Order this document
by EB312

Motorola Semiconductor Engineering Bulletin

EB312

Replacing 68HC11KA4/KA2 MCUs with 68HC11KS2/KS8 MCUs

By Mario Garcia and Hilario Garza
Transportation Systems Group
Austin, Texas

Introduction

The information in this engineering bulletin is presented to users of the 68HC11KAx Family (KA4/KA2) microcontrollers (MCUs) who are considering converting to the 68HC11KSx Family MCUs (KS2/KS8).

The 68HC11KSx Family was designed to provide more ROM and RAM for KA4 users. Also, a software selectable slow mode feature to reduce power consumption has been added.

Both device families have the same pinouts in the 68-pin PLCC (plastic leaded chip carrier) and 80-pin TQFP (thin quad flat pack) packages.

The 68HC11KSx MCUs have enhancements requiring special design considerations.

Table 1 lists the major features of each device in the families.

Table 1. 68HC11KAx and 68HC11KSx Features

Device Number	ROM in Bytes	RAM in Bytes	EEPROM in Bytes	I/O	Slow Mode
68HC11KA4	24 K	768	640	51	No
68HC11KA2	32 K	1 K	640	51	No
68HC11KS2	32 K	1 K	640	51	Yes
68HC11KS8	48 K	1.5 K	640	51	Yes

Basic Question and Answer

Can 68HC11KAx code run on 68HC11KSx without changes?

Most users can run 68HC11KAx code on 68HC11KSx devices without changes to code. However, the system memory configuration and software use of registers and control bits needs to be reviewed to determine if code changes are needed.

Pin Assignments

The 68HC11KAx and 68HC11KSx devices are pin-to-pin compatible.

Converting

Table 2 — showing differences, issues, and changes — describes the extent of analysis required for various 68HC11KAx Family applications. In each case, the version of 68HC11KAx devices affected and the issues involved in converting to 68HC11KSx devices are identified.

Difference: ROM

The 68HC11KS2 has 32 Kbytes of ROM at address locations $8000–$FFFF, and 68HC11KS8 has 48 Kbytes of ROM at address locations $4000–$FFFF.

The 68HC11KA4 has 24 Kbytes of ROM at address locations $A000–$FFFF, and the 68HC11KA2 has 32 Kbytes of ROM at address locations $8000–$FFFF.

Case 1

Case: Single-chip mode, custom (masked) 68HC11KA4 ROM devices

Issue: ROM code must be resubmitted.

Change required: Source code must be changed to blank out the extra ROM locations of the 68HC11KS2 ($8000–$9FFF) or 68HC11KS8 ($4000–$9FFF) and add the current 68HC11KA4 source code to locations $A000–$FFFF. Also, no reads and/or writes should occur to the system configuration option 3 register (OPT3, $002E).

NOTE: *Location $002E is a reserve register for the 68HC11KAx devices so most users' code should not be addressing this location. Object code (S-records) must be resubmitted to Motorola.*

Case 2

Case: Single-chip mode, custom (masked) 68HC11KA2 ROM devices

Issue: ROM code must be resubmitted.

Change required: No source code changes are needed if the 68HC11KA2 is being replaced with the 68HC11KS2 and no reads and/or writes occur to the system configuration option 3 register (OPT3, $002E).

NOTE: *Location $002E is a reserve register for the 68HC11KAx devices so most users' code should not be addressing this location. Object code (S-records) must be resubmitted to Motorola.*

Source code must be changed if the 68HC11KA2 is being replaced with the 68HC11KS8. Blank out the extra ROM locations of the 68HC11KS8 ($4000–$9FFF), and add current 68HC11KA2 source code to locations $A000–$FFFF. Also, no reads and/or writes should occur to the system configuration option 3 register (OPT3, $002E).

NOTE: *Location $002E is a reserve register for the 68HC11KAx devices so most users' code should not be addressing this location. Object code (S-records) must be resubmitted to Motorola.*

Case 3

Case: Expanded mode, custom (masked) 68HC11KA4 ROM devices

Issue: ROM code must be resubmitted.

Change required: Ensure that address space $8000– $9FFF is not used by external devices (memories and/or peripherals) when replacing it with a 68HC11KS2. Or, ensure that address space $4000–$9FFF is not used by external devices (memories and/or peripherals) when replacing it with a 68HC11KS8. Source code must be changed to blank out the extra ROM locations of the 68HC11KS2 ($8000–$9FFF) or 68HC11KS8 ($4000– $9FFF), and add current 68HC11KA4 source code to locations $A000–$FFFF. Also, no reads and/or writes should occur to the system configuration option 3 register (OPT3, $002E).

NOTE: *Location $002E is a reserve register for the 68HC11KAx devices so most users' code should not be addressing this location. Object code (S-records) must be resubmitted to Motorola.*

Case 4

Case: Expanded mode, custom (masked) 68HC11KA2 ROM devices

Issue: ROM code must be resubmitted.

Change required: No address space conflicts should occur when replacing a 68HC11KA2 with 68HC11KS2. No source code changes are needed if no reads and/or writes occur to the system configuration option 3 register (OPT3, $002E).

NOTE: *Location $002E is a reserve register for the 68HC11KAx devices so most users' code should not be addressing this location. Object code (S-records) must be resubmitted to Motorola.*

Ensure that address space $4000–$9FFF is not used by external devices (memories and/or peripherals) when replacing it with a 68HC11KS8. Source code must be changed to blank out the extra ROM locations of the 68HC11KS8 ($4000–$9FFF), and add current 68HC11KA2 source code to locations $A000–$FFFF. Also, no reads and/or writes should occur to the system configuration option 3 register (OPT3, $002E).

NOTE: *Location $002E is a reserve register for the 68HC11KAx devices so most users' code should not be addressing this location. Object code (S-records) must be resubmitted to Motorola.*

Table 2. Summary of Differences

Item	68HC11KAx	68HC11KSx	Issues
ROM	68HC11KA4 — 24 Kbytes located $A000 to $FFFF 68HC11KA2 — 32 Kbytes located $8000 to $FFFF	68HC11KS2 — 32 Kbytes located $8000 to $FFFF 68HC11KS8 — 48 Kbytes located $4000 to $FFFF	Source code changes; potential address conflicts in expanded mode
RAM	68HC11KA4 — 768 bytes located $0000 to $02FF 68HC11KA2 — 1 Kbyte located $0000 to $03FF	68HC11KS2 — 1 Kbyte located $0000 to $03FF 68HC11KS8 — 1.5 Kbytes located $0000 to $05FF	Potential address conflicts in expanded mode (If 128-byte register block is mapped at the same location as RAM, offset RAM locations by $0080.)
Bootstrap mode	Download terminates after all 768-RAM bytes (68HC11KA4) or 1-K RAM bytes (68HC11KA2) are written or after receiver detects 4 idle characters	Download terminates after all 1-K RAM bytes (68HC11KS2) or 1.5-K RAM bytes (68HC11KS8) are written or after receiver detects 4 idle characters	Download duration
Register	Location $002E is a reserve register	Location $002E is the system configuration option 3 register	Source code changes; potential of enabling the slow mode option, system clocks will slow down

Difference: RAM

The 68HC11KS2 has 1 Kbyte of RAM at address locations $0000–$03FF ($0080– $047F when 128-byte register block is mapped at the same location as RAM), and 68HC11KS8 has 1.5 Kbytes of RAM at address locations $0000–$05FF ($0080–$067F when 128-byte register block is mapped at the same location as RAM).

The 68HC11KA4 has 768 bytes of RAM at address locations $0000–$02FF ($0080–$037F when 128-byte register block is mapped at the same location as RAM), and the 68HC11KA2 has 1 Kbyte of RAM at address locations $0000–$03FF ($0080–$047F when 128-byte register block is mapped at the same location as RAM).

Case 1

Case: Single-chip mode, custom (masked) 68HC11KA4/KA2 ROM devices

Issue: None

Change required: None. However, the user may want to take advantage of the extra RAM space.

Case 2

Case: Expanded mode, custom (masked) 68HC11KA4 ROM devices

Issue: Potential conflict with external memories

Change required: Ensure that address space from $0300–$03FF ($037F–$047F when 128-byte register block is mapped at the same location as RAM) is not used by external devices (memories and/or peripherals) when replacing it with a 68HC11KS2. Or, ensure that address space $0300–$05FF ($037F–$067F when 128-byte register block is mapped at the same location as RAM) is not used by external devices (memories and/or peripherals) when replacing it with a 68HC11KS8.

Case 3

Case: Expanded mode, custom (masked) 68HC11KA2 ROM devices

Issue: Potential conflict with external memories

Change required: No address space conflicts should occur when replacing with 68HC11KS2.

Ensure that address space from $0400–$05FF ($047F–$067F when 128-byte register block is mapped at the same location as RAM) is not used by external devices (memories and/or peripherals) when replacing with a 68HC11KS8.

Difference: Bootstrap Mode

The bootloader function on the 68HC11KAx/68HC11KSx devices terminates a serial download to RAM after all 768 locations of RAM have been filled for the 68HC11KA4, 1-K locations for 68HC11KA2 or 68HC11KS2, and 1.5-K locations for 68HC11KS8.

The bootloader will also terminate downloading after four idle characters have been received by the serial unit. Program control then is passed to the RAM.

Case

Case: All 68HC11KA4 and some 68HC11KA2 devices

Issue: Length of download

Change required: None. However, a larger program may now be downloaded if replacing 68HC11KA4 with 68HC11KS2 or 68HC11KS8 and 68HC11KA2 with 68HC11KS8 due to the additional RAM.

Difference: Register

Reads, writes, and compares of the register in **Table 3** may have an effect on the system when switching between the 68HC11KAx and 68HC11KSx Families.

The bold text in **Table 3** represents the register or bit differences.

Table 3. Register Differences

Register Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$002E	—	SM	—	—	—	—	—	—	**OPT3**

OPT3

Bit 6 in the OPT3 enables the slow mode option when software is selected for the 68HC11KSx Family.

The OPT3 register is a reserve byte in the 68HC11KAx Family.

Special Section: CONFIG Register Verification and Reprogramming Routine

To guarantee proper operation of EEPROM-based M68HC11 devices, the CONFIG (configuration) register must be programmed correctly.

A CONFIG register verification and reprogramming routine should be included at the beginning of critical M68HC11 programs.

This code is an example of how to verify and, if necessary, reprogram the EEPROM CONFIG register to ensure proper operation. These same results can be accomplished also with less generic, user-specified code.

```
* (C) MOTOROLA, INC., 1995 (created 05/01/95)
*
* FILENAME: config.asm
*
* DESCRIPTION: This code checks the CONFIG register on an EEPROM-based
* HC11 device and reprograms it with the proper value if necessary.
*

REGBASE       equ     $0000                ;default beginning of HC11KSx registers

* Offsets from the beginning of the register block.

TOC4          equ     $1C
TCNT          equ     $0E
TFLG1         equ     $23
BPROT         equ     $35
OPTION        equ     $39
PPROG         equ     $3B
CONFIG        equ     $3F

* The following register bit constants are needed.

OC4F          equ     $10
PTCON         equ     $10
CME           equ     $08
BYTE          equ     $10
ERASE         equ     $04
EELAT         equ     $02
EEPGM         equ     $01

* Fill in the blank that follows with the desired CONFIG register value. Also, choose
the appropriate delay based on E frequency.
```

```
* Other user constants should follow, including:

MY_CONFIG     equ       $__                 ;CONFIG should be __
DELAY         equ       $9C40               ;$9C40 = 40,000 counts at 4 MHz
                                            ;$7530 = 30,000 counts at 3 MHz

* Program begins here.

Fill in the program starting address in the following blank.

START  org    $____                         ;program starts here

               lds      #$03FF              ;set a valid stack pointer for 68HC11KS2
                                            ;$047F for KS2 with REGBASE located at
                                             ;$0000
                                            ;$05FF for KS8
                                            ;$067F for KS8 with REGBASE located at
                                             ;$0000

               ldx      #REGBASE            ;set beginning of register block
               ldaa     CONFIG,X            ;read CONFIG
               cmpa     #MY_CONFIG          ;check for valid CONFIG
               beq      NORMAL              ;if CONFIG is OK, go on as usual

At this point, 49 cycles remain for modifications to be made to the time protected
registers (TMSK2, BPROT, OPTION, and INIT), if necessary, on all HC11 devices.

               bclr     BPROT,X,PTCON       ;clear CONFIG protect bit

* CONFIG erase sequence.

               ldaa     #BYTE + #ERASE + #EELAT
               staa     PPROG,X

The EEPROM erase sequence requires that some data be stored to the byte being erased.
The actual data stored and instructions used are irrelevant; it is only necessary to
complete a memory write cycle to the location in question.

               staa     CONFIG,X            ;store something to CONFIG

               ldaa     #BYTE + #ERASE + #EELAT + #EEPGM
               staa     PPROG,X
               jsr      EEDELAY             ;wait 10 ms
               clr      PPROG,X             ;finish erase sequence
```

```
* CONFIG program sequence.

              ldaa    #ERASE + #EELAT
              staa    PPROG,X
              ldaa    #MY_CONFIG           ;desired CONFIG value
              staa    CONFIG,X
              ldaa    #ERASE + #EELAT + #EEPGM
              staa    PPROG,X
              jsr     EEDELAY              ;wait 10 ms
              clr     PPROG,X              ;finish program sequence
* Now allow clock monitor to reset the HC11 and latch the new CONFIG
* register value.

              bset    OPTION,X,CME         ;enable clock monitor reset
              tpa                          ;get condition code register
              anda    #$7F                 ;enable stop mode
              tap
              nop
              stop                         ;enter stop mode and allow reset

* User program resumes normally here.

NORMAL    etc.

* This EEPROM delay subroutine may be used for any EEPROM programming
* or erase operations.

EEDELAY       ldd     TCNT,X               ;get current time

The #DELAY term used for program and erase operations is calculated as follows:

DELAY = ECLK  100

Typical decimal values for DELAY are 30000 at 3 MHz, 20000 at 2 MHz, and 10000 at
1 MHz. Note that these values apply when the timer prescale is divide-by-1.

              addd    #DELAY               ;add delay
              std     TOC4,X               ;allow match at end of delay
              ldaa    #OC4F                ;clear last OC4 match
              staa    TFLG1,X

* Wait for OC4 match (end of 10 ms delay) to occur.

DELAYLOOP brclr     TFLG1,X,OC4F,DELAYLOOP

              rts                           ;end of delay loop
```

Engineering Bulletin

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. Customer Focus Center, 1-800-521-6274

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

ASIA/PACIFIC: Motorola Semiconductors H.K. Ltd., Silicon Harbour Centre, 2 Dai King Street, Tai Po Industrial Estate,
Tai Po, New Territories, Hong Kong, 852-26629298

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/

MOTOROLA