Freescale Semiconductor
Application Note

Document Number: AN4201
Rev. 0, 11/2010

Migrating Applications from S12HY to S12XHY

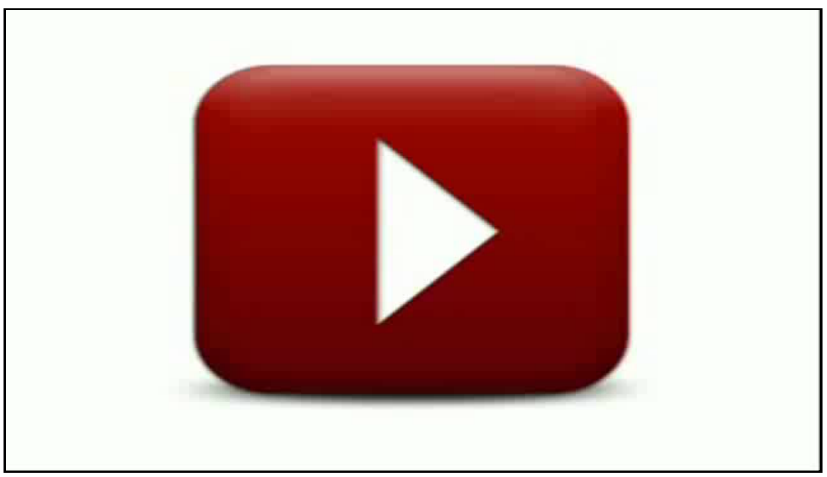
16-bit Automotive Cluster Migration

by: Steven McLaughlin
Microcontroller Support Group
East Kilbride
Scotland

1 Introduction

This application note was created to assist with the migration of cluster applications from the S12HY to the S12XHY micrcontroller. Both devices are part of the same family, they share a high degree of compatibility between their modules and therefore most application software from the S12HY will function on the S12XHY. Similarly, this document is also useful for migrating applications from established microcontrollers, namely the S12HZ and the S12XHZ.

The S12HY and S12XHY are both part of the S12 16-bit microcontroller family by Freescale. The S12XHY is an extension of the S12HY featuring higher performance and additional modules. For specific module information refer to the device's specific reference manual as well as any errata, which are both documented on the Freescale webpage.



Contents

Table 1 highlights the increased peripherals and performance that the S12XHY has over the S12HY. The S12XHY can run with a bus clock up to 40 MHz compared to the S12HY's maximum bus clock of 32 MHz, this is because of the addition of the CPU12XV1 16-bit core to the S12XHY. Direct compatibility remains with the cores. The S12XHY's core has an extension to the condition code register (CCR) that allows nesting of interrupts and blocking of lower priority interrupts.

A key difference in the table is the clock module. The clocks and reset generator (CRG) module is used on the S12XHY whereas the S12HY uses the clocks power and reset management unit (CPMU). The CPMU contains a 1 MHz internal reference clock which is not present on the CRG, however, both clock modules are capable of driving crystals in the range of 4–16 MHz in a pierce configuration.

The S12XHY has an extended nonvolatile memory (NVM), RAM, and introduces an additional MSCAN and SCI module over the S12HY and a new 112 LQFP package.

The S12XHY also introduces a stepper stall detect module which was not present on the S12HY. This effectively allows return-to-zero events of the stepper motor to be managed.

Table 1. Highlighted key differences – S12HY vs. S12XHY

S12HY	S12XHY
32 MHz HCS12 (V1) Core	40 MHz **HCS12X(V1)** Core
External 4–16 MHz pierce oscillator and **1 MHz internal RC oscillator**	External 4–16 MHz pierce oscillator
FM PLL	FM PLL
Up to 64 Kbytes P-flash	**Up to 256 Kbytes P-flash**
4 Kbytes D-flash – EEE	**8 Kbytes D-flash** – EEE
4 k RAM	**12 K RAM**
Pin out: 100/64 LQFP	Pin out: **112**/100 LQFP
4 x stepper motor control	4 x stepper motor control
n/a	**4 x stepper stall detect**
40 x 4 LCD controller	40 x 4 LCD controller
1 x MSCAN	**2 x MSCAN**
1 x SCI	**2 x SCI**
1 x SPI	1 x SPI
1 x IIC	1 x IIC
2 x 8 ch, 16-bit timer	2 x 8 ch, 16-bit timer
8 ch x 8-bit / 4 ch x1 6-bit PWM	8 ch x 8-bit / 4 ch x 16-bit PWM
8 ch 10-bit ADC	**12 ch** 10-bit ADC
22 Key wake-up pins	**25** Key wake-up pins

2 CPU and Instruction Set

The S12XHY features the HCS12XV1 core. This CPU includes enhancements to the programmer model stacking operations, and the instruction set. Direct compatibility remains with both cores.

The S12XHY's core has an extension to the Condition Code Register (CCR) that allows nesting of interrupts and blocking of lower priority interrupts. For the vast majority of users, the differences between the instruction set has minimal impact on the application.

2.1 Programmers model

The HCS12XV1 features an enhanced CCR. This register has been extended to 16 bits to allow stacking of the interrupt priority.

The additional bits in the CCR are shown in Figure 1. The IPL[2:0] indicates the interrupt level of the CPU for the current interrupt.

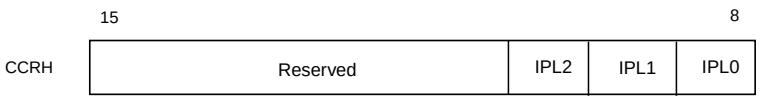


Figure 1. CCR High Byte

The CPU automatically updates the value of the IPL[2:0] to the value of the interrupt currently being serviced.

2.2 Interrupt stacking operation

The CCR has been extended from one to two bytes. This causes the interrupt stack frame to increase by one byte (from nine to ten bytes). Therefore, all stack relative accesses are modified by one byte. See Figure 2.

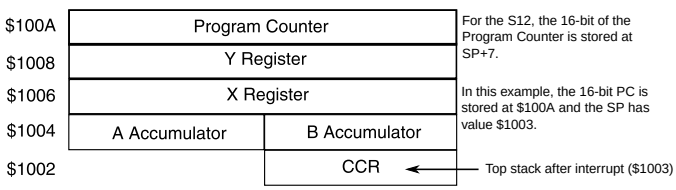


Figure 2. Stack frame example for CPU12

Figure 3 provides an example of a stack frame on the S12XHY after an interrupt has occurred.

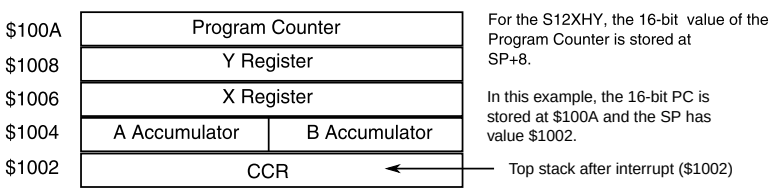


Figure 3. Stack frame example for HCS12XV1

In practice, the requirement to extract information, such as the program counter from an interrupt stack frame is an unusual activity (typically related to debug tools or perhaps task schedulers). Therefore, for the vast majority of users, this difference between the S12HY and S12XHY has little impact.

2.3 Instruction set

The S12XHY features are enhanced instruction set over the S12HY, however, the new CPU retains all of the existing S12HY CPU instructions.

There are four classes of new instructions:

1. New 16-bit, where only an 8-bit accumulator operation existed
2. New memory access instructions, allowing access to linear banks of up to 64 Kbytes
3. New instruction designed to optimize semaphore handling
4. New addressing modes for MOVe instructions

Class 1 improves the data manipulation capabilities of the CPU by allowing direct operation on larger data sizes. On the S12HY, most arithmetic and logical operations such as addition can only take place by using the A, B, or D accumulators. The S12XHY extends this capability to the X and Y registers and adds new instructions for the D register. All arithmetic and logical functions using the A or B accumulator now have a 16-bit counterpart using the X and Y register.

New instructions of this type are as follows:

- ADE—Add with carry, and ADD—Add without carry
- SBE—Subtract with carry, and SUB—Subtract without carry
- DEC—Decrement, and INC—Increment
- AND—Logical AND, OR—Logical OR, and EOR—Logical EXCLUSIVE OR
- NEG—Two's complement, and COM—One's complement
- CLR—Clear register
- BIT—Logical bit test, and TST—Test register
- LSL—Logical shift left, and LSR—Logical shift right
- ASR—Arithmetic shift right, and ASL—Arithmetic shift right
- ROR—Rotate right, and ROL—Rotate left

These new instructions have the same addressing modes as their 8-bit counterparts.

To improve the 32-bit capability of the D-Accumulator, ADED (add with carry) and SBED (subtract with carry) are added. In addition, the CPU provides a set of compare instructions carrying forward the carry and zero flag (CPED, CPEX, CPEY, CPES). This improves the capability to perform 32-bit compares.

While the existing architecture allows 8-bit, and read-modify-write instructions, the S12XHY extends this capability to 16-bit words and provides the following:

- NEGW—Two's complement, and COMW—One's complement
- DECW—Decrement 16-bit, and INCW—Increment 16-bit
- RORW—Rotate right, and ROLW—Rotate left
- LSRW—Logical shift right, and LSLW—Logical shift left
- ASRW—Arithmetic shift right, and ASLW—Arithmetic shift left
- CLRW—Clear memory, and TSTW—Test memory

The addressing modes are the same as their 8-bit counterparts. In general, these new 16-bit operations allow significantly faster manipulation of data compared to the S12HY CPU.

Class 2 provides access to global instructions available on the S12XHY MMC. This allows access to any 64 Kbyte page in global memory based on a new MCU register called GPAGE. The new instructions include all available addressing modes and concatenate the GPAGE register with the 16-bit address data. Global instructions are available for the following instructions:

- GLDAA —Load accumulator A, and GLDAB—Load accumulator B
- GLDD—Load accumulator D
- GLDX —Load X register, and GLDY—Load Y register
- GLDS—Load stack pointer
- GSTAA—Store accumulator A, and GSTAB—Store accumulator B
- GSTD—Store accumulator D
- GSTX—Store X register, and GSTY—Store Y register
- GSTS—Store stack pointer

The GPAGE register is 7 bits wide, therefore the global memory runs from 0x00_0000 to 0x7F_FFFF, and each location is accessible with a single instruction from anywhere in a program (after the GPAGE register is configured for that 64 Kbyte page).

Class 3 allows more efficient use of semaphores, which are important for real time operating systems (RTOS) and for sharing resources between tasks on the CPU. The new instruction is BTAS (bit test and set). Because this is a single instruction, it cannot be interrupted. Therefore, it is useful when requesting access to resources.

Software usually locks resources via a status bit in the RAM, when the bit is set the resource is in use. On the S12HY, you must take care that both tasks do not appear to have allocated the resource. This can occur if one task interrupts another immediately after a bit-test instruction. Therefore, tasks typically disable interrupts while checking and allocating resources. The BTAS

instruction removes this need, as it tests and sets the resource bit in a single instruction step. The BTAS follows the same syntax and allows the same addressing modes as the BSET instruction, except that the test is based on the original data and not on the data written back.

Class 4 is designed to improve the opportunity for compilers to use the memory-to-memory move instructions by allowing the use of all relevant S12XHY addressing modes, and not only those fitting in a single postbyte xb. See the CPU manual for more information on the newly added modes.

3 Interrupt Controller

The S12XHY features the S12XINTV2 interrupt controller module that provides eight interrupt priority levels (I-bit), (some text will mention seven levels, but level 0 equates to disabled interrupts). The XIRQ, SWI, BDM, unimplemented opcode, and system reset interrupts are available as before. In addition, the S12XHY introduces a new interrupt vector to allow handling of ‘spurious’ interrupts which can occur if an interrupt source is removed before the interrupt is managed.

The S12XHY also provides improved detection of invalid software operations which access areas of the MCU's memory that contain no resources. This enhancement applies in single-chip mode and causes a reset if the CPU accesses a memory location that does not address an on-chip memory or peripheral module. The reset vector fetched is the system reset at 0xFFFE.

On the S12HY, the priority of any interrupt is determined by its position in the interrupt vector table. This is valid on the S12XHY. Vectors closer to the top of the memory (0xFFFF) have a higher priority than those that are lower.

On the S12XHY, you can allocate each interrupt source to one of eight possible interrupt levels. You can change these levels at any time. This is not recommended while interrupts are active.

An interrupt can be taken only if it is enabled, the global mask (I-bit) clear, and it is a higher level than the current working interrupt. The CPU is aware of and stacks the interrupt level where it is working. For example, this means that the CPU cannot take a level 1 interrupt if it has not returned from processing a level 5 interrupt, even if the I-bit is clear. Conversely, the CPU cannot take a level 5 interrupt if it is working at level 1. The CPU will not take the level 5 interrupt if the I-bit is set. As with the S12HY, the I-bit is set automatically on entry to an interrupt, therefore the code within each Interrupt Service Routine (ISR) must explicitly clear the I-bit using the CLI instruction, if nested interrupts are desired. Customers who do not want to use nested interrupts still benefit from the seven different priority levels, as the highest priority interrupt will always be selected from those pending.

When the CPU returns from an interrupt, part of its new functionality is to recover the interrupt level at which it was working before the interrupt was taken. This is stored in the upper byte of the CCR (see Programmers model).

An additional feature of the interrupt module is the ability to specify the location of the interrupt vector table in the memory. This is achieved by using the Interrupt Vector Base Register (IVBR). The IVBR specifies the top eight bits of the vector table 16-bit address and can be changed at any time. This is not recommended while interrupts are active. The vector table always exists in the main 64 Kbyte map of the CPU. This ability to move the vector table allows you to have multiple vector tables for multiple mode operating systems, debugging systems, and bootloaders. The IVBR defaults to 0xFF out of reset for compatibility with the S12HY.

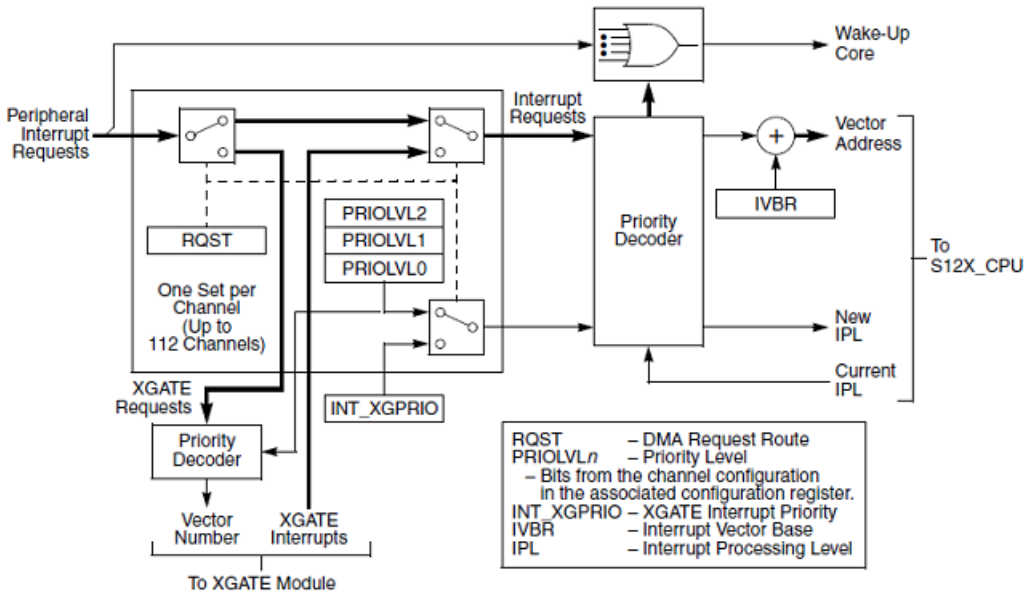


Figure 4. S12XHY interrupt module

NOTE

XGATE is not a feature on the S12XHY MCU.

Figure 5 illustrates a typical profile of the interrupt processing levels possible when using the interrupt controller.

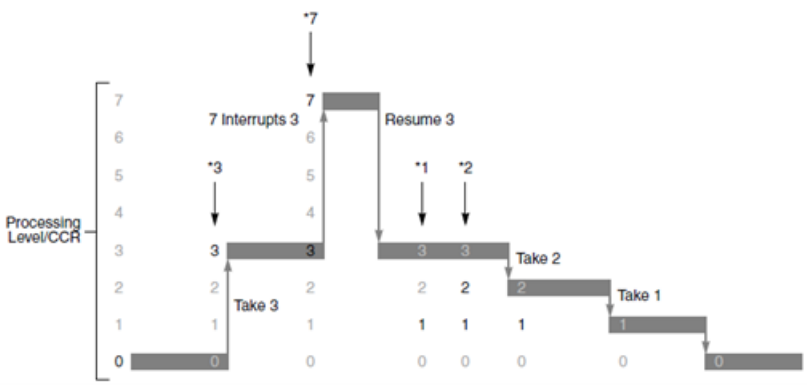


Figure 5. Interrupt profile

In this example, the CPU is initially not executing an interrupt service routine. At the point marked by *3, the CPU recognizes an interrupt with a priority level of three and the CPU begins executing the Interrupt Service Routine (ISR). While executing the level three ISR, an interrupt with a level 7 priority occurs and the CPU begins to execute the ISR at the point marked by *7. This ISR runs to completion and the level 3 ISR then resumes execution.

During the remaining execution time of level 3 ISR, a level 1 and level 2 interrupt, marked by *1 and *2 occur.

NOTE

Because these interrupts have a lower priority than the currently executing ISR, the CPU does not execute their ISRs. Instead, the interrupts remain pending.

At the completion of level 3 ISR, the CPU executes level 2 ISR first because it has the highest priority of the two pending interrupts. Finally, at the completion of level 2 ISR, the CPU executes level 1 ISR and runs to completion.

NOTE

For interrupt nesting to occur as shown in this example, the software must clear the I-bit in the CPU's CCR at the start of each ISR. You must take care not to set the interrupt level to

0 in the interrupt module. Doing so disables all interrupts from that interrupt source regardless of the settings of the peripheral's local interrupt enable bits.

4 Package and Pin-out Differences

Both devices are available in the 100 LQFP package however, differences arise between the two as the S12HY is available in the 64 LQFP, which is not available on the S12XHY and the alternative package for the S12XHY is 112 LQFP.

Although pin-out placement is not 100% compatible Figure 6 shows a high degree of similarity of pin placement around the MCU pinout_S12HY_S12XHY

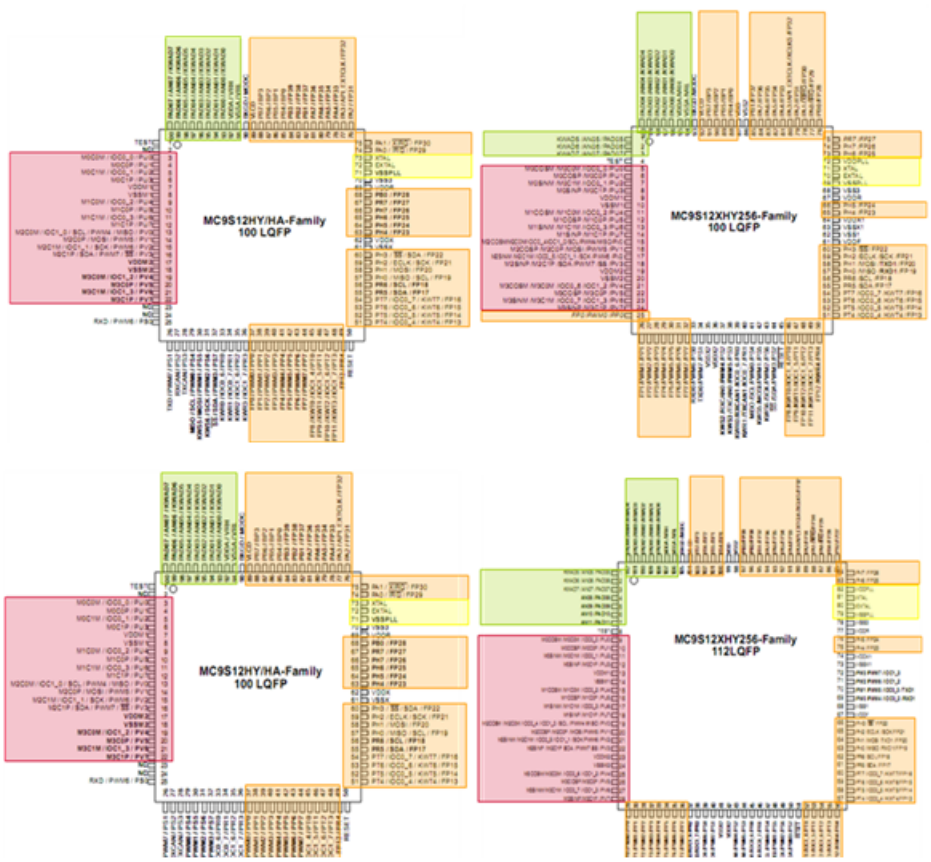


Figure 6. Pin-out for both 100 and 112 LQFP—S12HY vs. S12XHY

Analogue pins (green) are found on the north side of the MCU, oscillator and clock specific pins are placed on the east side (yellow) and pins which are used for motor control are found on the west side (red). LCD control pins are located north, east and south as shown (orange). The backplane and frontplane pins have been held in the same location.

This has been designed to ease any future transitions from S12HY to S12XHY. Re-designs of hardware shall not be overly cumbersome. This is true for both the 100 and 112 LQFP S12XHY devices.

The Port M[3:0] is additional on the S12XHY, which also features SCI1. It is worth noting there are several voltage supply pins which are additional:

- VDDF
- VDDPLL
- VDD

These are supplied via the internal voltage regulator.

The reset (with internal pull-up) and test (with internal pull down—grounds the test pin in production despite an internal pull down being available) pins featured on the S12XHY have the same functionality as on the S12HY.

4.1 Memory map

The memory mapping structure is different. Both devices use a paged memory scheme with the S12HY having a 256 K global memory map and the S12XHY using an 8 MB global memory map (the same as the S12XS and S12XE). Both devices share a common 64 K local memory map shown below in Figure 7.

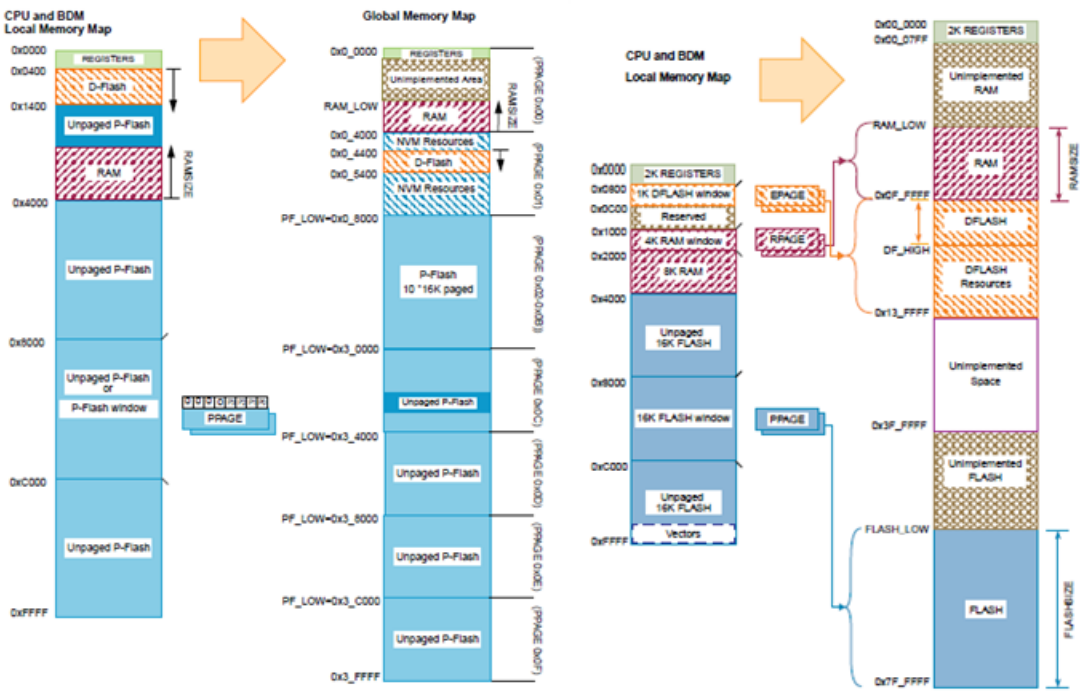


Figure 7. S12HY (left) and S12XHY (right) memory maps

The 64 K local memory map spans from 0x0000 to 0xFFFF for both devices as Figure 7 illustrates. The register has the same start address at 0x0000 and the S12XHY has additional modules that have been added. In comparison to the S12HY, registers have been added to what was previously reserved space.

Differences exist between the RAM, Program (P)-flash, and Data (D)-flash. It is possible to update these to initialize an S12HY project on the S12XHY. The following section outlines the distinct RAM and NVM differences and highlights the address ranges and pages.

4.1.1 Program flash (P–Flash)

Table 2. P-Flash — S12HY vs. S12XHY

Device	Address Range	Size	Paged/Unpaged
S12HY/HA64	0x1400 – 0x2FFF	7 K	Unpaged (Equivalent to PPAGE 0C_2)[1]
	0x4000 – 0x7FFF	16 K	Unpaged (equivalent to PPAGE 0D)
	0x8000 – 0xBFFF	16 K (x1.5)	Paged (PPAGE 0C-0F [0D and 0F not used])
	0xC000 – 0xFFFF	16 K	Unpaged (Equivalent to PPAGE 0F)
S12HY/HA48	0x4000 – 0x7FFF	16 K	Unpaged (Equivalent to PPAGE 0D)
	0x8000 – 0xBFFF	16 K (x1)	Paged (PPAGE 0D-0F [0D and 0F not used])
	0xC000 – 0xFFFF	16 K	Unpaged (Equivalent to PPAGE 0F)
S12HY/HA32	0x8000 – 0xBFFF	16 K (x1)	Paged (PPAGE 0E-0F [0F not used])
	0xC000 – 0xFFF	16 K	Unpaged (Equivalent to PPAGE 0F)
S12XHY256	0x4000 – 0x7FFF	16 K	Unpaged (Equivalent to PPAGE FD)
	0x8000 – 0xBFFF	16 K (x14)	Paged (PPAGE F0-FF [FD and FF not used])
	0xC000 – 0xFFFF	16 K	Unpaged (Equivalent to PPAGE FF)
S12XHY128	0x4000 – 0x7FFF	16 K	Unpaged (Equivalent to PPAGE FD)
	0x8000 – 0xBFFF	16 K (x6)	Paged (PPAGE F8 –FF [FD and FF not used])
	0xC000 – 0xFFFF	16 K	Unpaged (Equivalent to PPAGE FF)

1. Does not have the VAE bit on the S12XHY.

The P–Flash consists of both paged and unpaged memory. This application note is not a guide to this memory design. For guidance in paged memory, see the application note titled *HCS12X Family Memory Organization* (document AN2734) as well as on-demand training on www.freescale.com/training go to S12(X) Banked Memory Made Easy and MC9S12XE Memory Paging using Codewarrior Examples.

Table 2 illustrates that the local addresses remain the same with paged and unpaged regions remaining compatible. There are differences between the global addresses, however adjusting linker files to account for the additional memory (and address ranges) is required for migration.

Considering the largest flash size device, S12XHY256 and S12HY/HA64, the P-flash local areas are the same, with the exception that the S12HY/HA64 has an additional unpaged region which is equivalent to the paged RAM area on the S12XHY256. The S12XHY256 has additional paged memory windows that are accessed at 0x8000–0xBFFF. When migrating from the S12HY, these additional pages must be included within the system software. The PPAGE register within the MMC module is responsible for generating the 23-bit address required for accessing the paged memory. The S12HY MMC contains a PPAGE register which is not identical to the S12XHY as there are less pages.

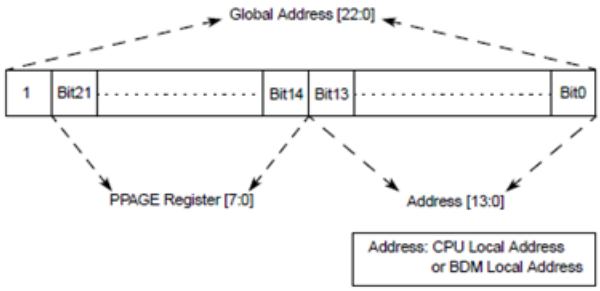


Figure 8. S12XHY PPAGE register

4.1.2 Data flash (D–Flash)

Table 3. D-Flash—S12HY vs. S12XHY

	Address Range	Size	Paged/Un-paged
S12HY/HA[1]	0x0400 – 0x13FF	4 K	Unpaged
S12XHY128/256	0x0800 – 0x0BFF	1 K win-dow (8 K in total)	Paged (EPAGE 0x00–0x07)

1. This is for all flash variants, S12HY/ HA64/ 48/ 32

Differences exist between the D-flash of both devices. Table 3 shows that the S12XHY has twice as much D-flash as the S12HY. This is accessed within a banked memory region accessible with different pages as defined by the EPAGE register. In total, the S12XHY has 8 K of D-flash which can be accessed via 8x1 K windows.

On the S12XHY the address range of 0x400–0x13FF; equivalent to the S12HY's D-flash is occupied as a reserved area. It is vital to update the address range for the D-flash to point to the new address as outlined in Table 3. Any writes to 0x400–0x13FF are ignored and the reads return zero for the S12XHY.

The EPAGE register which is not present on the S12HY is responsible for generating the 23-bit address for accessing the paged D-flash. This 23-bit address is constructed from the 8-bit EPAGE register, 10-bit CPU/BDM address information, and the fixed 5-bits as shown in Figure 9.

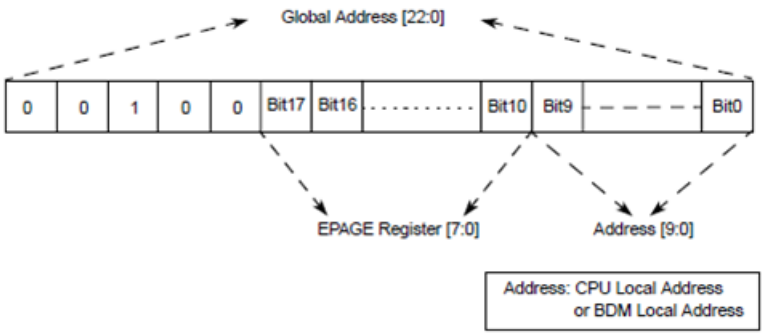


Figure 9. S12XHY EPAGE Register–Used to page 8x1 K blocks into the D-flash page window

D-Flash is typically used for implementing the emulated EEPROM (EEE). The Emulated EEPROM software driver discusses the Freescale EEE software driver and the steps taken to update this from the S12HY to the S12XHY.

4.1.2.1 Emulated EEPROM software driver

Freescale has produced a software driver capable of emulated EEPROM. It uses several partitions of the D-flash to perform continuous write and erase, contains features to protection against potential brownouts, and is fully configurable for the user.

This section has so far discussed the difference in the memory map. This is something that has to be noted when using this software driver. Currently, for the S12X family the emulated EEPROM driver has been written for S12P128, S12HY64, and S12SX256. The S12XS256 has an identical memory map to S12XHY256. Therefore, the S12XS256 emulated EEPROM driver can be used for the S12XHY256.

NOTE

The EEE driver has been generically developed for the latter devices. Selection of the appropriate MCU is performed via the **#define** as shown in Figure 10.

```
/*--------------- SUBJECT TO CHANGES ACCORDING TO COMPILER -------------*/
/*For selecting Codewarrior compiler define COMPILER_SELECT as CODE_WARRIOR   */
/*For selecting Cosmic compiler define COMPILER_SELECT as COSMIC_BANKED */
#define COMPILER_SELECT                 CODE_WARRIOR

/*For selecting S12XS family define SGF18_SELECT as S12XS_SGF18   */
/*For selecting S12P family define SGF18_SELECT as S12P_SGF18   */
/*For selecting S12HY family define SGF18_SELECT as S12HY_SGF18   */
#define SGF18_SELECT                    S12XS_SGF18

/*-------------------- Callback function enable ------------------*/
#define CALLBACK_ENABLE                 TRUE
```

Figure 10. #define — code to select the appropriate MCU for EEE in SGF18.h

Moving from the S12HY to the S12XHY EEE applications also require adjustments to the complier linker file because of the difference in the memory map stated in this section. The S12XHY requires additional segments to be added are outlined here. As a guide, the linker file for the S12XS256 can be used as a direct replacement of the S12HY64 when moving to the S12XHY256. Care must be taken that segments and placements names within the linker file are maintained.

4.1.3 RAM

Table 4. RAM — S12HY vs. S12XHY

	Address Range	Size	Paged/Unpaged
S12HY/HA[1]	0x3000 – 0x3FFF	4 K	Unpaged
S12XHY128/256	0x1000 – 0x1FFF	4 K Window	Paged (RPAGE 0xFD)
S12XHY128/256	0x2000 – 0x3FFF	8 K	Unpaged

1. This is for all flash variants, S12HY/ HA64/ 48/ 3

The S12XHY contains three times as much RAM as the S12HY. The RAM on the S12HY is an unpaged 4 Kbytes region at 0x3000–0x3FFF which equates to the same 4 K region on the S12XHY. However, the S12XHY has twice as much unpaged RAM and the unpaged region spans wider than the S12HY from 0x2000–0x3FFF. Additionally, the S12XHY features a 4 K paged RAM window that can be accessed at local address 0x1000–0x1FFF.

NOTE

This address on the S12HY equates to unpaged P-flash and any application code or similar non-volatile storage is not recommended to be stored at this address.

The RPAGE register within the MMC module is responsible for generating the 23-bit address to access the paged RAM on the S12XHY. The S12HY does not have paged RAM and hence no RPAGE register. The 23-bit global address is constructed from three fixed bits, 8-bit RPAGE, and a 12-bit CPU and BDM address as shown in Figure 11.

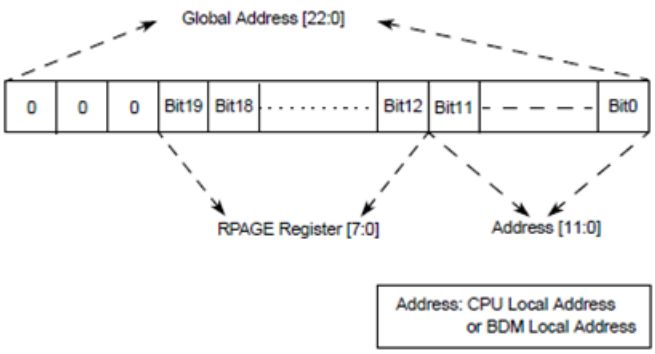


Figure 11. S12XHY RPAGE register – Used to page 4 K blocks in RAM paged window

NOTE

Both the S12HY and S12XHY contain the same LCD module. There is a dedicated 20 bytes of RAM. If required, this RAM can also be used for general purposes, if the LCD is disabled.

5 Clock Module and On-Chip Voltage Regulator

One major difference at the core of these devices is the clock module. The S12HY uses the clocks and power management unit (CPMU) which features a 1 MHz IRC and can also be used with an external oscillator with or without the PLL. The CPMU module also integrates the API, RTC, and VREG functionality of the chip. Although the device has this internal clock source option, it is recommended to use an external crystal, if CAN communications are to be used. External crystals in the range of 4–16 MHz may be used in the pierce configuration.

The S12XHY uses the clocks and reset generator (CRG) that does not have an internal clock source option, but similarly can use external crystals in the 4–16 MHz pierce configuration. The application note titled *Comparison of the S12XS CRG Module with S12P CPMU Module* (document AN3622) has been written to detail the differences between these two modules at the register level.

On the S12XHY the voltage regulator (VREG) is a separate module that is responsible for controlling low voltage interrupts and the autonomous periodic interrupt (API). The application code has to be slightly altered to make adjustments for migrating to the S12XHY. Because the registers that control the VREG function on the S12HY are similar to the S12XHY's VREG registers, this should be a minimal effort.

Table 5. Illustrating the equivalent VREG registers — S12HY vs. S12XHY

S12HY	S12XHY
High Temperature Control Register (CPMUHTCTL)[1]	High Temperature Control Register (VREGHTCL)
Low Voltage Control Register (CPMULVCTL)	VREG Control Register (VREGCTRL)

S12HY	S12XHY
Autonomous Periodical Interrupt Control Register (CPMUAPICTL)	Autonomous Periodical Interrupt Control Register (VREGAPICL)
Autonomous Periodical Interrupt Trimming Register (CPMUAPITR)	Autonomous Periodical Interrupt Trimming Register (VREGAPITR)
Autonomous Periodical Interrupt Rate High and Low Register (CPMUAPIRH / CPMUAPIRL)	Autonomous Periodical Interrupt Rate High and Low Register (VREGAPIRH / VREGAPIRL)
High Temperature Trimming Register (CPMUHTTR)	High Temperature Trimming Register (VREGHTTR)

1. Does not have VAE on the S12XHY

6 Motor Control and Stepper Stall Detect

Both devices use the same stepper motor control module and can drive up to four stepper motors directly from the MCU with no additional components. The motor control application code developed for the S12HY is operational on the S12XHY. Moreover, motor control applications that have been developed on other Freescale cluster microcontrollers such as the S12HZ, S12XHZ, and MPC560xS are compatible with the S12HY and S12XHY because of the common registers between them.

NOTE

The S12HY 64LQFP package, does not contain a second motor control supply (that is, VDDM2) therefore it is possible to drive three stepper motors, not four. The S12XHY, 100 and 112 LQFP packages do not have this limitation and have both VDDM1 and VDDM2 and therefore directly drive four stepper motors.

A hardware stepper stall detect (SSD) module is present on the S12XHY. Physically, the SSD functionality is bonded onto the Port U and Port V pins, (the same as the motor control pins). Therefore, the SSD functionality can be performed on external stepper motors where the SSD module takes control from the motor control module, can detect stall events, and return the pointer to zero. From an S12HY perspective this is a completely new functionality, because the S12HY does not contain a stepper stall detect module. The stepper stall detect module present on the S12XHY is similar to the SSD module on other Freescale cluster microcontrollers such as the S12HZ, S12XHZ, and MPC560xS. It is possible to use applications on the latter microcontrollers for the S12XHY.

A software driver for stepper stall detect was developed for the S12HY. The software driver uses the timer module to measure the decay time of the back-EMF of the stepper motor and determine if a stall condition has occurred. Compatibility with the motor pins has been maintained with the S12XHY, whereby the timer pins have been multiplexed with the motor control pins just like the S12HY. For customers who were using the stall detect software solution for their motor control applications, SSD software solution can easily be ported to the S12XHY. Figure 12 shows the coil minus pins of the motor control pins that are multiplexed with timer channels for input capture. The application note titled *High Speed Stall Detection on the S12HY and S12XHY Family* (document AN4024) has detailed information about this technique with sample software for both the S12HY and S12XHY.

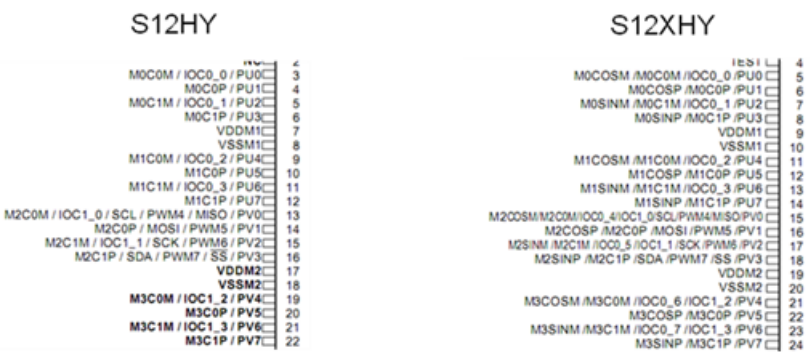


Figure 12. Motor control pins multiplexed with timer pins for a software stepper stall detect

7 LCD Module

Both the S12HY and S12XHY use the same LCD module capable of driving up to 160 segments, via 40 frontplanes and 4 backplane electrodes, however, there are slight differences that should be noted between the two devices. On the S12HY, the LCD module is always clocked via the internal IRC1M. But on the S12XHY, the LCD is clocked from the main OSC clock which means the LCDCR0_LCLK settings may have to be configured differently.

Another major difference with the LCD on the S12XHY, is it has the capability to be operated in pseudo stop mode, on the S12HY it cannot. This mode is useful for quicker recovery than full stop mode because the oscillator remains powered. Pseudo stop mode is entered via the **asm STOP** instruction, remembering that the CLKSEL_PSTP should be set, and the stop (S) bit of the CPU's condition code register (CCR) must be cleared prior to entry.

8 Communication Modules

Both the S12HY and S12XHY contain at least one SPI, IIC, SCI, and MSCAN module. The S12XHY adds an additional SCI and MSCAN module. Both devices use the same modules:

- SCI—S12SCIV5
- SPI—S12SPIV5
- MSCAN—S12MSCANV3
- IIC—IICV3

There is an identical compatibility with the registers to operate the communication functionality. Differences arise between some of the ports where signals are routed on both devices as detailed in Table 6.

Table 6. Routing possibilities of IIC, SPI, SCI, and MSCAN modules

	S12HY				S12XHY		
	De-fault	Option	Option	Option	Default	Option	Option
IIC	PS[7:4]	PR[6:5]	PR[6:5]	PR[6:5]	PS[7:4]	PR[6:5].	PV[3:0]
SPI	PS[7,4]	PV[3,0]	PH[3,0]		PS[7,4]	PV[3,0]	PH[3,0]
SCI0	PS[1:0]	—			PS[1:0]		
SCI1					PM[1:0]	PH[1:0]	
MS-CAN0	PS[3:2]				PS[3:2]		
MS-CAN1					PR[1:0]		

Out of reset, Table 6 shows that the default pins for the communication modules are identical and the majority of routing options are similar. Routing of the IIC and SPI is carried out via the PTSRR register, the only difference being that IIC cannot be routed to PH[3:0] on the S12XHY. It is possible on the S12HY.

9 Timer Module

The timer module is identical on both devices (TIM16B8CV2). They both have two timers, TIM0 (PT0–7) and TIM1 (PP0–7). Similar to the IIC and SPI, certain timer output compare/input capture pins can be routed to either port R or port T (or port V—S12XHY only). The PTTRR is the register responsible for routing the timer pins. On the S12XHY bits 0–7 are available to use, however the S12HY has bits 0, 1, 4, and 5 in use. Bits 2, 3, 6, and 7 are not implemented.

Table 7. Timer options as detailed by PTTRR register for the S12HY and S12XHY

	S12HY			S12XHY			
	Function	Default Pin	Option Pin	Function	Default Pin	Option Pin	Option Pin
Bit 7 - PTTRR [7:6]	—	—	—	I0C0_7	PT7	PR1	PV6
Bit 5 - PTTRR 5	I0C0_7	PT7	PR1	I0C0_5	PT5	PV2	—
Bit 4 - PTTRR 4	I0C0_6	PT6	PR0	I0C0_4	PT4	PV0	—
Bit 3 – PTTRR[3:2]	—	—	—	I0C0_6	PT6	PR0	PV4
Bit 1 - PTTRR 1	I0C1_7	PT3	PR3	I0C1_7	PT3	PR3	—
Bit 0 - PTTRR 0	I0C0_6	PT2	PR2	I0C1_6	PT2	PR2	—

The noticeable difference between the two devices is the PTTRR register. On the S12HY, bits 5 and 4 refer to a different timer channel than the S12XHY. However, the routing between the channel and the pad remains compatible therefore any alterations have to be considered in the software.

Appendix A Additional Information

The following material is helpful in transitioning from the S12HY to the S12XHY.

A.1 Application Notes

- AN3622—Comparison of the S12XS CRG Module with S12P CPMU Module
- AN3613—Using the MC9S12XS Family as a Development Platform for the MC9S12P Family
- AN3961—EEPROM Emulation for the MC9S12XS and MC9S12P Families Using AN2302/D as a Reference
- AN4024—High Speed Stall Detection on the S12HY Family
- AN4021—MC9S12HY-Family Demonstration Lab Training
- AN2734—HCS12X Family Memory Organization
- AN2974—Stepper motor quick start guide
- AN4037—Driving a Stepper Motor with MPC560xS Stepper Motor Control Module
- AN3330—Introduction to the Stepper Stall Detector Module
- AN3412—Dynamic LCD Driver Using GPIO Pins
- AN3219—TN/STN LCD Driver

A.2 Software

- AN3961 Code
- AN4021 Code
- Emulated EEPROM Driver
- AN4024 Code: SSD Software Example

A.3 Tools, Trainings, and Videos

- Freescale Instrument Cluster Homepage
- iPLL Calculator tool document and software
- On-demand training on www.freescale.com/training
  - S12(X) Banked Memory Made Easy
  - MC9S12XE Memory Paging using Codewarrior Examples
- S12HY64 Cluster Application Demonstration Video

Appendix B References

The following documents have assisted in the creation of this application note:

- Application note titled *S12HZ and S12XHZ Family Compatibility* (document AN3510)
- Application note titled *Comparison of the S12XS CRG Module with S12P CPMU Module* (document AN3622)
- MC9S12HY Reference Manual
- MC9S12XHY Reference Manual
- CPU12X Core Reference Manual

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or +1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Document Number: AN4201
Rev. 0, 11/2010