Order this document
by AN-HK-33/H
Rev. 1

Motorola Semiconductor Application Note

AN-HK-33

In-Circuit Programming of FLASH Memory in the MC68HC908JL3

By Roger Fan
Applications Engineering
Microcontroller Division
Hong Kong

This application note describes In-Circuit Programming (ICP) of the FLASH memory in the Motorola MC68HC908JL3 (JL3) microcontroller and its variants: MC68HRC908JL3, MC68HC908JK3, MC68HRC908JK3, MC68HC908JK1, and MC68HRC908JK1.

The text is divided into two parts:

- PART 1 — covers a general overview of ICP and techniques that can be applied to the JL3
- PART 2 — covers a low-cost ICP implementation for the JL3

For detailed specification on MC68HC908JL3, please refer to the datasheet: Motorola order number MC68HC908JL3/H.

PART 1
Introduction

In-circuit programming is a process by which the device is programmed or erased with the device on the final circuit board — the *target system*. This allows the *user code* to be changed without having to remove the device off the target system for reprogramming or initial programming.

On JL3, the 4k-bytes FLASH memory is allocated for the user code, with an additional 48-bytes of FLASH for user defined reset and interrupt vectors. A high voltage supply is not required by the JL3 for program or erase operations; as it is generated by an internal charge-pump.

This FLASH memory can be programmed or erased using software routines running either in *User mode* or *Monitor mode*, by writing to the FLASH Control register at address $FE08.

User Mode

In User mode, the JL3 is running the user code, that has been programmed in the FLASH memory. This is the mode in which the JL3 will be running during most of the time.

Monitor Mode

In Monitor mode, the JL3 is running code that has been permanently programmed into an area of memory in the JL3 during fabrication. The monitor code is used for communicating to an external host, connected via a serial link. Programming an initially blank JL3 FLASH memory is executed in monitor mode.

Initial FLASH Programming

The mode in which the JL3 enters is latched after a power-on-reset (POR), and depends on the logic level on the following pins: $\overline{\text{IRQ1}}$, $\overline{\text{RST}}$, PTB0, PTB1, PTB2, and PTB3. (For details, please refer to the Monitor ROM section in the datasheet.)

In-Circuit Programming in User Mode

ICP in user mode can be implemented so as to maintain target system operation while reprogramming the FLASH memory in the JL3. Reprogramming the FLASH memory in the JL3 involves two stages. The first stage is an erase operation to erase the existing data in the FLASH memory cell. The minimum erase size is 64-bytes, known as a *page*. The MASS bit in the FLASH Control register provides the option for erasing the entire FLASH array in one operation, known as *MASS erase*. It should be noted that an erased byte of FLASH memory reads as $FF. The second stage is the programming process, which programs the blank FLASH memory with new data. Thus, reprogramming involves: erase and program.

ICP code

Performing ICP in user mode requires that the erase and program routines — the *ICP code* — are to be stored in a part of non-volatile memory that can be called by the user program. This means the ICP code needs to be a routine that is part of the user code, and programmed into JL3's FLASH memory. With this in mind, ICP in user mode cannot be performed if the FLASH memory is initially blank; a blank device. Initial blank devices are programmed in Monitor mode (see next section for ICP in Monitor Mode).

With the ICP code programmed into the FLASH memory, it is called by software or hardware, and can operate in two ways:

The ICP code sets up the JL3 a communication link with an outside host system via the JL3 port pins, and then transfers control of the JL3 MCU to the host system. The host issues commands to erase the JL3's FLASH memory and downloads data to program the FLASH memory. In this case, the JL3 ICP code is acting as a command interpreter.

Alternatively, the ICP code can carry out the erase process and downloads new data from an external source for the programming. The source can be an intelligent host or an EPROM containing the new user code.

In both of the above methods, the ICP code must be loaded into the RAM area of memory, and the routine executed in the RAM area. Program or erase operations are not allowed while program is running in the FLASH area. If it was possible for the ICP code to execute in the FLASH area, there is the danger of erasing the ICP code itself.

Block Protected FLASH Memory

There is one situation where the FLASH memory cannot be erased: when it is *block protected*. The FLASH Block Protect register at address $FE09 is used to protect (prevent from erase or program) a block of, or the entire FLASH memory. By default, the entire JL3 FLASH memory is block protected, since the reset state of $FE09 is 00. The FLASH memory must be unprotected by setting the FLASH Block Protect register to $FF, prior to any program and erase operations.

In-Circuit Programming in Monitor Mode

In Monitor mode, the JL3 is running the *monitor code* that has been permanently programmed into an area of memory ($FC00 to $FDFF and $FF10 to $FFCF) in the JL3 during fabrication. First time programming of the JL3's FLASH memory can only be executed in monitor mode.

The monitor code consists of routines for communicating to a host connected using a serial link via pin PTB0. Once the link is established, control of the MCU is transferred to the host system. The host controls the MCU by directly writing to the MCU registers.
Monitor mode can be entered in two ways:

High Volt Entry to Monitor Mode

Similar to most Motorola MCUs, providing a high voltage ($1.5 \times V_{DD}$ for JL3) on the $\overline{IRQ1}$ pin during a POR will force the JL3 to enter monitor mode. With this high voltage entry method, the clock input to the MCU (at OSC1) must be either 4.9152 MHz or 9.8304 MHz. This clock divides to produce the 9600 baud communication speed on PTB0.

Blank Vector Entry to Monitor Mode

With the new FLASH memory implementation, there was a need to reduce the number of wire connections to the target system to program the MCU when ICP was required. The other method for entry to monitor mode is a blank reset vector. The only time when the reset vector is blank is when the entire JL3's FLASH memory is blank — the reset vector can only be erased by a mass erase operation. This monitor mode entry method does not need the high voltage to the $\overline{\text{IRQ1}}$ pin; and the clock at OSC1 must be 9.8304 MHz, to produce the 9600 baud communication speed on PTB0.

Implementing ICP in monitor mode has the advantage that no ICP code needs to be written for the user code. In addition, the *MCUscribe* program, a free Motorola utility, is available for the PC host system that talks to the MCU via PTB0 serial link.

Other ICP Considerations

Signal Conditioning

Normal system activities will usually be halted during an ICP operation, to allow an uninterrupted programming process. Therefore, at the start of the ICP process, the MCU should be configured such that no pin contention or runaway signal will occur during the ICP process. Also note that when the system is first switched-on with a MCU having a blank FLASH memory, the port pins default to their reset states.

Pin Isolation

If the MCU pins used for connecting to the external host are shared with the target system, make sure they are isolated to the proper logic level when the ICP connection is made.

PART 2
Introduction

The following ICP method is low-cost; with minimal system and user code changes. It involves two steps:

1. Erasing the FLASH memory in User mode.
2. Programming the FLASH memory in Monitor mode (blank vector entry) using Motorola's SPGMR08 Serial Programmer.

Bus Frequency Constraint

This ICP method uses a bus frequency at 2.4576 MHz for programming the FLASH (see Programming the FLASH Memory in Monitor Mode). For the blank vector entry method, this bus frequency can be generated using an external crystal oscillator circuit or a direct clock input at 9.8304 MHz (4 times the bus frequency). The 2.4576 MHz is used to derive the 9600 baudrate for the communication between MCU and Host.

Mass Erasing the FLASH Memory in User Mode

The program listing at the back of this application note contains the routine for mass erasing the MCU. Since this program is for demonstration purposes, only the MASS_ERASE subroutine is required for inclusion to the user program. Other parts of the program involves setting up the bus clock and polling the pins PTB0 and PTD3 for ICP request.

What the program does is this:

1. Check logic levels on PTB0 and PTD3; if true, proceed to mass erase.
2. Load MASS_ERASE routine to RAM memory.
3. Execute MASS_ERASE routine. The routine loops until the reset vector is blank.

On the JL3, an erase subroutine is available in the monitor ROM area. This subroutine is called after writing the two control bytes to the RAM locations $0088 and $0089.

In this implementation, PTB0 and PTD3 are used for setting up a request for mass erase operation. After a POR, when PTB0 = 1 and PTD3 = 0 (see figure 1), the user code will load the mass erase routine into RAM and perform a FLASH mass erase operation.

In the erase routine, the delay timing is based on a bus frequency of 2.4576 MHz, and the mass erase operation is repeated until the user vectors and the security bytes are erased. The time required for the mass erase operation is less than two seconds.

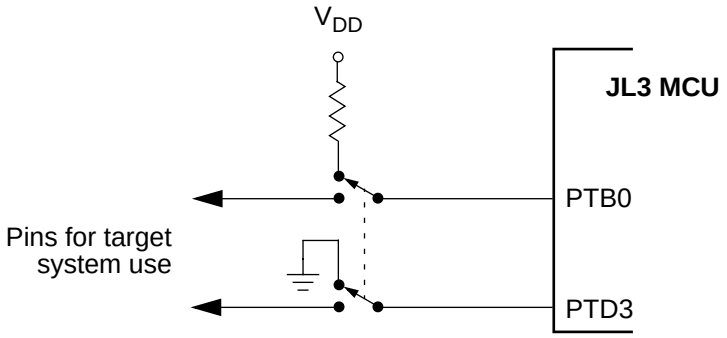


Figure 1. Mass Erase Port Pin Configuration

The flowchart in figure 2 shows the sequence of events for the mass erase operation.

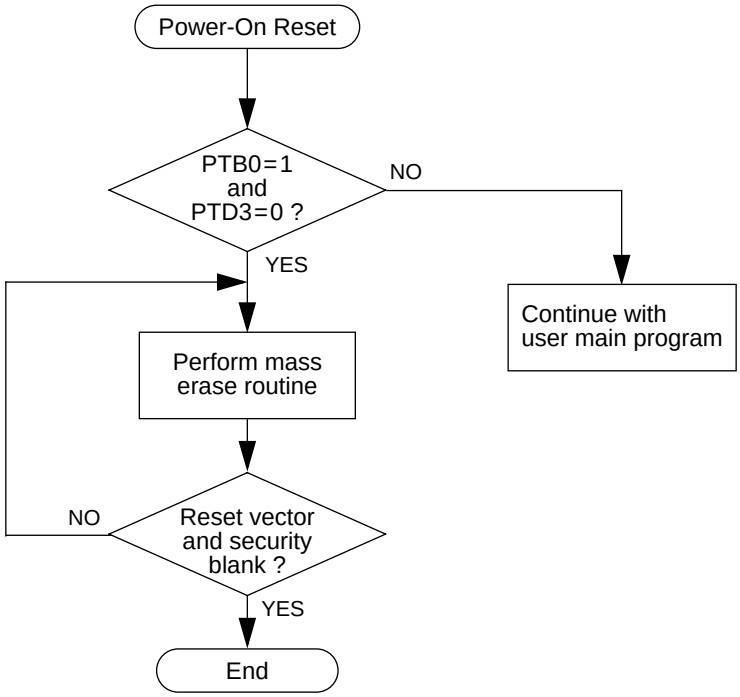


Figure 2. Mass Erase Flowchart

Procedure for mass erase

Using the sample program, this step-by-step procedure erases the JL3 FLASH in user mode:

1. Switch off the power to the target system.
2. Isolate port pins PTB0 and PTD3 from target system logic.
3. Set PTB0 to high via a pull-up resistor to V_{DD}.
4. Set PTD3 to ground directly to V_{SS}.
5. Switch on the power to the target system.
6. Wait 2 seconds.
7. Switch off power to the target system.
8. FLASH memory is now erased.

The next section describes the procedure for programming the JL3 FLASH memory using blank vector entry to monitor mode.

Programming the FLASH Memory in Monitor Mode

Programming the JL3's blank FLASH memory is achieved by running the MCU in monitor mode; and with a host connected using a serial link. Monitor mode can be entered in one of two ways after a power-on-reset:

- A high voltage ($1.5 \times V_{DD}$) applied on the $\overline{IRQ1}$ pin, or
- The FLASH memory is erased blank.

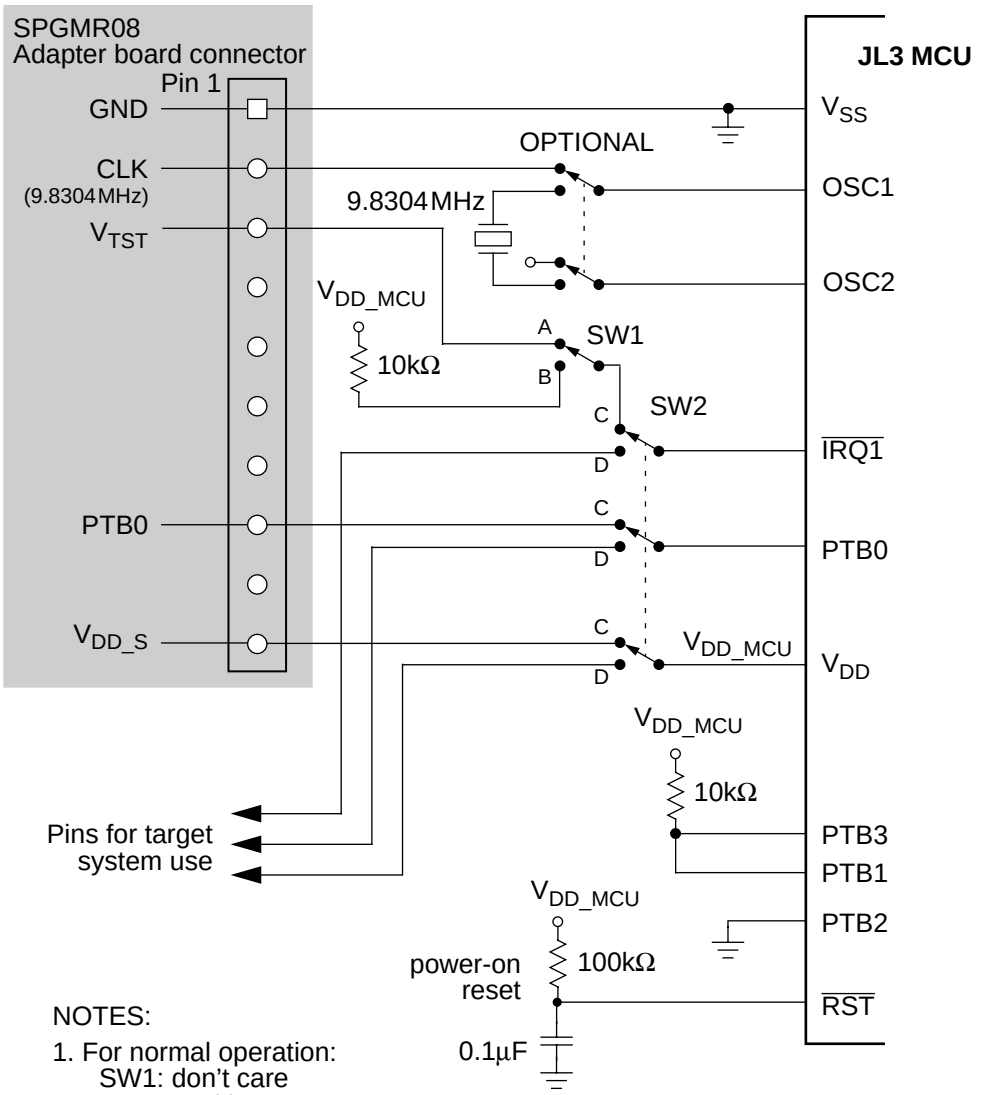
The latter method for entering monitor mode for programming the FLASH memory will be described here. With this method, the MCU enters monitor mode after a power-on reset when it detects that the reset vector, $FFFE–$FFFF, is blank (containing $FF).

The Motorola *SPGMR08 serial programmer* is used as the interface between the target system and the PC host system.

Figure 3 shows the connection to the SPGMR08. Three wires are used:

- PTB0 — This is the serial data link between the host and the MCU.
- V_{DD_S} — This line provides power and power-on reset synchronization between the host and MCU.
- GND — Common ground for the systems.

For this implementation, $\overline{IRQ1}$ is required to be pulled to V_{DD} for mode entry, and the clock frequency at OSC1 must be 9.8304 MHz (either from the crystal oscillator or a direct clock from SPGMR08). The V_{TST} connection is only necessary for a high voltage entry to monitor mode.



NOTES:

1. For normal operation:
   SW1: don't care
   SW2: position D
2. For **blank reset vector** monitor mode entry:
   SW1: position B
   SW2: position C
   PTB1, PTB2, and PTB3 pullup/down can be omitted.
3. For **high voltage** monitor mode entry:
   SW1: position A
   SW2: position C
   PTB1=1, PTB2=0, and PTB3=1 must be satisfied on mode entry.
4. If high voltage monitor mode entry is not required,
   SW1 can be omitted and signals default to position B.

Figure 3. Programming Setup

Once the programming system is connected as in figure 3, the programming is carried out by running the *MCUscribe* utility supplied with the SPGMR08. When MCUscribe has finished programming, set the jumpers back to their original position, and then select the "power-off" command on the MCUscribe utility screen menu.

The above ICP method has two limitations. They are:

1. The erase and program operations are for the entire 4k-bytes of FLASH memory — An erase operation erases all FLASH locations; a program operation programs all FLASH locations.
2. There must be no power outage during erase or program operations; otherwise, a high voltage must be applied to the $\overline{IRQ1}$ pin so that the MCU can enter Monitor mode. The alternative is to extract the MCU off the target system and reprogrammed using an external programmer.

Further cost-savings can be achieved by using the circuit in figure 4 to replace the SPGMR08 serial programmer.

Serial programming Schematic

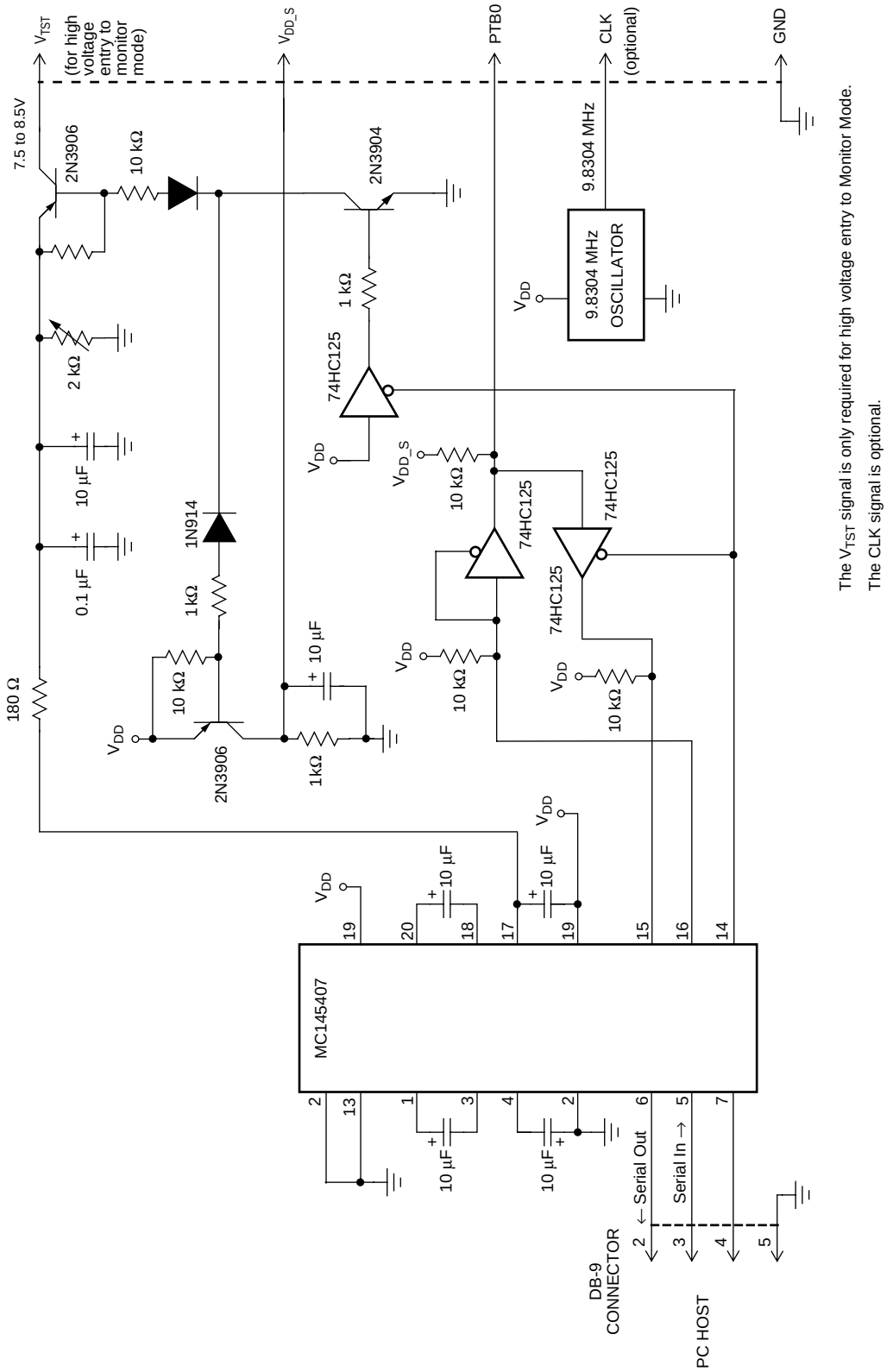


Figure 4. ICP Interface Circuit

Program Listing

```
;*****************************************************************
; Assembler Directives
; $base           10t
;*****************************************************************
; 68HC908JL3 User Mode FLASH Mass Erase
;
; Author    : Roger Fan
; File Name : jl3icp.asm
;*****************************************************************
; Description:
; This program allows the MCU to mass erase itself in user mode.
; The detect condition for mass erase is PTB0=1 & PTD3=0.
;
; For successful code execution, the user should set a bus frequency of
; 2.4576MHz. This can be derived from a 9.8304MHz xtal for the HC908 part.
;
; The program uses a subroutine, erase_cmd, located at $FC06 in the
; monitor ROM, for the mass erase operation.
;
; Jumper setting during power-up reset:
;            Jumper                  user mode             mass erase mode
;            -----                   ---------             ---------------
;            PTB0                       -                  pull-up(10k)
;            IRQ                     pull-up               pull-up
;            PTD3                    pull-up (10k)         short to ground
;*****************************************************************
; Version       Date                    Description
;
;   0.2      20/2/2000
;*****************************************************************
; MCU (JL3) I/O pin Assignment
;*****************************************************************
PTA          equ     0          ; Port A
PTB          equ     1          ; Port B
PTD          equ     3          ; Port D
DDRA         equ     4          ; Port A direction register
DDRB         equ     5          ; Port B direction register
DDRD         equ     7          ; Port D direction register
s_data       equ     0          ; Serial data used in monitor mode
Ps_data      equ     PTB        ; Port location of serial data
DDRs_data    equ     DDRB       ; Port direction location of serial data

;*****************************************************************
; FLASH Control Register
;*****************************************************************

FLCR         equ     $fe08      ; FLASH Control Register
HVEN         equ     3
MASS         equ     2
ERASE        equ     1
PGM          equ     0

FLBPR        equ     $fe09      ; FLASH Block Protect Register

;*****************************************************************
; External Subroutine Call Declaration
;*****************************************************************

erase_cmd    equ     $fc06      ; this routine is resident in the monitor rom,
                                ; and will erase an area unprotected when called

;*****************************************************************
; Constant declaration
;*****************************************************************

RAM_BEGIN     equ    $80     ; FLASH memory start address

;*****************************************************************
; RAM declaration       ; required by erase subroutine in monitor ROM
;*****************************************************************

ctrlbyt      equ     $88
cpuspd       equ     $89
CONFIG1      equ     $1F
CONFIG2      equ     $1E
RAM          equ     $90
MAIN         equ     $FB00
RSTVECTOR    equ     $FFFE
```

```
;--------------------------------------------------------------------
; Main Program
;--------------------------------------------------------------------
            org     MAIN
START:      rsp
            sei
            clr     DDRB            ; check user mode mass erase condition
            clr     DDRD            ; PTB0=5V & PTD3=GND in user mode condition
            brclr   0,PTB,USERCODE                          ; check PTB0=5V
            brset   3,PTD,USERCODE                          ; check PTD3=GND
            clr     CONFIG2
            mov     #$31,CONFIG1    ; disable COP & LVI
            clrx
NEXTRAM:
            lda     MASS_ERASE,x    ; Load mass erase code from FLASH to RAM
            sta     RAM,x
            incx
            cbeqx   #{ENDRAM-MASS_ERASE},RUNRAM
            bra     NEXTRAM
RUNRAM:
            jmp     RAM             ; Execute the mass erase
USERCODE:
            bra     *               ; Start of the user application code
;--------------------------------------------------------------------
; Mass Erase
;--------------------------------------------------------------------
MASS_ERASE:
            lda     #$ff            ; unprotect all FLASH area
            sta     FLBPR
            mov     #%01000000,ctrlbyt                      ; setup mass erase
            mov     #10,cpuspd
            ldhx    #$ffff
            jsr     erase_cmd       ; mass erase routine
            ldx     #$0A
Mem_check
            lda     $FFF6,x
            cmp     #$FF
            bne     M_erase
            decx
            bne     Mem_check
ICPMODE:
            bra     *               ; Waiting for power-off the device,
                                    ; then enter the ICP mode using SPGMR &
                                    ; MCUscribe
M_erase
            jmp     RAM
ENDRAM:
            org     RSTVECTOR
            fdb     START           ; RESET
```


How to reach us:

USA/EUROPE/Locations Not Listed: Motorola Literature Distribution; P.O. Box 5405, Denver, Colorado 80217. 1-800-441-2447 or 1-303-675-2140

JAPAN: Nippon Motorola Ltd. SPD, Strategic Planning Office 4-32-1, Nishi-Gotanda, Shinagawa-ku, Tokyo 141, Japan. 03-5487-8488

Mfax™, Motorola Fax Back System: RMFAX0@email.sps.mot.com; http://sps.motorola.com/mfax/; TOUCHTONE 1-602-244-6609; US and Canada ONLY 1-800-774-1848

HOME PAGE: http://motorola.com/sps/