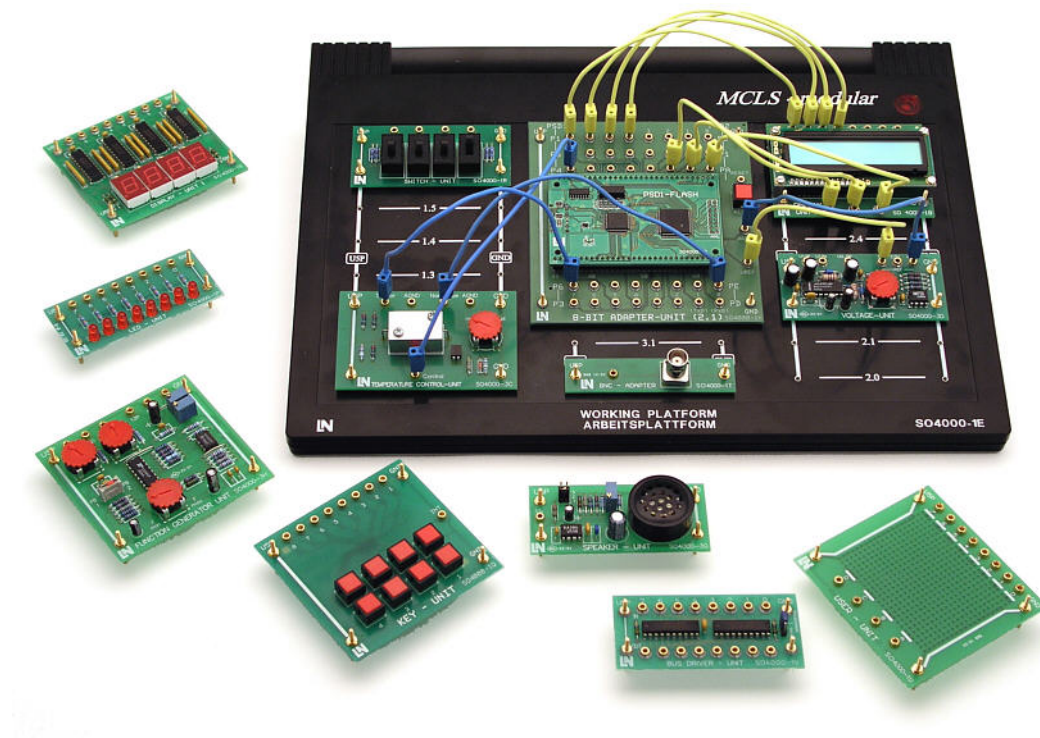




Programación de periféricos on-chip (C515C)

CMC 3



Guía de ejercicios para el estudiante

SH5004-1C

3ra. edición

Autor: equipo de autores ACMC Mittweida

Índice

Objetivo de los ensayos	1
Sinóptico de ejercicios.....	1
Instalación experimental.....	2
Introducción.....	3
CMC 3-1 Introducción en el funciona-miento y la utilización de interrupciones.....	5
Fundamentos preliminares.....	5
Ensayo CMC 3-1A.....	9
Lista de programa CMC 3-1A.....	12
Ensayo CMC 3-1B.....	14
Lista de programa CMC 3-1B.....	18
CMC 3-2 Programación de temporizador	20
Fundamentos preliminares.....	20
Ensayo CMC 3-2A.....	25
Lista de programa CMC 3-2A.....	28
Ensayo CMC 3-2B.....	30
Lista de programa CMC 3-2B.....	32
Ensayo CMC 3-2C	34
Lista de programa CMC 3-2C	39
Ensayo CMC 3-2D	43
Lista de programa CMC 3-2D	47
Ensayo CMC 3-2E.....	51
Lista de programa CMC 3-2E.....	55
CMC 3-3 Aplicación de un convertidor analógico-digital (convertidor A/D) integrado.....	56
Fundamentos preliminares.....	56
Ensayo CMC 3-3A.....	60
Lista de programa CMC 3-3A.....	63
Ensayo CMC 3-3B.....	68
Lista de programa CMC 3-3B.....	72

Objetivo de los ensayos

1. Los aprendices / estudiantes aprenden a aplicar la realización de problemas de control de la práctica en un microcontrolador.
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 3-1: Introducción en el funcionamiento y la utilización de interrupciones

- Funcionamiento básico de interrupciones
- Vector de interrupción y tabla de vectores de interrupción
- Fuentes de interrupción del C515C
- Aceptación de una interrupción
- ¿Qué es una interrupción?
- Autorización de una interrupción
- Prioridades de interrupción
- Ejemplos de aplicación para interrupciones externas
 - Complementar un estado de LED
 - Contar y visualizar accionamientos de tecla

CMC 3-2: Programación de temporizador

- Funciones básicas del temporizador
 - Conceptos importantes y básicos
 - Componentes principales de un temporizador
 - Las funciones contador y marcador de tiempo
- Los temporizadores „on chip“ del C515C
 - Timer 0, Timer 1 y Timer 2
 - Autorecarga, Modo de comparación, Modo de captación
- Ejemplos de aplicación de temporizadores
 - Parpadeo de un LED con 2 Hz
 - Emisión de un tono de 1 KHz por medio de altavoces
 - Emisión de una melodía
 - Medición de una frecuencia
 - Generación de señales PWM

CMC 3-3: Utilización de convertidores A/D

- Propiedades del convertidor A/D „on chip“
 - Parámetros
 - Relaciones de tiempo
- Ejemplos de aplicación
 - Medir una tensión continua externa (visualización de valores medidos en display LC)
 - Medir y regular una temperatura (visualización de valores medidos en display LC)

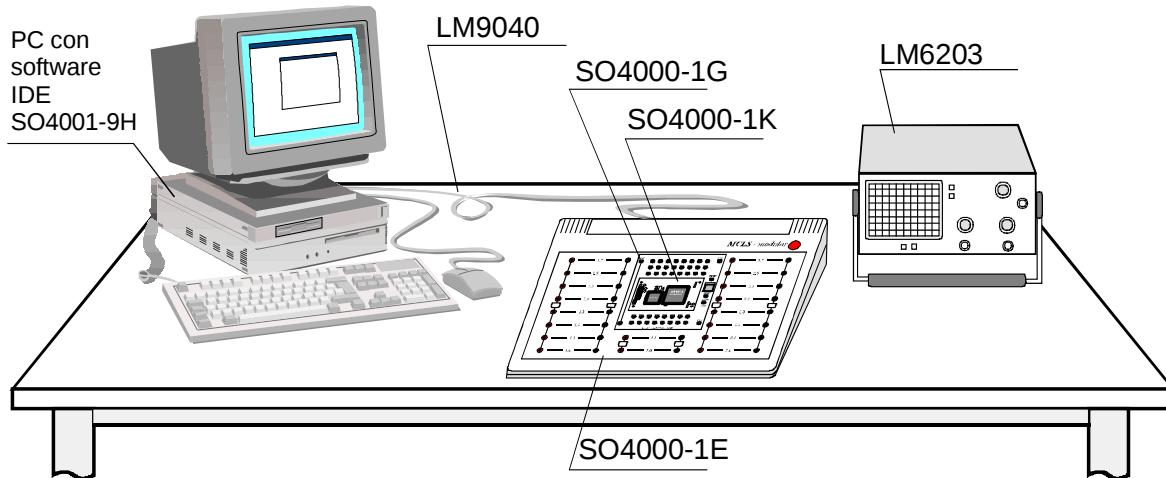


Fig. 01: Instalación en el puesto de trabajo

Instalación experimental

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.

Elementos y aparatos 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 teclas	SO4000-1Q
1	Unidad de display 1 (7 segmentos)	SO4000-1S
1	Unidad de display 2 (display LC)	SO4000-1B
1	Adaptador BNC	SO4000-1T
1	Unidad de control de temperatura	SO4000-3C
1	Unidad de tensión	SO4000-3D
1	Unidad de altavoz	SO4000-3G
1	Unidad de generador de funciones	SO4000-3H
1	Software IDE para MCLS (D,GB,F,E)	SO4001-9H
1	Osciloscopio con puntas de prueba	LM6203
1	Sonda de prueba lógica	LM8101
1	Cable de medición BNC / BNC	LM9034
1	Cable de interfase serial 9/9 polos	LM9040
5	Línea de medición 2mm 15cm azul	SO5126-5K
15	Línea medición 2mm 15cm amarillo	SO5126-5M
1	CMC 3 Programación de periféricos on-chip C515C	SH5019-1C
1	Maleta de almacenam. para MCLS	SO4000-1W

Introducción

Los microcontroladores han llegado a tener una importancia decisiva en muchos ámbitos de la vida cotidiana. Trabajan casi siempre en „segundo plano“, sin ser reconocidos, y forman la base de muchos aparatos electrónicos en el ámbito del „Embedded Control“. Los campos típicos de aplicación de los microcontroladores son:

- Electrónica de automóviles
- Técnica de medición
- Técnica de comunicación
- Técnica de medicina
- Técnica de electrodomésticos
- Electrónica recreativa

¿Qué es un microcontrolador?

Un microcontrolador es un sistema de microordenador completo en un chip. El núcleo del microcontrolador (Core), la memoria, los componentes de perifería y el sistema de interrupción están integrados juntos en un chip y conectados entre ellos por medio de uno ó varios buses. [1]

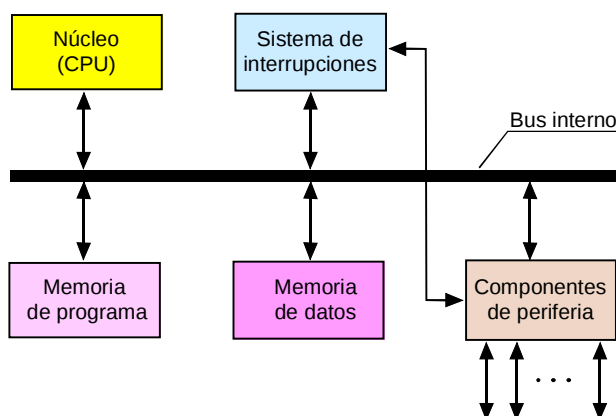


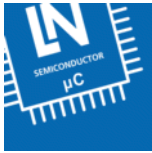
Fig. 02: Estructura básica de un microcontrolador [1]

La idea de la integración „on chip“ no se basaba en el objetivo de un alto rendimiento de procesamiento, sino en conseguir una integración funcional máxima con bajos costes de sistema. Además del núcleo (core) y los diferentes elementos de memoria, están integradas en el chip las funciones de perifería típicas de un sistema de microordenador.

¡Los componentes de perifería están „on chip“!

Ejemplos para esta perifería „on chip“:

- Puertos de entrada y salida
- Temporizadores
- Convertidores A/D
- Interfases de comunicación
- Interrupciones externas
- Temporizadores WatchDog
- Reloj de tiempo real (RealTimeClock)
- Funciones para ahorrar corriente



Puesto que casi todos los componentes „on chip“ son aptos para interrupciones, las funciones de perifera integradas son completadas con un sistema de interrupción programable.

Los microcontroladores con core idéntico y perifera „on chip“ diferente o memoria „on chip“ variada se denominan derivados o miembros de una familia MC.

En el marco de la presente serie de ensayos nos centramos en la programación, orientada en la aplicación, de componentes periféricos seleccionados de un microcontrolador que forma parte de la familia 8051 ampliamente divulgada. El microcontrolador C515C es un derivado de esta familia y está disponible en el módulo PSD1-FLASH del MCLS-modular® para los experimentos a realizar.

El manual de instrucciones del módulo PSD1-FLASH contiene muchos detalles relativos al microcontrolador empleado a los que nos referiremos en las siguientes instrucciones para los ensayos [2]. La estructura y el funcionamiento de los microcontroladores son objeto del manual de instrucciones CMC 1 [3]. Los fundamentos expuestos en este manual para la programación de microcontroladores y para la programación estructurada se consideran un requisito para comprender el presente documento.

El entorno de desarrollo de Windows integrado (IDE) que forma parte del MCLS-modular® permite, tras un periodo de adiestramiento corto, confeccionar, poner en funcionamiento y optimizar programas para un microcontrolador. En este IDE se encuentran implementadas las herramientas de desarrollo necesarias para la elaboración de programas de microcontrolador ejecutables. El trabajo inicial con este IDE y con el MCLS-modular® es facilitado por la posibilidad de la utilización inmediata de perfiles integrados ya preparados para el IDE.

Los ensayos con la perifera „on chip“ están orientados en una programación próxima a la máquina mediante el lenguaje de ensamblaje. Con ello se garantiza, incluyendo el depurador potente del módulo PSD1-FLASH, un acceso completo al hardware del microcontrolador y a los módulos de experimentación acoplados, lo que facilita la puesta en marcha de soluciones de programa propias.

Bibliografía:

[1] Beierlein, T.; Hagenbruch, O.: Taschenbuch Mikroprozessortechnik (Manual de la técnica de microprocesador), Fachbuchverlag Leipzig im Carl Hanser Verlag, 3ra. edición, 2004

[2] Manual de instrucciones del módulo PSD1-FLASH (SO4000-1G), LUCAS-NÜLLE Lehr- und Meßgeräte GmbH, 1ª edición 2004

[3] Introducción en la programación de microcontrolador (CMC 1), LUCAS-NÜLLE Lehr- und Meßgeräte GmbH, 3ra. edición, 2006

CMC 3-1 Introducción en el funciona-miento y la utilización de interrupciones

El funcionamiento de las interrupciones se presentará con el ejemplo de la activación de interrupciones externas mediante el accionamiento de teclas.

Fundamentos preliminares

Un evento externo puede interrumpir en una CPU la ejecución en curso de una secuencia de programa y conducir a la ejecución de un programa corto asociado al evento. Este evento que actúa desde fuera sobre la CPU y que casi nunca es previsible desde el punto de vista del tiempo, se denomina interrupción, y el programa asociado que contiene la función útil a realizar, rutina de servicio de interrupción (ISR).

Las fuentes de estas interrupciones pueden ser, entre otras:

- el desbordamiento de un temporizador
- el aviso de acabado de una conversión analógica-digital
- el cambio de nivel en un pin de puerto (interrupciones externas)
- la llegada completa de un carácter en el registro de desplazamientos de recepción del UART
- un evento de comparación en un temporizador

Una vez procesada la ISR, se continua con el programa anteriormente interrumpido en la correspondiente posición del programa.

Desde el punto de vista temporal, una solicitud de interrupción se desarrollará de la siguiente manera:

1. La fuente de interrupción genera una señal de solicitud de interrupción.
2. Se pone un bit de solicitud de interrupción asociado a la fuente de interrupción.
3. La lógica de consulta de interrupciones (lógica de interrupciones) detecta que se ha puesto un bit de solicitud de interrupción.
4. Se inicia un ciclo de aceptación de la interrupción de la CPU.
5. La dirección de retorno para el programa a interrumpir se salva en la pila.
6. El contador de programa (PC) de la CPU se pone en la correspondiente posición del vector de interrupción (IV) en la tabla de vectores de interrupción (IVT).
7. Después de un ciclo „Hole“ para el código de programa del IV empieza la ejecución de la ISR.

Los siguientes componentes tienen funciones clave en esta cadena de acciones:

- Bits de solicitud de interrupción
- Vector de interrupción (IV)
- Tabla de vectores de interrupción (IVT)
- Bits de activación de interrupción
- Registros de prioridad de interrupciones

De la estructura de las fuentes de interrupción del C515C (ver el anexo del manual de instrucciones del módulo PSD1-FLASH en [2]) se desprenden tanto las diferentes fuentes de interrupción como los bits de solicitud de interrupción asociados (p. ej. IE0, IEX5 y IADC).

Estos bits los contienen los registros de funciones especiales (SFR), y se pueden posicionar y también reposicionar por medio del software. Sin embargo y en el caso normal, una interrupción conducirá al posicionamiento del correspondiente bit y, con la entrada en la ISR, se realizará un reposicionamiento automático. Desde el punto de vista de las fuentes de interrupción, estos bits son p. ej. señales de desbordamiento de los temporizadores („overflow-flags“) o informaciones de disparador de las llamadas interrupciones externas („edge flags“).

En la zona inferior de direcciones de la memoria de programa se encuentra la IVT en la que se deben introducir, en las direcciones correspondientemente asociadas, los IV. Estas direcciones de vector de interrupción con la asociación a las fuentes de interrupción o los bits de solicitud de interrupción también se desprenden de [2]. En la arquitectura de 8051, ¡un IV supone un comando de salto a la ISR asociada!

Ejemplo de introducción de un IV en la IVT para el Timer 0:

```
SEGMENT CODE
ORG 000BH ; Posición del IV
JMP ISR_TIMER0 ; IV
```

Para que una interrupción lleve a la aceptación de la interrupción por parte de la CPU, es preciso autorizar la interrupción en cuestión y adicionalmente poner el indicador de habilitación de interrupción (EAL) global (después de un RESET, EAL es igual a 0).

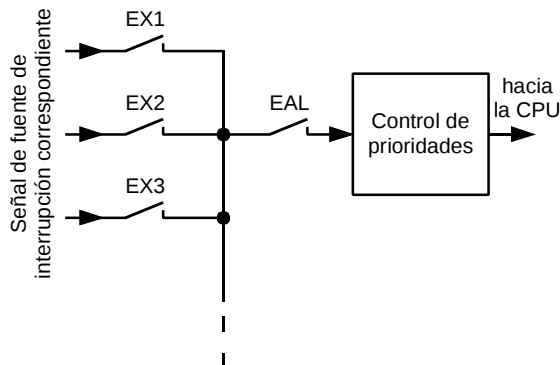


Fig. 101: Función de los bits de habilitación de interrupción („interrupt enable bits“)

Si se activan más de 1 interrupción dentro de una solución de programa, se deberán considerar los problemas de prioridad. Después de un reset, entra en vigor una prioridad base; no obstante es posible realizar, dentro de determinados límites, modificaciones de esta prioridad base por medio de los registros de funciones especiales IP0 e IP1. Véanse los detalles al respecto en la información compacta en [2].

De un modo general es válido:

¡Solamente una interrupción de prioridad más alta puede interrumpir una ISR en curso para una interrupción de prioridad más baja!

Las interrupciones de la misma prioridad quedan excluidas por la estructura de la prioridad base y por las posibilidades de asociar niveles de prioridad por medio de IP0 e IP1. Sin embargo existe la posibilidad de impedir, al principio de una ISR y por medio de EAL=0, una interrupción de la ISR por interrupciones de prioridad más alta.

¿Qué hace falta para inicializar una interrupción?

- Definir el modo de servicio del componente activador (si es necesario)
- Autorizar la interrupción por medio de una fuente de interrupción
- Permitir interrupciones de manera global en el sistema

Si durante la ejecución de un programa de aplicación se realizan modificaciones en el sistema de interrupciones (p. ej. si se pretende activar una nueva fuente de interrupción), ¡las aceptaciones de interrupciones por parte de la CPU se deberán bloquear de forma global durante el tiempo de inicialización!

¿Qué se debe tener en cuenta en la programación de una ISR?

1. ¡Todos los registros que se utilizan dentro de una ISR, se deberán guardar en la pila!
2. ¡A cada nivel de prioridad de interrupción utilizado se deberá asociar un banco de registro por separado para los registros de trabajo!
3. ¡La estructura de una ISR debería garantizar un tiempo de ejecución de programa lo más corto posible! Por este motivo se deberán evitar preferiblemente los bucles de programa en una ISR.

Ejemplo para 1. y 2.:

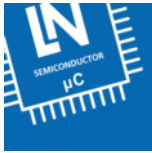
```
ISR_1: PUSH PSW
      PUSH ACC
      MOV  PSW, #REG_BANK1

      ; aquí figura la función útil

      POP  ACC
      POP  PSW
      RETI
```

Nota:

Para los procesos críticos desde el punto de vista del tiempo se deberá observar, que el tiempo de reacción de la interrupción (tiempo desde la aparición de una interrupción hasta el inicio de la ISR) puede ser de 4 a 9 ciclos de máquina, según la situación de salida. Esto solamente tiene validez si no está activada ninguna interrupción de prioridad más alta.



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 display 1 (7 segmentos)	SO4000-1S
1	Software IDE para MCLS (D,GB,F,E)	SO4001-9H
1	Cable de interfase serial 9/9 polos	LM9040
5	Línea de medición 2mm 15cm azul	SO5126-5K
12	Línea medición 2mm 15cm amarillo	SO5126-5M
1	CMC 3 Programación de periféricos on-chip C515C	SH5019-1C

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!

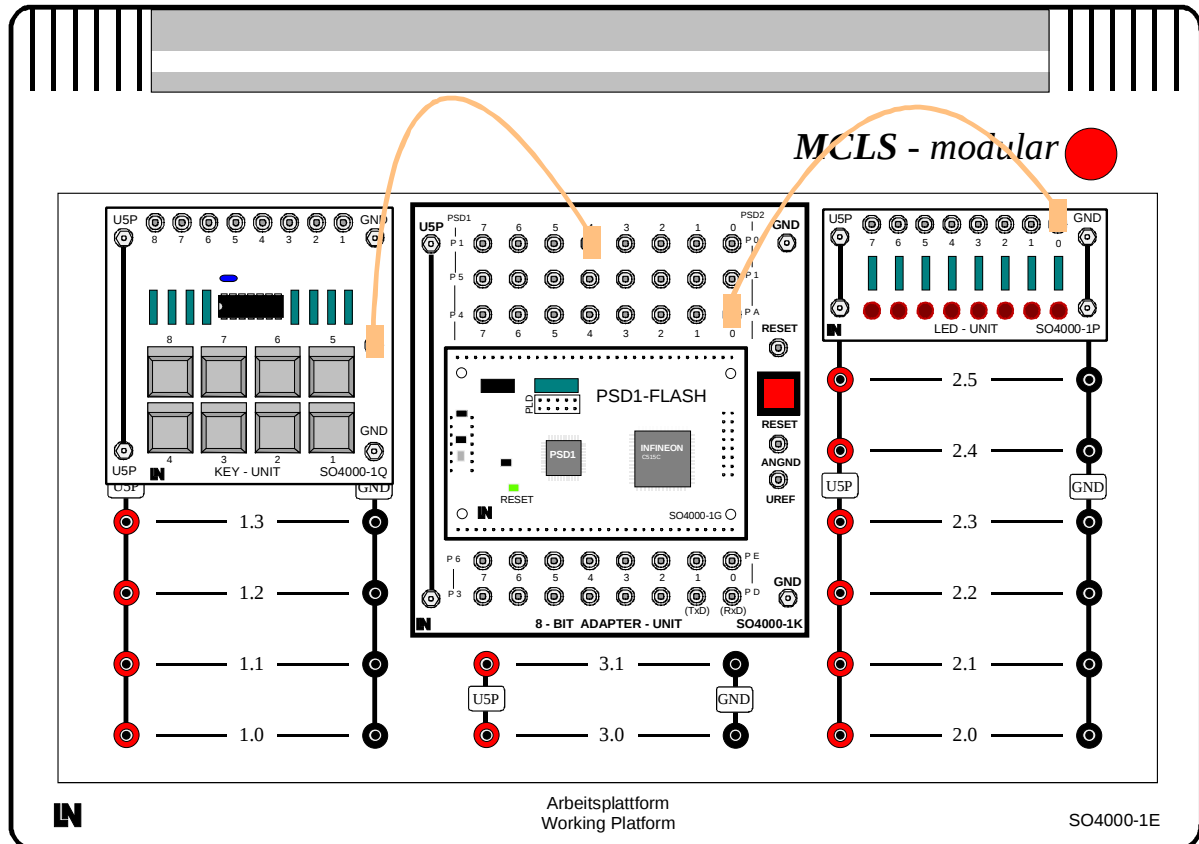


Fig. 102: Instalación de aparatos del ensayo CMC 3-1A

Ensayo CMC 3-1A

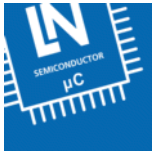
Pulsando una tecla se debe activar la interrupción externa 2 y se debe invertir el estado de encendido de un LED!

¿Qué conocimientos se pretenden transmitir?

- ¿Qué hace falta para inicializar una interrupción?
- ¿Cómo se introduce un vector de interrupción (IV) en la tabla de vectores de interrupción (IVT)?
- ¿Cuál es el aspecto de la estructura de una rutina de servicio de interrupción (ISR)?
- ¿Cómo se puede utilizar un punto de ruptura como medio de ensayo efectivo en la puesta en marcha de aplicaciones de interrupción?
- ¿Cómo se pueden evitar los efectos negativos del rebotamiento de las teclas?

Explicaciones relativas al ejercicio:

La salida INT de la unidad de teclas se conecta al correspondiente pin de entrada del microcontrolador para la interrupción externa 2. Para la visualización sencilla de la función de interrupción se conectará un LED de la unidad de LED a un pin del Puerto 4 y se activará correspondientemente dentro de la rutina de servicio de interrupción.



Ejercicios y soluciones

¡Realice la instalación experimental según la fig. 102!

Croquice un diagrama de bloques sencillo, del que se desprenda la asociación concreta de los pins del microcontrolador a la salida INT de la unidad de teclas y de la unidad de LED. Complete el croquis con la indicación del flanco que se genera mediante el accionamiento de una tecla.

Croquice la estructura de programa para el programa principal y la rutina de servicio de interrupción en forma de un organigrama del programa.

Elabore el texto fuente de ensamblador sobre la base del organigrama del programa. Observe el texto fuente del programa ensamblador cmc31a.asm.

Ensaye el programa traducido con la ayuda del depurador en servicio a pasos de proceso o en marcha de tiempo real en caso de utilizar un punto de ruptura en el IV.

1. Muévase en servicio a pasos de proceso (F8) al bucle sinfín del programa principal.
2. Coloque un punto de ruptura en la posición del IV en la IVT (dirección 004BH).
3. Inicie el servicio en tiempo real del programa (F9).
4. Accione una tecla cualquiera de la unidad de teclas.
5. En caso de una instalación experimental reglamentaria y una inicialización correcta de la interrupción se alcanza, mediante el accionamiento de la tecla, la posición del IV.
6. Continúe la ejecución del programa a pasos individuales hasta la función útil (CPL P4.0).

Ahora se pueden repetir las acciones 3 a 6.

Demuestre, en servicio de tiempo real, el rebotamiento de las teclas.

1. Elimine el punto de ruptura acordado.
2. Inicie el programa en tiempo real.
3. Accione las teclas de la unidad de teclas.

La inversión no consecuente del estado de encendido de los LED es un indicio para la existencia de rebotamientos de teclas.

Inmediatamente después de salir de la ISR (tiempo de ejecución 6µs, aproximadamente), se genera, debido a un rebotamiento de tecla posterior (según la tecla en el margen de hasta algunos ms), una nueva solicitud de interrupción y se activa otra activación no deseada de la ISR.

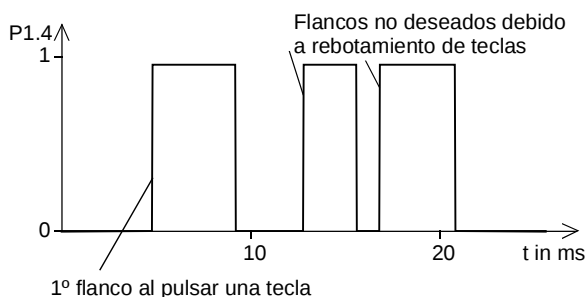
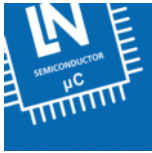


Fig. 103: Representación del rebotamiento de teclas

Complete la ISR de tal forma que el rebotamiento de las teclas no dé lugar a interrupciones no deseadas.

A ese fin se deben activar, en el texto fuente cmc31a.asm, las líneas de comando en ISR_INT2.

Varíe el retardo TIEMPO mediante diferentes constantes para la carga de R7. Observe que la ISR no debe presentar un tiempo de ejecución innecesariamente largo.



Lista de programa CMC 3-1A

```
; *****
; * Proyecto:          Interrupción externa 2 pulsando una tecla *
; * Archivo fuente:    cmc31a.asm                                *
; * Herramientas MC:   as, p2hex, debugger                        *
; * Perfil:            Perfil integrado PSD1-FLASH-C515C-AS       *
; * Recursos MC:       LED en P4.0,                               *
; *                   fuente de interrupción en P1.4 (int. ext.2) *
; * Editor:            HAG                                         *
; *****

; ----- Descripción de funcionamiento -----
; La salida de interrupción de la unidad de teclas está conectada
; a P1.4 la entrada para la interrupción externa 2. Un LED de la
; unidad de LED está conectada a P4.0 para la visualización de la
; interrupción mediante un accionamiento de tecla.
; -----

; -----Acuerdos necesarios para el ensamblador-----
CPU      80515          ; Selección de microcontrolador
INCLUDE  C515C.INC      ; Definiciones SFR

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

; -----
;          jmp      INICIO          ; Salto al programa principal

; ----- Introducción del IV en la IVT -----
ORG      004BH          ; Posición de IV para
                        ; la interrupción ext. 2
JMP      ISR_INT2       ; Vector de interrupción (IV)

; *****
; * Programa principal "Interrupción ext. 2 pulsando una tecla *
; *****
INICIO:
CALL     INIT_INT2      ; Inicializar interrupción ext. 2
JMP      $              ; Bucle sinfín como "final"
                        ; del programa principal
; *****

; -----
; Subprograma para la inicialización de la interrupción ext. 2
; -----
INIT_INT2:
CLR      EAL            ; Bloquear globalmente las aceptaciones
                        ; de interrupción
SETB     I2FR           ; Flanco 0-1 activa la interrupción
SETB     EX2            ; Autorizar interrupción por int. ext. 2
SETB     EAL            ; Autorizar globalmente las aceptaciones de
interr.
RET
```

```
; -----  
;                               ISR para interrupción ext. 2  
; -----  
ISR_INT2:  
    CPL    P4.0    ; Con accionam. de tecla invertir estado  
                ; de LED  
;    CALL   TIEMPO ; Tiempo para rebotamiento de tecla  
;    CLR    IEX2   ; Reposicionar bit de solicitud de interrupción  
    RETI  
  
; -----  
;    Tiempo para anular rebotamiento, aprox., 25ms  
; -----  
TIEMPO:  
    MOV     R7,#49  
T01:    MOV     R6,#000H  
T00:    DJNZ    R6,T00  
        DJNZ    R7,T01  
    RET  
  
END
```

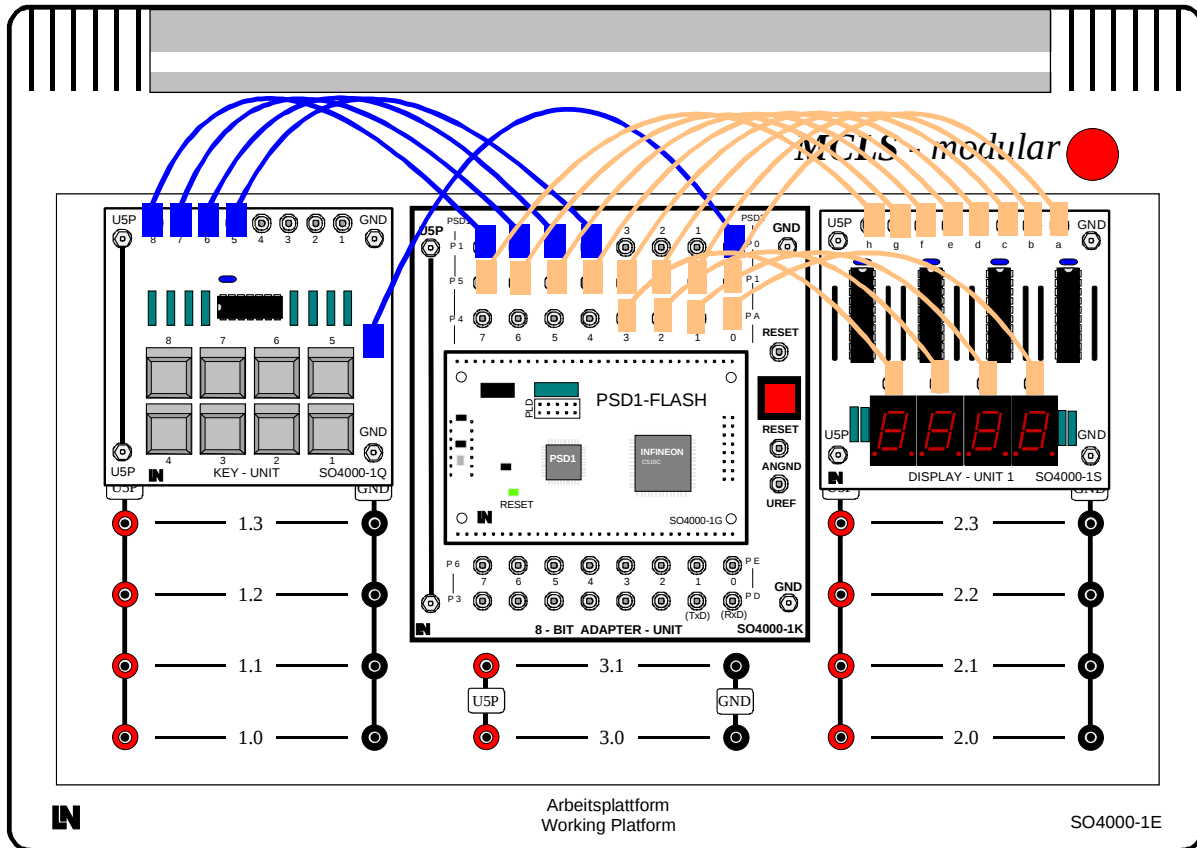


Fig. 104: Instalación de aparatos del ensayo CMC 3-1B

Ensayo CMC 3-1B

El número de accionamientos de teclas por medio de las teclas 5 a 8 de la unidad de teclas se debe registrar y visualizar por medio de la unidad de display 1. Para reconocer un accionamiento de tecla se utilizará la interrupción externa 4.

¿Qué conocimientos se pretenden transmitir?

- ¿Cómo se inicializa la interrupción ext. 4?
- ¿Cómo se introduce el IV para la interrupción ext. 4 en la IVT?
- ¿Cuál es el aspecto de la estructura de la ISR según la función útil a realizar?
- ¿Por qué se deben completar las medidas de anulación de rebotamientos del ensayo cmc31A?
- ¿Cómo se puede activar de manera sencilla la unidad de display 1?

Explicaciones relativas al ejercicio:

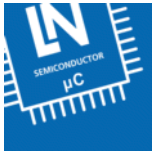
La salida INT de la unidad de teclas se conecta al correspondiente pin de entrada del microcontrolador para la interrupción externa 4. Adicionalmente se conectan las salidas lógicas de las teclas 5 a 8 a los pins P1.4 a P1.7 (tétrada superior del Puerto 1). Para visualizar el número de accionamientos de teclas se emplea la unidad de display 1. Los subprogramas para la activación del módulo de visualización incluyendo las indicaciones para la conexión a los Puertos 5 y 4 del microcontrolador se encuentran archivados en el fichero de inclusión FCT7SEGM.INC.

Ejercicios y soluciones

Realice la instalación experimental según la fig.104.

Croquice un diagrama de bloques del que se desprenda la asociación de los pins del microcontrolador a la unidad de TECLAS (salida INT, salidas de teclas 5 a 8) y a la unidad de display 1. Complete el croquis indicando el flanco que se genera a través del accionamiento de una tecla.

Croquice la estructura de programa para el programa principal y la de la rutina de servicio de interrupción en forma de un organigrama de programa.



Escriba el texto fuente de ensamblador sobre la base de un organigrama de programa. Observe el texto fuente del programa ensamblador cmc31b.asm.

Ensaye el programa traducido con la ayuda del depurador en servicio a pasos de proceso o en ejecución en tiempo real empleando un punto de ruptura en el IV.

1. Muévase en servicio a pasos de proceso (F8) hasta el bucle sinfín del programa principal.
2. Coloque un punto de ruptura en la posición del IV en la IVT (dirección 005BH).
3. Inicie la ejecución en tiempo real del programa (tecla F9 en el depurador).
4. Accione una tecla cualquiera de la unidad de teclas en la fila superior de teclas.
5. En caso de una instalación experimental reglamentaria y una inicialización correcta de la interrupción se alcanza, con el accionamiento de tecla, la posición del IV.
6. Continúe la ejecución del programa a pasos individuales en la ISR y compruebe la funcionalidad manteniendo pulsada la tecla.

Ahora, las acciones 3 a 6 se pueden repetir.

¿Por qué se debe instalar un retardo al principio de la ISR, antes de introducir el código de teclas?

Familiarícese con la función básica del subprograma para la activación de la unidad de display 1 en el archivo FCT7SEGM.INC. Consulte, para ello, el manual de instrucciones de la unidad de display 1.

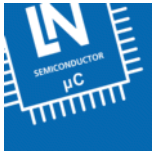
Del estudio de los textos fuente en FCT7SEGM.INC se desprende la activación fundamental de las indicaciones de segmentos:

1. Preparación de la cifra de 4 dígitos a representar por medio de las tétradas de DPTR.
2. Mostrar y tomar el código para una posición de indicación de la tabla de códigos.
3. Preparación del código de cifras para una posición de indicación en las conexiones de segmento a - h de la unidad de display 1 por medio del Puerto 5.
4. Pone el código de cifras en el correspondiente enclavamiento de la posición de indicación con un impulso 1 corto (bit individual de Puerto 4).

El contado y la visualización del número de accionamientos de tecla se realiza, de acuerdo con el siguiente texto fuente de programa, de manera hexadecimal.

```
INC_DPTR_DC:
    PUSH ACC          ; salvar acumulador
    MOV A,DPL         ; coger LT de DPTR
    ADD A,#1          ; +1
    DA A              ; ajuste decimal
    MOV DPL,A         ; recargar LT
    JNC IN0           ; si CF=0, al final
    MOV A,DPH         ; coger HT de DPTR
    ADD A,#1          ; +1
    DA A              ; ajuste decimal
    MOV DPH,A         ; recargar HT
IN0:
    POP ACC           ; restablecer acum.
    RET               ; comando de retorno
```

¿Cómo se puede realizar el contado y la visualización de los accionamientos de tecla de forma decimal?



Lista de programa CMC 3-1B

```
; *****
; * Proyecto:          Contar accionamientos de teclas          *
; * Archivo fuente:    cmc31b.asm                                *
; * Herramientas MC:as, p2hex, debugger                          *
; * Perfil:            Perfil integrado PSD1-FLASH-C515C-AS      *
; * Recursos MC:       Unidad de display 1 en P5 y P4.0 a P4.3   *
; *                   INT de la unidad de teclas en P1.1 (int. ext.4) *
; *                   y 5 a 8 en P1.4 a P1.7                      *
; * Editor:           HAG                                        *
; *****

; ----- Descripción de funcionamiento -----
; Se deben registrar y visualizar accionamientos de teclas de la
; unidad de teclas (fila superior de teclas).
; Para ello la salida de interrupción de la unidad de teclas está
; conectada a P1.1, la entrada para la interrupción externa 4. Puesto
; que solamente se utiliza la fila superior de teclas, se pueden
; conectar las salidas 5 a 8 a los pins de puerto P1.4 a P1.7.
; Para visualizar los accionamientos de teclas a contar sirve la
; unidad de display. La activación de los segmentos se realiza por
; medio del Puerto 5, la selección de posiciones por medio de la
; tétrada inferior del Puerto 4.
; -----

; -----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

; ----- Introducir el IV en la IVT -----
ORG      005BH      ; Posición IV para la int. ext. 4
JMP      ISR_INT4    ; Vector de interrupción (IV)
; -----

ORG      100H        ; Programa principal a partir de 100H
; *****
; * Programa principal "Contar accionamientos de tecla" *
; *****
START:
MOV      DPTR,#0000H ; Inicializar DPTR como registro
; contador
CALL     OUT_DPTR_7SEGM ; Contenido de DPTR en indicador
CALL     INIT_INT4     ; Inicializar interrupción ext. 4
JMP      $             ; Bucle sinfín como "final"
; del programa principal
; *****

; -----
; Subprograma para la inicialización de la interrupción ext. 4
; -----
INIT_INT4:
CLR      EAL          ; Bloquear globalmente las aceptaciones de int.
SETB     EX4          ; Autorizar interrupción por medio de int. ext. 4
SETB     EAL          ; Autorizar globalmente aceptación de int.
RET

; -----
; ISR para interrupción ext. 4
; -----
```

```

; -----
ISR_INT4:
    CALL TIEMPO ; Esperar hasta que código de teclas estable
;    CALL TIEMPO ; Necesario en caso de tiempo largo
;                ; de rebotamiento
    MOV A,P1 ; Introducir código de teclas
    ANL A,#11110000B ; Borrar LT
; .....Controlar código de teclas.....
    CJNE A,#11100000B,IS0 ; ¿Tecla 5?, si no, a IS0
    JMP IS4 ; en caso de sí, +1
IS0:
    CJNE A,#11010000B,IS1 ; ¿Tecla 6?, si no, a IS1
    JMP IS4 ; en caso de sí, +1
IS1:
    CJNE A,#10110000B,IS2 ; ¿Tecla 7?, si no, a IS2
    JMP IS4 ; en caso de sí, +1
IS2:
    CJNE A,#01110000B,IS3 ; ¿Tecla 8?, si no, a IS3
IS4:
    INC DPTR ; Registro contador +1
    CALL OUT_DPTR_7SEGM ; Contenido de DPTR en indicación
IS3:
    CLR IEX4 ; Reposicionar bit de solicitud de interrupción
    RETI

; -----
; Tiempo de anulación de rebotamiento, aprox., 25ms
; -----
TIEMPO:
    MOV R7,#49
T01:
    MOV R6,#000H
T00:
    DJNZ R6,T00
    DJNZ R7,T01
    RET

; -----

; Insertar funciones de indicación para indicación de 7 segmentos
    INCLUDE FCT7SEGM.INC ; Archivo de inclusión
;                          ; en el mismo directorio

END

```

CMC 3-2 Programación de temporizador

Con las aplicaciones de emisión de melodía, medición de frecuencia y generación de señales PWM se presenta la programación de funciones de temporizador.

Fundamentos preliminares

Los temporizadores forman parte de los componentes „on chip“ más importantes de un microcontrolador. Casi cada aplicación necesita, como mínimo, una función de temporizador, de forma que los microcontroladores a menudo tienen instalados varios temporizadores „on chip“.

Dentro de una estructura de temporizador, el componente principal está formado por un registro contador (RC). Según el temporizador, este RC tiene diferentes volúmenes de contado – son habituales anchuras de contado de 8, 16 y 24 bits. La entrada de contado de un RC se puede activar por medio de hardware y/o software. Un bit especial, que casi siempre es al mismo tiempo el bit de solicitud de interrupción, informa del desbordamiento del RC. Además de los RC que actúan de forma incremental, los hay de actuación decremental, siendo señalizado, en este último caso, un sub-desbordamiento por medio de un bit.

Un temporizador dispone, por regla general, de dos funciones base. Si la entrada de contado del RC está conectada a una señal externa, se habla de una función de contador (Counter). Si la entrada de contado es alimentada, en cambio, por el reloj de sistema dividido, el temporizador actúa como marcador de tiempo (Timer). Partiendo de estas funciones principales, se conocen otros modos de servicio, como p.ej. los modos de comparación y de captación, con diferentes funciones de control.

Los 3 temporizadores del SAB 80C515/C515C (Timer 0, Timer 1 y Timer 2) disponen de las siguientes propiedades fundamentales (ver figuras en [2]):

- Los RC tienen una anchura de 16 bits y trabajan de modo incremental.
- El acceso a los RC se puede realizar por medio de los SFR THx o TLx de 8 bits de manera secuencial (leyendo y escribiendo).
- En caso de desbordamiento del RC se pone un indicador de desbordamiento (TFx).
- El desbordamiento se puede utilizar para la generación de interrupciones.
- El proceso de contado se puede interrumpir por medio de señales de hardware y/o software.
- Como modos de servicio principales programables están disponibles las funciones Contador (Counter) y Marcador de tiempo (Timer).

Características del modo de marcador de tiempo (Timer)

- El RC se incrementa según el reloj dividido de oscilador del MC.
- Timer 0 y 1 dividen el reloj de oscilador por 6, de forma que en una frecuencia de oscilador de 6 MHz es preciso 1µs para un incremento. De ello resulta un tiempo máximo de 65536 µs para un periodo de contado.
- Para el Timer 2 se dispone adicionalmente de 1/12 del reloj de oscilador.
- Precargando el RC al principio del periodo de contado, el usuario puede determinar el tiempo hasta el desbordamiento del RC.

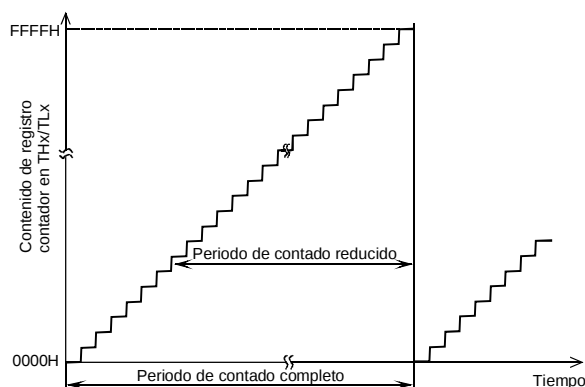


Fig. 201: Periodo de contado de un registro contador de 16 bits

Características del modo de contado (Counter)

- El RC se incrementa a través de un cambio de nivel de 1-0 en la correspondiente entrada de contado.
- La frecuencia de contado más alta posible es 1/12 de la frecuencia de oscilador, porque se necesitan, como mínimo, 2 ciclos de máquina para reconocer un cambio de nivel.
- La relación de exploración de la señal de entrada no es importante mientras que cada nivel perdure como mínimo 1 ciclo de máquina.

Timer 0 y Timer 1 presentan una estructura casi idéntica y se pueden utilizar en los siguientes modos de servicio:

- Timer/Counter de 13 bits (modo 0)
- Timer/Counter de 16 bits (modo 1)
- Timer/Count. de 8 bits con autorecarga (modo 2)
- Dos Timer/Counter de 8 bits (modo 3)

El Timer 2 dispone de una funcionalidad muy ampliada, por medio de la cual se tiene acceso a los siguientes modos de servicio:

- Timer/Counter de 16 bits con y sin autorecarga
- Modos de comparación 0 y 1
- Modo de capturación

Para realizar estos modos de servicio, el Timer 2 está equipado, en comparación con los Timer 0 y 1, adicionalmente con cuatro registros de 16 bits, un circuito de recarga, cuatro circuitos de comparación y cuatro circuitos de capturación.

Definiciones de conceptos

Recarga

Después de un desbordamiento del RC, el temporizador no sigue funcionando con 0000H, sino con el valor de carga posterior archivado en el registro de recarga. Modificando el valor de recarga se puede variar el periodo de temporizador.

Comparación

El contenido del RC se compara continuamente con el contenido del registro de comparación asociado. Si ambos coinciden, se pueden activar cambios de nivel en las salidas de comparación e interrupciones.

Capturación

Como consecuencia de un determinado evento, el contenido del temporizador en curso se copia en el registro de captación. El contenido archivado del registro de captación puede servir para la determinación de fechas de eventos. Los eventos que conducen a la captación pueden ser cambios de nivel externos en los pins de puerto asociados.

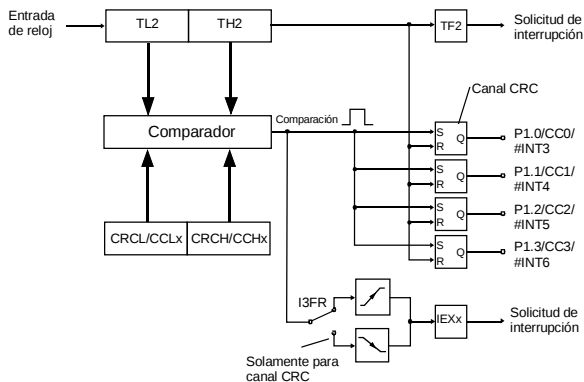


Fig. 202: Timer 2 en modo de comparación 0

Mediante el modo de comparación se pueden generar señales PWM, determinando el contenido del registro de comparación la relación de exploración, y el contenido del registro de autorecarga la duración de periodo de la señal. La generación de oscilaciones se desarrolla, una vez inicializado el temporizador, sin participación de la CPU. Solamente si cambian los parámetros de PWM, se deben volver a cargar los registros de recarga y de comparación.

Para el Timer 2, la emisión de señales PWM puede realizarse paralelamente para varios canales, coincidiendo las frecuencias de las señales de salida, y siendo posible ajustar diferentes relaciones de exploración para los canales.

Asociación de los recursos

Registro 16 bits	Pin de puerto	Modo posible
CRCH/CRCL	P1.0/#INT3/CC0	Modo comprar. 0 ó 1, captur., recarga
CCH1/CCL1	P1.1/INT4/CC1	Modo comprar. 0 ó 1, captación
CCH2/CCL2	P1.2/INT5/CC2	Modo comprar. 0 ó 1, captación
CCH3/CCL3	P1.3/INT6/CC3	Modo comprar. 0 ó 1, captación

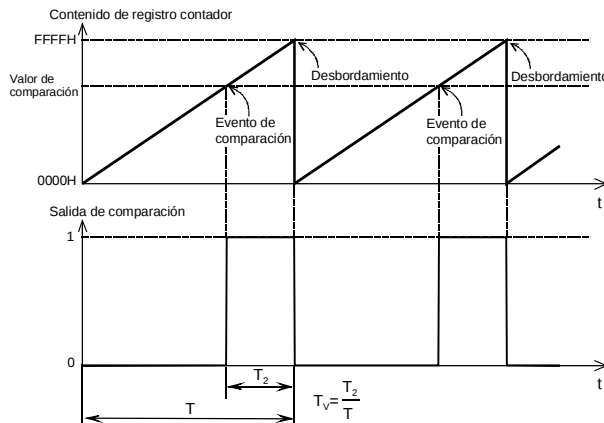


Fig. 203: Curso en el modo de comparación 0

Indicaciones relativas a la programación de los temporizadores

SFR participantes para Timer 0 y Timer 1

- TCON (registro de control de temporizador)
- TMOD (registro de modo de temporizador)
- TH0/TL0 o TH1/TL1 (registro contador)
- IEN0 (registro de activación de interr. 0)

Nota: ¡Los indicadores de desbordamiento TF0 y TF1 (en TCON) se reposicionan automáticamente por el hardware del temporizador!

SFR para Timer 2 (según modo de servicio)

- T2CON (registro de control de Timer 2)
- CCEN (registro de activ. de comprar.-captur.)
- CRCH/CRCL (registro para comprar., recarga o captur. en el canal P1.0/CC0)
- CCH1/CCL1 (registro para comprar. y captur. en el canal P1.1/CC1)
- CCH2/CCL2 (registro para comprar. y captur. en el canal P1.2/CC2)
- CCH3/CCL3 (registro para comprar. y captur. en el canal P1.3/CC3)
- IRCON (registro de control de interrupción)
- IEN0 (registro de activación de interr. 0)
- IEN1 (registro de activación de interr. 1)
- TH2/TL2 (registro contador)

Nota importante:

¡El indicador de desbordamiento TF2 (TF2 está en IRCON) se debe reposicionar por medio del software en la ISR!

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 display 1 (7 segmentos)	SO4000-1S
1	Adaptador BNC	SO4000-1T
1	Unidad de tensión	SO4000-3D
1	Unidad de altavoz	SO4000-3G
1	Unidad de generador de funciones	SO4000-3H
1	Software IDE para MCLS (D,GB,F,E)	SO4001-9H
1	Cable de interfase serial 9/9 polos	LM9040
3	Cable de medición 2mm 15cm azul	SO5126-5K
12	Cable medición 2mm 15cm amarillo	SO5126-5M
1	CMC 3 Programación de periféricos on-chip C515C	SH5019-1C

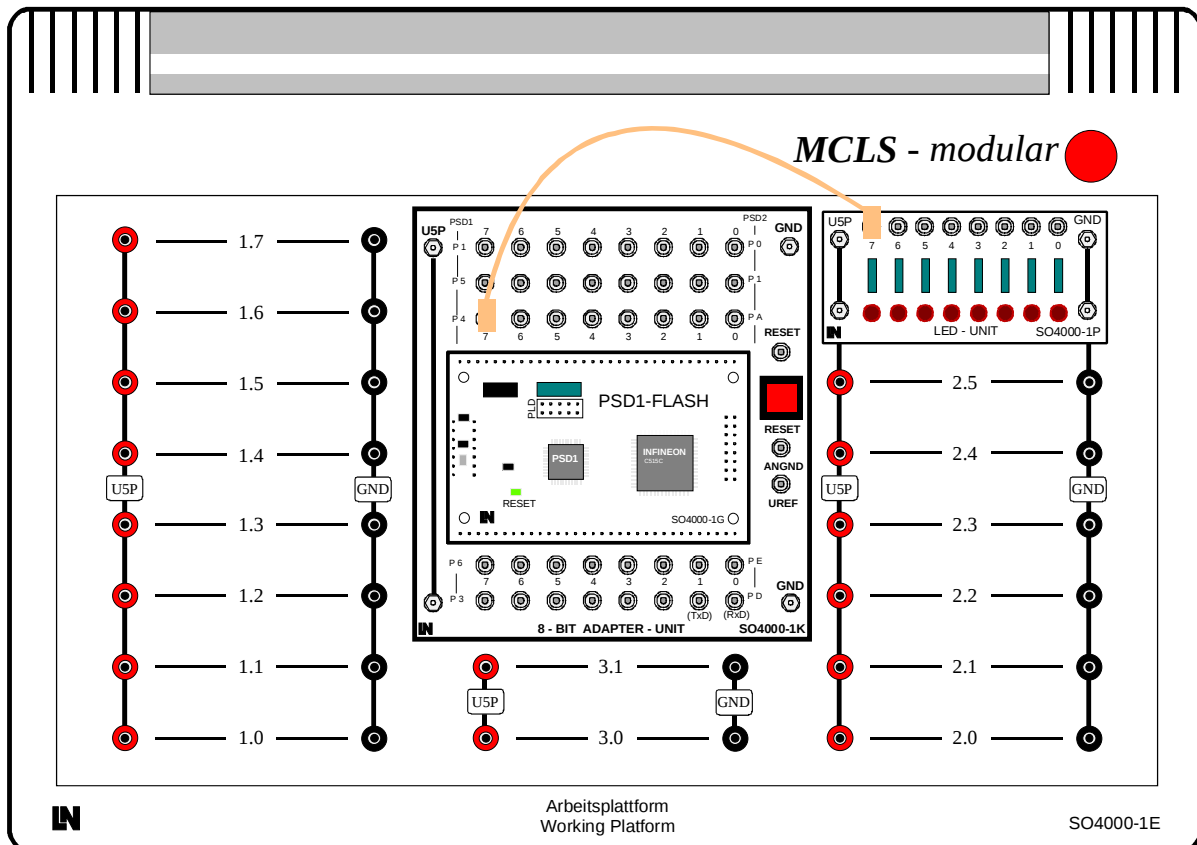


Fig.: 204: Instalación de aparatos del ensayo CMC 3-2A

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!

Indicación relativa a la literatura

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.

Ensayo CMC 3-2A

Programe un LED que parpadée con 2 Hz, debiendo el Timer 0 iniciar cíclicamente la correspondiente interrupción para la función útil.

Ejercicio:

¿Qué conocimientos se pretenden transmitir?

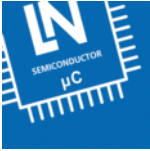
- ¿Qué m. de serv. se debe ajustar para Timer 0?
- ¿Cómo se puede realizar la frecuencia de 2 Hz (duración de periodo = 500ms)?
- ¿Dónde está el IV para el Timer 0?
- ¿Cómo se realiza la inicialización de Timer 0?
- ¿Cómo se puede realizar la recarga por medio del software?
- ¿Cuál es el aspecto de la ISR?

Explicaciones relativas al ejercicio:

Un LED de la unidad de LED se conecta a un pin de puerto (p.ej. P4.7).

Un problema del ejercicio reside en que la duración máxima de periodo de un temporizador se sitúa en 65536 µs, pero que se piden 250 ms por estado de encendido de LED (duración de periodo = 500 ms).

Otro problema supone el hecho de que para el Timer 0 no se dispone de autorecarga con una anchura de 16 bits.



Ejercicios y soluciones

Realice la instalación experimental según la fig.204.

Croquice la estructura de programa para el programa principal y la ISR.

Escriba el texto fuente de ensamblador para la inicialización del Timer 0. Parta de un periodo de contado de 50 ms para el temporizador. Un registro auxiliar se precarga con 5 para la realización del tiempo de periodo necesario. Observe el texto fuente del programa ensamblador cmc32a.asm.

Croquice el desarrollo del programa para la ISR.

Escriba el texto fuente de ensamblador para la ISR. Por medio de un registro auxiliar decremental la función útil se realiza solamente después de cada 5ª interrupción.

Observe el texto fuente del programa ensamblador cmc32a.asm.

Al principio de la ISR se vuelve a ajustar el valor de precarga en el registro contador del temporizador (RC).

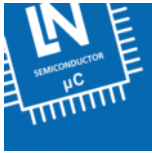
¿Por qué se detiene el temporizador durante la ejecución de los comandos de carga?

Ensaye la función del programa y mida la duración de periodo

La función de interrupción se puede ensayar de manera muy sencilla con un punto de ruptura apropiado.

Controle la duración de periodo realizada con un osciloscopio.

Modifique la frecuencia de parpadeo para el LED a 10 Hz.



Lista de programa CMC 3-2A

```
; *****
; * Proyecto: LED intermitente mediante función de marcador de tiempo*
; *           frecuencia de parpadeo 2 Hz                               *
; * Archivo fuente: cmc32a.asm                                           *
; * Herramientas MC:as, p2hex, debugger                                 *
; * Perfil:      Perfil integrado PSD1-FLASH-C515C-AS                   *
; * Recursos MC: un LED de la unidad de LED en P4.7                     *
; * Editor:      HAG                                                    *
; *****

; ----- Descripción de funcionamiento -----
; Timer 0 genera una interrupción de temporizador con una duración de
; periodo de 50 ms.
; Para obtener una frecuencia de parpadeo de LED de 2 Hz se
; decrementa, en la ISR, un registro auxiliar (R7) correspondientemente
; precargado.
; -----

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

CPU      80515          ; Selección de microcontrolador
INCLUDE  C515C.INC      ; Definiciones de SFR
INCLUDE  BITFUNCS.INC   ; Inserción de diversas funciones

; Definir constante de tiempo para registro contador de Timer 0
TCO EQU  -50000         ; 50000 * lus hasta desbordamiento

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 en la IVT -----
ORG      000BH          ; Posición de IV para desbordamiento de Timer0
JMP      ISR_T0          ; Vector de interrupción (IV)
; -----

ORG      100H           ; Programa principal a partir de 100H

; *****
; * Temporizador como marcador de tiempo para LED intermitente *
; *****
START:
MOV      R7,#05H        ; Precargar R7 como contador auxiliar
CALL     INIT_T0         ; Inicializar Timer 0
JMP      $              ; Bucle sin fin como "final"
; del programa principal
; *****
```

```

; -----
;   Subprograma para la inicialización de Timer 0
; -----
INIT_T0:
    CLR      EAL                ; Bloquear globalmente las aceptaciones
                                ; de interrupción
    MOV      TH0,#hi(TCO)
    MOV      TL0,#lo(TCO)
    ANL      TMOD,#11110000B
    ORL      TMOD,#00000001B ; Timer 0 como temporizador de 16 bits
    SETB     ET0                ; Interrupción en caso de desbordamiento
    SETB     TR0                ; Poner un Run Bit
    SETB     EAL                ; Autorizar globalmente las aceptaciones
                                ; de interrupción
    RET

; -----
;   ISR para Timer 0
; -----
ISR_T0:
; ....Recarga mediante software.....
    CLR      TR0
    MOV      TH0,#hi(TCO)
    MOV      TL0,#lo(TCO)
    SETB     TR0
    DJNZ     R7,IS0             ; ¿Registro auxiliar -1, =0?
    CPL      P4.7               ; Activar función útil : LED
    MOV      R7,#05H           ; Precargar de nuevo registro auxiliar
IS0:      RETI
; -----

END

```

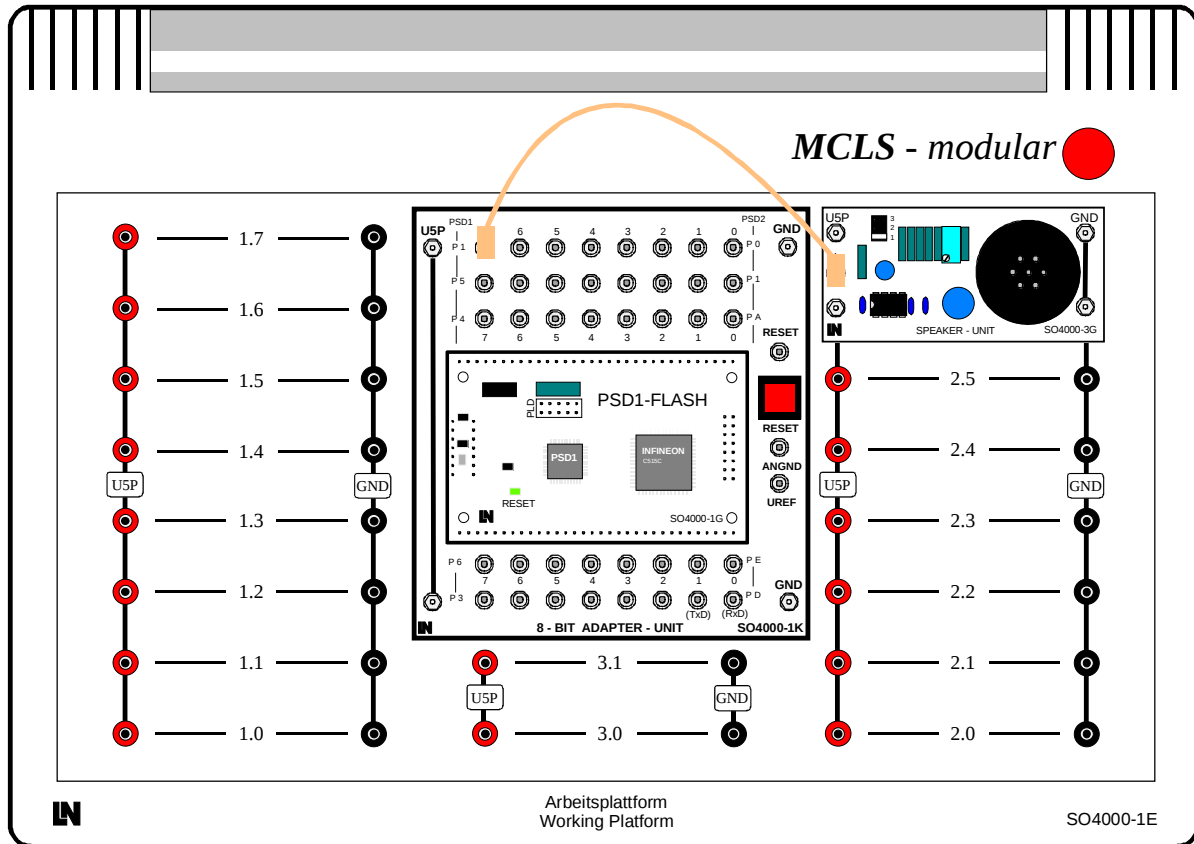


Fig.: 205: Instalación de aparatos del ensayo CMC 3-2B

Ensayo CMC 3-2B

Programe el Timer 2 como temporizador para generar un tono de 1 kHz. Envíalo por medio del pin de puerto P1.7 a la unidad de altavoz.

¿Qué conocimientos se pretenden transmitir?

- ¿Qué constante de tiempo se debe cargar en el registro contador o en el registro de recarga?
- ¿Cómo se debe inicializar el Timer 2?
- ¿Qué se debe observar particularmente en la programación de Timer 2?

Explicaciones relativas al ejercicio:

La unidad de altavoz se conecta al pin de puerto P1.7.

Para Timer 2 se dispone de un modo de recarga de 16 bits, de forma que la programación de Timer 2 como marcador de tiempo se hace muy sencilla.

Ejercicios y soluciones

Realice la instalación experimental según la fig.205.

Escriba el texto fuente de ensamblador para la inicialización del Timer 2. Familiarícese con la notación de los registros de recarga para el Timer 2.

Observe los comentarios en el texto fuente cmc32b.asm.

¿Qué constante de tiempo se debe seleccionar para una frecuencia de emisión de 1kHz?

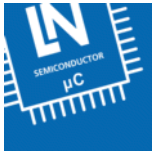
¿Por qué no es urgentemente necesaria la carga de los RC TH2/TL2 en la inicialización de Timer 2 con modo de recarga?

Escriba el texto fuente de ensamblador para la ISR. Observe el tratamiento especial del indicador de desbordamiento para Timer 2.

Observe los comentarios en el texto fuente cmc32b.asm.

Ensaye el programa y varíe la altura del tono generado modificando la constante de tiempo!

Asigne a la constante de tiempo sucesivamente diferentes constantes como valores de recarga y ensaye sus efectos.



Lista de programa CMC 3-2B

```
; *****
; * Proyecto: Emisión de tono de 1 kHz mediante marcador de tiempo *
; * Timer 2 como marcador de tiempo con recarga *
; * Archivo fuente: cmc32b.asm *
; * Herramientas MC: as, p2hex, debugger *
; * Perfil: Perfil integrado PSD1-FLASH-C515C-AS *
; * Recursos MC: Unidad de altavoz en P1.7 *
; * Editor: HAG *
; *****

; ----- Descripción de funcionamiento -----
; Timer 2 genera una interrupción de temporizador con una duración de
; periodo de 500 us, utilizándose la función de recarga.
; En la ISR se conmuta el pin de puerto P1.7 para la activación de la
; unidad de altavoz.
; -----

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

CPU      80515          ; Selección de microcontrolador
INCLUDE  C515C.INC      ; Definiciones SFR
INCLUDE  BITFUNCS.INC   ; Inserción de diversas funciones

; Definir constante de tiempo para registro contador de Timer 0
TCO EQU -500           ; 500 * lus hasta desbordamiento

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

JMP START              ; Salto al programa principal

; ----- Introducción del IV en la IVT -----
ORG 0002BH             ; Posición de IV para desbordamiento de Timer 2
JMP ISR_T2             ; Vector de interrupción (IV)
; -----

ORG 100H               ; Programa principal a partir de 100H

; *****
; * Temporizador como marcador de tiempo para LED intermitente *
; *****
START:
CALL INIT_T2           ; Inicializar Timer 2
JMP $                  ; Bucle sinfín como "final"
; del programa principal
; *****
```

```

; -----
;   Subprograma para la inicialización de Timer 2
; -----
INIT_T2:
    CLR      EAL                ; Bloquear globalmente las aceptaciones
de interrupción
    MOV      TH2,#hi(TCO)      ; Cargar RC
    MOV      TL2,#lo(TCO)
    MOV      CRCH,#hi(TCO)    ; Cargar registro de autorecarga
    MOV      CRCL,#lo(TCO)
    ANL      T2CON,#01100000B
    ORL      T2CON,#00010000B ; Autorecarga, reloj/12, temporizador
                                ; parado
    SETB     ET2                ; Autorizar interrupciones debidos a
                                ; desbordamiento
    SETB     T2I0               ; Modo de marcador de tiempo e inicio
    SETB     EAL                ; Autorizar globalmente aceptaciones
                                ; de interrupción
    RET

; -----
;                               ISR para Timer 2
; -----
ISR_T2:
    CLR      TF2                ; Reposicionar bit de solicitud
    CPL      P1.7               ; Función útil: Activar altavoz
    RETI

; -----

END

```

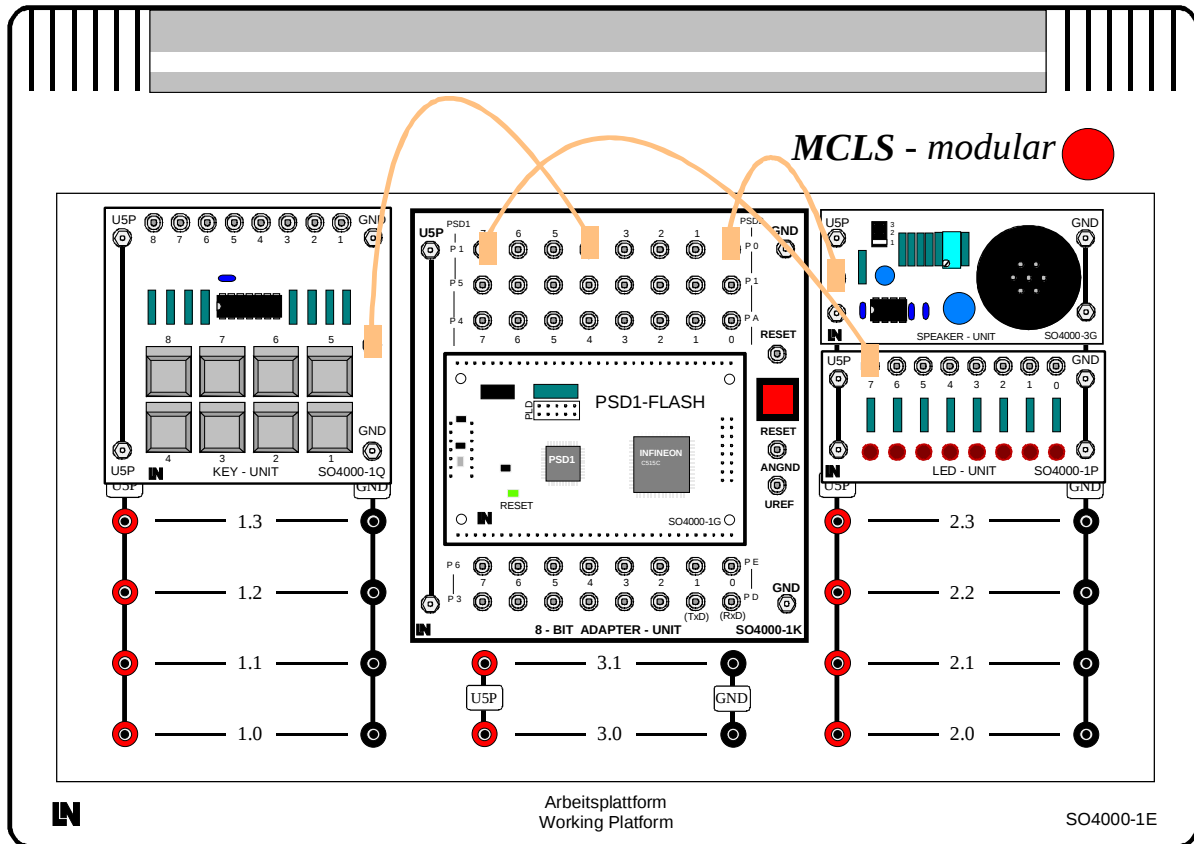


Fig.: 206: Instalación de aparatos del ensayo CMC 3-2C

Ensayo CMC 3-2C

Mediante el Timer 2 se debe realizar una emisión de una melodía. La información relativa a la melodía (altura y longitud de tono) está archivada en tablas a las que debe acceder correspondientemente el programa.

Como función adicional se debe activar la reproducción de la secuencia de tonos mediante el accionamiento de una tecla de la unidad de teclas.

¿Qué conocimientos se pretenden transmitir?

- Generación de la frecuencia de tonos por medio de Timer 2
- Generación de la longitud de tono mediante bucle de espera
- Metodica del acceso a los contenidos de tabla
- Definición de variables en tamaño de bits/ bytes
- Manera del control de proceso

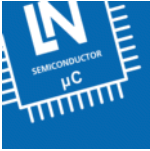
Explicaciones relativas al ejercicio:

La unidad de altavoz se conecta al pin de puerto P1.0 para la emisión de la melodía. Para la generación de un tono se utiliza el Timer 2 como marcador de tiempo con autorecarga de 16 bits, conmutándose correspondientemente el pin de puerto P1.0.

Una tabla contendrá los valores necesarios para las alturas de tono para media duración de periodo de la oscilación (corresponde al tiempo de periodo del marcador de tiempo). Una segunda tabla contiene los valores para longitud de tono. Estos valores son factores que determinan, por medio de un correspondiente retardo mediante bucle de espera, la longitud de tono.

Ejercicios y soluciones

Realice la instalación experimental según la fig.206. Seleccione la amplificación media para la unidad de altavoz.



Croquice un organigrama del programa del que se desprenda el control de proceso fundamental para la emisión de la melodía con los accesos a las tablas.

Escriba el texto fuente de ensamblador para la inicialización del Timer 2.

Observe el texto fuente del programa ensamblador cmc32c.asm.

La siguiente figura permite la asociación de las notaciones según escala y frecuencia de tono.

Determine, para una melodía corta de su elección, la secuencia de frecuencias individuales.

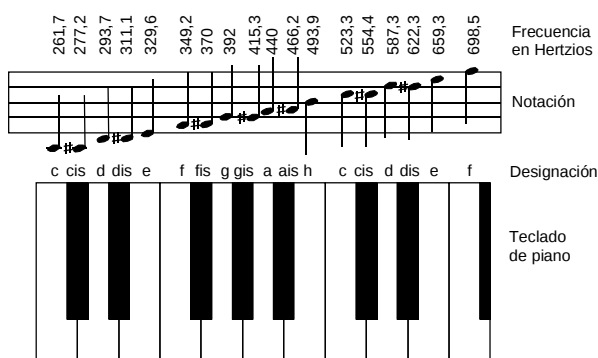


Fig 207: Asociación de las frecuencias a la notación

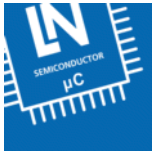
Para su orientación le indicamos a continuación las frecuencias de los primeros tonos según cmc32c.asm:

392 Hz → g
440 Hz → a
392 Hz → g
449 Hz → f

Calcule las constantes de tiempo necesarias para el Timer 2, si como función útil en la ISR se invierte el pin que activa la unidad de altavoz.

Para su orientación le indicamos a continuación las constantes de tiempo para los primeros tonos según cmc32c.asm:

-1275 → g
-1136 → a
-1275 → g
-1432 → f



Croquice el organigrama para la ISR y escriba el texto fuente para la ISR.

Archive las constantes de tiempo calculadas en una tabla en el texto fuente del programa y aclare el principio del acceso a las tablas por medio del comando

MOVC A,@A-DPTR.

Partiendo de un bucle de espera con 50 ms se puede controlar la longitud de tono utilizando un factor para la repetición.

Crée en el texto fuente para su melodía la lista de las longitudes de tono como resultado de estos factores. Observe el texto fuente del programa ensamblador cmc32C.asm.

¿Cómo se pueden definir variables en tamaño de byte/bit?

Consulte el manual de instrucciones para el ensamblador y compare la notación en el texto fuente cmc32c.asm

¿Cómo se puede realizar una localización controlada de estas variables en la memoria?

Lista de programa CMC 3-2C

```

; *****
; * Proyecto:           Emisión de una melodía                *
; * Archivo fuente:     cmc32c.asm                            *
; * Herramientas de MC: as, p2hex, debugger                    *
; * Perfil:             Perfil integrado PSD1-FLASH-C515C-AS   *
; * Recursos de MC:     Unidad de altavoz en P1.0,             *
; *                    unidad de teclas en P1.4, LED en P1.7    *
; * Editor:             HAG                                     *
; *****

; ----- Descripción de funcionamiento -----
; Un accionamiento de una tecla cualquiera de la unidad de teclas
; inicia de melodía (interrupción externa 2),
; Timer 2 genera las frecuencias de tonos en modo de marcador de
; tiempo, utilizándose la función de autorecarga, los valores
; para las frecuencias de los tonos y la longitud de los mismos
; están archivados en tablas y se procesan de manera secuencial,
; para la visualización de la longitud de tono se activa un LED
; -----

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

CPU      80515          ; Selección de microcontrolador
INCLUDE  C515C.INC      ; Definiciones de SFR
INCLUDE  BITFUNCS.INC   ; Inserción de diversas funciones

; Constante de tiempo para registro contador
TC_T2     EQU    -2      ; Valor por defecto para fase de inicio

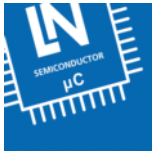
; Definición de una variable en tamaño de bit
; A partir de dirección de bit 0
        SEGMENT      bitdata
NO_TECLA      DB    ?    ; 1 = sin accionamiento de tecla

; Definición de variables en tamaño de byte
; a partir de dirección 30H
        SEGMENT      data
OFFSET_ALTURA      DB    ?    ; Offset para tabla de las alturas de tono
OFFSET_LONGITUDE     DB    ?    ; Offset para tabla de las longitudes de tono

; ----- A partir de aquí código de programa -----
        SEGMENT      code
ORG      0000H
LJMP     START          ; Salto al programa principal
; ----- Introducción de los vectores de interrupción -----
ORG      002BH          ; Posición IV para Timer 2
LJMP     ISR_T2          ; IV
ORG      004BH          ; Posición IV para int. ext. 2
LJMP     ISR_EXT2        ; IV
org      100H            ; Programa principal a partir de 100H

; *****
; *                    Emisión de melodía                    *
; *****
START:
        CALL  INIT_EXT2      ; Inicialización de la int.ext. 2
        CALL  INIT_T2        ; Inicializar Timer 2
        SETB  P1.7           ; Desconectar LED
ST1:    SETB  NO_TECLA

```



```
ST2:  JB      NO_TECLA,ST2          ; Esperar accionamiento de tecla
; .....
      MOV     OFFSET_ALTURA,#000H  ; Offset para tabla de alturas de
                                      ; tonos
      MOV     OFFSET_LONGITUDE,#000H ; Offset para tabla de longitudes
                                      ; de tonos
; .....
ST0:  MOV     DPTR,#TAB_ALTURA      ; Indicador en primer tono
      MOV     A,OFFSET_ALTURA
; -----
; Coger TC de tabla y cargar en registro de recarga
      MOVC    A,@A+DPTR
      MOV     CRCL,A
      INC     OFFSET_ALTURA
      MOV     A,OFFSET_ALTURA
      MOVC    A,@A+DPTR
      MOV     CRCH,A
      INC     OFFSET_ALTURA
; -----
; Reiniciar T2
      SETB    T2I0                  ; RUN para modo de marcador de tiempo
      CLR     P1.7                  ; Conectar LED
; -----
; Coger valor para longitud de tabla y realizar longitud de tono
      MOV     DPTR,#TAB_LONGITUD
      MOV     A,OFFSET_LONGITUDE
      MOVC    A,@A+DPTR
      JZ      END_SR
      INC     OFFSET_LONGITUDE
; .....
; Esperar duración del tono
      MOV     R2,A
ST4:  CALL    TIEMPO_50MS            ; A * 50MS
      DJNZ    R2,ST4
; .....
      CLR     T2I0                  ; Parar T2
      SETB    P1.7                  ; Desconectar LED
      CALL    TIEMPO_50MS            ; Pausa entre los tonos
; .....
      SJMP    ST0                  ; Siguiente tono
; -----
END_SR:
      CLR     T2I0                  ; Parar T2
      SJMP    ST1
; *****
; -----
; Subprograma para la inicialización de la interrupción externa 2
; -----
INIT_EXT2:
      CLR     EAL
      SETB    I2FR                  ; Flanco 0-1
      SETB    EX2
      SETB    EAL
      RET
; -----
; -----
; Subprograma para la inicialización de Timer 2
; -----
INIT_T2:
      CLR     EAL                  ; Bloquear globalmente interrupciones
      MOV     TH2,#HI (TC_T2)      ; Cargar RC
      MOV     TL2,#LO (TC_T2)
      MOV     CRCH,#HI (TC_T2)     ; Cargar registro de autorecarga
      MOV     CRCL,#LO (TC_T2)
```

```

ANL    T2CON,#01100000B
ORL    T2CON,#00010000B ; Autorecarga, reloj 1/12, temporizador
                        ; parado
SETB   ET2             ; Autorizar interrupción por Timer 2
SETB   EAL             ; Autorizar globalmente interrupciones
SETB   T2I0            ; Modo de marcador de tiempo e inicio
RET

; -----
;                               ISR para primera interrupción externa 2
; -----
ISR_EXT2:
    CPL    NO_TECLA
    MOV    R5,#10
I00:    CALL TIEMPO_50MS
    DJNZ   R5,I00
    CLR    IEX4
    RETI

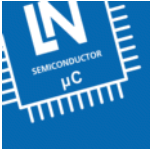
; -----
;                               ISR para Timer 2
; -----
ISR_T2:
    CLR    TF2          ; Reposicionar bit de desbordamiento
    CPL    P1.0         ; Función útil: generación de oscilaciones
    RETI

; -----
;                               Subprograma para retardo
; -----
TIEMPO_50MS:
    MOV    R7,#125      ; 125*
T01:    MOV    R6,#200
T00:    DJNZ   R6,T00    ; 2CM * 200 = aprox. 50000 CM
    DJNZ   R7,T01
    RET

; --Tabla de las constantes de tiempo para las alturas de tono--
; Ejemplo 1: TCO = -1275 -> Duración de periodo = 2,55 ms -> 392 Hz
; Ejemplo 2: la normal -> 440 Hz -> 2,2727 ms -> TCO = -1136
; -----
TAB_ALTURA:
    DW     -1275,-1136
    DW     -1275,-1432,-1517,-1432
    DW     -1275,-1136
    DW     -1275,-1432,-1517,-1432
    DW     -1275,-1275
    DW     -1136,-1012,-955
    DW     -1012,-1136
    DW     -1275
    DW     -1702,-1517,-1702,-1517
    DW     -1432,-1275,-1432
    DW     -1517,-1432,-1517,-1432
    DW     -1275,-1136,-1275
    DW     -955,-1012,-1136,-1275
    DW     -955,-1136,-1275,-1432
    DW     -1517,-1702
    DW     -1910

; ----- Tabla para las longitudes de tonos -----
; Factores para 50 ms
; -----
TAB_LONGITUD:
; 1 longitud de tono = 32 , 1/2 longitud de tono = 16 , 1/4 longitud
de tono = 8

```



```
DB      16,16
DB      12,4,8,8
DB      16,16
DB      12,4,8,8
DB      16,16
DB      16,8,8
DB      16,16
DB      32
DB      12,4,8,8
DB      12,4,16
DB      12,4,8,8
DB      12,4,16
DB      8,8,8,8
DB      8,8,8,8
DB      16,16
DB      32
DB      00H          ; Indicativo de final

END
```

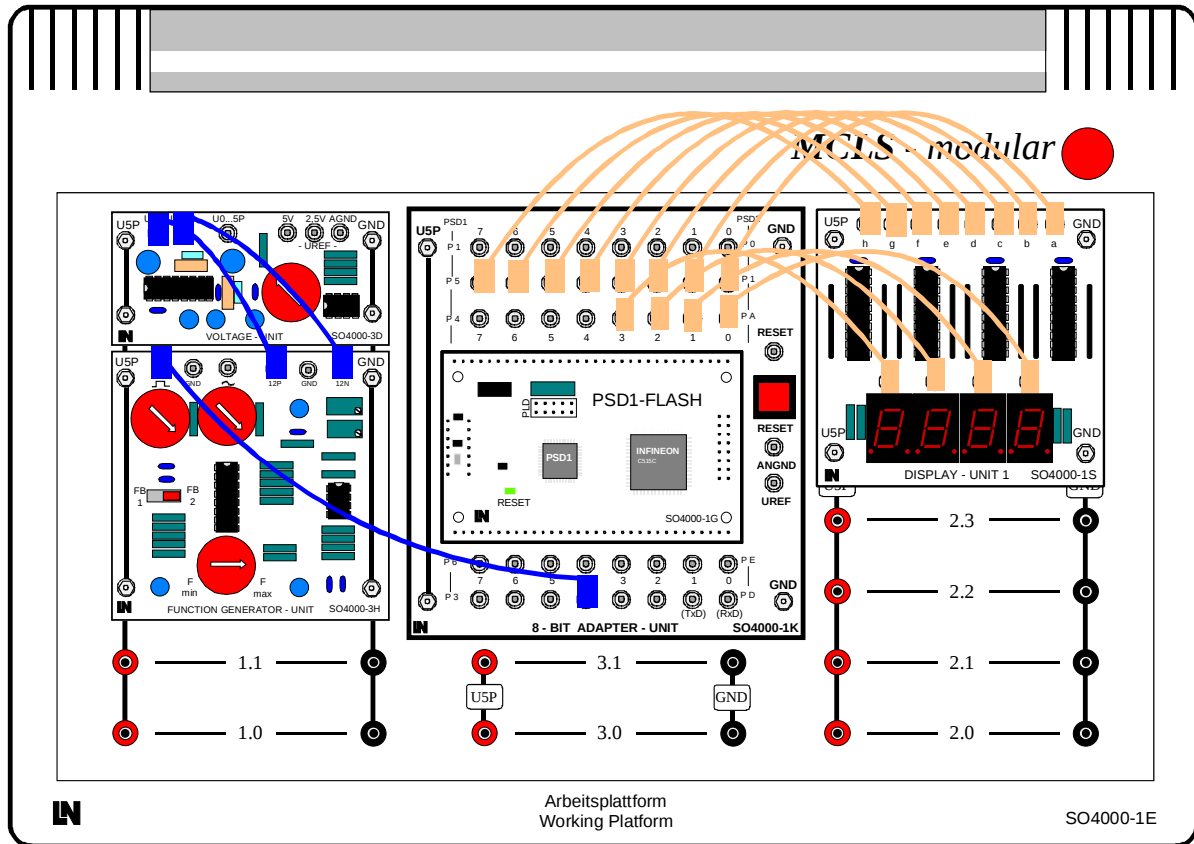


Fig.: 208: Instalación de aparatos del ensayo CMC 3-2D

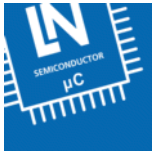
Ensayo CMC 3-2D

Mediante 2 temporizadores del C515C se deberá realizar la función de un contador de frecuencia.

La señal de contado está en la entrada de contado de Timer 0, el cual se configurará como contador. La realización del tiempo de brecha se puede efectuar de manera muy sencilla por medio de Timer 2.

¿Qué conocimientos se pretenden transmitir?

- Utilización de un temporizador como contador
- Utilización de un 2º temporizador como marcador de tiempo
- Generación del tiempo de brecha de 1 s
- Utilización de un SP para convertir el valor contado binario en un valor BCD



Explicaciones relativas al ejercicio:

La señal de salida de la unidad de generador de funciones se conecta a la entrada de contado de Timer 0 (P3.4). En el servicio de este módulo se debe observar que las tensiones de servicio U12P y U12N se deben suministrar por medio de la unidad de tensión.

Para visualizar la frecuencia hasta, como máximo, 9999 Hz, se puede utilizar la unidad de display 1. Los subprogramas para la activación del módulo de visualización incluyendo las indicaciones relativas a la conexión a los puertos 5 y 4 del microcontrolador se encuentran archivados en el fichero de inclusión FCT7SEGM.INC.

Ejercicios y soluciones

Realice la instalación experimental según la fig.208. Seleccione la salida de señal rectangular de la unidad de generador de funciones y ajuste una tensión de salida suficientemente grande.

Compruebe especialmente la alimentación de tensión adicional de la unidad de generador de funciones. Consulte también el manual de instrucciones del módulo.

¿Cómo se puede realizar con el Timer 2 un tiempo de brecha de 1 segundo?

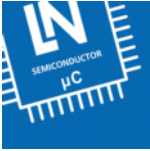
Croquice los organigramas para el programa principal y la ISR.

¿Cómo se inicializa el Timer 0?

INIT_T0:

¿Cómo se inicializa el Timer 2?

INIT_T2:



Discuta las variantes de solución para la conversión del resultado de contado binario en informaciones BCD.

Considere la utilización del comando DA A en relación con los comandos ADD A,op o ADDC A,op en el texto fuente cmc32d.asm.

Controle la función del subprograma CONV_BIN_BCD, controlando las zonas de memoria asignadas para el valor binario y el resultado BCD con la ayuda del depurador.

Coloque un punto de ruptura en la línea de comandos CALL FR_MEDIR_VIS y continúe la ejecución del programa a pasos individuales, y en CALL CONV_BIN_BCD con un paso de proceso. Observe la definición de las correspondientes variables en el texto fuente cmc32d.asm.

Lista de programa CMC 3-2D

```

; *****
; * Proyecto:           Medir frecuencias                      *
; * Archivo fuente:     cmc32d.asm                            *
; * Herramientas de MC: as, p2hex, debugger                    *
; * Perfil:             Perfil integrado PSD1-FLASH-C515C-AS   *
; * Recursos de MC:     Unidad de display 1 en P5 (segm. a-h en *
; *                     P5.0-P5.7) y en P4 (posiciones D0-D3    *
; *                     en P4.0-P4.3),                          *
; *                     Señal de salida rectangular de la      *
; *                     unid. de generador de func. en P3.4    *
; *                     (entrada de contado de Timer 0),       *
; *                     adicionalmente es necesaria la unidad  *
; *                     de tensión para la alimentación de    *
; *                     tensión del generador                  *
; * Editor:             HAG                                     *
; *****

; ----- Descripción de funcionamiento -----
; Para la medición de una frecuencia se necesitan 2 temporizadores del
; microcontrolador. Mientras que por medio del Timer 2 se prepara la
; base de tiempo de 1 s (función de marcador de tiempo), se cuentan,
; por medio del Timer 0, los impulsos del generador (función de
; contador).
; Los impulsos registrados en el registro contador del Timer 0
; dentro de un segundo se presentan como valor binario de 16 bits y se
; convierten para la visualización en un formato BCD comprimido.
; -----

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

        CPU            80515            ; Selección de microcontrolador
        INCLUDE        C515C.INC        ; Definiciones de SFR
        INCLUDE        BITFUNCS.INC     ; Inserción de diversas funciones

; ----- Definición de constantes -----
RESET EQU    0000H            ; Dirección de inicio de Reset
TCO_T2      EQU    -50000      ; Constante de tiempo para Timer 2
; 50ms hasta el desbordamiento
RL_T2 EQU    -50000           ; Valor de recarga para Timer 2
; 50ms hasta el desbordamiento

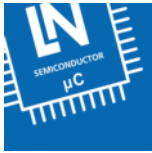
; ----- Segmento de datos para variables -----
        SEGMENT          data

                ORG      20H
VALOR_MEDIDO DW    ?          ; Valor medido de 16 bits
CTR          DB    ?          ; Contador auxiliar

                ORG      40h
FR_BCD_MSB   DB    ?          ; MSB del valor de frecuencia BCD
FR_BCD_MIB   DB    ?          ; Byte central de la frecuencia BCD
FR_BCD_LSB   DB    ?          ; LSB del valor de frecuencia BCD
FR_BIN_MSB   DB    ?          ; Valor de frecuencia binario MSB
FR_BIN_LSB   DB    ?          ; Valor de frecuencia binario LSB

; ----- a partir de aquí código de programa -----
        SEGMENT          code

```



```
org    RESET
LJMP   START          ; Salto al programa principal

; ----- Introducción del vector de interrupción -----
ORG    002BH          ; Posición del IV para Timer 2
LJMP   ISR_T2          ; IV
; -----

org    100H            ; Programa principal a partir de 100H

; *****
; *          Programa principal Medición de frecuencia          *
; *****
START:
MOV     CTR,#20         ; Inicializar contador auxiliar
CALL    INIT_T0         ; Inicializar Timer 0 como contador
CALL    INIT_T2         ; Timer 2 como marcador de tiempo de 50ms
                     ; con autorecarga
ST0:    CALL    FR_MEDIR_VIS ; Medir y visualizar frecuencia
CALL    TIEMPO          ; Retardo aproximadamente 130ms
SJMP    ST0             ; medir y visualizar continuamente

; -----
; Complejo de funciones para medición y visualización de frecuencia
; -----
FR_MEDIR_VIS:
MOV     FR_BIN_MSB,VALOR_MEDIDO+1 ; Preparar valor de frecuencia
MOV     FR_BIN_LSB,VALOR_MEDIDO   ; de 16 bits para conversión

CALL    CONV_BIN_BCD          ; Conversión binario en BCD

MOV     DPH,FR_BCD_MIB        ; Byte BCD central al indicador de datos
MOV     DPL,FR_BCD_LSB        ; LSB del byte BCD al indicador de datos

CALL    OUT_DPTR_7SEGM        ; Contenido de DPTR a indicación
RET

; -----
; Conversión de binario a BCD
; -----
CONV_BIN_BCD:
; Borrar margen de resultados
MOV     R1,#003H             ; 3 posiciones de memoria
MOV     R0,#FR_BCD_MSB
K4:     MOV     @R0,#000H      ; Cargar con 00
INC     R0
DJNZ    R1,K4

; .....
; Conversión de número binario de 16 bits en número BCD
MOV     R4,#10H              ; Convertir 16 bits
K0:     MOV     R0,#FR_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,#FR_BCD_LSB        ; Dirección BCD de destino 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            ; Siguiete dirección BCD de destino
DJNZ   R3,K2         ; ¿Ya se ha alcanzado la 3ª dirección de
                     ; destino?
DJNZ   R4,K0         ; ¿Ya se han convertido 16 bits?
RET

; -----
;                               Inicializar Timer 0
; -----
INIT_T0:
MOV     TH0,#000H      ; Reposicionar registro contador
MOV     TL0,#000H
ANL     TMOD,#11110000B
ORL     TMOD,#00000101B ; Como contador en el modo 1
SETB    TR0           ; Liberar contador
RET

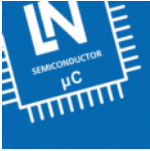
; -----
;                               Inicializar Timer 2
; -----
INIT_T2:
CLR     EAL           ; Bloquear interrupciones globalmente
CLR     TF2           ; Resposicionar indicador de
                     ; desbordamiento
MOV     TH2,#hi(TCO_T2) ; Precargar registro contador
MOV     TL2,#lo(TCO_T2) ; a 50ms
MOV     CRCH,#hi(TCO_T2) ; Ajustar 50 ms
MOV     CRCL,#lo(TCO_T2) ; en registro de recarga
ANL     T2CON,#01100000B
ORL     T2CON,#00010000B ; Autorecarga, reloj 1/12,
                     ; temporizador parado
SETB    ET2           ; Autorizar interrupciones por Timer 2
SETB    EAL           ; Autorizar globalmente interrupciones
SETB    T2I0          ; Modo de marcador de tiempo e inicio
RET

; -----
;                               ISR para Timer 2
; -----
ISR_T2:
CLR     TF2           ; Reposicionar indicador de desbordamiento
DJNZ    CTR,IS0       ; ¿Contador-1, =0?
MOV     CTR,#20       ; Cargar de nuevo contador auxiliar
MOV     VALOR_MEDIDO,TL0 ; TL0 = LT Valor medido
MOV     VALOR_MEDIDO+1,TH0 ; TH0 = HT Valor medido
MOV     TL0,#000H     ; Reposicionar registro contador de Timer 1
MOV     TH0,#000H
IS0:    RETI

; -----
;                               Subprograma para retardo
; -----
TIEMPO:
MOV     R7,#000H
T01:    MOV     R6,#000H
T00:    DJNZ    R6,T00
        DJNZ    R7,T01
        RET

; -----
;                               Insertar funciones de indicación para indicación de 7 segmentos
; -----

```



```
INCLUDE FCT7SEGM.INC    ; Archivo de inclusión  
                        ; en el mismo directorio  
  
END
```

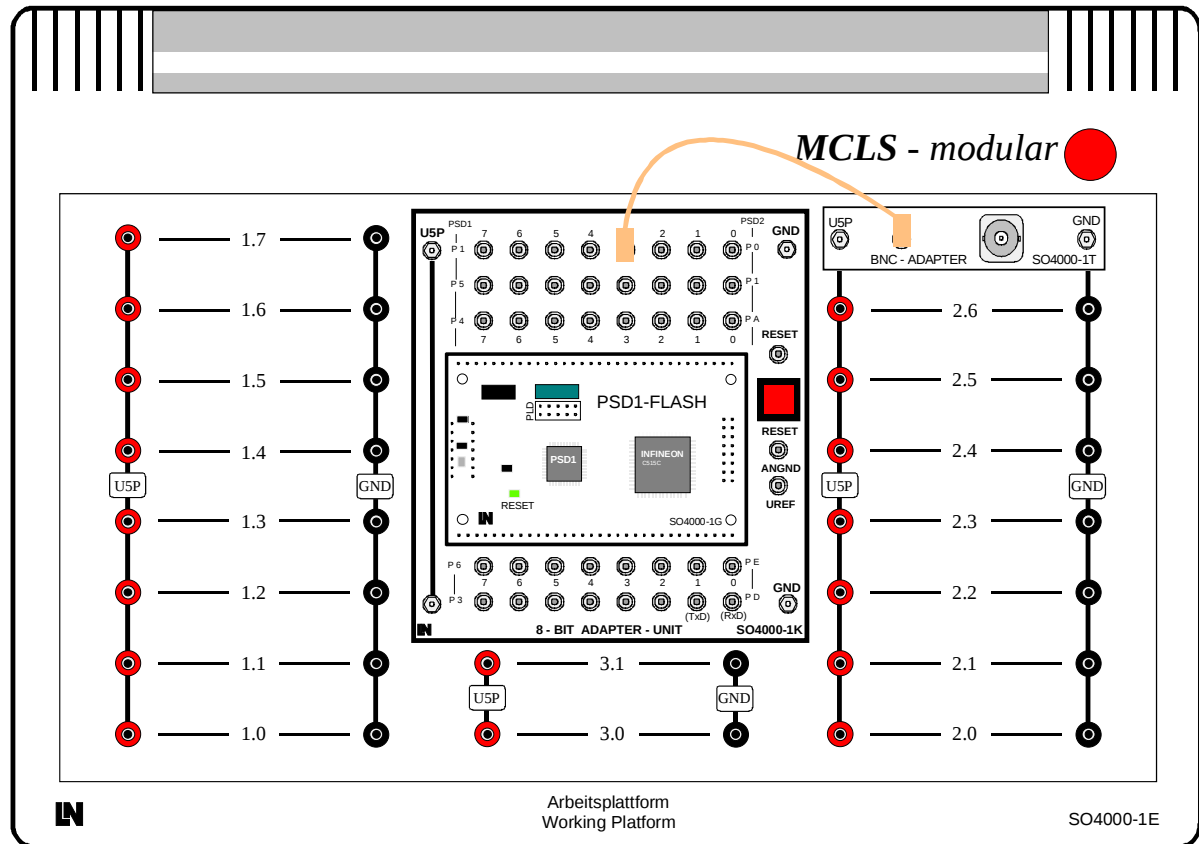


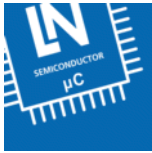
Fig.: 209: Instalación de aparatos del ensayo CMC 3-2E

Ensayo CMC 3-2E

Generación de señales PWM con el Timer 2 en modo de comparación. Mediante el Timer 2 se deberán generar señales PWM con diferentes frecuencias y relaciones de exploración. Se deberá utilizar el modo de comparación 0 del temporizador.

¿Qué conocimientos se pretenden transmitir?

- Principio de funcionamiento del modo de comparación 0
- Ajustar frecuencia y relación de exploración de la señal PWM
- Salida de señales al mismo tiempo por medio de varios canales con relación de exploración diferente pero con la misma frecuencia
- Generación de señales sin participación de la CPU una vez realizada la inicialización del temporizador
- El valor de comparación siempre debe ser superior al valor de recarga
- Valores límite para relaciones de exploración realizables



Explicaciones relativas al ejercicio:

El pin de salida del microcontrolador para la correspondiente señal de comparación (P1.1, P1.2 o P1.3) se conecta, por medio de un cable de medición apropiado (también se pueden utilizar el adaptador de BNC), al oscilógrafo.

Después del inicio del programa cmc32e.asm, las señales PWM se pueden verificar según su inicialización.

La modificación de los valores de inicialización permite variar la frecuencia y la relación de exploración de las señales PWM.

Ejercicios y soluciones

Realice la instalación experimental según la fig.209. Conecte un oscilógrafo a la correspondiente salida de señales PWM.

¿Qué registros de funciones especiales participan en la inicialización del Timer 2 para el modo de comparación 0?

Escriba el programa ensamblador para la inicialización del Timer 2 para la salida simultánea de las siguientes señales PWM:

Canal de comparación 2: $T = 5\text{ ms}$, $TV = 20\%$

Canal de comparación 3: $T = 5\text{ ms}$, $TV = 50\%$

Ensaye el programa.

Observe el texto fuente del programa ensamblador cmc32e.asm.

¿Por qué, con una generación simultánea de señales en varios canales, las señales PWM solamente se pueden generar con la misma frecuencia?

¿Qué valores de inicialización para el registro de recarga y el de comparación son necesarios para realizar una frecuencia de 1 KHz con una relación de exploración del 40%?

Modifique la inicialización en CMC302e.asm y verifique la señal PWM.

**¿Cuántas interrupciones se pueden realizar dentro de un periodo de temporizador?
¿Cuáles son?**

¿Por qué el valor de comparación debe ser siempre superior al valor de recarga?

¿De qué depende la posible resolución para la señal PWM?

La fig. 210 muestra los posibles valores extremos para relaciones de exploración generables. Cambie correspondientemente la inicialización para el Timer 2 e intente visualizar los impulsos con el oscilógrafo.

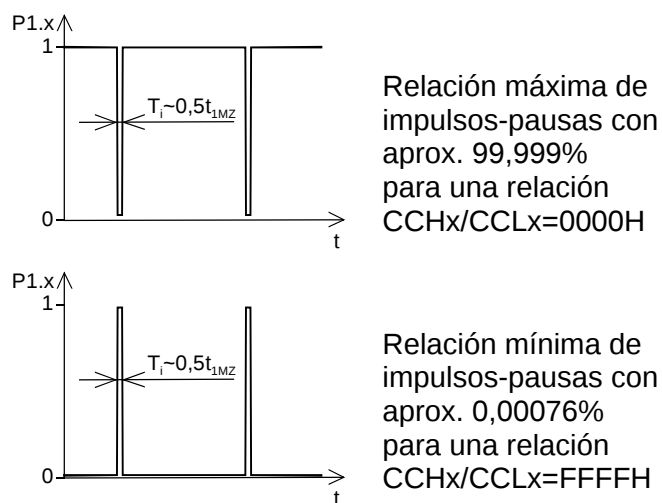
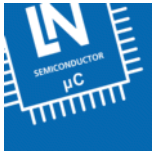


Fig. 210: Valores extremos para relaciones de exploración realizables



Modifique correspondientemente el texto fuente del programa ensamblador *cmc32e.asm*.

Controle las relaciones de exploración alcanzadas en las salidas de comparación por medio de un oscilógrafo.

Valor recarga (HEX)	Tiempo periodo (µs)	Frec. contado (Hz)	Resolución (Bit)	Posible relación de impulsos/pausas	
				1/X	X/1
0000	65536	15,25	16	1/65535	65535/1
8000	32768	30,51	15	1/32767	32767/1
C000	16384	61,03	14	1/16383	16383/1
E000	8192	122,07	13	1/8191	8191/1
F000	4096	244,14	12	1/4095	4095/1
F800	2048	488,28	11	1/2047	2047/1
FC00	1024	976,56	10	1/1023	1/1023
FE00	512	1953,12	9	1/511	511/1
FF00	256	3906,25	8	1/255	255/1
FF80	128	7812,50	7	1/127	127/1
FFC0	64	15625,00	6	1/63	63/1

Tab. 01: Tiempo de periodo, resolución y relaciones de impulsos-pausas para diferentes valores de recarga (fosc = 12 MHz)

Lista de programa CMC 3-2E

```

; *****
; * Proyecto:          Generación de señales PWM          *
; * Archivo fuente:    cmc32e.asm                          *
; * Herramientas de MC: as, p2hex, debugger                *
; * Perfil:            Perfil integrado PSD1-FLASH-C515C-AS *
; * Recursos de MC:    Canales CC2(P1.2) y CC3 (P1.3) para  *
; *                   la salida de las señales PWM,        *
; *                   adaptador BNC en el corresp. pin de  *
; *                   puerto p. la conexión de un oscilógrafo *
; * Editor:            HAG                                  *
; *****

; ----- Descripción de funcionamiento -----
; Generación de 2 señales PWM por medio del Timer 2
; Salida de las señales por medio de P1.2 (CC2) y P1.3 (CC3)
; Ajuste de la frecuencia común por medio del registro de recarga
; Ajuste de las relaciones de exploración por medio del registro
; de comparación
; -----

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

; ----- Definición de constantes -----
TCO_REL EQU -5000      ; 5 ms hasta el desbordamiento
CMP_CC2 EQU -1000      ; 1ms desde comparación 2 hasta desbordamiento
CMP_CC3 EQU -2500      ; 2,5ms desde comparación 3 hasta
                       ; desbordamiento

; ----- A partir de aquí código de programa -----
SEGMENT code
ORG      0000H          ; Código de programa a partir de 0000H

; *****
; * Programa principal Generación de PWM          *
; *****
START:
MOV      CRCH,#hi(TCO_REL) ; Cargar registro de recarga
MOV      CRCL,#lo(TCO_REL)
MOV      TH2,#hi(TCO_REL) ; Cargar registro de contado
MOV      TL2,#lo(TCO_REL)
MOV      CCH2,#hi(CMP_CC2) ; Cargar registro de comparación 2
MOV      CCL2,#lo(CMP_CC2)
MOV      CCH3,#hi(CMP_CC3) ; Cargar registro de comparación 3
MOV      CCL3,#lo(CMP_CC3)
MOV      CCEN,#10100000b ; CC3 y CC2 en modo de comparación
ANL      T2CON,#01100000B
ORL      T2CON,#00010000b ; Autorecarga, reloj 1/12,
                       ; temporizador parado
                       ; Modo de comparación 0
SETB     T2I0            ; Modo de marcador de tiempo e inicio
SJMP     $               ; Bucle sinfín como "final"
                       ; del programa principal

END

```

CMC 3-3 Aplicación de un convertidor analógico-digital (convertidor A/D) integrado

Con las aplicaciones de medición de tensión continua y regulación de temperatura se presenta el empleo del convertidor A/D „on chip“.

Fundamentos preliminares

El convertidor A/D integrado del C515C permite, con un volumen muy reducido de hardware, la solución de tareas metrológicas sin elevados requisitos hacia la precisión de medición.

El convertidor tiene las siguientes características:

- 8 canales de entrada (multiplex)
- Requiere una tensión de referencia externa
- Resolución de 10 bits en el margen de tensión de referencia ajustado
- Tiempo de conversión 8 ó 16 ciclos de máquina
- Inicio de conversión mediante comando software
- Solicitud de interrupción posible tras cada conversión

Como método de conversión se emplea la aproximación sucesiva con un grupo de condensadores. Partiendo de una tensión de ref. a aplicar externamente, se puede variar, dentro de unos límites definidos, la tensión de ref. interna que determina la ventana de medición. Por medio de 4 SFR programables por el usuario se puede programar el convertidor A/D y consultar el resultado de conversión.

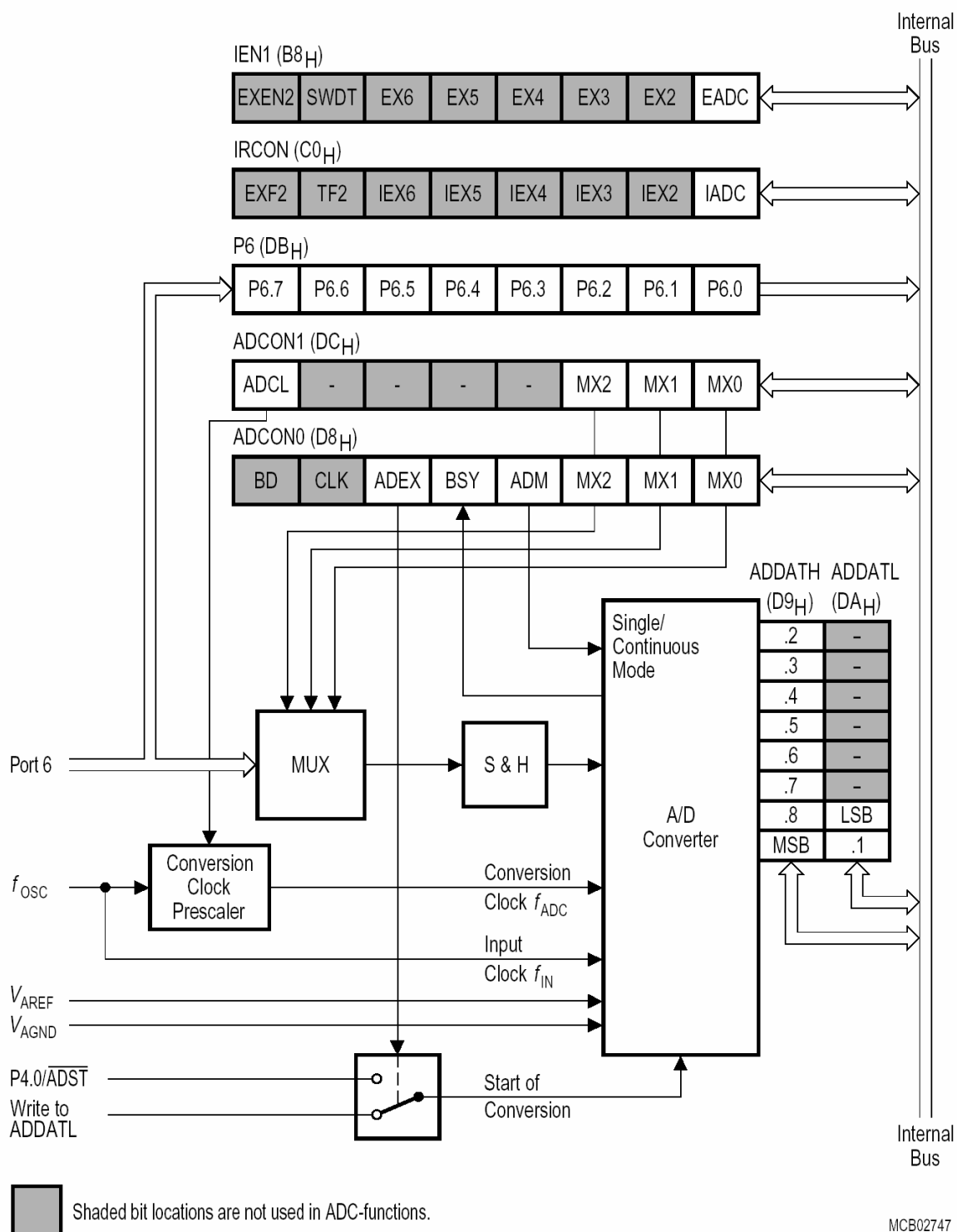
Indicaciones para la inicialización del convertidor A/D

- Inicialización del convertidor A/D por medio de los SFR ADCON1 y ADCON0
- Modo de conversión única o continua por medio del bit de modo ADM (ADCON0)
- MX0 a MX2 para la selección binaria de las entradas analógicas AN0 a AN7
- Bit de estado BSY=1, cuando conversión en curso, y BSY=0, cuando conversión finalizada
- El indicador de solicitud de interr. IADC en el SFR IRCON señala una conversión completada
- SFR ADDATH y ADDATL contiene el resultado de conversión
- El inicio de una conversión se realiza por medio de un comando de escritura en ADDATL

La base para la tensión de referencia interna es una tensión de referencia a aplicar externamente (Pin V_{AREF} t Pin V_{AGND}) que se deben poner a disposición por medio de una fuente de tensión de referencia externa, p. ej. por medio de la unidad de tensión.

Las siguientes condiciones marginales se deben observar, entre otras, en el ajuste y para las tensiones de entradas analógicas (V_{AINPUT}):

- $V_{AREF} = V_{CC} \pm 5\%$ ($V_{CC} = 5\text{ V} \pm 10\%$)
- $V_{AGND} = V_{SS} \pm 0,2\text{ V}$ ($V_{SS} = 0\text{ V}$)
- $V_{IntAREF} - V_{IntAGND} \geq 1\text{ V}$
- Valor mínimo para $V_{AINPUT} = V_{AGND} - 0,2\text{ V}$
- Valor máximo para $V_{AINPUT} = V_{AREF} + 0,2\text{ V}$



MCB02747

Fig. 301: Diagrama de bloques del convertidor A/D

Comportamiento de tiempo del convertidor A/D

Una conversión se inicia con un comando de escritura en el SFR ADDATL. Este comando de escritura siempre iniciará una conversión, incluso si una conversión solamente acaba de empezar. La nueva conversión se inicia con el siguiente comando de máquina y se pone el indicador BSY.

Una conversión se clasifica en 3 componentes temporales (tiempo de carga, tiempo de exploración, tiempo de conversión), incluyendo el tiempo de conversión el tiempo de exploración y el tiempo de carga. En el último ciclo de máquina del tiempo de conversión se carga el resultado de la conversión en ADDAT y se borra el indicador BSY.

De ello resulta que mediante el comando JB BSY,\$ se puede realizar de manera muy sencilla un sondeo para una conversión completada. El indicador de solicitud de interr. IADC en el SFR IRCON señala una conversión completada.

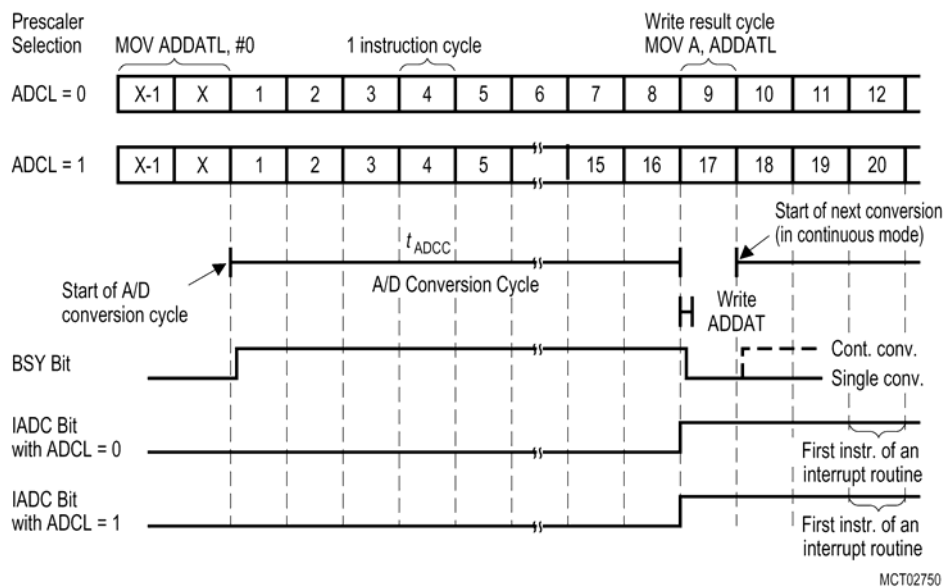


Fig. 302: Comportamiento de tiempo del convertidor A/D

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 display 2 (LCD)	SO4000-1B
1	Unidad de control 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	
4	Cable de medición 2mm 15cm azul	SO5126-5K
15	Cable medición 2mm 15cm amarillo	SO5126-5M
1	CMC 3 Programación de periféricos on-chip C515C	SH5019-1C

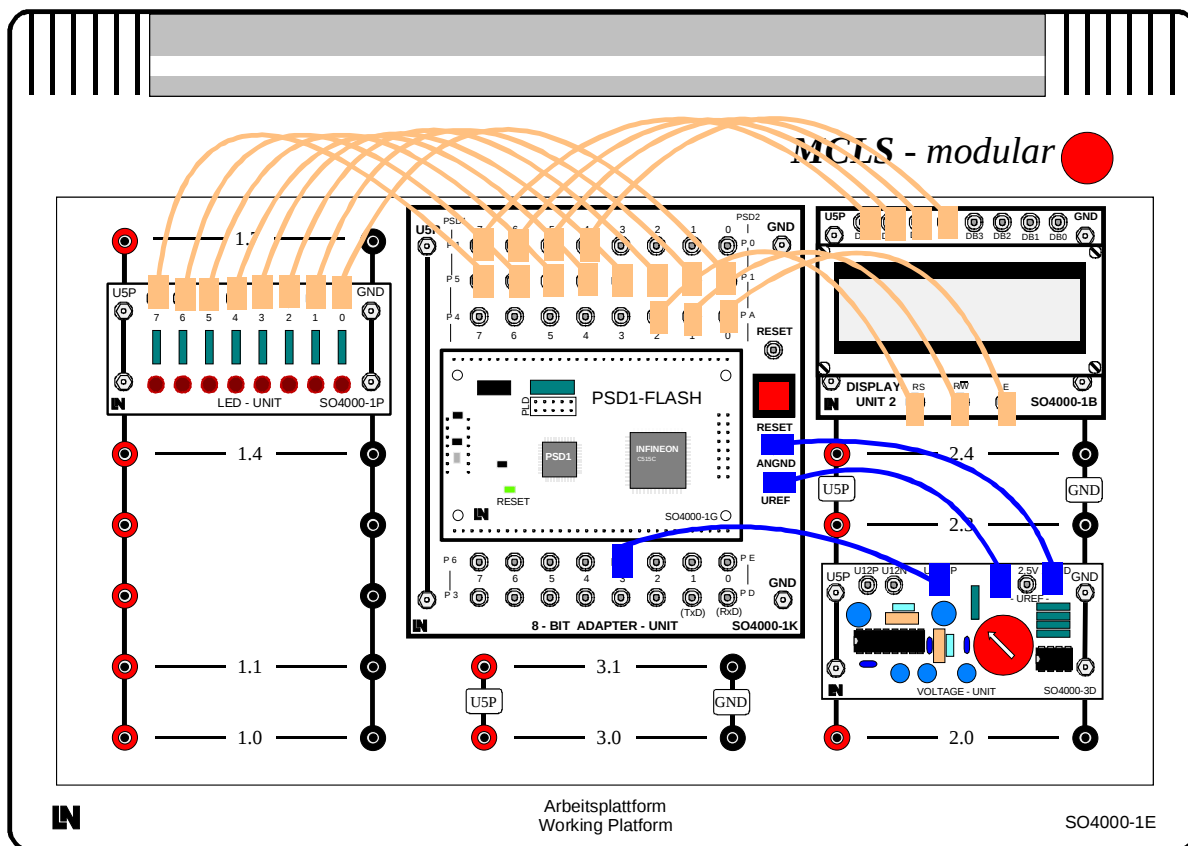


Fig.: 303: Instalación de aparatos del ensayo CMC 3-3A

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 3-3A

Medir una tensión continua con el convertidor analógico-digital integrado. Mediante el convertidor A/D integrado se deberá medir una tensión continua en una de las entradas analógicas. La visualización del valor medido se realizará en mV por medio de la unidad de LCD. Cada 800 ms se iniciará una medición nueva por medio de un marcador de tiempo.

¿Qué conocimientos se pretenden transmitir?

- Propiedades y principio de funcionamiento del convertidor A/D integrado
- Modos de servicio e inicialización del convertidor A/D
- Empleo de un temporizador para el proceso de medición
- Conversiones necesarias del valor medido binario para la visualización en la unidad LCD.

Explicaciones relativas al ejercicio:

Una tensión continua aplicada en la entrada analógica AN3 (como máximo 5 V) se debe medir cíclicamente cada 800 ms. Puesto que tanto la ventana de medición como la entrada analógica permanecen sin cambiar, se puede ajustar el servicio de medición continua en el convertidor A/D. Por medio del control de proceso se recoge el valor de conversión más actual de ADDATH y se convierte correspondientemente para la visualización en la unidad de LCD (de 4 dígitos en mV).

Los subprogramas para la activación del módulo de indicación incluyendo las indicaciones para la conexión a los puertos 1 y 4 del microcontrolador se encuentran archivados en el archivo de inclusión FCT_LCD.INC.

La tensión de referencia externa necesaria para el convertidor A/D y la tensión continua variable para AN3 (como máximo 5 V) se suministran por la unidad de tensión.

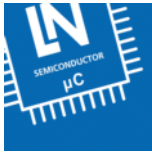
Ejercicios y soluciones

Realice la instalación experimental según la fig.303. Observe la polarización correcta de la tensión de referencia externa en la conexión de la unidad de tensión a la unidad de adaptador de 8 bits.

¿Qué SFRs participan en la inicialización del convertidor A/D integrado?

¿Por qué no es recomendable la programación de ADCON0 por medio de un comando MOV?

Croquice los organigramas para el programa principal y la ISR.



¿Después de qué tiempo a partir del inicio de conversión se puede contar con un resultado de conversión válido en ADDAT?

¿Qué tensión está asociada a un bit en el resultado de conversión? (Resolución de 10 bits)

¿Qué pasos de conversión son necesarios para que se pueda visualizar el resultado de conversión binaria en la unidad de LCD?

Controle los valores medidos visualizados de la instalación experimental comparándolos con resultados de medición de un voltímetro digital apropiado sobre el margen de medición.

Observe el escalonamiento marcado por la resolución de 8 bits de los valores medidos en la instalación experimental.

Indique las posibles causas para errores de medición en la instalación experimental.

Lista de programa CMC 3-3A

```

; *****
; * Proyecto:           Medir una tensión continua          *
; * Archivo fuente:     cmc33a.asm                          *
; * Herramientas MC:    as, p2hex, debugger                  *
; * Perfil:             Perfil integrado PSD1-FLASH-C515C-AS *
; * Recursos de MC:     Unidad LCD en P1.7-P1.4 y P4.0-P4.2  *
; *                    (ver detalles en FCT_LCD.INC),         *
; *                    Unidad de tensión para suministro de la *
; *                    tensión de ref. externa de +5V y una   *
; *                    tensión continua de entrada variable   *
; *                    para la entrada analógica AN3,         *
; *                    unidad LED para el control de la tensión *
; *                    continua digitalizada en P5            *
; * Editor:             HAG                                  *
; *****

; ----- Descripción de funcionamiento -----
; La tensión suministrada por la salida de tensión continua variable
; de la unidad de tensión (0 ... 5V) se mide y se visualiza por medio
; del convertidor analógico-digital integrado.
; Se utiliza la entrada analógica AN3 del convertidor.
; La visualización del valor de tensión medido se realiza por medio
; de la unidad de LCD.
; .....
; Para fines de control se puede ocupar la entrada analógica con la
; tensión de referencia (5V) o media tensión de referencia (2,5V).
; Para el control de la tensión continua de entrada digitalizada, ésta
; se saca en el puerto 5 para la activación de la unidad LED.
; -----

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

; ----- Algunas definiciones de constantes -----
RESET EQU    0000h      ; Dirección de inicio de reseteo

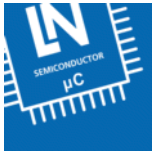
CT_T2 EQU    -20000     ; Constante de tiempo para Timer 2
                        ; 20ms hasta desbordamiento
RL_T2 EQU    -20000     ; Valor de recarga para Timer 2
                        ; 20ms hasta desbordamiento

; ----- Segmento de datos para variables -----
SEGMENT      DATA

      ORG     20h
CTR  DB      ?          ; Contador auxiliar
; Reservar posición de memoria para valores medidos (16 bits)
      ORG     40h
BCD_MSB DB      ?      ; MSB del valor medido BCD
BCD_MIB DB      ?      ; Byte central del valor medido BCD
BCD_LSB DB      ?      ; LSB del valor medido BCD
BIN_MSB DB      ?      ; Valor medido binario MSB
BIN_LSB DB      ?      ; Valor medido binario LSB

      ORG     50H
ASCII DB      ?        ; Reservar 4 bytes para valores medidos ASCII

```



```

        DB      ?
        DB      ?
        DB      ?
; -----

; -----A partir de aquí está el código de programa -----
        SEGMENT      CODE

        ORG      RESET                ; Salto al programa principal a partir
                                        ; de 100H

        LJMP     START

        ORG      002BH                ; Introducir en tabla IV
        LJMP     ISR_T2                ; vector de interrupción para Timer 2

        ORG      100H                ; Programa principal a partir de 100H

; *****
; *                               Programa principal Medición de tensión *
; *****
START:
        MOV      CTR,#40                ; Inicializar contador auxiliar
        CALL     INIT_ADU                ; Inicializar convertidor A/D "on chip"
        CALL     INIT_T2                ; Timer 2 como marcador de tiempo de
                                        ; 20ms con autorecarga

        CALL     INIT_LCD
        CALL     OUT_TEXT1                ; Editar texto en 2ª línea

ST0:
        SJMP     ST0                ; Medir e visualizar continuamente

; -----
;                               Inicializar e iniciar convertidor A/D "on chip"
; -----
INIT_ADU:
        ANL      ADCON0,#11100000B
        ORL      ADCON0,#00001011B    ; Conversión continua, canal 3
        MOV      ADDATL,#00H          ; iniciar ADC
        RET

; -----
;                               Inicializar Timer 2
; -----
INIT_T2:
        CLR      EAL
        MOV      TH2,#hi (CT_T2)      ; Precargar registro contador
        MOV      TL2,#lo (CT_T2)      ; a 20ms
        MOV      CRCH,#hi (RL_T2)     ; Ajustar registro de recarga
        MOV      CRCL,#lo (RL_T2)     ; a 20ms
        MOV      T2CON,#00010001B    ; Marcador de tiempo con autorecarga
        SETB     ET2                  ; Autorizar interrupción por T2
        SETB     EAL
        RET

; -----
;                               ISR contiene las funciones Medir y Visualizar
; -----
ISR_T2:
        CLR      TF2                ; Reposicionar indicador de desbordamiento
        DJNZ     CTR,IS0            ; Contador-1, =0?
; .....
```

```

MOV    CTR,#40      ; Recargar contador auxiliar
CALL   MEDIR        ; Coger valor de conversión, convertir en
                    ; valor de tensión y convertir en valor BCD
CALL   VISUALIZAR    ; Convertir valor medido BCD en signos ASCII
                    ; y en indicador LCD
ISO:   RETI

```

```

; -----
;                               Complejo de funciones MEDIR
; -----
MEDIR:
MOV    A,ADDATH      ; Introducir valor medido
CPL    A
MOV    P5,A          ; Visualización valor binario para control
                    ; por medio de UNIDAD LED
MOV    A,ADDATH      ; Normalización en 19,6 mV/Bit
MOV    B,#196
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

```

```

; -----
;                               Complejo de funciones Visualizar
; -----
VISUALIZAR:
; .....
; Conversión en signos 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
; .....
; Signos ASCII en indicador LCD
CALL   CURSOR_HOME
MOV    R7,#4
MOV    R0,#ASCII
A01:  MOV    A,@R0
CALL   WRITE_DREG
CALL   TIEMPO
INC    R0
DJNZ   R7,A01
RET

; -----
;                               Retardo de 70 us, aproximadamente
; -----
TIEMPO:
MOV    R2,#20h
T01:  DJNZ   R2,T01
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
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 de destino 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           ; Siguiete dirección BCD de destino
        DJNZ  R3,K2        ; ¿3ª dirección destino ya alcanzada?
        DJNZ  R4,K0        ; ¿Ya convertidos 16 bits?
        RET

; -----
;           Edición de un texto en segunda línea de display
; -----
OUT_TEXT1:
; Cursor al principio de la 2ª línea
        MOV   A,#40H
        CALL  CURSOR_POS
; .....
; Escribir texto en indicador LCD
        MOV   DPTR,#Text1
        MOV   B,#00H
E02:    MOV   A,B
        MOVC  A,@A+DPTR
        JZ    E01
        CALL  WRITE_DREG
        CALL  TIEMPO
        INC   B
        JMP   E02
E01:    RET

; -----
;           Acordar texto
; -----
Text1 db    "Valor med. en mV"
      db    00H                ; Indicativo de final

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

END

```

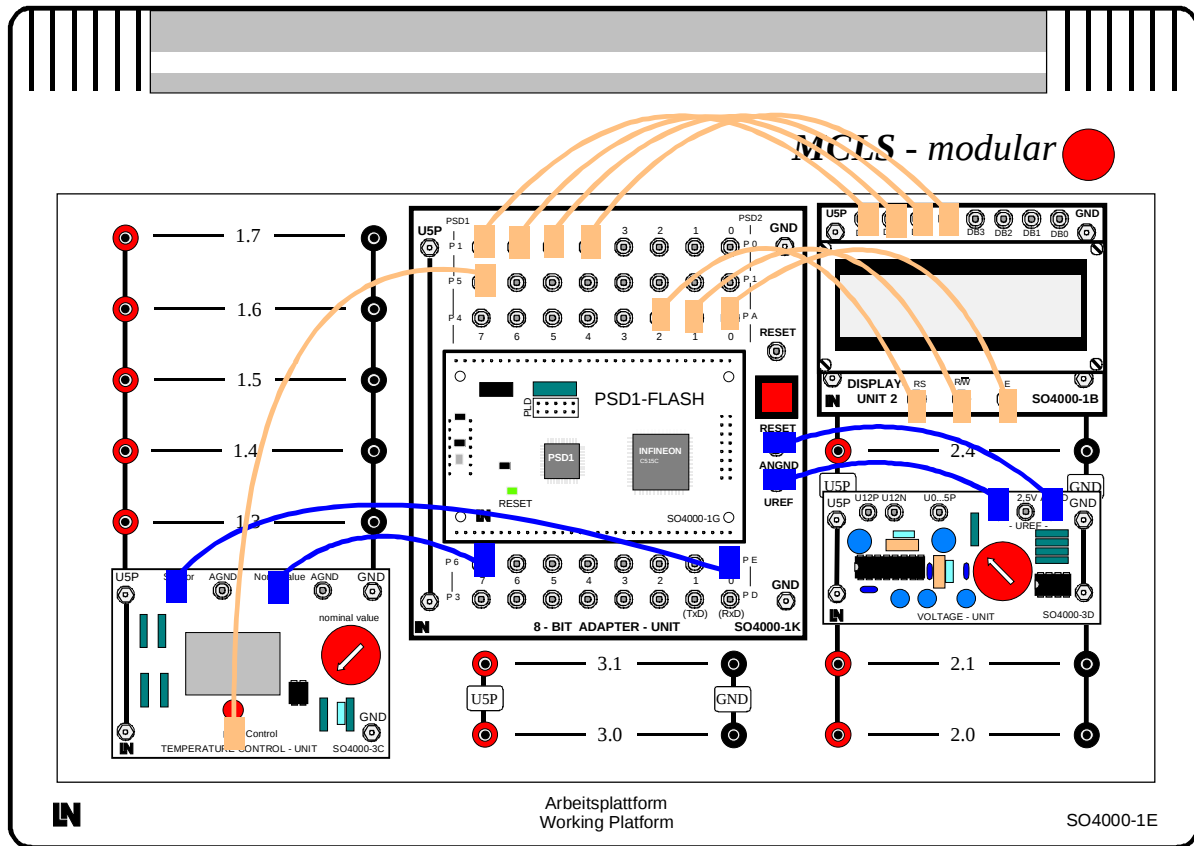


Fig.: 304: Instalación de aparatos del ensayo CMC 3-3B

Ensayo CMC 3-3B

Medir y regular una temperatura. Mediante el convertidor A/D integrado se deberá medir la temperatura de un paralelepípedo de metal y regularla según un valor nominal predeterminado por medio de una calefacción.

¿Qué conocimientos se pretenden transmitir?

- Utilización de diferentes entradas analógicas del convertidor A/D para la medición de los valores nominal y real
- Utilización de varias ventanas de medición para valores nominal y real
- Utilización del sondeo BSY para la conversión analógica-digital con medición única
- Generación de signos ASCII para visualización de texto y de valores medidos

Explicaciones relativas al ejercicio:

Para el control de la temperatura de un paralelepípedo de metal (unidad de control de temperatura) se deberá comparar un valor predeterminado (valor nominal) con el valor de medición de temperatura, conectando o desconectando la calefacción correspondientemente. Puesto que el proceso de calefacción se desarrolla muy inertemente, basta, para la regulación, con una medición alterna en intervalos de un segundo (alternando entre medición de valor nominal y de valor real) incluyendo la comparación de los valores medidos.

Las señales de sensor se miden por medio de dos diferentes entradas analógicas del convertidor A/D. Los valores nominal y real de la temperatura se deberán visualizar simultáneamente en la unidad de LCD.

El margen de la temperatura medible es de 0 a, como máximo, 99 °C. Por medio del potenciómetro de la unidad de temperatura se puede ajustar una tensión de 0 a 2,5 V como valor nominal. El programa deberá utilizarla como valor nominal predeterminado (0 a 99 °C).

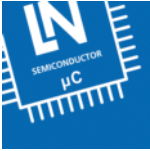
Ejercicios y soluciones

Realice la instalación experimental según la fig.304. Preste atención a una polarización correcta de la tensión de referencia externa en la conexión de la unidad de tensión a la unidad de adaptador de 8 bits.

¿Cómo se realiza la conmutación de las entradas analógicas?

¿Qué comando del ensamblador permite un sondeo del convertidor A/D con respecto al final de la conversión?

Croquice los organigramas para el programa principal y la ISR.



Interpreta las definiciones de texto dadas en el programa cmc33b.asm.

¿Por qué es necesario un pequeño retardo entre la salida de 2 signos ASCII en el indicador de LCD?

Controle las temperaturas medidas por medio de un aparato de medición de temperatura apropiado (p. ej. Tektronix TX3 con elemento térmico K). Observe las indicaciones del manual de instrucciones del aparato, relativas a la precisión, en el correspondiente margen de medición de temperatura.

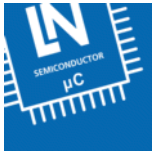
Indicación para la valoración de los resultados de medición:

Las desviaciones en el margen de $\pm 1^{\circ}\text{C}$ se deben considerar como muy pequeñas en caso de mediciones de temperatura sin calibración especial.

Indicaciones importantes:

¡No toque el cuerpo de prueba en la unidad de temperatura mientras que esté conectada su calefacción (LED encendido)!

Para excluir un peligro de lesiones debidas a altas temperaturas en el cuerpo de prueba (temperaturas hasta 90°C), la entrada al portal del módulo solamente debería estar conectada al microcontrolador durante la ejecución del programa. ¡En los demás casos, la entrada al portal se deberá conectar a GND!



Lista de programa CMC 3-3B

```
; *****
; * Proyecto:           Medir y regular una temperatura           *
; * Archivo fuente:     cmc33b.asm                                *
; * Herramientas MC:    as, p2hex, debugger                       *
; * Perfil:             Perfil integrado PSD1-FLASH-C515C-AS      *
; * Recursos de MC:     Unidad LCD en P1.7-P1.4 y P4.0-P4.2      *
; *                   (ver detalles en FCT_LCD.INC),              *
; *                   unidad de tensión para suministro de la    *
; *                   tensión de ref. externa de +5V,            *
; *                   unidad de control de temperatura:          *
; *                   señal de sensor en AN0, valor nominal      *
; *                   predeterminado en AN7, entrada control     *
; *                   de la calefacción (Control) en P5.7        *
; * Editor:             HAG                                       *
; *****

; ----- Descripción de funcionamiento -----
; Para el control de la temperatura de un paralelepípedo de metal en
; la unidad de control de temperatura, el valor nominal es comparado
; cíclicamente con el valor medido, conectando o desconectándose
; correspondientemente la calefacción.
; Las señales de sensor se miden por medio de dos diferentes entradas
; analógicas del convertidor analógico-digital "on chip".
; El valor nominal y el valor real de la temperatura se visualizan
; simultáneamente en la unidad de LCD.
; El margen de la temperatura medida es de 0 a 99 °C,
; el margen del valor nominal se puede ajustar de 0 a 99 °C
; en la unidad de control de temperatura (0 a 2,5 V).
; Un marcador de tiempo controla el proceso de medición. Alternando en
; intervalos de un segundo se miden y se visualizan los valores
; nominal y real. La regulación se limita a una comparación del valor
; nominal con el valor medido. Si el valor medido es inferior al valor
; nominal, se activa la calefacción por medio de P5.7=1.
; En otro caso, la calefacción es desconectada con P5.7=0.
; -----

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

; ----- Algunas definiciones de constantes -----
RESET EQU    0000h       ; Dirección de inicio de reseteo

CT_T2 EQU    -50000      ; Constante de tiempo para Timer 2
                        ; 50ms hasta desbordamiento
RL_T2 EQU    -50000      ; Valor de recarga para Timer 2
                        ; 50ms hasta desbordamiento

; ----- Segmento de datos para variables -----
SEGMENT      DATA
ORG          40H
CTR          DB          ?   ; Contador auxiliar
; Reservar posición de memoria para valores medidos
BCD_MSB      DB          ?   ; MSB del valor medido BCD
BCD_MIB      DB          ?   ; Byte central del valor medido BCD
BCD_LSB      DB          ?   ; LSB del valor medido BCD
BIN_MSB      DB          ?   ; Valor medido binario MSB
BIN_LSB      DB          ?   ; Valor medido binario LSB
```

```

VALOR_NOM    DB    ?    ; Valor nominal de la temperatura de 2 dígitos
VALOR_REAL   DB    ?    ; Valor real de la temperatura de 2 dígitos

ASCII        DB    ?    ; 4 bytes para valores medidos ASCII
             DB    ?
             DB    ?
             DB    ?

; -----

; ----- A partir está el código de programa -----
                SEGMENT      CODE

                ORG    RESET    ; Salto al programa principal a partir de 100H
                LJMP    START

                ORG    002BH    ; Introducir vector de interrupción para Timer 2
                LJMP    ISR_T2    ; en tabla IV

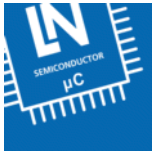
                ORG    100H    ; Programa principal a partir de 100H

; *****
; *          Programa principal Medir y regular una temperatura          *
; *****
START:
    MOV    CTR,#20    ; Inicializar contador auxiliar
    CALL   INIT_T2    ; Timer 2 como marcador de tiempo de 20ms con
                     ; autorecarga
    CALL   INIT_LCD    ; Inicializar indicador LCD
    CALL   OUT_TEXT    ; Editar texto
ST0:
    SJMP   ST0        ; Medir y visualizar continuamente
                     ; Funcionalidad está en ISR para Timer 2
; *****

; -----
;                               Inicializar Timer 2
; -----
INIT_T2:
    CLR    EAL
    MOV    TH2,#hi (CT_T2)    ; Precargar registro contador
    MOV    TL2,#lo (CT_T2)    ; a 50ms
    MOV    CRCH,#hi (RL_T2)    ; Ajustar registro de recarga
    MOV    CRCL,#lo (RL_T2)    ; a 50ms
    MOV    T2CON,#00010001B    ; Marcador de tiempo con autorecarga
    SETB   ET2                ; Autorizar interrupción por T2
    SETB   EAL
    RET

; -----
;                               ISR contiene las funciones Medir, Visualizar, Controlar
; -----
ISR_T2:
    CLR    TF2                ; Reposicionar indicador de
                             ; desbordamiento
    DJNZ   CTR,IS0            ; ¿Contador-1, =0?
; .....
; Cada segundo:
    MOV    CTR,#20            ; Recargar contador auxiliar
    CPL    F0                 ; Complementar indicador de usuario
    JNB    F0,IS1            ; Medir valores real y nominal
                             ; alternando
    CALL   ADU_INIT_AN0      ; Inicializar convertidor A/D para valor

```



```
CALL MEDIR_TEMP          ; real en AN0
                           ; Preparar valor medido para
                           ; visualización
CALL VISUALIZAR_TEMP      ; Visualizar valor medido
CALL TAKE_VALOR_REAL      ; Preparar valor real para comparación
CALL CONTROL              ; Comparación para valores real y
                           ; nominal
ISO: RETI

IS1: CALL ADU_INIT_AN7    ; Inicializar convertidor A/D para valor
                           ; nominal en AN7
CALL MEDIR_NOM            ; Preparar valor medido para
                           ; visualización
CALL VISUALIZAR_NOM       ; Visualizar valor medido
CALL TAKE_VALOR_NOM       ; Preparar valor nominal para
                           ; comparación.
CALL CONTROL              ; Comparación valores real y nominal
JMP ISO

; -----
; Inicializar convertidor A/D para valor de temperatura en AN0
; porque valor medido max. 1000 mV -> margen de medición 0 - 1,25 V
; -----
ADU_INIT_AN0:
ANL  ADCON0,#11100000B
ORL  ADCON0,#00000000B ; Una sola conversión, canal AN0
MOV  ADDATL,#00H       ; iniciar ADC
AD1: JB  BSY,AD1        ; Esperar hasta que finalice conversión
RET

; -----
; Inicializar convertidor A/D para valor nominal en AN7
; porque valor nominal aprox. 2500 mV -> margen de medición 0 - 2,5V
; -----
ADU_INIT_AN7:
ANL  ADCON,#11100000B
ORL  ADCON,#00000111B ; Una sola conversión, canal AN7
MOV  ADDATL,#00H       ; iniciar ADC
AD2: JB  BSY,AD2        ; Esperar hasta que finalice conversión
RET

; -----
; Medir la temperatura incluyendo preparación de valor medido
; -----
MEDIR_TEMP:
CALL SCAL_TEMP           ; scale to virtual measurement
                           ; window 0...1.25V
MOV  B,#49               ; Factor para normalización,
                           ; porque aprox. 4,9 mV/Bit
CALL MEDIR
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:
; 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
        MOV     ADDATH,A
        RET

; -----
;   Medir valor nominal incluyendo preparación de valor medido
; -----
MEDIR_NOM:
        CALL    SCAL_NOM    ; scale to virtual measurment window 0...2.5V
        MOV     B,#40      ; Factor para normalización,
                           ; porque aprox. 11 mV/Bit y margen de rotación
                           ; del elem. ajuste del valor nominal hasta
aprox. 99°C
        CALL    MEDIR
        RET

; -----
;   Scaling to vitual measurement window 0...2.5V
;   Inputs: 10-Bit value in ADDATH and ADDATL
;   Output: scaled 8-Bit value in A
; -----
SCAL_NOM:
; shift 8-Bit value in ADDATH one bit to the left
        MOV     A,ADDATH
        RL      A
        ANL     A,#11111110b
        MOV     B,A
; shift 8-Bit value in ADDATL one bit to the left
; and add to the scaled value in ADDATH
        MOV     A,ADDATL
        RL      A
        ANL     A,#00000001b
        ADD     A,B                ; scaled value in A
        MOV     ADDATH,A
        RET

; -----
;   Aceptación, normalización y conversión BCD del valor medido
; -----
MEDIR:
        MOV     A,ADDATH
        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

; -----
;                               Visualizar valor de temperatura
; -----
VISUALIZAR_TEMP:
    CALL KONV_ASCII         ; Generación carácter ASCII
; Carácter ASCII en indicador LCD
    MOV    A,#0CH           ; Posición en 1ª línea
    CALL   CURSOR_POS       ; Posicionamiento de cursor
    MOV    R7,#2             ; Número de caracteres
    MOV    R0,#ASCII+1
A01:    MOV    A,@R0
    CALL   WRITE_DREG       ; Carácter en indicador LCD
    CALL   WAIT_SHORT       ; Esperar un momento
    INC    R0
    DJNZ   R7,A01
    RET

; -----
;                               Visualización del valor nominal
; -----
VISUALIZAR_NOM:
    CALL   KONV_ASCII       ; Generación de caracteres ASCII
; Caracteres ASCII en indicador LCD
    MOV    A,#4CH           ; Posición en 2ª línea
    CALL   CURSOR_POS       ; Posicionamiento de cursor
    MOV    R7,#2             ; Número de caracteres
    MOV    R0,#ASCII+1
A02:    MOV    A,@R0
    CALL   WRITE_DREG       ; Caracteres en indicador LCD
    CALL   WAIT_SHORT       ; Esperar un momento
    INC    R0
    DJNZ   R7,A02
    RET

; -----
;                               Coger valor nominal de 2 dígitos de la temperatura
; -----
TAKE_VALOR_NOM:
    CALL   TRANS
    MOV    VALOR_NOM,A
    RET

; -----

```

```

;          Coger valor real de 2 dígitos de la temperatura
; -----
TAKE_VALOR_REAL:
    CALL    TRANS
    MOV     VALOR_REAL,A
    RET

; -----
;          Generar formato BCD comprimido para comparación de
;          los valores medidos
; -----
TRANS:
    MOV     A,BCD_MIB
    SWAP    A
    ANL     A,#0F0H
    MOV     B,A
    MOV     A,BCD_LSB
    SWAP    A
    ANL     A,#0FH
    ADD     A,B
    RET

; -----
;          Comparación del valor nominal con el valor real
;          Conectar o desconectar calefacción
; -----
CONTROL:
    MOV     A,VALOR_REAL
    CLR     CY
    SUBB    A,VALOR_NOM ; ¿Valor real < valor nominal?
    JC      C01          ; en caso de sí, conectar calefacción
    CLR     P5.7         ; en caso de no, desconectar calefacción
C02:  RET
; .....
C01:
    SETB    P5.7         ; Conectar calefacción
    JMP     C02

; -----
;          Generación de caracteres ASCII para indicador 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

; -----
;      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 resultados de la rotación
DEC     R0               ; Dirección valor binario MSB
DJNZ    R3,K1            ; Hacer lo mismo para MSB

; .....
MOV     R1,#BCD_LSB      ; Dirección BCD de destino 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

; -----
;      Edición de los textos
; -----
OUT_TEXT:
CALL    OUT_TEXT1
CALL    OUT_TEXT2
RET

OUT_TEXT1:
MOV     A,#00H
CALL    CURSOR_POS
MOV     DPTR,#Text1
CALL    INDICADOR
RET

OUT_TEXT2:
; Cursor al principio de la 1ª línea
MOV     A,#40H
CALL    CURSOR_POS
MOV     DPTR,#Text2
CALL    INDICADOR

```

```

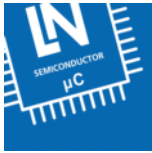
        RET
; .....
; Escribir texto en indicador LCD
INDICADOR:
        MOV     B,#00H
E02:    MOV     A,B
        MOVC    A,@A+DPTR
        JZ      E01
        CALL    WRITE_DREG
        CALL    WAIT_SHORT
        INC     B
        JMP     E02
E01:    RET

; -----
;                               Acuerdos de los textos
; -----
Text1 db      "Valor med.: xx ", 0DFH, "C"
        db      00H                ; Indicativo de final
; -----
Text2 db      "Valor nom.: xx ", 0DFH,"C"
        db      00H                ; Indicativo de final
; -----

; -----
;                               Insertar funciones para unidad 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