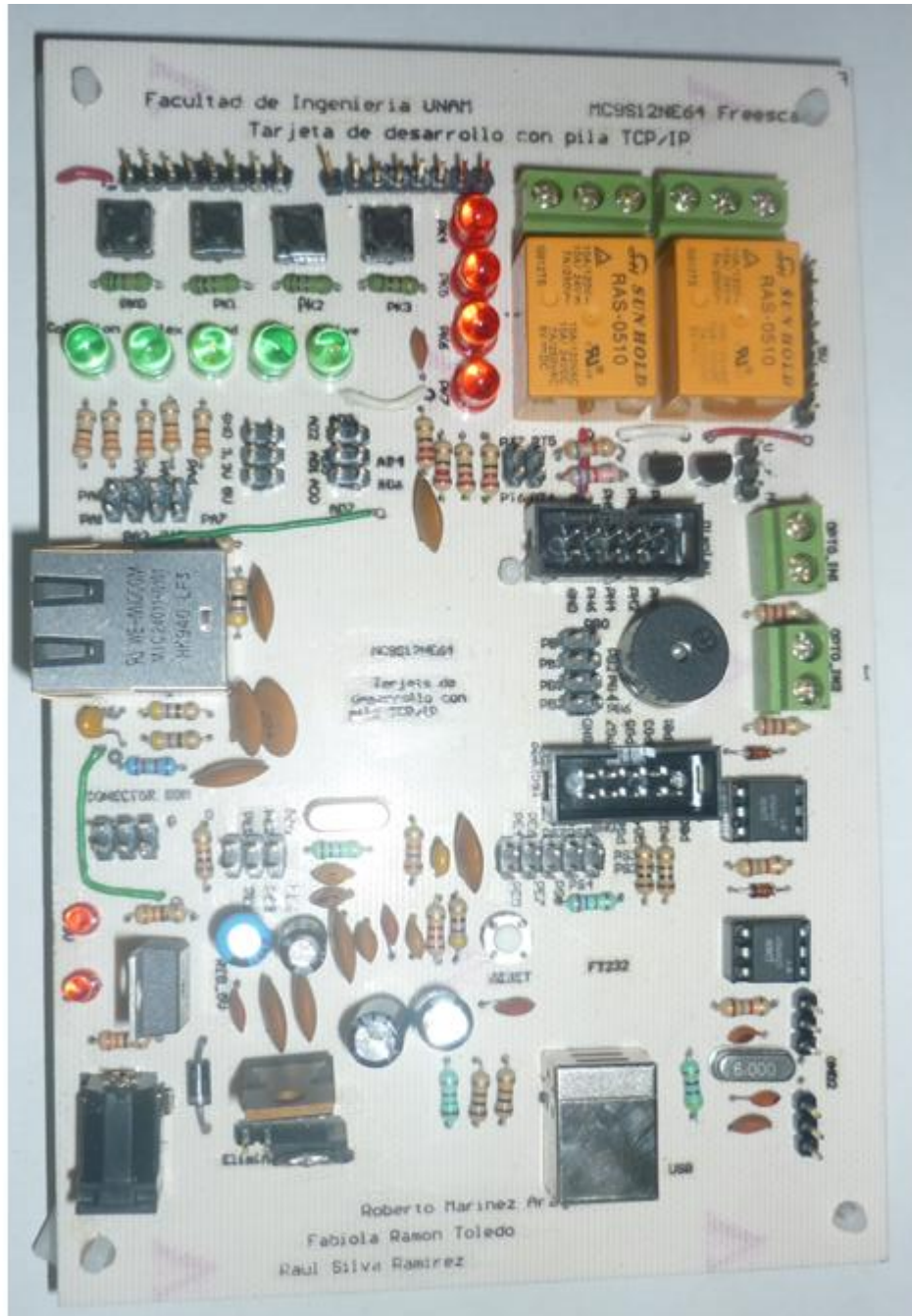


Figura 3.57 Tarjeta de desarrollo con pila TCP/IP