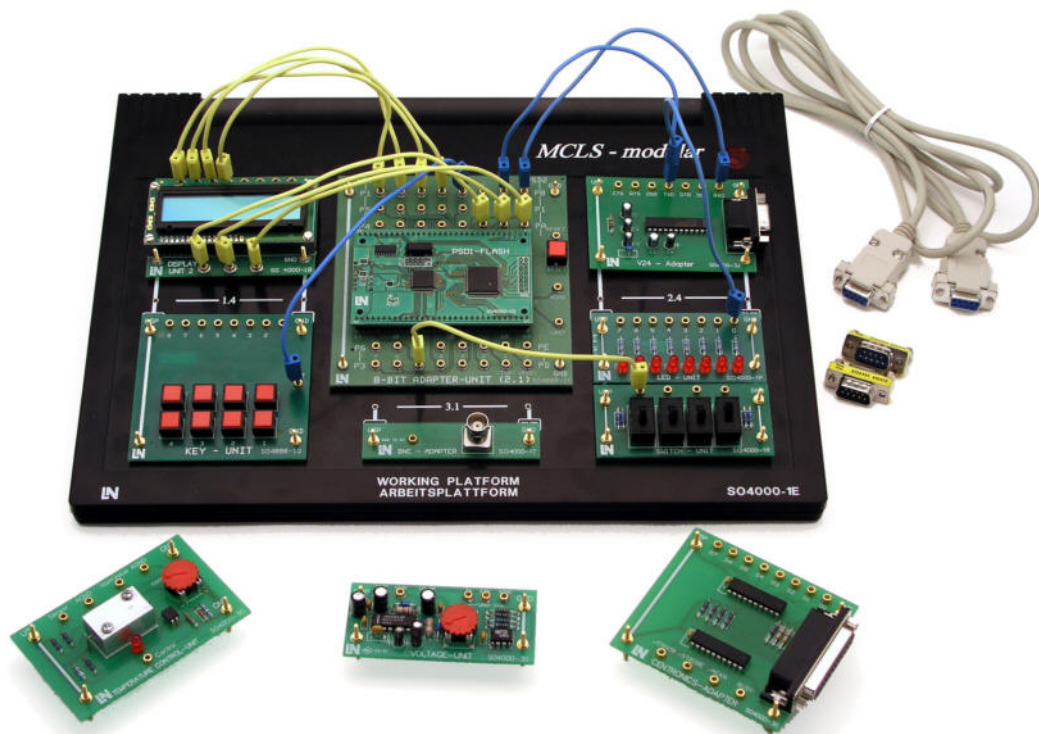




Programación de interfaces de transmisión de datos (C515C)

CMC 4



Guía de ejercicios para el estudiante

SH5004-1D

3ra. edición

Autor: equipo de autores APMC Mittweida

Índice

Objetivo de los ensayos	1
Sinóptico de ejercicios.....	1
Instalación experimental.....	1
Introducción.....	3
Transmisión serial de datos	4
Comunicación paralela.....	6
CMC 4-1 Comunicación de datos mediante la interfase serial del microcontrolador...8	
La interfase serial „on chip“	8
Ensayo CMC 4-1A, Transferencia de datos seriales hacia el PC	11
Ensayo CMC 4-1B, Como consecuencia de un comando envía una respuesta (texto) al PC	18
Ensayo CMC 4-1C, Cíclicamente se transmite un valor de medición del convertidor A/D „on chip“ del microcontrolador al PC	25
Ensayo CMC 4-1D	35
CMC 4-2 Comunicación serial entre 2 terminales de datos por medio de una interfase V.24	43
Utilización del adaptador de V24.....	43
El UART de software.....	44
Ensayo CMC 4-2A.....	47
Ensayo CMC 4-2B.....	56
CMC 4-3 Salida de datos a una impresora por medio de una interfase CENTRONICS67	
Utilización de un adaptador CENTRONICS	67
Utilización de impresoras existentes	67
Ensayo CMC 4-3A.....	70
Ensayo CMC 4-3B.....	78

Objetivo de los ensayos

1. Los aprendices / estudiantes adquieren conocimientos para la utilización de componentes seleccionados de la comunicación de datos para el intercambio de datos entre un microcontrolador y un aparato externo.
2. Los aprendices / estudiantes se familiarizan con la estructura del MC y de su perifería. Aprenden a emplear de manera óptima el conjunto de comandos del MC.
3. Los aprendices / estudiantes se familiarizan con la programación estructurada desde el planteamiento hasta la solución.

Sinóptico de ejercicios

CMC 4-1: Comunicación de datos por medio de la interfase serial del microcontrolador

- Funciones básicas de una interfase serial asíncrona (parámetros, protocolos)
- La interfase serial asíncrona del microcontrolador
- Ejemplos de aplicación para la interfase serial asíncrona (on chip)
 - Enviar un texto al PC
 - Enviar valores medidos al PC

CMC 4-2: Comunicación serial por medio de una interfase V.24 entre 2 terminales de datos (con y sin sincronización inicial de hardware)

- El adaptador V24
- El UART de software
- Ejemplos de aplicación:
 - Intercambio de mensajes de texto, activado por un accionamiento de tecla sin sincronización inicial de hardware
 - Intercambio de mensajes de texto, activado por un accionamiento de tecla con control de memoria tampón de recepción y sincronización inicial de hardware

CMC 4-3: Salida de datos a impresora por medio de una interfase CENTRONICS (interfase paralela)

- El protocolo de la interfase CENTRONICS
- Ejemplos de aplicación:
 - Salida de un texto de varias líneas en impresora
 - Salida cíclica de valores de medición de temperatura en impresora

Instalación experimental

Para una realización efectiva, sin errores y segura del ensayo se representan a continuación y por separado:

- Tabla con todos los módulos, aparatos, líneas de medición y accesorios necesarios.
- Instalación de los aparatos en la mesa de trabajo.

Se recomienda exponer los manuales de instrucciones relativos a los módulos empleados en la realización del ensayo. Representan una ampliación valiosa para los respectivos ejercicios.

Los puntos de medición, los interruptores y los elementos de ajuste en los módulos se explican por separado. Los manuales de instrucciones también contienen indicaciones de aplicación fundamentales.

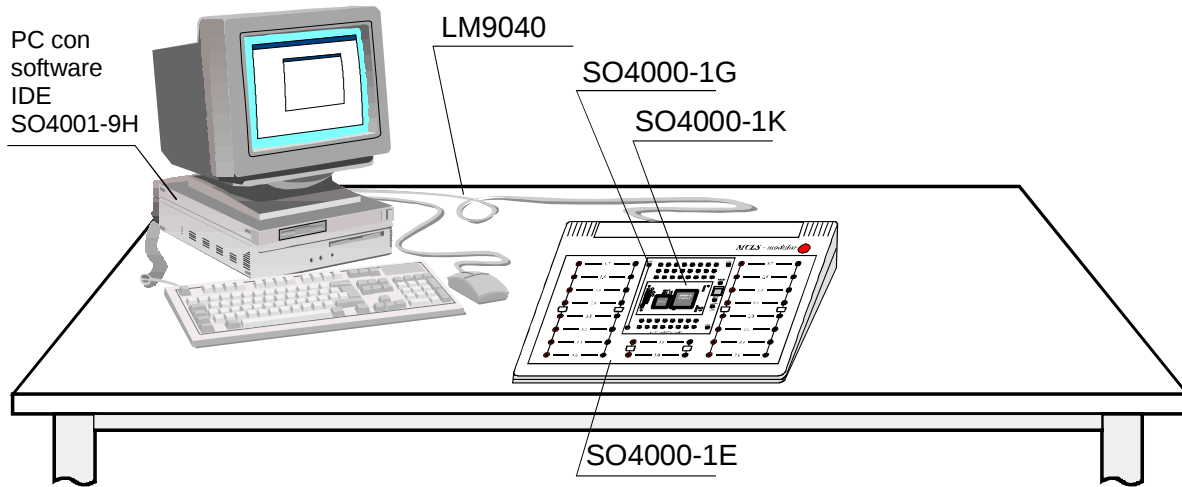


Fig. 01: Instalación en el puesto de trabajo

Componentes y aparatos necesarios

Unid	Denominación	Nº Id.
1 (2)	Plataforma con módulo de alim +5V	SO4000-1E
1 (2)	Fuente de alim. de enchufe AC 90...230V 45..65Hz, DC 9V 630mA	SO4000-1F
1 (2)	Módulo PSD1-Flash Controlador C515C	SO4000-1G
1 (2)	Unidad de adaptador de 8 bits	SO4000-1K
1 (2)	Unidad LED	SO4000-1P
1 (2)	Unidad de teclas	SO4000-1Q
1 (2)	Unidad de interruptores	SO4000-1R
1 (2)	Unidad de display 2 (display de LC)	SO4000-1B
1	Adaptador de BNC	SO4000-1T
1	Unidad de control de temperatura	SO4000-3C
1	Unidad de tensión	SO4000-3D
1 (2)	Adaptador de V24	SO4000-3J
1	Adaptador Centronics	SO4000-3K
1	Software IDE para MCLS (D,GB,F,E)	SO4001-9H
1	Sonda de prueba lógica	LM8101
1 (2)	Cable de interfase serial 9/9 polos	LM9040
1	Set de cables null-modem	LM9045
14 (20)	Cable de medición 2mm 15cm azul	SO5126-5K
10 (20)	Cable medición 2mm 15cm amarillo	SO5126-5M
1	CMC 4 Programación de interfaces de transmisión de datos	SH5019-1D
1	Maleta de almacenam. para MCLS	SO4000-1W

Nota: El número de piezas indicado entre paréntesis indica que en el segundo ensayo son necesarias dos instalaciones experimentales completas las cuales se unen con un cable null-modem para la comunicación entre ellas.

Introducción

La transmisión de información digitalizada es muy importante en la comunicación entre ordenadores, controles, aparatos de medición, sensores, actores y muchos más aparatos y componentes de la técnica de aparatos electrónicos, distinguiéndose el tipo de comunicación según diferentes criterios.

Si solamente 2 aparatos o componentes comunican entre ellos, se puede hablar de una *conexión punto a punto*. Un ejemplo típico es la conexión de una impresora a un PC como interfase periférica.

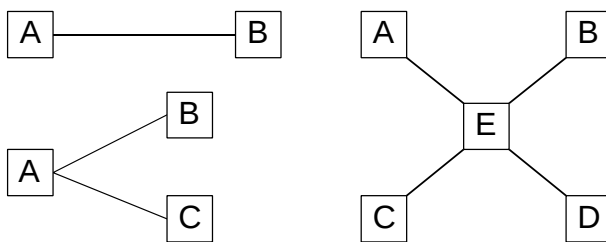
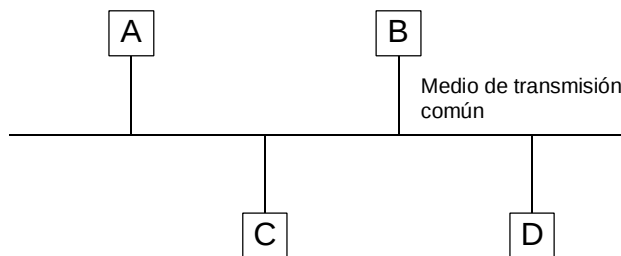


Fig. 02: Comunicación por medio de la conexión punto a punto

Sucede muy a menudo que varios aparatos o componentes utilizan el mismo medio para el intercambio de datos, de forma que se puede partir de un *sistema de bus*. Un ejemplo sería el bus I²C que une, en un solo módulo o en un solo aparato, varios IC's entre ellos para el intercambio de datos.



Sistema de bus = Topología de puntos múltiples

Fig. 03: Comunicación por medio de un sistema de bus

Otra característica de diferenciación es el *medio* que se utiliza para el transporte de los datos. Predomina la comunicación por hilos en las versiones técnicas más diferentes, haciéndose cada vez más importante la comunicación inalámbrica, p. ej. por medio de ondas radioeléctricas o mediante la luz.

Una importancia fundamental en todas las soluciones técnicas la tiene la cuestión de si la transmisión de la información digitalizada se realiza en forma paralela o serial. La ventaja principal de la transmisión serial es la posibilidad de poder transmitir información a grandes distancias, con tan solo 1 a 2 cables (también de forma inalámbrica). Sin embargo se presenta el inconveniente de que la velocidad de transmisión de datos es baja debido a la serialización. La transmisión paralela de datos es apropiada para tareas de comunicación con elevados requisitos hacia el tiempo, pero está limitada a distancias de transmisión cortas.

Transmisión serial de datos

En la comunicación serial existen numerosas ejecuciones técnicas que tocan especialmente los procedimientos para la sincronización, la codificación de señales, los protocolos de control y los procedimientos de seguridad. Por lo tanto, a continuación solamente contemplaremos más detalladamente algunos aspectos seleccionados de la transmisión asíncrona de datos.

En los procedimientos de transmisión serial, la transmisión asíncrona queda reservada para las velocidades de transmisión bajas. A menudo se realiza con la interfase V 24, existente en cada ordenador personal (PC) moderno en forma simplificada con 9 circuitos de intercambio. En la transmisión asíncrona se transmite, por regla general, un código de 8 bits del alfabeto de caracteres ASCII (incluyendo el bit de paridad).

Esta transmisión asíncrona se caracteriza por el hecho de que no existe ninguna trama de tiempo a través de un reloj durante todo el periodo de la transmisión.

El paso inicial que siempre constituye una interrupción del estado de reposo, sirve más bien para inicializar una fase síncrona que solamente se mantiene durante el tiempo que necesita la longitud de un carácter de 8 bits. A continuación sigue el paso de parada que puede consistir en 1,5 o 2 bits. El momento de la transmisión del siguiente carácter no está predeterminado, es decir, entre los caracteres puede haber pausas anisocrónicas de diferente duración. El periodo de tiempo entre el paso de inicio y el paso de parada sincroniza al receptor – ¡las pausas entre los caracteres caracterizan el asincronismo!

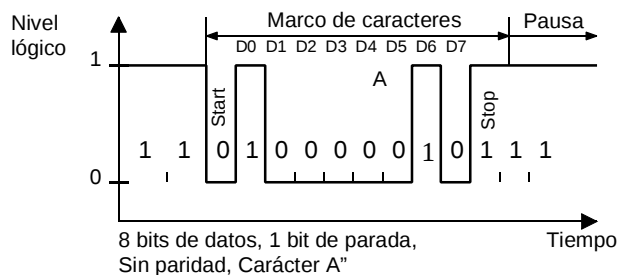


Fig. 04: Transmisión asíncrona de un carácter

En la transmisión síncrona, en cambio, se mantiene por medio de un reloj una trama de tiempo isocrónica fija durante todo el tiempo de la transmisión. El reloj es suministrado en el emisor por un generador de reloj.

Las interfaces normalizadas constituyen un requisito imprescindible para el intercambio internacional de datos. Por este motivo, el CCITT ha definido un número de interfaces en recomendaciones.

Las recomendaciones de la serie V, así como la recomendación V 24, se crearon originalmente para la transmisión asíncrona de datos por medio de la red telefónica analógica. Hoy en día, la interfase V 24 ha llegado a tener una gran importancia en la técnica de ordenadores, porque cada ordenador personal está equipado con esta interfase, aunque en forma simplificada. V 24 contiene la definición de, en total, 25 circuitos de intercambio y la ocupación de clavijas de un correspondiente conector de 25 polos. Es completada por la recomendación V 28 que contiene los parámetros eléctricos (tensiones, velocidad máxima de transmisión, aplicación de líneas no simétricas con tierra, etc.). A menudo se ve la denominación de interfase V 24/V 28, si se refiere tanto a los circuitos de intercambio como a los parámetros eléctricos. Un resumen de ambas recomendaciones también está comprendido en el estándar RS 232 C de la cía. Electronics Industries Association EIA (Asociación de la Industria Electrónica de EEUU); es decir, RS 232 C

incluye a V 24/V 28. Las normas DIN alemanas comprenden la interfase V 24/V28 en DIN 66020.

De los posibles 25 circuitos de intercambio de la V 24 hoy en día se utilizan, para la transmisión asíncrona mayoritariamente empleada, de manera casi exclusiva solamente 11 circuitos, como máximo, para la transmisión de datos. Los distintos circuitos de intercambio tienen diferentes denominaciones en los diferentes estándares indicados:

DIN	CCITT	EIA	Significado	DEE DÜE
D1	103	TD	Datos enviados	→
D2	104	RD	Datos recibidos	←
M1	107	DSR	UTD lista para funcionar	←
M2	106	CTS	Listo para enviar	←
M3	125	RI	Llamada entrante	←
M4	112		Conectar alta velocidad de transmisión	←
M5	109	DCD	Nivel de señal recibida	←
S1.2	108/2	DTR	TERM listo para funcionar	→
S2	105	RTS	Conectar unidad emisora	→
S4	111	SEL	Conectar alta velocidad de transmisión	→
T1	113	TCK	Reloj paso envío TERM	←
T2	114	XCR	Reloj paso envío UTD	←
T4	115	RCK	Reloj paso recepc. UTD	←
E2	102	GND	Tierra de servicio	↔

En la conexión de ordenadores personales por medio de la interfase V 24 sucede a menudo que se debe realizar una adaptación de conector de un conector de 25 polos a un conector de 9 polos.

Señal	Nº clavija del conector de 25 polos	Nº clavija del conector de 9 polos
DCD	8	1
RD	3	2
TD	2	3
DTR	20	4
GND	7	5
DSR	6	6
RTS	4	7
CTS	5	8
RI	22	9

Mediante estos circuitos de datos y de señales son posibles muchas variantes de la comunicación serial que se utilizan en la práctica.

Comunicación paralela

La transmisión de datos paralelos se realiza en interfases periféricas (p. ej. bus IEC, interfase CENTRONICS) y en interfases de comunicación internas de aparatos (p. ej. buses de procesador, puertos TTL). Como ejemplo para una interfase de comunicación paralela contemplaremos más detalladamente la interfase CENTRONICS que está concebida para la activación de impresoras y plotter desde un PC.

La salida de bytes (caracteres de impresión y de control) del PC por medio de la interfase hacia una impresora mediante una sincronización inicial („Handshake“) sencilla es característica. Para ese fin, la interfase dispone de 8 circuitos de datos, así como de algunos circuitos de control y de estado. La siguiente tabla muestra la ocupación de los contactos de la interfase Centronics del módulo empleado en el sistema de aprendizaje de microcontrolador.

Pin	Señal	Función
1	/STROBE	Datos están listos para captación
2	D0	Bit de datos D0
3	D1	Bit de datos D1
4	D2	Bit de datos D2
5	D3	Bit de datos D3
6	D4	Bit de datos D4
7	D5	Bit de datos D5
8	D6	Bit de datos D6
9	D7	Bit de datos D7
10	/ACKNLG	Confirmación de capt. de datos
11	BUSY	Estado de aparato, con Busy=1 aparato ocupado
12	sin ocupar	
13	sin ocupar	
14	sin ocupar	
15	/ERROR	Estado de aparato, con /ERROR=0 error de aparato
16	sin ocupar	
17	sin ocupar	
18	GND	Masa de aparato
19	GND	Masa de aparato
20	GND	Masa de aparato
21	GND	Masa de aparato
22	GND	Masa de aparato
23	GND	Masa de aparato
24	GND	Masa de aparato
25	GND	Masa de aparato

Un juego típico de señales en la interfase CENTRONICS muestra la fig. 05. Después que la fuente de datos (p.ej. un PC) haya preparado los datos (D0 a D7), señaliza por medio de un impulso STROBE que hay datos válidos para captar. La impresora evalúa esta información y empieza con la captación y el procesamiento del carácter de impresión. Durante esta captación de datos, la impresora pone el circuito BUSY al nivel 1 (ocupado). Una vez realizada la captación de datos, el circuito Busy vuelve al nivel 0, siendo generado al mismo tiempo un corto impulso de confirmación (impulso /ACKNLG < 4μs).

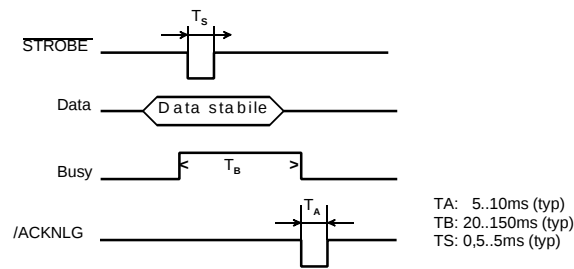


Fig. 05: Curso de la sincronización inicial

Bibliografía:

- [1] Beierlein, T.; Hagenbruch, O.: Taschenbuch Mikroprozessortechnik (Manual de la técnica de microprocesador), Fachbuchverlag Leipzig im Carl Hanser Verlag, 3ra. edición 2004
- [2] Dembowski, K.: Computerschnittstelln und Bussysteme (Interfaces de ordenador y sistemas de bus), Hüthig Verlag Heidelberg 1997

CMC 4-1 Comunicación de datos mediante la interfase serial del microcontrolador

La transferencia serial asíncrona de datos se presenta mediante la comunicación entre el microcontrolador y el PC, utilizándose la interfase serial (UART „on chip“) del microcontrolador.

La interfase serial „on chip“

La interfase serial disponible en el microcontrolador C515C es apta para dúplex total. La entrada está conectada en tampón, de forma que siempre se puede recibir otro byte antes de que el anterior haya sido evaluado. En el envío y también en la recepción se accede, por medio del registro de funciones especiales (SFR) SBUF, a registros físicamente separados. Un comando de escritura o un comando de lectura decide qué registro se va a utilizar.

La interfase serial se puede utilizar para tareas de comunicación serial tanto asíncronas como síncronas. Son posibles, en total, 4 modos de servicio diferentes:

- Modo 0: transmisión síncrona
- Modo 1: transmisión asíncrona para 8 bits con vel. de transmisión variable (UART de 8 bits)
- Modo 2: transmisión asíncrona para 9 bits con val. de transmisión fija (UART de 9 bits)
- Modo 3: transmisión asíncrona para 9 bits con vel. de transmisión variable (UART de 9 bits)

Los detalles relativos a los diferentes modos están indicados, p. ej., en el manual del usuario del C515C o en los manuales de los correspondientes derivados.

De cara a los experimentos descritos a continuación, es importante el modo 1 (UART de 8 bits), en el que se transmiten de forma serial caracteres de un tamaño de 8 bits, o sea 10 bits, en total, con el bit de inicio y el bit de parada.

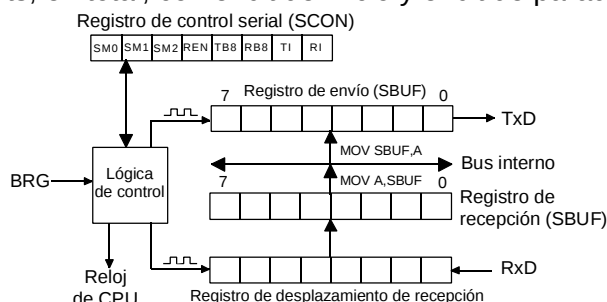


Fig. 101: Estructura básica de la interfase asíncrona en el modo 1

Para el modo 1, el reloj de transmisión se puede generar de dos maneras diferentes. Por una parte se puede utilizar el temporizador Timer 1 que ya no estará disponible para otras aplicaciones. Sin embargo, para el presente caso de aplicación ($f_{osc} = 6 \text{ Mhz}$) es preferible el generador interno de velocidad de transmisión (BRG), porque se pueden generar de manera muy fácil las velocidades de transmisión estandarizadas de 4800 ó 9600 baudios. Para conectar el BRG, se debe poner el bit BD en el SFR ADCON. Ahora se puede

programar la velocidad de transmisión por medio de SMOD=1 en el SFR PCON a 9600 baudios o con SMOD=0 a 4800 baudios.

La interfase serial se configura mediante el SFR SCON. Véanse otros detalles relativos al ajuste de la velocidad de transmisión en la información compacta del microcontrolador en el anexo del manual de instrucciones, o en los correspondientes manuales.

En el siguiente ejemplo de inicialización se programa la interfase serial para el modo 1 y 9600 baudios sin servicio de interrupción:

```

; -----
;   Inicialización de la interfase serial para el modo 1
; -----
INIT_SIO:
    ORL    PCON,#10000000B    ; SMOD = 1 para 9600 baudios
    SETB   BD                 ; Conectar
                                ; generador para vel. de transm.
    MOV     SCON,#01000000B    ; Transmisión asíncrona
                                ; con 8 bits de datos
    RET
; -----

```

Un papel central en la recepción o en el envío de datos por medio de la interfase serial asíncrona juegan los bits TI y RI en el SFR SCON (ver información compacta). RI se pone si en la memoria tampón de entrada (SBUF) se ha recibido un carácter completo. TI se pone si la memoria tampón de envío (SBUF) está completamente vaciada, o sea si el carácter ha sido enviado por completo. Ambos bits se pueden vigilar en el sondeo o se pueden utilizar para la activación de una interrupción, debiéndose realizar un respectivo reposicionamiento de los bits por medio del software. Si la interfase se utiliza en servicio de interrupción, el tratamiento se deberá realizar, debido al vector de interrupción común para enviar y recibir, en una rutina de servicio de interrupción (ISR) común.

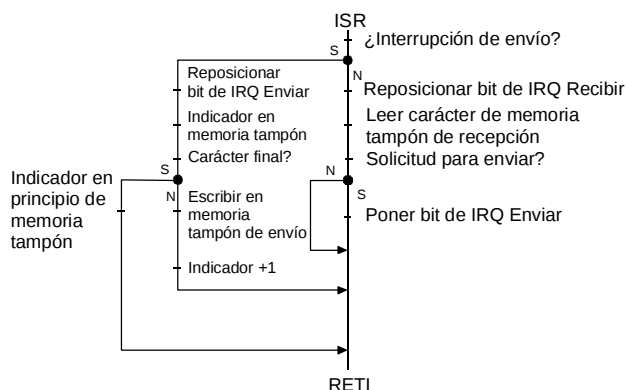
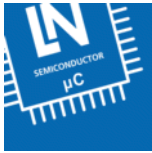


Fig. 102: Estructura de la ISR para enviar y recibir



Aparatos y accesorios necesarios

Unid	Denominación	Nº Id.
1	Plataforma con mód. de alim. +5V	SO4000-1E
1	Fuente de alim. de enchufe AC 90...230V 45..65Hz, DC 9V 630mA+	SO4000-1F
1	Módulo PSD1-Flash Controlador C515C	SO4000-1G
1	Unidad de adaptador de 8 bits	SO4000-1K
1	Unidad de LED	SO4000-1P
1	Unidad de teclas	SO4000-1Q
1	Unidad de tensión	SO4000-3D
1	Software IDE para MCLS (D,GB,F,E)	SO4001-9H
1	Cable de interfase serial 9/9 polos	
3	Cable de medición 2mm 15cm azul	SO5126-5K
1	Cable medición 2mm 15cm amarillo	SO5126-5M
1	CMC 4 Programación de interfaces de transmisión de datos	SH5019-1D

Instalación general de aparatos

¡Separe la alimentación de corriente del MCLS-modular®!

Conecte la plataforma de trabajo por medio del cable de interfase serial a su PC.

Encaje la unidad de adaptador de 8 bits con el Módulo PSD1-Flash adaptado en ésta en el campo de casquillos 4.0 - 4.5 en el centro de la plataforma. Seleccione los módulos de experimentación necesarios del ejercicio y conéctelos a los puertos elegidos por usted.

Conecte la alimentación de corriente al MCLS-modular®.

Nota relativa a la seguridad:

¡Asegúrese de conectar correctamente los potenciales de tensión (izda U5P, dcha GND)! No se deben aplicar tensiones extrañas a los puntos de medición llevados hacia fuera ¡ya que esto podría conducir a la destrucción de los componentes de las placas!

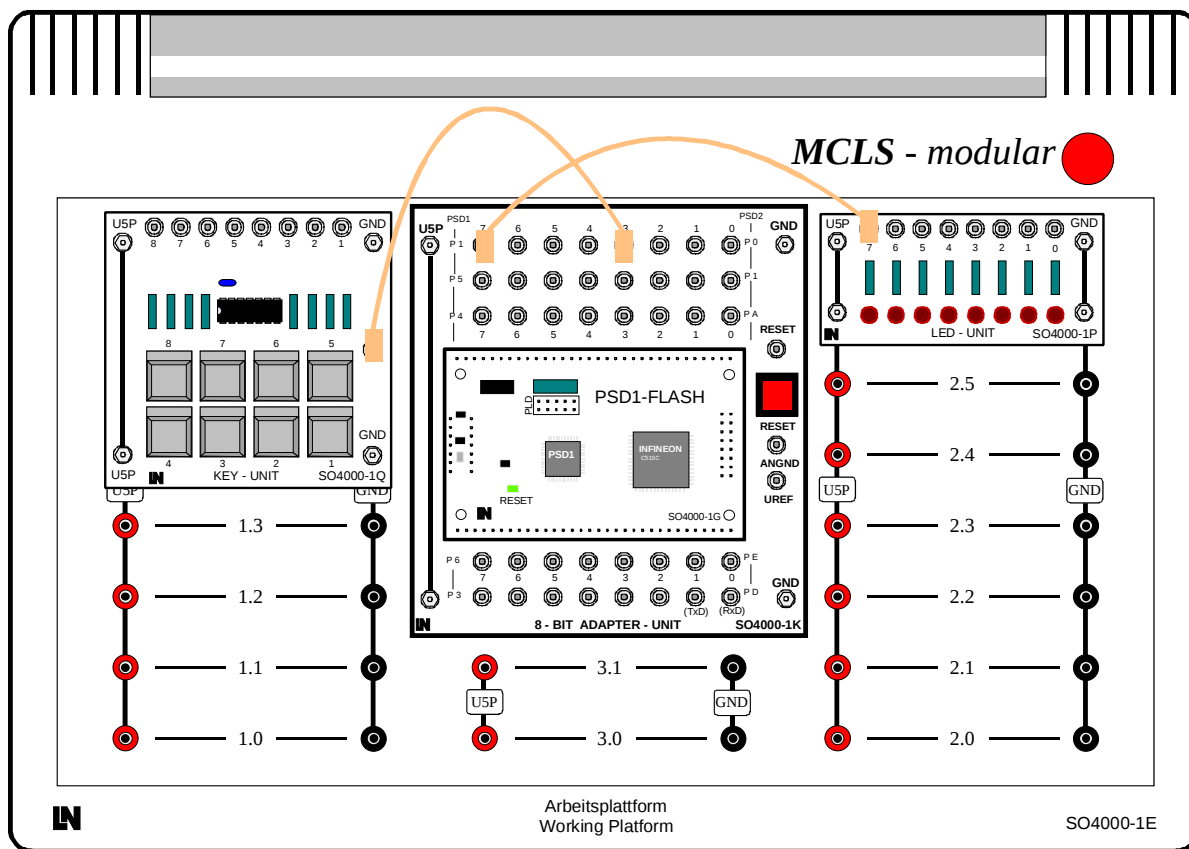


Fig. 103: Instalación de aparatos del ensayo CMC 4-1A

Ensayo CMC 4-1A, Transferencia de datos seriales hacia el PC

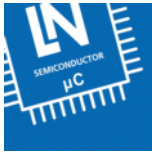
Pulsando una tecla se iniciará en el controlador una transferencia de datos seriales hacia el PC. La cadena de caracteres que llega al PC se deberá observar mediante un programa terminal.

¿Qué conocimientos se pretenden transmitir?

- ¿Cómo se inicializa la interfase serial del C515C para una transferencia asíncrona de datos?
- ¿Dónde se introduce el vector de interrupción común para enviar y recibir?
- ¿Cuál es la estructura de la ISR?
- ¿Cómo se realiza el acceso a la cadena de caracteres a enviar?
- ¿Cómo se realiza un „Line Feed“ o un „Carriage Return“ para el programa terminal?

Explicaciones relativas al ejercicio:

La pulsación de una tecla cualquiera de la unidad de teclas deberá activar la interrupción externa 6. Mediante esta interrupción (pulsación de tecla) se deberá iniciar el envío de una cadena de caracteres al PC, utilizándose la unión serial estándar entre el MCLS-modular y el PC (cable serial de 9 polos LM9040).



La cadena de caracteres que llega al PC se visualiza por medio del programa terminal integrado en el IDE. La interfase serial se deberá utilizar con 9600 baudios, 8 bits de datos, 1 bit de inicio y 1 de parada sin bit de paridad en modo de servicio asíncrono.

Para el control de la pulsación de tecla que inicia la transferencia de datos seriales se deberá activar un LED por medio del pin de puerto P1.7.

Ejercicios y soluciones

Croquice un diagrama de bloques sencillo del que se desprenda la asociación correcta de los pins del microcontrolador a la salida INT de la unidad de teclas, a la unidad de LED y a la interfase serial. Complete el croquis con la indicación del flanco que se genera por medio de una pulsación de tecla.

Realice la instalación experimental según la fig. 103.

Escriba el subprograma para la inicialización de la interfase serial utilizando el modo de servicio de interrupción.

```
; -----  
;   Inicialización de la interfase serial  
; -----  
INIT_SIO:
```

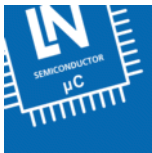
Croquice la estructura del programa para la ISR de la transferencia de datos seriales en forma de un organigrama de programa.

Defina como texto fuente de ensamblador la cadena de caracteres a enviar incluyendo los caracteres de control para „Line Feed“ y „Carriage Return“.

```
; -----  
; Cadena de caracteres para la transferencia serial al PC  
; -----  
TEXT01:
```

Diseñe el texto fuente de ensamblador para la ISR sobre la base del organigrama de programa incluyendo el acceso por caracteres a la cadena de caracteres.

```
; -----  
; ISR para interfase serial  
; -----  
ISR_SIO:
```



Introduzca los vectores de interrupción para la interrupción de tecla y para la interrupción de envío en la tabla de vectores de interrupción.

; ----- Introducción del IV para el UART -----

Ensaye la interrupción de tecla mediante un correspondiente punto de ruptura.

- El punto de ruptura se debe poner de forma óptima en la posición de introducción del IV (006BH).

Ensaye la función del programa mediante el programa terminal integrado.

- Cargue e inicie el programa por medio del programa terminal.
- Observe la visualización del texto en la ventana terminal al activarse una interrupción de tecla.

Nota:

Para trabajar con el terminal integrado, debe convertir el perfil integrado „PSD1-C515C-AS“ en un perfil definido por el usuario y, para trabajar con el proyecto, configurarlo para los ensayos.

Proceda de la siguiente manera:

- Proyecto → Microcontrolador → Seleccionar perfil integrado: „PSD1-C515C-AS“.
- Convertir en un perfil definido por el usuario pulsando la tecla de copia a la derecha del campo de selección.
- Asignar un nuevo nombre de perfil y confirmar el diálogo.
- Poner selección de perfil en „Perfiles de usuario“.
- Seleccionar el perfil nuevamente definido.
- Desactivar depurador y activar el terminal.
- Configurar la herramienta Terminal bajo Proyecto → Herramientas → Terminal con los siguientes parámetros:

Terminal a utilizar:	Terminal integrado
Archivo para descargar:	\$P\ \$F.hex
Comando inicio:	g0, conmutador en pos. central
- Ajuste de terminal bajo Proyecto → Terminal (interno):

Tipo de unión:	con cable
Juego de caracteres del monitor:	ANSI
Interfase actual:	COMx (su interfase actual)
Ampliar con:	CR

Lista de programa CMC 4-1A

```

; *****
; * Proyecto:          Enviar un texto por medio de la      *
; *                  interfase serial (SIO) del MC al PC    *
; * Archivo fuente:    cmc41a.asm                          *
; * Herramientas MC:   as, p2hex, terminal integrado        *
; * Perfil:            Perfil de usuario PSD1-C515C-AS mod. *
; * Recursos MC:       Interrupción ext 6 por medio de unid. *
; *                  de teclas, LED para control en P1.7    *
; * Editor:            HAG                                  *
; *****

; ----- Descripción de funcionamiento -----
; Pulsando una tecla (unidad de TECLAS: salida de interrupción
; suministra una señal para la interrupción externa 6) se inicia una
; transferencia de datos seriales de la interfase serial (SIO)
; del C515C al PC.
; En el lado del PC se puede observar la cadena de datos entrante
; mediante el programa terminal integrado.
; -----
; Modo de servicio del UART "on chip": 9600 baudios, 8 bits de datos,
; 1 bit de inicio y 1 bit de parada, sin bit de paridad
; -----
; A P1.7 está conectado un LED para el control de la pulsación de
; tecla (inicio de la transferencia de datos).
; -----

; -- Indicaciones para el ensayo al utilizar un terminal integrado --
; -----
; 1. Activación del terminal integrado en el perfil de proyecto
; 2. Especificar el archivo para descargar por medio de
;    -> Árbol de proyecto -> Herramientas -> Terminal
;    -> Archivo para descargar
; 3. Acordar los comandos para el inicio del programa en la ventana
;    de comandos mediante -> Árbol de proyecto -> Herramientas
;    -> Terminal -> Comandos (inicio de programa via cadena de
;    comandos 'G0')
; 4. Descargar el archivo Hex al destino del terminal
;    - Activar la ventana de mensajes del terminal
;    - Iniciar la descarga accionando
;      el ícono de herramienta "Download"
; 5. Inicio del programa accionando el correspondiente ícono para
;    RUN en la ventana de mensajes
; 6. Observe la salida del texto en la ventana de mensajes del
;    terminal pulsando una tecla en la unidad de TECLAS y el cambio
;    de estado del LED en P1.7 tras cada pulsación de tecla
; #####

; -----Acuerdos necesarios para el ensamblador-----

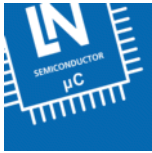
        CPU      80515          ; Selección de microcontrolador
        INCLUDE  c515c.inc      ; Definiciones de SFR

        SEGMENT code          ; A partir de aquí segmento de código
        ORG      0000H         ; Primer comando a partir de la
                                ; dirección 0000H

; -----
        JMP      START         ; Salto al programa principal

; ----- Introducción del IV para el UART -----
        ORG      0023H         ; Posición IV para el UART (SIO)
        JMP      ISR_SIO       ; Vector de interrupción (IV)

```



```
; ----- Introducción del IV para la int. ext. 6 -----
      ORG    006BH          ; Posición IV para la int. ext. 6
      JMP    ISR_INT6       ; Vector de interrupción (IV)

; *****
; *                               Programa principal                               *
; *****
START:
      CALL   INIT_INT6      ; Inicializar interrupción ext. 6
      CALL   INIT_SIO       ; Inicializar interfase serial
      SETB   EAL            ; Autorizar aceptaciones de interrupción
                           ; por CPU
      JMP    $              ; Bucle sinfín como "final"
                           ; del programa principal

; -----
;                               Inicialización de la interrupción ext. 6
;                               - Disparo por flanco 0-1 -
; -----
INIT_INT6:
      SETB   EX6            ; Autorizar interrupción externa 6
      RET

; -----
;                               Inicialización de la interfase serial
; -----
INIT_SIO:
      ORL    PCON,#10000000B ; SMOD = 1 para 9600 baudios
      SETB   BD              ; Conectar generador para vel.
                           ; de transmisión
      MOV    SCON,#01000000B ; Transmisión asíncrona
                           ; con 8 bits de datos
;      SETB   ES              ; Autorizar interrupción SIO
      RET

; -----
;                               ISR para interrupción ext. 6
; -----
ISR_INT6:
      CPL    P1.7           ; Con pulsación de tecla invertir estado LED

      MOV    SBUF,#' '      ; Carácter mudo en memoria tampón de envío
      SETB   TI              ; Poner solicitud para interrupción de envío
      MOV    DPTR,#TEXT1    ; Poner indicador en texto
      SETB   ES

      CALL   TIEMPO          ; Tiempo para eliminación de rebotamiento
                           ; de tecla
      CLR    IEX6            ; Reposicionar bit de solicitud de
                           ; interrupción
      RETI

; -----
;                               ISR para interfase serial
; -----
ISR_SIO:
      PUSH   ACC
      JNB    TI,ISR_END     ; si no hay interrupción de envío, al
                           ; final de la ISR
; .....
;      CLR    TI              ; Borrar solicitud de envío
;      MOV    A,#0            ; Índice a 0
;      MOVC   A,@A+DPTR       ; Acceso a cadena de caracteres
;      JZ     ISR0            ; Si carácter final alcanzado, a ISR0
```

```

        MOV     SBUF,A           ; Si no, cargar memoria tampón de envío
        INC     DPTR            ; Indicador en cadena de caracteres +1
; .....
ISR_END:
        POP     ACC
        RETI

ISR0:
        CLR     ES              ; Bloquear interrupción por SIO
        JMP     ISR_END

; -----
;          Tiempo de eliminación de rebotamiento aprox. 50ms
; -----
TIEMPO:
        MOV     R7,#98
T01:    MOV     R6,#000H
T00:    DJNZ    R6,T00
        DJNZ    R7,T01
        RET

; -----
;          Cadena de caracteres para la transferencia serial al PC
; -----
TEXT1:
        DB      "Hallo PC - the MCLS-modular transmit serial! "
        DB      0DH, 0AH,0      ; Carácter de control

END

```

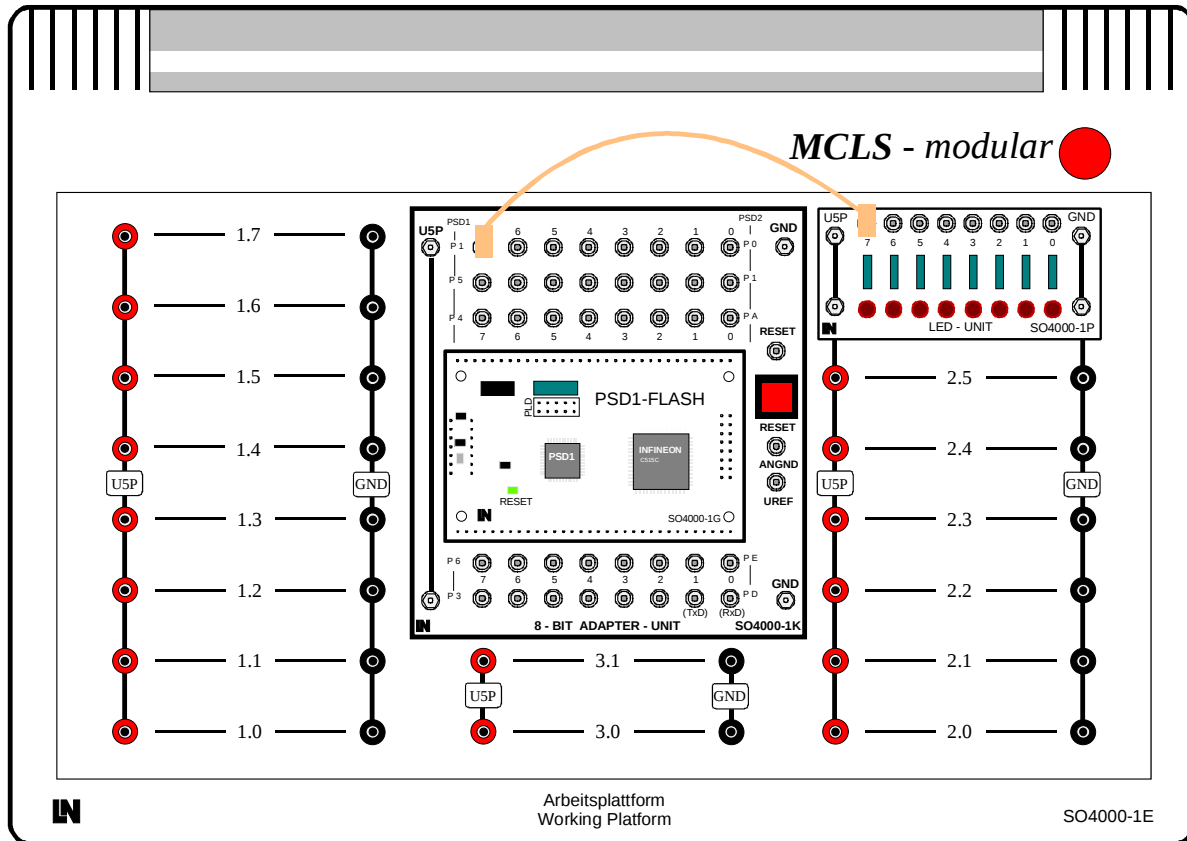


Fig. 104: Instalación de aparatos del ensayo CMC 4-1B

Ensayo CMC 4-1B, Como consecuencia de un comando envía una respuesta (texto) al PC

Como consecuencia de un comando desde el PC (solicitud) el microcontrolador envía una respuesta (texto) al PC.

¿Qué conocimientos se pretenden transmitir?

- Utilización del terminal integrado para enviar un comando de solicitud al microcontrolador.
- ¿Cómo se activa el canal de recepción de la interfase serial?
- ¿Cómo se evalúa la solicitud de envío dentro de la ISR en el lado del microcontrolador?

Explicaciones relativas al ejercicio:

Como consecuencia de una solicitud de envío desde el PC, el microcontrolador deberá contestar con un texto. Para la comunicación se utiliza la conexión serial estándar entre el MCLS-modular y el PC (cable serial de 9 polos LM9040).

Como solicitud de envío se envía el carácter „@“ por medio del terminal integrado desde el PC al microcontrolador. La respuesta del microcontrolador (cadena de caracteres) se visualiza por medio del programa terminal integrado. La interfase serial se deberá utilizar con 9600 baudios, 8 bits de datos, 1 bit de inicio y 1 bit de parada sin bit de paridad en modo de servicio asíncrono.

Ejercicios y soluciones

Croquice, para la instalación experimental, un diagrama de bloques sencillo.

Realice la instalación experimental según la fig. 104.

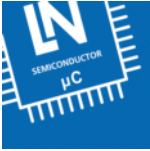
Acuerde la solicitud de envío „@“ en el terminal integrado.

↳ Observe el manual de instrucciones del terminal integrado.

¿Cómo se activa el canal de recepción de la interfase serial en el microcontrolador?

Escriba el texto fuente de ensamblador para la inicialización de la interfase serial.

```
; -----  
;      Inicialización de una interfase serial  
; -----  
INIT_SIO:
```



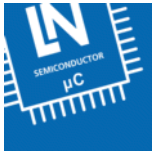
Croquice la estructura de programa para la ISR de la transferencia de datos seriales en forma de un organigrama de programa.

Diseñe el texto fuente de ensamblador para la ISR sobre la base del organigrama de programa.

```
; -----  
;          ISR para interfase serial  
; -----  
ISR_SIO:
```

Ensaye la función del programa mediante el programa terminal integrado.

- ↗ Cargue e inicie el programa mediante el programa terminal.
- ↗ Observe la visualización del texto en la ventana terminal al enviarse el carácter de solicitud acordado „@“.



Lista de programa CMC 4-1B

```
; *****
; * Proyecto:      Como consecuencia de un comando del PC se envía *
; *              un texto por medio de la interfase serial (SIO) *
; *              del MC al PC *
; * Archivo fuente: cmc41b.asm *
; * Herramientas de MC: as, p2hex, terminal integrado *
; * Perfil:        Perfil de usuario PSD1-C515C-AS mod. *
; * Recursos MC:   LED a P1.7 para ver si llegan datos del PC *
; * Editor:        HAG *
; *****

; ----- Descripción de funcionamiento -----
; Un comando (carácter '@') se envía por medio del terminal integrado
; desde el PC al MC,
; si el comando acordado se puede recibir correctamente por el MC,
; se envía una correspondiente respuesta al PC;
; en caso de otro carácter u en caso de un error no hay respuesta
; alguna.
; -----
; Modo de servicio del UART on chip: 9600 baudios, 8 bits de datos, 1
; bit de inicio y 1 bit de parada, sin bit de paridad, receptor ON
; -----
; A P1.7 está conectado un LED para controlar los datos que llegan
; desde el PC.

; - Indicaciones para el ensayo al utilizarse el terminal integrado -
; -----
; 1. Activar el terminal integrado en el perfil de proyecto
; 2. Especificar el archivo para descargar por medio de -> Árbol de
; proyecto -> Herramientas -> Terminal -> Archivo para descargar
; 3. Acordar los comandos en la ventana de comandos mediante -> Árbol
; proyecto -> Herramientas -> Terminal -> Comandos:
; - Inicio de programa por medio de 'G0'
; - Carácter a enviar '@' en ventana de comandos del terminal
; 4. Descargar el archivo Hex en el destino mediante el terminal
; - Activar la ventana de mensajes del terminal
; - Iniciar la descarga accionando
; el ícono de herramienta "Download"
; 5. Iniciar el programa accionando el ícono correspondiente a
; RUN en la ventana de mensajes
; 6. Accionar el ícono de comandos seleccionado como comando para
; el carácter a enviar (p.ej. ícono "Single Step") en la ventana
; de mensajes
; 4. Observe en la ventana de mensajes el texto de respuesta del
; destino y el cambio del LED en P1.7 después de cada activación
; de comando
; #####

; -----Acuerdos necesarios para el ensamblador -----

      CPU      80515          ; Selección de microcontrolador
      INCLUDE  c515c.inc      ; Definiciones de SFR
; -----
      SEGMENT code          ; A partir de aquí segmento de código
      ORG      0000H         ; Primer comando a partir de la
                           ; dirección 0000H

      JMP      START         ; Salto al programa principal

; ----- Introducción del IV para el UART -----
      ORG      0023H         ; Posición IV para el UART (SIO)
```

```

        JMP     ISR_SIO           ; Vector de interrupción (IV)

        ORG     0100H
; *****
; *                               Programa principal                               *
; *****
START:
        CALL    INIT_SIO         ; Inicializar interfase serial
        SETB    EAL              ; Autorizar aceptaciones de interrupción
                                ; por CPU
        MOV     DPTR,#TEXT2      ; Poner registro de indicador al principio
                                ; del texto
        CLR     P1.7            ; Conectar LED
        JMP     $               ; Bucle sinfín como "final"
                                ; del programa principal

; -----
;                               Inicialización de la interfase serial
; -----
INIT_SIO:
        ORL     PCON,#10000000B ; SMOD = 1 para 9600 baudios
        SETB    BD              ; Conectar generador para velocidad
                                ; de transmisión
        MOV     SCON,#01010000B ; Transmisión asíncrona
                                ; con 8 bits de datos, receptor ON!
        SETB    ES              ; Autorizar interrupción SIO
        RET

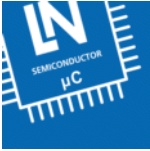
; -----
;                               ISR para interfase serial
; -----
ISR_SIO:
        PUSH    ACC
        JB      TI,I_SEND        ; ¿Interrupción de envío?
; .....
; Tratamiento de la interrupción de recepción
        CLR     RI              ; Reposicionar bit de solicitud de
                                ; interrupción Recibir
        MOV     A,SBUF          ; Coger carácter de la memoria tampón de
                                ; recepción
        CJNE    A,#'@',I_END     ; ¿Solicitud de PC correcta?
        SETB    TI              ; Poner bit de solicitud de
                                ; interrupción Enviar
        JMP     I_END

; .....
; Tratamiento de la interrupción de envío
I_SEND:
        CPL     P1.7            ; Conmutar LED como indicativo
        CLR     TI              ; Borrar solicitud de envío
        MOV     A,#0            ; Índice a 6 para comando correcto
        MOVC    A,@A+DPTR       ; Acceso a cadena de caracteres
        CJNE    A,#0FFH,I_CHARACTER ; Si carácter final no alcanzado,
                                ; al envío del carácter

; .....
; Tratamiento final del envío
        MOV     DPTR,#TEXT2      ; Poner registro de indicador
                                ; al principio del texto
        JMP     I_END

; .....
; Enviar carácter al PC
I_CHARACTER:
        MOV     SBUF,A           ; Si no, cargar memoria tampón de envío
        INC     DPTR            ; Indicador en cadena de caracteres +1
; .....

```



```
I_END:
    POP    ACC
    RETI

; -----
;          Cadena de caracteres para transferencia serial al PC
; -----
TEXT2:
    DB     "OK - here is my reply for you!" ; Cadena de caracteres
    DB     0DH, 0AH, 0FFH                  ; Carácter de control

END
```

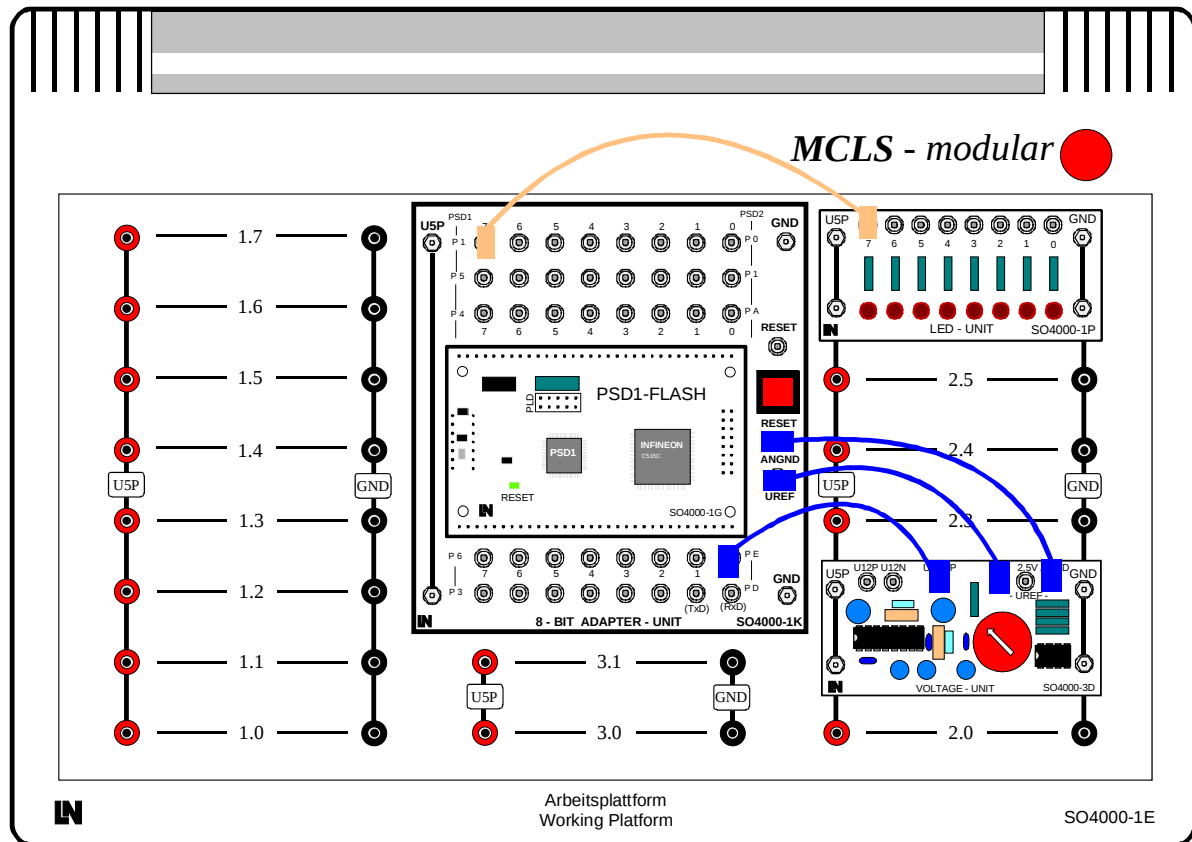


Fig. 105: Instalación de aparatos del ensayo CMC 4-1C

Ensayo CMC 4-1C, Cíclicamente se transmite un valor de medición del convertidor A/D „on chip“ del microcontrolador al PC

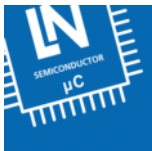
Cíclicamente, en intervalos de un segundo, se transmite un valor de medición de tensión del convertidor A/D „on chip“ del microcontrolador al PC. En el lado del PC se pueden observar los valores medidos entrantes mediante el programa terminal.

¿Qué conocimientos se pretenden transmitir?

- Utilización del terminal integrado para la visualización de valores medidos
- Generación de interrupciones cíclicas con el Timer 2 como marcador de tiempo
- Inicialización del convertidor A/D „on chip“
- Preparación del valor medido binario para la transmisión de datos seriales orientada en caracteres

Explicaciones relativas al ejercicio:

El Timer 2 deberá iniciar cíclicamente, en intervalos de un segundo, el envío de un valor de medición de tensión. Para la medición se utiliza el convertidor A/D „on chip“ del C515C. Los valores medidos se deberán transmitir con número correlativo y unidad de medida desde el microcontrolador al PC.



Para la comunicación se utilizará la conexión serial estándar entre el MCLS-modular y el PC (cable serial de 9 polos LM9040).

La interfase serial se utilizará con 9600 baudios, 8 bits de datos, 1 bit de inicio y 1 bit de parada, sin bit de paridad, en modo de servicio asíncrono.

Para el control de la interrupción de temporizador se conectará un LED a P1.7.

Ejercicios y soluciones

Realice la instalación experimental según la fig. 105.

Croquice, para la instalación experimental, un diagrama de bloques sencillo.

Escriba el programa ensamblador para la inicialización del Timer 2. La interrupción cíclica se deberá realizar en intervalos de 50 ms.

```
; ----- Constante de tiempo para el Timer 2-----  
  
;  
  
;  
;          Inicialización del Timer 2  
;          .   - cada 50 ms una interrupción -  
;  
INIT_T2:
```

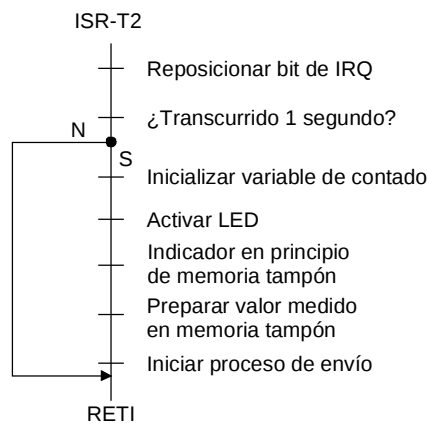
Escriba el programa ensamblador para la inicialización del convertidor A/D „on chip“ con conversión continua y entrada AN0!

```

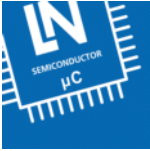
; -----
;      Inicializar convertidor A/D „on chip“
; -----
INIT_ADU:

```

Croquice la estructura de programa para la ISR del Timer 2 en forma de un organigrama de programa.



¿Qué conversiones son necesarias, partiendo del valor medido binario de 8 Bit (ADDATH), para una transferencia de datos seriales orientada en caracteres?



Diseñe el texto fuente de ensamblador para la ISR sobre la base del organigrama del programa.

```
; -----  
;           ISR para Timer 2  
; -----  
ISR_T2:
```

Ensaye la función del programa mediante el programa terminal integrado.

- ↪ Cargue e inicie el programa mediante el programa terminal.
- ↪ Observe la salida cíclica de valores medidos en la ventana del terminal.

Lista de programa CMC 4-1C

```

; *****
; * Proyecto: Envío cíclico de valores medidos por *
; * medio de la SIO del MC al PC *
; * Archivo fuente: cmc41c.asm *
; * Herramientas de MC: as, p2hex, terminal integrado *
; * Perfil: Perfil de usuario PSD1-C515C-AS mod. *
; * Recursos MC: LED para control en P1.7 *
; * Editor: HAG *
; *****

; ----- Descripción de funcionamiento -----
; Timer 2 inicia cíclicamente, siempre después de un segundo, el
; envío de un valor de medición de tensión del convertidor A/D "on
; chip" en la forma: '*NR abcd mV' (NR = número correlativo,
; abcd = valor de medición de tensión, mV = unidad de medida).
; En el lado del PC se pueden observar los valores medidos que llegan
; mediante el programa terminal integrado.
; Los valores de tensión alternos se pueden realizar p. ej. por
; medio del módulo de experimentación 'Generador de funciones' en
; la entrada (AN0)
; -----
; Modo de servicio del UART "on chip": 9600 baudios, 8 bits de
; datos, 1 bit de inicio y 1 bit de parada, sin bit de paridad,
; solamente servicio de envío
; -----
; A P1.7 está conectado un LED para controlar la interrupción del
; temporizador (inicio de la transferencia de datos).

; - Indicaciones para el ensayo al utilizarse el terminal integrado -
; -----
; 1. Activar el terminal integrado en el perfil de proyecto
; 2. Especificar el archivo a descargar por medio de -> Árbol
; de proyecto -> Herramientas -> Terminal -> Archivo a descargar
; 3. Acordar el comando para el inicio del programa en la
; ventana de comandos mediante -> Árbol de proyecto -> Herramientas
; -> Terminal -> Comandos (inicio de programa por medio de cadena
; de comandos 'G0' = RUN)
; 4. Descargar el archivo Hex al destino mediante el terminal
; - Activar la ventana de mensajes del terminal
; - Inicio de la descarga accionando
; el ícono de herramienta "Download"
; 5. Iniciar el programa accionando el ícono para
; RUN en la ventana de mensajes
; 6. Observe la salida continua de valores medidos en la ventana
; de mensajes del terminal y la conmutación del LED
; en P1.7 después de cada valor medido
; #####

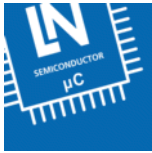
; -----Acuerdos necesarios para el ensamblador -----

CPU 80515 ; Selección de microcontrolador
INCLUDE c515c.inc ; Definiciones de SFR
INCLUDE bitfuncs.inc

; ----- Constante de tiempo para el Timer 2 -----
TCO EQU -50000 ; 50000*1µs hasta desbordamiento

; ----- Definir variables -----
SEGMENT data
ORG 40H
BCD_MSB DB ? ; Reservar un byte, respectivamente

```



```
BCD_MIB      DB      ?
BCD_LSB      DB      ?
BIN_MSB      DB      ?
BIN_LSB      DB      ?
NUMERO       DB      ?
VDC          DB      ?           ; Introducir variable de contado
INDICADOR    DB      ?           ; Indicador en memoria tampón
BUFFER       DB      15      DUP (?) ; 15 bytes para memoria tampón

; ----- A partir de aquí zona de códigos -----
      SEGMENT code           ; A partir de aquí segmento de código
      ORG      0000H         ; Primer comando a partir de la
                           ; dirección 0000H

      JMP      START         ; Salto al programa principal

; ----- Introducción del IV para el UART -----
      ORG      0023H         ; Posición IV para el UART (SIO)
      JMP      ISR_SIO       ; Vector de interrupción (IV)

; ----- Introducción del IV para el Timer 2 -----
      ORG      002BH         ; Posición IV para el Timer 2
      JMP      ISR_T2        ; Vector de interrupción (IV)

; *****
; *                               Programa principal                               *
; *****
START:
      MOV      NUMERO,#00H    ; Indicador para número de mediciones
      CALL     INIT_ADU       ; Inicializar convertidor A/D
      CALL     INIT_T2        ; Inicializar Timer 2
      CALL     INIT_BUFFER    ; Inicializar memoria tampón para datos
      CALL     INIT_SIO       ; Inicializar interfase serial
      SETB     EAL            ; Autorizar aceptaciones de
                           ; interrupción por CPU
      JMP      $              ; Bucle sinfín como "final"
                           ; del programa principal

; -----
;                               Inicializar convertidor A/D "on chip"
; -----
INIT_ADU:
      ANL      ADCON0,#11000000B
      ORL      ADCON0,#00001000B ; Conversión continua, entrada AN0
      MOV      ADDATL,#00H      ; start conversión
      RET

; -----
;                               Inicialización de Timer 2
;                               - cada 50 ms una interrupción -
; -----
INIT_T2:
      MOV      TH2,#HI(TCO)
      MOV      TL2,#LO(TCO)
      MOV      CRCH,#HI(TCO)
      MOV      CRCL,#LO(TCO)
      MOV      T2CON,#00010001B
      SETB     ET2
      MOV      VDC,#20         ; Inicializar variable de contado
      RET

; -----
```

```

;          Inicialización de la interfase serial
; -----
INIT_SIO:
    ORL    PCON,#10000000B    ; SMOD = 1 para 9600 baudios
    SETB   BD                  ; Conectar generador para velocidad
                                ; de transm.
    MOV    SCON,#01000000B    ; Transmisión asíncrona
                                ; con 8 bits de datos,
                                ; ¡solamente enviar!
    SETB   ES                  ; Autorizar interrupción SIO
    RET

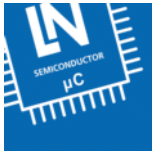
; -----
;          Inicialización de la memoria tampón de datos con cadena de
; caracteres
; -----
INIT_BUFFER:
    MOV    R0,#BUFFER
    MOV    DPTR,#STRING
INIT_NEXT:
    MOV    A,#0
    MOVC   A,@A+DPTR
    MOV    @R0,A
    JZ     INIT_END
    INC    R0
    INC    DPTR
    JMP    INIT_NEXT
INIT_END:
    RET

; ----- Cadena de caracteres -----
STRING    DB      "          mV",0DH,0AH,00H
; -----

; -----
;          ISR para Timer 2
; -----
ISR_T2:
    CLR    TF2                  ; Reposicionar bit IR
    DJNZ   VDC,IS0              ; ¿Transcurrido 1 segundo?
; Cada segundo
    MOV    VDC,#20              ; Volver a inicializar variable
                                ; de contado
    CPL    P1.7                 ; Activar LED para control
    MOV    INDICADOR,#Buffer    ; Indicador en principio de
                                ; memoria tampón
    CALL   MEDIR                 ; Preparar en memoria tampón
                                ; valor medido para transferencia serial
    SETB   TI                    ; Iniciar proceso de envío
IS0:
    RETI

; -----
;          ISR para interfase serial
; -----
ISR_SIO:
    PUSH   ACC
    JNB    TI,ISR_END            ; Si no hay interrupción de envío,
                                ; al final de ISR
; .....
    CLR    TI                    ; Borrar solicitud de envío

```



```
MOV    R0,INDICADOR
CJNE   @R0,#00H,ISR0    ; Si carácter final no alcanzado, a ISR0
MOV    INDICADOR,#Buffer ; Indicador en principio de memoria
                        ; tampón
        JMP    ISR_END    ; Al final de la ISR
ISR0:  MOV    SBUF,@R0    ; Si no, cargar memoria tampón de envío
        INC    INDICADOR  ; Indicador en siguiente posición
                        ; en memoria tampón
; .....
ISR_END:
        POP    ACC
        RETI

; -----
;                               Medir tensión en AN0, convertir
;                               y cargar en memoria tampón
; -----
MEDIR:
        MOV    A,ADDATH    ; Coger y normalizar valor medido
        MOV    B,#196
        MUL    AB          ; A partir de aquí valor binario
                        ; de 16 bits
; -----
; Convertir valor medido en formato BCD

        MOV    BIN_MSB,B
        MOV    BIN_LSB,A
        CALL   CONV_BIN_BCD
; -----
; Dividir valor medido BCD por 10
        MOV    A,BCD_LSB
        SWAP   A
        ANL    A,#00001111B
        MOV    B,A
; .....
        MOV    A,BCD_MIB
        SWAP   A
        ANL    A,#11110000B
        ADD    A,B
        MOV    BCD_LSB,A
; .....
        MOV    A,BCD_MIB
        SWAP   A
        ANL    A,#00001111B
        MOV    B,A
; .....
        MOV    A,BCD_MSB
        SWAP   A
        ANL    A,#11110000B
        ADD    A,B
        MOV    BCD_MIB,A
; -----
; Convertir valor medido BCD en caracteres ASCII
        MOV    R1,#BUFFER+5
        MOV    A,BCD_MIB
        SWAP   A
        ANL    A,#00001111B
        ADD    A,#30H
        MOV    @R1,A
; .....
        INC    R1
        MOV    A,BCD_MIB
        ANL    A,#00001111B
        ADD    A,#30H
```

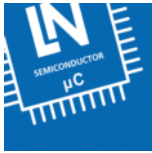
```

MOV    @R1,A
; .....
INC    R1
MOV    A,BCD_LSB
SWAP   A
ANL    A,#00001111B
ADD    A,#30H
MOV    @R1,A
; .....
INC    R1
MOV    A,BCD_LSB
ANL    A,#00001111B
ADD    A,#30H
MOV    @R1,A
; -----
; Completar memoria tampón con * y número de valores medidos enviados
MOV    R1,#Buffer
MOV    @R1,'#*'
MOV    A,NUMERO
ADD    A,#1
DA     A
MOV    NUMERO,A
; .....
INC    R1
SWAP   A
ANL    A,#00001111B
ADD    A,#30H
MOV    @R1,A
; .....
INC    R1
MOV    A,NUMERO

ANL    A,#00001111B
ADD    A,#30H
MOV    @R1,A
; .....
RET

; -----
;      Conversión de valor binario en valor BCD (comprimido)
; -----
CONV_BIN_BCD:
; Borrar zona de resultados
MOV    R1,#003H ; 3 posiciones de memoria
MOV    R0,#BCD_MSB
K4:    MOV    @R0,#000H ; Cargar con 00
INC    R0
DJNZ   R1,K4
; .....
; Conversión  Número de 16 bits en número BCD
MOV    R4,#10H ; Convertir 16 bits
K0:    MOV    R0,#BIN_LSB ; Dirección valor binario LSB
MOV    R3,#002H ; 2 bytes a convertir
CLR    C ; Borrar CF
K1:    MOV    A,@R0 ; Cargar valor binario
RLC    A ; Poner CF según rotación
MOV    @R0,A ; Recargar resultado de rotación
DEC    R0 ; Dirección valor binario MSB
DJNZ   R3,K1 ; Hacer lo mismo para MSB
; .....
MOV    R1,#BCD_LSB ; Dirección BCD destino LSB
MOV    R3,#003H ; Contador de bucles
K2:    MOV    A,@R1 ; Coger último valor destino
ADDC   A,@R1 ; Sumar CF

```



Programación de interfases de transmisión de datos CMC 4



```
DA      A           ; Corrección decimal
MOV     @R1,A       ; Archivar resultado
DEC     R1          ; Siguiente dirección BCD destino
DJNZ    R3,K2       ; ¿3ª dirección destino ya alcanzada?
DJNZ    R4,K0       ; ¿16 bits ya convertidos?
RET
```

END

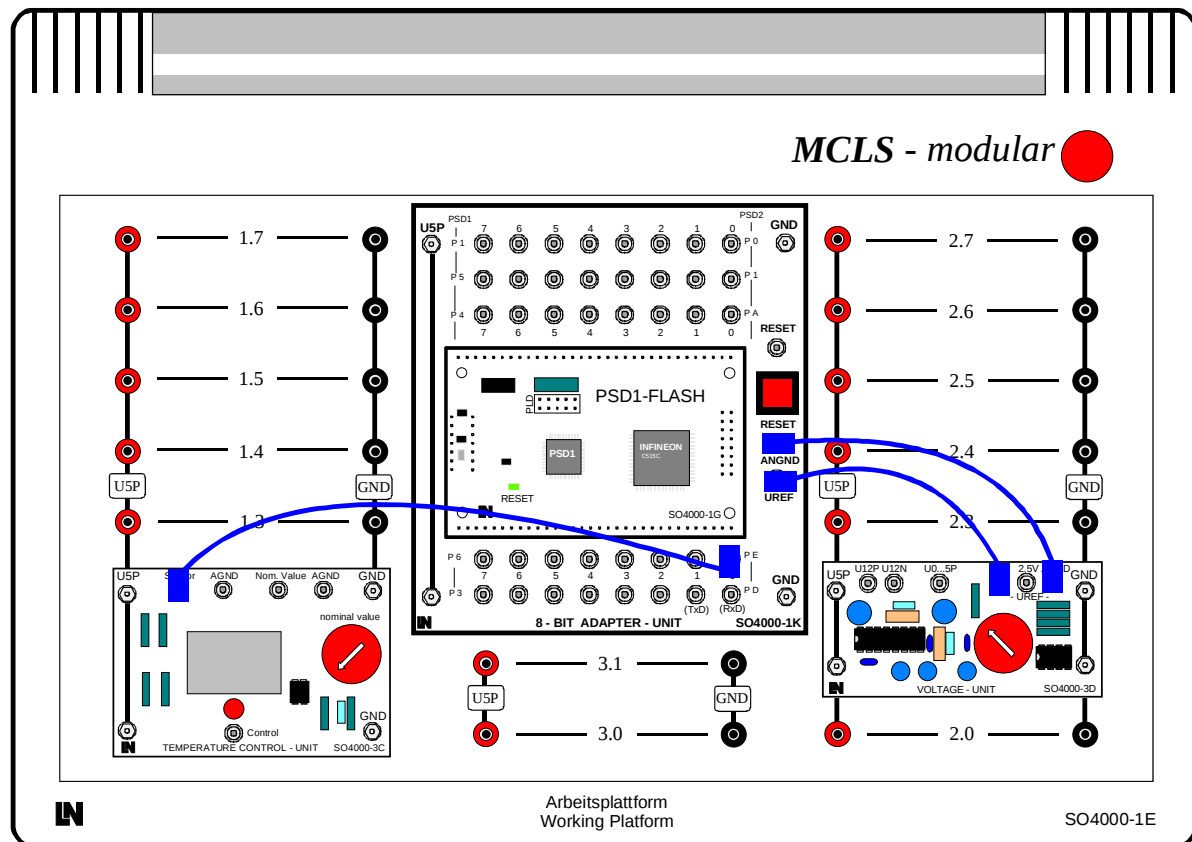


Fig. 112: Instalación de aparatos del ensayo CMC 4-1D

Ensayo CMC 4-1D

Después de una solicitud de envío del PC se deberá enviar un valor de medición de temperatura desde el microcontrolador al PC.

¿Qué conocimientos se pretenden transmitir?

- Utilización del terminal integrado para enviar un comando de solicitud
- Activación del canal de recepción en la interfase serial
- Visualización de valores medidos por medio del terminal integrado
- Evaluación de la solicitud de envío en el lado del microcontrolador
- Preparación de valores medidos binarios para la transmisión de datos seriales orientada en caracteres

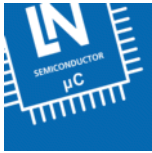
Explicaciones relativas al ejercicio:

Después de una solicitud de envío del PC, el microcontrolador deberá enviar un valor de medición de temperatura al PC.

Para la comunicación se utilizará la conexión serial estándar entre el MCLS-modular y el PC (cable serial de 9 polos LM9040).

La interfase serial se utilizará con 9600 baudios, 8 bits de datos, 1 bit de inicio y 1 bit de parada, sin bit de paridad en modo de servicio asíncrono.

Como solicitud de envío se envía el carácter „T“ del PC mediante el programa terminal al microcontrolador.



Ejercicios y soluciones

Croquice, para la instalación experimental, un diagrama de bloques sencillo.

Realice la instalación experimental según la fig. 106.

Escriba el programa ensamblador para la inicialización del convertidor A/D „on chip“ (ventana de medición 1,25 V, conversión continua, entrada AN0).

```
; -----  
; Inicializar convertidor A/D para valor de temperatura en AN0  
; porque valor medido máx. 1000 mV -> rango de medición 0-1,25 V  
; -----  
ADU_INIT_AN0:
```

Croquice la estructura de programa para la ISR de la interfase serial.

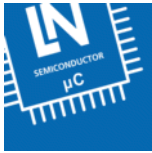
¿Qué conversiones son necesarias, partiendo del valor medido binario, para una transferencia de datos seriales orientada en caracteres?

Diseñe el texto fuente de ensamblador para la ISR sobre la base del organigrama del programa.

```
; -----  
;          ISR para interfase serial  
; -----  
ISR_SIO:
```

Ensaye la función del programa mediante el programa terminal integrado.

- ↗ Cargue e inicie el programa por medio del programa terminal.
- ↗ Observe la salida de valores medidos en la ventana terminal al pulsar la tecla de comando.



Lista de programa CMC 4-1D

```
; *****
; * Proyecto:          Solicitar valor medido, por medio de un *
; *                  comando desde el PC, al MCLS-modular      *
; * Archivo fuente:    cmc41d.asm                               *
; * Herramientas de MC: as, p2hex, terminal integrado          *
; * Perfil:            Perfil de usuario PSD1-C515C-AS mod.    *
; * Recursos de MC:    Sensor+ de la unidad de control de     *
; *                  temperatura en AN0 de la unidad PSD1      *
; *                  5V o GND de la unidad de tensión en      *
; *                  UREF y ANGND de la unidad PSD1-Flash     *
; * Editor:            HAG                                     *
; *****

; ----- Descripción de funcionamiento -----
; Como consecuencia de un comando "T" del PC se solicita un valor
; medido al MCLS-modular y se escribe en la ventana del terminal
; integrado.
; Como valor medido se utiliza la temperatura ambiente mediante
; la unidad de control de temperatura, que está conectada al canal
; análogo AN0 del convertidor A/D "on chip". El control de la
; calefacción por medio de la entrada al portal no se utiliza.
; -----
; Modo de servicio del UART "on chip": 9600 baudios, 8 bits de datos,
; 1 bit de inicio y 1 bit de parada, sin bit de paridad, solo servicio
; de envío
; -----

; #####
; - Indicaciones para el ensayo al utilizarse el terminal integrado -
; -----
; 1. Activar el terminal integrado en el perfil de proyecto
; 2. Especificar el archivo para descargar mediante-> Árbol de
;    proyecto -> Herramientas -> Terminal -> Archivo a descargar
; 3. Acordar el comando para el inicio del programa en la ventana de
;    comando mediante -> Árbol de proyecto -> Herramientas -> Terminal
;    -> Comandos (inicio del programa mediante cadena de comandos
;    'G0' = RUN)
; 4. Descargar el archivo Hex en el destino por medio del terminal
;    - Activar la ventana de mensajes del terminal
;    - Iniciar la descarga pulsando el ícono de herramienta
;    "Download"
; 5. Iniciar el programa pulsando el ícono correspondiente para
;    RUN en la ventana de mensajes
; 6. Acuerde el comando "T" para la solicitud de valores medidos
;    en la ventana de mensajes y accione el ícono asociado
; 7. Varíe la temperatura del cuerpo metálico p. ej. tocándolo
;    o conectando el portal a U5P.
; #####

; -----Acuerdos necesarios para el ensamblador -----
; CPU      80515          ; Selección de microcontrolador
; INCLUDE c515c.inc      ; Definiciones de SFR
; INCLUDE bitfuncs

; ----- Definir variables -----
; SEGMENT      data
; ORG          40H
BCD_MSB      DB      ?          ; Reservar un byte respectivamente
BCD_MIB      DB      ?
BCD_LSB      DB      ?
BIN_MSB      DB      ?
BIN_LSB      DB      ?
```

```

INDICADOR    DB    ?           ; Indicador en memoria tampón
ASCII        DB    ?           ; 2 bytes para valores medidos ASCII
              DB    ?
BUFFER       DB    15    DUP    (?) ; 15 bytes para memoria tampón

; ----- A partir de aquí zona de códigos -----
        SEGMENT code           ; A partir de aquí segmento de código
        ORG     0000H           ; Primer comando a partir de la
                                ; dirección 0000H
        JMP     START           ; Salto al programa principal

; ----- Introducción del IV para el UART -----
        ORG     0023H           ; Posición IV para el UART (SIO)
        JMP     ISR_SIO         ; Vector de interrupción (IV)

; *****
; *                               Programa principal                               *
; *****
START:
        CALL    ADU_INIT_AN0; Inicializar convertidor A/D
        CALL    INIT_BUFFER ; Inicializar memoria tampón para datos
        CALL    INIT_SIO     ; Inicializar interfase serial
        SETB    EAL           ; Autorizar aceptaciones de interrupción
                                ; por CPU
S01:    CALL    MEDIR_TEMP    ; Medir valor de temperatura
        CALL    TIEMPO       ; Retardo
        JMP     S01           ; Bucle sinfín como "final"
                                ; del programa principal

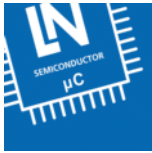
; -----
;                               Inicialización de la interfase serial
; -----
INIT_SIO:
        ORL     PCON,#10000000B ; SMOD = 1 para 9600 baudios
        SETB    BD             ; Conectar generador para velocidad
                                ; de transmisión
        MOV     SCON,#01010000B ; Transmisión asíncrona
                                ; con 8 bits de datos, enviar + recibir
        SETB    ES             ; Autorizar interrupción SIO
        RET

; -----
;                               Inicialización de la memoria tampón de datos con cadena
;                               de caracteres
; -----
INIT_BUFFER:
        MOV     R0,#BUFFER
        MOV     DPTR,#STRING
INIT_NEXT:
        MOV     A,#0
        MOVC    A,@A+DPTR
        MOV     @R0,A
        JZ      INIT_END
        INC     R0
        INC     DPTR
        JMP     INIT_NEXT
INIT_END:
        RET

; ----- Cadena de caracteres -----
STRING    DB     "Temp.: xx ",0F8H,"C",0DH,0AH,00H

; -----
;                               ISR para interfase serial
; -----

```



```
ISR_SIO:
    PUSH    ACC
    JNB     TI,ISR_EMPF ; si no hay interrupción de envío,
                        ; al final de la ISR
; Tratamiento de la interrupción de envío.....
    CLR     TI          ; Borrar solicitud de envío
    MOV     R0,INDICADOR
    CJNE    @R0,#00H,ISR0 ; Si carácter final no alcanzado, a ISR0
    MOV     INDICADOR,#BUFFER ; Indicador en principio de memoria
                        ; tampón
    JMP     ISR_END      ; Al final de la ISR
ISR0: MOV    SBUF,@R0    ; Si no, cargar memoria tampón de envío
    INC     INDICADOR    ; Indicador en siguiente posición en
                        ; la memoria tampón
; Final de la ISR.....
ISR_END:
    POP     ACC
    RETI
; Tratamiento de la interrupción de recepción.....
ISR_EMPF:
    CLR     RI          ; Borrar solicitud de recepción
    MOV     A,SBUF      ; Sacar carácter de memoria tampón
                        ; de recepción
    CJNE    A,'#T',ISR_END ; ¿Comando acordado?
    SETB    TI          ; De ser así, poner solicitud de envío
    JMP     ISR_END

; -----
; Inicializar convertidor A/D para valor de temperatura en AN0
; porque valor medido máx. 1000 mV -> rango de medición 0 - 1,25 V
; -----
ADU_INIT_AN0:
    ANL     ADCON0,#11000000B
    ORL     ADCON0,#00001000B ; Conversión continua, canal AN0
    MOV     ADDATL,#00h      ; Start de medición 0 - 1,25V
    RET

; -----
; Medir la temperatura
; Toma, normalización y conversión BCD del valor medido
; -----
MEDIR_TEMP:
    MOV     B,#49          ; Factor para normalización
                        ; porque aprox. 4,9 mV/Bit
AD1: JB     BSY,AD1        ; Esperar hasta que finalice
                        ; la conversión
    CALL    SCAL_TEMP      ; scale to virtual measurment window
                        ; 0...1.25V
    MUL     AB

; .....
    MOV     BIN_MSB,B      ; Preparar valor medido de 16 bits
    MOV     BIN_LSB,A
    CALL    CONV_BIN_BCD   ; Conversión binaria en BCD
; .....
; Dividir valor por 10
    MOV     A,BCD_LSB
    SWAP    A
    ANL     A,#00001111B
    MOV     B,A
    MOV     A,BCD_MIB
    SWAP    A
    ANL     A,#11110000B
    ADD     A,B
    MOV     BCD_LSB,A      ; BCD_LSB listo
    MOV     A,BCD_MIB
```

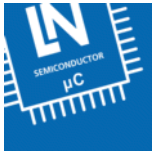
```

        SWAP    A
        ANL     A,#00001111B
        MOV     B,A
        MOV     A,BCD_MSB
        SWAP    A
        ANL     A,#11110000B
        ADD     A,B
        MOV     BCD_MIB,A          ; BCD_MIB listo
; Convertir valor medido BCD en caracteres ASCII.....
        MOV     R1,#BUFFER+7
        MOV     A,BCD_MIB
        ANL     A,#00001111B
        ADD     A,#30H
        MOV     @R1,A
        INC     R1
        MOV     A,BCD_LSB
        SWAP    A
        ANL     A,#00001111B
        ADD     A,#30H
        MOV     @R1,A
        RET

; -----
;           Scaling to virtual measurement window 0...1.25V
;           Inputs: 10-Bit value in ADDATH and ADDATL
;           Output: scaled 8-Bit value in A
; -----
SCAL_TEMP:
        PUSH    B
; shift 8-Bit value in ADDATH two bits to the left
        MOV     A,ADDATH
        RL      A
        RL      A
        ANL     A,#11111100b
        MOV     B,A
; shift 8-Bit value in ADDATL two bits to the left
; and add to the scaled value in ADDATH
        MOV     A,ADDATL
        RL      A
        RL      A
        ANL     A,#00000011b
        ADD     A,B          ; scaled value in A
        POP     B
        RET

; -----
;           Conversión de valor binario en valor BCD (comprimido)
; -----
CONV_BIN_BCD:
; Borrar zona de resultados
        MOV     R1,#003H          ; 3 posiciones de memoria
        MOV     R0,#BCD_MSB
K4:     MOV     @R0,#000H          ; Cargar con 00
        INC     R0
        DJNZ    R1,K4
; Conversión Número binario de 16 bits en número BCD.....
        MOV     R4,#10H          ; Convertir 16 bits
K0:     MOV     R0,#BIN_LSB        ; Dirección valor binario LSB
        MOV     R3,#002H          ; 2 bytes a convertir
        CLR     C                ; Borrar CF
K1:     MOV     A,@R0              ; Cargar valor binario
        RLC     A                ; Poner CF según rotación
        MOV     @R0,A            ; Recargar resultado de rotación
        DEC     R0              ; Dirección valor binario MSB
        DJNZ    R3,K1            ; Hacer lo mismo para MSB

```



```
; .....
      MOV    R1,#BCD_LSB      ; Dirección destino BCD LSB
      MOV    R3,#003H        ; Contador de bucles
K2:    MOV    A,@R1           ; Coger último valor destino
      ADDC   A,@R1           ; Sumar CF
      DA     A                ; Corrección decimal
      MOV    @R1,A           ; Archivar resultado
      DEC    R1              ; Siguiendo dirección destino BCD
      DJNZ   R3,K2           ; ¿3ª dirección destino ya alcanzada?
      DJNZ   R4,K0           ; ¿16 bits ya convertidos?
      RET

; -----
;                               Retardo aproximadamente 1 s
; -----
TIEMPO:
      MOV    R5,#20
T03:   CALL  TIEMPO_50
      DJNZ   R5,T03
      RET

; -----
;                               Retardo aproximadamente 50ms
; -----
TIEMPO_50:
      MOV    R7,#98
Z01:   MOV    R6,#000H
Z00:   DJNZ   R6,Z00
      DJNZ   R7,Z01
      RET

      END
```

CMC 4-2 Comunicación serial entre 2 terminales de datos por medio de una interfase V.24

Por medio de una interfase V.24 se presenta la comunicación serial entre 2 *MCLS-modular*. Ambos *MCLS-modular* están conectados entre ellos, como terminales de datos, mediante una conexión de „null modem“.

Utilización del adaptador de V24

Por medio del *MCLS-modular*[®] y el adaptador de V24 se pueden imitar partes de una interfase RS232 según CCITT V.24. El adaptador de V24 pone a disposición gran cantidad de las señales MODEM estandarizadas que permiten numerosos experimentos. Mediante el software son posibles diferentes variantes de la comunicación serial. Puesto que están disponibles diferentes circuitos de control y de estado en el adaptador, también se puede realizar una comunicación con una sincronización inicial de hardware.

La fig. 201 muestra la conexión de dos *MCLS-modular*[®] por medio de adaptadores V24. Los *MCLS-modular*[®] están conectados entre ellos como terminales de datos por medio de una unión null-modem.

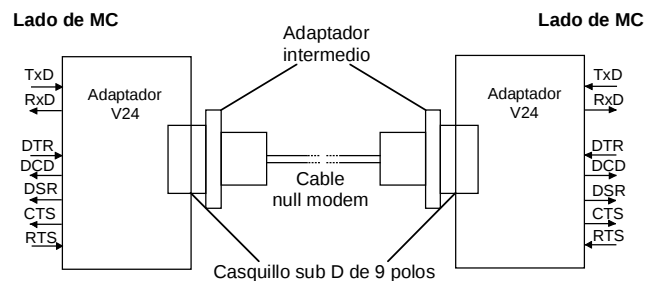


Fig. 201: Conexión de 2 *MCLS-modular* por medio de adaptadores V24

Para la realización de esta unión se emplea un cable de modem. Adicionalmente son necesarios adaptadores intermedios en cada conector Sub-D del adaptador V24. De ello resultan los niveles de ejemplo representados en la fig. 202 en los circuitos de datos, de control y de estado.

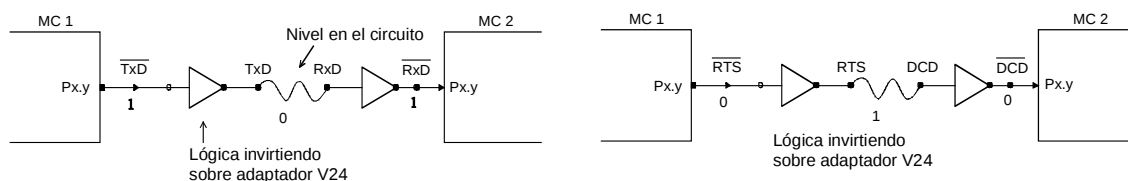


Fig. 202: Niveles lógicos en circuitos de datos o control y de estado (ejemplo)

A las correspondientes conexiones del adaptador de V24 se pueden conectar de forma óptima cables de medición, p. ej. para la observación del curso de las señales de interfase.

Esto permite controlar de forma sencilla la transmisión asíncrona, orientada en caracteres, de la información serial (en modo „Single-Shot“ de un osciloscopio).

En la transmisión serial de grandes cantidades de datos es conveniente la utilización de una sincronización inicial del hardware, porque en otro caso podrían perderse datos en el lado de recepción. Para la realización de una sincronización inicial existe, p. ej., la posibilidad de vigilar una memoria tampón de recepción con respecto a su nivel de relleno. Al alcanzarse un nivel de relleno a definir de la memoria tampón, el terminal receptor señaliza este estado por medio de RTS=0. En el lado de envío, DCD es igual a 0 y se interrumpe la transmisión. Después de un vaciado de la memoria tampón, RTS en el lado de recepción es igual a 1 y se puede continuar con la transmisión de datos. Este principio se realiza de esta forma en el ejemplo de programa cmc42B.

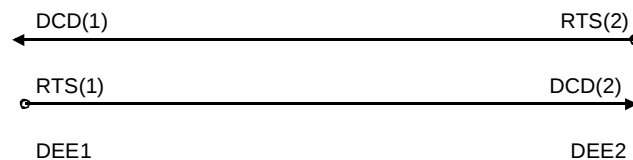


Fig. 203: Sincr. inicial del hardware por medio de RTS y DCD entre 2 terminales de datos (TERM)

El UART de software

La comunicación entre los terminales de datos se realiza por medio de un UART de software con una velocidad de transmisión de datos con aproximadamente 300 baudios, con 1 bit de inicio, 8 bits de datos, 1 bit de parada y sin bit de paridad. La muy baja velocidad de transmisión de datos se seleccionó con el fin de observar la sincronización inicial de hardware por medio de los cambios de nivel en RTS o DCD en los terminales de datos. Con un tamaño de memoria tampón de recepción acordado de 20 caracteres, un límite de memoria tampón vigilado de 16 caracteres y una longitud del texto a transmitir de >160 caracteres, este proceso se puede observar muy bien.

Otra ventaja reside en que la interfase serial integrada del microcontrolador está disponible sin restricciones para la depuración. Con ello, el desarrollo lógico de la sincronización inicial de hardware se puede comprender de manera muy sencilla.

La realización técnica de programa del UART de software se desprende de los textos fuente y no presenta particularidades. En caso de necesidad se puede realizar una adaptación a velocidades de transmisión más altas.

Aparatos y accesorios necesarios

Unid	Denominación	Nº Id.
2	Plataforma con mód. de alim. +5V	SO4000-1E
2	Fuente de alim. de enchufe AC 90...230V 45..65Hz, DC 9V 630mA	SO4000-1F
2	Módulo PSD1-Flash Controlador C515C	SO4000-1G
2	Unidad de adaptador de 8 bits	SO4000-1K
2	Unidad de LED	SO4000-1P
2	Unidad de display 2 (LCD)	SO4000-1B
2	Adaptador BNC	SO4000-1T
2	Unidad de interruptores	SO4000-1R
2	Unidad de teclas	SO4000-1Q
2	Adaptador de V24	SO4000-3J
1	Software IDE para MCLS (D,GB,F,E)	SO4001-9H
2	Cable de interfase serial 9/9 polos	LM9040
1	Kit de cable null-modem	LM9045
20	Cable de medición 2mm 15cm azul	SO5126-5K
20	Cable medición 2mm 15cm amarillo	SO5126-5M
1	CMC 4 Programación de interfaces de transmisión de datos	SH5019-1D

Instalación general de aparatos

¡Separe la alimentación de corriente del MCLS-modular®!

Conecte la plataforma de trabajo por medio del cable de interfase serial a su PC.

Encaje la unidad de adaptador de 8 bits con el Módulo PSD1-Flash adaptado en ésta en el campo de casquillos 4.0 - 4.5 en el centro de la plataforma.

Seleccione los módulos de experimentación necesarios del ejercicio y conéctelos a los puertos elegidos por usted.

Conecte la alimentación de corriente al MCLS-modular.

Nota relativa a la seguridad:

¡Asegúrese de conectar correctamente los potenciales de tensión (izda U5P, dcha GND)! No se deben aplicar tensiones extrañas a los puntos de medición llevados hacia fuera ¡ya que esto podría conducir a la destrucción de los componentes de las placas!

Prepare, para la realización del ensayo, el manual de instrucciones del entorno de desarrollo integrado (IDE) y el manual de instrucciones de la unidad PSD1-Flash. Utilice la información compacta en el anexo del manual de instrucciones del módulo PSD1-Flash.

Infórmase en los manuales de instrucciones de los módulos de experimentación que se van a emplear.

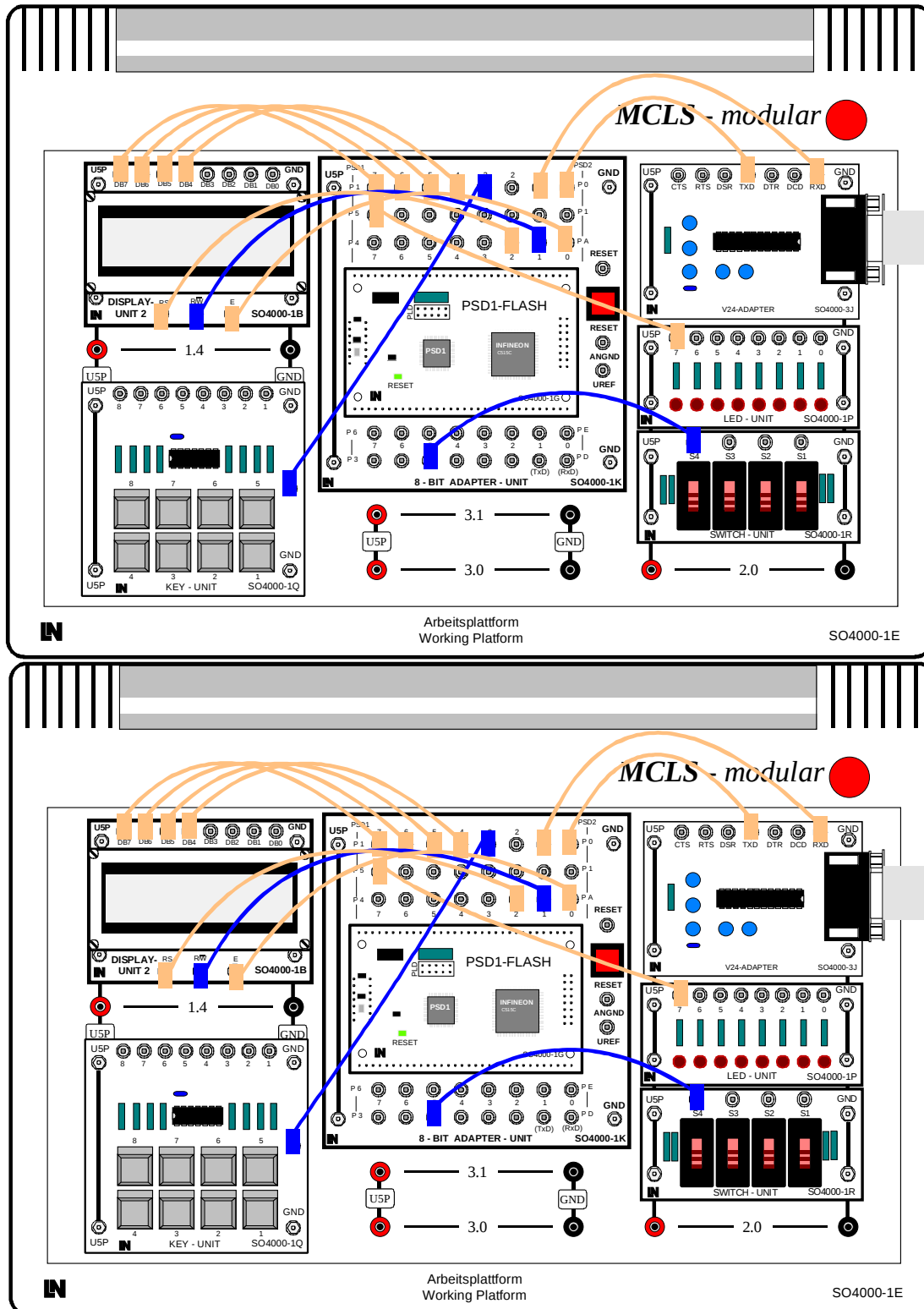


Fig.: 204: Instalación de aparatos del ensayo CMC 4-2A

Ensayo CMC 4-2A

Entre 2 *MCLS-modular* se debe realizar una transferencia de datos seriales sin sincronización inicial de hardware mediante los adaptadores de V24. La cadena de caracteres recibidos se deberá visualizar en el indicador LCD.

¿Qué conocimientos se pretenden transmitir?

- Utilización del adaptador de V24 para la conexión de 2 *MCLS-modular*[®]
- Utilización de las señales V24 RXD y TXD
- Utilización de un UART de software
- Utilización de una memoria tampón de recepción
- Visualizar el contenido de la memoria tampón en el indicador LCD

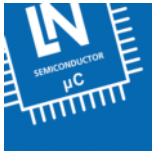
Explicaciones relativas al ejercicio:

Ambos *MCLS-modular*[®] se deben conectar entre ellos, por medio de un cable null-modem, como terminales de datos mediante adaptador V24. Como cable de conexión se utiliza un cable de modem y un adaptador intermedio, respectivamente, en cada extremo del cable. Un interruptor permitirá la definición de los *MCLS-modular*[®] como terminal de datos 1 (TERM1) y terminal de datos 2 (TERM2). El comienzo de la transmisión de datos se deberá iniciar pulsando una tecla.

Ejercicios y soluciones

Familiarícese con el adaptador V24 (ver manual de instrucciones). ¿Qué niveles de señal se utilizan para los circuitos de datos o para los circuitos de control y de estado?

Croquice la unión de los terminales de datos TERM1 y TERM2 por medio de las señales RXD y TXD.



¿Cómo se ha realizado la velocidad de transmisión del UART de software?

¿Qué tiempo se necesita para transmitir un carácter completo incluyendo bit de inicio y bit de parada con 300 baudios?

Croquice, para la instalación experimental, un diagrama de bloques sencillo. Realice la instalación experimental según la fig. 204.

Configure, por medio de la posición del interruptor en P3.5, TERM1 Y TERM2. Ensaye la función del programa cmc42A en tiempo real.

- ➡ Observe el proceso de recepción mediante la visualización en el display de LCD del respectivo terminal de datos receptor.

¿Qué tiempo se necesita para transmitir la cadena de caracteres completa en TEXT0 y TEXT2 (ver texto fuente de ensamblador cmc42a.asm) con 300 baudios?

Lista de programa CMC 4-2A

```

; *****
; * Proyecto:          Intercambio de datos sin sincronización inicial de      *
; *                   hardware por medio de la interfase V.24 con UART de SW*
; * Archivo fuente:    cmc42a.asm                                           *
; * Herramientas de MC: as, p2hex, depurador                                *
; * Perfil:            Perfil integrado PSD1-C515C-AS                        *
; * Recursos de MC:    P1.0 como TXD y P1.1 como RXD (UART de SW)          *
; *                   en el adaptador de V24,                               *
; *                   unidad LCD en P1.7-P1.4 y P4.0-P4.2                   *
; *                   P1.3 (int. ext. 6) para interrupción de teclas,       *
; *                   interruptor de la unidad de interruptores en P3.5,    *
; *                   en P5.7 opcionalmente un LED de la unidad de LED       *
; *                   para controlar la pulsación de teclas                 *
; * Editor:            HAG                                                  *
; *****

; ----- Descripción de funcionamiento -----
; 2 MCLS-modular deben comunicar entre ellos, por medio de una conexión null-
; modem, como equipos terminales de datos (TERM). A ese fin, un adaptador
; V24 (incluyendo adaptadores intermedios) suministra en cada extremo las
; señales de interfase V24 necesarias (RXD, TXD).
; La conexión de los dos TERM se realiza por medio de un cable de modem.
; Para enviar o para recibir las señales seriales se realiza
; la función UART por medio del software.
; Con ello, el UART "on chip" queda libre para funciones de depuración.
; -----
; El proceso de envío y recepción serial asíncrono para una cadena
; de caracteres se realiza sin protocolo de hardware o software, efectuándose
; la transmisión asíncrona orientada en caracteres por medio de un
; UART realizado por:
; 300 baudios, 1 bit de inicio, 8 bits de datos, 1 bit de parada, sin paridad
; -----
; En ambos MCLS-modular se carga y se inicia el software cmc42A.
; Mediante un conmutador deslizante en P3.5 se define el MCLS-modular
; que hará de TERM1 y de TERM2 para la comunicación.
; Pulsando una tecla (unidad de TECLAS) se puede iniciar opcionalmente
; un proceso de envío de TERM1 o TERM2. La cadena de caracteres
; que llega al correspondiente lado de recepción se introduce
; en una memoria tampón de recepción y se visualiza durante un tiempo corto
; (aproximadamente 3s) en el indicador de LCD.
; -----

; -----Acuerdos necesarios para el ensamblador -----

        CPU      80515                ; Selección de microcontrolador
        INCLUDE  c515c.inc            ; Definiciones de SFR

; Asociación de pins de puerto para del UART de SW
TX      EQU     P1.0                  ; Circuito de envío
RX      EQU     P1.1                  ; Circuito de recepción

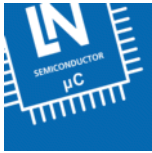
        SEGMENT      DATA
INDICADOR DB ?
TCO       DB ?

        SEGMENT      IDATA
REC_BUFFER DB 24      DUP      (?)

        SEGMENT      code             ; A partir de aquí segmento de código
ORG         0000H                    ; Primer comando a partir de la dirección 0000H
using      bank0                      ; Selección RB0

; -----
        JMP      START                ; Salto al programa principal

```



Programación de interfaces de transmisión de datos CMC 4



```
; ----- Introducción del IV para la int. ext. 6 -----
      ORG    006BH          ; Posición IV para la interrupción ext. 6
      JMP    ISR_INT6       ; Vector de interrupción (IV)

      ORG    100H

; *****
; *                               Programa principal                               *
; *****
START:
      MOV     SP,#40H
      CALL    INIT_INT6     ; Inicializar interrupción ext. 6
      CALL    INIT_SW_UART  ; Inicializar UART de SW
      CALL    INIT_LCD      ; Inicializar y borrar indicador LCD
      SETB    EAL           ; Autorizar aceptaciones de interrupción por CPU
      MOV     DPTR,#TEXT0   ; Direccional texto para inicio
      CALL    OUT_TEXT      ; Texto en indicador LCD
S00:   CALL    RECEIVE_STRING ; Esperar cadenas de datos seriales
      ; y, dado el caso, introducirlos y visualizarlos
      JMP     S00           ; Bucle sinfín como "final"
      ; del programa principal

; -----
;                               Inicialización de la interrupción ext. 6
;                               - Disparo por flanco 0-1 -
; -----
INIT_INT6:
      SETB    EX6          ; Autorizar interrupción externa 6
      RET

; -----
;                               Inicialización del UART de software
; -----
INIT_SW_UART:
      SETB    TX
      SETB    RX
      RET

; ----- Funciones UART de SW -----
; Imitación de la funcionalidad de UART para:
; 1 bit de inicio, 8 bits de datos, 1 bit de parada, sin bit de paridad
; 300 baudios
; -----
; Introducir 1 byte por medio de RXD a A
RECEIVE_BYTE:
      PUSH    AR7
      MOV     R7,#8        ; Contador de bits = 8
REC1:   JB     RX,REC1      ; Esperar hasta RXD=0
      CALL    WAIT         ; Esperar tiempo de bit
REC2:   CALL    WAIT
;
      MOV     CY,RX        ; Introducir bit
      RRC     A            ; Desplazar a acumulador
      DJNZ    R7,REC2      ; Bucle hasta contador de bit =0
      CALL    WAIT
      POP     AR7
      RET

; Enviar 1 byte en A por medio de TXD
TRANSMIT_BYTE:
      PUSH    AR7
      MOV     R7,#8        ; Contador de bits = 8
      CLR     TX           ; Bit de inicio
      CALL    WAIT         ; Esperar tiempo de bit
TRAN1:  RRC     A            ; Llevar bit a CY
      MOV     TX,CY        ; Sacar información de bit
      CALL    WAIT         ; Esperar tiempo de bit del bit de datos
```

```

        DJNZ R7,TRAN1          ; Bucle hasta contador de bits =0
        SETB TX
        CALL WAIT              ; Tiempo opcional para bit de parada
        CALL WAIT
        POP AR7
        RET

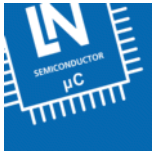
; Tiempo de bit para 300 baudios aproximadamente 3,333 ms (((71*2+2)*23)+10)us
WAIT: PUSH AR7
      PUSH AR6
      MOV R6,#23
WA1:  MOV R7,#71
WA2:  DJNZ R7,WA2
      DJNZ R6,WA1
      POP AR6
      POP AR7
      RET

; -----
;                               ISR para interrupción externa 6
;                               - Iniciar proceso de envío -
; -----
ISR_INT6:
      PUSH ACC
      PUSH DPH
      PUSH DPL
      JB P3.5,IS0              ; Si nivel 1, TERM2
; .....
      MOV DPTR,#Text1
IS1:  CPL P5.7                  ; Con fines de ensayo
      CALL SEND_STRING
; .....
      CALL TIEMPO              ; Tiempo para eliminación de rebotamiento de
tecla
      CALL TIEMPO
      CLR IEX6                  ; Reposicionar bit de solicitud de interrupción
      POP DPL
      POP DPH
      POP ACC
      RETI
; -----
IS0:  MOV DPTR,#Text2
      JMP IS1

; -----
;                               Enviar una cadena de caracteres con UART de SW
; -----
SEND_STRING:
      PUSH ACC
      PUSH DPH
      PUSH DPL
SE01: MOVX A,@DPTR
      JZ END_S                  ; ¿Final?
      CALL TRANSMIT_BYTE
      CALL TIEMPO              ; Tiempo de pausa
      CALL WAIT
      INC DPTR                  ; Contador +1
      JMP SE01
END_S: CALL TRANSMIT_BYTE      ; ¡Queda por enviar 0!
      POP DPL
      POP DPH
      POP ACC
      RET

; -----
;                               Recibir una cadena de caracteres con UART de SW
;                               Visualizar los caracteres en la unidad de LCD
; -----

```



```
; -----
RECEIVE_STRING:
    PUSH    ARO
    PUSH    ACC
    CALL    CLR_BUFFER      ; Borrar memoria tampón de recepción
    MOV     INDICADOR,#REC_BUFFER
RE0:  MOV     R0,INDICADOR
    CALL    RECEIVE_BYTE    ; Introducir byte
    CLR     CY
    SUBB    A,#00           ; ¿Carácter final?
    JZ      END_R           ; De ser así, al final del subprograma
; .....
    MOV     @R0,A
    INC     INDICADOR
    JMP     RE0
END_R:
    CALL    OUT_BUFFER      ; Memoria tampón de recepción en indicador
    CALL    TIEMPO_LARGO    ; aproximadamente 3s de texto de recepción
                                ; en indicador
    CALL    TIEMPO_LARGO
    CALL    TIEMPO_LARGO
    MOV     DPTR,#TEXT0     ; Visualizar disposición de recepción
    CALL    OUT_TEXT
    POP     ACC
    POP     ARO
    RET

; -----
;   Visualización de texto de 2 líneas en la unidad de LCD
;   - Cadena de caracteres tiene 0 como indicativo de final -
; -----
OUT_TEXT:
    PUSH    ACC
    MOV     A,#00H          ; Cursor en primera línea
    CALL    CURSOR_POS
    MOV     TCO,#000H       ; Contador de caracteres y offset
OUT_Z1:
    MOV     A,TCO
    MOVC    A,@A+DPTR       ; Coger carácter
    JZ      OUT_END         ; ¿Indicativo de final?
    CALL    WRITE_DREG      ; Carácter para visualizar
    CALL    TIEMPO_CORTO
    INC     TCO              ; Contador de caracteres +1
    MOV     A,TCO
    CLR     CY
    SUBB    A,#10H          ; ¿Aún línea 1?
    JC      OUT_Z1          ; De ser así, a OUT_Z1
; .....
    MOV     A,#40H          ; Cursor
    CALL    CURSOR_POS      ; al principio de la 2ª línea
; .....
OUT_Z2:
    MOV     A,TCO
    MOVC    A,@A+DPTR       ; Coger carácter
    JZ      OUT_END         ; ¿Indicativo de final?
    CALL    WRITE_DREG      ; Carácter para visualizar
    CALL    TIEMPO_CORTO
    INC     TCO              ; Contador de caracteres +1
    JMP     OUT_Z2
; .....
OUT_END:
    POP     ACC
    RET

; -----
;   Borrar memoria tampón de recepción
; -----
CLR_BUFFER:
    PUSH    ACC
```

```

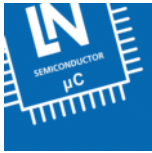
        PUSH  AR0
        MOV   A,#000
        MOV   TCO,#20H
        MOV   R0,#REC_BUFFER
CLR0:   MOV   @R0,A
        INC   R0
        DJNZ  TCO,CLR0
        POP   AR0
        POP   ACC
        RET

; -----
;   Escribir memoria tampón de recepción en 2 líneas en el indicador
; -----
OUT_BUFFER:
        PUSH  ACC
        PUSH  AR0
        MOV   A,#00H           ; Cursor en primera línea
        CALL  CURSOR_POS
        MOV   INDICADOR,#REC_BUFFER
        MOV   TCO,#000H       ; Reposicionar contador de caracteres
OUT_P_Z1:
        MOV   R0,INDICADOR     ; Indicador en memoria tampón de recepción
        MOV   A,@R0           ; Coger carácter de memoria tampón
        JZ    OUT_P_END       ; ¿Indicativo de final de la cadena
                                ; de caracteres?
; .....
        CALL  WRITE_DREG
        CALL  TIEMPO_CORTO
        INC   TCO              ; Contador de caracteres +1
        INC   INDICADOR       ; Indicador +1
        MOV   A,TCO
        CLR   CY
        SUBB  A,#10H          ; ¿Línea 2? Si no, a OUT_Z1
        JC    OUT_P_Z1
; .....
        MOV   A,#40H          ; Cursor en segunda línea
        CALL  CURSOR_POS
; .....
OUT_P_Z2:
        MOV   R0,INDICADOR
        MOV   A,@R0
        JZ    OUT_P_END
        CALL  WRITE_DREG
        CALL  TIEMPO_CORTO
        INC   INDICADOR
        JMP   OUT_P_Z2
; .....
OUT_P_END:
        POP   AR0
        POP   ACC
        RET

; -----
;   Tiempo corto, aproximadamente 1ms
; -----
TIEMPO_CORTO:
        PUSH  AR7
        PUSH  AR6
        MOV   R7,#2
T04:   MOV   R6,#000H
T03:   DJNZ  R6,T03
        DJNZ  R7,T04
        POP   AR6
        POP   AR7
        RET

; -----
;   Tiempo de eliminación de rebotamiento aproximadamente 50ms
; -----

```



```
; -----
TIEMPO:
    PUSH  AR7
    PUSH  AR6
    MOV    R7,#98
T01:  MOV    R6,#000H
T00:  DJNZ   R6,T00
    DJNZ   R7,T01
    POP    AR6
    POP    AR7
    RET

; -----
;                               Tiempo largo - aproximadamente 1s
; -----
TIEMPO_LARGO:
    PUSH  AR5
    MOV    R5,#20
T02:  CALL  TIEMPO
    DJNZ   R5,T02
    POP    AR5
    RET

; -----
;                               Cadena de caracteres para aviso de disposición
; -----
TEXT0:
    DB     "Ready for          Receive!          ", 0
; -----

; -----
;                               Cadena de caracteres para la transferencia serial a TERM2
; -----
TEXT1:
    DB     "Mail by          MCLS-modular 1!", 0
; -----

; -----
;                               Cadena de caracteres para la transferencia serial a TERM1
; -----
TEXT2:
    DB     "Mail by          MCLS-modular 2!", 0
; -----

; -----
;                               Insertar funciones para unidad de LCD
; -----
    INCLUDE FCT_LCD.INC      ; Archivo de inclusión
                             ; en el mismo directorio
; -----

END
```

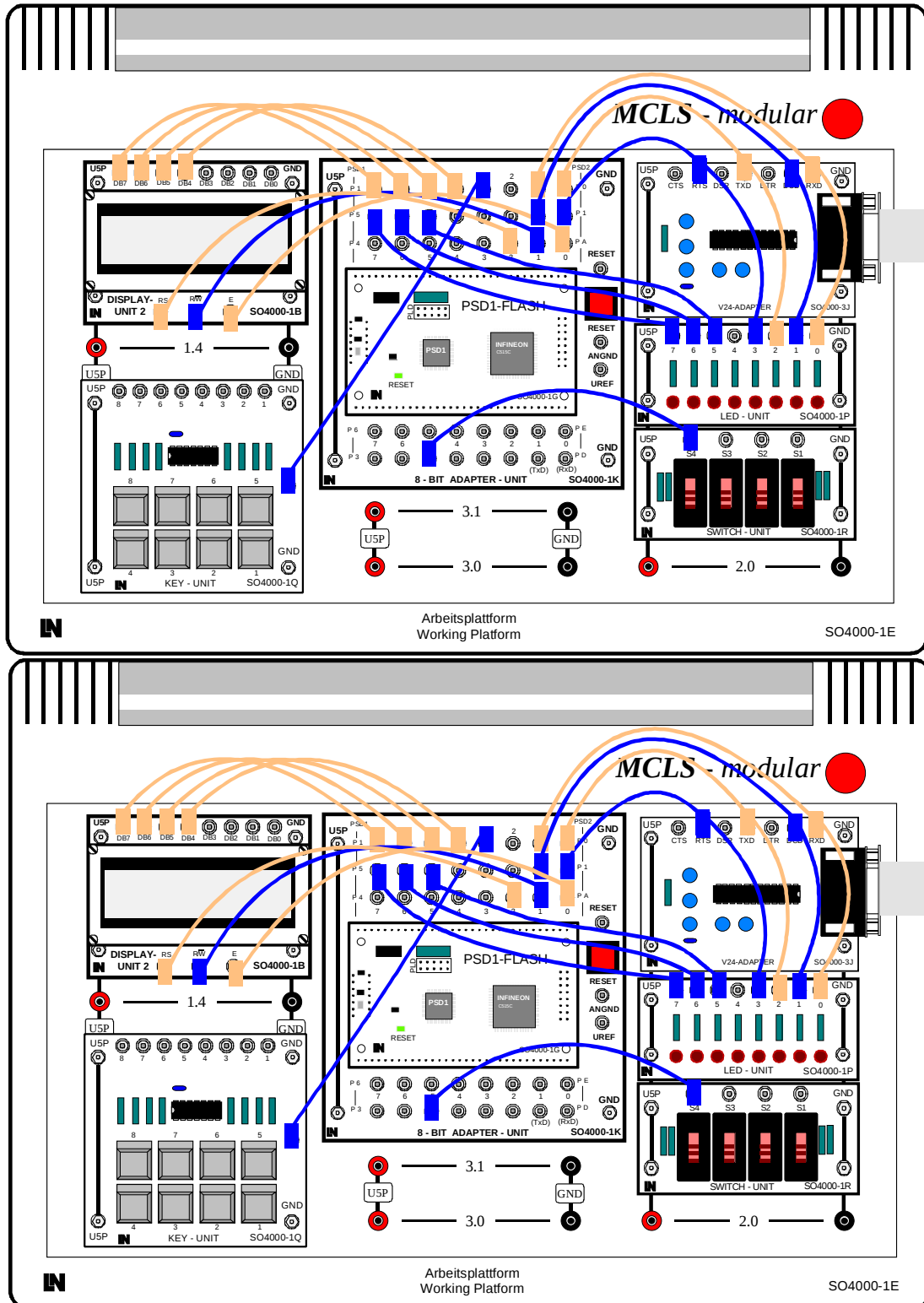
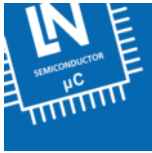


Fig. 205: Instalación de aparatos del ensayo CMC 4-2B



Ensayo CMC 4-2B

Entre 2 *MCLS-modular* se debe realizar una transferencia de datos seriales con sincronización inicial de hardware por medio de los adaptadores V24. La cadena de caracteres recibidos se deben visualizar en el indicador de LCD. Adicionalmente debe ser posible observar el juego de señales de RTS o DCD por medio de LEDs.

¿Qué conocimientos se pretenden transmitir?

- Utilización de las señales RTS y DCD para la sincronización inicial de hardware
- Utilización de una memoria tampón de recepción con control de un límite para la interrupción de la transmisión
- Visualización del contenido de la memoria tampón de recepción por medio del indicador de LCD

Explicaciones relativas al ejercicio:

Ambos *MCLS-modular*[®] se deben conectar, por medio de una conexión de null-modem, como aparatos terminales de datos (TERM) mediante adaptador V24. Como cable de conexión se utiliza un cable de modem y un adaptador intermedio, respectivamente, en cada extremo del cable. Por medio de un interruptor se debe definir el *MCLS-modular*[®] que hace de TERM1 y el que hace de TERM2. El comienzo de la transmisión de datos se iniciará pulsando una tecla.

Para la realización de la sincronización inicial se vigila el nivel de relleno de una memoria tampón de recepción. Al alcanzarse un nivel de relleno de la memoria tampón a definir, el TERM receptor señala este estado por medio de RTS=0. En el lado emisor, DCD también es igual a 0 y la transmisión se interrumpe. Una vez vaciada la memoria tampón, RTS en el lado receptor es igual a 1 y se puede continuar con la transmisión de datos.

Ejercicios y soluciones

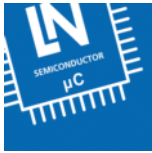
***Familiarícese con el texto fuente de ensamblador cmc42b.asm.
Deduzca de ello el organigrama para el programa principal START.***

Croquice la unión de los terminales de datos TERM1 y TERM2 por medio de las señales RXD, TXD, RTS y DCD.

Croquice, para la instalación experimental, un diagrama de bloques sencillo. Realice la instalación experimental según la fig. 205.

Configure, por medio de la posición del interruptor en P3.5, TERM1 y TERM2. Ensaye la función del programa cmc42b.asm en tiempo real y observe los indicadores de LED para RTS y DCD.

👉 Observe los comentarios en el texto fuente.



Croquice el organigrama de programa para la recepción de una cadena de caracteres según el subprograma RECEIVE_STRING en el texto fuente.

Aumente la velocidad de transmisión de datos mediante una correspondiente adaptación del UART de software.

- ➡ Proceda según los comentarios detallados realizados en el subprograma WAIT.
Seleccione los valores predeterminados para las constantes a cargar en R6 o R7.

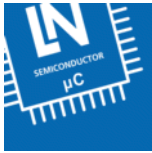
Lista de programa CMC 4-2B

```

; *****
; * Proyecto:          Intercambio de datos con sincronización      *
; *                  inicial de hardware por medio de la          *
; *                  interfase V.24                                *
; * Archivo fuente:    cmc42b.asm                                    *
; * Herramientas de MC: as, p2hex, depurador                        *
; * Perfil:            Perfil integrado PSD1-C515C-AS              *
; * Recursos de MC:    P1.0 como TXD y P1.1 como RXD (UART de SW) *
; *                  en el adaptador de V24,                      *
; *                  P5.0 en RTS adaptador de V24,                *
; *                  P5.1 en DCD adaptador de V24,                *
; *                  unidad de LCD en P1.7-P1.4 y P4.0-P4.2        *
; *                  P1.3 (int. ext. 6) para interrupción de      *
; *                  tecla,interruptor de la unidad de            *
; *                  interruptores en P3.5 P5.6, P5.5, P5.1 y P5.0 *
; *                  para estado de la conexión en la unidad de   *
; *                  LED, en P5.7 opcionalmente un LED de la      *
; *                  unidad de LED para el control del           *
; *                  accionamiento de teclas                      *
; * Editor:           HAG                                          *
; *****

; ----- Descripción de funcionamiento -----
; 2 MCLS-modular comunican por medio de una conexión de null-modem
; como terminales de datos (TERM) con sincronización inicial de
; hardware. A ese fin, un adaptador de V24 (incluyendo adaptadores
; intermedios) en cada extremo suministra las señales de interfase V24
; necesarias (RXD, TXD,RTS, DCD). La conexión de los dos TERM se
; realiza por medio de un cable de modem. Para enviar y recibir las
; señales seriales se realiza la función de UART por medio del
; software. Con ello el UART "on chip" queda libre para funciones de
; depuración.
; -----
; El proceso de envío y recepción serial asíncrono se realiza en
; este ejemplo de programa por medio de una sincronización inicial de
; hardware. En el lado receptor se controla una memoria tampón de
; recepción con respecto a si se alcanza un límite superior. Al
; alcanzarse este valor límite,se pone RTS=0 y con ello se controla
; DCD en el lado emisor.
; El proceso de envío se interrumpe, hasta que RTS en el lado receptor
; vuelve a ser igual a 1, lo que se consigue vaciando la
; memoria tampón de recepción en lado receptor.
; -----
; El proceso de envío y recepción serial asíncrono para una
; cadena de caracteres se realiza con protocolo de hardware o
; software, efectuándose la transmisión asíncrona orientada en
; caracteres por medio de un UART realizado por software:
; 300 baudios, 1 bit de inicio, 8 bits de datos, 1 bit de parada,
; sin paridad
; -----
; En ambos MCLS-modular se carga y se inicia el software cmc42b.asm.
; Mediante el conmutador deslizante en P3.5 se define
; qué MCLS-modular hará de TERM1 y de TERM2 para la comunicación.
; Pulsando una tecla (unidad de TECLAS) se puede iniciar un proceso
; de envío opcionalmente desde TERM1 o TERM2. La cadena de caracteres
; que llega al respectivo lado de recepción se introduce
; en una memoria tampón de recepción y, al alcanzarse un límite
; superior,se visualiza por medio de la unidad de LCD. Después del
; vaciado de la memoria tampón de recepción, se continua con la
; transferencia.
; -----
; Por medio de la unidad de LED se pueden señalar los siguientes

```



Programación de interfaces de transmisión de datos CMC 4



```
; estados de la transmisión de datos:
; LED en P5.6: ON, si memoria tampón de recepción vacía
; LED en P5.5: ON, si transmisión de datos finalizada
; LED en P5.1. LED ON si DCD = 0 (se controla desde el lado opuesto)
; LED en P5.0: LED ON si RTS = 0 (¡se ha alcanzado el límite en
; la memoria tampón de recepción!)
; -----
; Nota: ¡IC de interfase del adaptador de V24 invierte el nivel!
; -----

; ----- Acuerdos necesarios para el ensamblador -----

CPU      80515          ; Selección de microcontrolador
INCLUDE c515c.inc      ; Definiciones de SFR
INCLUDE bitfuncs

; Asociación de los pins de puerto para UART de SW
TX      EQU    P1.0      ; Circuito de envío
RX      EQU    P1.1      ; Circuito de recepción

; Asociación de los pins de puerto para indicaciones de estado
; de LED o señales de control RTS
BUFFER_VACIA EQU    P5.6 ; Comunica memoria tampón vaciada
; por medio de LED ON
STATUS_END   EQU    P5.5 ; Comunica que la transmisión de la
; cadena de caracteres ha finalizado
DCD          EQU    P5.1 ; Entrada en el lado emisor
; Evalua RTS del lado receptor e
; interrumpe, en caso de necesidad,
; el proceso de envío Con DCD = 0
; no hay disposición de recepción
RTS          EQU    P5.0 ; Salida en el lado receptor
; Comunica al emisor la disposición
; de recepción RTS = 1 corresponde
; a disposición de recepción (LED OFF)
LIMIT       EQU    16   ; Offset para límite de memoria tampón
; a vigilar

SEGMENT      DATA
INDICADOR   DB    ?      ; Indicador en memoria tampón de datos
TCO         DB    ?      ; Contador de caracteres

SEGMENT      IDATA
REC_BUFFER  DB    20     DUP    (?) ; Reservar 20 bytes para
; memoria tampón de datos

SEGMENT      code
ORG          0000H        ; Primer comando a partir de
; la dirección 0000H
using        bank0        ; Selección RB0

; -----
; JMP      START          ; Salto al programa principal

; ----- Introducción del IV para la int. ext. 6 -----
ORG    006BH      ; Posición IV para la interrupción ext. 6
JMP    ISR_INT6   ; Vector de interrupción (IV)
ORG    100H

; *****
; *                               *
; *****
START:
MOV     SP,#40H
CLR     RTS          ; Aún no hay disposición de recepción
```

```

        CALL  INIT_INT6           ; Inicializar interrupción externa 6
        CALL  INIT_SW_UART       ; Inicializar UART de SW
        CALL  INIT_LCD           ; Inicializar indicador LCD
        SETB  EAL                ; Autorizar aceptaciones de interrupción
                                ; por CPU
S00:    MOV   DPTR,#TEXT0         ; Direccionar texto para inicio
        CALL  OUT_TEXT           ; y visualizar
        CALL  CLR_BUFFER         ; Vaciar memoria tampón de recepción
        CLR   BUFFER_VACIA       ; Conectar indicación de estado
                                ; de LED para memoria tampón vacía
        CLR   STATUS_END         ; Señalizar final de la transmisión
        SETB  RTS                ; Poner estado en disposición
                                ; de recepción

; .....
        CALL  RECEIVE_STRING     ; Esperar corriente de datos seriales
        JMP   S00                ; Bucle sinfín como "final"
                                ; del programa principal
; *****

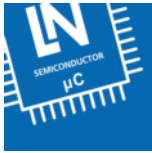
; -----
;                                     Borrar memoria tampón de recepción
; -----
CLR_BUFFER:
        PUSH  ACC
        PUSH  AR7
        PUSH  AR0
        MOV   R7,#20
        MOV   R0,#REC_BUFFER
        MOV   A,#20H
IN0:    MOV   @R0,A
        INC   R0
        DJNZ  R7,IN0
        POP   AR0
        POP   AR7
        POP   ACC
        RET

; -----
;                                     Inicialización de la interrupción ext. 6
;                                     - Disparo por flanco 0-1 -
; -----
INIT_INT6:
        SETB  EX6                ; Autorizar interrupción externa 6
        RET

; -----
;                                     Inicialización del UART de software
; -----
INIT_SW_UART:
        SETB  TX
        SETB  RX
        RET

; -----
;                                     ISR para interrupción ext. 6
;                                     - Iniciar el proceso de envío -
; -----
ISR_INT6:
        PUSH  ACC
        PUSH  DPH
        PUSH  DPL
        JB    P3.5,IS0           ; Si nivel 1, a TERM2
; .....
        MOV   DPTR,#Text1
IS1:    CPL   P5.7                ; Para fines de ensayo

```



```
CALL SEND_STRING
; .....
CALL TIEMPO ; Tiempo para eliminación
; de rebotamiento de tecla

CALL TIEMPO

CLR IEX6 ; Reposicionar bit de solicitud
; de interrupción

POP DPL
POP DPH
POP ACC
RETI

; -----
IS0: MOV DPTR,#Text2
JMP IS1

; -----
; Enviar una cadena de caracteres con UART de SW
; - incluyendo control DCD y señal de final de LED -
; -----
SEND_STRING:
PUSH ACC
PUSH DPH
PUSH DPL
SE1: JNB DCD,SE1 ; Esperar hasta DCD = 1
MOVX A,@DPTR
JZ END_S
CALL TRANSMIT_BYTE
CALL TIEMPO ; Tiempo entre 2 caracteres
CALL WAIT
INC DPTR ; Indicador +1
JMP SE1
END_S: CALL TRANSMIT_BYTE
CLR STATUS_END ; Comunicar final de transmisión
; por medio de LED

POP DPL
POP DPH
POP ACC
RET

; -----
; Recibir una cadena de caracteres con UART de SW
; Visualizar los caracteres en la unidad de LCD
; -----
RECEIVE_STRING:
PUSH ARO
PUSH ACC
RE2: MOV A,#00H ; Posición en línea 0
CALL CURSOR_POS ; Posicionamiento de cursor
MOV INDICADOR,#REC_BUFFER ; Indicador en principio
; de memoria tampón
SETB RTS ; Señalizar disposición
; de recepción con RTS = 1

; .....
NEXT:
; Control Limite Memoria tampón de recepción
CLR CY
MOV A,INDICADOR
SUBB A,#REC_BUFFER+LIMIT
JZ BUFFER_VACIAR ; Si se ha alcanzado el límite,
; vaciar memoria tampón

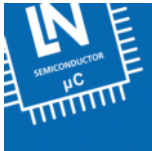
; .....
CALL RECEIVE_BYTE ; Introducir byte en A
CLR CY
SUBB A,#00 ; ¿Carácter final?
```

```

        JZ      END_R          ; De ser así, al final del subprograma
; .....
        SETB   STATUS_END     ; La transmisión aún no ha finalizado
        MOV    R0,INDICADOR
        MOV    @R0,A          ; Carácter a la memoria tampón
        INC    INDICADOR      ; Indicador +1
; .....
        JMP    NEXT          ; Introducir otros caracteres
END_R:
        CLR    RTS           ; No hay disposición de recepción,
                           ; señalar con RTS = 0
        CALL   OUT_BUFFER    ; Escribir contenido de memoria tampón
                           ; en indicador LCD
        CLR    STATUS_END     ; Comunicar final de la transmisión por
                           ; medio de LED
        CALL   TIEMPO_LARGO
        CALL   TIEMPO_LARGO
        CALL   TIEMPO_LARGO
        POP    ACC
        POP    AR0
        RET
; .....
BUFFER_VACIAR:
        CLR    RTS           ; No hay disposición de recepción,
                           ; señalar con RTS = 0
        SETB   BUFFER_VACIA   ; Señalizar memoria tampón llena
        CALL   INIT_LCD      ; Borrar display
        CALL   OUT_BUFFER    ; Escribir contenido de memoria tampón
                           ; en indicador
        CALL   TIEMPO_LARGO
        CALL   CLR_BUFFER     ; Borrar memoria tampón
        CLR    BUFFER_VACIA   ; Conectar indicación de estados de LED
                           ; para memoria tampón
                           ; vacía
        CALL   TIEMPO_LARGO   ; Esperar aproximadamente 1s
        JMP    RE2           ; Continuar
; -----
;   Escribir contenido de memoria tampón en indicador de LCD
; -----
OUT_BUFFER:
        PUSH   AR0
        PUSH   ACC
        MOV    R0,#REC_BUFFER

OUT1:  MOV     A,@R0
        CALL   WRITE_DREG     ; Carácter en indicador LCD
        CALL   TIEMPO_CORTO
        INC    R0
        CJNE   R0,#REC_BUFFER+16,OUT1
        POP    ACC
        POP    AR0
        RET
; -----
;   Salida de texto de 2 líneas en unidad LCD
;   - Cadena de caracteres tiene 0 como indicativo de final -
; -----
OUT_TEXT:
        PUSH   ACC
        MOV    A,#00H         ; Cursor en primera línea
        CALL   CURSOR_POS
        MOV    TCO,#000H      ; Contador de caracteres y Offset
OUT_Z1:
        MOV    A,TCO

```



```
        MOV    A,@A+DPTR          ; Coger caracteres
        JZ     OUT_END            ; ¿Indicativo de final?
        CALL   WRITE_DREG        ; Visualizar caracteres
        CALL   TIEMPO_CORTO
        INC    TCO                ; Contador de caracteres +1
        MOV    A,TCO
        CLR    CY
        SUBB   A,#10H            ; ¿Sigue la línea 1?
        JC     OUT_Z1            ; De ser así, a OUT_Z1
; .....
; Salida para 2ª línea
        MOV    A,#40H            ; Cursor
        CALL   CURSOR_POS        ; al principio de la 2ª línea
; .....
OUT_Z2:
        MOV    A,TCO
        MOV    A,@A+DPTR          ; Coger caracteres
        JZ     OUT_END            ; ¿Indicativo de final?
        CALL   WRITE_DREG        ; Visualizar caracteres
        CALL   TIEMPO_CORTO
        INC    TCO                ; Contador de caracteres +1
        JMP    OUT_Z2
; .....
OUT_END:
        POP    ACC
        RET

; -----
;           Tiempo de eliminación de rebotamiento aprox. 50ms
; -----
TIEMPO:
        PUSH   AR7
        PUSH   AR6
        MOV    R7,#98
T01:    MOV    R6,#000H
T00:    DJNZ   R6,T00
        DJNZ   R7,T01
        POP    AR6
        POP    AR7
        RET

; -----
;           Tiempo corto, aproximadamente 1ms
; -----
TIEMPO_CORTO:
        PUSH   AR7
        PUSH   AR6
        MOV    R7,#2
T04:    MOV    R6,#000H
T03:    DJNZ   R6,T03
        DJNZ   R7,T04
        POP    AR6
        POP    AR7
        RET

; -----
;           Tiempo largo, aproximadamente 1s
; -----
TIEMPO_LARGO:
        PUSH   AR5
        MOV    R5,#20
T02:    CALL   TIEMPO
        DJNZ   R5,T02
        POP    AR5
        RET
```

```

; ----- Funciones de UART de SW -----
; Imitación de la funcionalidad de UART para:
; 1 bit de inicio, 8 bits de datos, 1 bit de parada, sin bit de
; paridad 300 baudios
; -----
; Introducir 1 byte por medio de RXD en A
RECEIVE_BYTE:
    PUSH    AR7
    MOV     R7,#8          ; Contador de bits = 8
REC1: JB     RX,REC1        ; Esperar hasta RXD=0
    CALL    WAIT           ; Esperar tiempo de bit
REC2: CALL    WAIT
;    CALL    WAIT
    MOV     CY,RX          ; Introducir bit
    RRC     A              ; Desplazar en acumulador
    DJNZ    R7,REC2        ; Bucle hasta contador de bit =0
    CALL    WAIT
    POP     AR7
    RET

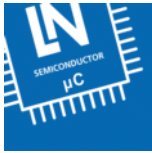
; -----
; Enviar 1 byte en A por medio de TXD
TRANSMIT_BYTE:
    PUSH    AR7
    PUSH    ACC
    MOV     R7,#8          ; Contador de bits = 8
    CLR     TX
    CALL    WAIT           ; Esperar tiempo de bit
TRAN1:
    RRC     A              ; Llevar bit a CY
    MOV     TX,CY          ; Sacar información de bit
    CALL    WAIT           ; Esperar tiempo de bit
    DJNZ    R7,TRAN1       ; Bucle hasta contador de bits =0
    SETB    TX
    CALL    WAIT           ; Tiempo opcional para bit de parada
    CALL    WAIT
    POP     ACC
    POP     AR7
    RET

; -----
; Tiempo de bit para 300 baudios aproximadamente 3,333 ms
; ((71*2+2)*23)+10)us
; -----
WAIT: PUSH    AR7
    PUSH    AR6
    MOV     R6,#23
WA1: MOV     R7,#71
WA2: DJNZ    R7,WA2
    DJNZ    R6,WA1
    POP     AR6
    POP     AR7
    RET

; -----
; Cadena de caracteres para aviso de disposición
; -----
TEXT0:
    DB      "Ready for      Receive!      ", 0
; -----

; -----
; Cadena de caracteres para la primera transferencia serial a TERM2
; -----
TEXT1:

```



```
DB      "The          "
DB      "communication "
DB      "between two MCLS"
DB      "-modular     "
DB      "with hardware- "
DB      "handshaking by "
DB      "RTS/DCD is    "
DB      "allready done! "
DB      "*** This mail "
DB      "sends DEE1 to  "
DB      "DEE2 ***      ", 0

; -----
; Cadena de caracteres para la primera transferencia serial a TERM1
; -----
TEXT2:
DB      "The          "
DB      "communication "
DB      "between two MCLS"
DB      "-modular     "
DB      "with hardware- "
DB      "handshaking by "
DB      "RTS/DCD is    "
DB      "allready done! "
DB      "*** This mail "
DB      "sends DEE2 to  "
DB      "DEE1 ***      ", 0

; -----
;                               Insertar funciones para unidad de LCD
; -----
INCLUDE FCT_LCD.INC      ; Archivo de inclusión
                        ; en el mismo directorio

END
```

CMC 4-3 Salida de datos a una impresora por medio de una interfase CENTRONICS

Utilización de un adaptador CENTRONICS

Mediante el *MCLS-modular*[®] y del adaptador de CENTRONICS que forma parte del sistema se puede realizar una salida de caracteres a una impresora por medio de una interfase CENTRONICS. El adaptador suministra una subcantidad de las señales de interfase estandarizadas en un casquillo Sub-D de 25 polos. Permite la asociación opcional de pins de puerto apropiados del microcontrolador a las señales de interfase CENTRONICS.

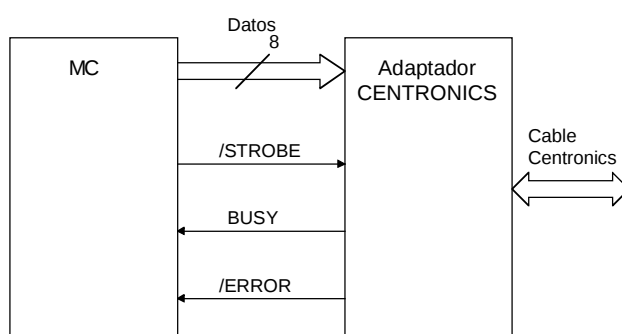


Fig. 301: Ejemplo para la conexión del adaptador CENTRONICS al microcontrolador

Básicamente, la entrega de caracteres a la impresora se puede realizar en el sondeo o de forma controlada por interrupción. Mientras que en el sondeo se vigila la señal BUSY, en el servicio de interrupción se puede utilizar la señal /ACKNLG para activar una interrupción. Resulta muy ventajosa la utilización de los pins para la activación de interrupciones externas, pudiéndose disparar, debido al impulso /ACKNLG muy corto, sobre el flanco ascendente o descendente (ver la fig. 05).

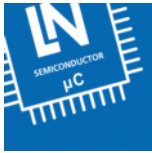
Para obtener más detalles relativos a la interfase CENTRONICS, consulte el manual de instrucciones del adaptador y las indicaciones en el punto D.

Utilización de impresoras existentes

Para los siguientes experimentos se puede utilizar, de un modo general, cualquier impresora con una interfase CENTRONICS. La impresora se conecta por medio de un cable CENTRONICS habitual al adaptador. La funcionalidad lógica necesaria de la interfase se realiza mediante software en el lado del microcontrolador.

Por razones de sencillez, se realiza una emulación EPSON dentro del software. Por ello, probablemente será necesario reconfigurar la impresora empleada para el conjunto de caracteres de control de EPSON. Las indicaciones correspondientes se encuentran normalmente en el manual de instrucciones de la impresora empleada o en la página Web del fabricante de impresoras.

Atención: Para la salida de caracteres por líneas en una impresora se necesita una impresora de líneas (impresora de agujas o impresora de chorro de tinta). Las impresoras láser son impresoras de páginas y no son apropiadas para esta aplicación.



Observe también que en algunos modelos de impresora no se dispone de todas las señales de estado o de control. Este es el caso p. ej. del estado de aparato para „Paper Error“ y de la señal de control „Auto Feed“.

Aparatos y accesorios necesarios

Unid	Denominación	Nº Id.
1	Plataforma con módulo de alim. +5V	SO4000-1E
1	Fuente de alim. de enchufe AC 90...230V 45..65Hz, DC 9V 630mA	SO4000-1F
1	Módulo PSD1-Flash Controlador C515C	SO4000-1G
1	Unidad de adaptador de 8 bits	SO4000-1K
1	Unidad de LED	SO4000-1P
1	Unidad de display 2 (LCD)	SO4000-1S
1	Adaptador de BNC	SO4000-1T
1	Unidad de teclas	SO4000-1Q
1	Adaptador Centronics	SO4000-3K
1	Unidad de temperatura	SO4000-3C
1	Unidad de tensión	SO4000-3D
1	Software IDE para MCLS (D,GB,F,E)	SO4001-9H
1	Cable de interfase serial 9/9 polos	LM9040
14	Cable de medición 2mm 15cm azul	SO5126-5K
10	Cable medición 2mm 15cm amarillo	SO5126-5M
1	CMC 4 Programación de interfaces de transmisión de datos	SH5019-1D

Prepare, para la realización del ensayo, el manual de instrucciones del entorno de desarrollo integrado (IDE) y el manual de instrucciones de la unidad PSD1-Flash. Utilice la información compacta en el anexo del manual de instrucciones del módulo PSD1-Flash.

Infórmase en los manuales de instrucciones de los módulos de experimentación que se van a emplear.

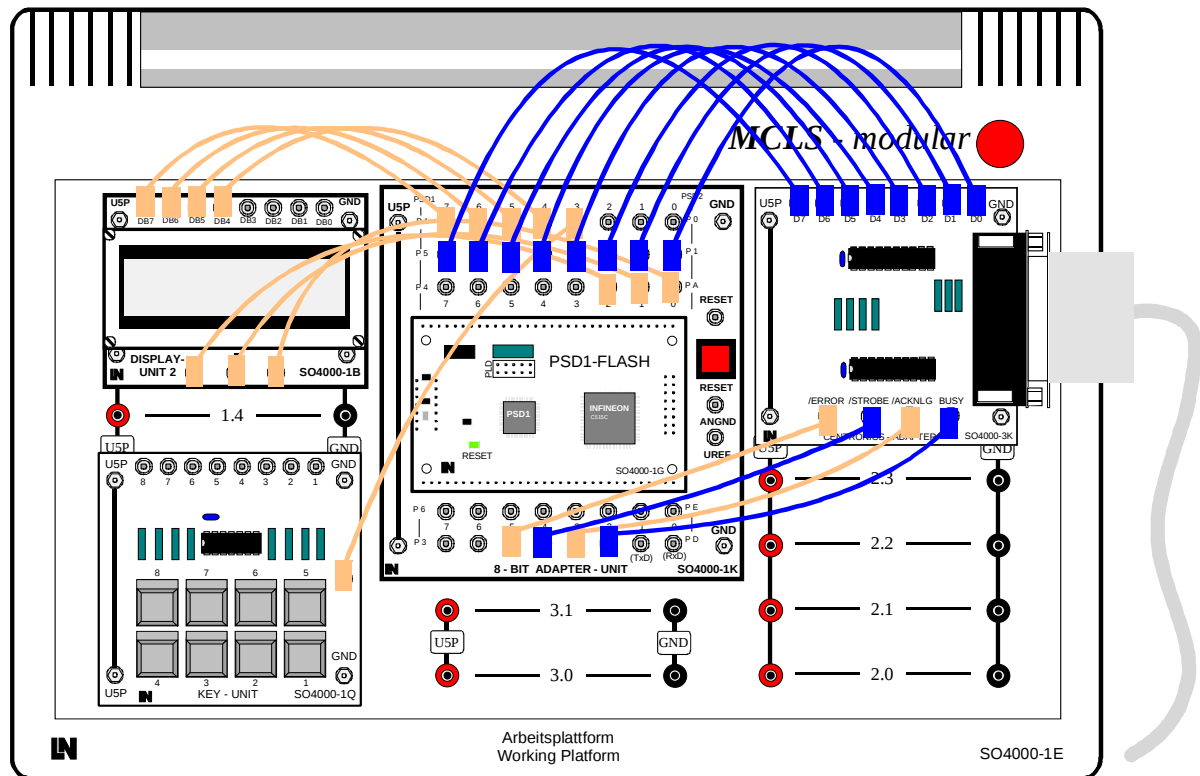


Fig. 302: Instalación de aparatos del ensayo CMC 4-3A

Instalación general de aparatos

¡Separe la alimentación de corriente del MCLS-modular®!

Conecte la plataforma de trabajo por medio del cable de interfase serial a su PC.

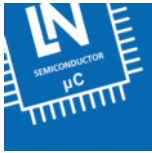
Encaje la unidad de adaptador de 8 bits con el Módulo PSD1-Flash adaptado en ésta en el campo de casquillos 4.0 - 4.5 en el centro de la plataforma.

Seleccione los módulos de experimentación necesarios del ejercicio y conéctelos a los puertos elegidos por usted.

Conecte la alimentación de corriente al MCLS-modular.

Nota relativa a la seguridad:

¡Asegúrese de conectar correctamente los potenciales de tensión (izda U5P, dcha GND)! No se deben aplicar tensiones extrañas a los puntos de medición llevados hacia fuera ¡ya que esto podría conducir a la destrucción de los componentes de las placas!



Ensayo CMC 4-3A

El *MCLS-modular* se conecta, por medio del adaptador CENTRONICS, a una impresora. Pulsando una tecla se inicia el proceso de impresión y se imprime un texto de varias líneas.

¿Qué conocimientos se pretenden transmitir?

- Utilización del adaptador CENTRONICS para la conexión de una impresora
- Utilización de la señal BUSY para el control de la impresora
- Realización de una impresión de texto

Explicaciones relativas al ejercicio:

Al *MCLS-modular* se conecta, por medio del adaptador CENTRONICS, una impresora apropiada. Un texto de varias líneas a definir se deberá sacar por impresora, realizándose el control de la salida de caracteres por medio de un sondeo del estado de impresora BUSY. Pulsando una tecla se iniciará el proceso de impresión. Adicionalmente se visualizará el inicio y el proceso de impresión en el indicador de LCD.

Ejercicios y soluciones

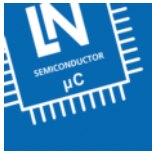
Familiarícese con el adaptador CENTRONICS (ver el manual de instrucciones). ¿Qué señales proporciona el tipo de impresora empleado para la comunicación?

Croquice la conexión del microcontrolador (especificando los pins de puerto) a la interfase CENTRONICS según el texto fuente de ensamblador cmc43a.asm.

Familiarícese con el desarrollo de programa según el texto fuente de ensamblador cmc43a.asm. Croquice el desarrollo del subprograma PRINT.

Explique la salida de un solo carácter según el subprograma OUT_CHAR.

Croquice, para la instalación experimental, un diagrama de bloques sencillo. Realice la instalación experimental según la fig. 302.



Ensaye la función del programa en tiempo real iniciando varias veces el proceso de impresión pulsando una tecla.

Varíe el texto a sacar por impresora y experimente con otros caracteres de control para el tipo de letra según las posibilidades de su impresora.

- ↪ Consulte los caracteres de control necesarios en el manual de instrucciones de la impresora empleada.

Lista de programa CMC 4-3A

```

; *****
; * Proyecto:          Activación de una interfase CENTRONICS      *
; * Archivo fuente:    cmc43a.asm                                  *
; * Herramientas de MC: as, p2hex, depurador                        *
; * Perfil:            Perfil integrado PSD1-C515C-AS              *
; * Recursos MC:       Puerto 5 para datos (D0-D7) a la interfase *
; *                   P3.5 para /ERROR, P3.4 para /STROBE,         *
; *                   P3.3 para /ACKNLG, P3.2 para BUSY,           *
; *                   unidad de LCD en P1.7-P1.4 y P4.0-P4.2,      *
; *                   P1.3 (int. ext. 6) para interr. de tecla,    *
; * Editor:            HAG                                         *
; *****

; ----- Descripción de funcionamiento -----
; Por medio del ADAPTADOR CENTRONICS el MCLS-modular activa una
; impresora con interfase CENTRONICS. En este ejemplo se inicia
; la salida de un texto de varias líneas pulsando una tecla
; (activación de la interrupción externa 6 por medio de la UNIDAD DE
; TECLAS).
; El control de la transmisión de datos por caracteres a la impresora
; se realiza mediante un sondeo de la señal de impresora BUSY.
; -----
; El inicio de impresión y su final se visualiza
; adicionalmente por medio de una visualización de texto en la
; unidad de LCD.
; Los eventuales mensajes de error de la impresora también
; se visualizan.
; -----
; Observe que la impresora utilizada por usted probablemente
; requiere otros caracteres de control que los indicados en el texto
; fuente.
; En caso de necesidad, realice mediante el manual de instrucciones
; de la impresora una adaptación de los caracteres de control o
; configure su impresora con respecto a la emulación de EPSON.
; ¡Observe también que no todas las interfases CENTRONICS de las
; impresoras ofrecen en el contacto 15 un estado de ERROR de la
; impresora!
; -----

; -----Acuerdos necesarios para el ensamblador -----

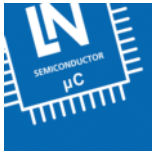
        CPU      80515                ; Selección de microcontrolador
        INCLUDE  c515c.inc            ; Definiciones de SFR
        INCLUDE  bitfuncs

; Asociación de los puertos o pins de puertos para
; la INTERFASE CENTRONICS
DATEN    EQU    P5                    ; Datos D0 a D7 de la interfase
ERROR    EQU    P3.5                  ; Señal de estado /ERROR
STROBE   EQU    P3.4                  ; Señal de control /STROBE
ACKNLG   EQU    P3.3                  ; Señal de confirmación /ACKNLG
BUSY     EQU    P3.2                  ; Señal de estado /BUSY

; Definición de caracteres de control (emulación de EPSON)
LF       EQU    0AH                    ; Avance de línea
CR       EQU    0DH                    ; Retorno de carro
ESC      EQU    1BH                    ; Escape
RES      EQU    '@'                    ; Escape + @ = Reset impresora
SO       EQU    0EH                    ; Impresión ancha
SI       EQU    0FH                    ; Impresión fina

        SEGMENT      BITDATA

```



Programación de interfaces de transmisión de datos CMC 4



```
ENABLE      DB      ?          ; Definir byte de estado
            SEGMENT DATA
TCO          DB      ?          ; Contador de caracteres
            SEGMENT code        ; A partir de aquí segmento de código
            ORG      0000H      ; Primer comando a partir de
                                ; la dirección 0000H
            using    bank0      ; Selección RB0

            JMP      START      ; Salto al programa principal

; ----- Introducción del IV para la int. ext. 6 -----
            ORG      006BH      ; Posición IV para la interrupción ext. 6
            JMP      ISR_INT6    ; Vector de interrupción (IV)
            ORG      100H

; *****
; *                               Programa principal                               *
; *****
START:
            CLR      ENABLE      ; Bloquear impresión
            CALL     INIT_INT6    ; Inicializar interrupción ext. 6
            CALL     INIT_LCD     ; Inicializar indicador de LCD
            MOV      DPTR,#TEXT0  ; Direccionar y visualizar texto
            CALL     OUT_TEXT     ; para inicio
            CALL     TIEMPO_LARGO
            CALL     TIEMPO_LARGO
S00:        JNB      ENABLE,S00   ; Esperar la autorización de impresión
            CALL     PRINT        ; Salida de texto en impresora
            CLR      ENABLE      ; Bloquear impresión
            JMP      S00          ; Bucle sinfín como "final"
                                ; del programa principal

; -----
;                               Inicialización de la interrupción externa 6
;                               - Disparo por flanco 0-1 -
; -----
INIT_INT6:
            SETB     EX6          ; Autorizar interrupción externa 6
            SETB     EAL
            RET

; -----
;                               Salida en impresora por medio de interfase CENTRONICS
; -----
PRINT:      PUSH     ACC
            PUSH     AR1
            PUSH     DPH
            PUSH     DPL
            JNB      ERROR,P01    ; Al mensaje de error, si ERROR=0
            MOV      DPTR,#Text1  ; Si no, visualizar
            CALL     OUT_TEXT     ; comienzo de transferencia de datos
            CALL     TIEMPO_LARGO

; .....
            MOV      R1,#000      ; Offset para indicador
            MOV      DPTR,#PRINT_TEXT ; Indicador en principio de texto
P03:        MOV      A,R1
            MOVC     A,@a+DPTR    ; Coger caracteres
            CJNE     A,#0,P04      ; ¿Indicativo de final alcanzado?
; .....
P02:        MOV      DPTR,#Text3  ; Visualizar
            CALL     OUT_TEXT     ; mensaje de listo
P05:        CALL     TIEMPO_LARGO
            CALL     TIEMPO_LARGO
            POP      DPL
            POP      DPH
```

```

        POP    AR1
        POP    ACC
        RET

; -----
P04:    CALL   OUT_CHAR          ; Sacar un solo carácter por impresora
        INC    R1                ; Offset +1
        SJMP   P03

; .....
P01:    MOV     DPTR,#Text2      ; Visualizar
        CALL   OUT_TEXT         ; mensaje de error
        JMP    P05

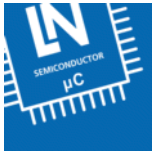
; -----
;                               Enviar un solo carácter a la impresora
; -----
OUT_CHAR:
        JB     BUSY,$            ; ¿Impresora lista?
                                   ; De no ser así, seguir consultando
        MOV     DATEN,A          ; De ser así, sacar carácter
                                   ; por impresora
        CLR     STROBE           ; Generar reloj de toma
        SETB    STROBE
        RET

; -----
;                               ISR para interrupción ext. 6
;                               - Iniciar proceso de impresión -
; -----
ISR_INT6:
        SETB    ENABLE          ; Autorizar impresión
        CALL    TIEMPO          ; Tiempo para eliminación
                                   ; de rebotamiento de tecla
        CALL    TIEMPO
        CLR     IEX6            ; Reposicionar bit de solicitud
                                   ; de interrupción
        RETI

; -----
;                               Visualización de texto de 2 líneas en la unidad de LCD
;                               - Cadena de caracteres tiene 0 como indicativo de final -
; -----
OUT_TEXT:
        PUSH    ACC
        MOV     A,#00H          ; Cursor en primera línea
        CALL    CURSOR_POS
        MOV     TCO,#000H       ; Contador de caracteres y offset
OUT_Z1:
        MOV     A,TCO
        MOVC    A,@A+DPTR       ; Coger caracteres
        JZ      OUT_END         ; ¿Indicativo de final?
        CALL    WRITE_DREG      ; Carácter para visualización
        CALL    TIEMPO_CORTO
        INC     TCO              ; Contador de caracteres +1
        MOV     A,TCO
        CLR     CY
        SUBB    A,#10H          ; ¿Aún línea 1?
        JC      OUT_Z1          ; De ser así, a OUT_Z1
; .....

        MOV     A,#40H          ; Cursor
        CALL    CURSOR_POS       ; al principio de la 2ª línea
; .....
OUT_Z2:
        MOV     A,TCO
        MOVC    A,@A+DPTR       ; Coger caracteres

```



Programación de interfaces de transmisión de datos CMC 4



```
JZ      OUT_END          ; ¿Indicativo de final?
CALL    WRITE_DREG       ; Caracteres para visualización
CALL    TIEMPO_CORTO
INC      TCO              ; Contador de caracteres +1
JMP      OUT_Z2

; .....
OUT_END:
POP      ACC
RET

; -----
;                      Tiempo corto aproximadamente 1ms
; -----
TIEMPO_CORTO:
PUSH     AR7
PUSH     AR6
MOV      R7,#2
T04:     MOV      R6,#000H
T03:     DJNZ     R6,T03
          DJNZ     R7,T04
POP      AR6
POP      AR7
RET

; -----
;                      Tiempo de eliminación de rebotamiento aprox. 50ms
; -----
TIEMPO:
PUSH     AR7
PUSH     AR6
MOV      R7,#98
T01:     MOV      R6,#000H
T00:     DJNZ     R6,T00
          DJNZ     R7,T01
POP      AR6
POP      AR7
RET

; -----
;                      Tiempo largo aprox. 1s
; -----
TIEMPO_LARGO:
PUSH     AR5
MOV      R5,#20
T02:     CALL     TIEMPO
          DJNZ     R5,T02
POP      AR5
RET

; -----
;                      Cadena de caracteres para aviso de disposición
; -----
TEXT0:
DB       "Ready for      Print!      ", 0

; -----
; -----
;                      Cadena de caracteres para inicio del proceso de impresión
; -----
TEXT1:
DB       "Data to Printer!      ", 0

; -----
; -----
;                      Cadena de caracteres para mensaje de error
```

```

; -----
TEXT2:
    DB    "Printerstatus  ERROR!          ", 0
; -----

; -----
;          Cadena de caracteres para mensaje de listo
; -----
TEXT3:
    DB    "Print ready!                  ", 0
; -----

; -----
;          Cadena de caracteres para salida en impresora
; -----
PRINT_TEXT:
    DB    ESC, RES
    DB    "MCLS-modular sends String to Printer"
    DB    LF, CR
    DB    "by CENTRONICS-Interface"
    DB    LF, CR, LF

    DB    SO
    DB    "MCLS-modular sends String to Printer"
    DB    LF, CR
    DB    "by CENTRONICS-Interface"
    DB    LF, CR, LF

    DB    SI
    DB    "MCLS-modular sends String to Printer"
    DB    LF, CR
    DB    "by CENTRONICS-Interface"
    DB    LF, CR, LF, LF
    DB    00                                ; Indicativo de final

; -----
;          Insertar funciones para unidad de LCD
; -----
    INCLUDE FCT_LCD.INC    ; Archivo de inclusión
                           ; en el mismo directorio

END

```

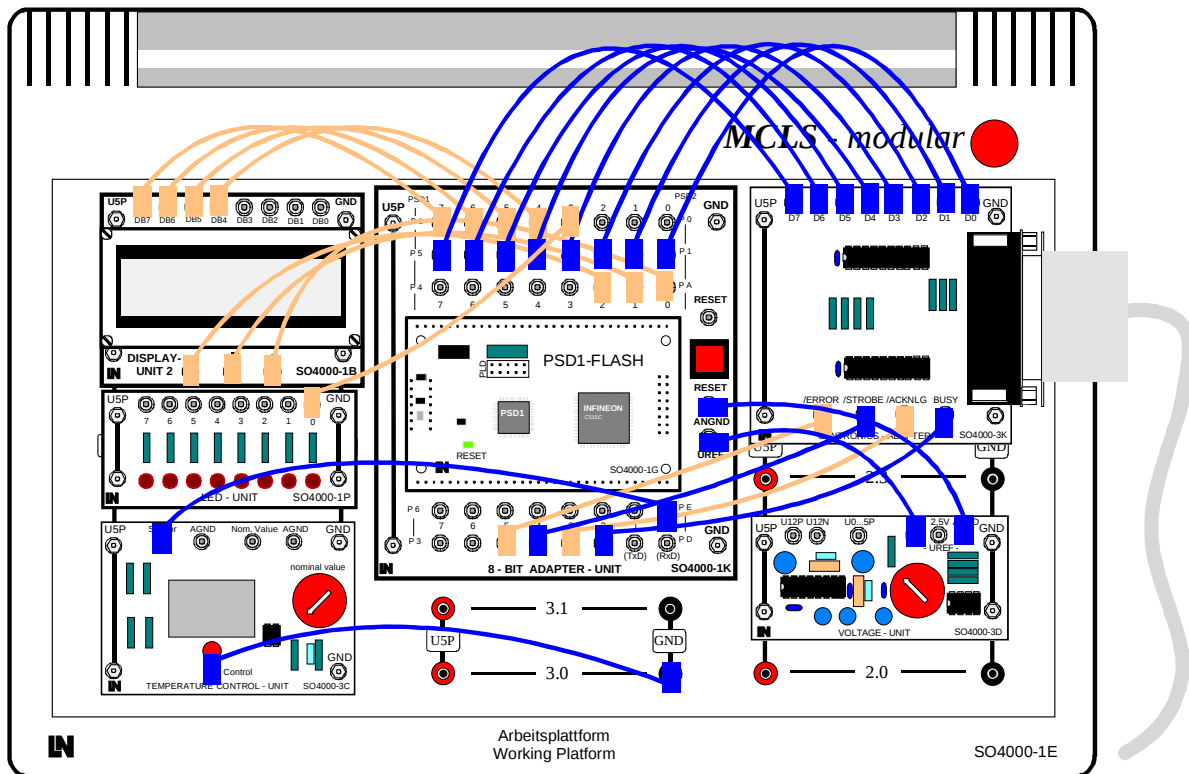


Fig. 303: Instalación de aparatos del ensayo CMC 4-3B

Ensayo CMC 4-3B

El *MCLS-modular* se conecta, por medio del adaptador CENTRONICS, a una impresora. Por medio de un marcador de tiempo se protocola, en intervalos de 10 segundos, un valor de medición de temperatura mediante la impresora. El control de la transmisión de datos por caracteres a la impresora se realizará de forma controlada por interrupciones.

¿Qué conocimientos se pretenden transmitir?

- Utilización del adaptador CENTRONICS para la conexión de una impresora
- Utilización de la señal /ACKNLG para el control de la impresora
- Realización de una salida cíclica de valores medidos

Explicaciones relativas al ejercicio:

Al *MCLS-modular*[®] se conecta, por medio del adaptador CENTRONICS, una impresora apropiada. Un valor de medición de temperatura se enviará cíclicamente, en intervalos de 10 segundos, a la impresora para protocolar el curso de la temperatura. El control del envío de caracteres a la impresora se realiza de forma controlada por interrupción mediante la señal /ACKNLG.

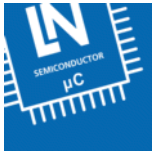
Ejercicios y soluciones

Familiarícese con el desarrollo temporal en la interfase CENTRONICS (ver manual de instrucciones). ¿Qué longitud típica tiene el impulso /ACKNLG?

¿Qué interrupción se utiliza para la evaluación del impulso /ACKNLG?

Familiarícese con el desarrollo del programa según el texto fuente de ensamblador cmc43b.asm. Croquice el desarrollo de programa de la rutina de servicio de interrupción ISR_INT1.

Croquice, para la instalación experimental, un diagrama de bloques sencillo. Realice la instalación experimental según la fig. 303.



Ensaye la función del programa en tiempo real. Caliente el cuerpo metálico de la unidad de temperatura a través de la correspondiente conexión de la entrada de control y observe la salida de los valores de medición de temperatura en el protocolo de la impresora.

➡ Observe el tiempo de ciclo de 10 segundos.

¿Por qué está indicado en el subprograma MEDIR_TEMP el factor para la normalización con 4,9mV/Bit?

¿Cómo se realiza en el programa cmc43B el tiempo de ciclo para el proceso de medición de 10s?

Lista de programa CMC 4-3B

```

; *****
; * Proyecto:          Imprimir valores medidos          *
; * Archivo fuente:    cmc43b.asm                        *
; * Herramientas de MC: as, p2hex, depurador             *
; * Perfil:            Perfil integrado PSD1-C515C-AS     *
; * Recursos de MC:    Puerto 5 para datos (D0-D7) a la interfase *
; *                  P3.5 para /ERROR, P3.4 para /STROBE, *
; *                  P3.3 para /ACKNLG, P3.2 para BUSY,   *
; *                  unidad de LCD en P1.7-P1.4 y P4.0-P4.2, *
; *                  LED en P1.3 - Control interr. de temporiz., *
; *                  señal +sensor de la unidad de control *
; *                  de temperatura en AN7 de la unidad PSD1, *
; *                  UNIDAD de tensión para preparación de la *
; *                  tensión de referencia externa UREF/UGND *
; * Editor:           HAG                                *
; *****

; ----- Descripción de funcionamiento -----
; Por medio de un marcador de tiempo (Timer 2) se mide cíclicamente,
; en intervalos de 10 segundos, la temperatura del sensor en la UNIDAD
; de control de temperatura y se envía por medio del ADAPTADOR
; CENTRONICS a la impresora.
; El control de la transmisión de datos por caracteres a la impresora
; para los valores de medición de temperatura se realiza de forma
; controlada por interrupciones por medio de la señal /ACKNLG.
; -----
; El inicio de la salida en impresora y su final se visualizan
; adicionalmente por medio de una salida de texto en la unidad de LCD.
; Los eventuales mensajes de error de la impresora también
; se visualizan.
; -----
; Observe que la impresora utilizada por usted probablemente
; requiere otros caracteres de control que los indicados en el texto
; fuente.
; En caso de necesidad, realice mediante el manual de instrucciones de
; la impresora una adaptación de los caracteres de control o configure
; su impresora con respecto a la emulación de EPSON.
; ¡Observe también que no todas las interfaces CENTRONICS de las
; impresoras ofrecen en el contacto 15 un estado de ERROR de la
; impresora!
; -----

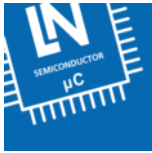
; -----Acuerdos necesarios para el ensamblador -----

        CPU      80515          ; Selección de microcontrolador
        INCLUDE  c515c.inc      ; Definiciones de SFR
        INCLUDE  bitfuncs

; ----- Constante de tiempo para Timer 2 -----
TCO     EQU     -50000          ; 50000*1us hasta el desbordamiento
; -----

; -----
; Asociación de los puertos o pins de puerto para
; la INTERFASE CENTRONICS
; -----
DATOS    EQU     P5              ; Datos D0 a D7 de la interfase
ERROR    EQU     P3.5           ; Señal de estado /ERROR
STROBE   EQU     P3.4           ; Señal de control /STROBE
ACKNLG   EQU     P3.3           ; Señal de confirmación /ACKNLG

```



Programación de interfaces de transmisión de datos CMC 4



```
BUSY      EQU    P3.2      ; Señal de estado /BUSY

; -----
;      Definición de caracteres de control (emulación EPSON)
; -----
LF      EQU    0AH      ; Avance de línea
CR      EQU    0DH      ; Retorno de carro
ESC     EQU    1BH      ; Escape
RES     EQU    '@'      ; Escape + @ = Reset Impresora

; -----
;      SEGMENT      DATA
; -----
; Variablen
VCO_CHARACTER DB    ?      ; Contador de caracteres
VCO_LINEAS   DB    ?      ; Contador de líneas
VCO          DB    ?      ; Variable de contador
; Reservar espacio de memoria para valores medidos
BCD_MSB     DB    ?      ; MSB del valor de medición BCD
BCD_MIB     DB    ?      ; Byte central del valor de med. BCD
BCD_LSB     DB    ?      ; LSB del valor de medición BCD
BIN_MSB     DB    ?      ; Valor de medición binario MSB
BIN_LSB     DB    ?      ; Valor de medición binario LSB
ASCII       DB    ?      ; 4 bytes para valores de medición ASCII
            DB    ?
            DB    ?
            DB    ?

; -----
;      A partir de aquí segmento de código
; -----
; -----
;      SEGMENT      code
;      ORG          0000H      ; Primer comando a partir de dirección
;                          ; 0000H
;      using        bank0      ; Selección banco de registro 0

;      JMP      START      ; Salto al programa principal

; ----- Introducción del IV para la int. ext. 1 -----
;      ORG      0013H      ; Posición IV para la interrupción ext. 1
;      JMP      ISR_INT1    ; Vector de interrupción (IV)

; ----- Introducción del IV para Timer 2 -----
;      ORG      002BH      ; Posición IV para la interrupción
;                          ; de Timer 2
;      JMP      ISR_T2      ; Vector de interrupción (IV)

;      ORG      100H

; *****
; *                      Programa principal                      *
; *****
START:
    CALL  INIT_ADU      ; Inicializar convertidor A/D
    CALL  INIT_LCD      ; Inicializar indicador de LCD
    MOV   DPTR,#TEXT0    ; Direccionar texto para inicio
    CALL  OUT_TEXT      ; y visualizar
    CALL  START_PRINTER  ; Salida en impresora del título
;                          ; de protocolo
    CALL  INIT_INT1     ; Inicialización de interrupción ext. 1
    CALL  INIT_T2       ; Inicializar Timer 2
    SETB  EAL           ; Autorizar interrupciones
S00:     JMP  S00        ; Bucle sinfín como "final"
;                          ; del programa principal
```

```

; -----
; Inicializar convertidor A/D para valor de temperatura en AN7 porque
; valor de medición máx. 1000 mV -> Rango de medición 0 - 1,25 V
; -----
INIT_ADU:
    ANL    ADCON0,#11000000B
    ORL    ADCON0,#00001111B    ; Conversión continua, canal AN7
    MOV    ADDATL,#00h          ; Start de medición 0 - 1,25V
    RET

; -----
; Inicialización de la interrupción ext. 1
; - Disparo por flanco 0-1 -
; -----
INIT_INT1:
    SETB   IT1                  ; Flanco 1-0 activa interrupción
    SETB   EX1                  ; Autorizar interrupción externa 1
    RET

; -----
; Inicialización de Timer 2
; - cada 50 ms una interrupción -
; -----
INIT_T2:
    MOV     TH2,#HI(TCO)
    MOV     TL2,#LO(TCO)
    MOV     CRCH,#HI(TCO)
    MOV     CRCL,#LO(TCO)
    MOV     T2CON,#00010001B
    SETB    ET2
    MOV     VCO,#200            ; 200*50ms, inicializar variable
                                ; de contado
                                ; Cada 10 segundos una impresión
    RET

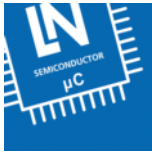
; -----
; Imprimir el título de protocolo
; incluyendo evaluación del estado de impresora
; -----
START_PRINTER:
P05:  JNB    ERROR,P01          ; Al mensaje de error, si ERROR=0
; .....
    MOV     R1,#000            ; Offset para indicador
    MOV     DPTR,#TEXT1        ; Indicador en principio de texto
P03:  MOV     A,R1
    MOVC    A,@A+DPTR          ; Coger caracteres
    CJNE    A,#0,P04           ; ¿Indicativo de final alcanzado?
; .....
    RET

; -----
P04:  CALL    OUT_CHAR          ; Sacar un solo carácter
    INC     R1                  ; Offset +1
    SJMP    P03

; .....
P01:  MOV     DPTR,#Text2        ; Visualizar
    CALL    OUT_TEXT            ; mensaje de error
    CALL    TIEMPO_LARGO
    JMP     P05

; -----
; Enviar un solo carácter a la impresora
; para impresión del título de protocolo sondeo de BUSY
; -----
OUT_CHAR:
    JB      BUSY,$              ; ¿Impresora lista?

```



Programación de interfases de transmisión de datos CMC 4



```
MOV    DATOS,A                ; De no ser así, seguir con consulta
CLR    STROBE                 ; De ser así, sacar caracteres
SETB   STROBE                 ; Generar reloj de toma
RET

; -----
;     ISR contiene las funciones Medir, Visualizar y
;     Activación de impresión para salida de valores medidos
; -----
ISR_T2:
    CLR    TF2                ; Reposicionar indicador de desbordamiento
Timer 2
    CPL    P1.3               ; Por medio de LED control de interrupción
Timer 2
    JNB    ERROR,IS1          ; Al mensaje de error, si ERROR=0
    DJNZ   VCO,IS0             ; Contador-1, =0?
; .....
; Cada 10 segundos
    MOV    DPTR,#TEXT0
    CALL   OUT_TEXT

    MOV    VCO,#200            ; Cargar nuevamente contador auxiliar
    CALL   MEDIR_TEMP           ; Preparar valor medido para
                                ; visualización
    CALL   VISUALIZAR_TEMP      ; Escribir nuevo valor medido en memoria
                                ; tampón y visualizar
; .....
    MOV    DPTR,#BUFFER        ; Indicador en principio de memoria
                                ; tampón
    MOV    VCO_CHARACTER,#0     ; Offset en principio de memoria tampón
; .....
    SETB   IE1                 ; Activar int. ext. - iniciar impresión
IS0:    RETI
; -----
IS1:    MOV    DPTR,#TEXT2      ; Visualizar
    CALL   OUT_TEXT            ; mensaje de error
    JMP    IS0
; -----
;     Salida de memoria tampón de valores medidos en impresora
; -----
ISR_INT1:
    PUSH   ACC
    MOV    A,VCO_CHARACTER
    MOVC   A,@A+DPTR           ; Coger caracteres de memoria tampón
    CJNE   A,#00,IS3           ; ¿Indicativo de final?
IS4:    POP    ACC
    RETI
IS3:    MOV    DATOS,A          ; Caracteres a puerto de datos
    CLR    STROBE              ; Generar reloj de toma
    SETB   STROBE
    INC    VCO_CHARACTER        ; Offset +1
    JMP    IS4
; -----
;     Medir la temperatura incluyendo preparación de valores medidos
; -----
MEDIR_TEMP:
    MOV    B,#49                ; Factor para normalización,
                                ; porque aprox. 4,9 mV/Bit
    CALL   MEDIR
    RET
```

```

; -----
;   Toma, normalización y conversión BCD del valor medido
; -----
MEDIR:
    CALL    SCAL_TEMP          ; Scaling to virtual window
    MUL     AB

; .....
    MOV     BIN_MSB,B          ; Preparar valor medido de 16 bits
    MOV     BIN_LSB,A          ;
    CALL    CONV_BIN_BCD       ; Conversión binaria en BCD
; .....
; Dividir valor por 10
    MOV     A,BCD_LSB
    SWAP    A
    ANL     A,#00001111B
    MOV     B,A

    MOV     A,BCD_MIB
    SWAP    A
    ANL     A,#11110000B
    ADD     A,B
    MOV     BCD_LSB,A          ; BCD_LSB listo

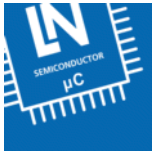
    MOV     A,BCD_MIB
    SWAP    A
    ANL     A,#00001111B
    MOV     B,A

    MOV     A,BCD_MSB
    SWAP    A
    ANL     A,#11110000B
    ADD     A,B
    MOV     BCD_MIB,A          ; BCD_MIB listo
    RET

; -----
;   Scaling to vitual measurement window 0...1.25V
;   Inputs: 10-Bit value in ADDATH and ADDATL
;   Output: scaled 8-Bit value in A
; -----
SCAL_TEMP:
    PUSH     B
; shift 8-Bit value in ADDATH two bits to the left
    MOV     A,ADDATH
    RL      A
    RL      A
    ANL     A,#11111100b
    MOV     B,A
; shift 8-Bit value in ADDATL two bits to the left
; and add to the scaled value in ADDATH
    MOV     A,ADDATL
    RL      A
    RL      A
    ANL     A,#00000011b
    ADD     A,B                  ; scaled value in A
    POP     B
    RET

; -----
;   Conversión de valor binario en valor BCD (comprimido)
; -----
CONV_BIN_BCD:
; Borrar zona de resultados
    MOV     R1,#003H            ; 3 posiciones de memoria
    MOV     R0,#BCD_MSB

```



Programación de interfaces de transmisión de datos CMC 4



```
K4:  MOV    @R0,#000H          ; Cargar con 00
      INC    R0
      DJNZ   R1,K4

; .....
;      ; Conversión    Número de 16 bits en número BCD
K0:  MOV    R4,#10H            ; Convertir 16 bits
      MOV    R0,#BIN_LSB       ; Dirección valor binario LSB
      MOV    R3,#002H          ; 2 bytes a convertir
      CLR    C                 ; Borrar CF
K1:  MOV    A,@R0              ; Cargar valor binario
      RLC    A                 ; Poner CF según rotación
      MOV    @R0,A             ; Recargar resultado de la rotación
      DEC    R0                ; Dirección valor binario MSB
      DJNZ   R3,K1             ; Hacer lo mismo para MSB

; .....
      MOV    R1,#BCD_LSB       ; Dirección de destino BCD LSB
      MOV    R3,#003H          ; Contador de bucles
K2:  MOV    A,@R1              ; Coger último valor de destino
      ADDC   A,@R1             ; Sumar CF
      DA     A                 ; Corrección decimal
      MOV    @R1,A             ; Archivar resultado
      DEC    R1                ; Siguiendo dirección de destino BCD
      DJNZ   R3,K2             ; ¿3ª dirección de destino ya alcanzada
?    DJNZ   R4,K0              ; ¿16 bits ya convertidos ?
      RET

; -----
;      Generación de caracteres ASCII para indicador de LCD
; -----
KONV_ASCII:
; Conversión en caracteres ASCII
      MOV    R0,#ASCII         ; Indicador en zona ASCII
      MOV    A,BCD_MIB
      SWAP   A
      ANL    A,#00001111B
      ADD    A,#30H
      MOV    @R0,A

; .....
      INC    R0
      MOV    A,BCD_MIB
      ANL    A,#00001111B
      ADD    A,#30H
      MOV    @R0,A

; .....
      INC    R0
      MOV    A,BCD_LSB
      SWAP   A
      ANL    A,#00001111B
      ADD    A,#30H
      MOV    @R0,A

; .....
      INC    R0
      MOV    A,BCD_LSB
      ANL    A,#00001111B
      ADD    A,#30H
      MOV    @R0,A
      RET

; -----
;      Valor medido a la memoria tampón y visualización del valor
;      de temperatura
; -----
VISUALIZAR_TEMP:
      CALL   KONV_ASCII        ; Generación de caracteres ASCII
```

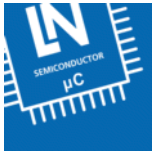
```

; Escribir valor medido en memoria tampón
MOV DPTR,#BUFFER+7
MOV A,ASCII+1
MOVX @DPTR,A
MOV DPTR,#BUFFER+8
MOV A,ASCII+2
MOVX @DPTR,A
; Escribir contenido de memoria tampón en indicador de LCD
MOV A,#040H ; Posición en 2ª línea
CALL CURSOR_POS ; Posicionamiento de cursor
MOV DPTR,#BUFFER
MOV B,#00H
E02: MOV A,B
MOVC A,@A+DPTR
JZ E01
CALL WRITE_DREG
CALL WAIT_SHORT
INC B
JMP E02
E01: RET

; -----
; Salida de texto de 2 líneas en la unidad de LCD
; - Cadena de caracteres tiene 0 como indicativo de final -
; -----
OUT_TEXT:
PUSH ACC
MOV A,#00H ; Cursor en primera línea
CALL CURSOR_POS
MOV VCO_CHARACTER,#000H ; Contador de caracteres y offset
OUT_Z1:
MOV A,VCO_CHARACTER
MOVC A,@A+DPTR ; Coger caracteres
JZ OUT_END ; ¿Indicativo de final?
CALL WRITE_DREG ; Caracteres al indicador
CALL TIEMPO_CORTO
INC VCO_CHARACTER ; Contador de caracteres +1
MOV A,VCO_CHARACTER
CLR CY
SUBB A,#10H ; ¿Aún 1ª línea?
JC OUT_Z1 ; De ser así, a OUT_Z1
; .....
; Salida para 2ª línea
MOV A,#40H ; Cursor
CALL CURSOR_POS ; al principio de la 2ª línea
; .....
OUT_Z2:
MOV A,VCO_CHARACTER
MOVC A,@A+DPTR ; Coger caracteres
JZ OUT_END ; ¿Indicativo de final?
CALL WRITE_DREG ; Caracteres al indicador
CALL TIEMPO_CORTO
INC VCO_CHARACTER ; Contador de caracteres +1
JMP OUT_Z2
; .....
OUT_END:
POP ACC
RET

; -----
; Tiempo corto aprox. 1ms
; -----
TIEMPO_CORTO:
PUSH AR7
PUSH AR6

```



```

        MOV     R7,#2
T04:    MOV     R6,#000H
T03:    DJNZ    R6,T03
        DJNZ    R7,T04
        POP     AR6
        POP     AR7
        RET

; -----
;           Tiempo de eliminación de rebotamiento aprox. 50ms
; -----
TIEMPO:
        PUSH    AR7
        PUSH    AR6
        MOV     R7,#98
T01:    MOV     R6,#000H
T00:    DJNZ    R6,T00
        DJNZ    R7,T01
        POP     AR6
        POP     AR7
        RET

; -----
;           Tiempo largo aprox. 1s
; -----
TIEMPO_LARGO:
        PUSH    AR5
        MOV     R5,#20
T02:    CALL    TIEMPO
        DJNZ    R5,T02
        POP     AR5
        RET

; -----
;   Cadena de caracteres para mensaje de listo por medio LCD
; -----
TEXT0:  DB      "Ready to Print!           ", 0
; -----

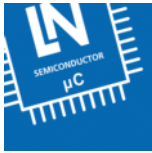
; -----
;   Cadena de caracteres para impresora para inicio de la impresión
; -----
TEXT1:  DB      ESC, RES
        DB      LF, CR
        DB      "Measurement to  Printer!   ", LF, LF, CR, 0
; -----

; -----
;   Cadena de caracteres para mensaje de error por medio de LCD
; -----
TEXT2:  DB      "Printerstatus  ERROR!       ", 0
; -----

; -----
;           Acuerdo memoria tampón de resultados
; -----
BUFFER  db      "Temp.: xx", "grd ", "C  ", LF, CR,
        db      00H                      ; Indicativo de final

; -----
;           Insertar funciones para unidad de LCD
; -----
```

```
; -----  
    INCLUDE FCT_LCD.INC      ; Archivo de inclusión  
                               ; en el mismo directorio  
  
END
```



Nos gusta trabajar con y para usted.

Usted es nuestro mejor colaborador. En base a las experiencias que haya tenido con nuestros equipos, puede contribuir a mejorarlos dándonos sugerencias o avisándonos de cualquier error encontrado. Toda observación que nos haga será tratada con importancia y tomada en cuenta a la hora de actualizar nuestras guías de ejercicios.

¡Muchas gracias por su interés y su colaboración!

**Asunto: Guía de
ejercicios**

Notas:

Fecha:

Copyright © 2006 LUCAS-NÜLLE GmbH. Reservados todos los derechos.

La presente guía de ejercicios está protegida por derechos de autor. Reservados todos los derechos. Este manual no podrá reproducirse, almacenarse o transmitirse de ninguna forma ni por ningún medio, sea éste fotocopia, microfilm o electrónico, o transformarse a un lenguaje que pueda ser leído por máquinas, y en especial las de procesamiento de datos, sin la previa autorización escrita por parte de LUCAS-NÜLLE GmbH.

Sólo se permitirá la reproducción de las hojas de trabajo del alumno, sin efectuar cambios de su contenido, dentro de la institución que haya adquirido la presente guía para fines de utilización en clase.

LUCAS-NÜLLE GmbH no se responsabiliza por daños ocasionados en los dispositivos debido a cambios de la información efectuados por terceros.

LUCAS-NÜLLE Lehr- und Messgeräte GmbH

Dirección: Siemensstr. 2 • D-50170 Kerpen (Sindorf) • Alemania

Apdo. postal: Postfach 11 40 • D-50140 Kerpen • Alemania

Telf.: + 49 / (0)2273 / 567-0 • Fax: +49 / (0)2273 / 567-30 • vertrieb@lucas-nuelle.com

MCLS - modular® es una marca registrada de Lucas-Nülle GmbH

Copyright: Prof. Dr.-Ing. Olaf Hagenbruch

Universidad de Ciencias Aplicadas de Mittweida

Lucas-Nülle Lehr- und Meßgeräte GmbH

Siemensstraße 2 · D-50170 Kerpen-Sindorf
Telefon +49 2273 567-0 · Fax +49 2273 567-30

www.lucas-nuelle.de