

# E2HC08: Entorn de Desenvolupament Lliure per al Microcontrolador 68HC08 per a GNU/LINUX

Agustí Fontquerni, Andreu Moreno, Francesc Trench i Agustín González

Escola Universitària Salesiana de Sarrià (EUSS)  
Passeig Sant Joan Bosco, 74  
08017 Barcelona  
afontquerni@euss.es

Maig de 2005

## Resum

L'àmbit docent dels sistemes basats en microprocessadors presenta certes mancances per ser duta a terme dins dels paràmetres del programari lliure. Existeixen opcions força consolidades per alguns microprocessadors com els PIC's. Nosaltres treballem amb la família 68HC08 de Motorola i en aquest sentit hem llançat el projecte E2HC08 amb l'objectiu d'aconseguir un entorn de desenvolupament visual per al microcontrolador Motorola 68HC08 totalment basat en programari lliure per ser utilitzat sobre un sistema base GNU/Linux.

## 1 Introducció

La renovació tecnològica de la maqueta de pràctiques del laboratori d'Informàtica Industrial basada en el històrica família de microprocessadors 68HC11 de Motorola [7, 3] de 8 bits, va comportar la selecció d'una nova base tecnològica per les pràctiques sobre microcontroladors. El fabricant proposava una evolució cap a la família 68HC12, amb la qual comparteixen moltes característiques però per finalitats docents suposava entrar en la parcel·la de microprocessadors de 16 bits, la qual ja tenim coberta amb altres arquitectures. No volíem perdre la simplicitat de l'entorn de 8 bits com a base d'una introducció al món dels microcontroladors. Finalment l'elecció fou la família de microcontroladors 68HC08 de 8 bits de Motorola que és la nova revisió de antiga família 68HC05.

Amb la nova família de microcontroladors, el fabricant proporciona un entorn de programari per desenvolupament de programes escrits en codi ensamblador, que pot obtenir-se gratuïtament i que funciona sobre plataforma propietària Windows. En el cas de voler desenvolupar en llenguatge C sense

restriccions, cal llavors recorre a plataformes propietàries que ja no estan disponibles gratuïtament.

Actualment, entre d'altres projectes lliures, existeix un port per al conegut compilador GCC per als microprocessadors 68HC11 i 68HC12 [4], integrat en la mateix línia principal de desenvolupament d'aquest compilador del projecte GNU.

Aquest fet, junt amb què la institució pretén migrar tot el programari a platafora GNU/Linux, ens va motivar al Departament d'Electrònica a iniciar el projecte E2HC08 per aconseguir una plataforma basada exclusivament amb programari lliure per al desenvolupament i depuració de les aplicacions per aquesta família de microcontroladors.

## 2 Antecedents

Les pràctiques que es realitzaven i es realitzen en el laboratori d'informàtica Industrial de la EUSS [2], són una introducció en el món dels microcontroladors industrials amb dos grans blocs, microcontroladors de 8 bits i microcontroladors de 32 bits.

En el primer gran bloc de 8 bits, que s'utilitzava la família de microcontroladors 68HC11, es disposava d'una plataforma de maquinari amb tres plaques de circuit imprès de doble capa interconnectades amb cables i connectors amb diversos perifèrics de diferent dificultat per a la realització de les pràctiques.

Existia un entorn de programari de desenvolupament visual programat en Visual Basic 6.0 per a plataforma Windows. Era un entorn propi que integrava el compilador `as9` [5] per DOS, amb un editor visual de text per al programa d'usuari, l'interpretació de la sortida d'errors d'aquest compilador i un entorn de depuració visual propi aprofitant la comunicació sèrie amb programa monitor BUFFALO gravat permanentment en el xip de memòria FLASH de la maqueta de pràctiques.

## 3 Plataforma actual de desenvolupament sobre programari propietari

La nova plataforma de maquinari que es disposa actualment, basada en el microcontrolador 68HC08GP32 de la família de microcontroladors 68HC08, ha permès una gran reducció de tamany amb la integració dels xips externs de memòria RAM i FLASH, ... de l'anterior versió de plataforma, dins el integrat 68HC08GP32 de muntatge superficial amb encapsulat PLCC. Aquesta reducció ha permès passar de 3 plaques de circuit imprès a sols una placa, eliminant tot el cablejat i tota la problemàtica existent associada a mal contactes, junt amb una gran disminució del cost de muntatge manual de l'equip.

Actualment s'utilitza el programari utilitzat per al desenvolupament i depuració es un programari propietari [8, 6] proporcionat gratuïtament per un partner del fabricant sobre sistema operatiu propietari.

## 4 Objectiu: Cap a una plataforma sobre programari lliure

L'objectiu que ens marquem es la obtenció, realització, integració,... d'una plataforma de desenvolupament totalment lliure per a la família de microcontroladors 68HC08.

El plantejament d'aquesta fita, ens porta a planificar i a prioritzar els recursos disponibles, quines són les passes a seguir.

Al començament s'inicià les proves per a conèixer el disseny tecnològic del maquinari que permet el control necessari per la depuració en el mateix circuit imprès del microcontrolador i que porta incorporat aquesta funcionalitat el mateix dispositiu. En aquestes proves s'utilitza el programari proporcionat pel fabricant.

El següent pas, ja sobre una plataforma lliure GNU/Linux, fou aconseguir el control del microcontrolador en mode depuració mitjançant la comunicació sèrie estàndard. Cal un desenvolupament del protocol utilitzat i les formes d'operar d'aquest mode.

Un cop fou establert l'enllaç, es desenvolupa per accedir a totes les funcions bàsiques que proporciona el mateix maquinari i que són molt elementals. A partir d'aquestes funcions bàsiques s'aconsegueix totes les operacions més complexes i comunes per a qualsevol entorn de depuració: inspecció de registres, inspecció de memòria, inspecció de codis, programació, borrat, punts de parada, control d'execució pas a pas,... així com la gestió i carrega d'arxius. Aquesta etapa portà a la obtenció d'una biblioteca dinàmica que proporciona les aquestes funcions de depuració, i a un exemple d'un complert d'un depurador en mode text basat en la biblioteca ncurses.

Actualment s'està finalitzant la integració de l'entorn de desenvolupament visual totalment basat en programari lliure amb un editor i un compilador. I posteriorment la integració d'aquest entorn amb la biblioteca de depuració de l'anterior fase.

I el futur proper ens planteja la integració amb altres projectes lliures de desenvolupament per a microcontroladors i el suport per a desenvolupament de programes d'usuari en llenguatge C de programació. Junt amb la realització d'un circuit de maquinari obert (open hardware) per a la introducció i el fàcil accés al desenvolupament de solucions basades en aquesta opció tecnològica totalment lliure.

El projecte E2HC08 està disponibles al portal sourceforge.net per al desenvolupament de programari lliure [1].

## 5 Característiques generals de la família de microcontroladors 68hc08

Es una família de microcontroladors de 8 bits de Motorola( desdel juliol del 2004 Freescale Semiconductors ) basada en un unitat de control amb un registre d'instrucció comú, la CPU08, amb un únic mapa d'adreces per memòria i perifèrics de 16 bits (fins a 64K). Alhora aquesta unitat de control es una revisió de la ja antiga família microcontrolador 68HC05, amb la qual el codi es compatible.

Les característiques d'aquesta família són la integració de gran part dels perifèrics externs en l'anterior família gràcies a la alta integració tecnològica que s'aconsegueix actualment. També es facilita molt la seva utilització en plaques, ja que sols es necessari per funcionar l'alimentació d'aquest integrat del microcontrolador i el seu circuit de reset associat. Les diferents versions que es disposen varien la capacitat de memòria RAM, memòria FLASH, ports entrada/sortida, temporitzadors, perifèrics de comunicació sèrie, conversor analògics digitals, ...

Aquesta família, té integrat ja en el mateix microcontrolador uns perifèrics i un reduït programa monitor en memòria ROM que permeten la existència d'una comunicació sèrie mitjançant un sol pin de l'integrat, la seva programació i la seva depuració en el mateix circuit imprès final del producte ensamblat. Sols s'ha de tenir certes limitacions en l'ús, i reservar molt pocs recursos per a fer possible aquestes funcionalitats. Aquests recursos afecten a la reserva d'un pin, les tensions inicials de certs pins del encapsulat, l'ús d'un segon banc de vectors d'interrupcions, la introducció d'un codi de seguretat, la programació del perifèric de break i l'enviament de les ordres via sèrie que notifiquen les operacions que ha de realitzar el reduït programa monitor incorporat.

## 6 Depuració en l'entorn E2HC08

### 6.1 Instruccions elementals del programa monitor incrustat

Per poder realitzar la depuració cal uns requisits de maquinari que proporcionen aquesta funcionalitat i que a continuació s'exposen:

- Read (4Ah): La instrucció read permet llegir una posició de memòria.
- Write (49h): La instrucció write escriu una dada en una posició de memòria.
- Indexed read (1Ah): Instrucció de lectura indexada. Mètode de lectura és molt més ràpid que l'anterior (Read)
- Indexed write (19h): La instrucció de escriptura indexada. Mètode de escriptura és molt més ràpid que l'anterior(Write)

- **Read SP (0Ch):** Aquesta instrucció s'utilitza per llegir l'adreça on apunta el stack pointer o punter a pila.
- **Run (28h):** Instrucció que executa el codi gravat a la memòria flash del dispositiu o programa d'usuari.

## 6.2 Interfície de la biblioteca de depuració del projecte E2HC08

- **llegir\_memoria():** llegeix 80 posicions del mapa de memòria del MCU. Es necessari passar una adreça a la funció i un array amb 80 posicions i aquesta l'emplena amb el valor de les dades contingudes a partir de l'adreça demanada, les dues primeres dades es llegiran amb un read, posteriorment es fan successius reads indexats per llegir la resta de dades per accelerar la lectura.
- **llegir\_ports():** es una variació de `llegir_memoria()` però sols es llegeixen el valor dels ports del MCU i que es situen a les posicions: 0000h, 0001h, 0002h, 0003h i 0008h del mapa de memòria, que corresponen respectivament als ports A, B, C, D i E.
- **llegir\_acumuladors():** llegeix la posició del punter a pila (sp: stack pointer), degut a que ens trobem en una interrupció quan s'executa el programa monitor, sabem que els registres del MCU es troben entre les posicions sp+1 i sp+6 de la memòria de pila. El registre H es troba a sp+1, el condition code register (CCR) a sp+2 i així successivament, els següents son l'acumulador A, el registre X, el byte de major pes del contador de programa (pc: program counter) i finalment el de menor pes a la posició sp+6.
- **registre\_break():** aquesta funció programa el perifèric "break", específic per el control del bus intern d'adreçament per provocar la interrupció del programa i per tant la funcionalitat d'un únic de punt de condició de parada al accedir el aquest bus d'adreçament a certa direcció de memòria.
- **esborrar\_pagina():** esborrar una pagina de la memòria flash.
- **esborrar\_flash():** esborrar la memòria flash sencera.
- **programar\_flash():** permet realitzar la programació de la memòria flash.
- **executar\_instruccio()** executa la instrucció següent de on apunta el PC (program counter) i ens permet anar executant les instruccions de una a una, o sigui pas a pas. Això es realitza ficant un punt de parada a pc+1 mitjançant la funció

- `registre_break()` després s'envia un run al MCU, al acabar cal restaurar el punt de parada amb el seu valor anterior, executar, executa el codi de memòria fins que troba un punt de parada. Si no s'ha introduït cap punt de parada es perd el control sobre l'execució del codi ja que el programa entra en una execució cíclica, de manera que cal realitzar un reset de l'alimentació per recuperar-lo i tornar a entrar en mode monitor.
- `llegir_fitxer_s19()`: llegeix el fitxer en format .S19 i els grava en una estructura en memòria pròpia.
- `llegir_codi_s19()`: busca dins el fitxer .S19 el codi de seguretat d'aquell programa per proporcionar accés a lectura de la memòria flash del microcontrolador.
- `carregar_fitxer_s19()`: grava a la memòria flash el programa ensamblador.
- `adresa_de_partida()`: busca la posició en memòria on comença el codi del programa per tal de posicionar el PC en aquest punt abans de realitzar la primera execució.
- `desensamblar()`: llegeix de la estructura interna pròpia de la biblioteca el codi de programa i escriu en pantalla l'adreça on es troba cada instrucció, seguit de la instrucció en hexadecimal i els seus operands en gris, i finalment el mnemònic en ensamblador de la instrucció i els seus operands.

### 6.3 Exemple ús de la biblioteca i executable operatiu: Depurador en mode text

Actualment disposem com a resultat del projecte d'un depurador totalment operatiu. Aquest té interfície d'usuari bastant amigable amb mode text però en finestres, utilitzant la biblioteca ncurses que permet una bona presentació i utilització (Veure Figura 1).

## 7 Conclusions

S'ha iniciat amb èxit el projecte E2HC08, amb una primera fase on s'ha obtingut un depurador en mode text com exemple d'ús de la biblioteca dinàmica de depuració per a microcontroladors 68HC08, la qual serà una de les bases que sustentarà l'entorn visual integrat de desenvolupament basat en programari lliure en el qual es vol arribar.

La dificultat en la programació quan aquesta es realitza en un frontera molt propera en el maquinari on es més difícil establir si problemes i er-

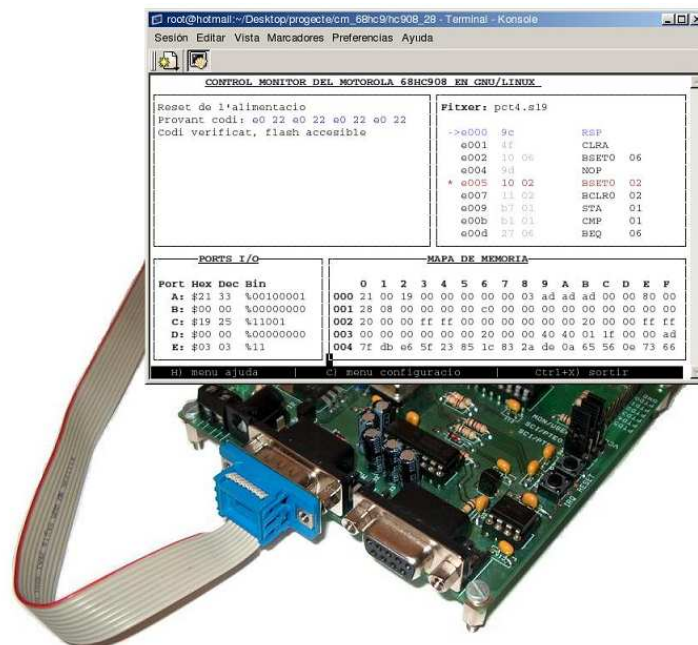


Figura 1: Exemple d'utilització de la biblioteca.

rors en el desenvolupament són degut a un incorrecte configuració d'aquest maquinari o una incorrecte programació del programari.

Falta molta feina per endavant per aconseguir l'objectiu final, tot i que l'accés i l'ús de codi lliure existent facilita molt el coneixement i el desenvolupament de les funcionalitats que es requereixen.

La documentació adequada sempre es un tema deficient: ja sigui per falta de fiabilitat o contrastació de la informació, ja sigui per obsoleta. La internacionalització de la documentació i del codi per un accés global.

L'entorn d'enginyeria on s'ha desenvolupat fins ara ha provocat mancances a l'estructuració i la integració en altres projectes més complexes d'estructures informàtiques més avançades.

S'obren unes interessants vies de futur per al desenvolupament de les part que falten per un entorn complert, i el coneixement tecnològic per a l'integració amb programari lliure existent i el qual la plataforma obtinguda per aquest projecte proporcioni per un ús funcional i complert.

## Referències

- [1] Debugger/ide for 68hc08 microcontroler. <http://sourceforge.net/projects/e2hc08/>.
- [2] Escola universitaria salesiana de sarria. <http://www.euss.es>.

- [3] Freescale semiconductos. <http://www.freescale.com>.
- [4] Gnu development chain for 68hc11 and 68hc12. <http://www.gnu.org/software/m68hc11/>.
- [5] The macroassembler as. <http://john.ccac.rwth-aachen.de:8000/as/>.
- [6] Metrowerks - codewarrior technology. <http://www.metrowerks.com>.
- [7] Motorola company. <http://www.motorola.com/mcu/>.
- [8] P&E microcomputer systems. <http://www.pemicro.com>.