Congr. Int. Ing. Electrón. Mem. Electro 2012, vol.34, pp. 282-287 Chihuahua, Chih. México
http://depi.itchihuahua.edu.mx/display/memorias_electro/MemoriaElectro2012.zip
ISSN 1405-2172

MINICON_08GT, TARJETA PARA DESARROLLO Y APRENDIZAJE CON EL MICROCONTROLADOR MC9S08GT60

Antonio Salvá Calleja, Víctor Manuel Sánchez Esquivel, Luis Antonio Altamirano Yépez
Facultad de Ingeniería, UNAM, División de Ingeniería Eléctrica
Ciudad Universitaria D.F., CP 04510
Tels: (55) 5622 – 3108, (55) 539 – 0879 Fax: (55) 5616 – 1855
salva@dctrl.fi-b.unam.mx, victor@dctrl.fi-b.unam.mx, altamirano@dctrl.fi-b.unam.mx

RESUMEN.
Los microcontroladores de la familia S08 son dispositivos fabricados por *Freescale*, versátiles y de bajo consumo de energía. Contienen puertos binarios, memorias RAM, EEPROM y FLASH, además de periféricos de uso frecuente en instrumentación y control. En este trabajo se presenta el desarrollo una tarjeta denominada MINICON_08GT, diseñada en el Departamento de Control y Robótica de la Facultad de Ingeniería de la UNAM, que permite al usuario tener herramienta integrada de fácil uso, mediante la cual se puedan realizar aplicaciones basadas en el microcontrolador MC9S08GT60.

Figura 1. Bloques funcionales de desarrollo PUMMA_08+ y MINICON_08GT.

1. INTRODUCCIÓN.

En la figura 1 se muestran los bloques funcionales que integran el sistema de desarrollo con la tarjeta MINICON_08GT. Éste está integrado por una computadora PC donde se ejecuta un programa denominado PUMMA_08+, el cual realiza la interfaz de usuario (cuenta con un ensamblador cruzado denominado ENS08 y un compilador cruzado de BASIC denominado MINIBAS8A). En este caso la tarjeta MINICON_08GT es la arquitectura destino (AD) sobre la que se desarrolla una aplicación, o bien, se prueben programas. Las características de la tarjeta MINICON_08GT son:

1. Está basada en el MCU MC9S08GT60 fabricado por FREESCALE.
2. Contiene Firmware Interlocutor Residente (FIR), denominado NBCP8 (Núcleo Básico de Comunicaciones con PUMMA_08+), compatible con el software manejador PUMMA_08+, mediante el cual se puede hacer depuración y cargado de programas y/o datos en memoria RAM o FLASH del MCU presente en la tarjeta.
3. Contiene postes (headers) a líneas de entrada y/o salidas de puertos y diversos puntos de prueba. En la figura 2 se muestra una fotografía de la tarjeta destino.
4. Contiene interfaz para desplegados alfanuméricos populares en la industria.
5. Capacidad para ejecución autónoma de programas.
6. Contiene el arreglo de seis postes estándar para interfaz del MCU por medio de hardware y software populares en la industria.

Como se puede ver en la Figura 1, al conjunto de bloques funcionales conformado por la tarjeta MINICON_08GT y el software manejador PUMMA_08+ se le denomina Ambiente Integrado para Desarrollo y Aprendizaje con microcontroladores 08, se denota AIDA08 por sus siglas.

Figura 2. Tarjeta destino MINICON_08GT

ISSN 1405-2172

2. SOFTWARE PUMMA_08+.

Las características principales del software manejador son:

- Ejecutable bajo WINDOWS (98/M/XP/V/7)
- Capacidad para cargar en la AD archivos S19, con datos o programas en lenguaje de máquina.
- Capacidad para ejecutar un programa previamente cargado en la AD.
- Capacidad para escribir datos a memoria o puerto en la AD.
- Capacidad para examinar memoria o puertos.
- Capacidad para programar la memoria FLASH contenida en el MCU presente en la AD.
- Contiene un editor de texto básico para que el usuario pueda escribir y almacenar programas.
- Contiene un ensamblador cruzado, denominado **ENS08**.
- Contiene un compilador cruzado de lenguaje **BASIC**, denominado **MINIBAS8A**.
- Capacidad para obtener, a partir del código fuente en ensamblador, presente en la ventana del editor, el archivo S19 que contiene el código de máquina ejecutable en el MCU de la AD.
- Capacidad para obtener, a partir del código fuente en BASIC, presente en la ventana del editor, el archivo S19 que contiene el código de máquina ejecutable en el MCU de la AD.
- Capacidad para ejecutar de inmediato el código generado a partir de un programa fuente en ensamblador y presente en la ventana del editor.
- Capacidad para ejecutar de inmediato el código generado a partir de un determinado programa escrito en lenguaje BASIC y presente en la ventana del editor.
- Contiene un emulador de terminal mediante el cual se realiza la consola de interfaz con el usuario para fines de la ejecución de programas que la requieran.
- Contiene un cargador en la AD de archivos objeto S19 que hayan sido generados por otro software de compilación y/o ensamble tal como CodeWarrior u otro.

3. ASPECTOS GENERALES DEL FUNCIONAMIENTO DE LA AD.

En la tabla 1 se muestra el mapa de memoria del MCU de la tarjeta destino, donde se aprecia el núcleo básico de comunicación de PUMMA_08+ (NBCP8) y las bibliotecas del compilador MINIBAS8A (BIBAS8), de ahí su valor agregado. Al conjunto conformado por NBCP8 y BIBAS8, se le llama software de base NBCP8_BIBAS8. A un circuito integrado MC9S08GT60 con este software de base se le llama Chipbas8_GT60. El NBCP8 se localiza en dos intervalos disjuntos de la memoria FLASH, los cuales son:

Intervalo 1 del NBCP8_BIBAS8, el cual está comprendido de la dirección 1080 a la dirección 17FF, donde radica la primera parte del conjunto de rutinas en ensamblador propias de MINIBAS8A.

TABLA 1. MAPA DE MEMORIA DEL Chpbas8 GT60.

Intervalo de direcciones (hex)	Contenido
0000 – 007F	Zona 1 de registros de control y operación (RCO)
0080 – 107F	Memoria RAM (4K)
1080 – 17FF	Zona 1 de firmware NBCP8_BIBAS8
1800 – 182B	Zona 2 de registros de control y operación (RCO)
182C - DFCB	Memoria FLASH de usuario
DFCC - DFFF	Vectores de usuario
E000 - FFFD	Zona 2 de firmware NBCP8_BIBAS8
FFFE - FFFF	Vector de reset del firmware NBCP8_BIBAS8

Intervalo 2 del NBCP8_BIBAS8, el cual está comprendido de la dirección E000 a la dirección FFFD, donde está colocada la segunda parte de las rutinas en ensamblador propias del compilador MINIBAS8A. Además de esto, en esta zona radica el firmware NBCP8, que contiene el receptor de comandos enviados por el usuario mediante el software manejador PUMMA_08+. Estos podrían ser entre otros: el requerimiento de lectura de una zona de memoria, el borrado de la memoria FLASH disponible para el desarrollo de las aplicaciones, o bien el grabado en memoria FLASH del código asociado con una determinada aplicación, previamente generado con las facilidades de ensamble y/o compilación propias del software PUMMA_08+, o bien otro software.

3.1. ACCIONES AL RESET DE LA AD.

Dependiendo del status lógico del bit PTC3 del puerto C del MCU y del contenido de las localidades cuyas direcciones son DFFE y DFFF,

ISSN 1405-2172

al darse un RESET podrán efectuarse una de las siguientes dos acciones:

1. Ejecución del receptor de comandos del NBCP8, lo cual es testificado por el parpadeo (blinking) del LED 1 presente en la tarjeta, esto hace que el usuario pueda efectuar diversos comandos desde el manejador PUMMA_08+.
2. Ejecución autónoma de un programa precargado en memoria FLASH a partir de la dirección denotada por el contenido de las localidades DFFE y DFFF a la cual se le denomina como ***vector de RESET de usuario***.

El nivel lógico del bit PTC3 es gobernado por un *jumper* excluyente presente en la tarjeta. Cuando se coloca un puente del lado con la leyenda PUMMA_08+ el nivel lógico de PTC3 será alto y se efectuará la acción uno. Cuando el puente es colocado del lado con la leyenda EA el nivel lógico de PTC3 será bajo, efectuándose la acción dos descrita anteriormente.

El software de base NBCP8_BIBAS8, el manejador PUMMA_08+ y el compilador cruzado MINIBAS8A fueron diseñados en el Departamento de Control y Robótica de la Facultad de Ingeniería de la UNAM. Para una descripción detallada de estos desarrollos, véanse las referencias [1], [2] y [3].

4. EJEMPLO ILUSTRATIVO.

Empleando la tarjeta se han desarrollado diversos ejemplos didácticos de aplicación. Para fines ilustrativos aquí se muestra la realización de un control simple en modo deslizante de una planta de orden dos con polos en -1.031 y -2.11. En la figura 3 se muestra el esquema de lazo cerrado implicado; donde el controlador es realizado con la tarjeta MINICON_08GT.

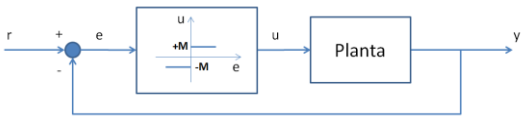

Figura 3. Control simple en modo deslizante de una planta de orden dos.

La función de transferencia (FT) de la planta es:

$$G_p(S) = \frac{1}{(1+.474S)(1+.97S)} \ldots\ldots(1)$$

Físicamente la planta presente en la figura 3 se realizó empleando el circuito mostrado en la figura 4.

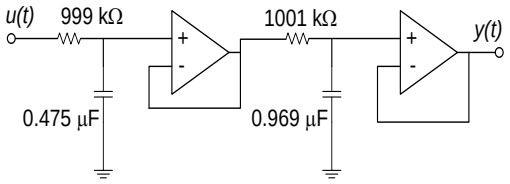

Figura 4. Planta de segundo orden

Puede verse que para cambios de tipo escalón en la señal de referencia r(t), se debe satisfacer la siguiente desigualdad:

$$|r(t)| < M \ \ldots\ldots\ldots (2)$$

Realización física del controlador

Para realizar físicamente el controlador que aparece en la figura 3, empleando para ello a un microcontrolador, es conveniente desarrollar el software implicado empleando un lenguaje de alto nivel. A continuación se describe en lo fundamental como se hizo esto usando el BASIC validado por el compilador cruzado MINIBAS8A. En primera instancia se requiere desarrollar una rutina, aquí denominada alconmd, ejecutable en el microcontrolador empleado, que efectúe las siguientes acciones:

1. Leer el convertidor análogo-digital (CAD) asociado con la señal analógica de error e(t) asignándose el valor entregado por el CAD a una variable entera aquí denominada byconv%.
2. Obtener el valor real de e(t) y asignar éste a una variable real de precisión sencilla aquí denominada ep.
3. Evaluar con la función signum la variable up empleando la siguiente expresión:

$$up = M \cdot \mathrm{sgn}(ep) \ldots\ldots\ldots \textbf{(3)}$$

4. Calcular el valor entero que se requiere colocar en el puerto de salida del microcontrolador que está ligado con el convertidor digital analógico (CDA), de modo que la salida de éste sea up. Este valor

ISSN 1405-2172

entero aquí se le asigna a la variable ubin%. De esta forma:

$$ubin\% = 12.75 \cdot (up + 10) \text{ .....(4)}$$

5. Se coloca en el CDA el valor entero calculado en el paso anterior.
6. Se retorna de la subrutina.

A continuación se muestra el código fuente de la rutina alconmd:

```
alconmd:
    gosub convad                ‘Paso 1.
    ep=.0784313*byconv%-10.     ‘Paso 2.
    up=md*sgn(ep)               ‘Paso 3.
    ubin%=12.75*(up+10.)        ‘Paso 4.
    iniens                      ‘Paso 5.
    lda ubin%+1
    sta pta@
    finens
    return                      ‘Paso 6.
```

La rutina alconmd es la que realiza propiamente el algoritmo de control implicado, y ésta ha de invocarse a intervalos iguales al periodo de muestreo. Para lograr lo anterior, en la fase de inicialización del software ejecutable en el microcontrolador del dispositivo Chipbas8gt60, se configura el temporizador de éste, de modo que se genere por medio del evento de sobreflujo una interrupción periódica a intervalos iguales al periodo de muestreo, y desde la rutina de servicio de interrupción asociada se invoca la rutina alconmd. Dada la temporalidad de la dinámica de la planta se asumió un período de muestreo de 10 [*ms*].

En la figura 5 se muestra un esquema a bloques de un sistema de lazo cerrado, donde el controlador es realizado en forma digital con la tarjeta MINICON_08GT. Se supone que el rango de valores que puede tomar la variable *e(t)* está comprendido entre -10 y +10 volts.

Mediante el bloque de adecuación de entrada (AE), este rango se transforma a otro que va de 0 a 3.3 volts, el cual es propio de los canales de entrada del convertidor analógico – digital (CAD) presente en el MCU. El AE se realizó empleando un circuito analógico basado en dos amplificadores operacionales.

Como interfaz de salida se emplea un convertidor digital analógico (CDA), el cual se conecta a un puerto de salida binario del microcontrolador empleado.

Como es sabido, en todo sistema de procesamiento digital, se requiere un filtro analógico antialias, el cual debe ser de tipo paso bajas, de modo que componentes de frecuencia espúreas cuyo valor sea mayor que la mitad de la frecuencia de muestreo sean eliminadas. Este bloque funcional debe filtrar la señal analógica *e(t)* a procesar por el microcontrolador antes de que ésta sea discretizada.

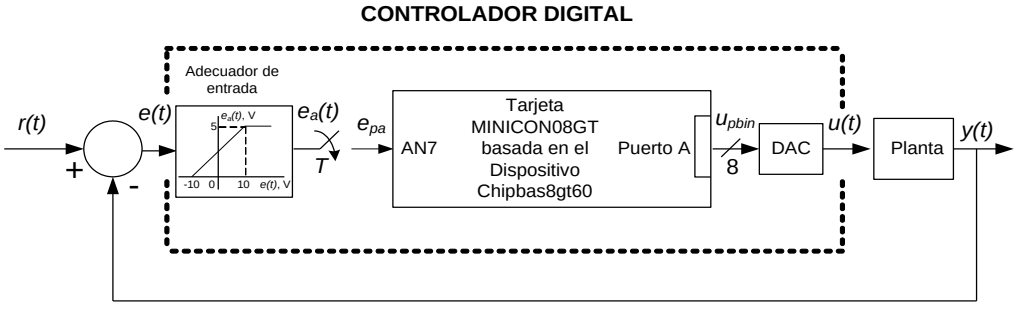

Figura 5. Sistema de control realizado con un dispositivo Chipbas8gt60

Para implementar el restador que aparece en la figura 5 se empleó el circuito mostrado en la figura 6. Éste está basado en un amplificador operacional configurado como restador. El interruptor S1 permite que la señal de referencia sea local o remota. En este último caso la señal de referencia es proporcionada al dispositivo chipbas8gt60 desde otra computadora ligada con éste de alguna forma.

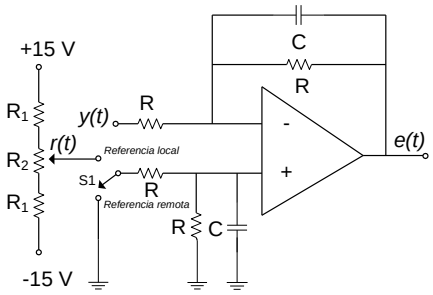

Figura 6. Circuito restador y de generación local de señal de referencia r(t).

En el dominio de Laplace, la relación entrada-salida del circuito mostrado en la figura 5 es la siguiente:

$$\frac{E(S)}{D(S)} = \frac{1}{RCS+1} \quad \ldots\ldots\ldots (5)$$

Cuando el interruptor S1 está en la posición de referencia local, la señal *e(t)* es la diferencia entre la señal de referencia local *r(t)* y la señal *y(t)* que representa a la variable a controlar. Para S1 en la posición de referencia remota, la señal *e(t)* es simplemente la señal *y(t)* con el signo invertido, efectuándose por software en el MCU la suma requerida para la obtención de la señal de error verdadera.

De acuerdo con la ecuación (5), claramente se aprecia un efecto de filtrado analógico paso bajas que realiza el filtro antialias requerido para fines del procesamiento digital implicado en el controlador digital descrito en este trabajo. Para los valores numéricos de los elementos del restador, la frecuencia de corte del filtro es 3.18 [Hz], lo que para un periodo de muestreo de 10 [ms] implica que para frecuencias mayores o iguales a la mitad de la frecuencia de muestreo, la atenuación del filtro antialias sea mayor que 23.9 dB.

Para fines ilustrativos, aquí se muestran algunos resultados experimentales. En las figuras 7 y 8 se observan respectivamente, la salida del controlador y la variable a controlar, cuando la señal de referencia es una onda cuadrada bipolar con temporalidad adecuada para los fines demostrativos requeridos.

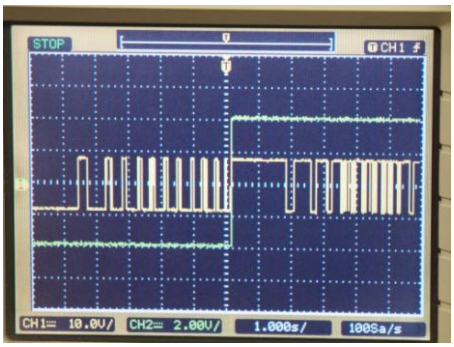

Figura 7. Señal de control para el sistema de la figura 3.

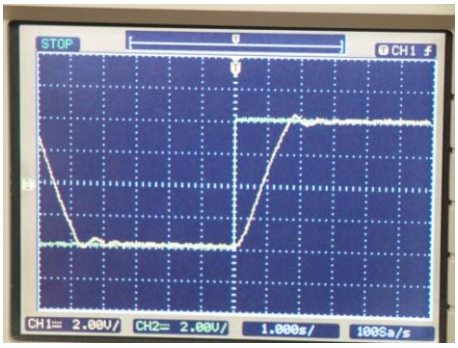

Figura 8. Variable a controlar en el esquema de la figura 3.

Para más detalles acerca de las herramientas de software y hardware empleadas en el diseño y construcción del controlador aquí ejemplificado, pueden verse las referencias [1], [2] y [3].

5. CONCLUSIONES.

La tarjeta MINICON_08GT y el software PUMMA_08+ es un binomio hardware - software que permite desarrollar aplicaciones basadas en microcontroladores S08 de FREESCALE; efectuándose esto de una manera ágil y sencilla para el diseñador. Entre los componentes que integran el sistema, puede decirse que el compilador MINIBAS8A, presente en el software manejador PUMMA_08+, es la aportación más relevante, sin restar importancia a los otros componentes.

En la industria existen herramientas más sofisticadas para el diseño de aplicaciones basadas en MCU's HC08; pero éstas están orientadas en cuanto a lenguajes de alto nivel únicamente al lenguaje C, que se ha vuelto un estándar de programación para estos dispositivos, desde luego muy poderosa y transportable, pero no es la única

forma de programar una computadora como podría ser el CPU de un microcontrolador S08.

El contar con la tarjeta MINICON_08GT permite al usuario desarrollar el software de su aplicación y probar su funcionalidad en ésta; sin embargo, si el usuario desea cargar en otra tarjeta el programa que haya desarrollado empleando ENS08 y/o MINIBAS8A, simplemente puede generar el código ejecutable (archivo np.s19) y colocar éste en el MCU S08 que use su producto, empleando para ello el cargador (downloader) de su preferencia.

6. REFERENCIAS

[1] Salvá Antonio y Altamirano Luis (2009). DISPOSITIVOS CHIPBAS8, MICROCONTROLADORES HC08 PROGRAMABLES EN LENGUAJE BASIC. Memoria del Simposio Anual de Automatización, Electrónica e Instrumentación, (SAAEI 2009). Celebrado en julio de 2009 en la Universidad Carlos III en Madrid España. El artículo puede verse en el archivo chipbas8_art_saaei09_def.pdf descargable desde http://dctrl.fi-b.unam.mx/~salva.

[2] Salvá Antonio (2011). Manual de usuario del sistema AIDA08. Archivo muaida08ve2011.pdf, descargable desde http://dctrl.fi-b.unam.mx/~salva

[3] Salvá Antonio y Altamirano Luis (2008). SAD_908, SISTEMA PARA APRENDIZAJE Y DESARROLLO CON MICROCONTROLADORES DE LA FAMILIA 68HC908. Memoria de ELECTRO 2008, Chihuahua México.

[4] FREESCALE, Data Sheet MC9S08GXXXA Microcontrollers, 2008. Archivo MC9S08GB60A.pdf descargable desde la página de FREESCALE. www.freescale.com