

Un nuevo enfoque en la enseñanza de microcontroladores basado en un diseño propio de bajo costo de un sistema de desarrollo del micro 80C32.

R. L. Millán y M. Perera
Grupo de Tecnología Electrónica.
Dpto. de Ingeniería de Sistemas y Automática
Escuela Superior de Ingenieros. Universidad de Sevilla.
Tlf: 95-4556857, Fax: 95-4556849, E-mail: millan@gte.esi.us.es

RESUMEN.- En este trabajo se destaca la creciente importancia de los diseños industriales basados en microcontroladores, y la conveniencia de mejorar la formación en dicha materia a los alumnos universitarios de estudios técnicos superiores. Para conseguir este objetivo se propone un sistema de desarrollo de bajo costo basado en el conocido microcontrolador 80C32 de Intel, con finalidad didáctica para que sea montado y programado por los propios alumnos. El sistema se ha utilizado para el desarrollo por parte de los alumnos de aplicaciones prácticas, propuestas, diseñadas e implementadas por ellos mismos, revisándose para ello los criterios evaluadores de la asignatura.

1.- INTRODUCCIÓN

El diseño electrónico basado en sistemas con microcontroladores está ampliamente extendido en la industria. Prueba de su enorme vigencia son las elevadas inversiones que realizan los fabricantes de microcontroladores en mejorar sus productos y en lanzar nuevos al mercado. Como ejemplo se puede citar a la empresa Hitachi, que diseña del orden de 300 nuevos microcontroladores al año, lo que equivale casi a un nuevo micro cada día.

Los sistemas microcontroladores presentan grandes ventajas respecto a otro tipo de diseño electrónico como pueden ser los circuitos integrados a medida (ASIC) o las placas de circuito impreso con componentes discretos. El bajo costo del diseño y su gran versatilidad lo hace apropiado para casi todo tipo de aplicaciones. Solamente en aquellos casos en los que se requiere una elevada velocidad o procesamiento en paralelo resulta desaconsejable. Sin embargo, la combinación de un microcontrolador con uno o varios circuitos de lógica programable de media escala de integración como pueden ser una PAL o una FPGA que se encarguen de aquellas etapas críticas de procesamiento, permiten que el sistema global pueda dar respuesta a casi todo tipo de problemas.

2.- OBJETIVOS

Vista la importancia de los microcontroladores, el objetivo persiguido consiste en mejorar la formación del alumno a nivel práctico en el diseño basado en estos dispositivos. Para la consecución de tal objetivo se ha diseñado una tarjeta de bajo costo y todo el software de desarrollo necesario para que cada alumno pueda montarla por sí mismo y operarla en su casa. El simple montaje de la placa conlleva unos beneficios prácticos para el alumno evidentes. En primer lugar comienza a perder ese miedo al mundo del hardware, que parece

extraordinariamente complicado, y se familiariza con todos los componentes electrónicos necesarios para la placa. Es muy fácil explicar cómo es y cómo funciona un decodificador o un convertidor de tensión de niveles TTL-RS232 pero estos conocimientos pueden caer en el olvido si no se practica con ellos, y qué mejor manera que tenerlos en la mano, ver cómo son de verdad y aprender cómo hay que situarlos en la placa y soldarlos.

3.- DESCRIPCIÓN DEL SISTEMA

La idea del sistema de desarrollo para el microcontrolador 80C32 de la familia MCS-51 de Intel [1, 2, 3] es poder elaborar en un ordenador un programa ejecutable por el micro y poder transmitir vía serie este programa al micro para ser ejecutado. A la vez se pretende que el sistema permita la monitorización del micro, en el sentido de que cuando éste ejecuta un programa que necesita entrada y salida de datos, el ordenador se convierte en terminal, visualizando los datos que transmite el micro y permitiendo enviarle datos que son introducidos por el teclado del ordenador.

3.1.- Hardware del sistema

La característica fundamental en este aspecto es su sencillez y bajo coste: 6.000 ptas en total: 1.200 ptas. como precio de la placa y 4.800 por los componentes.

El diseño de la placa se realizó en el Dpto. y su fabricación fue encargada a la empresa sevillana *Circuitrónica*. Se fabrica a doble capa (estañeada); la capa inferior con una capa verde aislamiento para evitar problemas de soldadura (cortocircuitos, etc) lo cual es muy conveniente ya que este montaje constituye, para la mayor parte de los alumnos, su primer contacto práctico con tal materia. El tamaño de la placa es de 13 x 8,5 cms.

Los componentes empleados son de barata y fácil adquisición en cualquier comercio de electrónica, e incluyen 1 microcontrolador 80C32 de Intel, 1 circuito de comunicación en paralelo PIA 8255, 1 memoria EPROM 8Kx8 2764, 1 memoria RAM 32Kx8 62256 para albergar datos y al programa de usuario, 1 convertidor de niveles TTL (0 a 5V) - RS232 (-10 a 10V) Maxim 232, 1 oscilador de 12 MHz, 1 regulador 7805, así como puertas Nand, zócalos, condensadores y conector de 9 pines para el puerto serie de un ordenador. La función de todo este hardware es proporcionar accesibilidad al 80C32 y dotarle de efectividad para desarrollar su labor de controlador. En la figuras 1 y 2 puede verse un esquemático general y una fotografía de la placa del sistema de desarrollo.

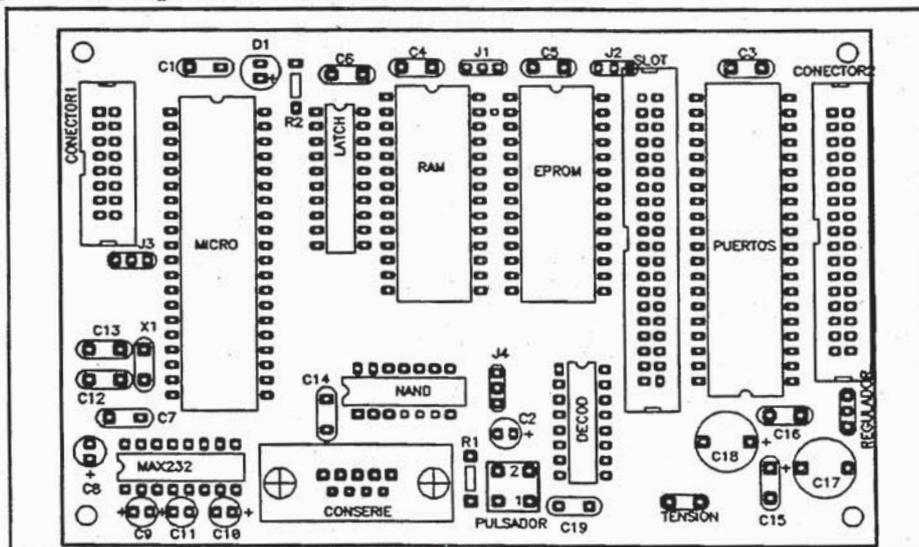


Figura 1.- Placa del sistema de desarrollo

Las características más reseñables del microcontrolador 80C32 son las siguientes: 256 bytes de memoria RAM, capacidad de direccionamiento externa de 64 Kb de memoria RAM, tres contadores/temporizadores programables, unidad serie USART, cuatro puertos de E/S de 8 bits, seis fuentes de interrupciones con dos niveles de prioridad, unidad lógica y aritmética, velocidad de 12 MHz y construcción en tecnología CMOS, lo que le permite disponer de dos modos de bajo consumo.

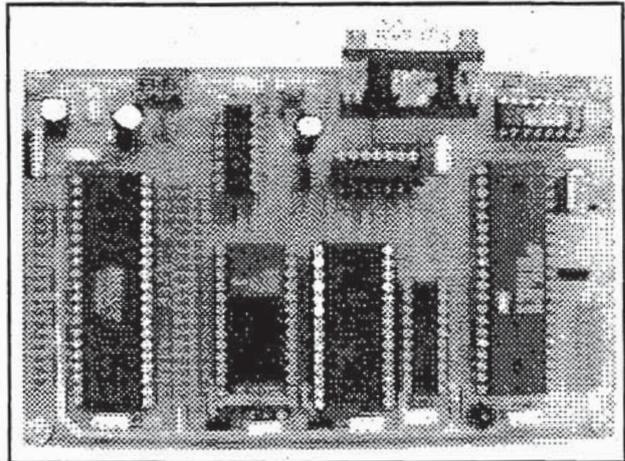


Figura 2.- Placa del sistema de desarrollo

3.2.- Software del sistema

El software del sistema de desarrollo está constituido por un programa monitor (denominado DMON) albergado en la memoria EPROM de la placa, y por el programa ESM, residente en el ordenador personal.

El programa DMON es como el sistema operativo del micro. Se encarga de la comunicación con el ordenador, recibiendo el programa que éste le transmite vía serie y alojándolo en la memoria RAM para su posterior ejecución. Posee además rutinas básicas de E/S de datos que están a disposición del usuario y un debugger o depurador que permite ejecutar programas paso a paso. Este programa, tras realizar algunas inicializaciones preliminares, se queda indefinidamente a la escucha de comandos provenientes vía serie. A la llegada de un comando, el programa responde bien enviando algún tipo de información o bien realizando alguna tarea. El monitor acepta comandos como: Ejecutar un programa, transmitir un programa y almacenarlo en la RAM externa, mostrar un bloque de memoria RAM externa, mostrar un bloque de memoria RAM interna, escribir en memoria RAM externa, escribir en memoria RAM interna, seleccionar la velocidad de comunicación vía serie y realizar la depuración o debugging. Además, DMON pone a disposición del usuario del sistema una librería de 12 rutinas de E/S de datos vía serie.

El funcionamiento del debugger es igualmente a base de comandos. Utiliza dos variables de la RAM interna del micro como contador de programa del usuario, para llevar un control de la dirección por donde va la ejecución del programa de usuario. Algunos de los comandos que acepta el debugger son: Leer por el puerto serie un valor para el contador de programa del usuario, ejecutar únicamente la instrucción apuntada por dicho contador de programa, ejecutar el programa de usuario hasta el final, ejecutar ininterrumpidamente una rutina del programa monitor, ejecutar ininterrumpidamente una subrutina del usuario, abandonar el debugger, etc. Es interesante el mecanismo de ejecución paso a paso, basado en el funcionamiento de las interrupciones. Cuando se produce una petición de interrupción en el momento en que se está atendiendo otra de igual o superior prioridad, la segunda no es atendida hasta que se termina de ejecutar la rutina de servicio de la primera. Entonces, la CPU ejecuta una única instrucción del programa principal y pasa a ejecutar la rutina de servicio de la segunda interrupción. El debugger utiliza la interrupción del puerto serie. Provoca la petición de interrupción del puerto internamente (por software). Inmediatamente se abandona el programa principal (bucle indefinido en espera de comandos) y se pasa a ejecutar la rutina de servicio. Dentro de esta rutina se provoca también internamente la segunda petición de interrupción, pero antes se prepara todo para que cuando se termina de ejecutar, la instrucción del programa principal

que se ejecute no sea aquella por la que iba cuando se produjo la interrupción (la del bucle indefinido), sino la que nosotros deseamos en el programa de usuario. Al ejecutarse la instrucción del usuario deseada, se pasa a ejecutar de nuevo la rutina de servicio para atender la segunda petición de interrupción. En este caso, la rutina no hace nada, y devuelve el control al bucle indefinido de espera de comandos.

El programa **ESM** se ejecuta en el ordenador personal, y es el interlocutor del programa **DMON**. Su programación se realizó en lenguaje C, por las posibilidades que ofrece a bajo nivel. Sus funciones principales son las de transmitir vía serie un programa ejecutable al micro y la de monitorizar a éste convirtiendo al ordenador en un terminal para la E/S interactiva de datos durante la ejecución de un programa de aplicación en el micro.

La interfase hombre-máquina de este programa corre a cargo de un sistema de menús y ventanas, presentando una pantalla con dos ventanas en la que la inferior actúa como dispositivo de salida de datos del 80C32 y la superior como menú de usuario seleccionables mediante los cursores. En la ventana principal se dispone de opciones para editar el código fuente del programa que se está desarrollando (se hace una llamada a un editor externo), ensamblarlo (se crea un fichero que contiene el código máquina del programa), transmitirlo al micro y ejecutarlo (especificando la dirección del programa en la memoria RAM). También permite ejecutar comandos del DOS sin salir del entorno. En estado de reposo, el programa muestra en la ventana inferior cualquier carácter que le sea transmitido desde el micro. Están activas tres teclas de función:

- a) F10: el teclado pasa a convertirse en un dispositivo de E/S de datos para el micro, siendo transmitido vía serie el código ASCII de cualquier tecla que se pulse. El teclado vuelve a su normal funcionamiento cuando se pulsa de nuevo F10.
- b) F2: provoca un reset en la tarjeta.
- c) F1: da acceso al menú de configuración. Es posible elegir el puerto serie usado para la comunicación, su velocidad, el formato en que es generado por el ensamblador el fichero de código máquina (Intel, Motorola o Binario), si utilizar o no el debugger, el editor externo a usar para editar el programa de aplicación, etc. Toda esta información se almacena en un fichero de configuración.

El entorno de depuración consta de tres ventanas: ventana de comunicación (muestra todos los caracteres transmitidos), ventana de estado (muestra información sobre el estado de la ejecución) y ventana de información (muestra contenidos de registros, memoria RAM externa o interna y variables definidas en RAM externa, en formatos binario, hexadecimal o decimal. Es posible modificar los contenidos de registros y variables). Se dispone de una opción de ayuda. Las operaciones que permite realizar el debugger, a través de la pulsación de determinadas secuencias de teclas, son: Ejecutar hasta el final, ejecutar hasta una cierta dirección, interrumpir la ejecución, ejecutar una sola instrucción, ejecutar de corrido una subrutina como si fuese una sola instrucción, introducir un break-point en una dirección y eliminar break-points. Toda la información de funcionamiento del debugger se almacena en un fichero de configuración, lo cual es muy cómodo tras alguna situación problemática que obligue a resetear al micro y al ordenador.

4.- METODOLOGÍA

El montaje y testeo del sistema de desarrollo corre a cargo de cada alumno. Para ello, cuenta con las herramientas disponibles en el laboratorio y con el asesoramiento e instrucción del profesor. A continuación se especifican las actividades a realizar:

- a) Estudio teórico en clase del micro 80C32, de la placa de desarrollo y del software. En esta fase se facilita a los alumnos bastantes ejemplos en ensamblador de cómo manejar los

periféricos, tanto los internos del micro como la PIA externa. Se estudia también el programa monitor de la EPROM. Los alumnos utilizan para probar los ejemplos de clase un simulador del micro denominado *Emily*, disponible en Internet.

b) Montaje individual de la placa.

c) Testeo de la placa.

d) Programación y prueba de alguna aplicación concreta en el sistema real [2].

Todo este trabajo realizado por el alumno se valora dentro de la correspondiente asignatura. En el segundo parcial se valora un 50% la nota del examen y el otro 50% por un proyecto de mediana complejidad realizado por grupos no superiores a tres alumnos. En cada proyecto cada grupo debe diseñar una tarjeta de aplicación que será controlada por la tarjeta microcontroladora, debiéndose programar el micro para que controle el hardware realizado. Algunos ejemplos de proyectos realizados por los alumnos como aplicación del sistema de desarrollo son: *Cerradura electrónica con control de acceso; Control de lámparas por ángulo de disparo; Programador de EPROM, E2PROM y RAM "viva"; Control de una máquina expendedora automática; Polímetro, etc.*

Toda la experiencia adquirida por el alumno en el sistema de desarrollo es aprovechada además en las prácticas de la asignatura *Electrónica Aplicada* de 6º curso de la titulación de Ingeniería Industrial (especialidad Eléctrica), en el control de tiristores.

5.- RESULTADOS

Esta experiencia se inició en el curso 1994-95. Se montaron un total de 92 kits del microcontrolador de los cuales 36 correspondieron a la asignatura *Electrónica Industrial* de 5º curso de la titulación de Ingeniería Industrial (especialidad Eléctrica), y 56 a la asignatura *Sistemas Electrónicos Digitales* de 3er curso de la titulación de Ingeniería de Telecomunicaciones. Este número de placas corresponde a un 60% y a un 30% del número de alumnos matriculados en cada una de las respectivas asignaturas. En el segundo año el interés ha crecido extraordinariamente, montándose un total de 157 placas de las cuales 59 han correspondido a la asignatura *Electrónica Industrial*, lo que supone un 90% del alumnado, y las 88 restantes a *Sistemas Electrónicos Digitales* lo que supone un 50% del total. Además, grupos de alumnos de otras especialidades también se han interesado en montar su propio sistema y actualmente siguen las clases de *Electrónica Industrial* como alumnos oyentes, lo que constituye buena prueba del gran interés y aceptación del sistema realizado.

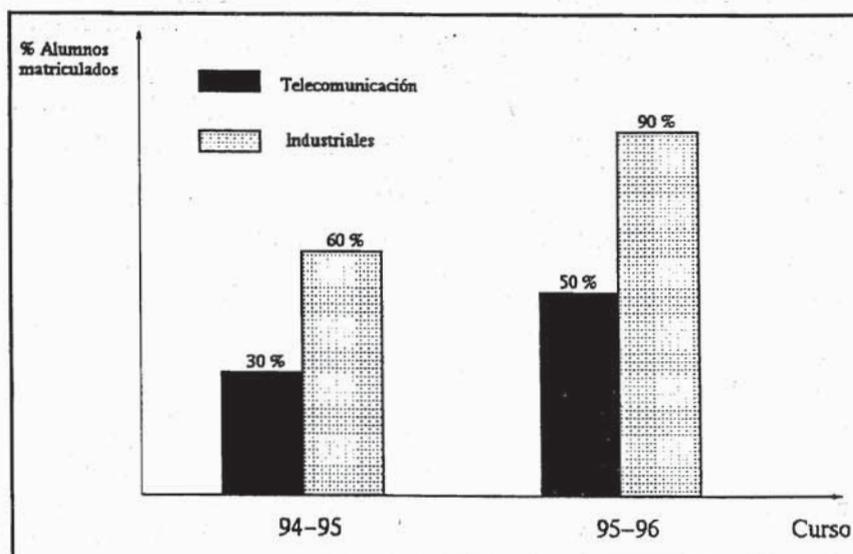


Figura 3.- Estadísticas de uso del sistema

Al final del curso pasado se pasaron unas encuestas a los alumnos para valorar la nueva experiencia del sistema de desarrollo, siendo la respuesta global muy positiva.

Para el profesor resulta un estímulo que nuevas iniciativas en la docencia tengan una gran aceptación, lo que anima a mejorar todo el sistema de desarrollo. Este segundo año la placa es una versión mejorada en la que se ha variado el mapa de memoria, se contempla la posibilidad de dar un reset al micro desde el ordenador personal, se dispone de 7 líneas de activación de periféricos en el slot de expansión y se ha disminuido en buena medida el tamaño de la placa.

Finalmente, destacar que el resultado obtenido más importante es, sin duda, el afianzar en los alumnos, por medio de la práctica, los contenidos teóricos impartidos en clase.

6.- FUTUROS TRABAJOS Y AMPLIACIONES

A nivel hardware el sistema no se cambiará pues posibles mejoras supondrían la introducción de nuevos componentes que encarecerían el precio de la placa y podrían hacerla prohibitiva para una parte del alumnado. Las posibles mejoras se enfocan pues al software: realizar un nuevo ensamblador y mejorar el sistema de depuración, en su interfase hombre-máquina y en el método de ejecución paso a paso, que no será por interrupciones, sino por software (se podría decir que el micro se simularía a sí mismo). Esta labor, fundamentalmente informática, correría a cargo de un grupo de alumnos voluntarios interesados, debido a que cae fuera de los objetivos docentes de la asignatura. No obstante, sería de gran ayuda para la realización de futuras aplicaciones prácticas utilizando el sistema de desarrollo. Actualmente está en proceso de diseño un sistema de desarrollo más potente basado en el conocido microcontrolador 68HC11 de Motorola.

7.- REFERENCIAS

- [1] Martínez Sánchez, V. A. "Desarrollo y programación de sistemas digitales. Familia de microprocesadores Intel MCS-51. 8031-8051-8751". Ed. Ra-Ma, 1993.
- [2] Yuralan, S. y Ahluwalia, A. "Programming and Interfacing the 8051 Microcontroller". Ed. Addison-Wesley, 1995.
- [3] "Microcomputer Components. SAB-C501. 8-bit CMOS Microcontroller". Siemens.