

```

*-----*
* Program Number: Display 68K RAM/ROM Map
* Written by      : John Monahan
* Date Created   : 11/10/2011
* Description    : Boot from high ROM
*

```

```

BELL      EQU      $07
BLANK     EQU      $20
CR        EQU      $0D
LF        EQU      $0A

```

```

ORG      $00FD0000

```

```

DC.L     $00FD8100 ;SSP = FD8100H
DC.L     $00FD0020 ;PC = FD1010H

```

```


```

```

ORG      $00FD0020 ;Will start monitor here

```

```

MOVE.L   #$00FF0000,A0 ;Point to status Port 0H (CRT, Propeller Console IO Board)
MOVE.L   #$00FF0001,A1 ;Point to hardware Port 01H (CRT, Propeller Console IO Board)
MOVE.B   #6,D0          ;For Trap #15 D0 = #6 (always), to display a character
                        ;(Character will always be in D1).

```

```

MOVE.B   #'#',D1       ;Show we are alive
BSR      CRTOUT

```

```

MOVE.L   #$00000000,A3 ;Pointer to RAM area

```

```

NEWLINE: MOVE.L   #64,D3          ;64 characters across per line

```

```

MOVE.B   #CR,D1        ;SEND cr/lf
BSR      CRTOUT
MOVE.B   #LF,D1
BSR      CRTOUT

```

```

MOVE.L   A3,D1         ;Move address @A3 to D1 and print it
SWAP     D1            ;Swap down upper word
LSR.W   #4,D1         ;Shift upper byte to lower 4 bits
ANDI.B   #$0F,D1      ;SHOW NIBBLE
ORI.B   #$30,D1       ;CONVERT TO ASCII
CMPI.B   #$39,D1      ;SEE IF IT IS>9
BLE.S   HEXOK
ADD.B   #7,D1         ;ADD TO MAKE 10=>A

```

```

HEXOK:   BSR      CRTOUT

```

```

MOVE.L   A3,D1
SWAP     D1            ;Swap down upper word again
ANDI.B   #$0F,D1      ;SHOW LOWER NIBBLE
ORI.B   #$30,D1       ;CONVERT TO ASCII
CMPI.B   #$39,D1      ;SEE IF IT IS>9
BLE.S   HEXOK1
ADD.B   #7,D1         ;ADD TO MAKE 10=>A

```

```

HEXOK1:  BSR      CRTOUT

```

```

MOVE.L   A3,D1         ;Move address @A3 to D1 again
LSR.W   #8,D1         ;Shift upper byte to lower 8 bits
LSR.W   #4,D1         ;Shift upper byte to lower 4 bits
ANDI.B   #$0F,D1      ;SAVE LOWER NIBBLE
ORI.B   #$30,D1       ;CONVERT TO ASCII
CMPI.B   #$39,D1      ;SEE IF IT IS>9
BLE.S   HEXOK2
ADD.B   #7,D1         ;ADD TO MAKE 10=>A

```

```

HEXOK2:  BSR      CRTOUT

```

```

;Address lower high byte nibble
MOVE.L   A3,D1         ;Move address @A3 to D1
LSR.W   #8,D1         ;Shift upper byte to lower 8 bits
ANDI.B   #$0F,D1      ;SAVE LOWER NIBBLE
ORI.B   #$30,D1       ;CONVERT TO ASCII
CMPI.B   #$39,D1      ;SEE IF IT IS>9

```

```

BLE.S   HEXOK3
ADD.B   #7,D1           ;ADD TO MAKE 10=>A
HEXOK3: BSR           CRTOUT

;Address upper high byte nibble
MOVE.L  A3,D1           ;Move address @A3 to D1
LSR.W   #4,D1           ;Shift upper byte to lower 4 bits
ANDI.B  #$0F,D1         ;SAVE LOWER NIBBLE
ORI.B   #$30,D1         ;CONVERT TO ASCII
CMPI.B  #$39,D1         ;SEE IF IT IS>9
BLE.S   HEXOK4
ADD.B   #7,D1           ;ADD TO MAKE 10=>A
HEXOK4: BSR           CRTOUT

;Address lower high byte nibble
MOVE.L  A3,D1           ;Move address @A3 to D1
ANDI.B  #$0F,D1         ;SAVE LOWER NIBBLE
ORI.B   #$30,D1         ;CONVERT TO ASCII
CMPI.B  #$39,D1         ;SEE IF IT IS>9
BLE.S   HEXOK5
ADD.B   #7,D1           ;ADD TO MAKE 10=>A
HEXOK5: BSR           CRTOUT

;Address upper lower byte nibble
MOVE.B  #BLANK,D1
BSR     CRTOUT

START1: MOVE.B  (A3),D1   ;Is there RAM/ROM there
        NOT.B  D1
        MOVE.B D1,(A3)   ;See if we can flip bits
        MOVE.B (A3),D6
        CMP.B  D1,D6
        BNE   NOT_RAM
        NOT.B  D1        ;Put back original data
        MOVE.B D1,(A3)
        MOVE.B #'R',D1
        BRA   DONE_TEST

NOT_RAM: MOVE.B  (A3),D1
        CMPI.B #$FF,D1
        BEQ   EMPTY

EMPTY:  MOVE.B  #'p',D1
        BRA   DONE_TEST
DONE_TEST: MOVE.B #'.',D1
        BSR   CRTOUT

ADD.L   #$2000,A3       ;No matter what point to next 2K byte section

SUBI.L  #1,D3
TST.L   D3              ;Have we done 32X2 characters across
BEQ     NEWLINE

MOVE.L  A3,D4
CMPI.L  #$ff0000,D4     ;Have we done all the RAM area
BLE     START1
DONE:   BRA     DONE     ;Loop continously

;Send ASCII character in D1
CRTOUT: MOVE.B  (A0),D5   ;Check CRT status is ready to recieve character
        ANDI.B #$04,D5
        TST.B  D5
        BEQ   CRTOUT
        MOVE.B D1,(A1)   ;Output ASCII (in D1) to hardware port 01H
;
TRAP   #15
RTS    ;Return from subroutine

END    $00FD0000

```