```
;This program displays the contents of the bootstrap ROM in a Motorola MC68705P3 or P5 microcontroller. It can be modified to do the same with
;the MC68705U3, U5, R3 and R5. I haven't tried but it should be possible to modify it for other Motorola microcontrollers

;To setup the hardware you can place the microcontroller on a breadboard and connect XTAL pin to the EXTAL pin (to use RC mode)
;Connect nine LEDs as follows – connect all the LED anodes to +5V
;Connect each LED cathode to its own 1K resistor and connect the other side of 8 resistors to Port A; connect the 9th resistor to bit 7 of Port B
;The 8 Port A LEDS will display each byte of data, the 9th LED acts as a 'data ready' signal

;The idea is each time the 'data ready' LED flashes you need to quickly write down the value being displayed on the 8 data LEDs
;If you miss any values don't worry, just keep going, you can re-run the program and fill in the missing values later. The circuit can be improved by
;using BCD to 7-segment decoders and 7-segment displays but since it normally is only used once it's hardly worth the trouble
;An alternative is to remove all the delay loops in the program and feed the Port A and 'data ready signals into an I/O card connected to a computer
;or other microcontroller and have it log the data
;originally written by Peter Ihnat in May 2012
;properly documented in Feb 2016; also added references to MC68705S3 but I haven't tested it on an S3 micro

;equates
PortA   equ     0000H
PortB   equ     0001H
;data direction registers
DDRA    equ     0004H
DDRB    equ     0005H

;variables used by delay subroutine
        org     0010H                   ;note: for MC68705S3 use 0018H instead of 0010H
cntr    rmb     1
cntr1   rmb     1


;*********************
;main program starts here
;(at start of EPROM)
;*********************
```

```
        org     0080H
Start:  rsp

;setup I/O ports - Port A displays a byte of data on 8 LEDs
;a 'data ready' LED connected to bit 7 of Port B is flashed when correct data is displayed on Port A
;note: outputting a 0 turns on an LED, 1 turns it off
;first setup Ports A and B as outputs and set all bits high (LEDs off)
        lda     #0FFH
        sta     PortA
        sta     DDRA
        sta     PortB
        sta     DDRB
        clrx                    ;reg X will be used as an offset to access each value in the bootstrap area
        clr     cntr            ;used in delay subroutine

;fetch a byte from the bootstrap area, invert it and display on 8 LEDs
loop:   lda     0785H,X         ;note that address 0785H is the start of the bootstrap ROM for the MC68705P3 and P5
                                ;for the MC68705U3, R3, U5 and R5 use 0F80H
                                ;for the MC68705S3 use 0F20H
        coma
        sta     PortA

;flash 'data ready' LED
        bclr    7,PortB         ;turn LED on
        bsr     delay
        bset    7,PortB         ;turn LED off
        bsr     delay

;increment pointer to the next byte
        incx
        cpx     #73H            ;check if end of bootstrap code has been reached
                                ;note that 73H (115 bytes) is the length of the bootstrap ROM for the MC68705P3 and P5
                                ;for the MC68705U3, R3, U5 and R5 use 120 bytes or 78H
```

```
                                        ;for the MC68705S3 use 216 bytes or 0D8H
        bne     loop
        bra     *                       ;all done so go into infinite loop


;************
;SUBROUTINES
;************

;this is a very simple delay routine. Increase the value   #30H   if you want the data to be displayed on the LEDs for a longer time
delay:  lda     #30H
        sta     cntr1
del:    bsr     wait
        dec     cntr1
        bne     del
        rts

wait:   bsr     deccntr
        bsr     deccntr
        bsr     deccntr
        bsr     deccntr
        bsr     deccntr
        bsr     deccntr
        bsr     deccntr
        bsr     deccntr
        bsr     deccntr
        bsr     deccntr
        bsr     deccntr
        bsr     deccntr
        bsr     deccntr
deccntr:
        dec     cntr
        bne     deccntr
        rts
```

```
;**********
;vectors, etc
;**********

;interrupts (not used here)
Timer:  rti
Ext:    rti
SW:     rti

;set MOR so micro is in RC mode
        org     0784H           ;use address 0784H for the MC68705P3 and P5
                                ;use address 0F38H for MC68705U3, U5, R3 and R5
                                ;use address 0F1EH for MC68705S3
        fdb     80H
;setup all the vectors for correct operation
        org     07F8H           note that address 07F8H is the location of vectors for the MC68705P3 and P5
                                ;for the MC68705U3, R3, U5 and R5 use 0FF8H
                                ;for the MC68705S3 use 0FF8H

        fdb     Timer           ;vector to Timer interrupt routine
        fdb     Ext             ;vector to External interrupt routine
        fdb     SW              ;vector to SW interrupt routine
        fdb     Start           ;vector to start of main program
```