

Programador para microcontroladores cop8

Introducción:

El proyecto consta de la implementación de un programa (en realidad de dos) que permiten la programación y lectura de los microcontroladores cop8 de Nacional Semicundutors, debido a que estos dispositivos no son muy populares en comparación a los PIC'S o AVR'S, casi no se encuentran alternativas gratuitas y/o económicas tanto de hardware como software para su programación. La forma de programación consta de un conjunto de señales y niveles de voltajes específicos que se aplican a ciertos pines del dispositivo. Cada función leer, escribir, comprobar datos del microcontrolador debe cumplir con una secuencia de señales desde y hacia el dispositivo.

Solución del problema:

El software se encarga de generar las secuencias de los pulsos necesarias para la lectura o programación del dispositivo, esto se hace a través del puerto paralelo el cual se conecta por medio de un cable al hardware implementado. Toda la información sobre las secuencias ocupadas, pines del microcontrolador a ocupar, niveles de voltajes, tiempos mínimos entre cada evento etc... fueron recogidas del paper "Programming Specification and Application Notes for In Circuit Programming (ICP) and Personal Programmer Design" de Nacional. Los pines del puerto paralelo son mapeados en una variable, es decir cada pin es representado por un bit dentro de la variable, un 1 lógico dentro de un bit de la variable corresponderá a VCC en el pin asociado, y un 0 lógico corresponderá a GND. También se tiene en cuenta de que existen pines de salidas (PC -> μ C) y de entradas (μ C -> PC), a continuación se muestra el esquema escogido para la asociación de los pines de salida con los bits de la variable.

bits de la variable	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
							V P P	G 6	V C C 1	V C C 0	P G M E	H O L E	S E L E	P C I D D	A N K	C L K								
DB-25F Pines							17	16	14	1	9	8	7	6	5	4	3	2						
bits de los puertos							7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0		
Dirección de los Puertos en hexadecimal							37A				378													
							3BE				3BC													
							27A				278													

Se observa de la figura los nombres escogidos para cada bit clk, din, pcadd etc....estos corresponden a los mismos nombres definidos en el paper.

Para asociar un bit de la variable a un pin específico se hace a través de un objeto de la clase Pinout, el bit al cual representará este objeto se pasa como argumento del constructor, por ejemplo:

Pinout clk(1), crea un objeto que representa al bit 1 de la variable o lo mismo decir al pin 3 del puerto paralelo. El constructor tiene un segundo parámetro como argumento que elige al tipo de configuración de puerto que posee el PC, ya que existen tres direcciones de memoria por cada puerto LPT1, LPT2 y LPT3, el orden de las direcciones dependen del PC, en la mayoría de los casos el puerto LPT1 ocupa las direcciones 378, 379 y 37A (valore en hexadecimal), por defecto el constructor setea esta configuración, por lo que en la mayoría de los casos no hay necesidad de ingresar este parámetro. Los métodos para esta clase son los siguientes:

```
void setval(unsigned int v, unsigned int *out);  
void resetall(unsigned int *out);
```

setval setea el bit asociado al valor lógico "v" dentro de la variable pasada por referencia en el segundo parámetro.

resetall pone en 0 a todos los bit de la variable.

También existe la clase Pinin que permite crear un objeto asociado a un pin de entrada de la misma forma que el caso anterior. Existe un solo método para esta clase `unsigned int getval(void)` este retorna el valor lógico del pin asociado.

Otra clase que se define denominada **intelhex** permite la lectura de un archivo con formato hexadecimal de Intel. Esta clase es heredada de ifstream y permite la creación del objeto de la siguiente forma;

```
intelhex input("archivo.hex");
```

Esta clase define dos métodos:

```
int getdec(int &n)  
int getnb(void)
```

El primero obtiene el siguiente byte del programa en forma decimal. Y el segundo retorna el número de bytes que contiene el código dentro del archivo.

Modo de uso:

Como se dijo anteriormente este proyecto se compone de dos programas, una para la programación del dispositivo (pgmcp8.exe) y otro para la lectura (readcp8.exe). Al ejecutar el primero este pregunta por el nombre del archivo en formato hexadecimal, luego despliega el número de bytes que contiene el programa y además pregunta por el registro de configuración (econ) necesario para el microcontrolador, luego comienza con la programación.

El programa de lectura solo pide que se ingrese el número de bytes que se desea leer.