AN00143: Interfacing FlashRunner with Freescale S12/S12X Devices

by Andrea Favret (andrea.favret@smh-tech.com)

FlashRunner is a Universal In-System Programmer, which uses the principles of In-Circuit Programming to program Freescale S12 and S12X family microcontrollers.

This Application Note assumes that you are familiar with both FlashRunner and the main features of the S12/S12X families. Full documentation about these topics is available in the FlashRunner user's manual and in device-specific datasheets.

1. Introduction

In-system programming of S12/S12X microcontrollers is performed by entering the device's BDM (Background Debug Module) mode, which allows the programming of the MCU memory, through a synchronous serial protocol.

In order to use FlashRunner to perform in-system programming, you need to implement the appropriate in-circuit programming hardware interface on your application board.

SMH Technologies

Revision 1.1 – July 2008

freescale™ Alliance Member

E-mail (general information): info@smh-tech.com

E-mail (technical support): support@smh-tech.com

Web: http://www.smh-tech.com

DC10044

2. Hardware Configuration

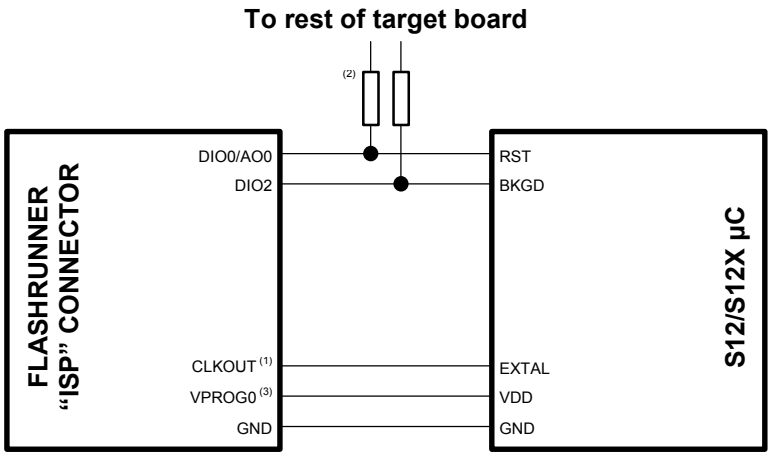
The microcontroller lines needed to implement the BDM interface and program an S12/S12X device are the following:

- **BKGD:** Single-wire background interface pin. The BKGD pin is used for bidirectional communication of active background mode commands and data between FlashRunner and the target MCU.
- **VDD:** Device power supply voltage. The range of operating voltage is typically 3.3–5.0V, depending on the MCU.
- **RST:** Device reset input/output pin.
- **GND:** Device power supply ground.

If you want FlashRunner to clock your target device, you must also connect FlashRunner's CLKOUT pin to the EXTAL pin of your target device.

Freescale has defined a standard 6-pin connector (BDM) that allows an interface pod to be connected to any target S12/S12X family MCU.

The lines mentioned above must be connected to the FlashRunner's "ISP" connector according to the following diagram:



Notes

(1) Connect this line if you want the target device to be clocked by FlashRunner. Please note that the voltage level of FlashRunner CLKOUT signal is that specified by the VDD parameter (see later). This could be an issue for certain devices for which the clock signal has a voltage level different than the supply voltage's. In this case you cannot use FlashRunner to clock the target device.

(2) If the lines needed to enter monitor mode are used for other purposes in the application, series resistors should be implemented to avoid a conflict in case the rest of the target board forces the signal level. If these lines are used as outputs, these resistors are not necessary.

(3) It is not necessary to connect this line if you externally power the MCU.

Note: *FlashRunner drives the RST line of the target microcontroller in order to enter the BDM mode. Make sure there is no external watchdog/LVD circuitry in the target board that interferes with the correct RST driving.*

3. Specific TCSETPAR Programming Commands

Overview

`TCSETPAR` commands set device-specific and programming algorithm-specific parameters. These commands must be sent after the **`TCSETDEV`** command and before a **`TPSTART`** / **`TPEND`** command block.
All of the FlashRunner programming capabilities rely on the background monitor mode of the target device. In order to enter this special mode (which establishes a communication channel between the target device and FlashRunner) and configure it properly, all of the following parameters (expection made for the Reser driving mode) must be correctly specified through the relative **`TCSETPAR`** commands (although the order with which these parameters are set is not important):

- V_{DD};
- Auxiliary V_{DD};
- Power up time;
- Power down time;
- Reset up time;
- Reset down time;
- Reset driving mode;
- External oscillator frequency;
- PLL frequency;
- PLL frequency multiplier;
- PLL frequency divisor;
- CLKOUT frequency.

TCSETPAR VDD

Command syntax:

`TCSETPAR VDD <voltage mV>`

Parameters:

`voltage mV`: Target device supply voltage, expressed in millivolts.

Description:

This command is used to properly generate the voltage level of the ISP signals. Additionally, the specified voltage is routed to the VPROG0 line of the FlashRunner “ISP” connector, which can be used as a supply voltage for the target board.

TCSETPAR VDD_AUX

Command syntax:

`TCSETPAR VDD_AUX <voltage mV>`

Parameters:

`voltage mV`: Auxiliary supply voltage, expressed in millivolts, in the range 3000-14500mV.

Description:

This command is used to generate an optional, auxiliary voltage level for user purposes. The specified voltage is routed to the VPROG1 line of the FlashRunner “ISP” connector.

If you don’t need to use the auxiliary V_{DD} line, specify a value of `0`.

TCSETPAR PWUP

Command syntax:

`TCSETPAR PWUP <time ms>`

Parameters:

`time ms`: Power rising time, expressed in milliseconds.

Description:

This command is necessary because, to enter the BDM mode, FlashRunner must properly drive the V_{DD} line during the power-on reset.

The V_{DD} rising time (PWUP) is expressed in milliseconds and depends on the features of your target board. Make sure to choose a value large enough to ensure that the V_{DD} signal reaches the high logic level within the specified time. Note that, if the V_{DD} line has a high load, a longer time is required for the V_{DD} signal to reach the high logic level. If PWUP is not long enough, FlashRunner could not be able to enter the BDM mode.

TCSETPAR PWDOWN

Command syntax:

`TCSETPAR PWDOWN <time ms>`

Parameters:

`time ms`: Power falling time, expressed in milliseconds.

Description:

The V_{DD} falling time (PWDOWN) is expressed in milliseconds and depends on the features of your target board. Make sure to choose a value large enough to ensure that the V_{DD} signal reaches the low logic

level within the specified time. Note that, if the V_{DD} line has a high load, a longer time is required for the V_{DD} signal to reach the low logic level.

TCSETPAR RSTUP

Command syntax:

`TCSETPAR RSTUP <time µs>`

Parameters:

`time µs`: Reset rising time, expressed in microseconds.

Description:

The Reset rising time (RSTUP) is expressed in microseconds and depends on the features of your target board. Make sure to choose a value large enough to ensure that the Reset signal reaches the high logic level within the specified time. Note that, if the Reset line has a high load, a longer time is required for the Reset signal to reach the high logic level. If RSTUP is not long enough, FlashRunner could not be able to enter the BDM mode.

Note: *on S12X devices, the Reset rising time must be shorter than the COP module timeout.*

TCSETPAR RSTDOWN

Command syntax:

`TCSETPAR RSTDOWN <time µs>`

Parameters:

`time µs`: Reset falling time, expressed in microseconds.

Description:

The Reset falling time (RSTDOWN) is expressed in microseconds and depends on the features of your target board. Make sure to choose a value large enough to ensure that the Reset signal reaches the low logic level within the specified time. Note that, if the Reset line has a high load, a longer time is required for the Reset signal to reach the low logic level. If RSTDOWN is not long enough, FlashRunner could not be able to enter the BDM mode.

TCSETPAR RSTDRV

Command syntax:

`TCSETPAR RSTDRV PUSHPULL|OPENDRAIN`

Options:

`PUSHPULL`: Reset line is driven in push-pull mode.

`OPENDRAIN`: Reset line is driven in open-drain mode.

Description:

Specifies how the Reset line is driven by FlashRunner.

TCSETPAR FOSC

Command syntax:

`TCSETPAR FOSC <frequency Hz>`

Parameters:

`frequency Hz`: External oscillator frequency, expressed in Hertz.

Description:

Specifies the target microcontroller's external clock frequency.

TCSETPAR PLLFREQ

Command syntax:

`TCSETPAR PLLFREQ <frequency Hz>`

Parameters:

`frequency Hz`: PLL frequency, expressed in Hertz.

Description:

This command specifies the target microcontroller's internal PLL frequency, which is twice the bus frequency.

Please note that programming time can be reduced by driving the internal PLL circuitry to the maximum allowed frequency. Make sure that the value you set for the **`frequency Hz`** parameter is included in the VCO clock frequency range, specified in the specific device datasheet.

To disable the PLL, set **`PLLFREQ`**, **`REFDIV`** and **`SYNR`** parameters to **`0`**.

The PLL frequency must be calculated with the following formula:

$$PLLFREQ = \begin{cases} 0 \ \text{(to disable the PLL)} \\ 2 \cdot FOSC \cdot \dfrac{SYNR + 1}{REFDIV + 1} \end{cases}$$

TCSETPAR REFDIV

Command syntax:

`TCSETPAR REFDIV <divisor>`

Parameters:

`divisor`: PLL divisor.

Description:

This command specifies the target microcontroller's PLL divisor. Please refer to the specific device datasheet for the range of allowed PLL divisors.

To disable the PLL, set **`PLLFREQ`**, **`REFDIV`** and **`SYNR`** parameters to **`0`**.

TCSETPAR SYNR

Command syntax:

`TCSETPAR SYNR <multiplier>`

Parameters:

`multiplier`: PLL multiplier.

Description:

This command specifies the target microcontroller's PLL multiplier. Please refer to the specific device datasheet for the range of allowed PLL multiplier.

To disable the PLL, set **`PLLFREQ`**, **`REFDIV`** and **`SYNR`** parameters to **`0`**.

TCSETPAR CLKOUT

Command syntax:

`TCSETPAR CLKOUT 25000000|12500000|6250000|0`

Command options:

Frequency of the clock signal to be generated at the CLKOUT pin of the FlashRunner "ISP" connector, expressed in Hertz. The available clock frequency values are 25MHz, 12.5MHz, 6.25MHz and 0.

Description:

Generates an auxiliary clock signal at the CLKOUT pin of the FlashRunner “ISP” connector. This signal can be used to speed up programming (when you want to use a clock faster than that provided by your target board).

Make sure that the clock frequency you select is not greater than the maximum allowed frequency for your device. Note that, since all devices feature an internal frequency divisor, the actual bus frequency will be a fraction of the CLKOUT frequency.

If you don’t need to use this feature, specify `0` as the CLKOUT frequency.

4. Specific TPCMD Programming Commands

Overview

`TPCMD` commands perform a programming operation (i.e. mass erase, program, verify, etc.) These command must be sent within a **`TPSTART`** / **`TPEND`** command block.

Freescale S12/S12X-specific target programming commands are the following:

- **`TPCMD UNSECURE`**;
- **`TPCMD BLANKCHECK`**;
- **`TPCMD MASSERASE`**;
- **`TPCMD PROGRAM`**;
- **`TPCMD VERIFY`**;
- **`TPCMD RUN`**.

TPCMD UNSECURE

Command syntax:

`TPCMD UNSECURE`

Command parameters:

None.

Description:

This command unsecures the target device. It erases all of the Flash and EEPROM contents. It programs the value **`0xFE`** to the **`0xFF0F`** location in order to unsecure the device.

TPCMD BLANKCHECK

Command syntax:

`TPCMD BLANKCHECK F|E <tgt start addr> <len> P|G`

Command parameters and options:

`F|E`:	Specifies Flash (**`F`**) or EEPROM (**`E`**) memory.
`tgt start address`:	Device memory location from where the blankcheck operation will start.
`len`:	Number of locations to be blankchecked.
`P|G`:	Target addressing type: Paged (**`P`**) or Global (**`G`**).

Description:

Blankchecks Flash or EEPROM memory. Blankchecks **`len`** locations starting from the address specified by **`tgt start address`**.

The **`tgt start address`** and (**`tgt start address`** + **`len`** – 1) locations must be in the same page.

If the Paged (**`P`**) addressing mode is specified, **`tgt start address`** is in the <page number><page offset> form, where <page offset> is in the **`8000`** – **`BFFF`** range.

If the Global (**`G`**) addressing mode is specified, **`tgt start address`** is in linear form.

Examples of Paged and Global addressing modes for various devices are shown in the Appendix.

TPCMD MASSERASE

Command syntax:

`TPCMD MASSERASE F|E`

Command options:

`F|E`:	Specifies Flash (**`F`**) or EEPROM (**`E`**) memory.

Description:

Mass erases Flash or EEPROM memory. If you want to be able to mass erase secured devices, they must be unsecured first (through the **`TPCMD UNSECURE`** command).

TPCMD PROGRAM

Command syntax:

```
TPCMD PROGRAM  F|E <src offset> <tgt start addr> <len>
               B|L P|G
```

Command parameters and options:

F|E:	Specifies Flash (**F**) or EEPROM (**E**) memory.
src offset:	Offset from the beginning of the source memory.
tgt start addr:	Device memory location from where the program operation will start.
len:	Number of locations to be programmed.
B|L:	Source addressing type: Banked (**B**) or Linear (**L**).
P|G:	Target addressing type: Paged (**P**) or Global (**G**).

Description:

Programs **len** locations of Flash or EEPROM memory starting from the **tgt start addr** address.

The **tgt start address** and (**tgt start address + len** – 1) locations must be in the same page.

If the Paged (**P**) addressing mode is specified, **tgt start address** is in the <page number><page offset> form, where <page offset> is in the **8000** – **BFFF** range.

If the Global (**G**) addressing mode is specified, **tgt start address** is in linear form.

If the Banked (**B**) source addressing mode is specified, **src offset** is in the <page number><page offset> form, where <page offset> is in the **8000** – **BFFF** range.

If the Linear (**L**) source addressing mode is specified, **src offset** is in linear form.

Examples of Paged and Global addressing modes for various devices are shown in the Appendix.

TPCMD VERIFY

Command syntax:

```
TPCMD VERIFY  F|E R|S <src offset> <tgt start addr>
              <len> B|L P|G
```

Command parameters and options:

F|E:	Specifies Flash (**F**) or EEPROM (**E**) memory.
R|S:	Specifies the verifying method. Choose R for a slower, but safer method that reads back all the written data. Choose S for a faster, but less safe method that only compares the checksum.
src offset:	Offset from the beginning of the source memory.
tgt start addr:	Device memory location from where the verify operation will start.
len:	Number of locations to be verified.
B|L:	Source addressing type: Banked (**B**) or Linear (**L**).
P|G:	Target addressing type: Paged (**P**) or Global (**G**).

Description:

Verifies **len** locations of Flash or EEPROM memory starting from the **tgt start addr** address. The verify operation is carried out by actually reading back all data from the device.

The **tgt start address** and (**tgt start address** + **len** – 1) locations must be in the same page.

If the Paged (**P**) addressing mode is specified, **tgt start address** is in the <page number><page offset> form, where <page offset> is in the **8000** – **BFFF** range.

If the Global (**G**) addressing mode is specified, **tgt start address** is in linear form.

If the Banked (**B**) source addressing mode is specified, **src offset** is in the <page number><page offset> form, where <page offset> is in the **8000** – **BFFF** range.

If the Linear (**L**) source addressing mode is specified, **src offset** is in linear form.

Examples of Paged and Global addressing modes for various devices are shown in the Appendix.

TPCMD RUN

Command syntax:

`TPCMD RUN`

Command parameters:

None.

Description:

Runs the target application.

Speeding up programming

*The memory on S12 and S12X family devices can be mapped by non-consecutive (banked) address ranges. On the other hand, the FlashRunner binary file format (frb) consists of consecutive (linear) addresses. When the source file to be programmed is of the banked type, it can be linearized using a Freescale utility (**`SRecCvt`**). After this operation, the resulting S19 file can be converted to the frb format using the FlashRunner **`fr_bin2frb`** utility. This final frb file is usually smaller than a frb file generated directly from a banked source file, and in turn this translates into shorter programming times.*

*On the FlashRunner CD you can find a MS-DOS batch file (**`fr_s12x2frb`**) which automatically launches the **`SRecCvt`** and **`fr_bin2frb`** utilities to generate a linearized frb file.*

*The syntax of the **`fr_s12x2frb`** utility is the following:*

`fr_s12x2frb S12|S12X F|E B|L <memory size> <input file> <output file>`

where:

*- **`S12`** or **`S12X`** indicates the target microcontroller family;*
*- **`F`** or **`E`** indicates the memory type (Flash or EEPROM);*
*- **`B`** or **`L`** indicates the source file addressing type (Banked or Linear);*
*- **`<memory size>`** is the size (in KB) of the specified target Flash or EEPROM memory;*
*- **`<input file>`** is the S19 source file to be linearized;*
*- **`<output file>`** is the resulting linearized file.*

*The address range of the resulting linearized file is **`0`** to **`<memory size> - 1`**.*

Example of conversion from a banked S12 S19 file to a linearized FlashRunner binary file:

`fr_s12x2frb S12 F B 512 banked.s19 linear.frb`

Example of conversion from a banked Flash S12X S19 file to a linearized FlashRunner binary file:

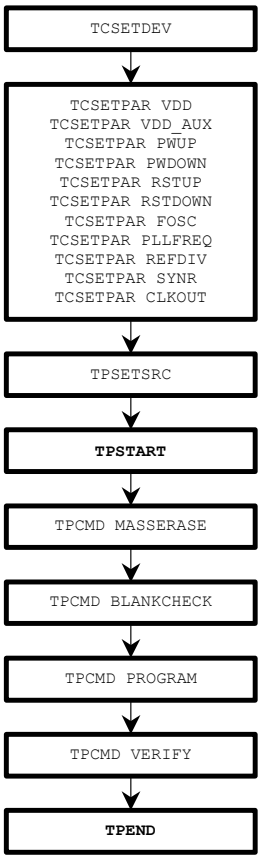
`fr_s12x2frb S12X F B 512 banked_F.s19 linear.frb`

Example of conversion from a banked EEPROM S12X S19 file to a linearized FlashRunner binary file:

`fr_s12x2frb S12X E B 4 banked_E.s19 linear.frb`

5. Typical Programming Flow

The following flow chart illustrates typical steps to help you write your own script file.



6. Script Examples

The example below shows a typical programming flow for a Freescale MC9S12E128 device.

```
;
; FLASHRUNNER SCRIPT EXAMPLE FOR FREESCALE MC9S12E128
;
; Use this example as a starting point for your specific programming needs
;
; ----------
;
; Hardware connections
;
; DIO0/AO0 (RESET)
; DIO2     (BKGD)
; CLKOUT   (CLOCK - optional)
;

; Turns off logging
#LOG_OFF
; Halt on errors
#HALT_ON FAIL

; Sets device
TCSETDEV FREESCALE MC9S12E128 HCS12

;-------------------------
;FLASHRUNNER I/O Settings
;-------------------------

; Target voltage, mV (change as needed)
TCSETPAR VDD 5000

; VPROG1 voltage, mV (from 3000 to 14500, 0 to disable)(change as needed)
TCSETPAR VDD_AUX 0

; Clock oscillator frequency driven by FlashRunner, Hz
; The possible frequencies are: 25000000, 12500000, 6250000, 0 (DISABLED)
TCSETPAR CLKOUT 0

; RESET down time (from 0 us to 65535 us)
TCSETPAR RSTDOWN 10
; RESET up time (from 0 us to 65535 us)
TCSETPAR RSTUP 100

; Power down time (from 0 ms to 65535 ms)
TCSETPAR PWDOWN 10
; Power up time (from 0 ms to 65535 ms)
TCSETPAR PWUP 10

;--------------------
;HCS12 ALGO Settings
;--------------------

; External clock source frequency, Hz (change as needed)
; For this device the maximum FOSC is 50000000 Hz
TCSETPAR FOSC 16000000

; Enables and sets the internal PLL frequency, Hz
; Use the following formula to calculate the PLL parameter:
;      PLLFREQ = 2*FOSC*(SYNR+1)/(REFDIV+1)
; For this device the maximum PLLFREQ is 50000000 Hz
; Set all parameters to 0 to disable the internal PLL frequency

TCSETPAR PLLFREQ 0
TCSETPAR REFDIV 0
TCSETPAR SYNR 0

;---------------------------
;Start Programming operation
;---------------------------

; Starts programming block
TPSTART

; Image file to be programmed (must be placed in the \BINARIES directory)
TPSETSRC FILE TEST.FRB

; Chip unsecuring. Enable this command if the device is protected
```

```
; Note: This command erases Flash memory and EEPROM memory
TPCMD UNSECURE

;--------------
;FLASH commands
;--------------

; Mass erases Flash memory
TPCMD MASSERASE F

; Blank checks Flash memory (change address and length as needed)
; Paged FLASH: 0x8000 TO 0xBFFF addressed through PPAGE.
TPCMD BLANKCHECK F $388000 $4000 P
TPCMD BLANKCHECK F $398000 $4000 P
TPCMD BLANKCHECK F $3A8000 $4000 P
TPCMD BLANKCHECK F $3B8000 $4000 P
TPCMD BLANKCHECK F $3C8000 $4000 P
TPCMD BLANKCHECK F $3D8000 $4000 P
TPCMD BLANKCHECK F $3E8000 $4000 P
TPCMD BLANKCHECK F $3F8000 $4000 P

; Programs Flash memory (change source and target address and length as needed)
; Paged FLASH: 0x8000 to 0xBFFF addressed through PPAGE.
TPCMD PROGRAM F $388000 $388000 $4000 B P
TPCMD PROGRAM F $398000 $398000 $4000 B P
TPCMD PROGRAM F $3A8000 $3A8000 $4000 B P
TPCMD PROGRAM F $3B8000 $3B8000 $4000 B P
TPCMD PROGRAM F $3C8000 $3C8000 $4000 B P
TPCMD PROGRAM F $3D8000 $3D8000 $4000 B P
TPCMD PROGRAM F $3E8000 $3E8000 $4000 B P
TPCMD PROGRAM F $3F8000 $3F8000 $4000 B P

; Verifies Flash memory, read-out method (change source and target address and length as needed)
; Paged FLASH: 0x8000 to 0xBFFF addressed through PPAGE.
TPCMD VERIFY F R $388000 $388000 $4000 B P
TPCMD VERIFY F R $398000 $398000 $4000 B P
TPCMD VERIFY F R $3A8000 $3A8000 $4000 B P
TPCMD VERIFY F R $3B8000 $3B8000 $4000 B P
TPCMD VERIFY F R $3C8000 $3C8000 $4000 B P
TPCMD VERIFY F R $3D8000 $3D8000 $4000 B P
TPCMD VERIFY F R $3E8000 $3E8000 $4000 B P
TPCMD VERIFY F R $3F8000 $3F8000 $4000 B P

; Ends programming block
TPEND
```

The example below shows a typical programming flow for a Freescale MC9S12XDP512 device.

```
;
; FLASHRUNNER SCRIPT EXAMPLE FOR FREESCALE MC9S12XDP512
;
; Use this example as a starting point for your specific programming needs
;
; ----------
;
; Hardware connections
;
; DIO0/AO0 (RESET)
; DIO2     (BKGD)
; CLKOUT   (CLOCK - optional)
;

; Turns off logging
#LOG_OFF
; Halt on errors
#HALT_ON FAIL

; Sets device
TCSETDEV FREESCALE MC9S12XDP512 HCS12

;------------------------
;FLASHRUNNER I/O Settings
;------------------------

; Target voltage, mV (change as needed)
TCSETPAR VDD 5000

; VPROG1 voltage, mV (from 3000 to 14500, 0 to disable)(change as needed)
```

```
TCSETPAR VDD_AUX 0

; Clock oscillator frequency driven by FlashRunner, Hz
; The possible frequencies are: 25000000, 12500000, 6250000, 0 (DISABLED)
TCSETPAR CLKOUT 0

; RESET down time (from 0 us to 65535 us)
TCSETPAR RSTDOWN 10
; RESET up time (from 0 us to 65535 us)
; Note: the Reset up time must be shorter than the COP module timeout.
TCSETPAR RSTUP 10

; Power down time (from 0 ms to 65535 ms)
TCSETPAR PWDOWN 10
; Power up time (from 0 ms to 65535 ms)
TCSETPAR PWUP 10

;--------------------
;HCS12 ALGO Settings
;--------------------

; External clock source frequency, Hz (change as needed)
; For this device the maximum FOSC is 80000000 Hz
TCSETPAR FOSC 16000000

; Enables and sets the internal PLL frequency, Hz
; Use the following formula to calculate the PLL parameter:
;      PLLFREQ = 2*FOSC*(SYNR+1)/(REFDIV+1)
; For this device the maximum PLLFREQ is 50000000 Hz
; Set all parameters to 0 to disable the internal PLL frequency

TCSETPAR PLLFREQ 0
TCSETPAR REFDIV 0
TCSETPAR SYNR 0

;---------------------------
;Start Programming operation
;---------------------------

; Starts programming block
TPSTART

; Image file to be programmed (must be placed in the \BINARIES directory)
TPSETSRC FILE TEST.FRB

; Chip unsecuring. Enable this command if the device is protected
; Note: This command erases Flash memory and EEPROM memory
TPCMD UNSECURE

;--------------
;FLASH commands
;--------------

; Mass erases Flash memory
TPCMD MASSERASE F

; Blank checks Flash memory (change address and length as needed)
; Paged FLASH: 0x8000 to 0xBFFF addressed through PPAGE.
TPCMD BLANKCHECK F $E08000 $4000 P
TPCMD BLANKCHECK F $E18000 $4000 P
TPCMD BLANKCHECK F $E28000 $4000 P
TPCMD BLANKCHECK F $E38000 $4000 P
TPCMD BLANKCHECK F $E48000 $4000 P
TPCMD BLANKCHECK F $E58000 $4000 P
TPCMD BLANKCHECK F $E68000 $4000 P
TPCMD BLANKCHECK F $E78000 $4000 P
TPCMD BLANKCHECK F $E88000 $4000 P
TPCMD BLANKCHECK F $E98000 $4000 P
TPCMD BLANKCHECK F $EA8000 $4000 P
TPCMD BLANKCHECK F $EB8000 $4000 P
TPCMD BLANKCHECK F $EC8000 $4000 P
TPCMD BLANKCHECK F $ED8000 $4000 P
TPCMD BLANKCHECK F $EE8000 $4000 P
TPCMD BLANKCHECK F $EF8000 $4000 P
TPCMD BLANKCHECK F $F08000 $4000 P
TPCMD BLANKCHECK F $F18000 $4000 P
TPCMD BLANKCHECK F $F28000 $4000 P
TPCMD BLANKCHECK F $F38000 $4000 P
TPCMD BLANKCHECK F $F48000 $4000 P
TPCMD BLANKCHECK F $F58000 $4000 P
TPCMD BLANKCHECK F $F68000 $4000 P
TPCMD BLANKCHECK F $F78000 $4000 P
TPCMD BLANKCHECK F $F88000 $4000 P
TPCMD BLANKCHECK F $F98000 $4000 P
TPCMD BLANKCHECK F $FA8000 $4000 P
```

```
TPCMD BLANKCHECK F $FB8000 $4000 P
TPCMD BLANKCHECK F $FC8000 $4000 P
TPCMD BLANKCHECK F $FD8000 $4000 P
TPCMD BLANKCHECK F $FE8000 $4000 P
TPCMD BLANKCHECK F $FF8000 $4000 P

; Programs Flash memory (change source and target address and length as needed)
; Paged FLASH: 0x8000 to 0xBFFF addressed through PPAGE.
TPCMD PROGRAM F $E08000 $E08000 $4000 B P
TPCMD PROGRAM F $E18000 $E18000 $4000 B P
TPCMD PROGRAM F $E28000 $E28000 $4000 B P
TPCMD PROGRAM F $E38000 $E38000 $4000 B P
TPCMD PROGRAM F $E48000 $E48000 $4000 B P
TPCMD PROGRAM F $E58000 $E58000 $4000 B P
TPCMD PROGRAM F $E68000 $E68000 $4000 B P
TPCMD PROGRAM F $E78000 $E78000 $4000 B P
TPCMD PROGRAM F $E88000 $E88000 $4000 B P
TPCMD PROGRAM F $E98000 $E98000 $4000 B P
TPCMD PROGRAM F $EA8000 $EA8000 $4000 B P
TPCMD PROGRAM F $EB8000 $EB8000 $4000 B P
TPCMD PROGRAM F $EC8000 $EC8000 $4000 B P
TPCMD PROGRAM F $ED8000 $ED8000 $4000 B P
TPCMD PROGRAM F $EE8000 $EE8000 $4000 B P
TPCMD PROGRAM F $EF8000 $EF8000 $4000 B P
TPCMD PROGRAM F $F08000 $F08000 $4000 B P
TPCMD PROGRAM F $F18000 $F18000 $4000 B P
TPCMD PROGRAM F $F28000 $F28000 $4000 B P
TPCMD PROGRAM F $F38000 $F38000 $4000 B P
TPCMD PROGRAM F $F48000 $F48000 $4000 B P
TPCMD PROGRAM F $F58000 $F58000 $4000 B P
TPCMD PROGRAM F $F68000 $F68000 $4000 B P
TPCMD PROGRAM F $F78000 $F78000 $4000 B P
TPCMD PROGRAM F $F88000 $F88000 $4000 B P
TPCMD PROGRAM F $F98000 $F98000 $4000 B P
TPCMD PROGRAM F $FA8000 $FA8000 $4000 B P
TPCMD PROGRAM F $FB8000 $FB8000 $4000 B P
TPCMD PROGRAM F $FC8000 $FC8000 $4000 B P
TPCMD PROGRAM F $FD8000 $FD8000 $4000 B P
TPCMD PROGRAM F $FE8000 $FE8000 $4000 B P
TPCMD PROGRAM F $FF8000 $FF8000 $4000 B P

; Verifies Flash memory, read-out method (change source and target address and length as needed)
; Paged FLASH: 0x8000 to 0xBFFF addressed through PPAGE.
TPCMD VERIFY F R $E08000 $E08000 $4000 B P
TPCMD VERIFY F R $E18000 $E18000 $4000 B P
TPCMD VERIFY F R $E28000 $E28000 $4000 B P
TPCMD VERIFY F R $E38000 $E38000 $4000 B P
TPCMD VERIFY F R $E48000 $E48000 $4000 B P
TPCMD VERIFY F R $E58000 $E58000 $4000 B P
TPCMD VERIFY F R $E68000 $E68000 $4000 B P
TPCMD VERIFY F R $E78000 $E78000 $4000 B P
TPCMD VERIFY F R $E88000 $E88000 $4000 B P
TPCMD VERIFY F R $E98000 $E98000 $4000 B P
TPCMD VERIFY F R $EA8000 $EA8000 $4000 B P
TPCMD VERIFY F R $EB8000 $EB8000 $4000 B P
TPCMD VERIFY F R $EC8000 $EC8000 $4000 B P
TPCMD VERIFY F R $ED8000 $ED8000 $4000 B P
TPCMD VERIFY F R $EE8000 $EE8000 $4000 B P
TPCMD VERIFY F R $EF8000 $EF8000 $4000 B P
TPCMD VERIFY F R $F08000 $F08000 $4000 B P
TPCMD VERIFY F R $F18000 $F18000 $4000 B P
TPCMD VERIFY F R $F28000 $F28000 $4000 B P
TPCMD VERIFY F R $F38000 $F38000 $4000 B P
TPCMD VERIFY F R $F48000 $F48000 $4000 B P
TPCMD VERIFY F R $F58000 $F58000 $4000 B P
TPCMD VERIFY F R $F68000 $F68000 $4000 B P
TPCMD VERIFY F R $F78000 $F78000 $4000 B P
TPCMD VERIFY F R $F88000 $F88000 $4000 B P
TPCMD VERIFY F R $F98000 $F98000 $4000 B P
TPCMD VERIFY F R $FA8000 $FA8000 $4000 B P
TPCMD VERIFY F R $FB8000 $FB8000 $4000 B P
TPCMD VERIFY F R $FC8000 $FC8000 $4000 B P
TPCMD VERIFY F R $FD8000 $FD8000 $4000 B P
TPCMD VERIFY F R $FE8000 $FE8000 $4000 B P
TPCMD VERIFY F R $FF8000 $FF8000 $4000 B P

;----------------
; EEPROM Commands
;----------------
; Mass erases EEPROM memory
TPCMD MASSERASE E

; Blank checks EEPROM memory (change address and length as needed)
; Paged EEPROM: 0x800 to 0xBFF addressed through EPAGE.
TPCMD BLANKCHECK E $FC0800 $400 P
```

```
TPCMD BLANKCHECK E $FD0800 $400 P
TPCMD BLANKCHECK E $FE0800 $400 P
TPCMD BLANKCHECK E $FF0800 $400 P

; Programs EEPROM memory (change address and length as needed)
; Paged EEPROM: 0x800 to 0xBFF addressed through EPAGE.
TPCMD PROGRAM E $FC0800 $FC0800 $400 B P
TPCMD PROGRAM E $FD0800 $FD0800 $400 B P
TPCMD PROGRAM E $FE0800 $FE0800 $400 B P
TPCMD PROGRAM E $FF0800 $FF0800 $400 B P

; Verifies EEPROM memory, read-out method (change address and length as needed)
; Paged EEPROM: 0x800 to 0xBFF addressed through EPAGE.
TPCMD VERIFY E R $FC0800 $FC0800 $400 B P
TPCMD VERIFY E R $FD0800 $FD0800 $400 B P
TPCMD VERIFY E R $FE0800 $FE0800 $400 B P
TPCMD VERIFY E R $FF0800 $FF0800 $400 B P

; Ends programming block
TPEND
```

The FlashRunner's system software setup will install script examples specific for each device of the S12/S12X family on your PC.

7. Programming Times

The following table shows programming times for selected Freescale S12/S12X family devices.

Device	Mem. Size	PLL Freq.	Operations	Time
MC9S12A128B	128KB Flash	50 MHz	Program	5.95 s
MC9S12A128B	128KB Flash	50 MHz	Erase + Unsecure + Blank Check + Program + Verify	10.00 s
MC9S12XDP512	512KB Flash	50 MHz	Program	14.99 s
MC9S12XDP512	512KB Flash	50 MHz	Erase + Program + Verify	30.62 s

Programming times depend on Programming Algorithm version, target board connections, communication mode, target microcontroller mask, and other conditions. Programming times for your actual system may therefore be different than the ones listed here. SofTec Microsystems reserves the right to modify Programming Algorithms at any time.

8. References

FlashRunner user's manual

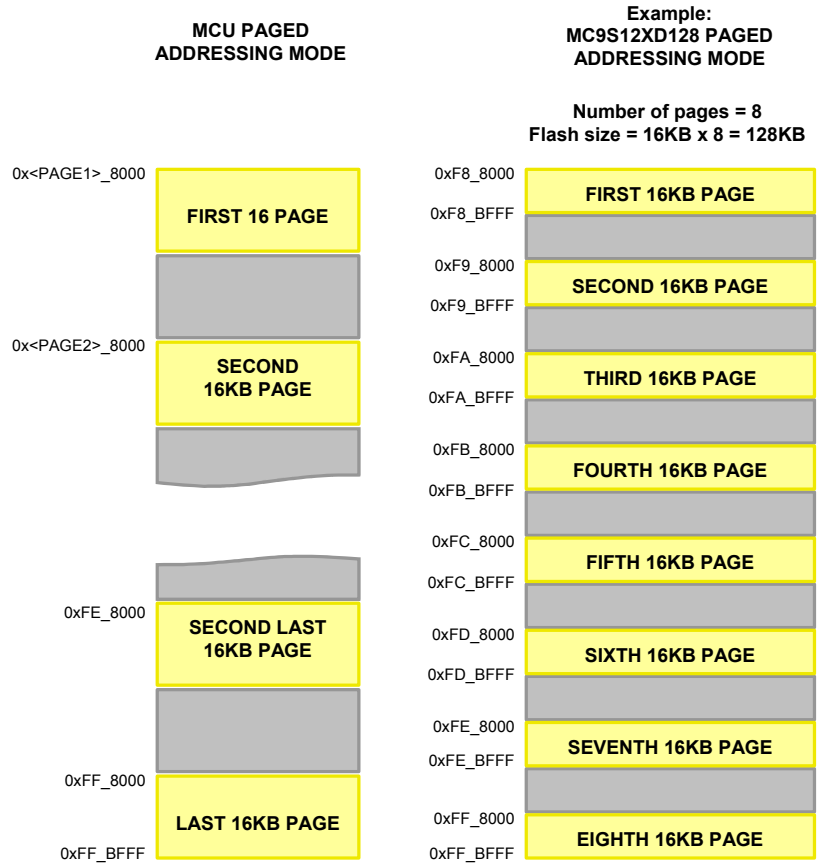
Microcontroller-specific datasheets

9. Appendix: Example of Addressing Modes

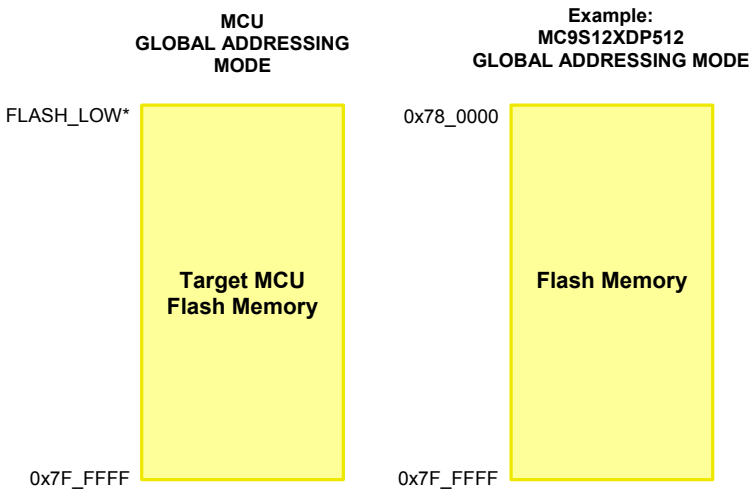
S12X MCUs (128KB and 512KB Flash Memory)

Paged Addressing Mode

Pages	PPAGE Address
Page 1	0x100 - number of pages
...	...
Second Last Page	Last Page - 1
Last Page	0x100 - 1 = 0xFF



Global Addressing Mode

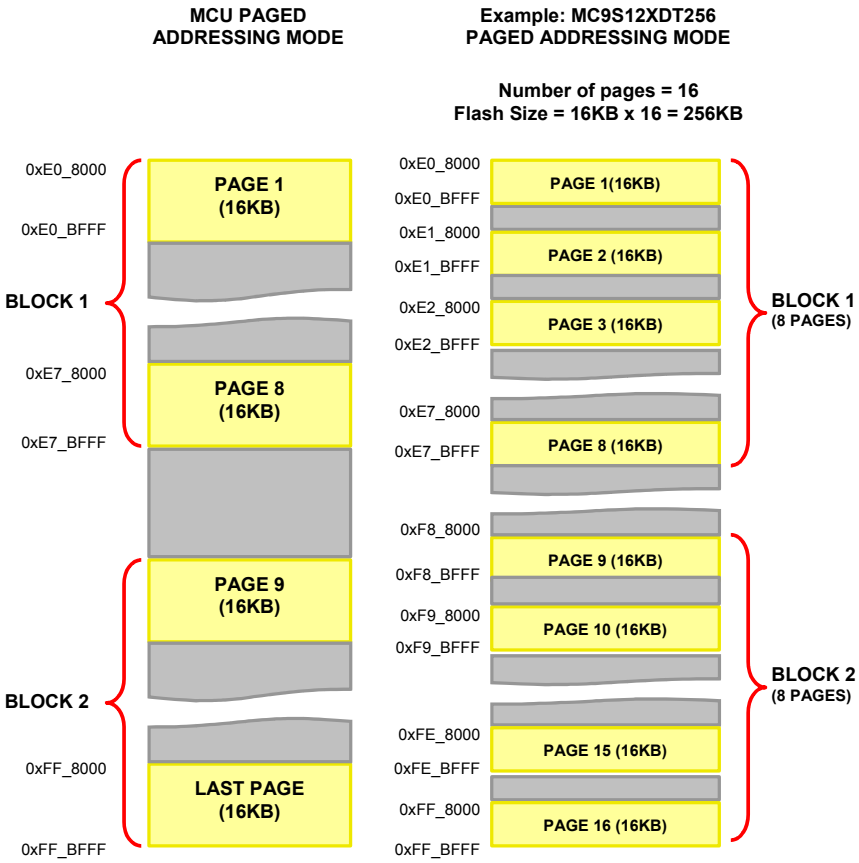


* The FLASH_LOW value is specified in the device datasheet.

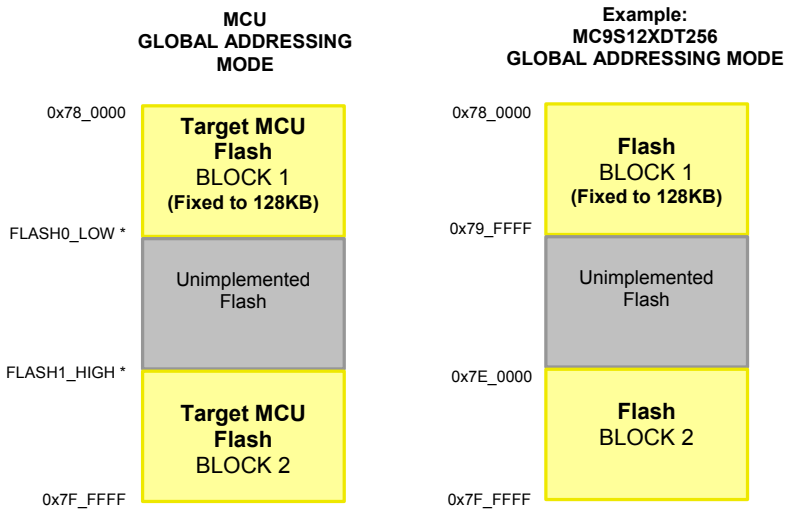
S12X MCUs (256KB and 384KB Flash Memory)

Paged Addressing Mode

Pages	PPAGE Address
Page 1 Block 1	0xE0
Page 2 Block 1	0xE1
...	...
Page 8 Block 1	0xE7
Page 9 Block 2	0x100 - number of pages + 8
...	...
Last Page Block 2	0x100 - 1 = 0xFF



Global Addressing Mode



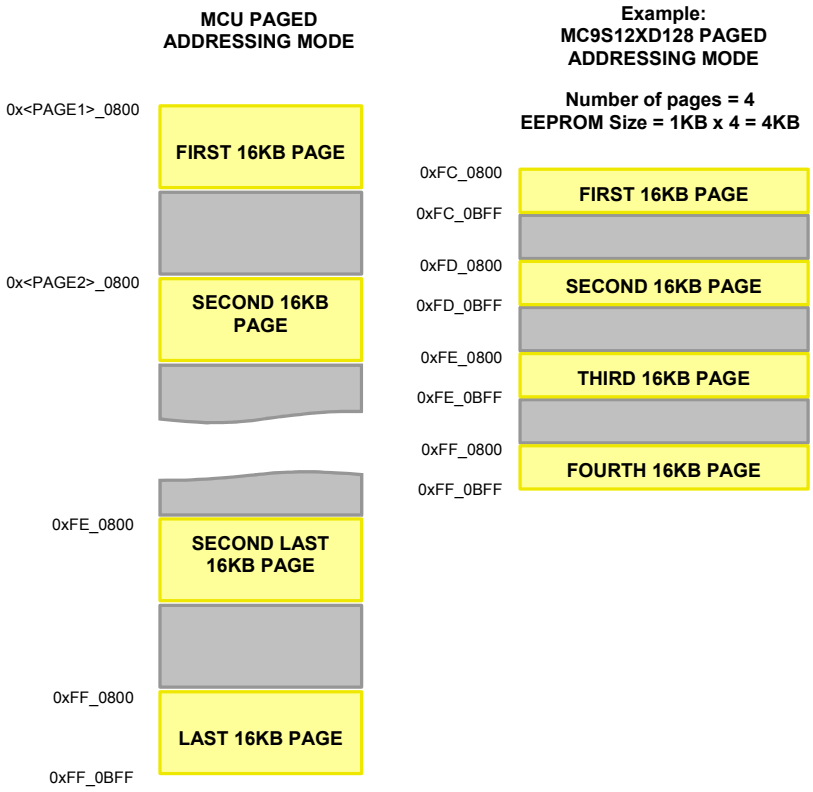
* FLASH0_LOW and FLASH1_HIGH values are specified in the device datasheet.

S12X MCUs (EEPROM Memory)

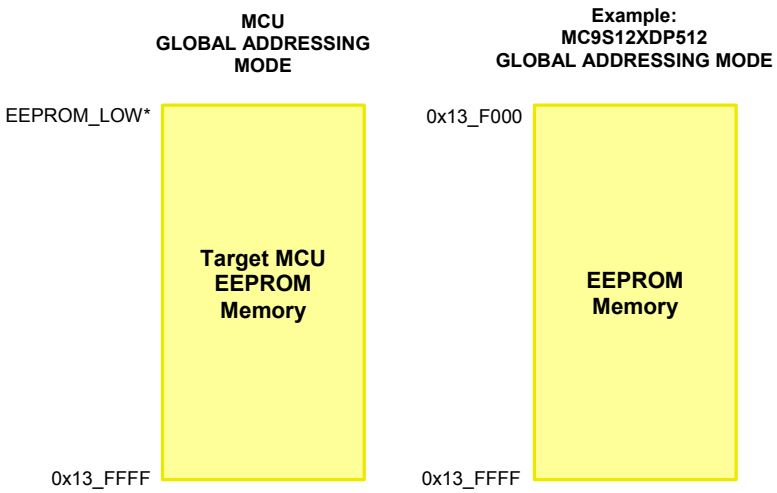
Paged Addressing Mode

The EEPROM memory of S12X MCUs is set in pages of 1KB from address 0x0800 to 0x0BFF.

Pages	PPAGE Address
Page 1	0x100 - number of pages
...	...
Second Last Page	Last Page - 1
Last Page	0x100 - 1 = 0xFF



Global Addressing Mode



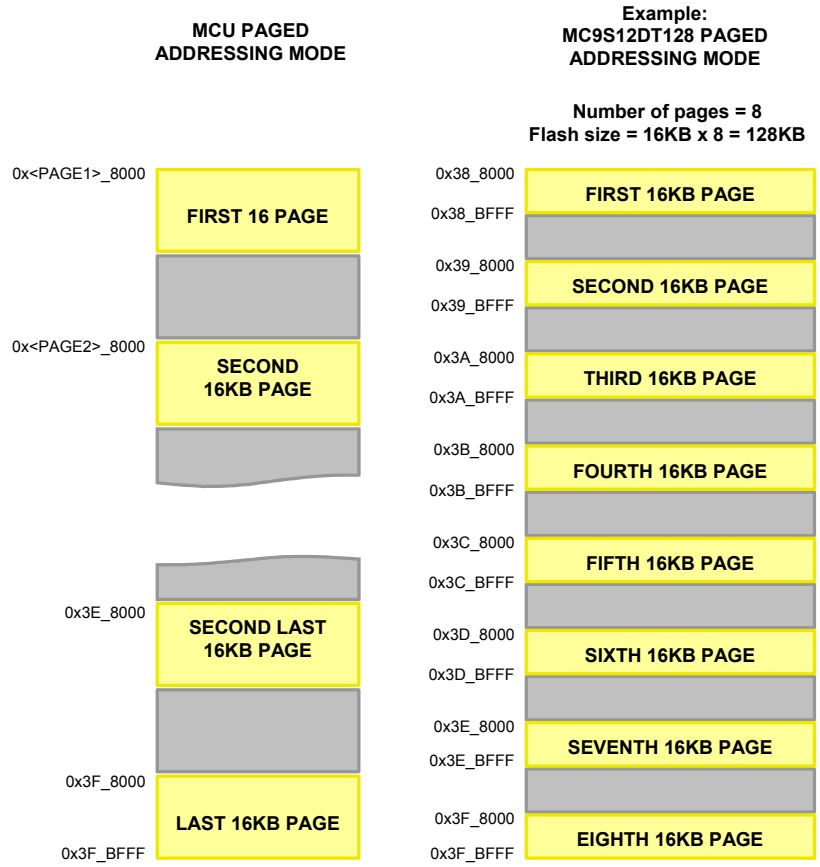
* The EEPROM_LOW value is specified in the device datasheet.

S12 MCUs (Flash Memory)

Paged Addressing Mode

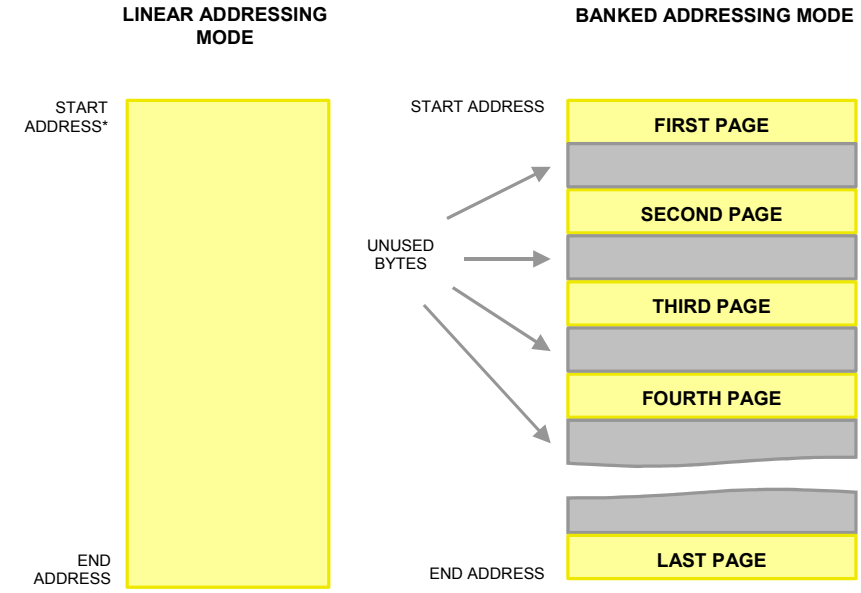
S12 MCUs can be addressed in PAGED mode only.

Pages	PPAGE Address
Page 1	0x40 - number of pages
...	...
Second Last Page	Last Page - 1
Last Page	0x40 - 1 = 0x3F



Note: *The EEPROM memory of S12 devices can be mapped anywhere. By default, FlashRunner maps it to 0x0000.*

FlashRunner Binary File Types for S12 and S12X MCUs



* This address should correspond to the address that contains the first Flash or EEPROM data of your code. Use the Offset parameter in the FRB conversion, to set it.