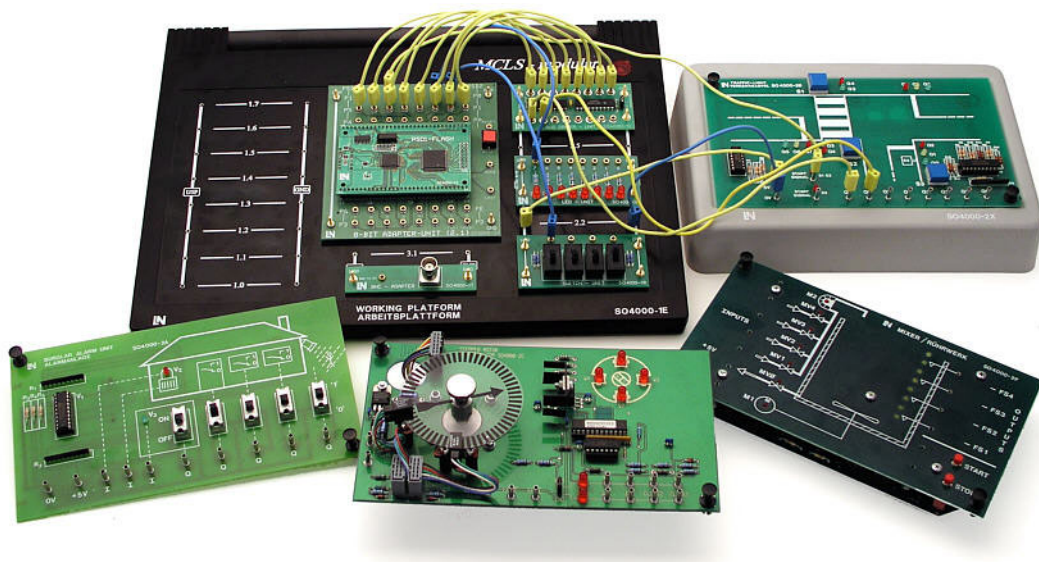




Aplicaciones con el Microcontrolador 8051 C515C

CMC 2



Guía de ejercicios para el estudiante

SH5004-1B

3ra. edición

Autor: equipo de autores APMC Mittweida

Índice

Objetivo de los ensayos.....	1
Resumen de ejercicios.....	1
Instalación del ensayo	1
Introducción	3
CMC 2-1 Introducción en la programación de un microcontrolador C515C.....	6
Fundamentos preliminares	6
Instalación del ensayo CMC 2-1	7
Ejercicio CMC 2-1: Técnicas de programación, análisis de problema, técnicas de diseño (diagramas de flujo), lista de programa.....	8
Soluciones de los borradores y de la lista de programa	19
CMC 2-2 Control de una instalación de alarma	22
Fundamentos preliminares	22
Instalación del ensayo CMC 2-2	22
Ejercicio CMC 2-2: Programe el control de flujo de la instalación de alarma empleando subprogramas	24
CMC 2-3 Control de un semáforo	32
Fundamentos preliminares	32
Instalación del ensayo CMC 2-3	33
Ejercicio CMC 2-3: Programe el control de proceso para el semáforo empleando interrupciones y tablas de valores	34
CMC 2-4 Control de un motor paso a paso	45
Fundamentos preliminares	45
Instalación del ensayo CMC 2-4	46
Ejercicio CMC 2-4: Programe un control de proceso para el motor paso a paso empleando una tabla de valores y subprogramas.	47
CMC 2-5 Control de una instalación ventiladora de túnel	63
Fundamentos preliminares	63
Instalación del ensayo CMC 2-5	63
Ejercicio CMC 2-5: Programe un control de proceso para la instalación ventiladora de túnel empleando tablas de valores y subprogramas.....	65
CMC 2-6 Control de un mecanismo agitador	76
Fundamentos preliminares	76
Instalación del ensayo CMC 2-6	76
Ejercicio CMC 2-6: Programe un control de proceso para el control de un mecanismo agitador empleando subprogramas y bits de memoria.....	78

Objetivo de los ensayos

1. Los aprendices / estudiantes aprenden a aplicar la conversión 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.

Resumen de ejercicios

CMC 2-1: Introducción en la programación de un microcontrolador C515C

En este ensayo, usted conocerá los primeros pasos en la programación de usuario de un microcontrolador C515C. Al mismo tiempo, se familiarizará con la plataforma de desarrollo integrada y el sistema de depuración.

- Operaciones de puerto sencillas

CMC 2-2: Control de una instalación de alarma con el MC C515C

En este ensayo usted aprenderá el principio del control para instalaciones de alarma.

- Empleo de subprogramas

CMC 2-3: Control de una instalación de semáforo con el MC C515C

En este ensayo usted conocerá el control secuencial de una instalación de semáforo.

- Tablas de valores
- Desarrollos continuos y controlador según el requisito
- Empleo de interrupciones

CMC 2-4: Control de un motor paso a paso con el MC C515C

En este ensayo le damos a conocer el control de un motor paso a paso.

- Desarrollos continuos
- Tablas de valores
- „Números grandes“ y registro de 2 Bytes „DPTR“

CMC 2-5: Control de una instalación ventiladora de túnel con el MC C515C

En este ensayo usted conocerá el control de una instalación ventiladora de túnel.

- Tablas de valores
- Subprogramas

CMC 2-6: Control de un mecanismo agitador con el MC C515C

En este ensayo usted aprenderá el control de un mecanismo agitador.

- Técnica de subprograma y entrega de parámetros
- Poner bits de memoria accionando teclas

Instalación del ensayo

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

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

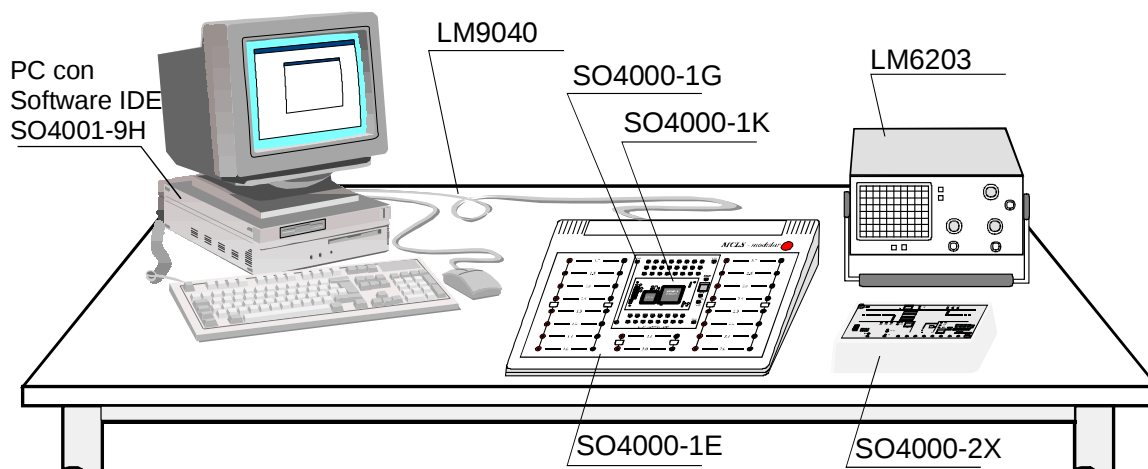


Fig. 01: Instalación en el puesto de trabajo

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

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

Componentes y aparatos necesarios

Can	Denominación	Nº ID
1	Plataforma con módulo de alim. +5V	SO4000-1E
1	Fuente de alim. de enchufe CA 90...230V 45..65Hz, CC 9V 630mA	SO4000-1F
1	Mód. PSD1-Flash Controlador C515C	SO4000-1G
1	Unidad de adaptador de 8 bits	SO4000-1K
1	Unidad LED	SO4000-1P
1	Unidad de teclas	SO4000-1Q
1	Unidad de interruptores	SO4000-1R
1	Adaptador BNC	SO4000-1T
1	Unidad de driver de bus	SO4000-1V
1	Instalación de alarma	SO4000-2A
1	Semáforo	SO4000-2B
1	Motor paso a paso con disco codif.	SO4000-2C
1	Ventilación de túnel	SO4000-2P
1	Agitador (Autoklave)	SO4000-2F
1	Soporte para módulos	SO4000-2X
1	Software IDE para MCLS (D,GB,F,E)	SO4001-9H
1	Osciloscopio incl. puntas de prueba	LM6203
1	Sonda de prueba lógica	LM8101
1	Cable de medición BNC / BNC	LM9034
1	Cable de interfaz serial 9/9 polos	LM9040
6	Línea de medición 2mm 15cm azul	SO5126-5K
9	Línea de medición 2mm 15cm amarillo	SO5126-5M
9	Línea de medición 2mm 30cm azul	SO5126-5S
1	Línea med. Adapt. 4/2mm 50cm rojo	SO5126-6S
1	Línea med. Adapt. 4/2mm 50cm azul	SO5126-6U
12	Línea de medición 2mm 30cm amarillo	SO5126-5U
1	CMC 2 Aplicaciones con el microcontrolador 8051 C515C	SH5019-1B
1	Soporte de almacenamiento para módulos de aplicación	SO4000-1Y

Introducción

Los microcontroladores han llegado a tener una importancia decisiva que no siempre es perceptible a primera vista. Estos sistemas casi siempre trabajan „en segundo plano“, sin ser reconocidos, pero sí de manera muy eficaz. La alta o más alta integración de procesador, RAM, ROM, EEPROM, Flash y de otras funciones permite unos campos de aplicación los cuales hace unos años aún eran inimaginables. Los sistemas de microcontrolador se emplean hoy p.e. en :

- Controles
- Sistemas de regulación
- Sistemas de dirección
- Sistemas de transmisión

El sistema de aprendizaje de microcontrolador modular MCLS-modular® se desarrolló especialmente para la formación en el campo de la „Técnica de microcontrolador“. Permite estudiar la estructura de hardware, así como la programación apoyada por software de un microcontrolador de la serie de construcción 8051.

El entorno de desarrollo integrado de Windows IDE permite, tras una breve iniciación, elaborar, ensayar y optimizar programas de microcontrolador. Para el empleo en el MCLS-modular, nos orientamos en el empleo de depuradores de monitores o simuladores con función de cargador de arranque inicial o de programación para la detección de fallos en programas. Depende del sistema final para el cual deben estar disponibles las correspondientes herramientas, qué herramienta se va a emplear. Con ello, el desarrollo de la programación corresponde a la vida cotidiana industrial.

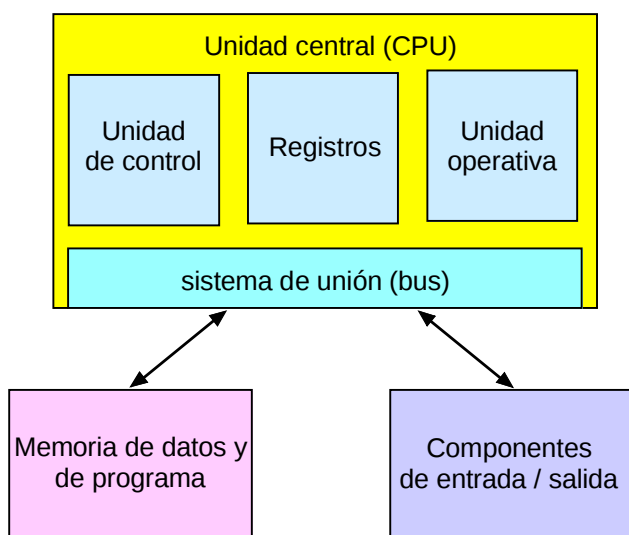
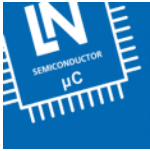


Fig. 02: Estructura fundamental de un ordenador

Los microprocesadores son el elemento central de cualquier microordenador. Como ordenador designamos generalmente una instalación de procesamiento de datos controlada por programas. Qué datos se procesan, o sea el objeto de aplicación, es determinado por la estructura y con ello la capacidad necesaria del microprocesador empleado. Esto se hace más claro cuando se compara un ordenador grande con una tarjeta de chip, ya que en ambos casos se trata de un ordenador.



Originalmente, el microprocesador - el CPU de un sistema de ordenador realizado en un chip - estaba concebido para solucionar cálculos numéricos. Hoy, los microprocesadores se emplean en un amplio entorno técnico, o sea el volumen de funciones se ha ampliado y especializado. Pensando en la variedad de campos de empleo, como los ordenadores de bordo de vehículos modernos, los procesadores de señales en la técnica de comunicación o también los ordenadores personales potentes, podemos realizar la siguiente clasificación:

- Microprocesadores estándar
- Microprocesadores de alto rendimiento
- Microcontroladores
- Procesadores de señales digitales

En este manual de ensayos, nos vamos a dedicar a la programación de microcontroladores generales. El manual de instrucciones CMC 1 se dedica a la estructura y al funcionamiento de los microcontroladores. Los fundamentos indicados en este manual para la programación de microcontroladores y para la programación estructurada se dan por sabidos en el manual CMC 2.

El desarrollo de programa en microcontroladores se realiza de manera próxima al procesador mediante el lenguaje de ensamblaje y el ensamblador correspondiente al microcontrolador o en un lenguaje de alto nivel (C o C++) y un compilador. El ensamblador o el compilador traducen el código fuente escrito en formato de texto a un código binario comprensible para el microcontrolador. Puesto que un microcontrolador representa un microordenador específico, tanto el ensamblador empleado como el compilador deben disponer de los conjuntos de comandos correspondientes al microcontrolador. En otro caso, el código generado no se puede ejecutar o se ejecuta incorrectamente.

La pregunta cuál de las dos herramientas de desarrollo se emplea depende de:

- la disponibilidad de estas herramientas para el MC,
- el requisito de capacidad de tiempo real,
- la compacidad del código generado,
- el volumen de las actividades de programación.

Dentro de los trabajos de desarrollo se emplean otras herramientas adicionales. Los dispositivos de enlace (Linker) insertan las funciones existentes de bibliotecas en el código de programa. Los convertidores generan del archivo de Ensamblador un archivo Intel Hex apto para descargar. Los sistemas de depuración sirven para la puesta en servicio y la optimización de código.

Qué herramientas de desarrollo finalmente se emplean depende de la aplicación por la selección del microcontrolador y los posibles costes de desarrollo. En los microcontroladores de 8 bit es muy habitual trabajar con Ensambladores y Depuradores de monitor. En los microcontroladores de mayor integración, sobre todo en los microcontroladores de 16 y 32 bits se emplean plataformas de lenguaje de alto nivel para el desarrollo de aplicación. Éstas son apoyadas en los microcontroladores más nuevos por Depuradores „on chip“. Sin embargo, estos entornos de programación confortables presuponen la existencia del interfaz necesario al microcontrolador.

Por desgracia, la tendencia hacia microcontroladores cada vez más potentes, con periféricas complejas y confortables sistemas de desarrollo de lenguajes de alto nivel, implica el peligro que su uso sea irreflexivo y suponga un derroche.

Al principio de cada programación debe haber un análisis detallado del problema a solucionar.

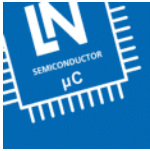
El ejercicio casi siempre verbalmente descrito debe ser convertido en un principio en una forma descriptiva apropiada (p.e. diagrama de flujo). Por medio de escalones intermedios se elabora una secuencia de comandos de microcontrolador, cuyo procesamiento secuencial permite solucionar la tarea. De la secuencia de comandos elaborada de esta manera (programa) se genera un archivo cargable (p.e. archivo Intel-Hex) el cual se carga en el controlador de destino.

A continuación el programa es depurado, es decir se buscan, se detectan y se eliminan los fallos. Para ello se utiliza preferiblemente el servicio por pasos individuales. Por este proceso también se pasa al emplear lenguajes de alto nivel, y los correspondientes compiladores, dispositivos de enlace y herramientas se encargan de gran parte de la complicada actividad de programación.

Partiendo del ejercicio, se elabora en primer lugar un diseño aproximado relativo al hardware necesario, con el objetivo de emplear el mayor número posible de componentes de perifería „on chip“.

Los diagramas de flujo se elaboran desde el nivel de borrador hasta el nivel de comandos (diagrama de flujo de programa DFP). Partiendo del DFP se puede escribir directamente la lista de programa. Todos los comandos se deberán dotar de un comentario breve y de valor informativo. De un modo general se aspira a una programación modular, o sea el programa es dividido en elementos pequeños (subprogramas) (p.e. operaciones de entrada / salida, operaciones aritméticas, bucles de espera). Estos subprogramas constituyen componentes de programa de uso múltiple.

[1] Taschenbuch Mikroprozessortechnik (Libro de bolsillo Técnica de microprocesador), Thomas Beierlein, Olaf Hagenbruch, 3. 3ra. edición 2004, Fachbuchverlag Leipzig



CMC 2-1 Introducción en la programación de un microcontrolador C515C

Realización de una luz de desplazamiento sucesivo sencilla con control de la dirección de desplazamiento

Fundamentos preliminares

Al principio de cada programación debe haber un análisis detallado del problema a solucionar.

El ejercicio casi siempre verbalmente descrito debe ser convertido en una forma descriptiva apropiada (p.e. diagrama de flujo). Por medio de escalones intermedios se elabora una secuencia de comandos de microcontrolador, cuyo procesamiento secuencial permite solucionar la tarea. De la secuencia de comandos elaborada de esta manera (programa) se genera un archivo cargable (p.e. archivo Intel-Hex) el cual se carga en el controlador de destino. A continuación el programa es depurado, es decir se buscan, se detectan y se eliminan los fallos. Para ello se utiliza preferiblemente el servicio por pasos individuales.

Partiendo del ejercicio, se elabora en primer lugar un diseño aproximado, relativo al hardware necesario, con el objetivo de emplear el mayor número posible de componentes de perifería „on chip“.

Los diagramas de flujo se elaboran desde el nivel de borrador hasta el nivel de comandos (diagrama de flujo de programa DFP). Partiendo del DFP se puede escribir directamente la lista de programa. Todos los comandos se deberán dotar de un comentario breve y de valor informativo. De un modo general se aspira a una programación modular, o sea el programa es dividido en elementos pequeños (subprogramas) (p.e. operaciones de entrada / salida, operaciones aritméticas, bucles de espera). Estos subprogramas constituyen componentes de programa de uso múltiple.

Normalmente un microcontrolador es iniciado por medio de una señal RESET externa. Después de aplicar la señal, el contador de programa está en la dirección 0000H y todos los registros de funciones especiales han sido sometidos a una inicialización estándar. En la dirección 0000H debe estar el primer comando del programa. Los primeros comandos de un programa deben llevar a los componentes „on chip“ del microcontrolador al estado inicial deseado (inicialización). Durante la inicialización se cargan p.e. los registros de mando de la perifería „on chip“ (temporizador, interface, convertidor AD, etc.) y se ajusta el sistema de interrupción. A continuación sigue el programa principal.

Instalación del ensayo CMC 2-1

Aparatos y accesorios necesarios

Can	Denominación	Nº ID
1	Plataforma con mód. de alim. +5V	SO4000-1E
1	Fuente de alim. de enchufe CA 90...230V 45..65Hz, CC 9V 630mA	SO4000-1F
1	Módulo PSD1-Flash Controlador C515C	SO4000-1G
1	Unidad de adaptador de 8 bits	SO4000-1K
1	Unidad LED	SO4000-1P
1	Unidad de teclas	SO4000-1Q
1	Software IDE para MCLS (D,GB,F,E)	SO4001-9H
1	Cable de interfaz serial 9/9 polos	LM9040
1	Línea de medición 2mm 15cm azul	SO5126-5K
8	Línea de medición 2mm 15cm amarillo	SO5126-5M
1	CMC 2 Aplicaciones con el microcontrolador 8051 C515C	SH5019-1B

Instalación de aparatos

¡Separe la alimentación de corriente del MCLS-modular !

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

Encaje la unidad de adaptador de 8 bit 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 necesarias 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 de 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.

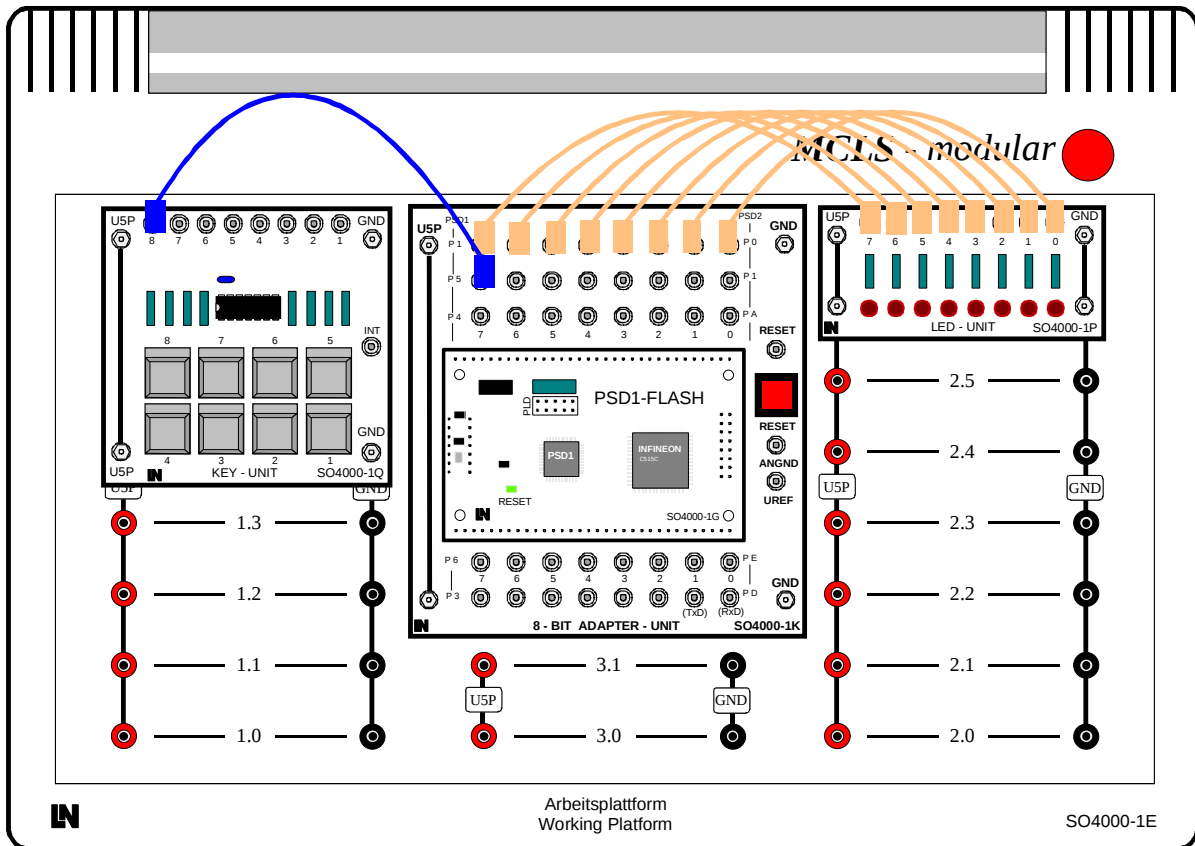


Fig. 101: Instalación de aparatos del ensayo CMC 2-1

Ejercicio CMC 2-1:

Técnicas de programación, análisis de problema, técnicas de diseño (diagramas de flujo), lista de programa

Objetivo :

- Fundamentos generales de la programación
- Programación continua
- Distribuidor
- Técnica de bucles
- Técnica de subprogramas
- Análisis de problema de un ejercicio
- Plan borrador como diagrama de flujo
- Plan de algoritmos como diagrama de flujo
- Plan de programa como diagrama de flujo a nivel de comandos
- Símbolos habituales de diagramas de flujo
- Elaboración de una lista de programación

Ejercicio:

Elabore el borrador de concepto, de algoritmos y de programa (como diagrama de flujo respectivamente) para una luz de desplazamiento sucesivo. Realice previamente un análisis de problemas.

La luz de desplazamiento sucesivo se debe indicar en una línea de LED con 8 LED. La salida y el control de los LED se realiza por medio del puerto P1. La dirección de desplazamiento de la luz de desplazamiento sucesivo se cambiará por medio de un pulsador en el puerto P5.7. El tiempo para los cambios entre los distintos LED deberá ser de aproximadamente 131 ms. Escriba la lista de programa (texto fuente) partiendo del borrador de programa.

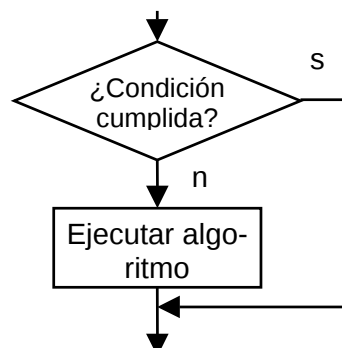
Conocimiento básico :

- Fundamentos generales de la programación
 - ⇒ Configurar los programas de manera sencilla, clara, fiable y fácilmente aplicable
 - ⇒ La división en programas parciales es conveniente (programación modular)
 - ⇒ Documentar de manera clara las prestaciones y los interfaces de los programas parciales
 - ⇒ Estandarizar los interfaces
 - ⇒ Mantener el volumen de programación dentro de un margen económico
 - ⇒ Tener en cuenta la economía de hardware (espacio en memoria, recursos „on-chip“)

NOTA : Intente conseguir siempre una programación estructurada ya que aumenta la claridad y facilita la detección de errores.

ATENCIÓN : *Emplée la programación „Top-Down“, o sea una planificación de programa que empiece con las propiedades principales del sistema (nivel más alto) y continúe con los detalles (nivel bajo).*

- Programación continua
 - ⇒ Comandos secuenciales
- Distribuidor
 - ⇒ Ramificación de programa
 - ⇒ es realizada por una operación de decisión



- Técnica de bucles

- ↳ Utilizar bucles de programa

- ⇒ Bucle de programa :
Ejecutar varias veces una secuencia de comandos con operandos eventualmente cambiados
 - ⇒ Componentes de un bucle de programa
 - Inicialización del criterio de bucle (p.e. registro)
 - Procesamiento de la función útil
 - Modificación del criterio de bucle
 - Consulta final

NOTA : Los bucles de programa reducen el volumen de programa y de memoria.

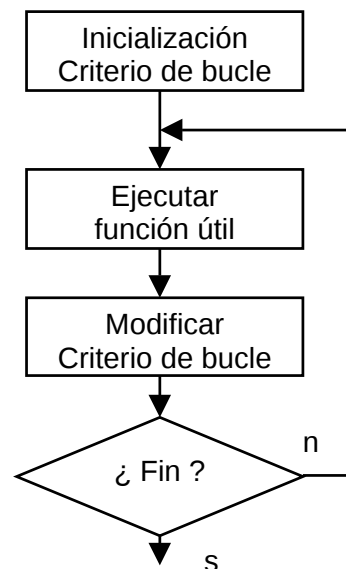


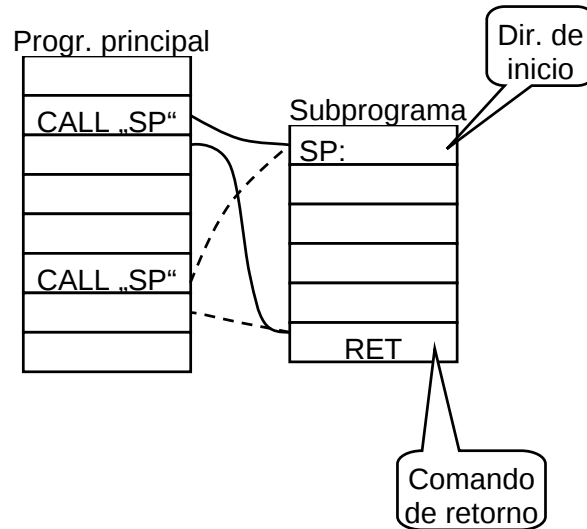
Fig.102: Bucle típico de programa

- Técnica de subprograma

- ↳ Utilización de subprogramas

- ⇒ Subprograma (SP) :
 - Secuencia de comandos que se necesita muchas veces en el programa. (Subrutina, procedimiento)
 - Se accede mediante el programa principal
 - La dirección donde se debe continuar con el programa principal después del SP, se archiva en el Stack (memoria de apilamiento)
 - Es finalizado con un comando de retorno especial (p.e. RET)

Los subprogramas permiten una configuración de programa clara, así como la reducción del volumen de programación y de memoria.



ATENCIÓN : Los contenidos de los registros empleados en el SP se deberían guardar en el Stack (memoria de apilamiento) tras entrar en el SP y recuperarse de allí antes del retorno del SP

- Análisis de problema de una tarea
 - ⇒ Objetivo : De la tarea verbal se elabora una planificación aproximada del hardware necesario con el objetivo de utilizar el mayor número de componentes de perifería „on chip“ posible.
 - ⇒ Se debe contestar a las siguientes preguntas:
 - ⇒ ¿Es posible dividir el problema en problemas parciales?
 - ⇒ ¿Qué se debe controlar, planificar, calcular?
 - ⇒ ¿Qué perifería „on chip“ se necesita? ¿Presenta una potencia suficiente?
 - ⇒ ¿Se necesitan interrupciones?
 - ⇒ ¿Cuántas entradas/salidas se necesitan?
 - ⇒ ¿Son de esperar estados críticos en cuanto al tiempo?
 - ⇒ ¿Existen soluciones ya conocidas que podrían ser utilizadas?
 - ⇒ ¿Qué estado inicial debe presentarse?
- Borrador de proyecto como diagrama de flujo
 - ⇒ Descripción aproximada del desarrollo de procesamiento en un orden cronológico
 - ⇒ Para cada bloque se introduce el procedimiento a realizar (¿Qué se debe hacer?)

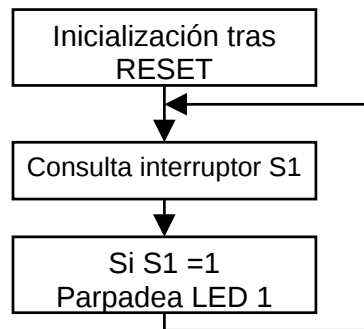


Fig. 103: Diagrama de flujo a nivel de proyecto

ATENCIÓN : ¡ El borrador de proyecto se debe elaborar independientemente del procesador!

- Borrador de algoritmos como diagrama de flujo (independiente del procesador)
 - ↳ Más detalles relativos al nivel de borrador hasta obtener instrucciones de procesamiento fáciles de realizar.
 - ↳ Debe estar determinado para cada bloque qué acciones se deben realizar en qué orden (sin entrar en detalles de procesador)
 - ↳ Debe ser reconocible cómo se debe realizar la operación necesaria.
 - ↳ Utilizable como diagrama de flujo de programa (DFP)

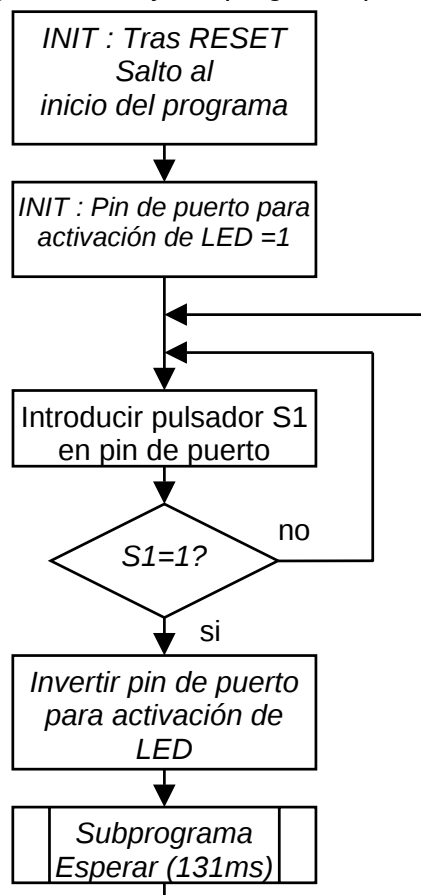


Fig. 104: Diagrama de flujo a nivel de algoritmos

ATENCIÓN : ¡El borrador de algoritmos se debe elaborar independientemente del procesador !

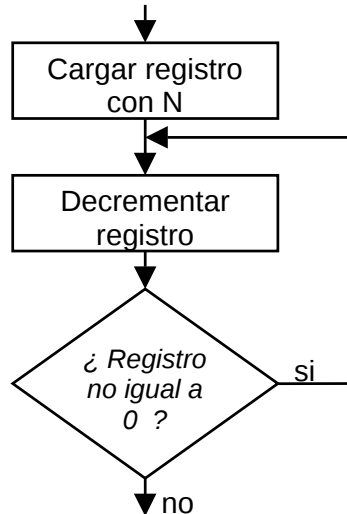


Fig. 105: Subprograma Esperar (diagrama de flujo a nivel de algoritmos)

- Borrador de programa como diagrama de flujo a nivel de comandos (dependiente de procesador)
 - ↗ Más detalles relativos al nivel de algoritmos empleando el conjunto de comandos
 - ↗ Disolución de todos los algoritmos
 - ↗ Como máximo 3 comandos por bloque
 - ↗ El programa se puede deducir directamente
 - ↗ Diagrama de flujo de programa (DFP)

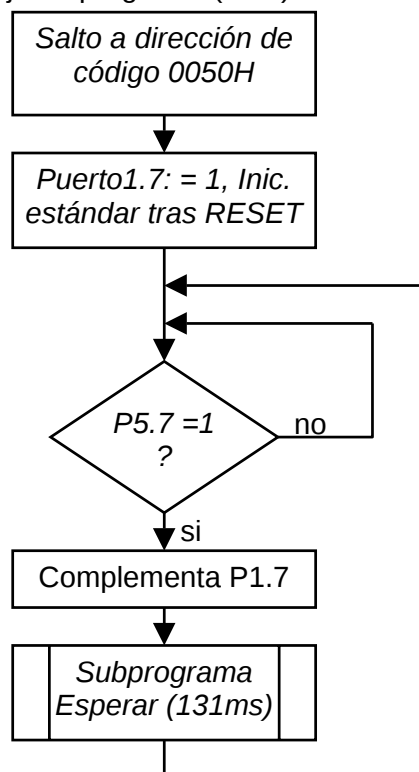


Fig. 106: Diagrama de flujo a nivel de comandos

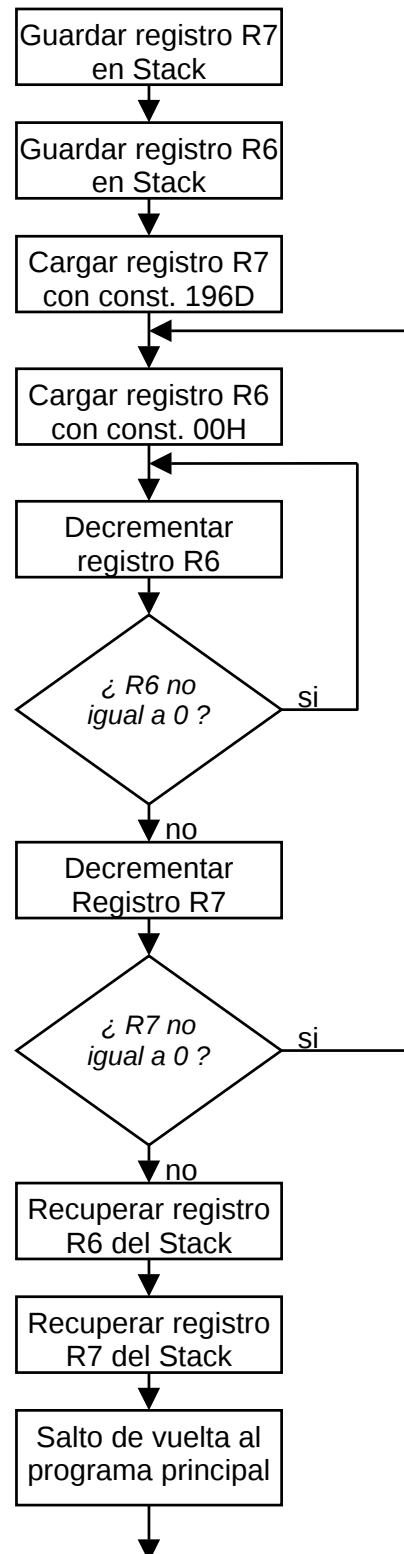
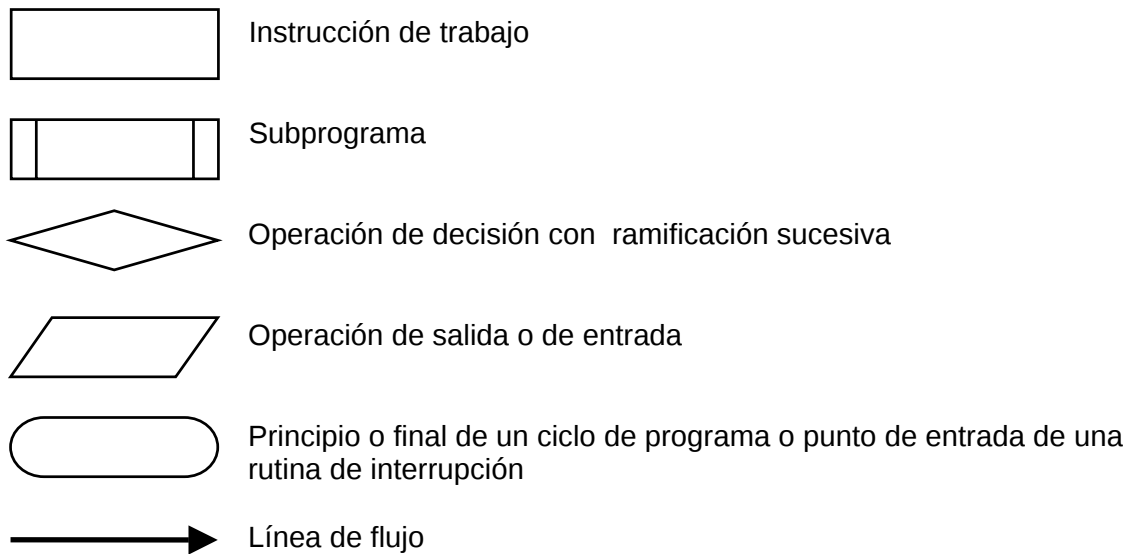


Fig. 107: Subprograma Esperar (diagrama de flujo a nivel de comandos)

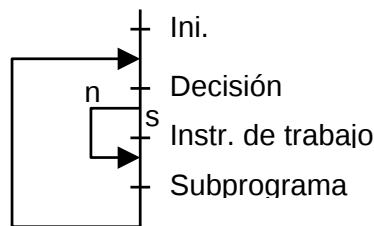
ATENCIÓN : ¡ El borrador de programa se debe elaborar dependientemente del procesador!

ATENCIÓN : ¡ Cada bloque debe contener como máximo 3 comandos !

- Símbolos habituales de diagramas de flujo



Otro método igualmente habitual aparte del método de líneas de flujo presentado es el método de líneas de dirección.



Método de líneas de dirección

La secuencia de borrador de proyecto, borrador de algoritmos y borrador de comandos corresponde al principio del diseño „Top-Down“.

• Elaboración de una lista de programa

↳ Estructura

- ⇒ Cabeza de programa :
 - Título de programa
 - Nombre del programador
 - Número de versión
 - Fecha de elaboración
- ⇒ Descripción breve :
 - Función
 - Condiciones de entrada y salida
- ⇒ Instrucciones de Ensamblador :
 - Archivos Include especiales
- ⇒ Asignaciones de valores :
 - Asignaciones de símbolo
 - Asignaciones de direcciones
- ⇒ Programa principal

- Comandos
- Comentarios
- El último comando es el salto al principio del programa principal o a sí mismo
- ⇒ Subprogramas
 - Eventual cabeza de subprograma (como cabeza de programa)
 - Archivar el contenido de los registros empleados en el Stack
 - Comandos
 - Comentarios
 - Recuperar el contenido guardado de los registros del Stack
 - Retorno al programa principal
- ⇒ Tablas de constantes
 - Valores binarios constantes como fuente de datos

La utilización constante de asignaciones de valores al principio de la lista permite la modificación rápida de constantes y direcciones en todo el programa.

ATENCIÓN : *¡ Los comentarios deben explicar la operación y no el comando !*

- ⇒ Reglas de escritura
- ⇒ Observar Ensamblador

Procedimiento :

- Conteste, partiendo del problema, las preguntas del análisis de problema.
- Elabore el diagrama de flujo para el borrador de proyecto.
 - ⇒ Clasificar la tarea en problemas parciales (resultado del análisis de problema)
 - ⇒ Dibujar un borrador según el método de líneas de dirección
- Elabore el diagrama de flujo para el borrador de algoritmos
 - ⇒ Convertir los problemas parciales del borrador de proyecto en algoritmos generales
 - ⇒ Definición general de los componentes de perifería
 - ⇒ Determinar el componente de perifería para la edición en la línea LED
 - ⇒ Determinar el componente de perifería para la introducción del estado de interruptor
 - ⇒ Determinar generalmente las memorias intermedias (p.e. registros, memoria de datos)
 - ⇒ Determinar las ramificaciones, los bucles y los subprogramas
 - ⇒ Dibujar el borrador de programa principal y subprograma según el método de líneas de flujo

- Elabore el diagrama de flujo para el borrador de programa (para C515C)
 - ⇒ Convertir todos los algoritmos del borrador de algoritmos en comandos concretos
 - ⇒ Definición de los componentes de perifería
 - ⇒ Designar componentes de perifería para la edición en línea LED
 - ⇒ Designar componentes de perifería para la introducción del estado de interruptor
 - ⇒ Determinar memorias intermedias (designación de registros, dirección de memoria de datos)
 - ⇒ Descodificar ramificaciones, bucles, subprogramas en todos los algoritmos
 - ⇒ Dibujar un borrador de programa principal y subprograma según el método de líneas de flujo
- Elabore del diagrama de flujo del borrador de programa la lista de programa
 - ⇒ Elabore el proyecto „CMC2-1“, utilice el Perfil integrado PSD1_C515C_AS para el microcontrolador C515C con las herramientas Ensamblador, Convertidor y Depurador. Inserte un nuevo archivo de texto fuente „cmc21.asm“.
 - ⇒ Introduzca la lista de programa según la estructura general en el archivo de texto fuente. Observe las reglas del Ensamblador.
 - ⇒ Guarde los contenidos de los registros empleados en el subprograma en el Stack
- Cree de este archivo el archivo Intel-Hex y cárguelo en el controlador.
- Ensaye su programa mediante el Depurador
- Corrija los errores de programa hasta que el programa se ejecute sin errores

Preguntas :

¿Qué se entiende por programación „Top-Down“?

¿Qué secuencia de borradores corresponde a este principio?

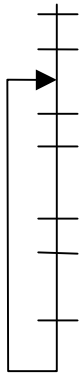
¿Qué dependencia del procesador existe respectivamente en el borrador de proyecto, el borrador de algoritmos y el borrador de programa?

¿Cuántos comandos debe haber como máximo en un bloque ?

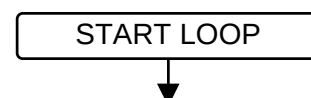
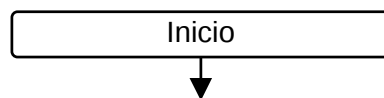
¿Qué métodos para la elaboración de diagramas de flujo conoce usted?

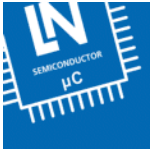
Soluciones de los borradores y de la lista de programa

Borrador de proyecto

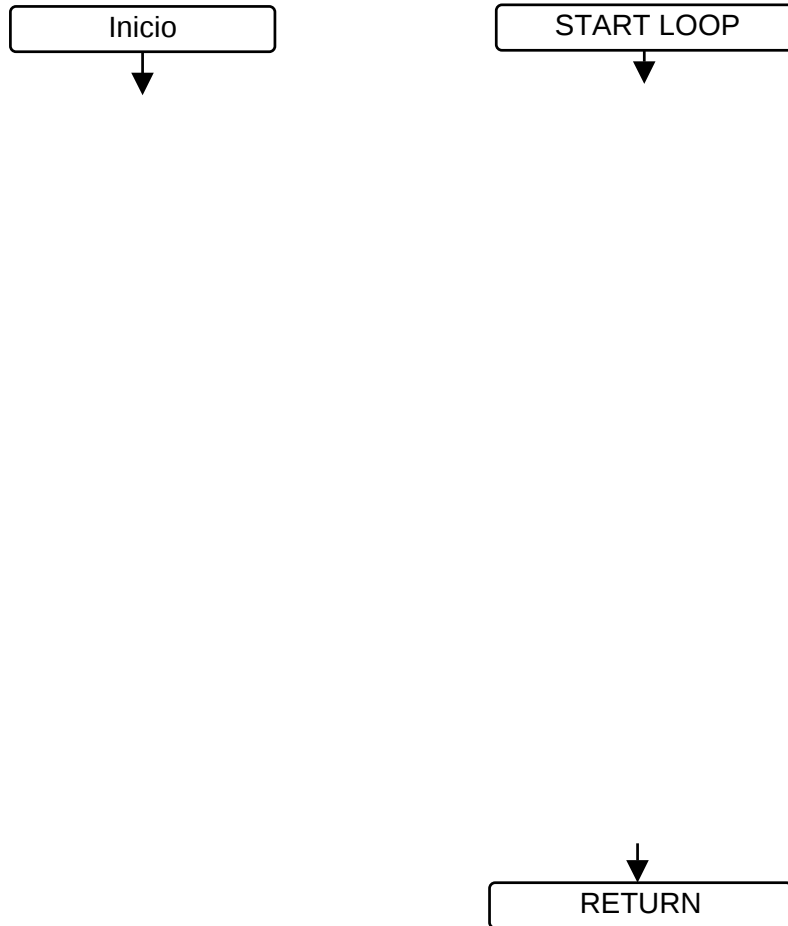


Borrador de algoritmos





Borrador de programa



Lista de programa

```

; *****
; * Proyecto:      Luz LED de despl.suc.del 23.09.2005      *
; * Arch. Proy.:   CMC2-1                                    *
; * Arch. fuente:  cmc21.asm                                  *
; * MC-Tools:      as, p2hex, debugger                       *
; * Perfil:         Perfil integrado PSD1_C515C_AS           *
; * Recursos MC:   LED-Unit compl. en P1                     *
; * Encargado:                                           *
; *****
;                               Descripción de funcionamiento
; Todos los LED de la unidad LED están conectados al
; puerto Port1 completo (P1.0 - P1.7) del MC.
; Una tecla de la unidad de teclas se conecta al
; pin de puerto Portpin P5.7.
; Por medio del nivel lógico ajustado mediante el conmutador
; deslizante se determina la dirección de la luz de desplazamiento
; sucesivo.
; La consulta del interruptor se realiza por medio del
; comando de ensayo de bit 'JB Bit,ziel'.
; El patrón de bits para la activación del puerto se desplaza
; por medio de los comandos de rotación 'RL A' o 'RR A'.
; -----
;                               Acuerdos necesarios para el Ensamblador
;
; CPU      80515      ; Selección MC
; INCLUDE  c515c.inc   ; Definición SFR (registro de funciones
;                       ; especiales)
; SEGMENT  code        ; Segmento de código
; ORG      0000H       ; Primer comando a partir de 0000H
; *****
; *   PROGRAMA PRINCIPAL Luz de desplazamiento sucesivo LED   *
; *****

; *****
;                               Subprograma LOOP, aproximadamente 131 ms
; -----

```

END

CMC 2-2 Control de una instalación de alarma

Control de una instalación de alarma con el MC C515C empleando subprogramas

Fundamentos preliminares

El modelo de una instalación de alarma de edificios está equipado con

- un interruptor general para la activación de la instalación de alarma,
- un contacto de puerta y dos contactos de ventana,
- un sensor para la vigilancia del entorno,
- un LED para el estado de la instalación y otro como alarma, así como con
- un zumbador como alarma acústica.

Con el interruptor general se conecta la instalación de alarma, siendo vigilados todos los sensores y contactos. Un LED de control de color verde indica la “disposición de servicio”.

Si se activa uno (o varios) de los contactos en la puerta y en las ventanas, deberá parpadear el LED rojo y sonar la alarma acústica. La instalación da la alarma hasta que usted confirma, o sea hasta que desconecta el interruptor general de la instalación de alarma. Sin embargo, la alarma acústica se desconecta como más tarde después de 10 segundos.

En cuanto se active el sensor de entorno, deberá parpadear el LED de alarma de color rojo. La alarma de entorno se desactiva cuando usted desconecta el interruptor general o cuando el intruso se aleja.

Instalación del ensayo CMC 2-2

Aparatos y accesorios necesarios

Can	Denominación	Nº ID
1	Plataforma con mód. de alim. +5V	SO4000-1E
1	Fuente de alim. de enchufe CA 90...230V 45..65Hz, CC 9V 630mA	SO4000-1F
1	Módulo PSD1-Flash Controlador C515C	SO4000-1G
1	Unidad de adaptador de 8 bits	SO4000-1K
1	Instalación de alarma	SO4000-2A
1	Unidad de driver de bus	SO4000-1V
1	Adaptador BNC	SO4000-1T
1	Soporte para módulos	SO4000-2X
1	Software IDE para MCLS (D,GB,F,E)	SO4001-9H
1	Osciloscopio incl. puntas de prueba	LM6203
1	Sonda de prueba lógica	LM8101
1	Cable de medición BNC / BNC	LM9034
1	Cable de interfaz serial 9/9 polos	LM9040
8	Línea de medición 2mm 15cm amarillo	SO5126-5M
9	Línea de medición 2mm 30cm azul	SO5126-5S
1	Línea de medición 2mm 30cm amarillo	SO5126-5U
1	CMC 2 Aplicaciones con el microcontrolador 8051 C515C	SH5019-1B

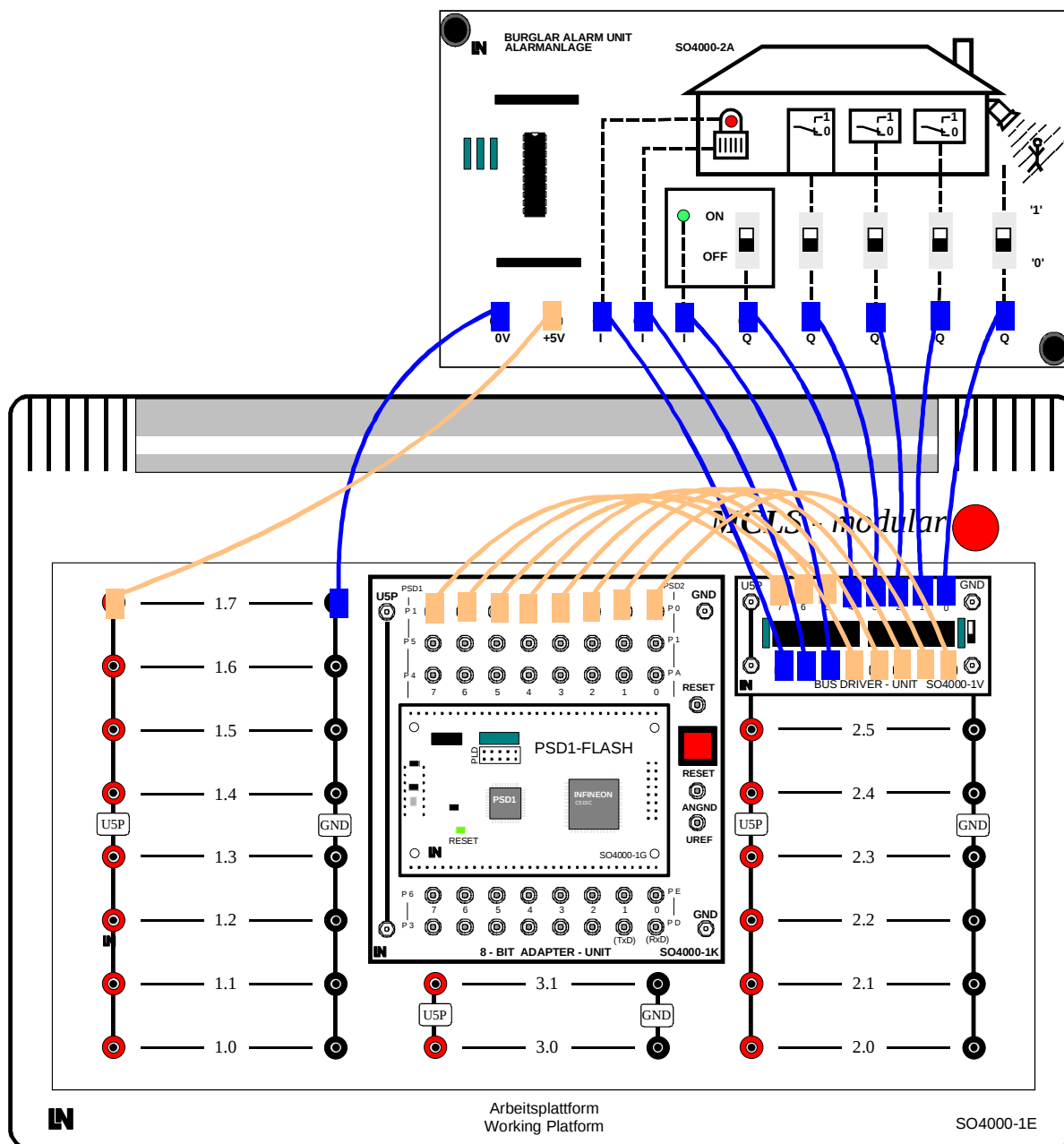


Fig.: 201: Instalación de aparatos del ensayo CMC 2-2

Instalación de aparatos

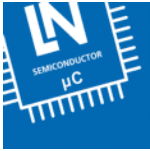
¡Separe la alimentación de corriente del MCLS-modular !

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

Encaje la unidad de adaptador de 8 bit 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 de 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.

Ejercicio CMC 2-2:

Programa el control de flujo de la instalación de alarma empleando subprogramas

Paso 1:

Analice el problema y elabore un borrador de proyecto. Emplee la técnica de subprogramas para el control de tiempo del LED de alarma rojo intermitente.

Paso 2:

Elabore el plan de programa para el subprograma "DELAY" a partir de la dirección 40h. El subprograma DELAY retardará el proceso medio segundo y por lo demás no tendrá otras funciones.

Nota: Reflexione cómo se obtiene medio segundo por medio de comandos de ensamblador extremadamente rápidos en el campo de μs (microsegundos). Observe también los diferentes tiempos de ciclo de los distintos comandos.

Documente su análisis de problema.

Compruebe la corrección de su programación con el osciloscopio.

Paso 3:

Elabore, empleando "DELAY", el borrador de programa para el subprograma "FLASH". El subprograma FLASH hace que el LED de alarma rojo parpadée con 1 Hz.

Paso 4:

Elabore ahora el programa principal que controla la instalación de alarma completa, tal y como se describe en los fundamentos preliminares.

Soluciones de los borradores y de la lista de programa

Descripción del hardware:

Seleccionar procesador: C515C

Cpu: 80515

Actuador / Sensor	Puerto	Identificador
Sensor de movimiento	P1.0	Sensor
Contacto de ventana 1	P1.1	
Contacto de ventana 2	P1.2	
Contacto de puerta	P1.3	
Interruptor	P1.4	On
LED listo para funcionar	P1.5	Ready
Zumbador de alarma	P1.6	Sound
LED de alarma	P1.7	Alarm

Nota:

Observe la carga admisible de los puertos del microcontrolador empleado. El C515C puede excitar, para Active Low, aproximadamente 16mA frente a VCC, y para Activ High, solamente 0,5mA frente a GND. Utilice para la excitación de la perifería la unidad de driver de bus. Véase más informaciones relativas al microcontrolador en la descripción original del microcontrolador Infineon C515C o en el Internet.

Análisis de problema para el subprograma DELAY:

Frecuencia de reloj:

Ciclo de máquina:

0,5 seg. corresponden a:

Comando de salto seleccionado

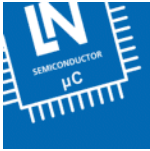
para el bucle de contador:

Número de CM por comando:

Núm. de bucles de contador:

Anchura de registro:

Solución propuesta:



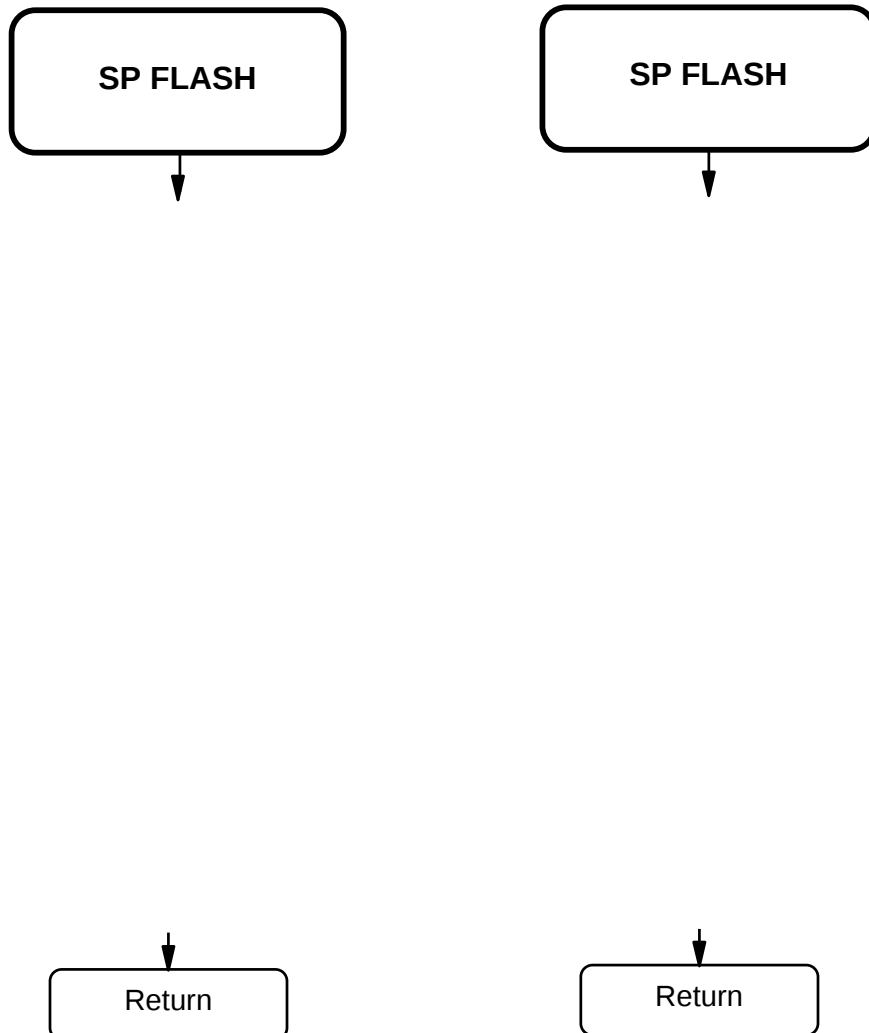
Borrador de proyecto

**INST. DE ALARMA
CMC 2-2**



Borrador de programa:





Lista de programa

```

;*****
;* Proyecto: "Control de una instal. de alarma",cmc22.asm      *
;* Fecha:      10.12.99                                       *
;* Tools:      as, p2hex, debugger                             *
;* Encargado: Grosse                                           *
;*                                                         *
;* Interface de hardware                                       *
;* P1.0   conectado al sensor de movimiento                    *
;* P1.1   conectado al contacto de ventana 1                   *
;* P1.2   conectado al contacto de ventana 2                   *
;* P1.3   conectado al contacto de puerta                      *
;* P1.4   conectado al interruptor de puesta en marcha         *
;* P1.5   conectado al LED Listo para funcionar               *
;*         por medio de unid.driver de bus (sin inversión)     *
;* P1.6   conectado al zumbador de alarma                      *
;*         por medio de unid.driver de bus (sin inversión)     *
;* P1.7   conectado al LED de alarma                           *
;*         por medio de unid.driver de bus (sin inversión)     *
;*****

cpu      80515          ; Seleccionar procesador -> C515C
include  c515c.inc      ; Insertar registro de funciones
                        ; especiales

;***** Definir variables específicas de programa *****

        segment data
        ORG 2Fh
AlarmTime db      ?      ; Contador de tiempo para alarma acústica
                        ; de 10 segundos
                        ; Entradas y salidas en Port1:
Alarm      bit      P1.7  ; LED de alarma en P1.7
Sound      bit      P1.6  ; Zumbador de alarma en P1.6
Ready      bit      P1.5  ; LED "Listo para funcionar" en P1.5
On          bit      P1.4  ; Interruptor general en P1.4
Sensor     bit      P1.0  ; Iniciador de entorno en P1.0

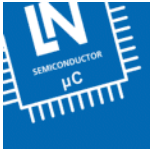
;////////////////////////////////////

        segment code
        ORG 0h
ajmp      MAIN          ; Salto al programa principal

;***** Subprogramas ( SP ) *****
        ORG 40h

;=====
; SP:          DELAY
; Objetivo:    Retardar medio segundo
; Acceso:      Por medio de SP FLASH
;=====
; Información de fondo:
; Microcontrolador C515C con reloj de 12 MHz
; ==> 1 µs por ciclo de máquina
; ==> 0,5 segundo corresponde a 500.000 ciclos de máquina
;-----

```



```
;=====
;      SP:          FLASH>[Alarma]
;      Objetivo:    Hacer parpadear LED de alarma
;      Acceso:      Mediante MAIN
;      Subprograma: DELAY
;=====
```

```
;***** Programa principal ( MAIN ) *****
;=====
;      MAIN:        Programa principal "ALARM CMC2-2"
;      Objetivo:    Controlar toda la instalación de alarma
;      Subprograma: FLASH>[Alarma]
;      Entradas:    Port1.0-4
;      Salidas:     Port1.5-7
;=====
```

;----- Comprobar CONTACTOS Puerta/Ventanas -----

;----- SENSOR Comprobar entorno -----

END

CMC 2-3 Control de un semáforo

Control de un semáforo con el MC C515C empleando interrupciones y tablas de valores.

Fundamentos preliminares

En este ensayo contemplamos el principio de un control para semáforos de tráfico. El modelo del cruce comprende una calle central, una calle secundaria con bucle de iniciador y un paso de peatones con teclas de solicitud.

Mediante un control sencillo, estos semáforos se conectan repetidamente en un orden determinado

Rojo - Rojo/Amarillo - Verde - Amarillo - Rojo
(ciclo continuo).

Por medio de un programa ampliado, la instalación de semáforo se conectará de manera flexible y „conforme al tráfico“ según los requisitos.

La siguiente tabla describe 6 fases de semáforo. Se indica el estado de conexión para cada semáforo, la duración de todas las fases y la conexión para los bits de salida.

Fase N°:		1	2	3	4	5	6
Duración [sec]:		1	1	4	1	1	8
Port:	Conexión:						
5.7	Q7 Rojo Calle central		X	X	X	X	
5.6	Q6 Amar. Calle central	X				X	
5.5	Q5 Verde Calle central						X
5.4	Q4 Rojo Peatones	X	X		X	X	X
5.3	Q3 Verde Peatones			X			
5.2	Q2 Rojo Calle secund.	X	X			X	X
5.1	Q1 Amar. Calle sec.		X		X		
5.0	Q0 Verde Calle sec.			X			

Instalación del ensayo CMC 2-3

Aparatos y accesorio necesarios

Can	Denominación	Nº ID
1	Plataforma con módulo de alim. +5V	SO4000-1E
1	Fuente de alim. de enchufe CA 90...230V 45..65Hz, CC 9V 630mA	SO4000-1F
1	Módulo PSD1-Flash Controlador C515C	SO4000-1G
1	Unidad de adaptador de 8 bits	SO4000-1K
1	Unidad LED	SO4000-1P
1	Unidad de interruptores	SO4000-1R
1	Adaptador BNC	SO4000-1T
1	Unidad de driver de bus	SO4000-1V
1	Semáforo	SO4000-2B
1	Soporte para módulos	SO4000-2X
1	Software IDE para MCLS (D,GB,F,E)	SO4001-9H
1	Osciloscopio incl. puntas de prueba	LM6203
1	Sonda de prueba lógica	LM8101
1	Cable de medición BNC / BNC	LM9034
1	Cable de interfaz serial 9/9 polos	LM9040
2	Línea de medición 2mm 15cm azul	SO5126-5K
8	Línea de medición 2mm 15cm amarillo	SO5126-5M
1	Línea de medición 2mm 30cm azul	SO5126-5S
1	Línea de medición 2mm 30cm amarillo	SO5126-5U
1	CMC 2 Aplicaciones con el microcontrolador 8051 C515C	SH5019-1B

Instalación de aparatos

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

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

Encaje la unidad de adaptador de 8 bit 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 de 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.

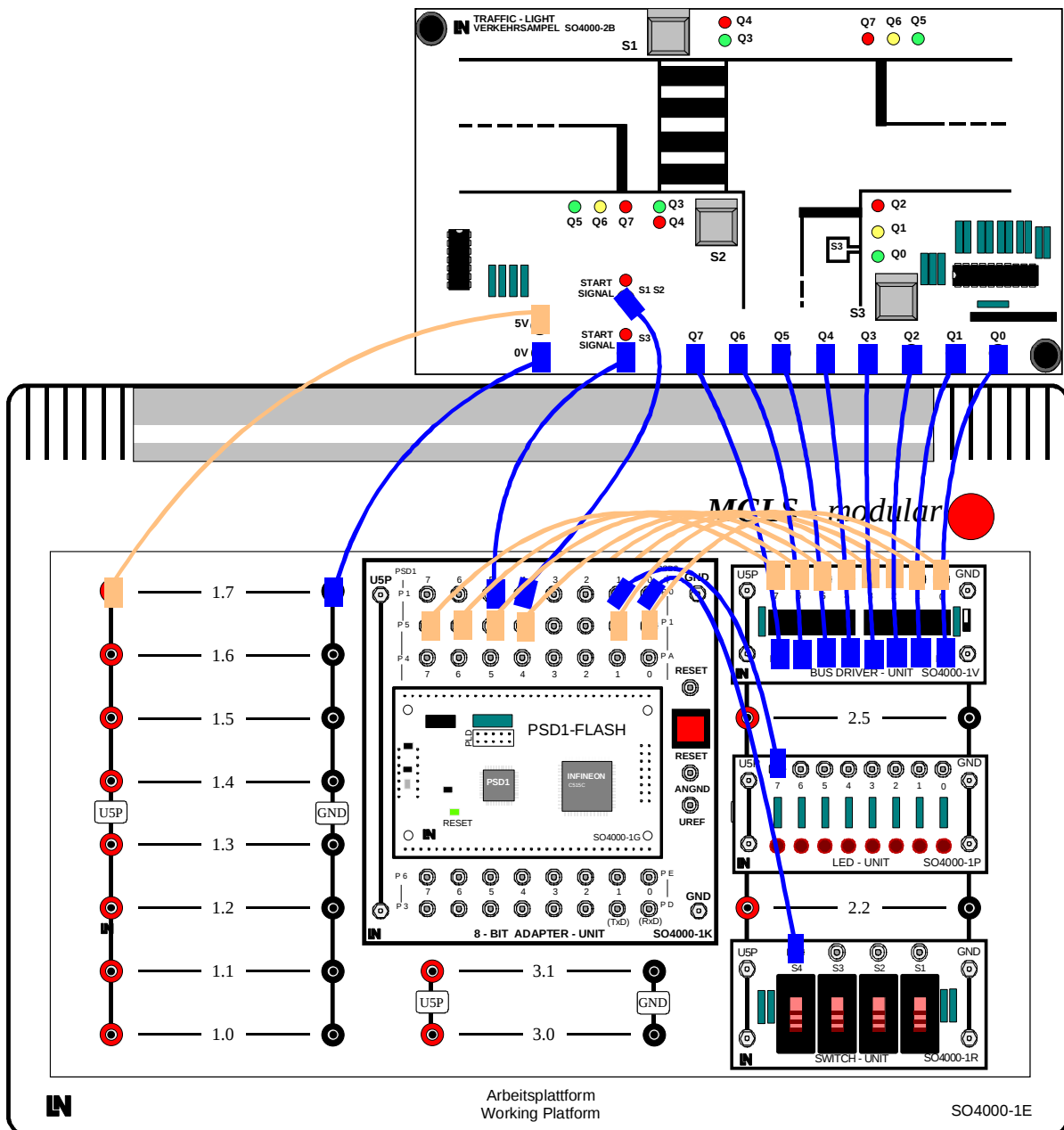


Fig.: 301 Instalación de aparatos del ensayo CMC 2-3

Ejercicio CMC 2-3:

Programa el control de proceso para el semáforo empleando interrupciones y tablas de valores

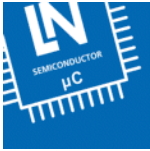
Paso 1:

Realice un análisis de problema y conteste a las siguientes preguntas.

¿Cuáles son los valores de salida de las distintas fases de semáforo como secuencia de bits, en representación Hex y como número decimal? ¡Rellene la siguiente tabla de valores!

Fase	Q7	Q6	Q5	Q4	Q3	Q2	Q1	Q0	hex	dez
1										
2										
3										
4										
5										
6										

¿Cómo se puede archivar el valor de salida y la duración en forma de tabla en la memoria de códigos (a partir de la dirección 200h)?



¿Cómo se puede introducir el valor de salida y la duración en forma de tabla en el RAM (a partir de la dirección 30h)?

¿Cómo se reconoce el final de la tabla?

Paso 2:

Elabore, a partir de la dirección 100h, un subprograma DELAY1 que suministre un segundo como base de tiempo. Analice el hardware empleado.

Ensaye el resultado con el osciloscopio.

Frecuencia de reloj:

Ciclo de máquina:

1,0 segundo corresponde a:

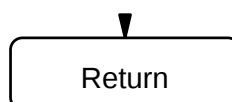
**Comando de salto seleccionado
para el bucle de contador:**

Núm. de CM por comando:

Núm. de bucles de contador:

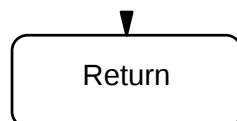
Anchura de registro:

Solución propuesta:



Paso 3:

Desarrolle de esto un subprograma DELAY al que entrega el parámetro < DelayTime > para la duración en segundos.



Paso 4:

Escriba el programa principal que ejecute de manera continua las 6 fases de semáforo que salen en el Port5.

Paso 5:

Amplíe el programa principal de tal forma que sea posible cambiar con el Bit0 en Port1 entre el modo de servicio continuo y el modo de servicio controlado según los requisitos (conecte "GND" o "+5V" a P1.0 por medio de un interruptor):

- Bit0 = 0:
Salida de fases continuas como en el paso 4
- Bit0 = 1:
"Verde " para la calle central, o sea la fase 6 se prolonga hasta que se presente una petición del iniciador de la calle secundaria o de una tecla de peatón.

Paso 6:

Amplíe el programa a partir de la dirección 80h con una rutina de servicio de interrupción (ISR) "MARK" para el mismo objetivo, o sea para cambiar entre modo de servicio continuo y modo de servicio controlado según los requisitos.

En cuanto se presente una petición, se accede a la rutina de ISR MARK:

MARK interrumpe el programa actual y pone inmediatamente (!!!) un bit de memoria. El bit de memoria decide, si permanece el „Verde permanente“ o si se ejecutan las demás fases. Indíque el bit de memoria por medio de un LED en Port 1.1. ¡Atención! Los LED de la unidad de LED son Active Low.

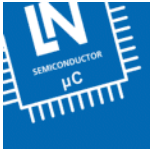
Para la petición externa de interrupción, úna la „Señal de inicio S1, S2“ con el puerto 1.4 y la „Señal de Inicio S3“ con el puerto 1.5.

Introduzca en la dirección 03h el comando de salto a ISR MARK.

Paso 7:

Análisis del resultado:

¿Es justificable, desde el aspecto técnico del tráfico, el estado de base 0 tras un Reset? ¿Cuál es la causa? Haga una propuesta para solucionar este problema.



SEMÁFORO
CMC 2-3



Lista de programa

```

;*****
;* Proyecto: "Control de un semáforo ", cmc23a.asm *
;* Fecha: 10.12.99 *
;* Tools: as, p2hex, debugger *
;* Encargado: Grosse *
;* *
;* Interfaces de hardware *
;* P1.0 conectado a UNIDAD DE INTERRUPTORES (Op. de interruptor) *
;* P1.1 conectado a UNIDAD de LED (Memoria de indicador) *
;* P1.4 conectado a semáforo S1 S2 (Tecla Paso de peatones) *
;* P1.5 conectado a semáforo S3 (Iniciador Calle secundaria) *
;* *
;* P5.0-7 conectado a semáforo Q0-7 *
;* por medio de unidad de driver de bus (sin inversión) *
;* *
;*****

cpu 80515 ; Seleccionar procesador -> C515C
include c515c.inc ; Insertar registro de funciones especiales

;***** Definir variables específicas del programa *****

segment data
ORG 2fh
DelayTime db ? ; Contador de tiempo para duración de fase
; de semáforo
Adres0 equ 30h ; Dirección de inicio de la tabla de fases
Marker bit 0 ; 20h.0 = Bit de memoria para interrupción
; activada
; Entradas en Port1:
Option bit P1.0 ; Control de ciclo de fases:
; "0" para continuo,
; "1" para controlado según petición
Indicador bit P1.1 ; Indica el estado del bit de memoria "Marker"
Iniciador bit P1.5 ; Bucle de inducción de la calle secundaria

;////////////////////////////////////

segment code
ORG 00h ; Vector de Reset
ajmp MAIN ; Salto al programa principal

ORG 4Bh ; Vector de interrupción EXTI2
ajmp MARK ; Salto a la rutina de servicio de
; interrupción

;***** Rutina de servicio de interrupción ( ISR ) *****
ORG 80h

;=====
; ISR: MARK>[Marker]
; Objetivo: En caso de petición de interrupción poner memoria
; Acceso: Mediante interrupción externa INT2 en P1.4
;=====

;***** Subprograma ( SP ) *****

```

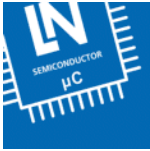
ORG 100h

```
;=====
;      SP:          DELAY(DelayTime)
;      Objetivo:    < DelayTime > * Retardar un segundo
;      Acceso:      Mediante MAIN
;=====
;      Información de fondo:
;      Microcontrolador C515C con reloj de 12 MHz.
;      ==> 1 µs por ciclo de máquina
;      ==> 1 segundo corresponde a 1.000.000 ciclos de máquina.
;-----
```

```
;=====
;      SP:          INIT
;      Objetivo:    Inicializar los puertos I/O P1 y P5
;                  y activar el control de interrupción
;      Acceso:      Mediante MAIN
;=====
```

```
;=====
;      SP:          TABEL
;      Objetivo:    Cargar la tabla para las 6 fases de semáforo
;                  en la memoria RAM
;      Acceso:      Mediante MAIN
;=====
```

```
;***** Programa principal ( MAIN ) *****  
;=====  
;      MAIN:          Programa principal "SEMÁFORO CMC2-3"  
;      Objetivo:      Controlar el semáforo  
;      Subprogramas:  INIT, TABEL, DELAY(DelayTime)  
;      Interrupción: INT2 en P1.4 da acceso a ISR MARK  
;      Entradas:      Port1.0,4,5  
;      Salidas:       Port5  
;=====
```



```
;***** Fin de programa *****  
END
```

CMC 2-4 Control de un motor paso a paso

Control de un motor paso a paso con el MC C515C, empleando tablas de valores y registros de 2 bytes.

Fundamentos preliminares

Un motor paso a paso acciona un dispositivo de posicionamiento. Para la activación por el microcontrolador C515C, el módulo de motor paso a paso pone a disposición tres entradas:

- "CLOCK"
cada vez que se reposicione una señal (transición "1-0") en la entrada "Clock1", el motor se mueve un 'paso'.
- "SENTIDO DE GIRO"
en esta entrada "R/ /L" se puede predeterminar el sentido de giro deseado, "1" para derecha, "0" para izquierda.
- "ANCHURA DE PASO"
esta entrada "H/ /F" define la anchura de paso a ejecutar, "1" para medio paso, "0" para paso completo.

La asociación para las salidas a controlar en el puerto Port1 será: Bit7 para anchura de paso, Bit6 para sentido de giro, Bit5 para Clock. Por ejemplo, con "111xxxxx" está determinado el modo de servicio "Medio paso / con giro a la derecha".
(x significa en este caso "0" o "1" libremente elegible)

Un movimiento regular repetido se puede describir de manera resumida con 3 Bytes:

- Modo de servicio: Medio paso-dcha / -izda,
Paso completo-dcha / -izda
- Número de pasos: 1...250 pasos
- Frecuencia 1...250 Hz

Vamos a archivar en una tabla una secuencia completa de movimientos, de varios pasos, con diferentes modo de servicio y frecuencia. El principio y el final de la tabla recibirán como indicativo FDh y Feh respectivamente.

Para una regulación del número de revoluciones con graduación muy precisa se empleará una base de tiempo de 100 microsegundos. De la frecuencia deseada (Fs) del motor paso a paso se calcula el tiempo de espera entre los distintos 'pasos' de la siguiente manera:
Constante DELAY Const := 10.000 / Fs

Nota: ¡¡¡CUIDADO CON LAS ALTAS TEMPERATURAS EN EL MOTOR!!!

Instalación del ensayo CMC 2-4

Aparatos y accesorios necesarios

Can	Denominación	Nº ID
1	Plataforma con mód. de alim. +5V	SO4000-1E
1	Fuente de alim. de enchufe CA 90...230V 45..65Hz, CC 9V 630mA	SO4000-1F
1	Módulo PSD1-Flash Controlador C515C	SO4000-1G
1	Unidad de adaptador de 8 Bit	SO4000-1K
1	Unidad LED	SO4000-1P
1	Unidad de interruptores	SO4000-1R
1	Adaptador BNC	SO4000-1T
1	Unidad de driver de bus	SO4000-1V
1	Motor paso a paso con disco codif.	SO4000-2C
1	Soporte para módulos	SO4000-2X
1	Software IDE para MCLS (D,GB,F,E)	SO4001-9H
1	Osciloscopio incl. puntas de prueba	LM6203
1	Sonda de prueba lógica	LM8101
1	Cable de medición BNC / BNC	LM9034
1	Cable de interfaz serial 9/9 polos	LM9040
6	Línea de medición 2mm 15cm azul	SO5126-5K
4	Línea de medición 2mm 15cm amarillo	SO5126-5M
1	Línea de med. adap. 4/2mm 50cm rojo	SO5126-6S
1	Línea de med. adap. 4/2mm 50cm azul	SO5126-6U
1	CMC 2 Aplicaciones con el microcontrolador 8051 C515C	SH5019-1B

Instalación de aparatos

¡Separe la alimentación de corriente del MCLS-modular !

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

Encaje la unidad de adaptador de 8 bit 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!

Atención: ¡ Usted debe unir los potenciales de masa del modelo y de la plataforma entre ellos !

Indicación de 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.

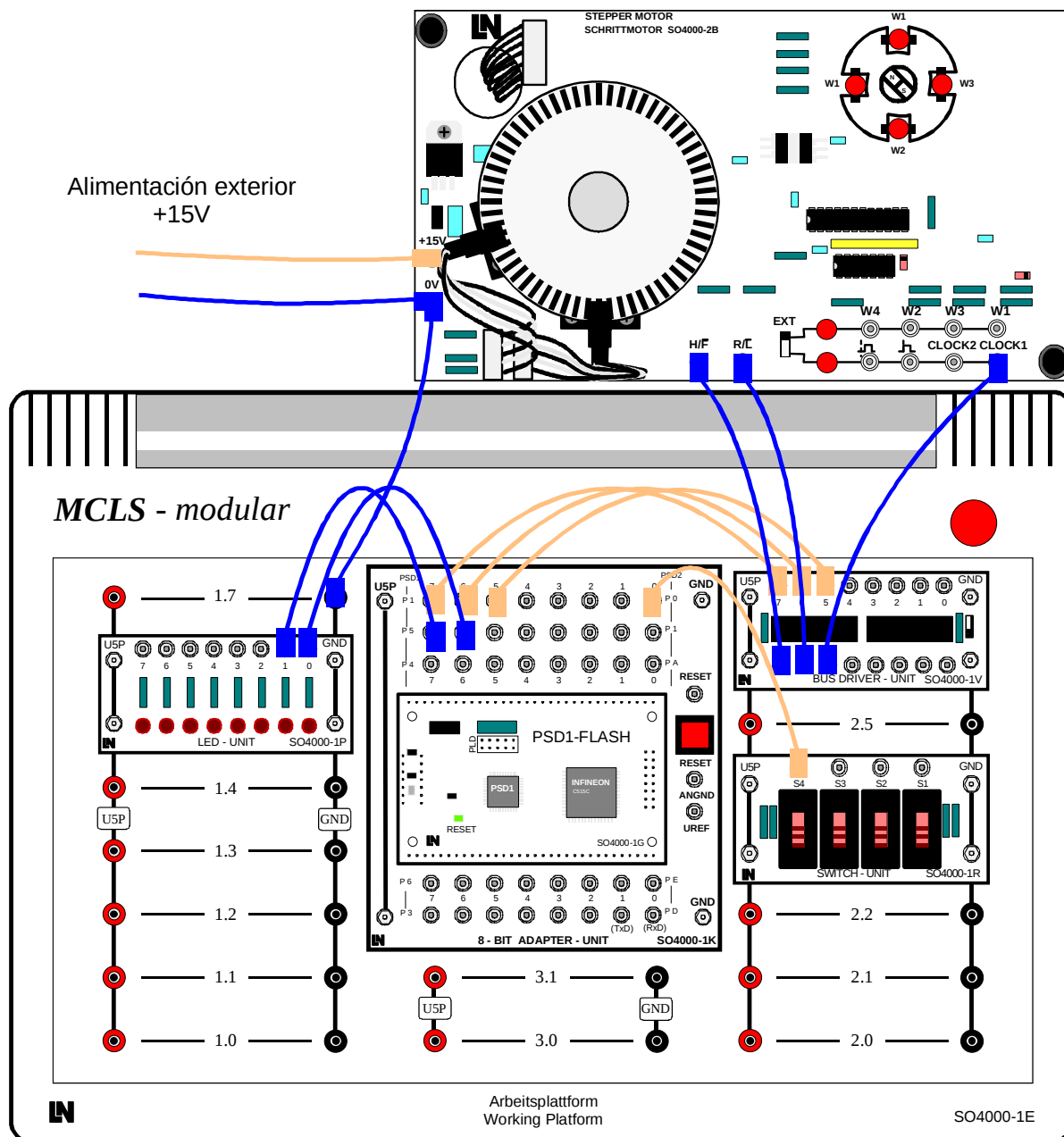


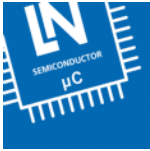
Fig.: 401 Instalación de aparatos del ensayo CMC 2-4

Ejercicio CMC 2-4:

Programe un control de proceso para el motor paso a paso empleando una tabla de valores y subprogramas.

Paso 1:

Realice un borrador de una tabla completa con 4 fases para un proceso de movimiento del motor paso a paso con diferente número de pasos, sentido de giro alterno y frecuencia constante.



```
mov 10h,#0FDh ;Indicativo Principio de la tabla
mov 11h,#0C0h ;Fase 1
                ;Modo de servicio = Medio paso dcha
mov 12h,#5d    ;Nº de pasos = 5
mov 13h,#100d  ;Frecuencia = 100Hz
mov 14h,#80h   ;Fase 2
                ;Modo de servicio = Medio paso izda
mov 15h,#10d   ;Nº de pasos = 10
mov 16h,#100d  ;Frecuencia = 100Hz
mov 17h,#40h   ;Fase 3
                ;Modo servicio = Paso completo dcha
mov 18h,#15d   ;Nº de pasos = 15
mov 19h,#100d  ;Frecuencia = 100Hz
mov 1Ah,#00h   ;Fase 4
                ;Modo servicio = Paso completo izda
mov 1Bh,#20d   ;Nº de pasos = 20
mov 1Ch,#100d  ;Frecuencia = 100Hz
mov 1Dh,#0Feh  ;Indicativo Final de tabla
```

Paso 2:

¿Qué relación existe

- (1) entre el límite superior de frecuencia y el indicativo de Principio de la tabla?
- (2) entre el modo de servicio y el indicativo de Final de la tabla?

FDh = 11111101b

253Hz = 11111101b

Podemos ver que el indicativo (FDh) para el principio de la tabla y la frecuencia (253Hz) presentan la misma secuencia de bits en el sistema binario. Por lo tanto tienen el mismo aspecto para el microcontrolador. FD es la representación Hex para el número decimal 253. En la consulta del principio de tabla, una frecuencia de 253Hz conduciría a la conclusión errónea que el final de la tabla ya se hubiera alcanzado y que no se debieran ejecutar más pasos.

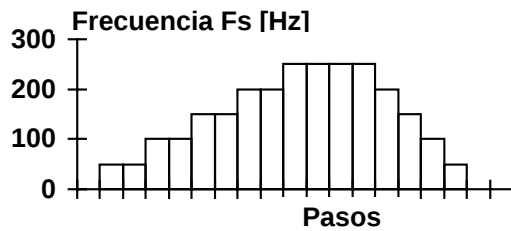
Entre el modo de servicio C0h = 11000000b y el indicativo de final de la tabla C0h = 11000000b existe la misma relación:

C0h es la representación Hex para el número decimal 192.

Conclusión: ¡La asociación de indicativos, frecuencias y modos de servicio siempre debe ser inequívoca!

Paso 3:

Programe, a partir de la dirección 40h, una tabla que acelere el motor paso a paso con número de revoluciones ascendente hasta una frecuencia máxima de 250 Hz y que a continuación reduzca la frecuencia hasta la parada, eso es, la llamada „rampa de arranque y de frenado“.



Nota:

Evitamos la división complicada $\{10.000 / F_s\}$. Calcule la respectiva constante de DELAY y archive los correspondientes valores en la tabla. Siempre tiene que reservar dos bytes (una palabra) para ello, porque los valores para Const pueden situarse entre 40 y 10.000.

Paso 4:

Prepare el subprograma corto "DELAY" que retarda el proceso con una base de tiempo de 100 µs. Compruebe su resultado con el osciloscopio.

Frecuencia de reloj:

Ciclo de máquina:

100µs corresponden a:

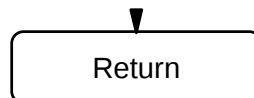
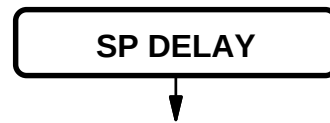
Comando de salto seleccionado

para el bucle de contador:

Nº de CM por comando:

Nº de bucles de contador:

N = Número de CM de los demás comandos del subprograma



Si se guardan el acumulador y la palabra de estado de programa, la constante de Delay debe ser 47.

$$\text{Tiempo} = 1+1+1+(47*2)+1+1+2 = 101\mu\text{s}$$

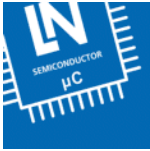
Paso 5:

Elabore el subprograma "STEP" que incorpore como parámetros los valores para el modo de servicio, el número de pasos, la frecuencia y la constante de DELAY para controlar la ejecución de los pasos conformemente al tiempo.

SP STEP



Return



Paso 6:

Elabore el programa principal que seleccione los valores de la tabla y los entregue al subprograma "STEP".

Dependiendo del Bit0 en Port1, el programa deberá detenerse opcionalmente al final de la tabla o reiniciarse desde el principio. (Para ello, úna opcionalmente "0V" o "5V" con P1.0 por medio de un interruptor).

Visualice el modo del sentido de giro y de la anchura de paso por medio de LEDs adicionalmente conectados a los puertos 5.6 y 5.7. (Si emplea la unidad de LEDs, ¡observe que éstos son Active Low!)

Paso 7:

El programa principal ampliado deberá editar los mismos valores una vez más después de haber ejecutado la tabla - esta vez en orden inverso hasta alcanzar el principio de la tabla - para luego volver a empezar desde el principio.

Paso 8:

Modifique (con 2 líneas) el programa principal para que el motor paso a paso ejecute estos pasos durante la „lectura hacia atrás“ con el sentido de giro contrario.

Paso 9: Ejercicio adicional

La división "10.000 / Fs" es algo más difícil porque la función "DIV AB" solamente procesa valores de 8 bits.

Programa una "división de los grandes números" realizando la división „al revés“:

Sume repetidamente Fs; cuénte el número de las adiciones y la suma intermedia hasta que alcance exactamente o supere por primera vez 10.000.

Notas:

Utilice el registro de 2 bytes „DPTR“ como contador.

Elimine de la tabla los valores que usted ha calculado para la constante de DELAY Const.

Lista de programa hasta el paso 8

```

;*****
;* Proyecto: "Control de un motor paso a paso",cmc24a.asm      *
;* Fecha:      10.12.99                                         *
;* Tools:      as, p2hex, debugger                             *
;* Encargado: Grosse                                           *
;*
;* Interfaces de hardware                                       *
;* P1.0  conectado a SWITCH-UNIT          (Opción de interruptor) *
;* P1.5  Clock1 Motor paso a paso         *
;*        por medio de unidad de driver de bus (sin inversión) *
;* P1.6  R/L      Motor paso a paso         *
;*        por medio de unidad de driver de bus (sin inversión) *
;* P1.7  H/F      Motor paso a paso         *
;*        por medio de unidad de driver de bus (sin inversión) *
;*
;* P5.6 conectado a unidad LED (indicación R/L)                *
;* P5.7 conectado a unidad LED (indicación H/F)                *
;*
;*****
;////////////////////////////////////
;
;          Solución propuesta Ensayo CMC2-4 por medio de tabla
;////////////////////////////////////

        cpu      80515          ; Seleccionar procesador -> C515C
        include  c515c.inc      ; Insertar registro de funciones
                                ; especiales

;***** Definir variables específicas del programa *****

        segment data
Adres0      equ      10h        ; Dirección de inicio de la tabla

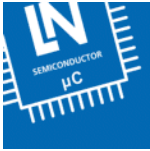
        ORG 3Bh
Mode        db        ?        ; Modo de servicio (C0h,80h,40h,00h)
Number      db        ?        ; Número de pasos (1-250)
FrequS      db        ?        ; Frecuencia (1-250 Hz)
ConstH      db        ?        ; Constante de DELAY (High)
ConstL      db        ?        ; Constante de DELAY (Low)

StopGo      bit       P1.0      ; Entradas y salidas en Port1 :
Clock       bit       P1.5      ; Bit de entrada para empezar/terminar
Right       bit       P1.6      ; Impulso de aceptación de Clock en P1.5
HaStep      bit       P1.7      ; Marcha a la dcha/izda en P1.6
                                ; Medio paso/Paso completo en P1.7

;////////////////////////////////////
        segment code
        ORG 00h
        ajmp     MAIN          ; Salto al programa principal

;***** Subprograma ( SP ) *****
        ORG 40h
;=====
;          SP:          TABEL
;          Objetivo:    Cargar la tabla de parámetros en RAM
;          Acceso:      Mediante MAIN
;=====

```



```
;      Nota: Calcular previamente la constante de DELAY
;      e introducirla en la tabla
;-----
```

```
;=====
;      SP:          DELAY
;      Objetivo:    Retardar 100 microsegundos
;      Acceso:      Mediante SP STEP
;=====
;      Información de fondo:
;      Microcontrolador C515C con reloj de 12 MHz
;      ==> 1 μs por ciclo de máquina
;      ==> 100 μs corresponden a 100 ciclos de máquina
;-----
```

```

;=====
;      SP:          STEP (Mode Number FreqUS Const)
;      Objetivo:     Controlar la ejecución de los pasos
;      Acceso:       Mediante MAIN
;      Subprograma:   DELAY
;      Salidas:      Port1.5-7 (=Clock, RL, HV)
;=====
;      Nota: Calcular previamente la constante de DELAY
;      e introducirla en la tabla
;-----

```

```

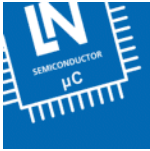
;----- Esperar tiempo de retardo -----

```

```

;----- Ejecutar paso -----

```

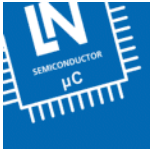


```
;***** Programa principal ( MAIN ) *****  
  
;=====  
;      MAIN:          Programa principal "CMC2-4"  
;      Objetivo:      Controlar los movimientos del motor paso  
;                    a paso  
;      Subprogramas:  TABEL, STEP (Mode Number FreqS Const)  
;      Entradas:      Port1.0  
;      Salidas:       Port1.5-7  
;=====  
;      Nota: Calcular previamente la constante de DELAY  
;      e introducirla en la tabla  
;-----
```

;----- HACIA DELANTE -----

;----- HACIA ATRÁS -----

;***** Fin de segmento de programa *****
END



Lista de programa según ejercicio adicional Paso 9

```
;*****
;* Proyecto: "Control de un motor paso a paso ",cmc24b.asm*
;* Fecha: 10.12.99*
;* Tools: as, p2hex, debugger*
;* Encargado: Grosse*
;*
;* Interfaces de hardware*
;* P1.0 conectado a unidad de interruptores (Opción interrupt.)*
;* P1.5 Clock1 Motor paso a paso*
;* mediante unidad de driver de bus (sin inversión)*
;* P1.6 R/L Motor paso a paso*
;* mediante unidad de driver de bus (sin inversión)*
;* P1.7 H/F Motor paso a paso*
;* mediante unidad de driver de bus (sin inversión)*
;*
;* P5.6 conectado a unidad LED (indicación R/L)*
;* P5.7 conectado a unidad LED (indiciación H/F)*
;*
;*****
;////////////////////////////////////
;
; Solución propuesta Ensayo CMC2-4 con división
;////////////////////////////////////

cpu 80515 ; Seleccionar procesador C515C
include c515c.inc ; Insertar registro de funciones
; especiales

;***** Definir variables específicas del programa *****
segment data
Adres0 equ 10h ; Tabla de dirección de inicio

ORG 37h
Mode db ? ; Modo de servicio (C0h,80h,40h,00h)
Number db ? ; Número de pasos (1-250)
FrequeS db ? ; Frecuencia (1-250 Hz)
ConstH db ? ; Constante de DELAY (High)
ConstL db ? ; Constante de DELAY (Low)
SumH db ? ; Suma intermedia (High) )
SumL db ? ; Suma intermedia (Low) ) para SP (
TenH db ? ; Diez mil (High) ) DIVplus (
TenL db ? ; Diez mil (Low) )
; Entradas y salidas en Port1 :
StopGo bit P1.0 ; Bit de entrada para empezar/terminar
Clock bit P1.5 ; Impulso de aceptación de Clock en P1.1
Right bit P1.6 ; Marcha a la dcha/izda en P1.2
HaStep bit P1.7 ; Medio paso/Paso completo en P1.3
;////////////////////////////////////

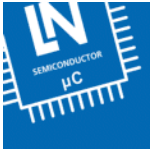
segment code
ORG 00h
ajmp MAIN ; Salto al programa principal

;***** Subprogramas ( SP ) *****
ORG 40h

;=====
; SP: TABEL
; Objetivo: Introducir la tabla de parámetros en RAM
; Acceso: Mediante MAIN
```

;=====

```
;=====
;      SP:          DELAY
;      Objetivo:    Retardar 100 microsegundos
;      Acceso:      Mediante SP STEP
;=====
;      Información de fondo:
;      Microcontrolador C515C con reloj de 12 MHz
;      ==> 1 µs por ciclo de máquina
;      ==> 100 µs corresponden a 100 ciclos de máquina
;-----
```



```
;=====
;      SP:          STEP (Mode Number Frequs)
;      Objetivo:    Controlar la ejecución de los pasos
;      Acceso:      Mediante MAIN
;      Subprogramas: DELAY
;      Salidas:     Port1.5-7 (=Clock, RL, HV)
;=====
```

```
;----- Esperar tiempo de retardo -----
```

```
;----- Ejecutar paso -----
```

```
;=====
;      SP:          DIVplus(FrequeS)>[DPTR]
;      Objetivo:    Dividir números grandes {10.000/fs},
;                  Devolver constante de DELAY a DPTR
;=====
;      "División" como procedimiento inverso:
;      Sumar repetidamente la frecuencia fs;
;      Contar el número de adiciones y la suma intermedia,
;      hasta que se alcance exactamente o se supere por primera
;      vez 10.000.
;-----
```

```

;***** Programa principal ( MAIN ) *****
;=====
;      MAIN:          Programa principal "CMC2-4b"
;      Objetivo:      Controlar el movimiento del motor paso a paso
;      Subprograma:    TABEL, STEP (Mode Number FreqUS)
;      Entradas:       Port1.0
;      Salidas:        Port1.1-3
;=====

;----- HACIA DELANTE -----

;----- HACIA ATRÁS -----

;***** Fin de segmento de programa *****
END

```

CMC 2-5 Control de una instalación ventiladora de túnel

Control de una instalación ventiladora de túnel con el MC C515C empleando tablas de valores y la técnica de subprogramas.

Fundamentos preliminares

En un túnel de carretera, 4 sensores de sustancias nocivas I1...I4 registran la contaminación del aire. Según la necesidad, el control conectará y desconectará 3 turbinas de ventilación. Una instalación de semáforo regula el flujo de la circulación a través del túnel cuando están conectadas las 4 turbinas de ventilación.

Especialmente en caso de un atasco en el túnel, la instalación de semáforo impide la entrada. Un atasco es registrado por un bucle de iniciador I5 que vigila el flujo de la circulación.

Las fases de semáforo están definidas de la siguiente manera:

	Fase N°:	1	2	3	4
	Duración [sec]:	1	10	1	10
Port1 Bit:	Conexión:				
7	L3 Rojo		X	X	
6	L2 Amarillo	X		X	
5	L1 Verde				X

Instalación del ensayo CMC 2-5

Aparatos y accesorios necesarios

Can	Denominación	Nº ID
1	Plataforma con mód. de alim. +5V	SO4000-1E
1	Fuente de alim. de enchufe CA 90...230V 45..65Hz, CC 9V 630mA	SO4000-1F
1	Módulo PSD1-Flash Controlador C515C	SO4000-1G
1	Unidad de adaptador de 8 bits	SO4000-1K
1	Adaptador BNC	SO4000-1T
1	Unidad de driver de bus	SO4000-1V
1	Ventilación de túnel	SO4000-2P
1	Soporte para módulos	SO4000-2X
1	Software IDE para MCLS (D,GB,F,E)	SO4001-9H
1	Osciloscopio incl. puntas de prueba	LM6203
1	Sonda de prueba lógica	LM8101
1	Cable de medición BNC / BNC	LM9034
1	Cable de interfaz serial 9/9 polos	LM9040
6	Línea de medición 2mm 15cm amarillo	SO5126-5M
7	Línea de medición 2mm 30cm azul	SO5126-5S
6	Línea de medición 2mm 30cm amarillo	SO5126-5U
1	CMC 2 Aplicaciones con el microcontrolador 8051 C515C	SH5019-1B

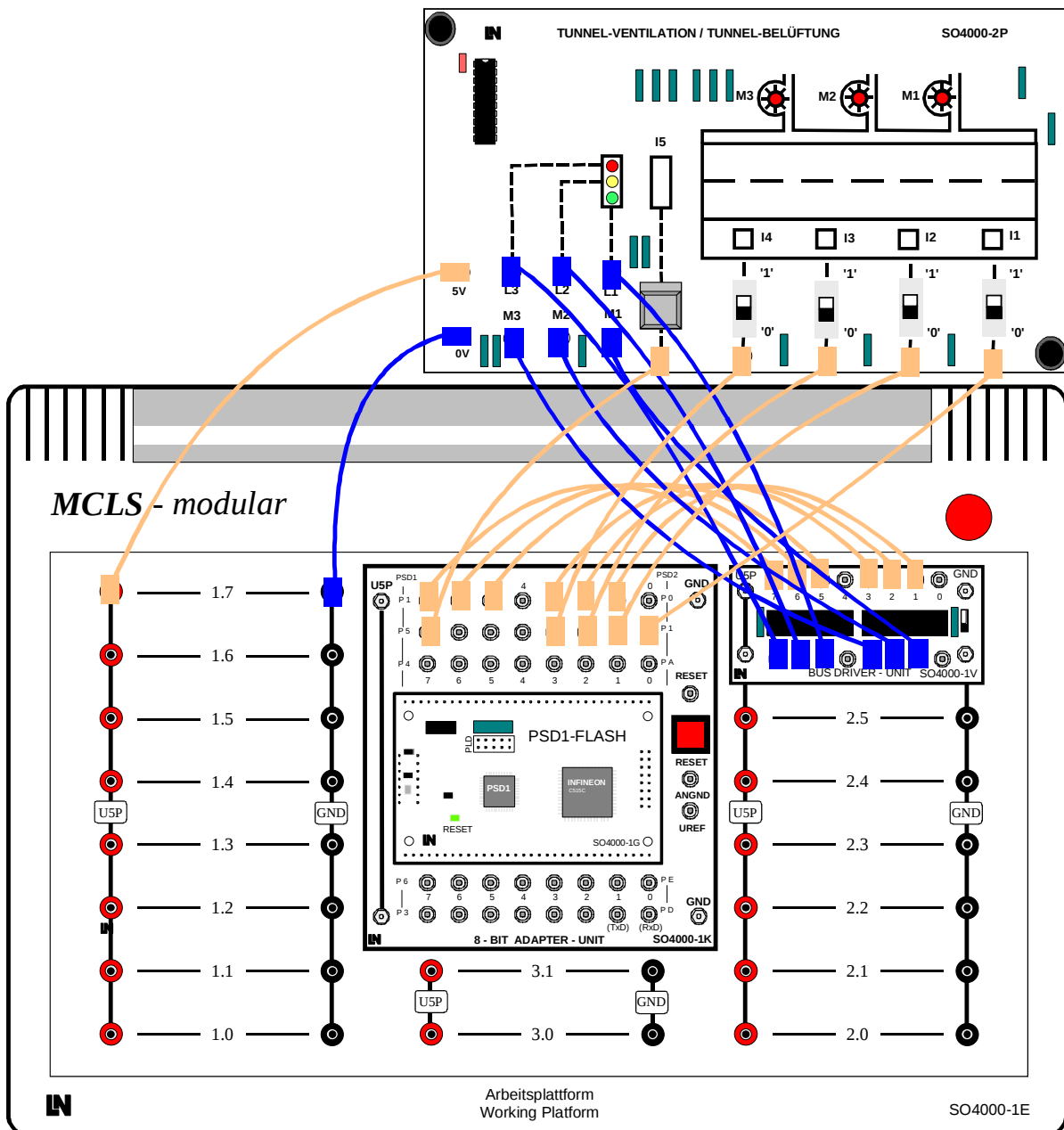


Fig.: 501 Instalación de aparatos del ensayo CMC 2-5

Instalación de aparatos

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

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

Encaje la unidad de adaptador de 8 bit 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 de 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.

Ejercicio CMC 2-5:

Programa un control de proceso para la instalación ventiladora de túnel empleando tablas de valores y subprogramas

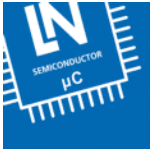
Paso 1:

Realice un análisis de problema para el control de semáforo y conteste a las siguientes preguntas.

¿Cuáles son los valores de salida de las distintas fases de semáforo como secuencia de bits, en representación Hex y como número decimal? ¡Rellene la siguiente tabla de valores!

Fase	Q7	Q6	Q5	Q4	Q3	Q2	Q1	Q0	hex	dez
1										
2										
3										
4										
Fin										

¿Cómo se puede cargar el valor de salida y la duración en forma de tabla en el RAM (a partir de la dirección 30h)?



¿Cómo se puede reconocer el final de la tabla?

Paso 2:

Para preparar el control de semáforo elabore, a partir de la dirección 40h, un subprograma DELAY que retarde con una base de tiempo de 1 segundo y espere varios segundos, según el parámetro entregado < Time >.

Ensaye el resultado con el osciloscopio.

Frecuencia de reloj:

Ciclo de máquina:

1,0 segundo corresponde a:

Comando de salto seleccionado

para el bucle de contador:

Núm. de CM por comando:

Núm. de bucles de contador:

Anchura de registro:

Solución propuesta:

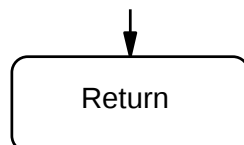
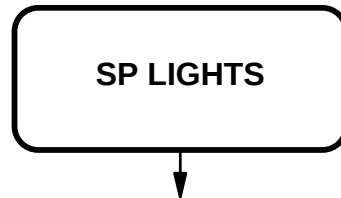
SP DELAY



Return

Paso 3:

Desarrolle un subprograma que recorra una vez las 4 fases de semáforo.



Paso 4:

Elabore el programa principal que conecte los ventiladores de la siguiente manera:

- El ventilador M1 se pone en marcha si se activa 1 sensor,
- los ventiladores M1 y M2 funcionan con 2 sensores activados,
- con más de 2 sensores activados funcionan todos los ventiladores.

Amplíe el programa principal de tal forma que el semáforo

- recorra las fases de semáforo cuando haya 4 sensores activados,
- permanezca en el estado „Verde“ cuando haya menos de 4 sensores activados.

Paso 5:

En caso de atasco/accidente se activa el bucle de iniciador I5. Todos los ventiladores deberán arrancar inmediatamente y el semáforo deberá pasar a „Rojo“ para 10 segundos. En caso de un atasco/accidente de duración más larga, el semáforo permanecerá en el estado „Rojo“. Inserte esta consulta al principio del bucle del programa principal.

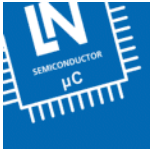


Diagrama de flujo de programa para el control de la instalación ventiladora de túnel:

INSTAL. DE TÚNEL
CMC 2-5



Lista de programa

```

;*****
;* Proyecto: "Control de una instalación ventiladora de túnel" *
;*          cmc25.asm *
;* Fecha:    10.12.99 *
;* Tools:    as, p2hex, debugger *
;* Encargado: Grosse *
;*          *
;* Interfaces de hardware *
;* P1.1 Ventilador M1 mediante unid. de driver de bus (sin inversión) *
;* P1.2 Ventilador M2 mediante unid. de driver de bus (sin inversión) *
;* P1.3 Ventilador M3 mediante unid. de driver de bus (sin inversión) *
;*          *
;* P1.5 Semáforo "Verde" *
;*      mediante unidad de driver de bus (sin inversión) *
;* P1.6 Semáforo "Amarillo" *
;*      mediante unid. de driver de bus (sin inversión) *
;* P1.7 Semáforo "Rojo" *
;*      mediante unid. de driver de bus (sin inversión) *
;*          *
;* P5.0 conectado al sensor de gases de escape I1 *
;* P5.1 conectado al sensor de gases de escape I2 *
;* P5.2 conectado al sensor de gases de escape I3 *
;* P5.3 conectado al sensor de gases de escape I4 *
;* P5.7 conectado al sensor de atasco I5 *
;*          *
;*****

    cpu      80515          ; Seleccionar procesador -> C515C
    include  c515c.inc      ; Insertar registro de funciones especiales

;***** Definir variables específicas del programa *****

    segment data
    ORG 2Ch
Adres0      equ      30h    ;Dirección de inicio de la tabla

SumSen      db      ?      ; Número de sensores activados
Count       db      ?      ; Contador para prueba de sensores
DelayTime   db      ?      ; Contador de tiempo para duración de
                                ; fases de semáforo
Vnew        db      ?      ; Memoria intermedia Valor teórico
                                ; Ventilador
                                ; Entradas y Salidas:
Green       bit      P1.5   ; Semáforo "Verde"
Red         bit      P1.7   ; Semáforo "Rojo"
Crash       bit      P5.7   ; (Pulsador) Accidente/Atasco en el túnel

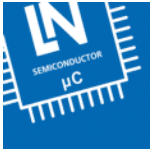
;////////////////////////////////////

    segment code
    ORG 00h
    ajmp     MAIN          ; Salto al programa principal

;***** Subprogramas ( SP ) *****
    ORG 40h

;=====
;      SP:          DELAY(DelayTime)
;      Objetivo:    < DelayTime > * retardar un segundo
;      Acceso:      Mediante MAIN
;=====

```

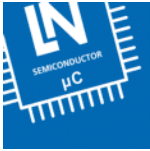


```
;      Información de fondo :  
;      Microcontrolador C515C con reloj de 12 MHz.  
;      ==> 1 µs por ciclo de máquina  
;      ==> 1 segundo corresponde a 1.000.000 ciclos de máquina.  
;-----
```

```
;=====  
;      SP:          INIT  
;      Objetivo:    Iniciar los puertos I/O P5 P1 y cargar  
;                  la tabla de 4 fases de semáforo en el RAM  
;      Acceso:      Mediante MAIN  
;=====
```

```
;=====
;      SP:          SENSOR>[SumSen]
;      Objetivo:     Contar los sensores activados
;      Acceso:       Mediante MAIN
;      Entradas:     Port5.0-3 (=sensores)
;=====
```

```
;=====
;      SP:          VENTI(SumSen)
;      Objetivo:     Conectar o desconectar los ventiladores
;                   según el número de sensores activados
;      Acceso:       Mediante MAIN
;      Salidas:      Port1.1-3 (=Ventiladores)
;=====
```



```
;=====
;      SP:           LIGHTS
;      Objetivo:     Recorrer las 4 fases de semáforo
;      Acceso:       Mediante MAIN
;      Subprograma:  DELAY(DelayTime)
;      Salidas:      Port1.5-7 (=Semáforos)
;=====
```

```
;=====
;      SP:           STOP
;      Objetivo:     En caso de atasco/accidente en el túnel,
;                   semáforo "Rojo"
;                   y conectar todos los ventiladores
;      Acceso:       Mediante MAIN
;      Subprogramas: DELAY(DelayTime)
;      Salidas:      P1.1-3,5-7 (=semáforos y ventiladores)
;=====
```

```

;=====
;      SP:          FREE
;      Objetivo:     Después de un atasco conectar semáforo "Verde"
;      Acceso:       Mediante MAIN
;      Subprograma:  DELAY(DelayTime)
;      Salidas:      P1.5-7 (=semáforos)
;=====

```

```

;***** Programa principal ( MAIN ) *****

```

```

;=====
;      MAIN:        Programa principal "TUNEL CMC2-5"
;      Objetivo:     Control de la instalación ventiladora de túnel
;      Subprogramas: DELAY(DelayTime), INIT, FREE, LIGHTS,
;                   STOP, SENSOR>[SumSen], VENTI(SumSen)
;      Entradas:     Port5.0-3,7 (=sensores, iniciador de atasco)
;      Salidas:      Port1.1-3,5-7 (=semáforos, ventiladores)
;=====

```

```

;***** Fin de programa *****
END

```

CMC 2-6 Control de un mecanismo agitador

Control de un mecanismo agitador con el MC C515C empleando la técnica de subprogramas y bits de memoria.

Fundamentos preliminares

El modelo de una instalación mezcladora para líquidos simula el relleno de un depósito mezclador con cuatro diferentes líquidos por medio de las válvulas de admisión y la bomba. Ocho LEDs indican el nivel de relleno creciente, y el nivel se puede consultar por medio de cuatro sensores de nivel de relleno. El mecanismo agitador mezcla los líquidos en los tiempos predeterminados por accionamiento por motor. Por medio de la válvula de purga el depósito es vaciado.

Dos pulsadores permiten enviar las señales de inicio y de parada a la instalación.

Mediante un programa de control se realizará el siguiente ciclo con 9 fases:

1. Vaciar depósito
2. Esperar la señal de inicio
3. Rellenar líquido 1 hasta alcanzar nivel de rell. 2
4. Rellenar líquido 2 hasta alcanzar nivel de rell. 3
5. Mezclar líquidos durante 2 segundos
6. Rellenar líquido 3 hasta alcanzar nivel de rell. 4
7. Mezclar líquidos durante 4 segundos
8. Vaciar depósito, consultando si hay señal de parada
9. Continuar con fase 3 si no se envió la señal de parada durante la fase 8, en otro caso pasar a fase 2.

Instalación del ensayo CMC 2-6

Aparatos y accesorios necesarios

Can	Denominación	Nº ID
1	Plataforma con mód. de alim. +5V	SO4000-1E
1	Fuente de alim. de enchufe CA 90...230V 45..65Hz, CC 9V 630mA	SO4000-1F
1	Mód. PSD1-Flash Controlador C515C	SO4000-1G
1	Unidad de adaptador de 8 bits	SO4000-1K
1	Adaptador BNC	SO4000-1T
1	Unidad de driver de bus	SO4000-1V
1	Mecanismo agitador	SO4000-2F
1	Soporte para módulos	SO4000-2X
1	Software IDE para MCLS (D,GB,F,E)	SO4001-9H
1	Osciloscopio incl. puntas de prueba	LM6203
1	Sonda de prueba lógica	LM8101
1	Cable de medición BNC / BNC	LM9034
1	Cable de interfaz serial de 9/9 polos	LM9040
7	Línea de medición 2mm 15cm amar.	SO5126-5M
8	Línea de medición 2mm 30cm azul	SO5126-5S
7	Línea de medición 2mm 30cm amar.	SO5126-5U
1	CMC 2 Aplicaciones con el microcontrolador 8051 C515C	SH5019-1B

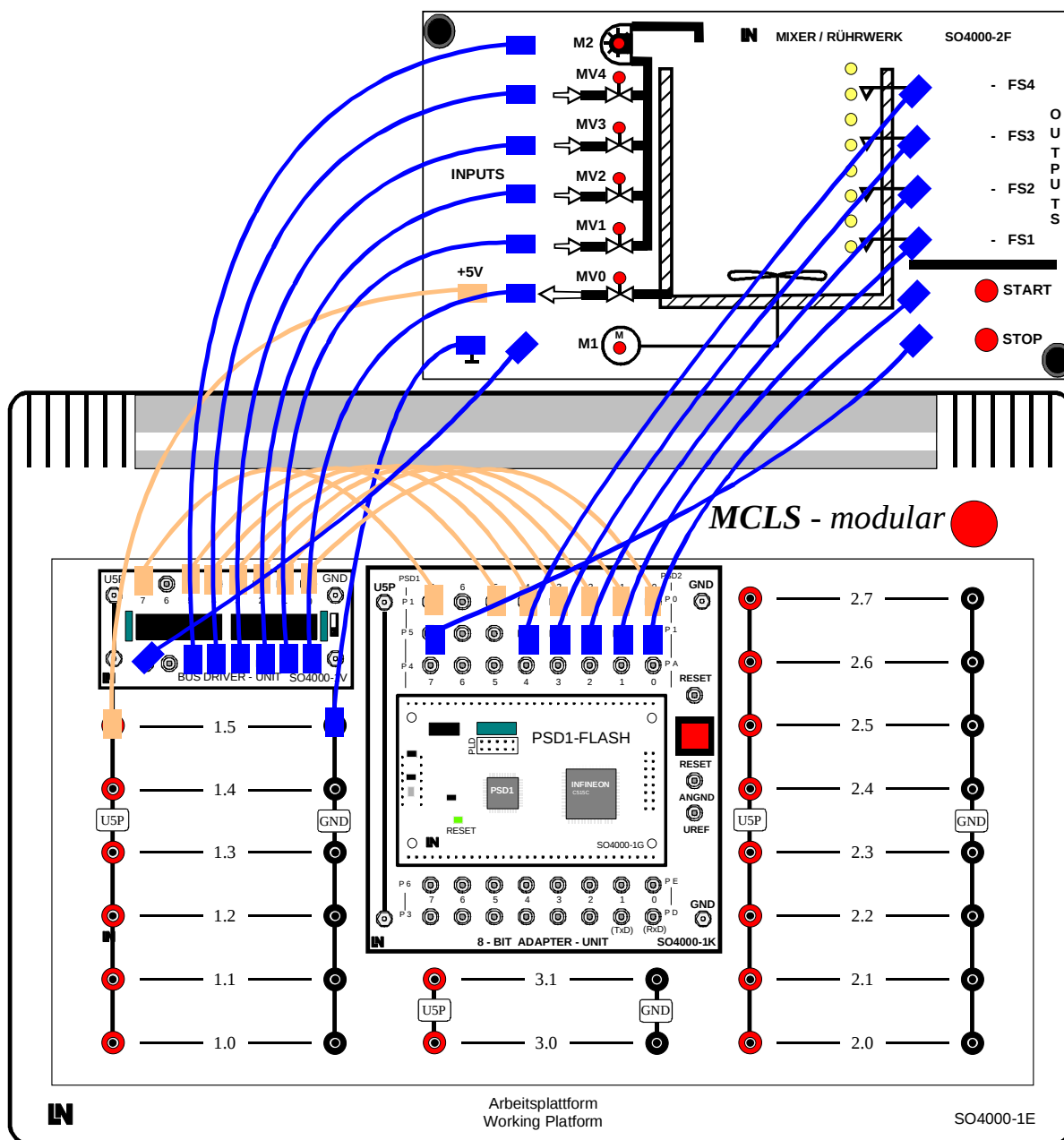


Fig.: 601 Instalación de aparatos del ensayo CMC 2-6

Instalación de aparatos

¡Separe la alimentación de corriente del MCLS-modular !

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

Encaje la unidad de adaptador de 8 bit 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 de 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.

Ejercicio CMC 2-6:

Programa un control de proceso para el control de un mecanismo agitador empleando subprogramas y bits de memoria

Paso 1:

Elabore, a partir de la dirección 40h, el subprograma "DELAY" que, correspondientemente al parámetro de entrega < Time >, retarde el proceso 1 segundo.

Frecuencia de reloj:

Ciclo de máquina:

1,0 segundo corresponde a:

Comando de salto seleccionado

para el bucle de contador:

Núm. de CM por comando:

Núm. de bucles de contador:

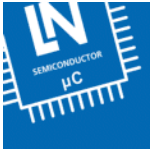
Anchura de registro:

Solución propuesta:

SP DELAY



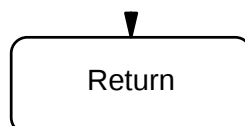
Return



Paso 2:

Elabore un subprograma "FILL". FILL rellenará un depósito con un determinado líquido hasta un nivel de relleno predeterminado.

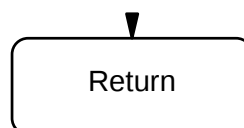
< Fluid > y < Level > son los correspondientes parámetros de entrega.



Paso 3:

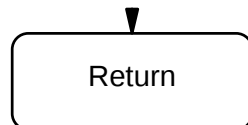
Desarrolle un subprograma "EMPTY". EMPTY vaciará el depósito hasta que todos los iniciadores estén libres.

Amplíe el subprograma "EMPTY" de tal forma que al mismo tiempo efectúe una consulta de la tecla de parada y en caso de accionamiento ponga un bit de memoria.



Paso 4:

Elabore un subprograma "MIX". MIX conectará el motor del mecanismo agitador durante un tiempo <Time> determinado.



Paso 5:

Programe ahora el programa principal que controle el mecanismo agitador por medio de los subprogramas preparados en todas las fases y funciones.

AGITADOR
CMC 2-6

Lista de programa

```

;*****
;* Proyecto: "Control de un mecanismo agitador "      cmc26.asm      *
;* Fecha:      10.12.99                                           *
;* Tools:      as, p2hex, debugger                               *
;* Encargado: Grosse                                              *
;*                                                     *
;* Interfaces de hardware                                         *
;* P1.0 Válvula de purga MV0 med. unid. driver de bus (sin inversión)*
;* P1.1 Válvula MV1 mediante unid. driver de bus (sin inversión)  *
;* P1.2 Válvula MV2 mediante unid. driver de bus (sin inversión)  *
;* P1.3 Válvula MV3 mediante unid. driver de bus (sin inversión)  *
;* P1.4 Válvula MV4 mediante unid. driver de bus (sin inversión)  *
;* P1.5 Bomba M2 mediante unid. driver de bus (sin inversión)     *
;* P1.7 Mec. agitador M1 med. unid. driver de bus (sin inversión) *
;*                                                     *
;* P5.0 conectado a la tecla de inicio                           *
;* P5.1 conectado al sensor de nivel de relleno FS1               *
;* P5.2 conectado al sensor de nivel de relleno FS2               *
;* P5.3 conectado al sensor de nivel de relleno FS3               *
;* P5.4 conectado al sensor de nivel de relleno FS4               *
;* P5.7 conectado a la tecla de parada                           *
;*                                                     *
;*****

cpu      80515                ; Seleccionar procesador -> C515C
include  c515c.inc            ; Insertar registro de funciones especiales

;***** Definir variables específicas de programa *****

segment data
ORG 2fh
DelayTime db      ?          ; Contador de tiempo
Fluid     db      ?          ; Líquido
Level     db      ?          ; Nivel de relleno

Stop      bit      1          ; 20h.1 Memoria tecla de parada
; Entradas en Port3:
On         bit      P5.0      ; Tecla de inicio en P5.0
Off        bit      P5.7      ; Tecla de parada en P5.7
; Salidas en Port1:
VaOut     bit      P1.0      ; Válvula de purga en P1.0
Pumpe     bit      P1.5      ; Bomba en P1.5
Motor     bit      P1.7      ; Motor en P1.7

;////////////////////////////////////

segment code
ORG 0h
ajmp      MAIN                ; Salto al programa principal

;***** Subprogramas ( SP ) *****
ORG 40h

```

```

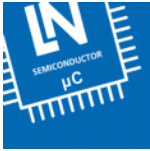
;=====
;      SP:          DELAY(DelayTime)
;      Objetivo:    < DelayTime > * Retardar 1 segundo
;      Acceso:      Mediante MAIN o MIX
;=====
;      Información de fondo:
;      Microcontrolador C515C con reloj de 12 MHz
;      ==> 1 µs por ciclo de máquina
;      ==> 1 segundo corresponde a 1.000.000 ciclos de máquina
;-----

```

```

;=====
;      SP:          EMPTY>[Stop]
;      Objetivo:    Controlar el vaciado del depósito;
;                  Poner memoria cuando la tecla de parada
;                  está pulsada
;      Acceso:      Mediante MAIN
;      Subprograma: DELAY(DelayTime)
;=====

```



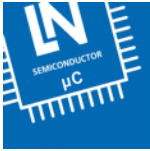
```
;=====
;      SP:          FILL(Fluid, Level)
;      Objeto:      Rellenar el líquido hasta el nivel
;                  de relleno predeterminado
;      Acceso:      Mediante MAIN
;      Subprograma: DELAY(DelayTime)
;=====
```

```
;=====
;      SP:          MIX(DelayTime)
;      Objetivo:    Controlar el motor del mecanismo agitador
;      Acceso:      Mediante MAIN
;      Subprograma: DELAY(DelayTime)
;=====
```

```
;***** Programa principal ( MAIN ) *****
```

```
;=====
;      MAIN:          Programa principal "AGITADOR CMC2-6"
;      Objetivo:      Controlar el proceso en el agitador
;      Subprogramas:  DELAY(DelayTime), EMPTY>[Stop],
;                   MIX(DelayTime), FILL(Fluid, Level)
;      Entradas:      Port5.0,1-4,7
;      Salidas:       Port1.0,1-4,5,7
;=====
```

```
;***** Fin de programa *****
      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