

COMENTARIO TÉCNICO

Buceando en el HC908.....



Por Ing. Daniel Di Lella
Dedicated Distributor Field Application
www.edudevices.com.ar
dilella@arnet.com.ar



www.edudevices.com.ar

Fe de Erratas

Hola amigos, en los artículos anteriores habíamos visto como sacarle el mayor provecho a la familia **HC908APxx**, en los mismos se deslizó un error en la rutina de borrado de página y en la de borrado en masa. En las mismas se cometió el error en un lazo de demora que deja muy “marginal” la efectividad del borrado en ambas rutinas. Además aprovechando la oportunidad, se agregó en la rutina de MASS ERASE (borrado en masa) un mecanismo de verificación de borrado en la zona de vectores (Reset Vector, en particular) que será vital cuando se utiliza la rutina de borrado en masa para volver al MCU al estado de “virginidad” cuando se quiere efectuar la re – programación total del mismo a pocos hilos, o sea en modo “monitor forzado” sin pin IRQ a + 9V. A continuación, incluiremos las rutinas corregidas.

```
base 10T                ;Base Decimal por Default
include 'ap32_registers.inc' ;Equates grales. p/ AP32
```

```
*****
* EQUATES *
*****
```

```
RAMSPACE      EQU $0060                ;Czo de la RAM uso GRAB/BORR. FLASH
DATSTRC       EQU RAMSPACE+8          ;RAM para uso del "MONITOR ROM"
Q_RAM_Blk_Erase EQU $0080              ;Czo RAM para programa LIMPIEZA GRAL. FLASH
RAMUSER       EQU $00C0                ;Comienzo de la RAM p/ usuario
RAMEND        EQU $0860                ;Fin de la RAM + 1 en el AP32 / AP64
ROMSTART      EQU $7860                ;Czo de la FLASH para PROGRAMA
PRGRNGE       EQU $FC34                ;Sub-Rutina en ROM escritura datos en FLASH
```

* EQUATES P/ BORRADO DE FLASH *

b_ERASE EQU 2 ;Bits control bomba de carga
b_HVEN EQU 8
b_MASS EQU 6
MORVALUE EQU %11111111 ;Valor standard del MOR

BLOCKSIZE EQU \$200
TESTCOUNT EQU \$20
FLASHSTART EQU \$860
RAM_BEGIN EQU \$60
RAM_END EQU \$85F

* VARIABLES EN RAM *

ORG RAMSPACE ; Comienzo RAM para AP32
; reservada hasta DATSTRC P/ uso
; rutinas ROM prog/erase FLASH.

ORG DATSTRC ; RAM p/ uso en prog/erase flash

CPUSPD RMB 1 ; CPU speed -> 4 x Fop aprox. (FOSC)
DATASIZE RMB 1 ; N° de Bytes de datos a grabar.
STARTADDR RMB 2 ; Start Address p/ R o W un rango
DATA RMB 1 ; DATOS en RAM transitorios

V_FLASH_ADDR RMB 2 ; Address czo. página a borrar
; cargar con \$FFFF p/ MASS ERASE

;

* MASS_ERASE - Rutina de borrado GRAL. de la FLASH en los HC908APxx *
* Toda la rutina está en FLASH y luego se copia a RAM p/ ejecutarla *
* desde allí. (FBUS = 2,4576 MHZ) *
* *
* VARIABLES INVOLUCRADAS: *
* *
* V_FLASH_ADDR -> Dirección de czo. página a borrar (512 bytes) *
* En este caso se debe cargar con \$FFFF *
* *
* ATENCION: Reservar 60 BYTES en la RAM a partir de "Q_RAM_Blк_Erase" *

MASS_ERASE

```
lda #$FF
sta FLBPR          ; Without flash protection !!
```

*--- erase block -----

erase1_M:

```
pshh              ; "H:X" content will be corrupted in the jump
pshx              ;
```

```
jsr FLASH_ERASE_M
```

```
pulx              ;
pulh              ;
```

```
ICPCLEAN_M  NOP          ; LOOP infinito hasta P.O.R del MCU
             BRA ICPCLEAN_M      ;
             RTS              ;
```

* Block Erase *

FLASH_ERASE_M:

```
bsr BlkErase2RAM_M ; copy block erase routine to RAM
ldhx V_FLASH_ADDR  ; load H:X with the block address
lda #b_MASS         ; MUST load Acc with b_MASS (ERASE+MASS=1)
jsr Q_RAM_Blk_Erase ; execute block erase in RAM
rts
```

*-----

BlkErase2RAM_M:

```
ldhx #Blk_Erase_Len_M ; get blk erase routine length
                          ; NB: Assume "Blk_Erase_Len" is one byte long
```

BE2RAM1_M:

```
lda (Block_Erase-1),x ; load from FLASH
sta {Q_RAM_Blk_Erase-1},x ; copy to RAM
dbnzx BE2RAM1_M       ; NB: Assume "Blk_Erase_Len" is one byte long
                          ; need modification if length over 1 byte
rts
```

Block_Erase_M:

```
sta FLCR          ; set ERASE bit
sta ,x            ; write any data to block
bsr Dly_5us_M
lda FLCR
ora #b_HVEN
sta FLCR          ; set HVEN bit
```

ldx #255 ;(2)

Blk_Erase_Time_M:

```
bsr Dly_1ms_M ;[14]
dbnzx Blk_Erase_Time_M ;(3)
ldhx #FLCR
lda #%00001000 ; clear ERASE bit
sta ,x
bsr Dly_100us_M
clr ,x ; clear HVEN bit
NOP ;
NOP ;
NOP ;
LDA $FFFE ;Reset Vector is Blank??
CMP #$FF ;not ? -> new mass erase cycle
BNE Block_Erase_M ;yes ? -> cycle over
LDA $FFFF ;
CMP #$FF ;
BNE Block_Erase_M ;
rts
```

Blk_Erase_Exit_M:

* ----- *

```
Dly_5us_M: ;For 2,5 MHz bus
; [4] cycles
lda #4 ; [2]
dbnza $ ; [3]
rts ; [4]
```

```
Dly_100us_M: ; [4] cycles
lda #100 ; [2]
dbnza $ ; [3]
rts ; [4]
```

```
Dly_1ms_M: ; [4] cycles
lda #200 ; [2]
Dly_1ms_lp mov PORTB,PORTB ; dummy for 10 bus clk
mov PORTB,PORTB ;
dbnza Dly_1ms_lp ; [3]
rts ; [4]
```

Dly_5us_Exit_M:

Blk_Erase_Len_M equ {Dly_5us_Exit_M - Block_Erase_M}

```
-----
;
```

```

;-----
;-----
*****
* PAGE_ERASE - Rutina de borrado de Página en los HC908APxx *
* Toda la rutina está en FLASH y luego se copia a RAM p/ ejecutarla *
* desde allí. (FBUS = 2,4576 MHZ) *
* *
* VARIABLES INVOLUCRADAS: *
* *
* V_FLASH_ADDR -> Dirección de czo. página a borrar (512 bytes) *
* *
* ATENCION: Reservar 60 BYTES en la RAM a partir de "Q_RAM_Blк_Erase" *
*****

```

PAGE_ERASE

```

        lda #$FE
        sta FLBPR          ; protect vector table only

```

*--- erase block -----

erase1:

```

        pshh              ; "H:X" content will be corrupted in the jump
        pshx              ;
        jsr FLASH_ERASE
        pulx              ;
        pulh              ;
        RTS              ;

```

```

*****
* Block Erase *
*****

```

FLASH_ERASE:

```

        bsr BlkErase2RAM ; copy block erase routine to RAM
        ldhx V_FLASH_ADDR ; load H:X with the block address
        lda #b_ERASE      ; MUST load Acc with b_ERASE
        jsr Q_RAM_Blк_Erase ; execute block erase in RAM
        rts

```

*-----

BlkErase2RAM:

```

        ldhx #Blk_Erase_Len ; get blk erase routine length
                                ; NB: Assume "Blk_Erase_Len" is one byte long

```

BE2RAM1:

```

        lda (Block_Erase-1),x ; load from FLASH
        sta {Q_RAM_Blк_Erase-1},x ; copy to RAM
        dbnzx BE2RAM1         ; NB: Assume "Blk_Erase_Len" is one byte long
                                ; need modification if length over 1 byte
        rts

```

```

Block_Erase:
    sta FLCR           ; set ERASE bit
    sta ,x            ; write any data to block
    bsr Dly_5us
    lda FLCR
    ora #b_HVEN
    sta FLCR         ; set HVEN bit

```

* ----- *

```

Blk_Erase_Time:
    ldx #20           ;(2)
    bsr Dly_1ms      ;[14]
    dbnzx Blk_Erase_Time ;(3)
    ldhx #FLCR
    lda #%00001000   ; clear ERASE bit
    sta ,x
    bsr Dly_5us
    clr ,x           ; clear HVEN bit
    rts

```

Blk_Erase_Exit:

* ----- *

```

Dly_5us:
    ;For 2,5 MHz bus
    ; [4] cycles
    lda #4           ; [2]
    dbnza $          ; [3]
    rts              ; [4]

Dly_1ms:
    ; [4] cycles
    lda #200        ; [2]
Dly_1ms_lpb
    mov PORTB,PORTB ; dummy for 15 bus clk
    mov PORTB,PORTB ;
    dbnza Dly_1ms_lpb ; [3]
    rts              ; [4]

```

Dly_5us_Exit:

Blk_Erase_Len equ {Dly_5us_Exit - Block_Erase}

```

;-----
;-----

```

Bueno amigos, en el próximo número veremos como hacer un control remoto por infrarrojo utilizando la técnica de PWM (modulación por ancho de pulso) con el modulo de Timer, en función ICAP (Input Capture) disponible en todos los HC908, hasta la próxima!!!

NOTA: Este artículo fue realizado cuando las primeras máscaras de los HC908APxx aparecían en el mercado. En la actualidad ya no se presentan ninguno de los errores aquí citados, pero el documento es válido si se lo utiliza como idea para solucionar problemas o situaciones similares en algún otro dispositivo de la familia HC908.

www.edudevices.com.ar

