

DISEÑOS AUTÓMOS CON MICROPROCESADORES

B. VAZQUEZ, L. CLOSAS

Departament d'Enginyeria Electrònica. Universitat Politècnica de Catalunya. c/Gran Capitan s/n Campus Nord mòdul C4 08071 Barcelona. closas@eel.upc.es

Con la óptica de dotar al estudiante de autonomía en el diseño de sistemas que usan microcontrolador, suministramos al alumno los programas más comunes y un programador de un microcontrolador de amplio uso comercial con el que podrá hacer aplicaciones de forma fácil i económica. El diseño del programador es el objeto de este trabajo

1. Introducción

Al realizar una aplicación con microcontrolador nos encontramos con la necesidad de grabar el código de la aplicación en un soporte no volátil.

El programador desarrollado, que después describiremos en detalle, consta de una placa que nos genera las señales necesarias para programar un 87c51 junto con el software de grabación de dicho microcontrolador. El control del proceso de grabación lo realizamos desde el PC, pudiendo se mirar si la EPROM está borrada, cargar el fichero para su grabación, leer la EPROM y desencadenar el proceso de grabación.

2. Funcionamiento del circuito

En la figura 1 vemos el esquema propuesto para la placa del grabador. Básicamente la placa consta de dos microcontroladores 87c51 y una fuente de alimentación que nos suministra los $12,75v \pm 0,25$ que necesitamos en el proceso de grabación de la EPROM interna [2] del 87c51.

El primer microcontrolador (U1) (por medio de sus patillas R_x , T_x , y con ayuda del circuito integrado U9) es el encargado de comunicarse con el PC, y guardar el programa a grabar en la RAM (U3) externa de 32Kbytes.

Además este microcontrolador tiene conectado directamente una interfase 8255 (U6) que nos permite ampliar el número de puertos de salida. Pasando de tener cuatro puertos a tener seis. Observe que el \overline{CS} es activado por el bit A₁₅ del bus de direcciones. Todos los movimientos a RAM externa a partir de la posición 8000H activarán el 8255. Tal como se puede apreciar en la figura anterior. En reposo, es decir, si no ejecuta ninguna tarea, el microcontrolador (U1) esta ejecutando un bucle, en espera de la llegada de un byte de control.

```

; Menú de selección de proceso. Bucle principal
MENU2: SJMP MENU
        JNB TI,$
        CLR TI
        MOV A,#NACK
        MOV SBUF,A
MENU:   MOV R0,#0;Inicialización de los registros
        MOV R1,#0; R0, R1 y R4.
        MOV R4,#pulsos;Número de pulsos
PROG\ . JNB RI,$;Espera recibir indicador del proceso a
        CLR RI; seguir.
        MOV A,SBUF
        CJNE A,#'B',S1
        MOV A,#ACK
        JNB TI,$
        CLR TI
        MOV SBUF,A
        SJMP BLANCO
S1:    CJNE A,#'L',S2
        MOV A,#ACK
        JNB TI,$
        CLR TI
        MOV SBUF,A
        SJMP LECTURA
S2:    CJNE A,#'ENG',MENU2
        SJMP GRABAR

```

Cuando el usuario del PC manda una orden por el puerto serie se activa el flag RI de recepción del puerto serie del microcontrolador (U1), provocando su salida de un bucle (que está ejecutando indefinidamente a través de la instrucción JNB RI,\$) y pasa a ejecutar una rutina diferente para cada una de las tres órdenes posibles, ejecutando saltos a las rutinas de BLANCO, LECTURA y GRABAR: Una vez se ha producido el volcado del fichero a grabar, que ha de estar en formato Intel hexadecimal, desde el PC, el microcontrolador U1 genera las señales necesarias para la programación del 87c51 (U8). Las direcciones, datos y señales de programación (P_{2.7} "señal de control" y PROG "señal de grabación") necesarias.

Con el código cargado en la RAM de la placa, se inicia la programación del microcontrolador. Para esto, se leen los datos de la RAM y son direccionados a la EPROM del microcontrolador con la ayuda de la interfase 8255, del cual solo utilizamos dos de los tres puertos disponibles, suficiente para direccionar hasta un total de 16Kbytes.(PA₀-PA₇ y PB₀-PB₅) Por último, se comprueba que la grabación ha sido correcta, comunicandolo al PC. Con el siguiente programa generamos las señales necesarias [1] para programar el microcontrolador:

1. Activación señal P2.7, la cual permite la escritura de la memoria.
2. Direccionamiento del byte a grabar a través del 8255.
3. Carga del byte a grabar del micro U1 en el puerto P1.
4. Conmutación de V_{pp} de 5 a 12,75V (mediante P3.5).
5. Inicialización del timer de generación de pulsos de grabación y activación.
6. Inicialización de la señal PROG (señal de grabación)

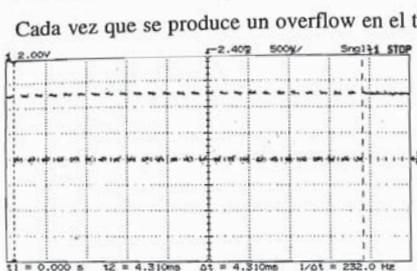
```

rutina de programación de la eeprom
PROGRAM: MOV A,R1
        MOV DPTR,#PB55
        MOVX @DPTR,A
        MOV P1,R2
        SETB P3.5
        MOV TH0,#0AFH
        MOV TL0,TH0
        LCALL TIEMPO
        SETB TR0
        CLR P3.2; ;PROG\
        RET
;final rutina de programación de la eeprom

```

En la rutina anterior se lanza un pulso de programación. Se activan las señales de control y se dispara el contador que nos da la duración del pulso de programación. Al cumplir el tiempo se

ejecuta la interrupción que pulsará a cero la señal de grabación. Para grabar cada byte se necesitan 25 pulsos como estos. Los pulsos necesarios para la programación según el fabricante [2] son los de la figura 2:



Cada vez que se produce un overflow en el timer0 se provoca una interrupción que genera un pulso de grabación. Al producirse el overflow se ejecuta la rutina de interrupción del timer0 que describimos a continuación. Esta rutina provoca que el microcontrolador vaya conmutando la señal de programación, PROG, a través del puerto P3.2. A la vez que se va decrementado el contador de pulsos hasta conseguir 25 pulsos necesarios a nivel bajo.

Figura. 2 Pulsos de grabación del 87c51

```
; rutina de interrupción
PULSO: CPL P3,2
      DEC R4
      CJNE R4 #0,SEGUIR
```

```
CLR TR0
MOV A,#0
SEGUIR:RETI
;final de la rutina de interrupción
```

3. Software del programador

Para que el programador pueda funcionar hemos desarrollado un programa de comunicaciones (menu2.exe) instalado en un PC que nos permite controlar todo el proceso de grabación de un microcontrolador del tipo 87c51. En la figura 2 mostramos la pantalla de un PC en el que estamos ejecutando el programa de comunicaciones menu2.exe. Además, para que la placa pueda comunicarse con el PC hemos desarrollado otro programa de comunicaciones que controla la placa (grabador.asm) y que está instalado en la memoria EPROM de la placa. Partes de dicho programa ya hemos descrito anteriormente en el apartado 2 de este trabajo.

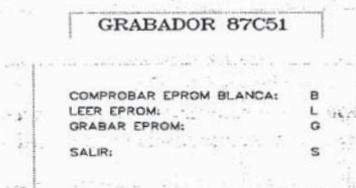


Figura 3: Entorno de grabación

4. Conclusiones

Se ha realizado un diseño de forma fácil y económica que es de gran utilidad al alumno para motivarle y para que pueda construir diseños con pocos medios.

Referencias

- [1] J.A. González Vázquez . Introducción a los microcontroladores 8x52 y 8x51. McGraw 1993
- [2] 80C51 Based 8 bit microcontrollers. Data Book Philips 1997