Application Note

Report No:

AN122

Title:

In-System Programming (ISP) of Atmel AT91SAM7 FLASH Microcontrollers using the JTAG Programming Interface

Author:	Date:	Version Number:
John Marriott	25th January 2010	1.13

High-speed JTAG ISP Programmers...
PPM4
EPSILON
ATMEL
AT91SAM7
FS2009
ISP nano
...designed for Production Programming

Contents

1.0 Introduction

Equinox Technologies manufacture a range of programmers suitable for high-speed programming of Atmel AT91SAM7 FLASH microcontrollers via the ***'ARM JTAG Debug Interface'***. This application note describes how to develop and implement ***In-System Programming (ISP)*** support for the Atmel AT91SAM7 FLASH microcontroller families using the ***'ARM JTAG Debug Interface'***. The document details how to make a JTAG ***'Programming Project'*** which will operate on any Equinox ISP programmer including a full description of how to implement JTAG In-System Programming (ISP) of the Atmel AT91SAM7S, AT91SAM7SE, AT91SAM7X and AT91SAM7L families.

1.1 Features

The Equinox programming range includes solutions for development, low / mid / high volume production and field programming of Atmel AT91SAM7 microcontrollers.

General features…..

- High-speed In-System Programming (ISP) support of Atmel AT91SAM7 microcontrollers
- Programming solutions for development, low / mid / high volume production and field programming of Atmel AT91SAM7 microcontrollers
- Programs the on-chip FLASH Memory of AT91SAM7 microcontrollers
- Uses industry standard ***'ARM JTAG Debug Interface'*** port as the ISP interface
- Very high-speed programming due to local data storage and optimised programming algorithms
- Programmers can be used in ***"Standalone Mode"*** (no PC required)
- Supports high-speed program / verify of the on-chip FLASH in a singe operation.
- Optimised Erase operations - blanks the on-chip FLASH in less than 200ms
- Supports programming of non volatile ***'Fuse Bits'***
- Supports blowing of the 'Security Fuse' to protect code from being read out
- Supports reading back / logging of all ***'FLASH Page Lock Bits'***
- Supports programming of AT91SAM7 devices when placed in a ***'JTAG Chain'*** (JTAG daisy-chain mode)

In 'Development Mode'…..

- Powerful yet simple-to-use Development Suite called ***'EDS'***
- All aspects of programming the AT91SAM7 device can be controlled from ***EDS***

In 'Production Mode'…..

- Programmers can be used in ***"Standalone Mode"*** (no PC required)
- A single ***'Standalone Programming Project'*** can Erase the device, program /verify the FLASH, program the Fuse Bits and blow the ***'Security Fuse'*** in a single operation.
- Up to 64 x AT91SAM7S ***'Standalone Programming Projects'*** can be stored inside an FS2003, FS2009, PPM3-MK2, PPM4-MK1 or ISPnano programmer.
- Programmer can store multiple versions of firmware for different 'customer product versions'.

1.2 Programmers supporting AT91SAM7 devices

1.2.1 Complete AT91SAM7 Programming Solutions

The table below lists the off-the-shelf AT91SAM7 programming systems available from Equinox….

Order Code	Description
EPSILON5(SAM7)	• Portable high-speed In-System (ISP) Programmer for JTAG programming of Atmel AT91SAM7 ARM7 microcontroller family. • Standalone capability (1 project) • USB / RS232 connectivity • Includes AT91SAM7 ISP Connector Cable
EPS-AT91SAM7JTAG-BUNDLE	• Portable high-speed In-System (ISP) Programmer for JTAG programming of Atmel AT91SAM7 ARM7 microcontroller family. • Standalone capability (1 project) • This bundle pack includes a standard EPSILON5 Programmer which is enabled for SPI programming of Atmel AVR devices and also JTAG programming of Atmel AT91SAM7 microcontrollers. • USB / RS232 connectivity • Includes AT91SAM7 ISP Connector Cable
FS2009(SAM7)	• Portable high-speed In-System (ISP) Programmer for JTAG programming of Atmel AT91SAM7 ARM7 microcontroller family. • Standalone capability using LCD / keypad (64 projects) • USB / RS232 connectivity • Includes AT91SAM7 ISP Connector Cable

Please note:

- The AT91SAM7 Device Library is pre-enabled on each programmer.
- The AT91SAM7 ISP Connector Cable is included with each system.

1.2.2 Upgrading an existing programmer for AT91SAM7 support

Most Equinox ISP Programmers can be upgraded to support high-speed programming of Atmel AT91SAM7 microcontrollers via the ***'ARM JTAG Debug Interface'*** .The table below lists all the Equinox ISP programmers which are capable of programming AT91SAM7 microcontrollers.

Programmer	AT91SAM7 Support	Requirements	Upgrade Order Code
EPSILON5(UN)	Upgrade	License upgrade + CAB-ARM7-20W ISP Cable	EPSILON5-UPG15
FS2003(UN)	Upgrade	License upgrade + CAB-ARM7-20W ISP Cable	FS2003-UPG15
FS2009(UN)	Upgrade	License upgrade + CAB-ARM7-20W ISP Cable	FS2009-UPG15
PPM3-MK2	Upgrade	License upgrade + EQ-IO-CON-10 Connector Module + EQ-SFM-MAX-V1.3 High-speed I/O Driver Module	PPM3A1-UPG15
PPM4-MK1	Upgrade	License upgrade + EQ-IO-CON-10 Connector Module + EQ-SFM-MAX-V1.3 High-speed I/O Driver Module	PPM4MK1-UPG15
ISPNano Series 1	Upgrade	License upgrade	ISPNano-UPG15
ISPNano Series 2	Upgrade	License upgrade	ISPNano-UPG15
ISPNano Series 3	Upgrade	License upgrade	ISPNano-UPG15
ISPNano Series 3 ATE	Upgrade	License upgrade	ISPNano-UPG15

Please note:

- A chargeable ***'License Upgrade'*** is required to enable the AT91SAM7 support on any of these programmers.
- A suitable ***'AT91SAM7 ISP Connector Cable'*** is also required to connect from the programmer to the AT91SAM7 Target Board. This is provided with the License Upgrade.
- The ***programmer firmware*** will probably also need to be upgraded in order to support AT91SAM7 device programming – see section 1.3.
- It is also recommended that EQTools build 1156 or above is used when programming AT91SAM7 devices.

1.3 Programmer firmware versions for AT91SAM7 support

Most Equinox ISP Programmers can be upgraded to support high-speed programming of Atmel AT91SAM7 microcontrollers via the ***'ARM JTAG Debug Interface'*** .The table below lists all the Equinox ISP programmers which are capable of programming AT91SAM7 microcontrollers. A chargeable ***'License Upgrade'*** is required to enable the AT91SAM7 support on any of these programmers. A suitable ***'AT91SAM7 ISP Connector Cable'*** is also required to connect from the programmer to the AT91SAM7 Target Board. This is supplied with the License Upgrade.

Fig. 1.3 Programmer firmware versions for AT91SAM7 JTAG In-System Programming Support

Programmer	**AT91SAM7 Support**	**Requirements**
EPSILON5	5.22	Contains all algorithms
FS2003	5.22	Contains only AT91SAM7 and AVR JTAG algorithms
FS2009	5.22	Contains only AT91SAM7 and AVR JTAG algorithms
PPM3-MK2	5.21	Contains all algorithms
PPM4-MK1	5.21	Contains all algorithms
ISPnano Series I/II	5.22	Contains all algorithms
ISPnano Series III	5.22	Contains all algorithms

Please note:

- Due to limited firmware storage space on the FS2003 and PPM3-MK2 programmers, these programmers cannot support both the AT91SAM7 firmware and other microcontrollers at the same time.
- It is necessary to upload a special version of firmware for programming AT91SAM7 devices which will then not support programming of any other devices e.g. Atmel AVR microcontrollers.
- It is possible to switch between the conventional firmware and ***'AT91SAM7'*** firmware at any time by simply uploading the relevant ***'Firmware upgrade project'*** to the programmer.

1.4 Device Support

1.4.1 Overview

There are various different families of Atmel AT91SAM7 microcontrollers, each with a different performance / peripheral mix. The Device Support is now split into a different section for each different family.

1.4.2 AT91SAM7 – "S" Family

The AT91SAM7 – "S" Family contains the following devices:

Device	**On-chip FLASH Size (kb)**	**FLASH Page Size (bytes)**	**Minimum firmware version**	**Programming Support available**	**Tested on actual device**
AT91SAM7S16	16	64	4.00	YES	-
AT91SAM7S161	161	64	4.00	YES	-
AT91SAM7S32	32	128	4.00	YES	-
AT91SAM7S321	321	128	4.00	YES	-
AT91SAM7S64	64	128	4.00	YES	YES
AT91SAM7S128	128	256	4.00	YES	-
AT91SAM7S256	256	256	4.00	YES	YES
AT91SAM7S512	512	256	4.00	YES	YES

1.4.3 AT91SAM7 – "SE" Family

The AT91SAM7 – "SE" Family contains the following devices:

Device	**On-chip FLASH Size (kb)**	**FLASH Page Size (bytes)**	**Minimum firmware version**	**Programming Support available**	**Tested on actual device**
AT91SAM7SE32	32	128	4.00	YES	-
AT91SAM7SE256	256	256	4.00	YES	-
AT91SAM7SE512	512	256	4.00	YES	YES

1.4.4 AT91SAM7 – "L" (Low Power) Family

The AT91SAM7 – "L" Family contains the following devices:

Device	**On-chip FLASH Size (kb)**	**FLASH Page Size (bytes)**	**Minimum firmware version**	**Programming Support available**	**Tested on actual device**
AT91SAM7L64	64	256	4.00	-	-
AT91SAM7L128	128	256	4.00	-	-

1.4.5 AT91SAM7 – “A” Family

The AT91SAM7 – “A” Family contains the following devices:

Device	On-chip FLASH Size (kb)	FLASH Page Size (bytes)	Minimum firmware version	Programming Support available	Tested on actual device
AT91SAM7A3	256	256	4.00	YES	YES

1.4.6 AT91SAM7 – “X” Family

The AT91SAM7 – “X” Family contains the following devices:

Device	On-chip FLASH Size (kb)	FLASH Page Size (bytes)	Minimum firmware version	Programming Support available	Tested on actual device
AT91SAM7X128	128		4.00	YES	-
AT91SAM7X256	256		4.00	YES	-
AT91SAM7X512	512		4.00	YES	YES

1.4.7 AT91SAM7 – “XC” Family

The AT91SAM7 – “XC” Family contains the following devices:

Device	On-chip FLASH Size (kb)	FLASH Page Size (bytes)	Minimum firmware version	Programming Support available	Tested on actual device
AT91SAM7XC128	128		5.07	YES	-
AT91SAM7XC256	256		5.07	YES	-
AT91SAM7XC512	512		5.07	YES	-

1.5 Upgrading your Equinox Programmer to support AT91SAM7 JTAG Programming

1.5.1 Overview

The Atmel AT91SAM7 JTAG algorithms are not supported as standard on any Equinox programmers (except for the options described below*). It is necessary to purchase a ***'License Upgrade'*** for '***AT91SAM7 JTAG'*** support from Equinox. Equinox will then send you a ***'JTAG Upgrade License String'*** which will upgrade your programmer to support programming of this device family.

Please note
The following standalone programmer and ***'Bundle'*** options have the license pre-installed, therefore these instructions are not necessary:

- Epsilon5(SAM7)
- EPS-AT91SAM7JTAG-BUNDLE
- FS2009(SAM7)

1.5.2 Purchasing an AT91SAM7 License

All Equinox ISP programmers require the purchase of a ***'License Upgrade'*** to enable AT91SAM7 JTAG support. Please see the table in section 1.1 for the relevant upgrade for your programmer.

1.5.3 How do I enable the programmer for AT91SAM7 JTAG?

To enable your programmer to support JTAG ISP programming, please purchase the relevant JTAG Upgrade from Equinox or an Equinox distributor:

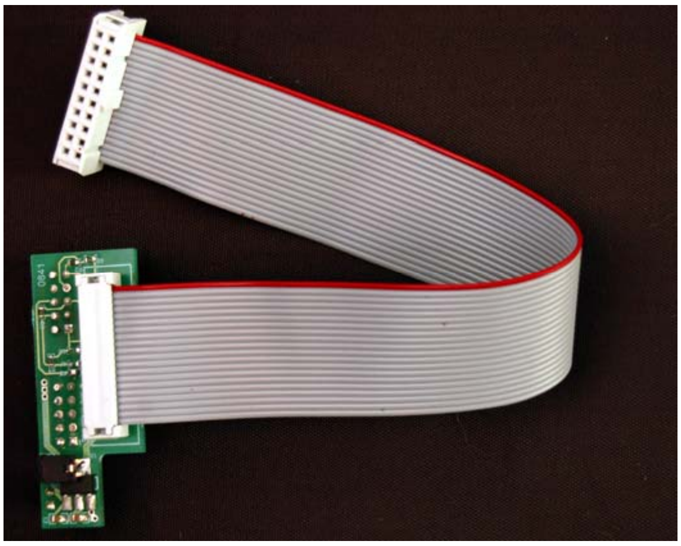
1. **If you purchase the upgrade directly from Equinox**
   - Equinox will email you a 'JTAG License String'.
   - This string can be entered directly into the ***<Enter License>*** screen in EQTools.

2. **If you purchase the upgrade from a distributor**
   - The distributor will send you the Upgrade Pack by courier.
   - Within the Upgrade Pack you will find an Upgrade Form with a Code String on it.
   - Email this Code String plus your programmer Serial Number to support@equinox-tech.com
   - Equinox will then send you a ***'AT91SAM7 JTAG License String'*** which is keyed to your programmer Serial Number.
   - This string can be entered directly into the ***<Enter License>*** screen in EQTools.

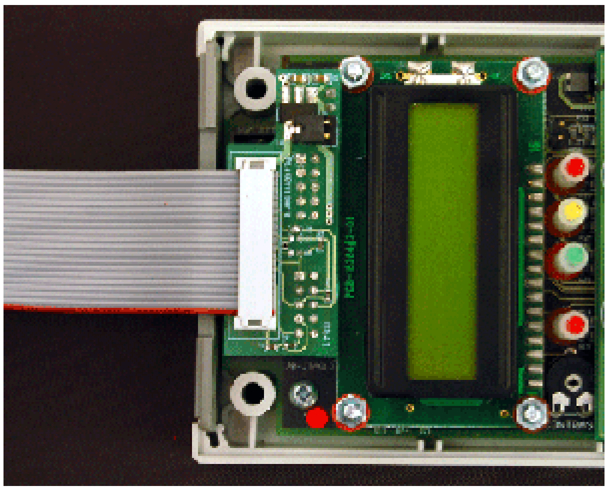
1.5.4 Upgrading an Epsilon5, FS2003 or FS2009 Programmer to support AT91SAM7 JTAG programming

To upgrade an Epsilon5, FS2003 or FS2009 programmer to support AT91SAM7 JTAG programming, please follow the steps below:

- Order an ***'AT91SAM7 JTAG License'*** from Equinox
- Enter the ***'AT91SAM7 Upgrade License String'*** given to you by Equinox into EQTools
- Make sure you have the required version of Programmer Firmware to support the device you wish to program – see section 1.2.
- You will also receive a special ***'SAM7 Programming Cable'*** with this upgrade. See Appendix 1 for full details of how to connect this cable to your programmer.



- This cable has a small circuit board on one end which plugs into the ISP headers inside the programmer. On the other end it features a 20-way IDC plug which will plug into the 20-way IDC ***"ARM Debug Connector"*** socket on any ARM Target Board.
- Remove the top cover of the programmer
- Plug the circuit board end of the ***'SAM7 Programming Cable'*** into the programmer ISP headers – see example FS2003 / FS2009 picture below.

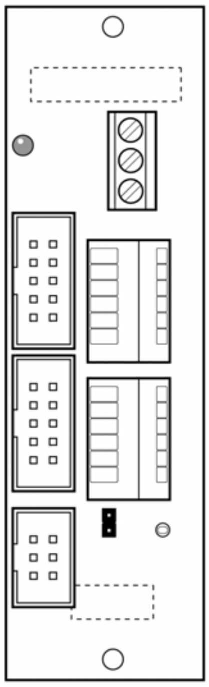


- Plug the 20-way IDC plug end of the cable into the mating IDC socket on your Target System.
- You are now ready to program an AT91SAM7 device via JTAG.

1.5.5 Upgrading a PPM3-MK2 or PPM4-MK1 Programmer to support AT91SAM JTAG

To upgrade a PPM3-MK2 or PPM4-MK1 programmer to support AT91SAM7 JTAG, please follow the steps below:

- Order a ***'AT91SAM7 JTAG upgrade'*** from Equinox Technologies
- Enter the ***'JTAG Upgrade License String'*** given to you by Equinox into EQTools
- The JTAG upgrade also includes a new ***'I/O Connector Module'*** for the PPM4-MK1 called the **'EQ-IO-CON-10'**. This module features a 20-way JTAG header for connection to the AT91SAM7 Target Board. It also includes the High Speed / High Current Special Function Module **'EQ-SFM-MAX-V1.3'.** This significantly reduces the overall programming times for many high capacity devices. For full instructions on fitting this module please refer to ***Application Note AN115*** provided with your upgrade
- Make sure you have the required version of Programmer Firmware to support the device you wish to program – see section 1.2.
- Plug the **'EQ-IO-CON-10'** module into the PPM3-MK2 / PPM4-MK1 programmer
- Plug the 20-way ISP cable supplied with the programmer into the **'JTAG'** ISP Header on the **'EQ-IO-CON-10'** module.
- Connect the other end of the 20-way ISP cable to the JTAG port on your Target Board
- You are now ready to program a Target Chip via JTAG

EQ-IOCON-10 	**I/O Connector Module 10 (AT91SAM7 JTAG) – Fast Connect Version** I/O connector module for In-System Programming (ISP) of Atmel AT91SAM7 microcontrollers using JTAG protocol **Features:** • Plugs into suitable Equinox programmer e.g. PPM4-MK1 • Generic 20-way ARM JTAG IDC connector • Single-in-line header with all programmer I/O brought out for wire-wrapping to bed-of-nails probe wires • Screw terminals for power connections • Target Vcc Status LED • Link to connect / isolate the programmer Vcc from the Target Vcc

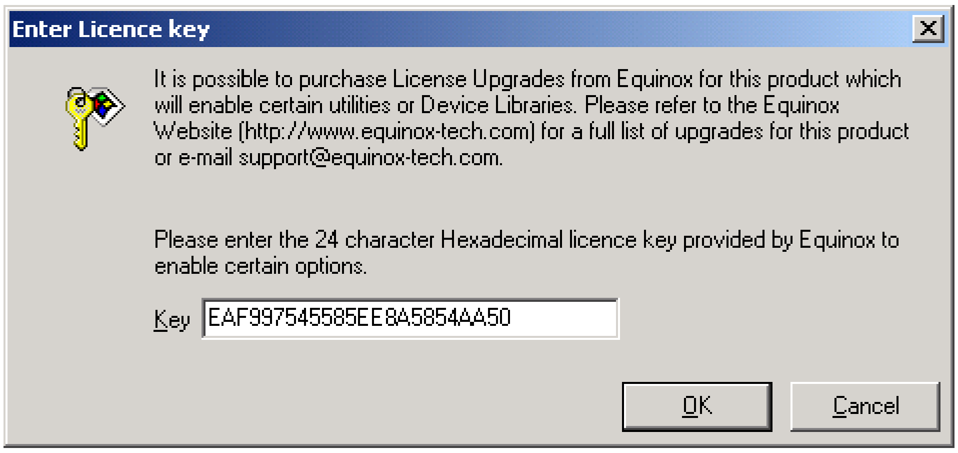
1.5.6 Entering the License String to upgrade your programmer

Once you have received the License String from Equinox, please follow the steps below to apply the upgrade to your programmer:

- Launch EQTools → The EQTools 'Welcome Screen' is displayed.
- Close down the EQTools 'Welcome Screen'
- From the top menu bar, select ***<Programmer><Programmer Info>***

→ the Programmer Information screen is displayed
- Click the ***<Enter License>*** button

→ The ***<Enter License Key>*** screen is displayed.


Enter Licence key

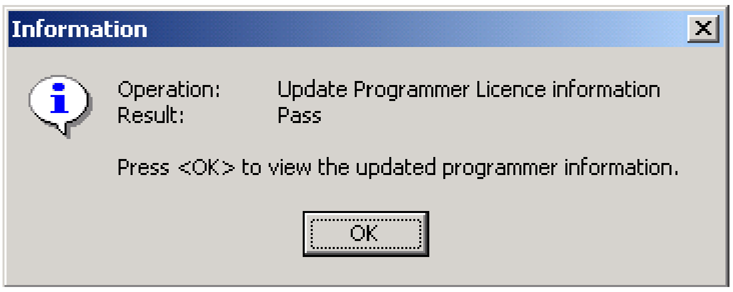
It is possible to purchase License Upgrades from Equinox for this product which will enable certain utilities or Device Libraries. Please refer to the Equinox Website (http://www.equinox-tech.com) for a full list of upgrades for this product or e-mail support@equinox-tech.com.

Please enter the 24 character Hexadecimal licence key provided by Equinox to enable certain options.

Key EAF997545585EE8A5854AA50

Enter the License String you were sent by Equinox
- Click ***<OK>***

→ EQTools should acknowledge that the attached programmer has been upgraded.


Information

Operation: Update Programmer Licence information
Result: Pass

Press <OK> to view the updated programmer information.

- Click ***<OK>***
- If you now check the Programmer Info screen, you should find that the entry for ***'AT91SAM7 ARM7 JTAG'*** is now ENABLED.
- Your programmer is now upgraded to support JTAG programming of Atmel AT91SAM ARM microcontrollers.

2.0 AT91SAM7 - JTAG Programming Interface

2.1 Overview

There a number of different methods of programming the on-chip FLASH of an AT91SAM7 microcontroller as detailed in the table below.

Interface		
FFPI	Fast FLASH Programming Interface	The FFPI Interface is a special programming interface designed for off-board programming of the AT91SAM7 devices. It is not suitable for ISP programming as it uses too many processor I/O pins.
USB	USB Interface	Uses the Atmel SAM-BA Boot Loader stored in ROM / FLASH.
SPI	SPI Interface	Uses the Atmel SAM-BA Boot Loader stored in ROM / FLASH.
JTAG	ARM Debug Port	Uses the ARM Debug Port for both In-System Debugging and Programming.

The Equinox range of programmers use the ***'ARM Debug JTAG Interface'*** for programming AT91SAM7 devices. This interface provides a method for both ***In-System Debugging (ISD)*** and ***In-System Programming (ISP)*** of Atmel AT91SAM7 Microcontrollers. The JTAG Interfaces uses an industry-standard set of signals to provide the connection between the programmer / debugger and the AT91SAM7 microcontroller. The actual JTAG Header (connector) used by Atmel and Equinox is the standard 20-pin JTAG connector used for most popular ARM7 microcontrollers.

In the development environment, the JTAG Interface can be used for ***In-System Debugging*** of the code running on the actual Target System. This method of operation requires the use of a third party In-System Debugger. Using this equipment, it is possible to download code (firmware) to a Target Chip and then execute and debug this code under PC control. The Debugger Software allows you to set breakpoints in the code, read / write memory locations, look at register contents etc. This functionality is not supported by any Equinox programmer.

In a production environment, the JTAG Interface can be use for high-speed ***In-System Programming (ISP)*** of the Target Microcontroller. This method of operation requires the use of any Equinox ISP Programmer which has been enabled to support the ***'AT91SAM7 JTAG'*** algorithms.

The Equinox ISP Programmer range supports high-speed ***In-System Programming (ISP)*** of a single or multiple AT91SAM7 microcontrollers on a Target Board using the 4-wire JTAG bus. Support has now been added for programming of any AT91SAM7 microcontroller when connected as part of a ***'JTAG Chain'***. This mode allows multiple JTAG devices to be in-system programmed using a single JTAG bus.

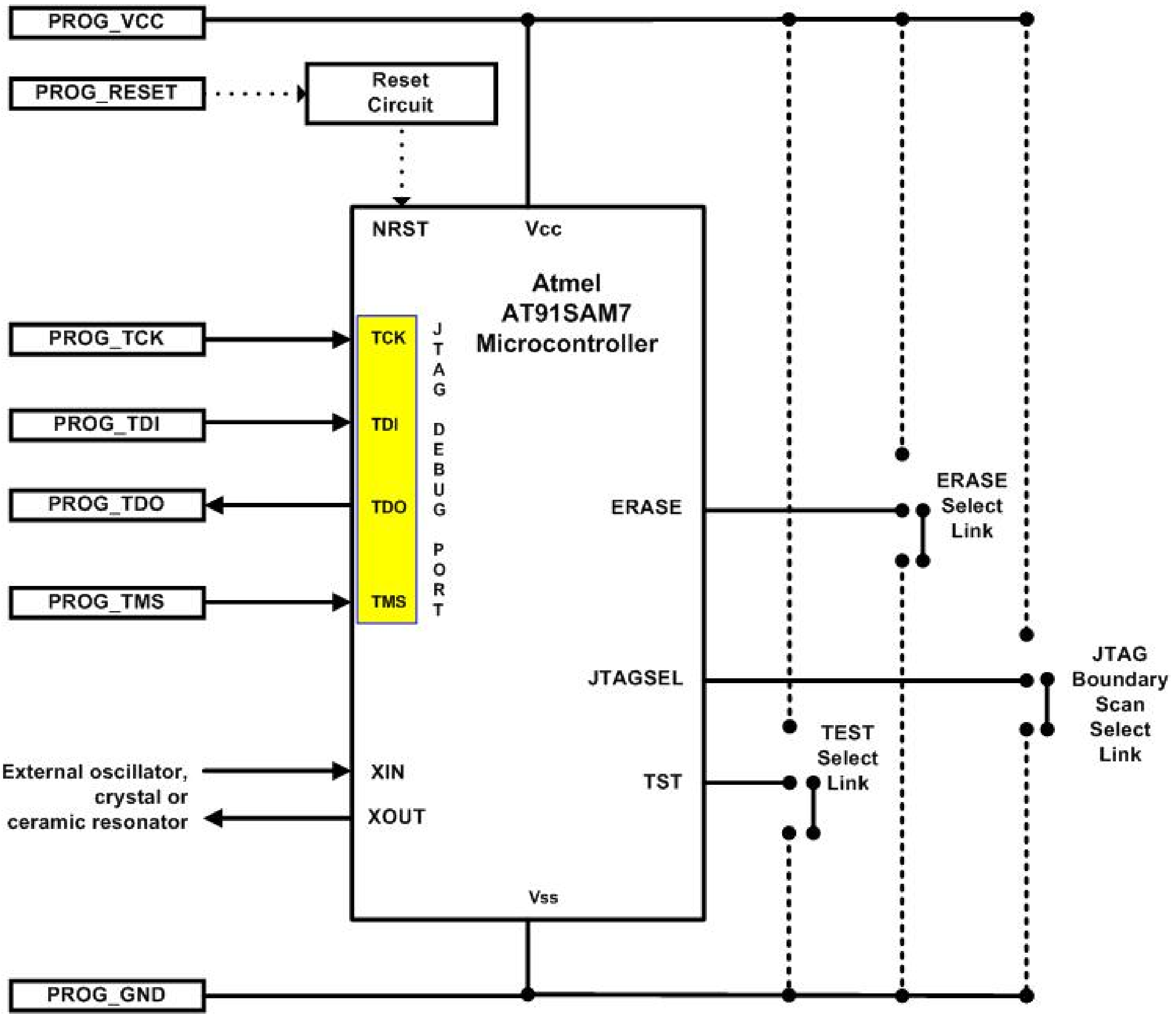
2.2 JTAG Features

- Fast Programming speeds
- Simple 4-wire JTAG bus connection + RESET signal
- JTAG programming does not depend on the AT91SAM7 oscillator frequency so JTAG programming will always work
- JTAG interface is compatible with any third party In-System Debugger so same interface can be used for development and production.
- Both ***'Single microcontroller'*** and ***'JTAG-in-a-chain'*** implementations are supported

2.3 JTAG single-chip In-System Programming (ISP) Schematic

The diagram below details the connections required to implement JTAG In-System Programming of a single Atmel AT91SAM7 microcontroller using an Equinox ISP programmer.

Fig 2.3 –AT91SAM7 – JTAG Programming Interface connections



AT91SAM7 – JTAG Programming Interface - signal names and directions

Programmer Signal Name	Signal description	Signal direction (from Programmer)	Connect to AT91SAM7 Microcontroller Pin	Signal direction (from Microcontroller)
PROG_TCK	Test Clock Pin	Output	TCK	Input
PROG_TDI	Test Data Input	Output	TDI	Input
PROG_TDO	Test Data Output	Input	TDO	Output
PROG_TMS	Test Mode Select	Output	TMS	Input
PROG_RESET	RESET	Output	nRST	Input / Output

2.4 JTAG Pins considerations

The JTAG port on an AT91SAM7 device is made up of the following pins: TDI, TDO, TCK and TMS and nTRST (optional). On the AT91SAM7 Target System, it is recommended that a pull-up resistor to the main microcontroller Vcc (e.g. 3.3V) is placed on the TCK, TMS and TDI pins. The TDO pin can be left floating.

Programmer Signal Name	Signal description	Signal direction (from Programmer)	Pull-up resistor on Programmer Interface Cable	Pull-up resistor on Target System
PROG_TCK	Test Clock Pin	Output	47k ohm	47k – 100k ohm
PROG_TDI	Test Data Input	Output	47k ohm	47k – 100k ohm
PROG_TDO	Test Data Output	Input	47k ohm	Floating
PROG_TMS	Test Mode Select	Output	47k ohm	47k – 100k ohm

Please note:

- The Equinox ***'SAM7 ISP Cable'*** which connects the Equinox Epsilon5, FS2003 and FS2009 programmers to the AT91SAM7 Target System features 47k ohm pull-up resistors on the TCK, TMS and TDI pins. These resistors can be removed if required.
- The Equinox ***'EQ-IOCON-10'*** I/O Connector Module which connects the Equinox PPM3-MK2 and PPM4-MK1 programmers to the AT91SAM7 Target System features 47k ohm pull-up resistors on the TCK, TMS and TDI pins. These resistors can be removed if required.

2.5 Erase Pin considerations

The ***'ERASE'*** pin of an AT91SAM7 device is used to force a complete hardware ***'Chip Erase'*** of the device.

Erase pin	Connect to:	Comment
During normal operation	Connect ERASE pin to GROUND (0V).	This protect the ERASE pin from spurious noise in harsh environments.
To perform a Chip Erase	Connect ERASE pin to Vcc for > 300ms	This causes the device to erase the GPNVM Fuse Bits, the entire FLASH contents and the "Security Bit".

If you do not plan to lock the AT91SAM7 device by programming the ***"Security Bit"***, then you can simply connect the ***'ERASE'*** pin to GROUND on the Target System.

Warning!

- If the ***"Security Bit"*** of the AT91SAM7 device is programmed to a '1', then the JTAG Port of the device is then disabled. The device will no longer respond to any JTAG programmer which is connected to the JTAG Port. This means that you cannot reprogram the device again!!!
- If you require a method to "unlock" an AT91SAM7 device once the ***"Security Bit"*** has been programmed, then it is recommended that provision for a link on the ***ERASE*** pin to both Vcc and GROUND is made.

2.6 TEST Pin

The ***'TST'*** pin of an AT91SAM7 device is used to enter ***'Fast FLASH Programming Mode (FFPI)'*** or ***'SAM-BA Boot Recovery'*** mode. The Equinox SAM7 programmers do not use either of these modes so the use of the ***'TST'*** pin is not required. It is therefore recommend that this pin is connected to GROUND.

Please note:
If you wish to be able to use the Atmel SAM-BA software then you may want to make the ***'TST'*** pin user-selectable.

2.7 JTAGSEL Pin

The ***'JTAGSEL'*** pin is used to force the AT91SAM7 device to enter ***'JTAG Boundary Scan'*** mode. This mode is **NOT** used by Equinox ISP programmers for programming these devices. The ***'JTAGSEL'*** pin can therefore be connected to GROUND.

Please note:
If you wish to be able to use a third party Boundary Scan Tester on an AT91SAM7 Target Board, then you may want to make the ***'JTAGSEL'*** pin user-selectable.

2.8 RESET (NRST) Pin

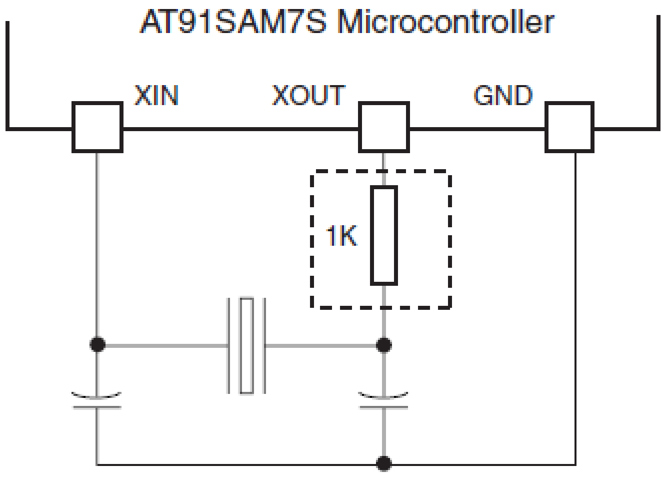
The ***'NRST'*** pin on an AT91SAM7 device is controlled by the on-chip ***'RESET Controller (RSTC)***'. The external Equinox ISP programmer does not actually need control of the ***'NRST'*** pin for programming purposes but control of the ***'NRST'*** pin may be required to control the execution of user-firmware once it has been programmed into the FLASH.

Important notes:

- Equinox programmers do **NOT** drive the ***RESET*** pin (nRST) as standard. Instead the RESET pin is tri-stated so the programmers has no control over the reset timing of the AT91SAM7 device.
- The ***'NRST'*** pin on an AT91SAM7 device is actually **disabled** as a RESET pin as standard.
- It is up to the user firmware running in the AT91SAM7 device to enable the RESET pin in order to allow an external RESET event.
- This means that the ***'NRST'*** pin cannot be used to guarantee that user code will not execute after a POR (Power-on-RESET) condition.
- It is therefore very important to make sure that the user firmware does not place the AT91SAM7 device into any type of 'Sleep Mode' immediately after reset otherwise an external JTAG programmer may not be able to enter programming mode.
- This can also mean that the very first time a virgin device is programmed, it will program OK. However on subsequent programming attempts the ***nRST*** pin may force the device into the reset condition thus stopping the external programming from connecting to the device.

2.9 External Oscillator source

The AT91SAM7 devices initially use their ***'Internal Oscillator'*** to execute from after a POR (Power-on-RESET) condition. This oscillator runs from anywhere between 22 kHz and 46 kHz so the AT91SAM7 will execute code but this execution will be very slow. An Equinox JTAG programmer will use this ***'Internal Oscillator'*** to initially 'connect to' the target AT91SAM7 device via JTAG. Once this connection has been made, the programmer will then attempt to switch to a faster external oscillator source such as a crystal or an oscillator signal which must be connected to the XIN / XOUT pin of the device – see example connection of external crystal in the illustration below.



If a suitable external oscillator is not available then the programmer will use the ***'Internal Oscillator'*** instead. This will make any programming operation very slow!

Please note:
An external oscillator source with a frequency of >= 3 .0 MHz is required for fast programming.

Please refer to Appendix 2 for further details about configuring the oscillator for programming.

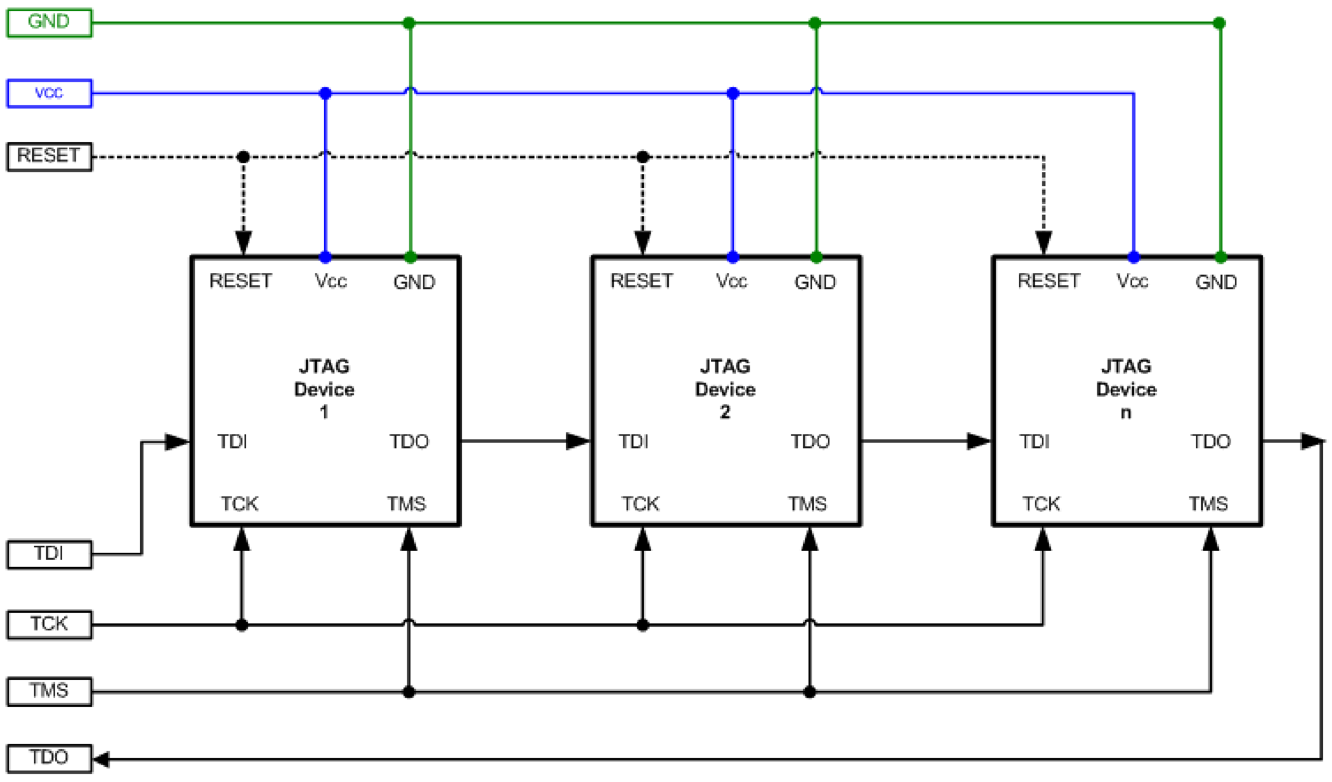
2.10 Target Vcc requirements

The power supply to the AT91SAM7 device must meet very stringent power-up slope requirements otherwise the device may not function correctly. These devices require a fast power-up slope otherwise the device may not start-up correctly leading to erratic execution of code. Please refer to the device datasheet and any relevant errata sheet for more detailed information.

2.11 JTAG-in-a-chain In-System Programming (ISP) Schematic

The diagram below details the connections required to implement JTAG In-System Programming of either a single or multiple Atmel AT91SAM7 Microcontrollers which are connected in a JTAG Chain' arrangement.

Fig 2.6 – AT91SAM7 – JTAG Programming Interface connections



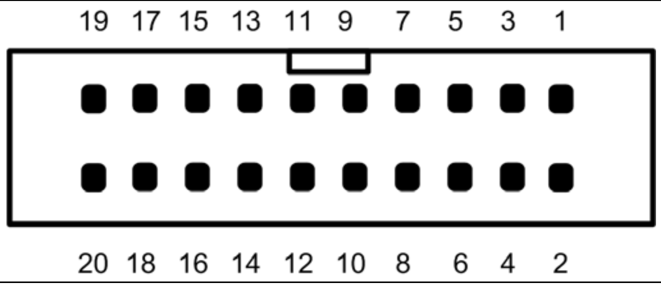
The TDI signal is fed into the TDI input on the first JTAG Device in the chain. The data path then goes through the first device and comes out on the TDO pin. The TDO pin is connected to the TDI pin of the next JTAG Device in the chain.

AT91SAM7 – JTAG Programming Interface - signal names and directions

Programmer Signal Name	Signal description	Signal direction (from Programmer)	Connect to AT91SAM7 Microcontroller Pin	Signal direction (from Microcontroller)
PROG_TCK	Test Clock Pin	Output	TCK	Input
PROG_TDI	Test Data Input	Output	TDI	Input
PROG_TDO	Test Data Output	Input	TDO	Output
PROG_TMS	Test Mode Select	Output	TMS	Input
PROG_RESET	RESET	Output	RESET	Input

2.12 ARM JTAG Debug Interface (20-way)

All Equinox SAM7 ISP programmer solutions offer the use of the standard 20-way ***'ARM JTAG Debug Header'*** to connect to an AT91SAM7 Target System. This connector is a 20-way 2.54mm female IDC (Insulation Displacement Connector).

	***ARM JTAG Debug Header*** The connector is a 20-pin bump-polarised IDC connector with 0.1" pin spacing. Pin 1 is the top right pin as shown in the diagram opposite.

The pin-out of this connector is detailed in the table below:

Pin No	Signal	Type	Description
1	**VTref**	Input	**Target System Reference Voltage** This pin is used by the programmer to sense / power the programmer Line Driver circuitry.
2	**Vsupply**	No connect	**Not used by Equinox Programmers**
3	**nTRST**	Output	**No connect to Equinox Programmer** This signal is not used by Equinox Programmers. It is pulled high with a 47k ohm resistor on the programmer cable / connector module.
4	**PROG_GND**	Passive	**Ground Connection** Common ground connection between programmer and Target System.
5	**PROG_TDI**	Output	**JTAG TDI – Test Data Input pin** Data signal from programmer to Target Device JTAG port.
6	**PROG_GND**	Passive	**Ground Connection** Common ground connection between programmer and Target System.
7	**PROG_TMS**	Output	**JTAG TMS – Test Mode Select pin** Mode Select Signal from programmer to Target Device JTAG port.
8	**PROG_GND**	Passive	**Ground Connection** Common ground connection between programmer and Target System.
9	**PROG_TCK**	Output	**JTAG TCK – Test Clock Signal pin** Clock signal from programmer to Target Device JTAG port.
10	**PROG_GND**	Passive	**Ground Connection** Common ground connection between Programmer and

			Target System.
11	**RTCK**	No connect	**Input Return Test Clock signal from Target System** This signal is not used by Equinox programmers.
12	**PROG_GND**	Passive	**Ground Connection** Common ground connection between Programmer and Target System.
13	**PROG_TDO**	Input	**JTAG TDO – Test Data Output pin** JTAG data output from Target device JTAG port to programmer. Typically connected to TDO pin on AT91SAM7 device.
14	**PROG_GND**	Passive	**Ground Connection** Common ground connection between Programmer and Target System.
15	**PROG_RESET**	O	**Microcontroller RESET control signal** This pin connects to the main RESET pin of the Target AT91SAM7 microcontroller or to the AT91SAM7 RESET circuit on the Target System.
16	**PROG_GND**	Passive	**Ground Connection** Common ground connection between Programmer and Target System.
17	**-**	No connect	**No connect to Equinox Programmer**
18	**PROG_GND**	Passive	**Ground Connection** Common ground connection between Programmer and Target System.
19	**-**	No connect	**No connect to Equinox Programmer**
20	**PROG_GND**	Passive	**Ground Connection** Common ground connection between Programmer and Target System.

3.0 Creating an EDS (Development) Project

3.1 Overview

This section describes how to make a ***'Programming Project'*** for an Atmel AT91SAM7 device.

3.2 Information required to create a AT91SAM7 Project

The following information is required about the Target Board in order to create an AT91SAM7 JTAG Programming Project:

#	Information / data required	Example
1	AT91SAM7 Device part number	AT91SAM7S256
2	JTAG connections / connector on Target board	20-way IDC connector
3	JTAG configuration	i. Single JTAG device or ii. JTAG device is part of a 'JTAG chain'
4	JTAG chain configuration parameters These parameters are required if the device to be programmed is part of a 'JTAG chain'. If a single device is to be programmed via JTAG, then simply set all the 'JTAG Chain' parameters to '0'.	• Devices before: 0 • Devices after: 0 • Bits before: 0 • Bits after: 0
5	Target oscillator configuration to be used during programming – see appendix 3	• Internal RC Oscillator • External crystal oscillator • External clock signal
6	Target external device **oscillator frequency**	e.g. 12 MHz
7	Target System Vcc voltage	e.g. 3.3V
8	Target System maximum current consumption	e.g. 100mA
9	FLASH area 'Program File'	Binary (*.bin) or Intel Hex (*.hex)
10	Configuration Fuse values These fuse values describe how the 'Configuration Fuses' in the AT91SAM7 device are to be programmed.	i. Boolean fuse values: e.g. GPNVM0, GPNVM1 etc
11	***"Security Bit"*** settings	The "Security Bit" is used to disable the JTAG port after programming is complete.
12	Reset circuit parameters	e.g. • Capacitor / Resistor circuit • Watchdog supervisor circuit • Voltage monitoring circuit

3.3 Creating an EDS (Development project)

The simplest way to create a Programming Project for a JTAG device is to use the ***EDS (Development Mode)*** Wizard.

The steps required to create a project are as follows:

- Select **<Create a new Development (EDS) Project**>

→ the EDS (Development) Wizard will launch

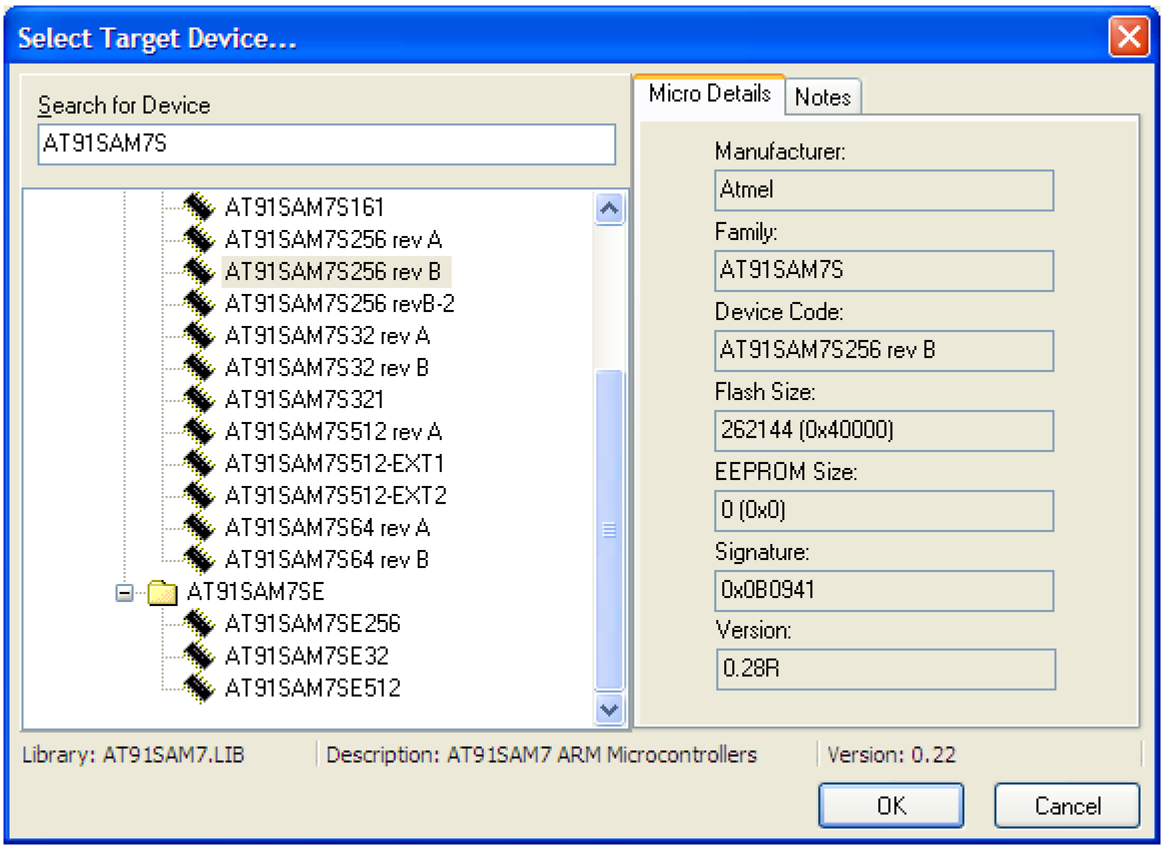
- Select the correct AT91SAM7 device
- Select the correct AT91SAM7 'Oscillator Source' and 'Oscillator frequency'
- Set up the correct 'Target Voltage' and any other Voltage / Current related parameters
- Specify the data file to be programmed into the FLASH (optional at this stage)

3.4 Selecting the correct Target Device

It is important to select the correct 'Target Device' when programming an AT91SAM7 device. The part number of the device should be printed on the top of the chip e.g. ***'AT91SAM7S256'***.

3.4.1 Device selection

- Click **<Next>** → the **<Select Target Device>** screen will be displayed.
- Type in the ***'Device Part Number'*** e.g. ***AT91SAM7S256*** into the ***'Search for Device'*** field → a list of all matching devices will be displayed in the box underneath.


Select Target Device...

Search for Device: AT91SAM7S

AT91SAM7S161
AT91SAM7S256 rev A
AT91SAM7S256 rev B
AT91SAM7S256 revB-2
AT91SAM7S32 rev A
AT91SAM7S32 rev B
AT91SAM7S321
AT91SAM7S512 rev A
AT91SAM7S512-EXT1
AT91SAM7S512-EXT2
AT91SAM7S64 rev A
AT91SAM7S64 rev B
AT91SAM7SE
AT91SAM7SE256
AT91SAM7SE32
AT91SAM7SE512

Manufacturer: Atmel
Family: AT91SAM7S
Device Code: AT91SAM7S256 rev B
Flash Size: 262144 (0x40000)
EEPROM Size: 0 (0x0)
Signature: 0x0B0941
Version: 0.28R

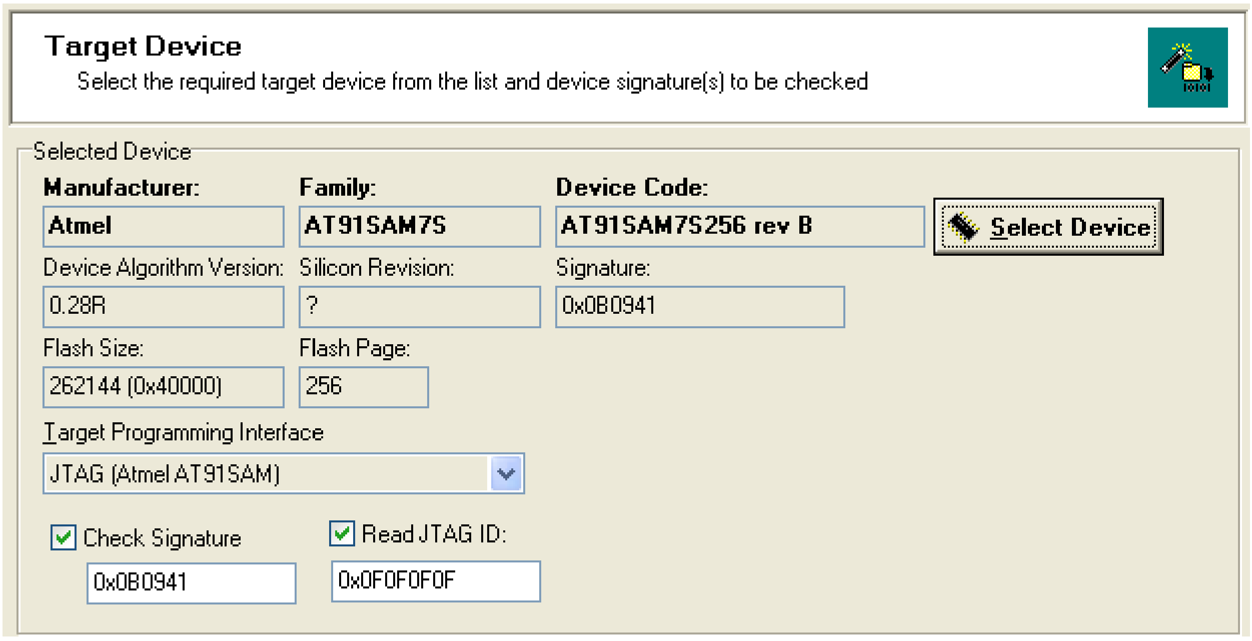
Library: AT91SAM7.LIB | Description: AT91SAM7 ARM Microcontrollers | Version: 0.22

- Select the required device from the list and then click **<OK>**

→ The device is now selected.

3.4.2 Silicon revision and Chip ID / Signature

- On the next screen, check that the device selection and all other device parameters are correct



Target Device
Select the required target device from the list and device signature(s) to be checked

Selected Device

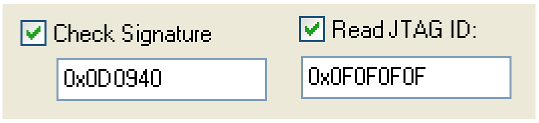
Manufacturer: Atmel	**Family:** AT91SAM7S	**Device Code:** AT91SAM7S256 rev B	
Device Algorithm Version: 0.28R	Silicon Revision: ?	Signature: 0x0B0941	
Flash Size: 262144 (0x40000)	Flash Page: 256		

Target Programming Interface: JTAG (Atmel AT91SAM)

☑ Check Signature: 0x0B0941

☑ Read JTAG ID: 0x0F0F0F0F

- The project is set to automatically read and validate the ***'JTAG ID'*** of the Target Device by default. The ***'JTAG Revision'*** is not validated if the first digit of the ***'JTAG ID'*** is '0'.
- Atmel may have released multiple ***'silicon versions'*** of the same AT91SAM7 device. Each of these versions usually has a different / unique ***'Chip ID'*** / Signature.
- The actual ***'Chip ID'*** for the AT91SAM7 device being programmed can be found in the Atmel datasheet. Alternatively, it can be read from the target device using EDS.



☑ Check Signature: 0x0D0940

☑ Read JTAG ID: 0x0F0F0F0F

- If you are unsure of the actual ***'Chip ID'*** for the AT91SAM7 device being programmed, select the nearest device for now and then read the actual ***'Chip ID'*** back using EDS.
- Click **<Next>** to advance to the next screen

3.5 Target Oscillator Settings

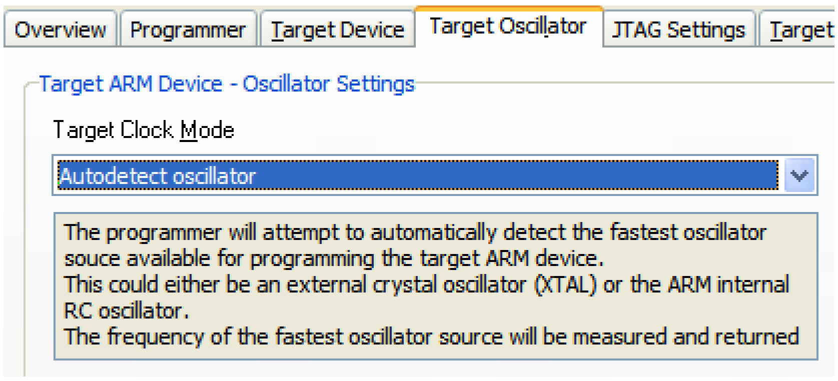
When creating a ***'Programming Project'*** for an Atmel AT91SAM7 device using the Equinox EQTools software, it is necessary to specify the correct ***oscillator source*** and ***oscillator frequency*** in the ***'Target Oscillator Settings'*** section of the project. These settings are used to calculate the correct ***'FLASH timing parameters'*** for the AT91SAM7 device which are then used during the programming process. The programmer requires the fastest possible oscillator in order to program the FLASH as fast as possible.

If you are using EQTools build 2000 or above and programmer firmware >= 5.20, then the programmer supports selection of all possible AT91SAM7 oscillator sources. It also supports the option to automatically detect the fastest oscillator available.

Supported oscillator sources:

- Internal RC Oscillator
- External crystal oscillator
- External oscillator signal

The screen below allows you to set up the ***'Target Clock Mode'*** and the ***'Target Oscillator frequency'***.



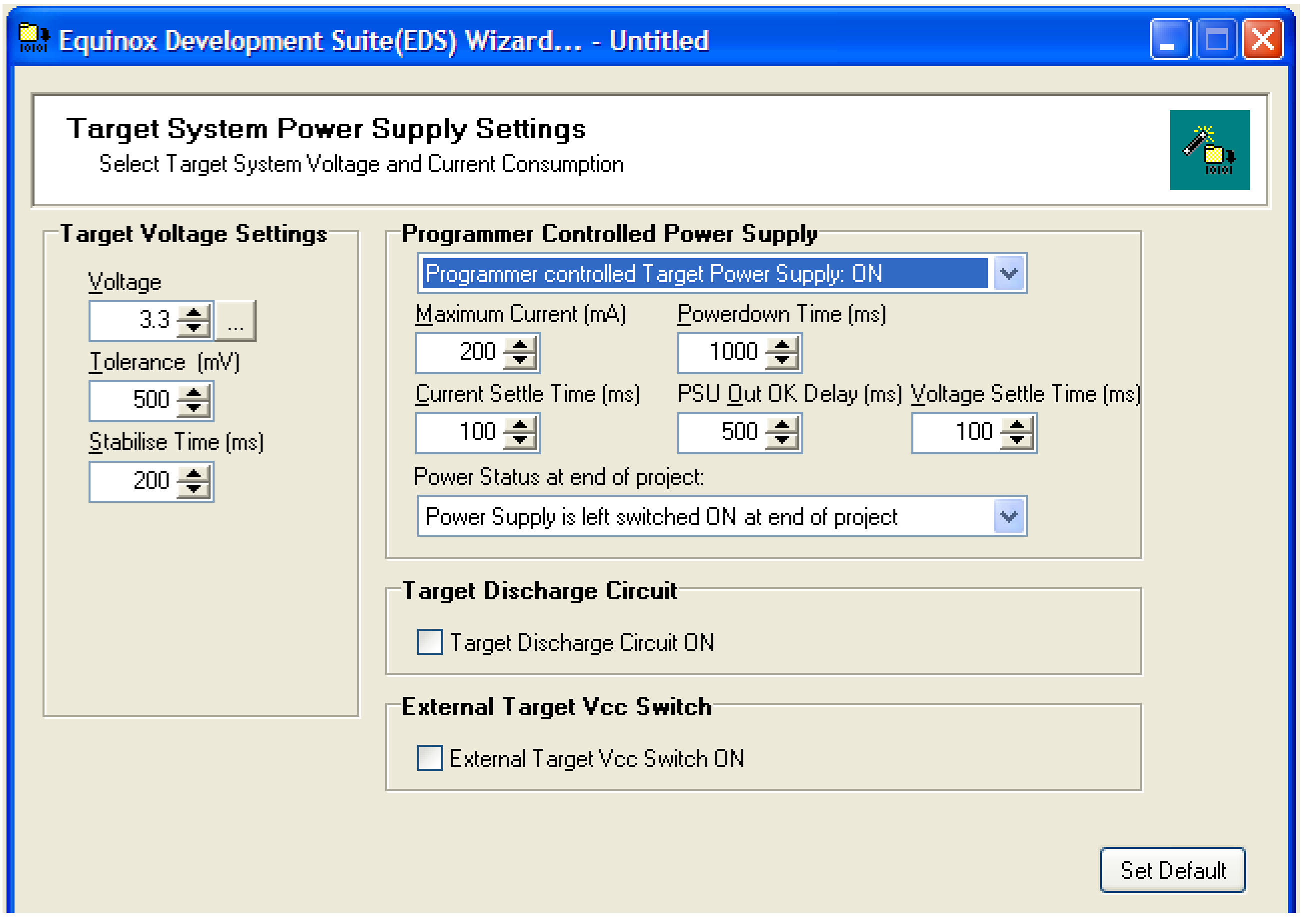
Target Clock Mode: Autodetect oscillator

The programmer will attempt to automatically detect the fastest oscillator souce available for programming the target ARM device.
This could either be an external crystal oscillator (XTAL) or the ARM internal RC oscillator.
The frequency of the fastest oscillator source will be measured and returned

Please refer to ***Appendix 3*** for detailed instructions of how to select the correct ***oscillator source*** and ***oscillator frequency***.

3.6 Target System – Power Supply Settings

This screen allows you to set up the Power Supply characteristics of your Target System.


Equinox Development Suite(EDS) Wizard... - Untitled

Target System Power Supply Settings
Select Target System Voltage and Current Consumption

Target Voltage Settings
Voltage: 3.3
Tolerance (mV): 500
Stabilise Time (ms): 200

Programmer Controlled Power Supply
Programmer controlled Target Power Supply: ON
Maximum Current (mA): 200
Powerdown Time (ms): 1000
Current Settle Time (ms): 100
PSU Out OK Delay (ms): 500
Voltage Settle Time (ms): 100
Power Status at end of project: Power Supply is left switched ON at end of project

Target Discharge Circuit
☐ Target Discharge Circuit ON

External Target Vcc Switch
☐ External Target Vcc Switch ON

i. Select the Target Voltage

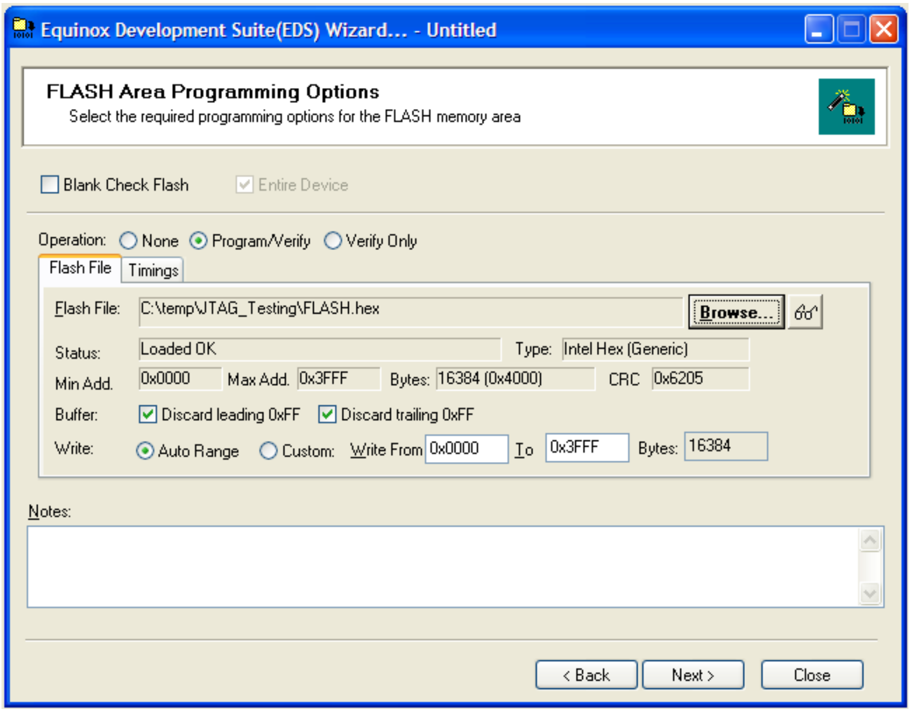
- This should be the voltage at which the Target Device itself e.g. AT91SAM7 IC is being powered at during the programming operation. This is usually 3.0 – 3.3V.
- Set the ***'Voltage Tolerance'*** to be as wide as possible e.g. 500mV to allow for power supply variations. If the programmer is powering the Target System, this will also give a faster power-up time.
- It may also be possible to power the entire Target System by feeding in a higher voltage e.g. +5V into the power supply input on the Target System.

ii. Set up the Target Powering and current parameters

- This option is only available for the PPM3-MK2, PPM4-MK1 and ISPnano programmers.
- If the programmer is to power the Target System, select ***<Programmer controlled Target Power Supply: ON>***
- Set the ***'Maximum Current'*** to the maximum possible current which the Target System could draw from the programmer.
- Leave all other settings as default.

3.7 Specifying the FLASH (Code) File

This screen allows you to specify the Code (firmware) file which is to be programmed into the FLASH area of the Target Device. This is an optional step – you can also specify the file once you are in the Development Suite (EDS).


Equinox Development Suite(EDS) Wizard... - Untitled

FLASH Area Programming Options
Select the required programming options for the FLASH memory area

Blank Check Flash [ ] Entire Device [✓]

Operation: None [✓ Program/Verify] Verify Only

Flash File | Timings

Flash File: C:\temp\JTAG_Testing\FLASH.hex

Status: Loaded OK Type: Intel Hex (Generic)

Min Add. 0x0000 Max Add. 0x3FFF Bytes: 16384 (0x4000) CRC 0x6205

Buffer: [✓] Discard leading 0xFF [✓] Discard trailing 0xFF

Write: [✓ Auto Range] Custom: Write From 0x0000 To 0x3FFF Bytes: 16384

Notes:

i. Blank Check the FLASH

- If the chip has been erased at the start of the programming cycle, then the FLASH should already be blank (i.e. all locations contain the value 0xFF).
- If you want to be absolutely sure the FLASH is blank, you can enable the ***'Blank Check Flash'*** option. This will perform a full Blank Check of the FLASH area to check that all locations are set to 0xFF.
- Warning – this check can be time-consuming and will increase the overall programming time!

ii. Selecting the FLASH File

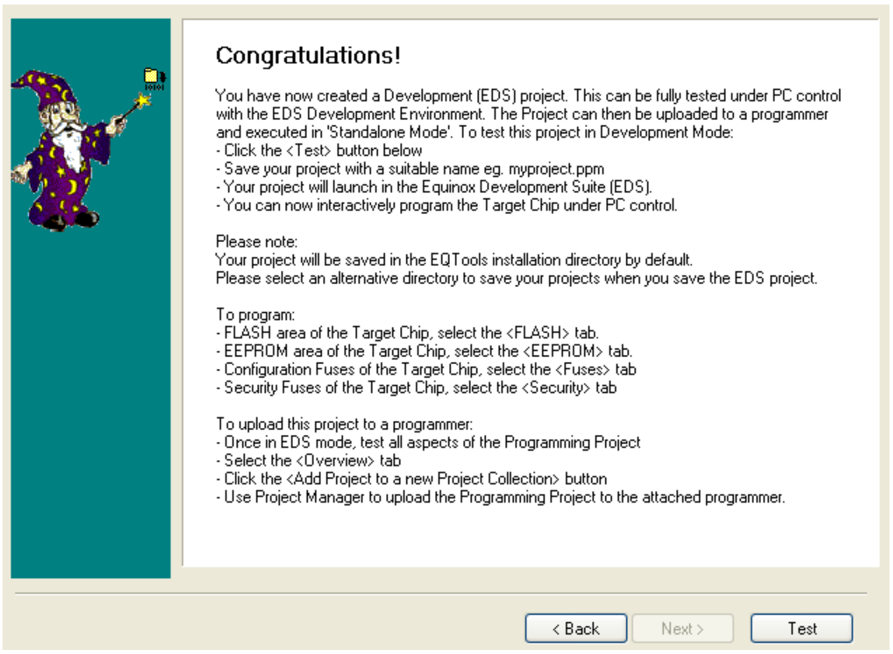
- Click the ***<Browse>*** button
- Browse to and select the file you wish to load and then select ***<OK>***
- If the input file is a BINARY file then the wizard will load the data in from file starting at address 0x0000 and continuing contiguously to the end of the file.
- If the input file is an ***'INTEL HEX'*** or ***'Motorola S-Record'*** file then the wizard will load in from file from the start address specified in the file to end address specified in the file.

Important note:

The ***'Internal FLASH Memory'*** of an AT91SAM7 device can be mapped to two different address ranges. EQTools expects the start address of data in the ***'INTEL HEX'*** or ***'Motorola S-Record'*** file to be ***0x000000***. If the start address in the file is ***0x100000*** then it will be necessary to resave the file with an offset of ***0x000000***.

3.8 Launching EDS at the end of the EDS Wizard

Once you reach the end of the EDS Wizard, click the ***<Test>*** button to launch the project in the Equinox Development Suite (EDS).



Congratulations!

You have now created a Development (EDS) project. This can be fully tested under PC control with the EDS Development Environment. The Project can then be uploaded to a programmer and executed in 'Standalone Mode'. To test this project in Development Mode:
- Click the <Test> button below
- Save your project with a suitable name eg. myproject.ppm
- Your project will launch in the Equinox Development Suite (EDS).
- You can now interactively program the Target Chip under PC control.

Please note:
Your project will be saved in the EQTools installation directory by default.
Please select an alternative directory to save your projects when you save the EDS project.

To program:
- FLASH area of the Target Chip, select the <FLASH> tab.
- EEPROM area of the Target Chip, select the <EEPROM> tab.
- Configuration Fuses of the Target Chip, select the <Fuses> tab
- Security Fuses of the Target Chip, select the <Security> tab

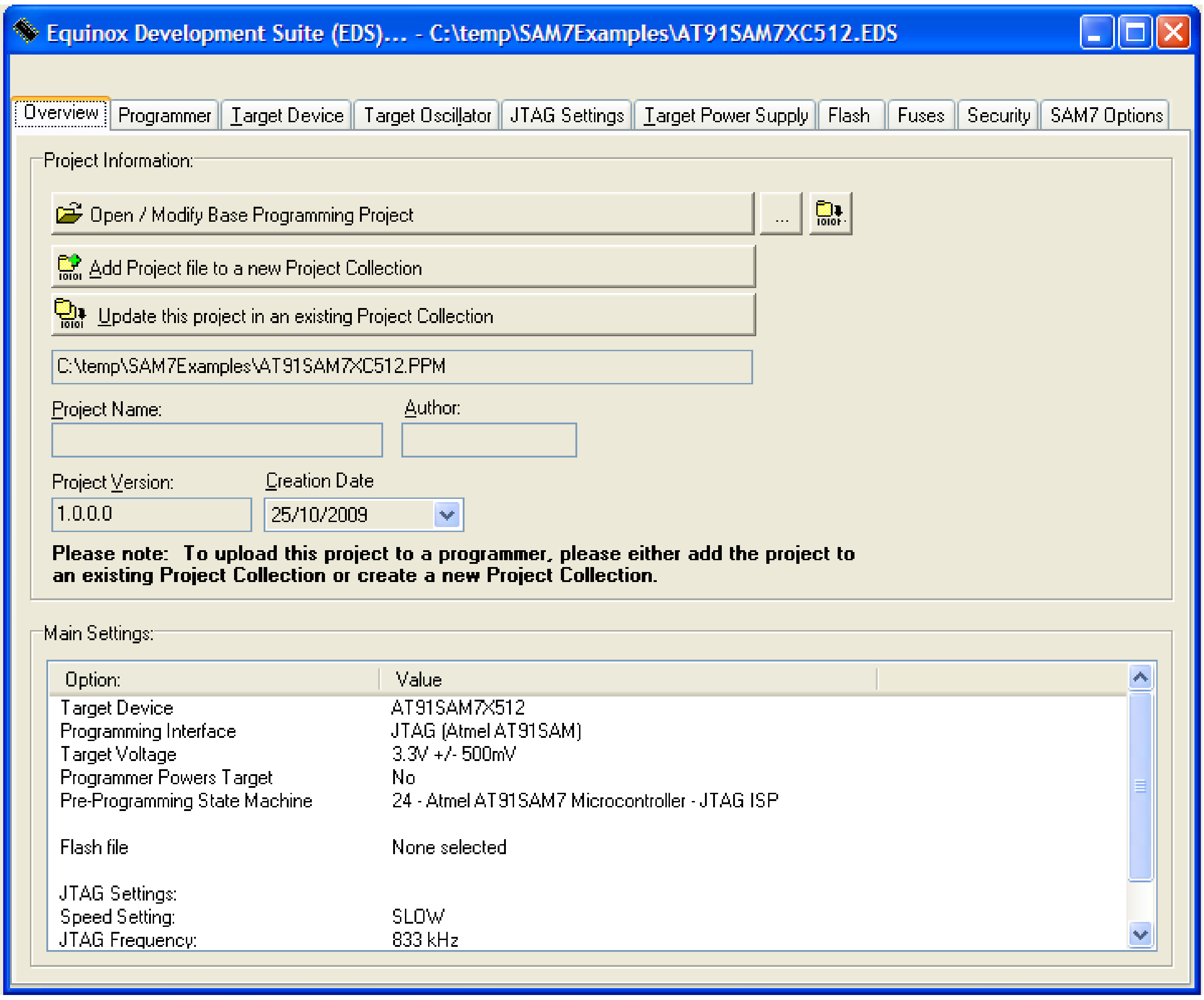
To upload this project to a programmer:
- Once in EDS mode, test all aspects of the Programming Project
- Select the <Overview> tab
- Click the <Add Project to a new Project Collection> button
- Use Project Manager to upload the Programming Project to the attached programmer.

Enter a name for the EDS project e.g. ***AT91SAM7S256*** and click the ***<Test>*** button
→ Your project will now launch in EDS (Development) Mode.

4.0 Testing a Project in Development (EDS) Mode

4.1 Introduction to EDS

If you have clicked the **<Test>** button at the end of the EDS Wizard, then an EDS (Development Mode) session will now launch.


Equinox Development Suite (EDS)... - C:\temp\SAM7Examples\AT91SAM7XC512.EDS
Overview
Programmer
Target Device
Target Oscillator
JTAG Settings
Target Power Supply
Flash
Fuses
Security
SAM7 Options
Open / Modify Base Programming Project
Add Project file to a new Project Collection
Update this project in an existing Project Collection

Project Information:

C:\temp\SAM7Examples\AT91SAM7XC512.PPM

Project Name:

Author:

Project Version: 1.0.0.0

Creation Date: 25/10/2009

Please note: To upload this project to a programmer, please either add the project to an existing Project Collection or create a new Project Collection.

Main Settings:

Option:	Value
Target Device	AT91SAM7X512
Programming Interface	JTAG (Atmel AT91SAM)
Target Voltage	3.3V +/- 500mV
Programmer Powers Target	No
Pre-Programming State Machine	24 - Atmel AT91SAM7 Microcontroller - JTAG ISP
Flash file	None selected
JTAG Settings:	
Speed Setting:	SLOW
JTAG Frequency:	833 kHz

4.2 EDS - Default settings for JTAG, statemachine etc

The following default settings will be used:

- SLOW JTAG speed at maximum SLOW frequency
- Single JTAG device (no JTAG Chain)
- Target System not powered by programmer (unless enabled during the EDS Wizard)
- The default AT91SAM7 - JTAG pre-programming state machine – ***Statemachine '24'*** will be used.
- The ***RESET*** pin will **NOT** be driven by the programmer by default. Instead it will be tri-stated at all times.
- The ***"Configuration Fuse"*** Write is disabled (can be enabled in EDS)
- The ***"Security Bit"*** Write is disabled (can be enabled in EDS)

At this stage there are still a few parameters which may need to be set up / checked before the programmer will communicate with the Target Device on the Target Board.

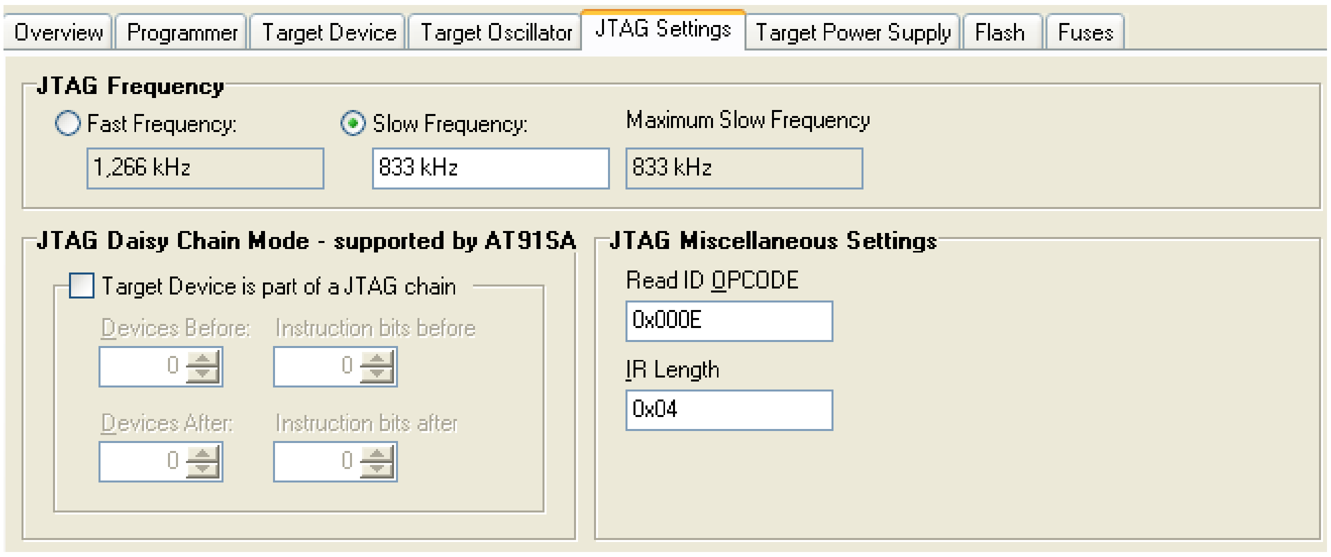
Please follow the instructions in the next sections which explain how to set up the:

- JTAG Frequency
- JTAG chain settings
- Oscillator source and frequency
- Test the Target Voltage

4.3 JTAG Settings

The JTAG Frequency must be set up before any programming operation can take place.

To set up the JTAG Frequency, select the **<JTAG Settings>** tab.



Overview | Programmer | Target Device | Target Oscillator | JTAG Settings | Target Power Supply | Flash | Fuses

JTAG Frequency

Fast Frequency: 1,266 kHz [✓ Slow Frequency: 833 kHz] Maximum Slow Frequency: 833 kHz

JTAG Daisy Chain Mode - supported by AT91SA

☐ Target Device is part of a JTAG chain

Devices Before:	Instruction bits before
0	0
Devices After:	**Instruction bits after**
0	0

JTAG Miscellaneous Settings

Read ID OPCODE: 0x000E

IR Length: 0x04

4.3.1 JTAG frequency

There are two choices of JTAG frequency setting:

i. Slow JTAG (default setting)

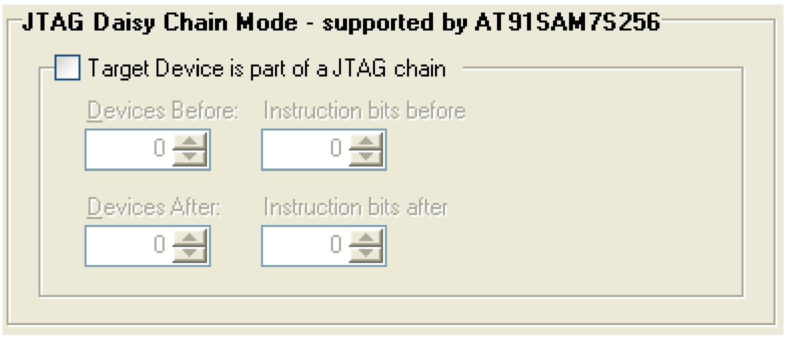
- Selecting the ***SLOW JTAG*** option allows you to specify a JTAG frequency from 30 kHz up the maximum SLOW JTAG frequency.
- This option should be used if there are reliability problems with JTAG programming using the ***'FAST JTAG'*** option. If the JTAG frequency is slowed down, the reliability of programming often increases.

ii. FAST JTAG

- Selecting the ***'FAST JTAG'*** option selects a single high-speed JTAG frequency which is fixed for the selected programmer.
- This option should be tried to see if reliable programming of the Target System is possible. If programming proves to be unreliable, then try using the ***'SLOW JTAG'*** instead.

4.3.2 JTAG Chain settings

To set up the position of the Target AT91SAM7 Device in a JTAG Chain, select the **<JTAG Settings>** tab.



JTAG Daisy Chain Mode - supported by AT91SAM7S256

☐ Target Device is part of a JTAG chain

Devices Before: 0
Instruction bits before: 0
Devices After: 0
Instruction bits after: 0

i. Single JTAG Device
If the programmer is only connected to ONE JTAG device, then you can leave all the settings as their default value of '0'. This means the Target Device is the first and only device in the JTAG Chain.

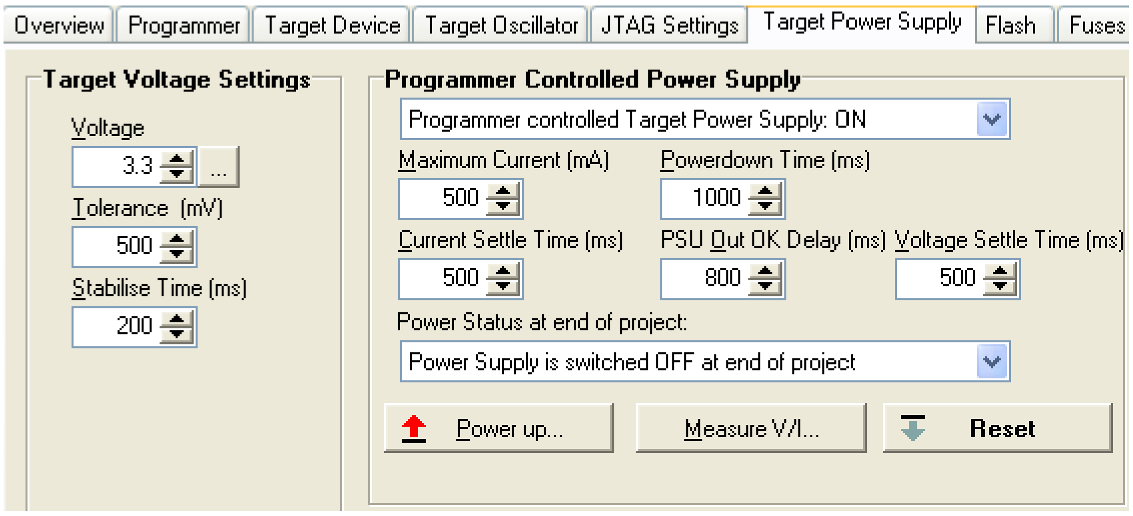
ii. Device is part of a JTAG Chain
If the Target Device to be programmed is connected so it is part of a JTAG chain, then it is necessary to specify the number of ***'JTAG devices'*** and ***'Instruction Bits'*** both BEFORE and AFTER the Target Device in the Chain – see example JTAG Chain below.

4.4 Checking the AT91SAM7 – Target Voltage

It is a good idea to check that the target AT91SAM7 device is powered at the correct voltage before trying to program it. An AT91SAM7 device normally runs at between 3.0 and 3.3V with the core running at 1.8V. The programmer ***'Target Vcc'*** pin should be connected to the 3.3V rail on the Target System allowing the programmer to measure the Target Voltage (even if the programmer is not powering the Target System).

To check the Target Voltage using the programmer, please follow the instructions detailed below….

- Select the ***<Target Power Supply>*** tab



Target Voltage Settings
Voltage: 3.3
Tolerance (mV): 500
Stabilise Time (ms): 200

Programmer Controlled Power Supply
Programmer controlled Target Power Supply: ON
Maximum Current (mA): 500
Powerdown Time (ms): 1000
Current Settle Time (ms): 500
PSU Out OK Delay (ms): 800
Voltage Settle Time (ms): 500
Power Status at end of project:
Power Supply is switched OFF at end of project

If the programmer is going to power the Target System…..

- Set up the voltage / current parameters accordingly (see programmer User Manual for detailed instructions)
- The ***'Target Voltage'*** should be set the actual voltage which the AT91SAM7 device is running at e.g. 3.3V.
- The programmer will then generate JTAG signals which swing between 0V and the ***'Target Voltage'***.
- Click the ***<Power up>*** button to power up the Target System.
- → The programmer will then switch on the programmer controlled power supply and the Target System should power up to the specified voltage.
- The measured ***'Target Voltage'***. will be continuously displayed. If it is not, then you can simply click the ***<Measure V/I>*** button.
- The voltage should be within 3.0 and 3.3V.

If the programmer is <u>NOT</u> powering the Target System…

Switch on the independent power supply which is connected to the Target System.
Click the ***<Measure>*** button to measure the ***'Target Voltage'***.

- The voltage should be within 3.0 and 3.3V.

4.5 Testing JTAG communication with the AT91SAM7 device

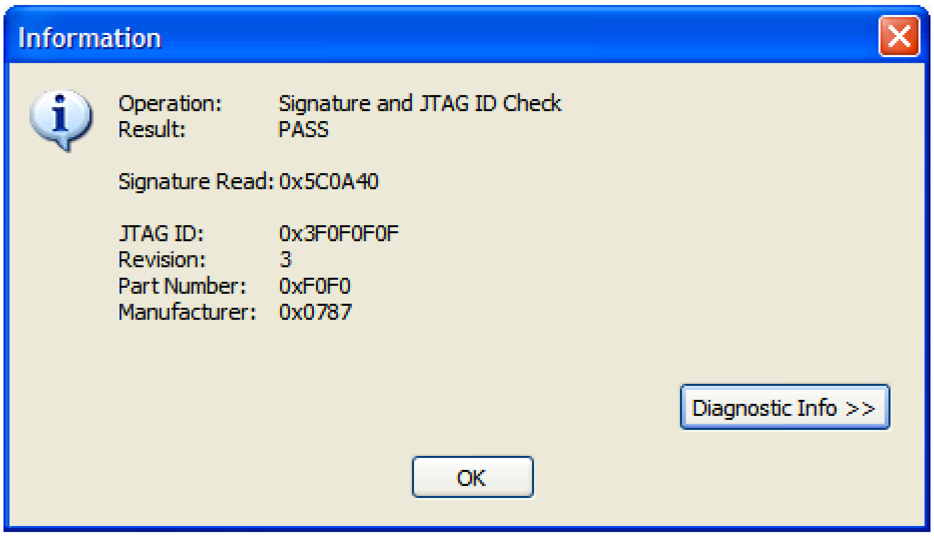
4.5.1 Overview

To make sure that the programmer can communicate to the Target JTAG device, try reading back the ***Device Signature (SAM7 Chip ID)*** and ***JTAG ID*** as follows:

- Select the **<Target Device>** tab
- Click the **<Check ID>** button

→ The programmer will now try to communicate with the Target Chip via the JTAG Interface

→ If the Target Chip responds correctly, then EDS will report ***'Signature and JTAG ID Check – Result: Pass'***.



```
Operation:      Signature and JTAG ID Check
Result:         PASS

Signature Read: 0x5C0A40

JTAG ID:        0x3F0F0F0F
Revision:       3
Part Number:    0xF0F0
Manufacturer:   0x0787
```

- The ***Device Signature (Chip ID)*** is displayed e.g. ***0xD0940***
- The full ***JTAG ID*** is displayed: e.g. ***0x3F0F0F0F***
- The ***JTAG ID*** is then split into the ***'Silicon Revision'***, ***'Part Number'*** and ***'Manufacturer'***. This is not relevant for AT91SAM7 devices. The ***'Chip ID'*** holds this information instead.

This message means that the programmer has established a connection via the JTAG interface to the specified target AT91SAM7 device and that the device has the correct ***'Chip ID'*** as specified in the device library.

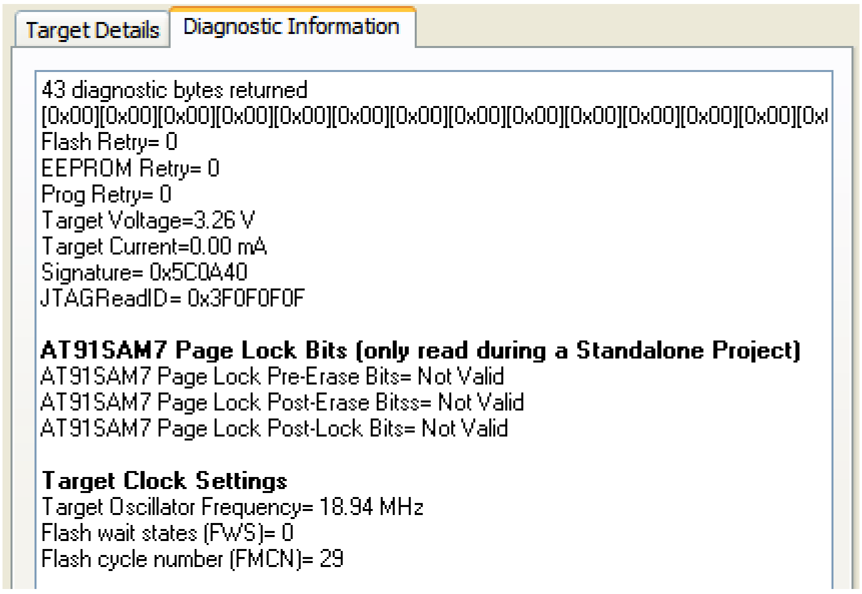
4.5.2 Diagnostic Info

Every time the programmer enters programming mode, it will return detailed diagnostic information about the target device. This information includes the Target Voltage, oscillator frequency and FLASH timings.

To view the *'Diagnostic information'*:

- Click the ***<Diagnostic Info>*** button on any EDS screen
- Select the ***<Diagnostic Information>*** tab

→ The diagnostic information is displayed as shown below…..



```
43 diagnostic bytes returned
[0x00][0x00][0x00][0x00][0x00][0x00][0x00][0x00][0x00][0x00][0x00][0x00][0x00][0x
Flash Retry= 0
EEPROM Retry= 0
Prog Retry= 0
Target Voltage=3.26 V
Target Current=0.00 mA
Signature= 0x5C0A40
JTAGReadID= 0x3F0F0F0F

AT91SAM7 Page Lock Bits (only read during a Standalone Project)
AT91SAM7 Page Lock Pre-Erase Bits= Not Valid
AT91SAM7 Page Lock Post-Erase Bitss= Not Valid
AT91SAM7 Page Lock Post-Lock Bits= Not Valid

Target Clock Settings
Target Oscillator Frequency= 18.94 MHz
Flash wait states (FWS)= 0
Flash cycle number (FMCN)= 29
```

Please note:
The detailed ***'Diagnostic Information'*** requires that firmware 5.15 or above is installed in the programmer.

4.5.3 Possible failure messages

The action of performing a ***<Check ID>*** can produce any of the following error messages:

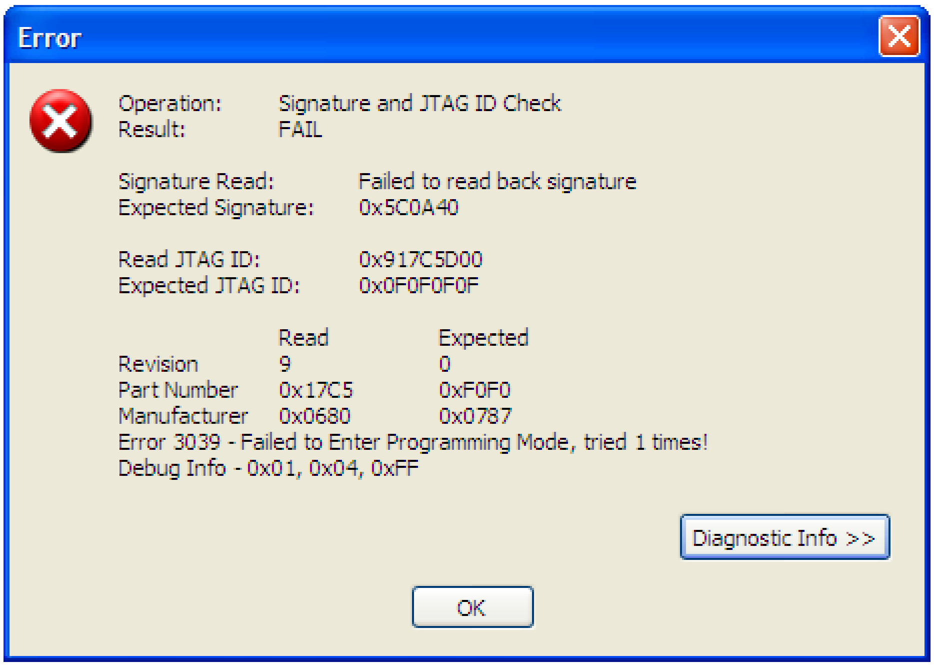
i. Error 3039 / 3044 – Failed to enter programming mode

ii. Error 44 / 3041 – Signature failure: Read back: ***0x??????*** Expected***: 0x??????***

These errors are discussed in the next two sections.

4.5.4 Error 3044 / 3039 – Failed to enter programming mode

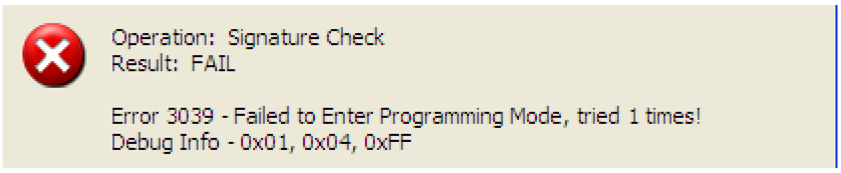
This error message means that the programmer attempted to place the target AT91SAM7 device into programming mode, but the operation failed for some reason.



```
Operation:     Signature and JTAG ID Check
Result:        FAIL

Signature Read:       Failed to read back signature
Expected Signature:   0x5C0A40

Read JTAG ID:         0x917C5D00
Expected JTAG ID:     0x0F0F0F0F

               Read          Expected
Revision       9             0
Part Number    0x17C5        0xF0F0
Manufacturer   0x0680        0x0787
Error 3039 - Failed to Enter Programming Mode, tried 1 times!
Debug Info - 0x01, 0x04, 0xFF
```

This could be due to many things including incorrect JTAG / power wiring, no power on the Target System, JTAG frequency too high, incorrect oscillator source specified etc. etc.

Please note:
The error message also displays 2 or 3 ***'Debug Info'*** bytes.



```
Operation:  Signature Check
Result:  FAIL

Error 3039 - Failed to Enter Programming Mode, tried 1 times!
Debug Info - 0x01, 0x04, 0xFF
```

The first '***Debug Info'*** byte indicates where in the sequence of entering programming mode that the process failed and can be useful in pin-pointing where the problem lies. You can look up what these bytes mean in appendix 2 of this application note.

If you receive this error, please check the following:

Hardware checks…..

- The JTAG connections between the programmer and the Target System are correct.
- There is definitely power applied to the Target System and to all the JTAG devices if the Target Device is part of a JTAG chain. (Please note – The error message will display the Target Voltage – click the ***Diagnostic Info*** button.)
- The ***TST*** pin on the AT91SAM7 device is connected to 0V.
- The ***JTAGSEL*** pin on the AT91SAM7 device is connected to 0V.

- The ***ERASE*** pin on the AT91SAM7 device is connected to 0V.
- Check that the ***RESET*** pin of the AT91SAM7 device is not being held low by an external reset circuit on the Target Board. This will prevent the device from running the debugger code.
- If there is a ***RESET*** button or RESET jumper on the Target Board, try resetting the device.
- Try removing power from the Target Board and then switching power back on again.

EQTools checks…..

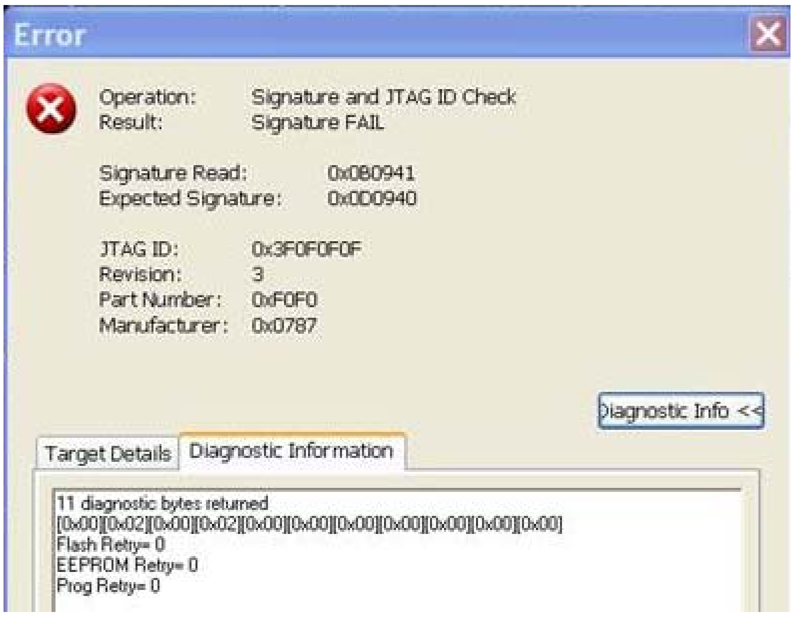
- The ***'Communications Timeout'*** is set to at least **1000ms** in ***the 'Communications and Scripting setup'*** screen.
- The ***'JTAG Chain'*** settings are correct for the Target Device being programmed.
- Try slowing down the ***'JTAG Frequency'*** and then try to check the Device Signature again.

If none of the above suggestions prove to be successful, then the problem could be related to one of the issues listed below:

- The ***"Security Bit"*** in the target AT91SAM7 devices is not already programmed to a **'1'** – see section 4.9 for further details.
- Check that any ***'SAM7 firmware'*** which is already in the FLASH of the SAM7 device does not continuously reset the processor or stop an external programmer from entering debug mode.

4.5.5 Error 41 or 3041 – Signature FAIL message

This error message means that the programmer managed to communicate OK with (connect to) the target AT91SAM7 device via the JTAG interface, but the ***Device Signature (Chip ID)*** read back does not match the ***Chip ID*** in the device library.



```
Operation:      Signature and JTAG ID Check
Result:         Signature FAIL

Signature Read:       0x080941
Expected Signature:   0x0D0940

JTAG ID:        0x3F0F0F0F
Revision:       3
Part Number:    0xF0F0
Manufacturer:   0x0787
```

```
11 diagnostic bytes returned
[0x00][0x02][0x00][0x02][0x00][0x00][0x00][0x00][0x00][0x00][0x00]
Flash Retry= 0
EEPROM Retry= 0
Prog Retry= 0
```

Please note:
To view the full diagnostic information about this error, click the ***<Diagnostic>*** info button.

This message confirms the following:
- The connections between the programmer and AT91SAM7 device are OK.
- The programmer is correctly communicating with (connecting to) the AT91SAM7 device.

If you receive this error, please check the following:
- The problem is simply that the ***'Chip ID' (Signature)*** read back from the AT91SAM7 device does not match the ***'Chip ID'*** in the programming project.
- Check that you have selected the correct AT91SAM7 device in your project.
- Check if the ***'Chip ID'*** read back from the AT91SAM7 device matches one of the ***'Chip IDs'*** from a different silicon revision of the same device. This information can be found in the Atmel device datasheet.
- The ***'JTAG Chain'*** settings are correct for the Target Device being programmed.
- Try slowing down the ***'JTAG Frequency'*** and then try to check the Device Signature again.
- If the Signature looks like a valid ***'Chip ID'***, make sure that you have selected the correct JTAG Device in the chain. It is possible that the programmer is actually communicating with a different device by mistake and hence reading the wrong signature.
- For further diagnostics information, click the ***[Diagnostic Info>>]*** button.

If you still cannot find a device with the correct 'Chip ID':
- Try disabling the ***'Signature Check'*** option on the ***Target Device*** tab.

4.6 Erasing the FLASH area

4.6.1 Overview

The FLASH area must be erased so that each location contains the value 0xFF before any new data can be programmed into it. It is possible to erase the area of a Target Device by clicking the ***<Erase>*** button. This send the ***'Chip Erase'*** command to the target AT91SAM7 device which will then erase the entire FLASH.

Please note:

- The Configuration Fuses (GPNVM fuses) are not affected by a Chip Erase operation.
- The Erase operation will automatically unlock any locked sectors of the FLASH memory.

4.6.2 Erasing the FLASH area

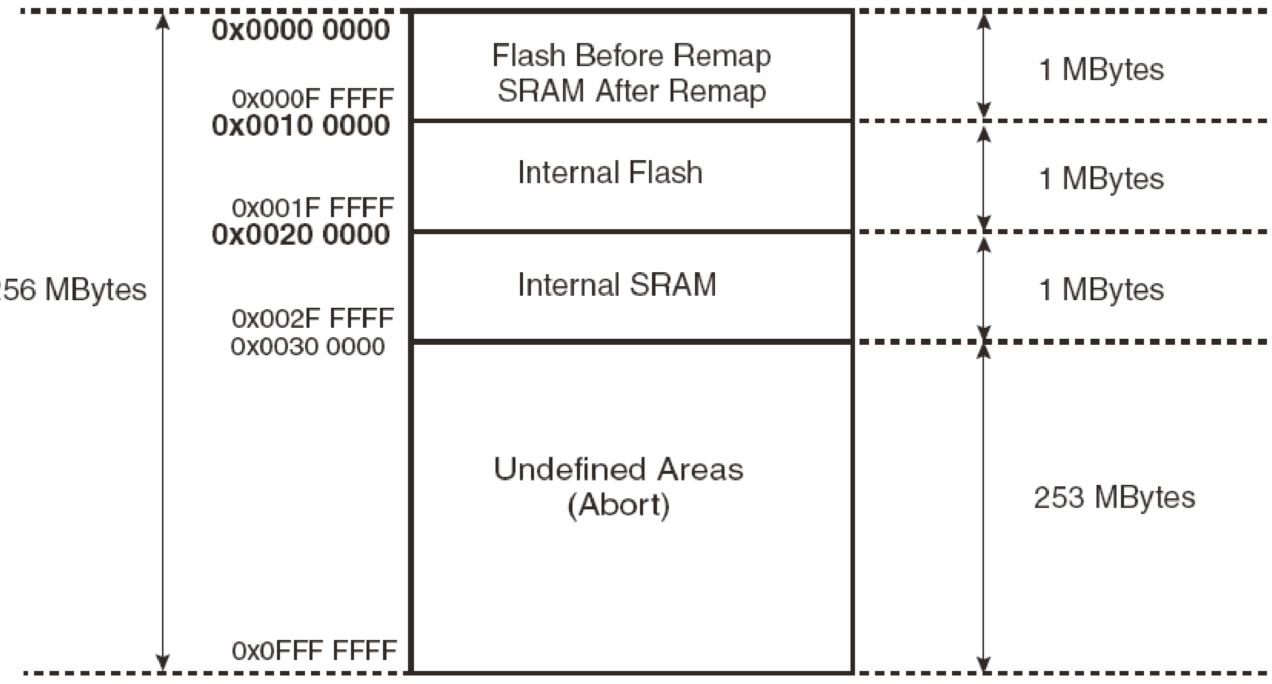
The only way to erase the FLASH area of the Target Device under programmer control is to use the ***'Chip Erase'*** command:

- Select the **<FLASH>** tab
- Click the **<Erase>** button
- This will send the ***'Chip Erase'*** command to the Target Device.
- The Target Device will then erase the FLASH
- To confirm that the FLASH area is definitely blank, you can choose to perform a ***'Blank Check'*** operation.

4.7 Programming the FLASH Area

4.7.1 Overview of AT91SAM7 Memory Map

The ***'Internal FLASH Memory'*** of an AT91SAM7 device can be mapped to two different address ranges. EQTools expects the start address of data in the ***'INTEL HEX'*** or ***'Motorola S-Record'*** file to be 0x000000. This is the default value before the device remaps the FLASH – see memory map diagram below.



Important note:
If the start address of the data in the ***'Intel Hex'*** or ***'Motorola S-Record'*** input file is ***0x100000*** then it will be necessary to resave the file with an offset of ***0x000000.***

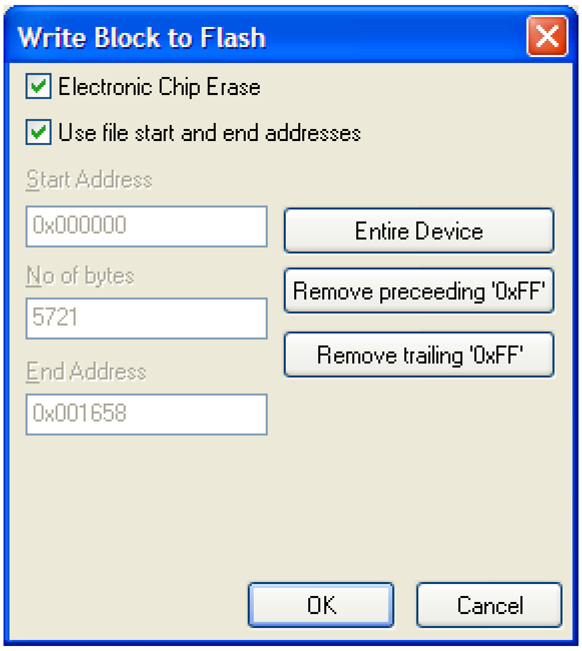
4.7.2 Instructions to program the FLASH area

The instructions detailed below describe how to program the contents of a file into the FLASH area of the Target AT91SAM7 device:

- Select the **<FLASH>** tab
- If you have not already selected a data file to program, click the ***'Edit buffer'*** check box and then click the **<Load>** button to select a suitable ***Binary*** or ***Intel Hex*** file.

Important note:
If the start address of the data in the ***'Intel Hex'*** or ***'Motorola S-Record'*** input file is ***0x100000*** then it will be necessary to resave the file with an offset of 0x000000.

- The contents of the specified file should now be displayed in the Buffer Window.
- Click the **<Write>** button


Write Block to Flash

☑ Electronic Chip Erase
☑ Use file start and end addresses

Start Address: 0x000000
No of bytes: 5721
End Address: 0x001658

- EDS will automatically perform a ***Chip Erase*** by default which will erase the entire FLASH before programming any data into it.
- Select the address range you wish to program.
- EDS will automatically use the ***'Start'*** and ***'End'*** address of the FLASH input file unless otherwise specified. This reduces the total data actually programmed to the number of bytes in the input file rounded to the end of the nearest FLASH Page.
- If you want to program the entire FLASH range, click the **<Entire Device>** button.
- Click **<OK>** to program the FLASH of the Target Chip.
- The programmer should now start to program the chip.
- If any of the AT91SAM7 ***'Sector Lock Bits'*** are enabled, the programmer will automatically disable them so the relevant pages of FLASH can be programmed.
- The BUSY LED will illuminate on the programmer.

- The programmer will program the contents of the Buffer Window into the FLASH area of the Target Device.
- Each block of data is programmed and then verified so if a failure occurs it will be notified immediately.
- To verify that the data has been programmed correctly, click the **<Verify>** button.

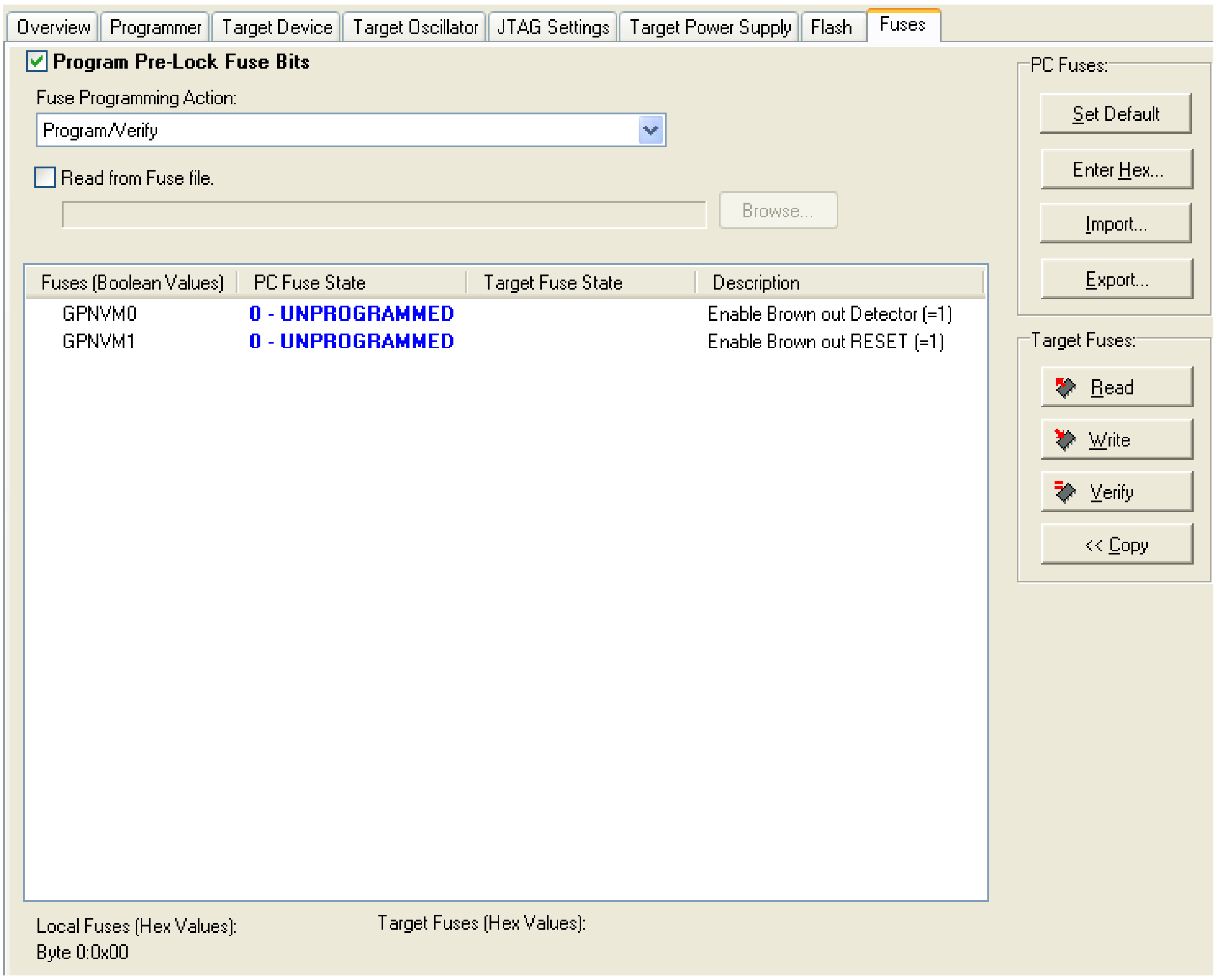
4.8 Programming the Configuration Fuses

4.8.1 Overview

The Configuration Fuses of an Atmel AT91SAM7 device can be programmed / read using the ***<Fuses>*** tab.

Instructions:

- Select the ***<Fuses>*** tab
- If this is a new EDS project, then the Fuses will be disabled.
- Check the ***'Program Pre-Lock Bits'*** box → the Fuses can now be programmed



☑ Program Pre-Lock Fuse Bits

Fuse Programming Action: Program/Verify

☐ Read from Fuse file.

Fuses (Boolean Values)	PC Fuse State	Target Fuse State	Description
GPNVM0	0 - UNPROGRAMMED		Enable Brown out Detector (=1)
GPNVM1	0 - UNPROGRAMMED		Enable Brown out RESET (=1)

Local Fuses (Hex Values):
Byte 0:0x00

Target Fuses (Hex Values):

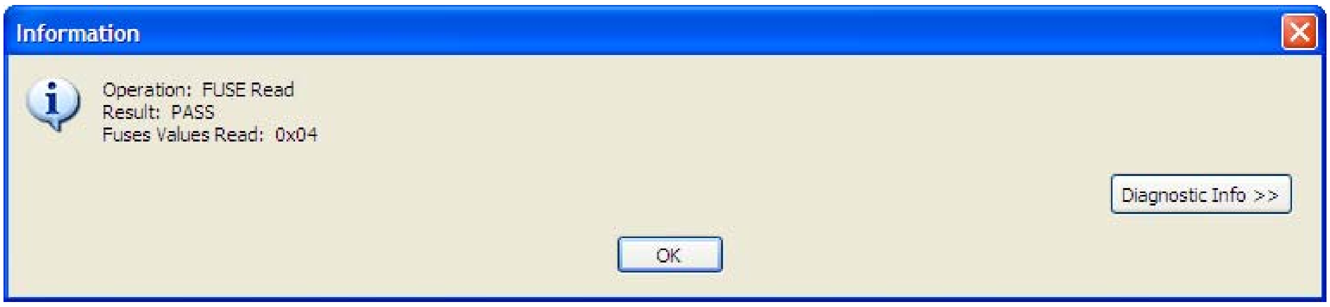
- The values of the Fuses which could be programmed into the Target Chip are shown in the ***'PC Fuse State'*** column. The initial Fuse values are the default Fuse values for a virgin chip.
- The ***'Target Fuse State'*** column displays the current value of the Fuses of the actual Target Device. They are initially set to be blank or ***'?'*** until the first read or write operation is performed.
- The Fuse Hex values are shown for the ***'PC Fuse State'*** at the bottom of the screen.

- A red 'x' next to a fuse indicates a 'Dangerous Fuse'. Programming one of these fuses incorrectly could result in the chip no longer responding to the programmer.

4.8.2 Reading the Fuses from a Target Device

To read the Fuse Values from a Target Device:

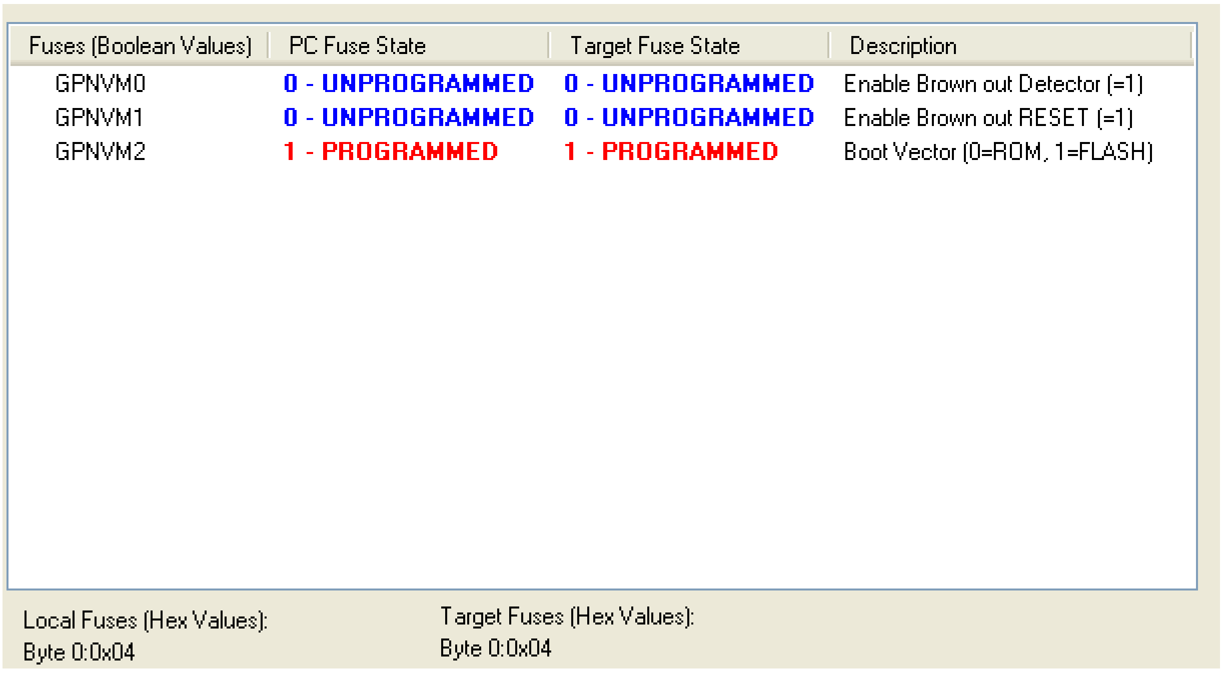
- Click the **<Read>** button
- The Hex values of the 'Fuse Bytes' which are read back are displayed as follows:



Information

Operation: FUSE Read
Result: PASS
Fuses Values Read: 0x04

- The Fuse values from the Target Device are now displayed in the ***'Target Fuse State'*** column.



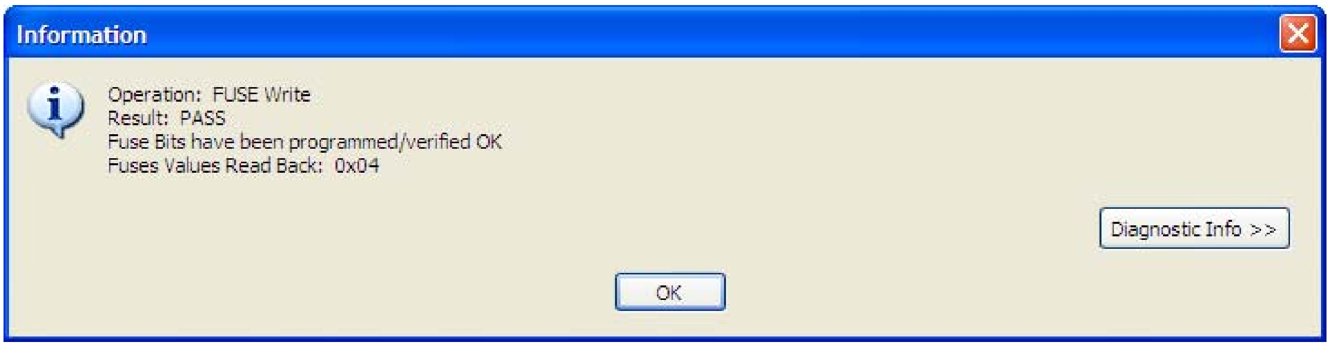
Fuses (Boolean Values)	PC Fuse State	Target Fuse State	Description
GPNVM0	0 - UNPROGRAMMED	0 - UNPROGRAMMED	Enable Brown out Detector (=1)
GPNVM1	0 - UNPROGRAMMED	0 - UNPROGRAMMED	Enable Brown out RESET (=1)
GPNVM2	1 - PROGRAMMED	1 - PROGRAMMED	Boot Vector (0=ROM, 1=FLASH)

Local Fuses (Hex Values):
Byte 0:0x04

Target Fuses (Hex Values):
Byte 0:0x04

4.8.3 Writing the Fuses into a Target Device

- To program the Fuses values in the ***'PC Fuse State'*** column into the Target Device, click the **<Write>** button
- The programmer will then report a PASS or FAIL for programming the Fuses.



Information

```
Operation:  FUSE Write
Result:  PASS
Fuse Bits have been programmed/verified OK
Fuses Values Read Back:  0x04
```

Warning!

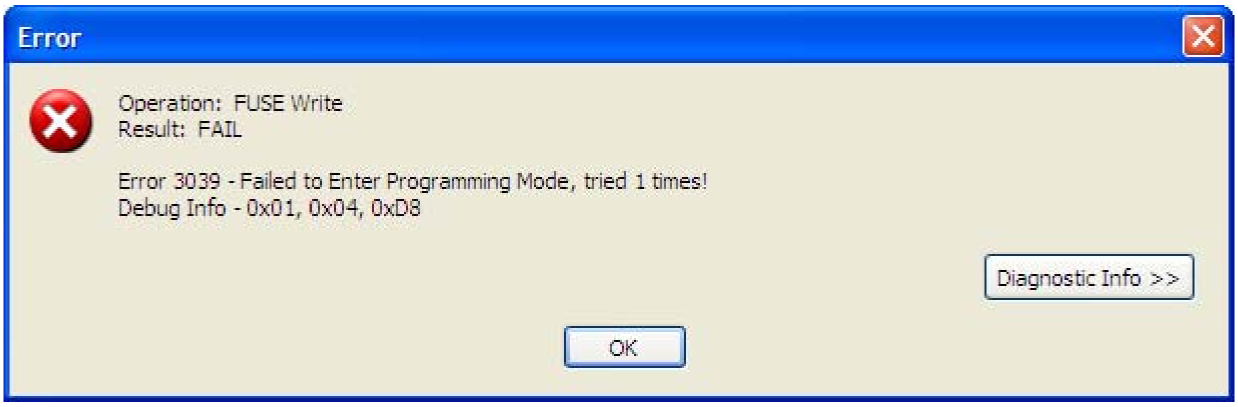
- Atmel state that for many of the AT91SAM7 microcontrollers the ***'GPNVM Fuse Bits'*** can only be programmed up to **a maximum of 100 times**.
- To avoid 'wearing out' of the ***'GPNVM Fuse Bits'***, it is therefore recommended that the fuses are only programmed when absolutely necessary.

4.8.4 Fuse Write Failure messages

The Fuse Write operation can fail for the following reasons:

- The Target Device will not enter programming mode (Error 3039).
- One of the fuses will not program for some reason. E.g. The fuses have 'worn out' because they have been programmed more than 100 times.

If you receive ***'Error 3039 – Failed to enter programming mode'***, please refer to section 3.7.3 for help.



Error

```
Operation:  FUSE Write
Result:  FAIL

Error 3039 - Failed to Enter Programming Mode, tried 1 times!
Debug Info - 0x01, 0x04, 0xD8
```

4.8.5 Verifying the Fuses of a Target Device

To verify the Fuse Values in a Target Device with the Fuse Values in the ‘PC Fuse State’ column:

- Click the **<Verify>** button
- Any differences in the Fuse Values between the ‘PC Fuse State’ settings and the ‘Target Fuse State’ settings will now be displayed.

4.8.6 Using a ‘Fuse File’ to import Fuse settings into a project

It is possible to export the ***‘Fuse Values’*** for a particular device to a so-called ***‘Fuse File’*** so that a single copy of the fuses is stored in one place. This ***‘Fuse File’*** can then be shared amongst many projects if required. See section 4 for further details about using ***‘Fuse Files’*** .

4.9 Programming the AT91SAM7 “Security Bit”

4.9.1 Overview

The AT91SAM7 microcontroller families all feature a single ***“Security Bit”***. Once this fuse it enabled (set to ‘1’), then the AT91SAM7 device will no longer respond to any command sent to the device from the programmer. This allows you to lock the AT91SAM7 device so that it is then no longer possible to read the program data from the device or to write an further data to the device.

Warning!

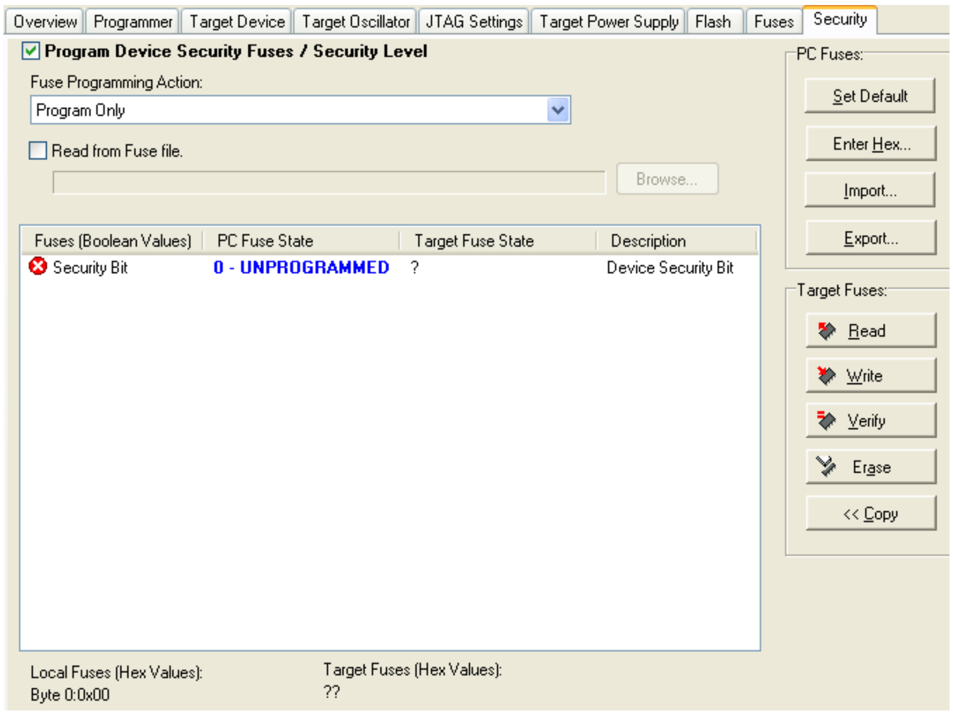
- If the ***“Security Bit”*** of the AT91SAM7 device is programmed to a ‘1’, then the JTAG Port of the device is then disabled. The device will no longer respond to any JTAG programmer which is connected to the JTAG Port. This means that you cannot reprogram the device again!!!
- If you require a method to “unlock” an AT91SAM7 device once the ***“Security Bit”*** has been programmed, then it is recommended that provision for a link on the ERASE pin to both Vcc and GROUND is made.

4.9.2 Programming the “Security Bit”

The instructions below detail how to enable the AT91SAM7 ***“Security Bit”***:

Instructions:

- Select the **<Security Bits>** tab
- If this is a new EDS project, then the ‘Security Fuses’ will be disabled.
- Check the ***‘Program Device Security Fuses’*** box → the Security Fuses can now be programmed



☑ Program Device Security Fuses / Security Level

Fuse Programming Action: Program Only

☐ Read from Fuse file.

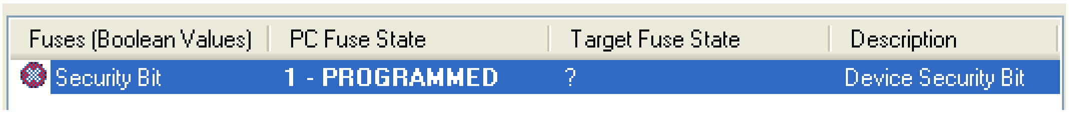
Fuses (Boolean Values)	PC Fuse State	Target Fuse State	Description
Security Bit	0 - UNPROGRAMMED	?	Device Security Bit

Local Fuses (Hex Values):
Byte 0:0x00

Target Fuses (Hex Values):
??

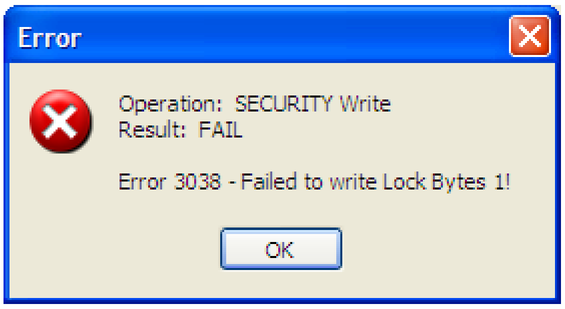
- Set the ‘Fuse Programming Action’ to ‘Program only’

- Double-click the ‘’ fuse to change its value to ‘1’



Fuses (Boolean Values)	PC Fuse State	Target Fuse State	Description
Security Bit	1 - PROGRAMMED	?	Device Security Bit

Click the ***‘Write’*** button to program the “Security Bit” to ‘1’.
→ The programmer can no longer communicate with the target AT91SAM7 device.
→ A ‘Security Write’ error message is displayed.



Error

Operation: SECURITY Write
Result: FAIL

Error 3038 - Failed to write Lock Bytes 1!

- The Target Device is now locked
- It is no longer possible for the programmer to communicate with the target AT91SAM7 device.

4.9.3 Recovering a AT91SAM7 device

Once the AT91SAM7 ***“Security Bit”*** has been programmed to a ‘1’, then the target device will no longer respond to any command sent by the programmer via the JTAG port.

The only way to recover the AT91SAM7 device is as follows:

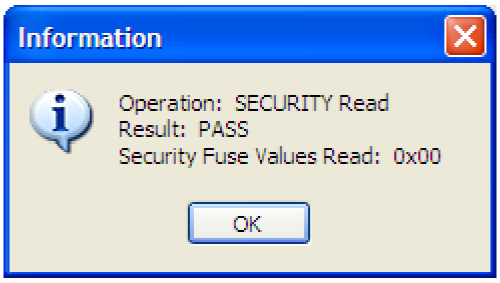
- Connect the ERASE pin of the AT91SAM7 device to Vcc (e.g. 3.3V) for > 300ms
- Re-connect the Erase pin to 0V

→ The AT91SAM7 device has now been erased and the ***“Security Bit”*** has been set back to a ‘0’.
→ The programmer should now be able to communicate with the AT91SAM7 device again.

4.9.4 Reading the "Security Bit" from a Target Device

To read the Security Bit Values from a Target Device:

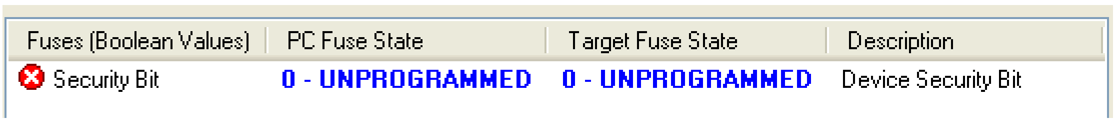
- Click the **<Read>** button
- The Hex values of the 'Fuse Bytes' which are read back are displayed as follows:



Information

Operation: SECURITY Read
Result: PASS
Security Fuse Values Read: 0x00

- The Security Bit values from the Target Device are now displayed in the ***'Target Fuse State'*** column.



Fuses (Boolean Values)	PC Fuse State	Target Fuse State	Description
Security Bit	0 - UNPROGRAMMED	0 - UNPROGRAMMED	Device Security Bit

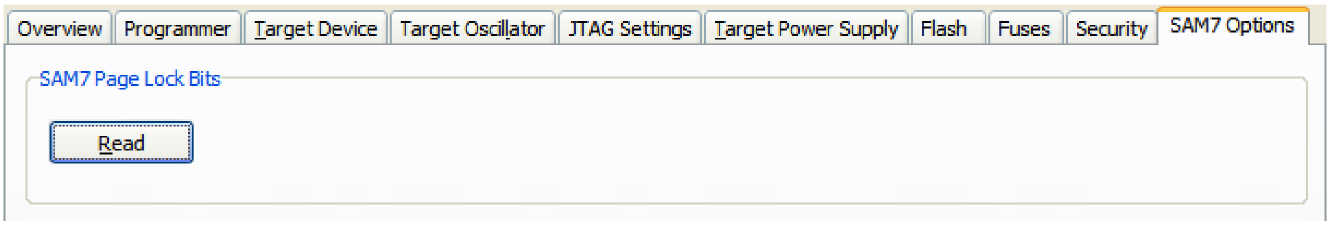
Please note:
If the ***"Security Bit"*** has been programmed to a ***'1'***, then reading back of the fuse will fail as the device is now locked.

4.10 Reading the FLASH Page Lock Bits

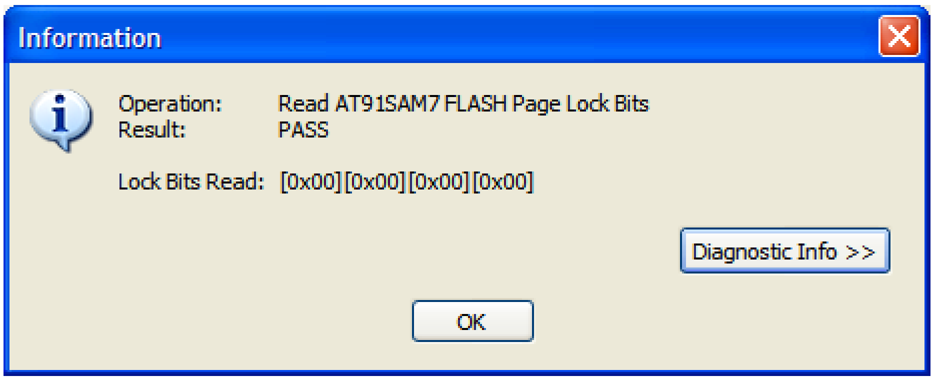
The Atmel AT91SAM7 devices feature special ***'FLASH Page Lock Bits'*** which are used to lock certain pages / sectors of the FLASH memory area. EQTools supports reading back of these bits and displays them as hex bytes.

To read the ***'FLASH Page Lock Bits'***:

- Select the ***<SAM7 Options>*** tab in EDS



- Click the ***<Read>*** button
- The ***'FLASH Page Lock Bits'*** are now displayed as four hex bytes.



```
Operation:      Read AT91SAM7 FLASH Page Lock Bits
Result:         PASS

Lock Bits Read: [0x00][0x00][0x00][0x00]
```

Some devices feature 16 x ***'FLASH Page Lock Bits'*** and other feature 32 x ***'FLASH Page Lock Bits'*** Lock Bits. The hex bytes are displayed in lowest address first.

Please note:

- Some devices feature 16 x ***'FLASH Page Lock Bits'*** and other feature 32 x ***'FLASH Page Lock Bits'*** Lock Bits.
- The hex bytes are displayed in lowest address first.
- If the AT91SAM7 device is a virgin (never programmed before) device then all lock bits should be '0'.
- This functionality requires that firmware version 5.14 or above is installed in the programmer.
- It is NOT possible to write the ***'FLASH Page Lock Bits'*** using EQTools. This should be handled by your own AT91SAM7 firmware.

4.11 Exporting an EDS Project to a Standalone Project

Once you have fully tested your EDS Development Project, it is possible to add the project to a ***Project Collection*** and then upload it to a programmer as a so-called ***'Standalone Project'***. The project can then be executed on a programmer without requiring any form of PC control.

Please follow the instructions detailed in Section 6 to upload your EDS project to a programmer.

5.0 Exporting / Importing Fuse Settings to / from File

5.1 Overview

One of the new power features of EQTools is the ability to export the ***'Fuse Settings'*** for a Programming Project to a ***Fuse File (*.eff)***. This allows the settings for all the fuses to be contained in one Fuse File which can then be imported into any of the Fuse tabs in Project Manager or in the ***<Fuses>*** tab in EDS. In this way, the values of all the Fuses for a particular project can be shared with other projects. This helps to ensure that the correct fuse values are specified in all projects.

5.2 Exporting the Fuse Settings to a Fuse File

To export the settings of the 'Local Fuses' column to a fuse File:

- Select the EDS ***<Fuses>*** Tab
- Set up the ***'Local Fuses'*** to the correct values for your Target Device.
- Click the ***<Export>*** button → a file browser is displayed.
- Enter a suitable name for your Fuse File e.g. ***project_fuses.eff***
- Click ***<Save>*** → The ***'Local Fuses'*** column is transferred to your specified ***Fuse File (*.eff)***.

5.3 Copying the Fuses from a Target Device

To copy the Fuses from the Target Device to a Fuse File:

- Select the EDS ***<Fuses>*** Tab
- Click the ***<Read>*** button
  → the Fuses are read from the Target Device and are then displayed in the ***'Target State'*** column.
- Click the ***<<Copy*** button
  → the Fuse settings read from the Target Device are copied into the ***'Local State'*** fuse column.
- Click the ***<Export>*** button → a file browser is displayed.
- Enter a suitable name for your Fuse File e.g. ***project_fuses.eff***
- Click ***<Save>*** → The ***'Local Fuses'*** column is transferred to your specified ***Fuse File (*.eff)***.

5.4 Importing the Fuse Settings from a Fuse File

To import the settings of the ***'Local Fuses'*** column from a Fuse File:

- Select the EDS ***<Fuses>*** Tab
- Click the ***<Import>*** button → a file browser appears
- Browse to and select your ***Fuse File (*.eff)***
  → The Fuse settings are then automatically copied from the ***Fuse File*** to the ***'Local Fuses'*** column.
- To program these settings into a Target Device, click ***<Write>.***

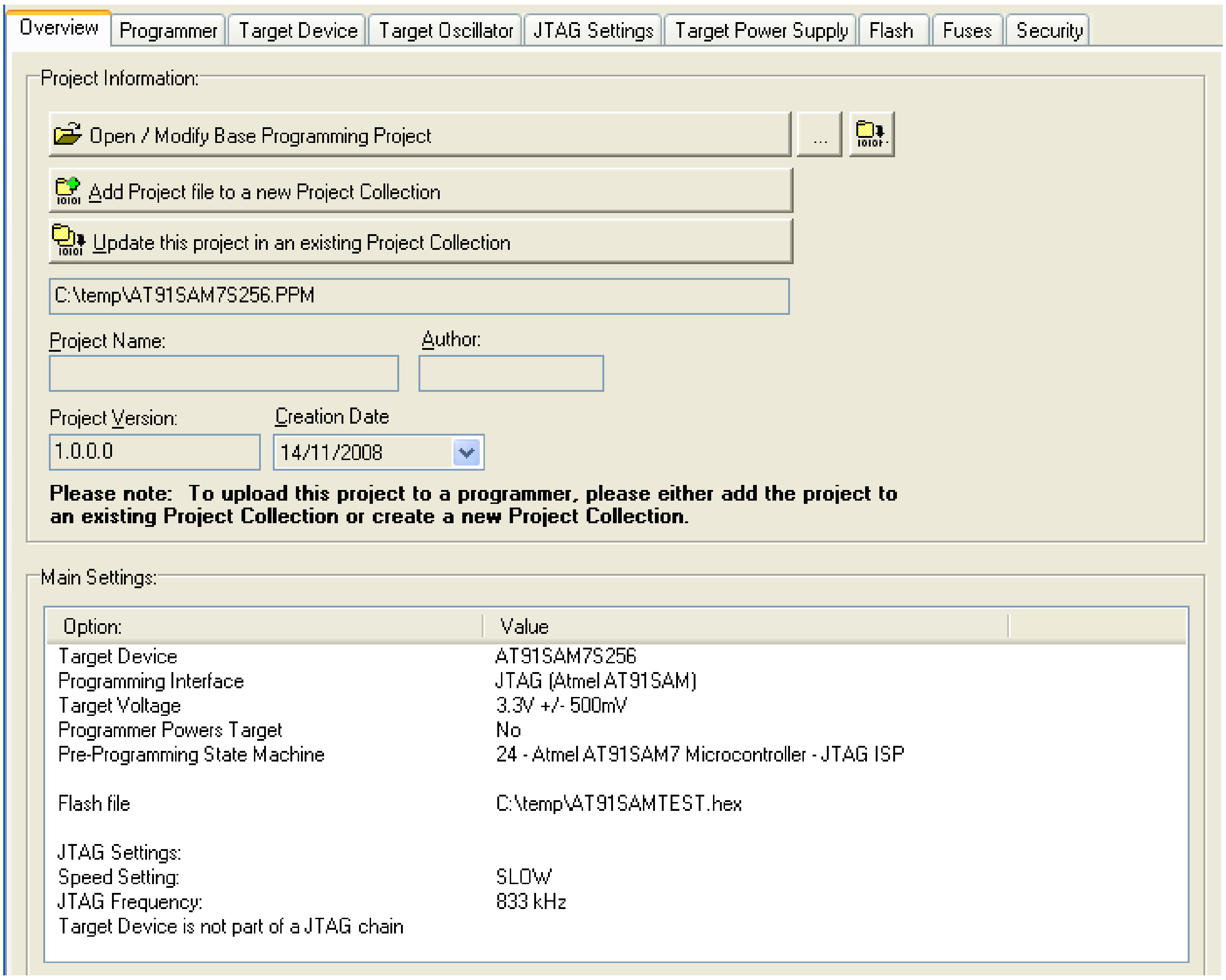
6.0 Creating a Standalone Programming Project

6.1 Overview

Once you have tested your project fully in ***EDS*** (Development Mode), it is possible to then make this project into a ***'Standalone Project'*** which can be uploaded to a programmer. This single standalone project file (*.prj) will contain all the information required to program the Target Device including FLASH file, EEPROM file, Fuse settings, Security Settings etc.

6.2 Creating a Standalone Project from EDS (Development Mode)

In EDS (Development Mode), select the ***<Overview>*** tab


Overview
Programmer
Target Device
Target Oscillator
JTAG Settings
Target Power Supply
Flash
Fuses
Security
Project Information:
Open / Modify Base Programming Project
...
Add Project file to a new Project Collection
Update this project in an existing Project Collection
Main Settings:

C:\temp\AT91SAM7S256.PPM

Project Name:	
Author:	
Project Version:	1.0.0.0
Creation Date	14/11/2008

Please note: To upload this project to a programmer, please either add the project to an existing Project Collection or create a new Project Collection.

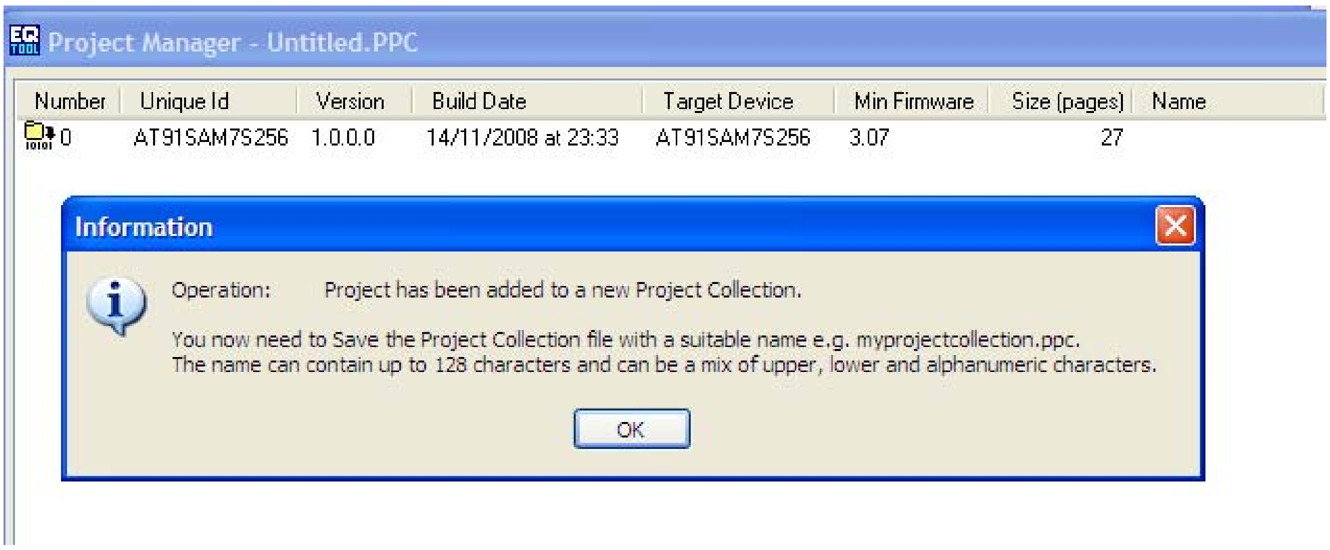
Option:	Value
Target Device	AT91SAM7S256
Programming Interface	JTAG (Atmel AT91SAM)
Target Voltage	3.3V +/- 500mV
Programmer Powers Target	No
Pre-Programming State Machine	24 - Atmel AT91SAM7 Microcontroller - JTAG ISP
Flash file	C:\temp\AT91SAMTEST.hex
JTAG Settings:	
Speed Setting:	SLOW
JTAG Frequency:	833 kHz
Target Device is not part of a JTAG chain	

- If this is the first time the EDS Project has been uploaded to a programmer, click the ***<Add Project File to a new Project Collection>*** button.
- If the EDS Project has already been uploaded to a programmer before, click the ***<Update this project in an existing Project Collection>*** button.

6.3 Add Project File to a new Project Collection

When the ***<Add Project File to a new Project Collection>*** button is pressed, the EDS project will be automatically added to a new ***'Project Collection'***.

- The EDS Project will appear in a ***'Project Manager'*** window.



Number	Unique Id	Version	Build Date	Target Device	Min Firmware	Size (pages)	Name
0	AT91SAM7S256	1.0.0.0	14/11/2008 at 23:33	AT91SAM7S256	3.07	27	

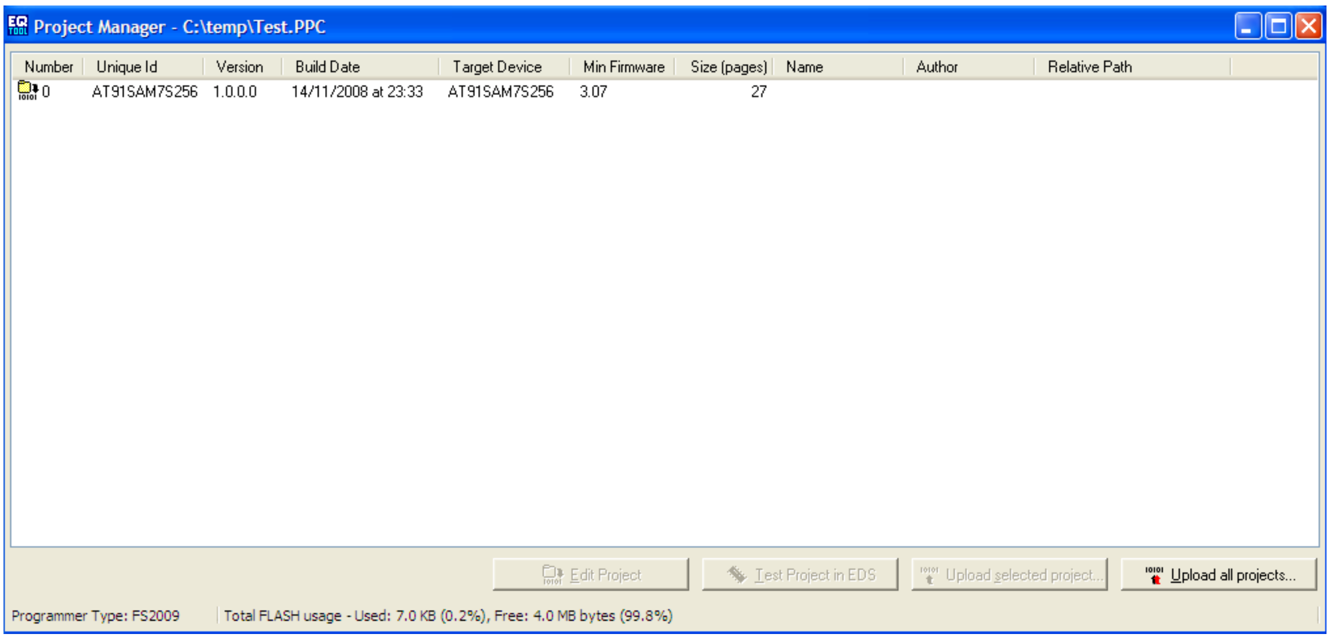
Operation: Project has been added to a new Project Collection.

You now need to Save the Project Collection file with a suitable name e.g. myprojectcollection.ppc.
The name can contain up to 128 characters and can be a mix of upper, lower and alphanumeric characters.

- You will then be prompted to save the ***'Project Collection'***.
- Choose a suitable name e.g. ***Test.ppc*** and click the ***<Save>*** button.
- The Project Manager window is now displayed – see section 6.4.

6.4 Uploading a Project to a programmer

The ***Project Manager*** window displays all the projects in your ***Project Collection***.


Project Manager - C:\temp\Test.PPC
Edit Project
Test Project in EDS
Upload selected project...
Upload all projects...

Number	Unique Id	Version	Build Date	Target Device	Min Firmware	Size (pages)	Name	Author	Relative Path
0	AT91SAM7S256	1.0.0.0	14/11/2008 at 23:33	AT91SAM7S256	3.07	27			

Programmer Type: FS2009 | Total FLASH usage - Used: 7.0 KB (0.2%), Free: 4.0 MB bytes (99.8%)

In this example we have only one project called ***'AT91SAM7S256'***.

The ***'Unique ID'*** is the ***'Project Name'*** which is also the file name you saved the project with in EDS.

To upload the project to the programmer:

- Click the ***<Upload all projects>*** button
  → uploads all the projects in the collection to the programmer.

or

- Click once on the project you wish to upload in the ***Project Manager*** window and then click the ***<Upload selected project>*** button
  → uploads only the selected project in the collection to the programmer.

Follow the on-screen ***Upload Wizard*** instructions to complete the uploading of the project(s) to the programmer(s).

6.5 Re-testing a Project in EDS (Development mode)

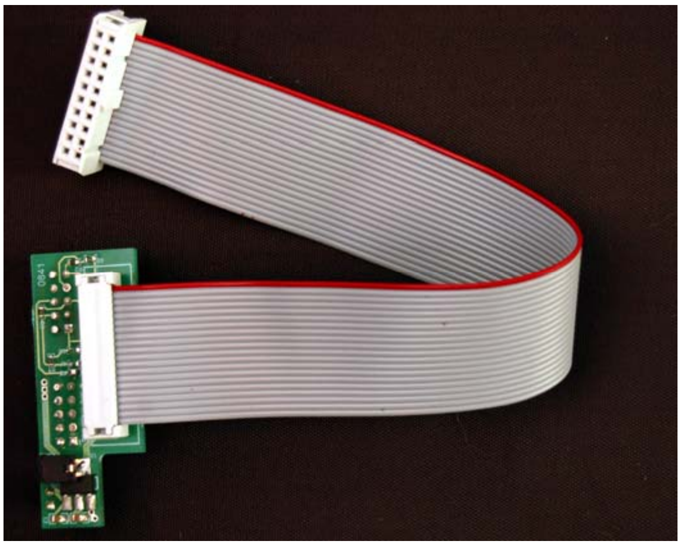
If you want to re-test a Programming Project in EDS (Development Mode), the simplest method to do this is as follows:

- Use ***Project Manager*** to open your ***Project Collection (*.ppc file)***
- Click once on the project which you wish to test in EDS mode. This will select the project.
- Click the ***<Test in EDS>*** button
  → The selected project will now be opened in EDS (Development Mode).
  → You can now test your project in EDS (Development Mode).

Appendix 1 – AT91SAM7 ISP Cable for Epsilon5, FS2003 and FS2009 programmers

1.0 Overview

It is necessary to use a special ***'AT91SAM7 ISP Cable'*** to connect between an Epsilon5, FS2003 or FS2009 programmer and an AT91SAM7S ***'Target System'***. This cable converts the programmer pin-out to the standard 20-way ARM JTAG pin-out suitable for plugging into a 20-way IDC socket on an AT91SAM7 Target System. It also provides a safe way of powering the Target System from the programmer at 3.3V if required.



1.1 Features

- Compatible with Equinox Epsilon5, FS2003 and FS2009 programmers
- Converts the programmer pin-out to the standard 20-way IDC ARM JTAG connector suitable for plugging into an AT91SAM7S Target Board
- Supports powering of the Target System with a regulated 3.3V supply from the programmer
- Supports powering of the programmer from a Target System running at 3.0 – 5.V
- Provides 47k ohm pull-up resistors on the JTAG signal lines

1.2 Programmer compatibility

The ***'AT91SAM7 ISP Cable'*** is compatible with the following Equinox programmers:

- Epsilon5 MK2, Epsilon5 MK3
- FS2003
- FS2009

1.3 Power Supply - Selection Jumper

If you have version ***'V1.1'*** of the ***'AT91SAM7 ISP Cable'*** then you will find a ***'3-way Jumper Link'*** on the circuit board. This jumper is used to configure how the Programmer and Target System are powered. The jumper is labelled ***'IN'*** and ***'OUT'*** on the actual PCB. Please refer to the table below to select the correct powering method for your application.

Link pins	Link pos	Powering scenario	Voltage (V)
1-2	IN	**Target System powers the Programmer** • The Target System voltage is fed directly to the programmer and will be used to power the programmer.	3.0 – 5.0V
2-3	OUT	**Programmer powers the Target System at 3.3V (default)** • The programmer must be powered from an external power supply via the DC Jack Socket in the range of 3.5 to 5.0V. • A 3.3V regulator on the ***'AT91SAM7 ISP Cable'*** then feeds 3.3V down to the Target System.	3.3V

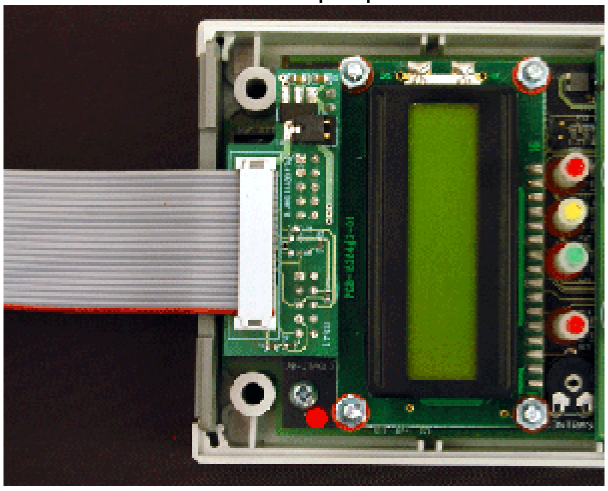
Warning!
If you select link position ***'IN'*** and then input +5V into the DC Jack Socket of the programmer, this will feed +5V directly to the Target System. This +5V could damage circuitry on the Target System if it is only designed to run at 3.3V.

1.4 Cable Installation Instructions

These instructions detail how to fit the***'AT91SAM7 ISP Cable'*** to an Epsilon5, FS2003 or FS2009 programmer.

Instructions:

- The ***'SAM7 Programming Cable'*** has a small circuit board on one end which plugs into the ISP headers inside the programmer. On the other end it features a 20-way IDC plug.
- Remove the top cover of the programmer.
- Make sure the both the programmer and your Target System are powered OFF.
- Plug the circuit board end of the ***'SAM7 Programming Cable'*** into the programmer ISP headers – see example picture of cable fitted to FS2003 / FS2009 programmer below.



- Make sure that the 2 x 10-way connectors are correctly aligned so that the 2-way header J9 aligns with the 2-way header on the cable.
- Decide on how you wish to power the programmer and Target Board – set the ***'Power Supply – Selection Jumper'*** accordingly – see section 1.3
- You are now ready to program an AT91SAM7 device via JTAG.

1.5 Getting Started

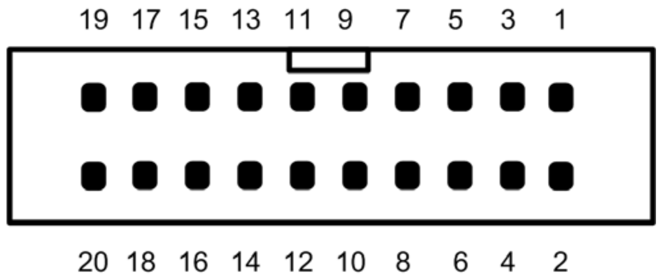
Once you have the installed into the programmer and connected to your Target System, then you are ready to program an AT91SAM7 device.

Instructions:

- Check that you have selected the correct position for the ***'Power Supply – Selection Jumper'*** – see section 1.3
- To be on the safe side, measure the voltage on pin 1 of the 20-way IDC connector. If the programmer is powering the Target System, then this voltage should 3.3V.
- Connect the ***'SAM7 Programming Cable'*** to the 20-way IDC connector on your SAM7 Target System
- Power up the programmer first → this makes sure that the programmer Line Drivers are powered BEFORE you power up the Target Board.
- Power up the Target Board
- Check that the programmer and Target Board power up OK.

1.6 ARM JTAG Debug Header

The pin-out of the 20-way IDC connector end of the cable is shown in the diagram below.

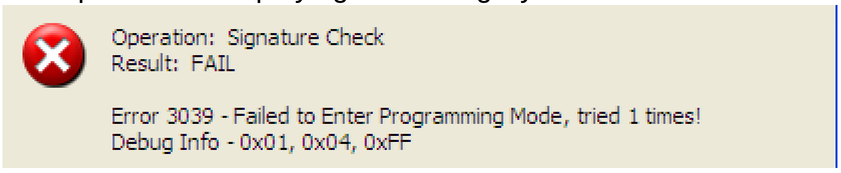
	***ARM JTAG Debug Header*** The connector is a 20-pin bump-polarised IDC connector with 0.1" pin spacing. Pin 1 is the top right pin as shown in the diagram opposite.

Appendix 2 – AT91SAM7 Debug Bytes

1.0 Overview

When an Equinox programmer is attempting to enter programming mode with (connect to) an AT91SAM7 ARM device, it must work through a sequence of actions to force the device into the correct mode for JTAG programming. If the programmer cannot connect with an Atmel AT91SAM7 device, it will output a series of ***'Debug Bytes'*** which can be used to help to find out why the connection failed. These ***'Debug bytes'*** are displayed as series of 2 or 3 bytes depending on where in the connection sequence the connection failed.

Example – EDS displaying the Debug Bytes


```
Operation:  Signature Check
Result:  FAIL

Error 3039 - Failed to Enter Programming Mode, tried 1 times!
Debug Info - 0x01, 0x04, 0xFF
```

The first debug byte indicates where in the sequence of entering programming mode that the process failed. The table below details the meaning of each possible value of the first debug byte displayed.

1.1 Explanation of each Debug Byte value

The table below details the meaning of each possible value of the first debug byte displayed.

1st Debug Byte	Corresponding Error Message	See section
0x00	• Failed to halt the target ARM device and read registers.	1.2
0x01	• Failed to change the target ARM device to supervisor mode and optionally from THUMB mode.	1.3
0x02	• Failed to read the target ARM device 'Chip ID' (Signature) immediately after changing to internal clock	1.4
0x03	• Failed to load 'Equinox demon' into the target ARM device internal memory.	1.5
0x04	• Failed to execute 'Equinox demon'	1.6
0x05	• 'Equinox Demon' did not respond within 250ms	1.7
0x06	• Equinox Demon' is not ready to receive a command	1.8
0x07 0x08 0x09	• 'Equinox Demon' did not respond correctly to 'Config command'	1.8

1.2 Debug 0x00 – Failed to halt ARM device

If the first debug byte is 0x00, then this means that the programmer has attempted to communicate with the target ARM device via the JTAG interface, but the programmer was not able to halt the execution of the ARM core.

Possible causes of this error:

- Power supply problems with the AT91SAM7 device.
- Poor signal integrity of the JTAG signals from the programmer
- The JTAG signals from the programmer are not at the same voltage as the AT91SAM7 device is running at.
- The AT91SAM7 device already has some firmware running in it which is somehow locking out the JTAG port e.g. it is sending the AT91SAM7S device into 'Sleep Mode'.
- The AT91SAM7 device is running from its internal 32 kHz oscillator – see errata information from Atmel about this problem.

Check list:

- Check the JTAG connections between the programmer and the AT91SAM7 device.
- Check the voltage on the ***'VDD_IN'*** and ***'VDD_FLASH'*** on the AT91SAM7 device. Is it approximately 3.3V?
- If the programmer should be powering the Target System, check that the ***'Programmer controlled Target System'*** in ON and set to eg. 3.3V.
- Check that the firmware which is being programmed into the AT91SAM7 device does not somehow disable the JTAG port. The JTAG port can be disabled by placing the device in certain 'Sleep Modes'.

1.3 Debug 0x01 – Failed to change ARM to supervisor mode

If the first debug byte is 0x01, then this means that the programmer has communicated with the target ARM device via the JTAG interface, halted the ARM core, but then was not able to switch the ARM to ***'Supervisor Mode'***. This error is usually caused by power supply issues or poor JTAG signal integrity.

Possible causes of this error:

- Power supply problems with the AT91SAM7 device.
- Poor signal integrity of the JTAG signals from the programmer to the AT91SAM7 device.
- The JTAG signals from the programmer are not at the same voltage as the AT91SAM7 device is running at.
- Firmware which has already been programmed into the AT91SAM7 device on a previous programming attempt is constantly resetting the processor or perhaps putting it into a 'Sleep Mode'. This will prevent an external JTAG programmer from communicating with the chip.
- Firmware which has already been programmed into the AT91SAM7 device on a previous programming attempt is switching to an oscillator which is either not fitted on the Target Board or not supported by the Equinox algorithm.

Check list:

- Check the JTAG connections between the programmer and the AT91SAM7 device.
- Check that there is a separate GROUND wire directly from the programmer GROUND (0V) connection of the Programmer I/O Connector to the GROUND of the Target Board. Make sure this is the same GROUND to which the AT91SAM7 is referenced to.
- Check that there are resistor pull-ups on the JTAG TDI, TCK, TMS signals lines to the voltage rail from which the AT91SAM7 device is powered. These pull-ups should preferably be mounted on the Target Board itself. If they are missing, then they should be mounted as close to the Target Board as possible i.e. on the Test Probes.
- Check the integrity of the JTAG signals using an oscilloscope.
- Try reducing the JTAG frequency in the programming project to e.g. 100kHz and see if this fixes the problem. If it does, then it is likely that there are signal reflections on the JTAG lines.
- Check with your 'AT91SAM7 Firmware Developer' to see whether they have enabled the AT91SAM7 RESET (nRST) pin in their firmware.
- Check if the ***nRST*** pin on the AT91SAM7 device (if enabled) is spuriously resetting the processor.
- Check with your 'AT91SAM7 Firmware Developer' to see whether they are doing any form of ***'Software RESET'*** in their code which would prevent an external programmer from communicating with the device.

1.4 Debug 0x02 – Failed to read the target ARM device 'Chip ID'

If the first debug byte is 0x02, then this means that the programmer has communicated with the target ARM device via the JTAG interface, halted the ARM core, switched to ***'Supervisor Mode'*** but was then not able to read the ***'Chip ID'*** from the ARM core. The ***'Chip ID'*** identifies the actual device and also specifies the core type, size of FLASH etc.

Possible causes of this error:

- Same as when debug byte is 0x01.

Check list:

- Same as when debug byte is 0x01.

1.5 Debug 0x03 – Failed to load Equinox demon into ARM

If the first debug byte is 0x03, then this means that the programmer has communicated with the target ARM device via the JTAG interface, halted the ARM core, switched to ***'Supervisor Mode'*** and read / validated the ***'Chip ID'***. It has then attempted to load the ***'Equinox demon'*** into SRAM on in the ARM but this process has failed for some reason.

1.6 Debug 0x04 – Failed to execute 'Equinox demon'

If the first debug byte is 0x04, then this means that the programmer has communicated with the target ARM device via the JTAG interface, halted the ARM core, switched to ***'Supervisor Mode'*** and read /

validated the ***'Chip ID'***. It has successfully loaded the ***'Equinox demon'*** into SRAM in the ARM device but when it has tried to execute the ***'Equinox demon'*** the process has failed.

Check list:

- Check that a valid external oscillator is available on the AT91SAM7 device. This oscillator must be an external crystal or clock signal running at >=1MHz.

1.7 Debug 0x05 – 'Equinox Demon' did not respond within 250ms

If the first debug byte is 0x05, then this means that the programmer has communicated with the target ARM device via the JTAG interface, halted the ARM core, switched to ***'Supervisor Mode'*** and read / validated the ***'Chip ID'***. It has successfully loaded the ***'Equinox demon'*** into SRAM in the ARM device and execution has started. However, when the programmer has tried to communicate with the ***'Equinox demon'*** but no response has been received back within the timeout period of 250ms.

Check list:

- Check that a valid external oscillator is still available / running on the AT91SAM7 device. This oscillator must be an external crystal or clock signal running at >=1MHz.
- Check that there is still power on the AT91SAM7 device.
- Check that the AT91SAM7 device has not rebooted due to eg, an external RESET on the nRST pin.

1.8 Debug 0x06 – 0x09

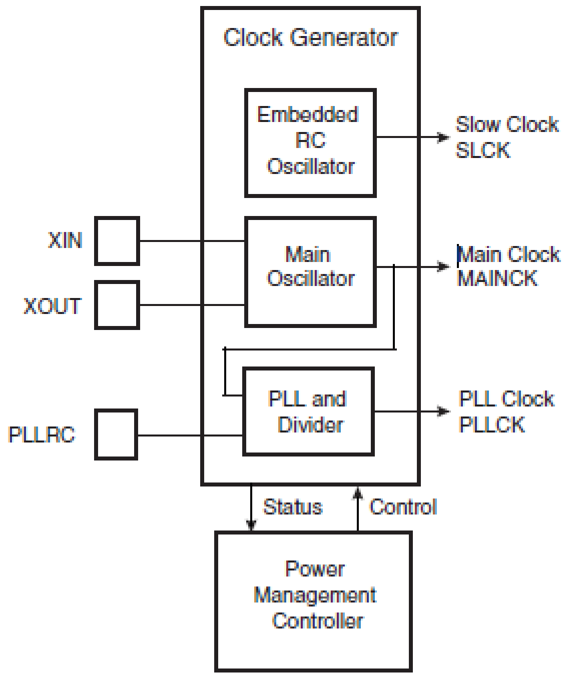
If the first debug byte is 0x06, 0x07, 0x08 or 0x09 then please report this problem to Equinox.

Appendix 3 – Target Oscillator settings

1.0 Overview

The AT91SAM7 microcontroller features a fully user-configurable ***'Clock Generator Module'*** which allows the device to select from various different clock sources including internal oscillator, external crystal and external oscillator signal. The device always starts to run from its ***internal oscillator*** after power-up (Power-On-RESET) which means that it always has a safe clock to run from. It is then up to the user firmware (or an external programmer / debugger) running on the microcontroller to select any other oscillator source. As the ***internal oscillator*** is very slow, most applications demand that a faster external crystal or external oscillator signal is selected.

The block diagram below shows the functional layout of the oscillator selection / routing logic inside an AT91SAM7 device.



An AT91SAM7 device can be clocked from any of the different sources detailed below.

#	Oscillator source	Frequency range	Comment
1	**Internal RC Oscillator**	22 – 46 kHz	• This oscillator is always available as it is an on-chip peripheral. • It can be used for programming but the programming will be very slow.
2	**External Crystal Oscillator**	3 – 20 MHz	• An external crystal oscillator should be connected across the ***Xin*** and ***Xout*** pins.
3	**External Oscillator Signal**	3 – 20 MHz	• An external oscillator signal should be connected to the ***Xin*** pin.

1.1 AT91SAM7 - Oscillator modes

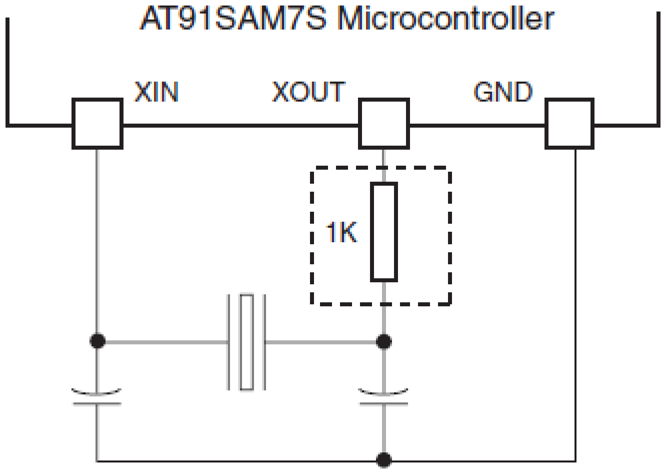
This section gives a brief overview of the different ***'Oscillator Modes'*** which the AT91SAM7 devices support.

1.1.1 Internal RC Oscillator

An AT91SAM7 device initially runs from a very slow ***'internal RC oscillator'*** when it is first powered up (Power-on-RESET) or after a firmware reset. This oscillator is usually in the range of **22 – 46 kHz** and varies slightly with batch, voltage and temperature. This oscillator is always present and so can always be used for programming. However, because the ***'internal RC oscillator'*** is very slow, the programming process will be extremely slow eg. > 3 minutes for a 512 kb FLASH device.

1.1.2 External crystal oscillator

An AT91SAM7 device can be clocked by an **External crystal oscillator** connected across the ***Xin*** and ***Xout*** pins as shown in the diagram below.



The oscillator could be a ***'crystal oscillator'*** or a ***'crystal oscillator module'***.
It will usually have the ***'oscillator frequency'*** printed on it in MHz e.g. 16.00 MHz. You will need to know this frequency as the programmer needs the value so that it can calculate the correct ***'FLASH timing parameters'.***

Please note:
The programmer can measure the frequency of this crystal at the start of a programming operation and can then automatically switch the device to use the crystal. This allows the programmer to program the device at a much faster speed.

1.1.3 External oscillator signal

An AT91SAM7 device can be clocked by an ***external oscillator signal*** connected to the ***Xin*** pin. This oscillator signal could be generated from a self-contained ***'oscillator module'*** or from any other logic device on the same Target Board e.g. another microcontroller or a CPLD / FPGA.

The programmer can switch the device to this external clock signal which will yield much faster programming times. The AT91SAM7 devices **do not** support measurement of the frequency of this oscillator source so the user must manually measure the frequency with eg. a frequency counter. This measured frequency is then specified in the 'programming project' allowing the programmer to calculate the correct ***'FLASH timing parameters'*** for programming the FLASH.

1.2 How to check which oscillator source is available

An AT91SAM7 device can be clocked from any of the different ***'oscillator sources'*** as detailed in section 1.3. It is very important to select the correct ***'oscillator source'*** and ***'oscillator frequency'*** in the EQTools 'Programming Project' as these parameters are used to calculate the correct ***'FLASH timing parameters'*** for programming the FLASH. If you are not sure exactly how the AT91SAM7 device on your Target Board is clocked, you will need to find out in order to program the device correctly.

Please use the techniques below to establish the type and frequency of the oscillator connected to the AT91SAM7 device.

- Check what components are connected to the ***Xin*** and ***Xout*** pins of the AT91SAM7 device.
- If the device is clocked by a ***crystal oscillator***, then you should find a crystal (XTAL) or crystal pack will be connected across the ***Xin*** and ***Xout*** pins of the AT91SAM7 device. The crystal value is usually written on the can of the crystal.
- If the device is clocked by an ***external oscillator signal***, then you should find some sort of oscillator circuit will be connected to the ***Xin*** pin of the AT91SAM7 device. The oscillator frequency should be marked on the circuit schematic, but if it is not then you will need to measure the frequency on the ***Xin*** pin of the AT91SAM7 device.
- If there are no components connected to the ***Xin*** and ***Xout*** pins of the AT91SAM7 device, then the device can only run from its ***internal RC oscillator***.

2.0 Oscillator Selection in EQTools

2.1 Overview

When creating a ***'Programming Project'*** for an Atmel AT91SAM7 device using the Equinox EQTools software, it is necessary to specify the correct ***oscillator source*** and ***oscillator frequency*** in the ***'Target Oscillator Settings'*** section of the project. These settings are used to calculate the correct ***'FLASH timing parameters'*** for the AT91SAM7 device which are then used during the programming process. The programmer requires the fastest possible oscillator in order to program the FLASH as fast as possible.

2.2 Clocking the AT91SAM7 device during programming

Equinox AT91SAM7 programmers work by downloading a highly optimised ***'programming agent'*** into the SRAM on the target AT91SAM7 device via the ARM JTAG debug interface. This ***'programming agent'*** is then executed by the target AT91SAM7 device from SRAM and is used to sequence the programming / verifying of the internal FLASH of the device. The agent will execute at whatever speed the device is currently being clocked at. The faster the oscillator frequency then the faster the agent will execute and hence the faster the FLASH programming time. As an AT91SAM7 device boots from a very slow internal oscillator, it is desirable to switch the device to a faster ***external crystal*** or other ***external oscillator signal*** as this will yield much faster programming times.

2.3 AT91SAM7 – Target Clock Mode

The EQTools software / programmer firmware supports selection of the following AT91SAM7 oscillator sources:

- Internal RC Oscillator
- External crystal oscillator
- External oscillator signal

2.4 AT91SAM7 – Target Oscillator Frequency

The ***'Target Oscillator frequency'*** is the frequency which the target AT91SAM7 ARM Device is being clocked at during the In-System Programming (ISP) process. It is essential to know how the AT91SAM7 device is being clocked so that the correct ***'FLASH timing parameters'*** are selected for programming the FLASH.

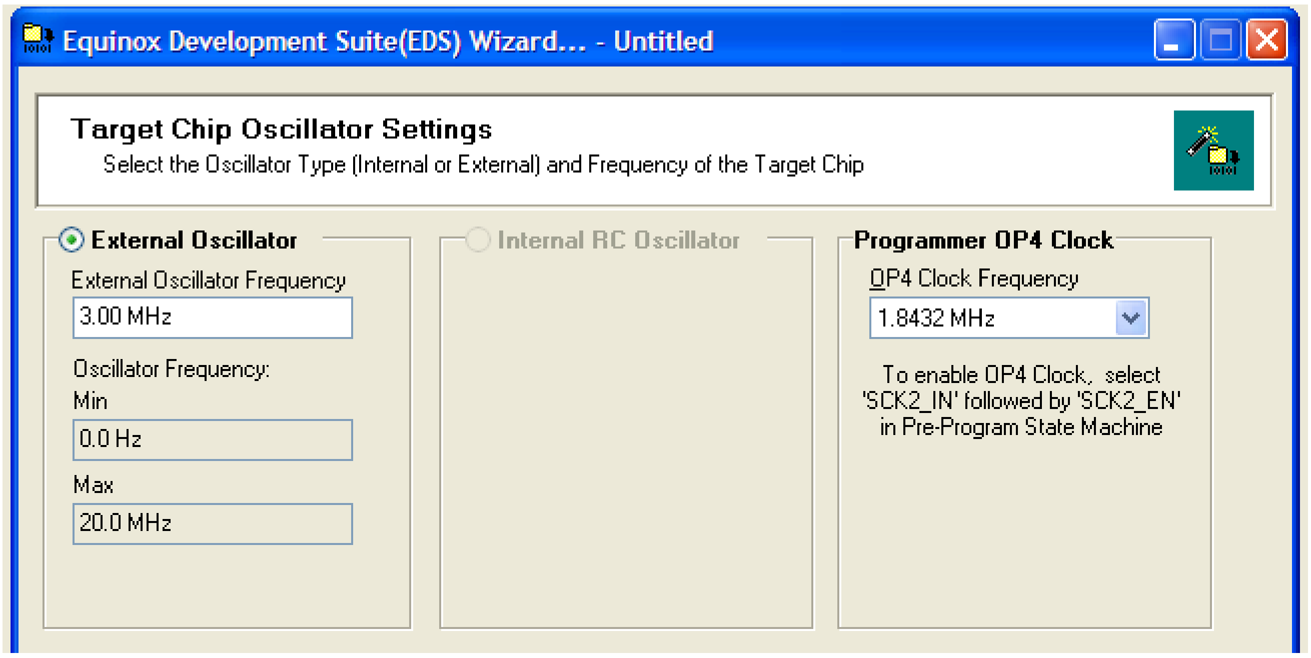
2.5 Target Oscillator Settings (EQTools < build 1151)

If you are using EQTools build 1150 and programmer firmware < 5.18 or below, then the programmer will only support ***'internal RC oscillator'*** or ***'External Crystal oscillator'***. It will try to auto-detect the ***'External Crystal oscillator'*** but if it does not find a crystal then it will automatically use the ***'internal RC oscillator'***.

Supported oscillator sources:

- Internal RC Oscillator
- External crystal oscillator (auto-detect frequency)

The screen below allows you to set up the ***'Target Oscillator'*** settings.


Equinox Development Suite(EDS) Wizard... - Untitled

Target Chip Oscillator Settings
Select the Oscillator Type (Internal or External) and Frequency of the Target Chip

[✓ External Oscillator]
External Oscillator Frequency: 3.00 MHz
Oscillator Frequency:
Min: 0.0 Hz
Max: 20.0 MHz

Internal RC Oscillator

Programmer OP4 Clock
OP4 Clock Frequency: 1.8432 MHz
To enable OP4 Clock, select 'SCK2_IN' followed by 'SCK2_EN' in Pre-Program State Machine

Instructions:

- Set the ***'External Oscillator Frequency'*** to the external crystal oscillator frequency (if there is one fitted on your Target Board).
- Many AT91SAM7 microcontrollers actually run from an ***Internal Oscillator*** to start with. The programmer then automatically swaps the oscillator settings of the Target Device to use the fastest available oscillator for programming. This is usually an external crystal oscillator.
- You do not need to change any other parameters on this screen.

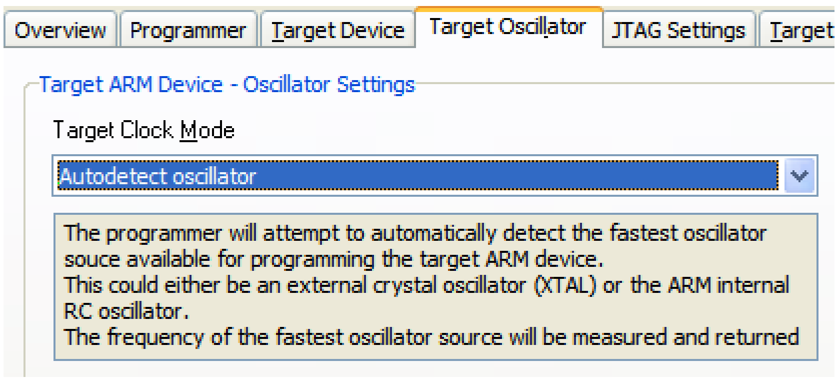
2.6 Target Oscillator Settings (EQTools > build 2000)

If you are using EQTools build 2000 or above and programmer firmware >= 5.20, then the programmer supports all possible AT91SAM7 oscillator sources. It also supports the option to automatically detect the fastest oscillator available.

Supported oscillator sources:

- Internal RC Oscillator
- External crystal oscillator
- External oscillator signal

The screen below allows you to set up the ***'Target Clock Mode'*** and the ***'Target Oscillator frequency'***.



Target ARM Device - Oscillator Settings

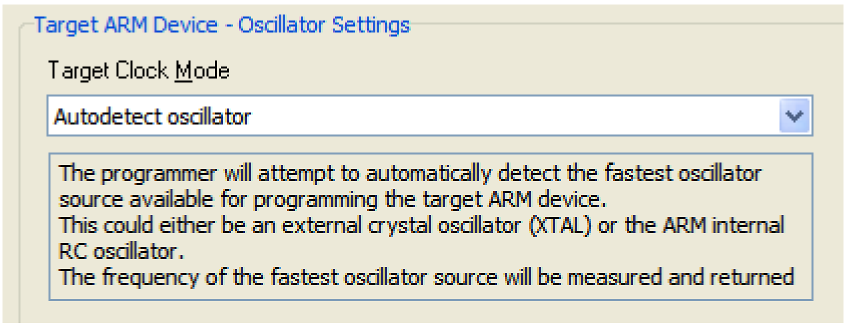
Target Clock Mode: Autodetect oscillator

The programmer will attempt to automatically detect the fastest oscillator souce available for programming the target ARM device.
This could either be an external crystal oscillator (XTAL) or the ARM internal RC oscillator.
The frequency of the fastest oscillator source will be measured and returned

Select the ***'Target Clock Mode'*** from the options below:

i. Auto detect oscillator (default)

- The programmer will attempt to automatically detect the fastest oscillator source available for programming the target ARM device.
- This could either be an ***external crystal oscillator (XTAL)*** or the ARM ***internal RC oscillator***.
- The frequency of the fastest oscillator source will be measured and returned in the ***'Programmer diagnostics'***.
- This mode is equivalent to the older EQTools build <1150 and firmware <5.20 standard operation.



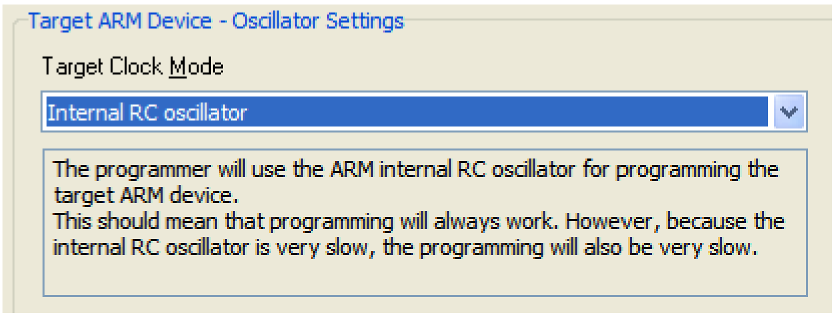
Target ARM Device - Oscillator Settings

Target Clock Mode: Autodetect oscillator

The programmer will attempt to automatically detect the fastest oscillator source available for programming the target ARM device.
This could either be an external crystal oscillator (XTAL) or the ARM internal RC oscillator.
The frequency of the fastest oscillator source will be measured and returned

ii. Internal RC oscillator

The programmer will use the ARM ***internal RC oscillator*** for programming the target ARM device. This should mean that programming will always work. However, because the internal RC oscillator is very slow, the programming will also be very slow.

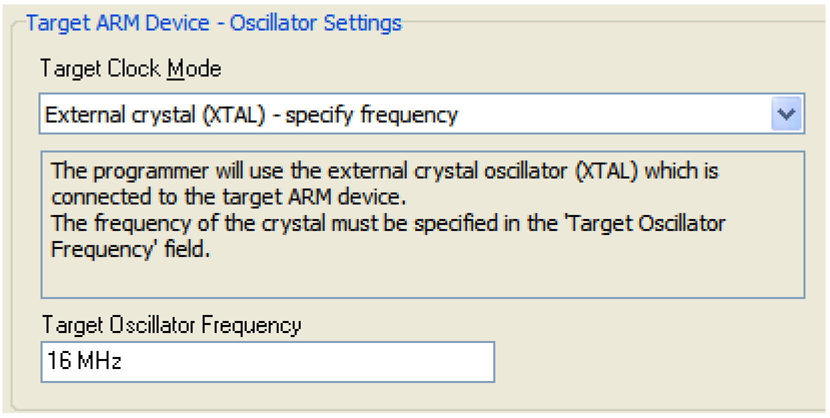

Target ARM Device - Oscillator Settings

Target Clock Mode: Internal RC oscillator

The programmer will use the ARM internal RC oscillator for programming the target ARM device.
This should mean that programming will always work. However, because the internal RC oscillator is very slow, the programming will also be very slow.

iii. External crystal (XTAL)

The programmer will use the external crystal oscillator (XTAL) which is connected to the target ARM device.


Target ARM Device - Oscillator Settings

Target Clock Mode: External crystal (XTAL) - specify frequency

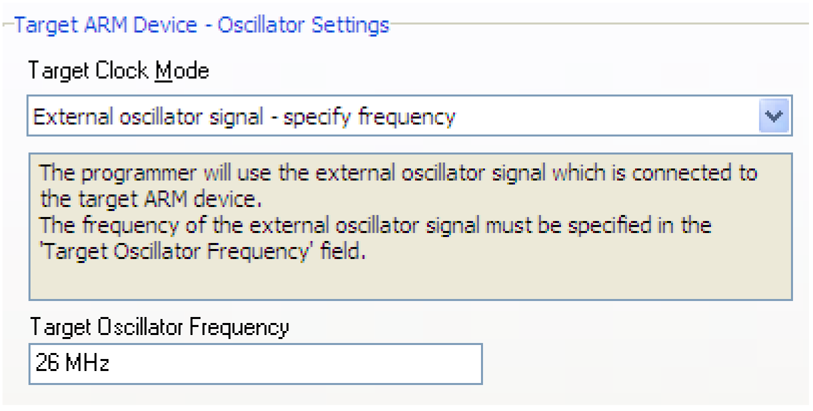
The programmer will use the external crystal oscillator (XTAL) which is connected to the target ARM device.
The frequency of the crystal must be specified in the 'Target Oscillator Frequency' field.

Target Oscillator Frequency: 16 MHz

The frequency of the crystal must be specified in the ***'Target Oscillator Frequency'*** field.
The programmer does **NOT** perform a measurement of the oscillator frequency.

iv. external oscillator signal

The programmer will use the ***external oscillator signal*** which is connected to the target ARM device.


Target ARM Device - Oscillator Settings

Target Clock Mode: External oscillator signal - specify frequency

The programmer will use the external oscillator signal which is connected to the target ARM device.
The frequency of the external oscillator signal must be specified in the 'Target Oscillator Frequency' field.

Target Oscillator Frequency: 26 MHz

The frequency of the ***external oscillator signal*** must be specified in the ***'Target Oscillator Frequency'*** field.

v. specialx - custom algorithm

This is a custom algorithm.
Please do not use unless instructed to by Equinox.

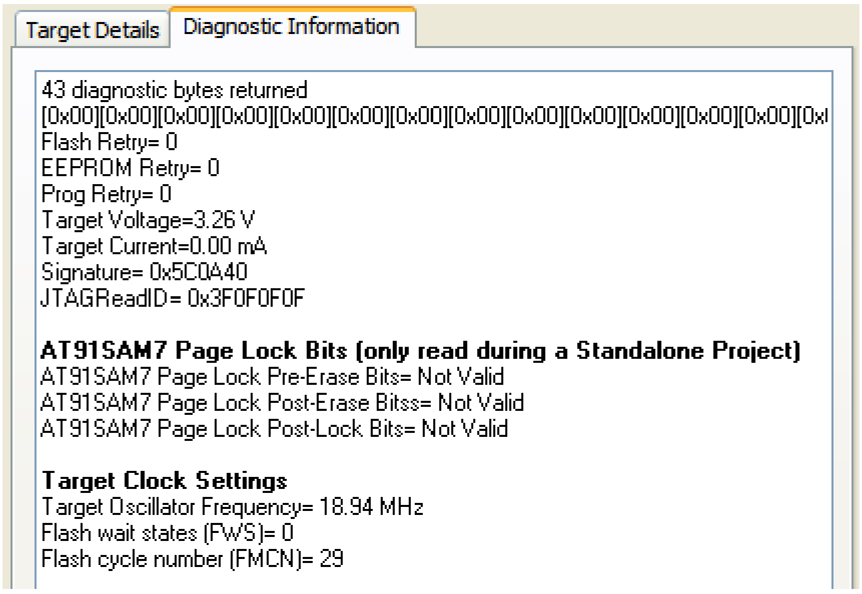
2.7 Testing the Oscillator Settings

Once you have selected the correct ***'oscillator source'*** and '***oscillator frequency'***, it is then possible to perform a simple check to make sure that the target AT91SAM7 device.

Instructions:

- Go to the ***<Target Device>*** screen
- Click the ***<Check ID>*** button → the programmer will try to connect to the target AT91SAM7 device.
- If EDS connects OK, it should display the following message ***'Signature and JTAG ID Check: PASS'***.
- Click the ***<Diagnostics>*** button on this message
- Select the ***<Diagnostic Information***> tab

→ The diagnostic information is displayed as shown below…..



```
43 diagnostic bytes returned
[0x00][0x00][0x00][0x00][0x00][0x00][0x00][0x00][0x00][0x00][0x00][0x00][0x00][0x
Flash Retry= 0
EEPROM Retry= 0
Prog Retry= 0
Target Voltage=3.26 V
Target Current=0.00 mA
Signature= 0x5C0A40
JTAGReadID= 0x3F0F0F0F

AT91SAM7 Page Lock Bits (only read during a Standalone Project)
AT91SAM7 Page Lock Pre-Erase Bits= Not Valid
AT91SAM7 Page Lock Post-Erase Bitss= Not Valid
AT91SAM7 Page Lock Post-Lock Bits= Not Valid

Target Clock Settings
Target Oscillator Frequency= 18.94 MHz
Flash wait states (FWS)= 0
Flash cycle number (FMCN)= 29
```

- The ***'Target Clock Settings'*** read from the target AT91SAM7 device are displayed.
- These should match the oscillator settings in the project.
- If the project is set to measure the crystal frequency, then the measured frequency will be displayed on this screen.