Application Note

Report No:

AN145

Title:

In-System Programming (ISP) of Sigma Z-Wave 500 series devices and modules

Author:	Date:	Version Number:
John Marriott	18th August 2014	0.66

Contents

1.0 Introduction

Equinox Technologies manufacture a comprehensive range of programmers suitable for high-speed In-System Programming (ISP) of ***Sigma 500 series Z-WAVE devices***. This application note describes how to develop and implement ***In-System Programming (ISP)*** support for the Z-WAVE devices using the ***'SPI Programming Interface'***. The document details how to make a ***'Programming Project'*** which will operate on any Equinox ISP programmer including a full description of how to implement ***In-System Programming (ISP)*** of Z-WAVE devices.

1.1 Features

The Equinox programming range includes solutions for development, low / mid / high volume production and field programming of ***Sigma 500 series Z-WAVE SOC (System on Chip) devices***. and ***Z-WAVE modules***.

General features…..

- High-speed In-System Programming (ISP) support of Sigma 500 series Z-WAVE SOC devices and modules
- Programming solutions for development, low / mid / high volume production and field programming of Z-WAVE devices
- Programs the on-chip FLASH Memory and NVR area of Z-WAVE devices
- Uses a high-speed ***'SPI bus'*** port as the ISP interface
- Very high-speed programming due to fast SPI programming interface, local user 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 single operation.
- Fully user-configurable pre-programming statemachine supports custom target reset circuits

In 'Development Mode'…..

- Powerful yet simple-to-use Development Suite called ***'EDS'***
- All aspects of programming the ***Z-WAVE*** device can be controlled from ***EDS***
- Program and read back the Sigma device on-chip FLASH memory under PC control
- All projects can be developed and tested on a real device before uploading a ***'Standalone Programming Project***' to the programmer
- Tested ***'Programming Projects'*** can then be uploaded to the programmer for use in ***'Standalone Mode'***

Production Programming solutions…..

- Programmers can be used in ***'Standalone Mode'*** (no PC required)
- A single ***'Standalone Programming Project'*** can Erase the device and program /verify the FLASH area in a single operation.
- Up to 64 x Z-WAVE ***'Standalone Programming Projects'*** can be stored inside the ISPnano programmer.

- Programmer can store multiple versions of firmware for different ***'customer product versions'***.
- Support for programming unique data per device including serial numbers, MAC addresses, calibration data, barcode data etc.
- ***ConsoleEDS*** **–** powerful ***'console application'*** allows the programmer to be controlled from any custom remote application.
- ***ISP-PRO*** **–** powerful production control / sequencing utility supports controlling of up to 32 programmers from the same PC.
- ***ISPnano-MUX*** programmer family - supports sequential programming of up to 8 x independent Target Boards (UUTs) on a ***'PCB Panel'***
- ***ISPnano-GANG*** programmer family - supports concurrent gang programming of up to 32 x independent Target Boards (UUTs) on a ***'PCB Panel'***

1.2 Programmers supporting Z-WAVE 500 series devices

The ***'Z-WAVE 500 series'*** devices are currently only supported by the Equinox ***'ISPnano'*** family of production ISP Programmers. The ***'ISPnano'*** programmers can be upgraded to support high-speed programming of via the ***'SPI Programming Interface'*** .

The table below lists all the Equinox ISP programmers which are capable of programming ***'Z-WAVE 500 series'*** devices....

Programmer	Sigma Z-wave Support	Requirements	Upgrade Order Code
• **ISPnano Series III** • **ISPnano Series III ATE** • **ISPnano Series IV ATE** • **ISPnano-MUX**	Upgrade	License upgrade	ISPnano-UPG35

Please note:

- A chargeable ***'License Upgrade'*** is required to enable the ***'Z-WAVE 500 series'*** device support on any of these programmers.
- The ***programmer firmware*** will probably also need to be upgraded in order to support ***'Z-WAVE 500 series'*** device programming – see section 1.3.
- It is also recommended that EQTools version 4 build 3498 or above is used when programming ***'Z-WAVE 500 series'*** devices.

1.3 Calibration overview

1.3.1 Overview

The Z-Wave 500 series SOC devices and modules must be calibrated at the customer production programming stage before they will operate correctly.

The calibration procedure(s) required depend on whether you are programming a Z-Wave module, SOC (just the bare IC) or 'Bare die' version of the Z-Wave product.

The table below details which calibration procedure(s) is / are required for the different Z-Wave product types......

Sigma product family	Z-Wave Product type	Customer Crystal (XTAL) calibration required	Customer TX calibration required	Equinox IOMOD10 Calibration module required
ZM5101	**SiP Module**	**NO** *(Sigma factory calibrated)*	**YES**	**NO**
ZM5xxx **ZDB5xxx** *(Except ZM5101)*	**Module**	**NO** *(Sigma factory calibrated)*	**NO** *(Sigma factory calibrated)*	**NO**
SD35xx	**SOC** *(System On chip - Bare IC)*	**YES**	**YES**	**YES**
ZW05xx	**Bare die**	**YES**	**YES**	**YES**
ALL	**Any product type where the 'NVR Area' has been erased or corrupted.**	**YES**	**YES**	**YES**

1.3.2 Crystal (XTAL) calibration

The ***'Crystal (XTAL) calibration'*** procedure tunes the TX- and RX radio frequency of the Z-Wave device so as to give the minimum frequency error. This calibration must be carried out on the final crystal which will be used with the Z-Wave device.

Important notes:

- The ***'Crystal (XTAL) calibration'*** procedure is performed by Sigma at the factory for all Z-Wave modules as these modules have the final crystal already fitted to them.
- If you are programming SOCs (bare ICs) or 'Bare Die' products, then these devices are **NOT** pre-calibrated by Sigma at the factory. You will need to perform the ***'Crystal (XTAL) calibration'*** procedure on these devices at the customer production programming stage.

1.3.3 TX calibration

The ***'TX calibration'*** procedure is required to tune the Z-Wave frequency separation during modulation to an optimum value.

Important notes:

- The ***'TX calibration'*** procedure is currently NOT performed by Sigma at the factory.
- The customer must therefore perform the ***'TX calibration'*** procedure on ALL Z-Wave modules, SOCs and 'Bare die' devices at the ***'customer production programming'*** stage.#

1.3.4 Calibration recovery after accidental NVR erasure / corruption

If the ***'NVR Area'*** of a 500 series Z-Wave device is accidentally erased or corrupted, then both the ***'Crystal (XTAL)'*** and ***'TX'*** calibration parameters may be invalid. This means that the Z-Wave device will no longer function properly.

If this happens, then it is necessary to fully re-calibrate the Z-Wave device. This recalibration process involves performing both the ***'Crystal (XTAL) calibration'*** and the ***'TX calibration'*** procedures and also programming some default 'factory settings' for the device or module back into the 'NVR Area' of the device.

This procedure requires the following equipment and other information:

- An ***ISPnano Series 4*** or ***ISPnano-MUX*** programmer
- An Equinox ***'IOMOD10 - Sigma Calibration Module'*** - plugged into the programmer
- A special ***'Restore NVR Calibration'*** script
- A custom parameter file to restore the relevant ***'factory parameters'*** to the device

1.3.5 Programmer selection guide for Z-Wave calibration

The table below details which Equinox programmers are capable of performing the ***'Crystal (XTAL) calibration'*** and ***'TX calibration'*** procedures.

Programmer name	Crystal (XTAL) calibration supported	TX calibration supported	IOMOD10 Calibration module required
ISP nano Series III	**NO**	**YES**	**Not applicable**
ISP nano Series III - ATE	**NO**	**YES**	**Not applicable**
ISP nano Series IV - ATE	**YES**	**YES**	**YES** 1 x **IOMOD10** module for '**XTAL Calibration'**
ISP nano gang Multi-Channel Gang Production ISP Programming Systems	**NO**	**YES**	**Not applicable**
ISP nano mux **2, 4 or 8 Channels**	**YES**	**YES**	**YES** Requires 2, 4 or 8 x **IOMOD10** modules for '**XTAL Calibration'**

1.4 Device Support

1.4.1 Overview

The Equinox ISPnano programmer range supports the following ***'Z-WAVE 500 series'*** - ***SOC (System on Chip)*** devices and ***Z-WAVE modules***...

Z-Wave Next Gen SoCs (500 series):
SD3502, SD3503

Z-Wave Next Gen modules (500 series):
ZM5101, ZM5202, ZM5304

1.5 Upgrading your Equinox Programmer to support Sigma 500 series Z-WAVE device programming

1.5.1 Overview

The Sigma ***'Z-WAVE 500 series'*** algorithms are not supported as standard on any Equinox programmers. It is necessary to purchase a ***'License Upgrade'*** for ***'Z-WAVE 500 series'*** support from Equinox. Equinox will then send you a ***'Upgrade License String'*** which will upgrade your programmer to support programming of this device family.

1.5.2 Purchasing a Sigma Z-WAVE 500 series License

All Equinox ISP programmers require the purchase of a ***'License Upgrade'*** to enable ***'Z-WAVE 500 series'*** programming support. Please see the table in section 1.2 for the relevant upgrade for your programmer.

1.5.3 How do I enable the programmer for Z-WAVE programming?

To enable your programmer to support ***'Z-WAVE 500 series'*** ISP programming, please purchase the relevant 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 ***'License String'*** which is keyed to your programmer Serial Number.
   - This string can be entered directly into the ***<Enter License>*** screen in EQTools.

1.5.6 Entering the License String to upgrade your programmer

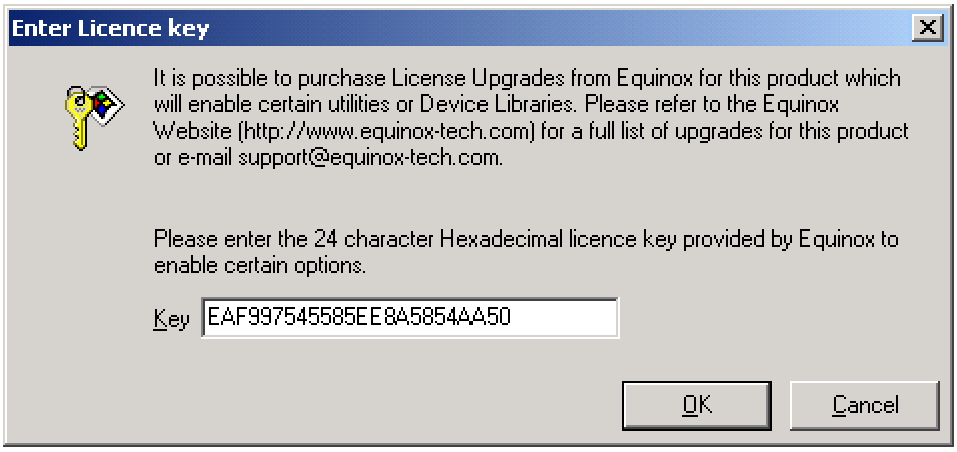
Once you have received the License String from Equinox, please follow the steps below to apply the upgrade to your programmer:

- 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.

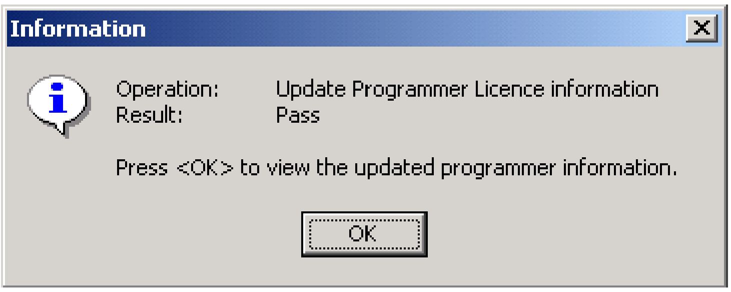
Please enter the 24 character Hexadecimal licence key provided by Equinox to enable certain options.

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 ***'Sigma 500 Series devices' i***s now ENABLED.

1.6 Programmer firmware versions for Sigma 500 series support

Most Equinox ISP Programmers can be upgraded to support high-speed programming of ***'Z-WAVE 500 series'*** microcontrollers via the ***'SPI Programming Interface'*** .The table below lists all the Equinox ISP programmers which are capable of programming***'Z-WAVE 500 series'***. A chargeable ***'License Upgrade'*** is required to enable the ***'Z-WAVE 500 series'*** support on any of these programmers.

Fig. 1.3 Programmer firmware versions for **'Z-WAVE 500 series'** *In-System Programming (ISP) Support*

Programmer	***'Z-WAVE 500 series' support***
ISPnano Series III	Please contact Equinox
ISPnano Series IV	Please contact Equinox
ISPnano-MUX 2 / 4 / 8	Please contact Equinox

Please note:

- Due to limited firmware storage space and the lack of required hardware on the EPSILON5-MK4 and FS2009 / FS2009USB and PPM4-MK1 programmers, these programmers cannot support the ***'Z-WAVE 500 series'*** devices.

2.0 SPI Programming Interface

2.1 Overview

The ***'Z-WAVE 500 series'*** devices are programmed using three different physical ***'programming interfaces'*** as detailed in the table below.

Interface		
USB	USB Interface	Uses the USB port of the Z-Wave device to program the on-chip FLASH memory. This programming mode still requires an external device programmer to set the device into 'programming mode' via the SPI or UART interface before the UART programming interface can be used.
SPI	SPI Programming Port	Uses an SPI Port + RESET pin as an In-System Programming (ISP) interface..
UART	UART interface	Uses a 2-pin UART interface as an In-System Programming (ISP) interface..

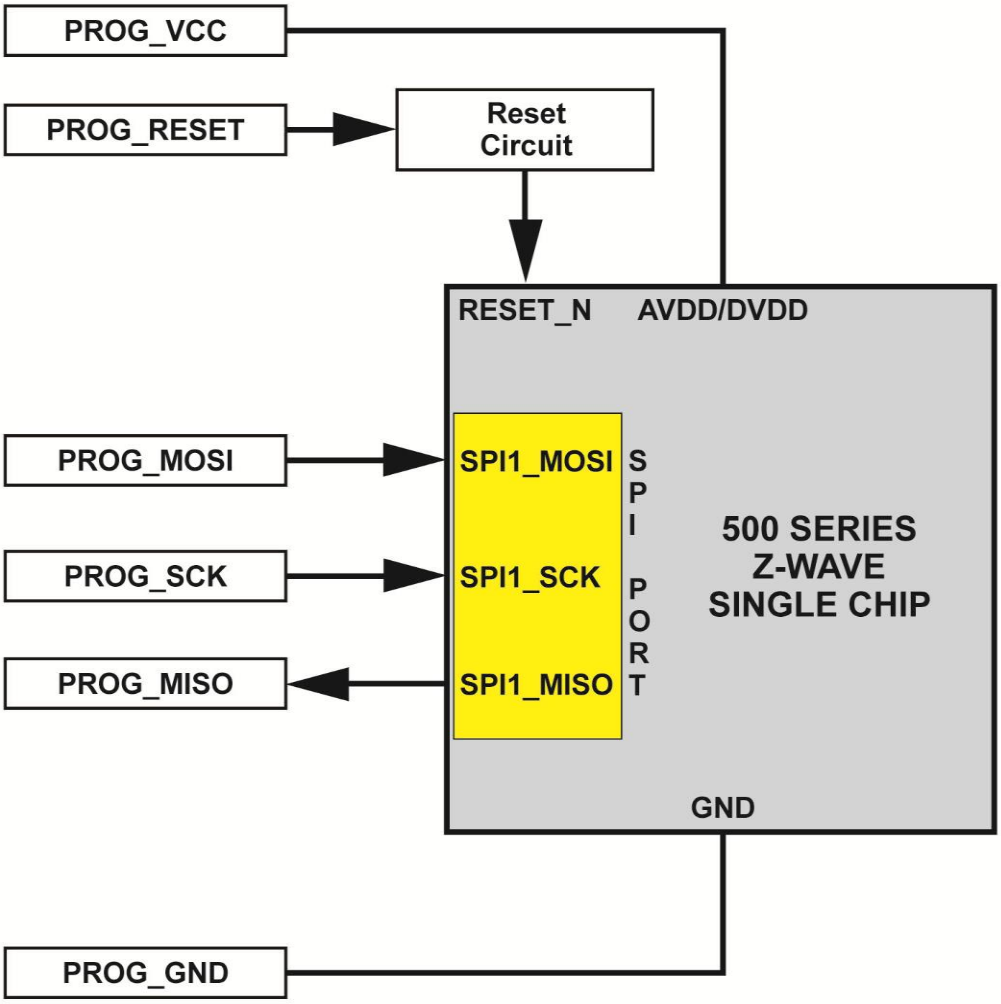
2.2 SPI - Programming Interface - Features

- Fast Programming speeds
- Simple 3-wire SPI bus connection + ***RESET_N*** signal

2.3 Z-WAVE single-chip In-System Programming (ISP) Schematic

The diagram below details the connections required to implement In-System Programming of a single ***'Z-WAVE 500 series'*** device using an Equinox ISP programmer.

Fig 2.3 – ***'Z-WAVE 500 series'*** device – SPI Programming Interface connection



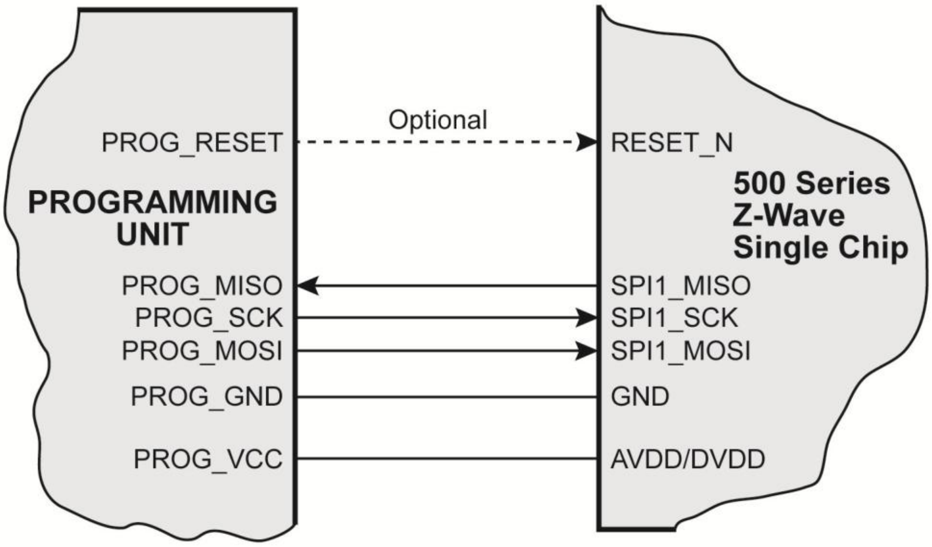


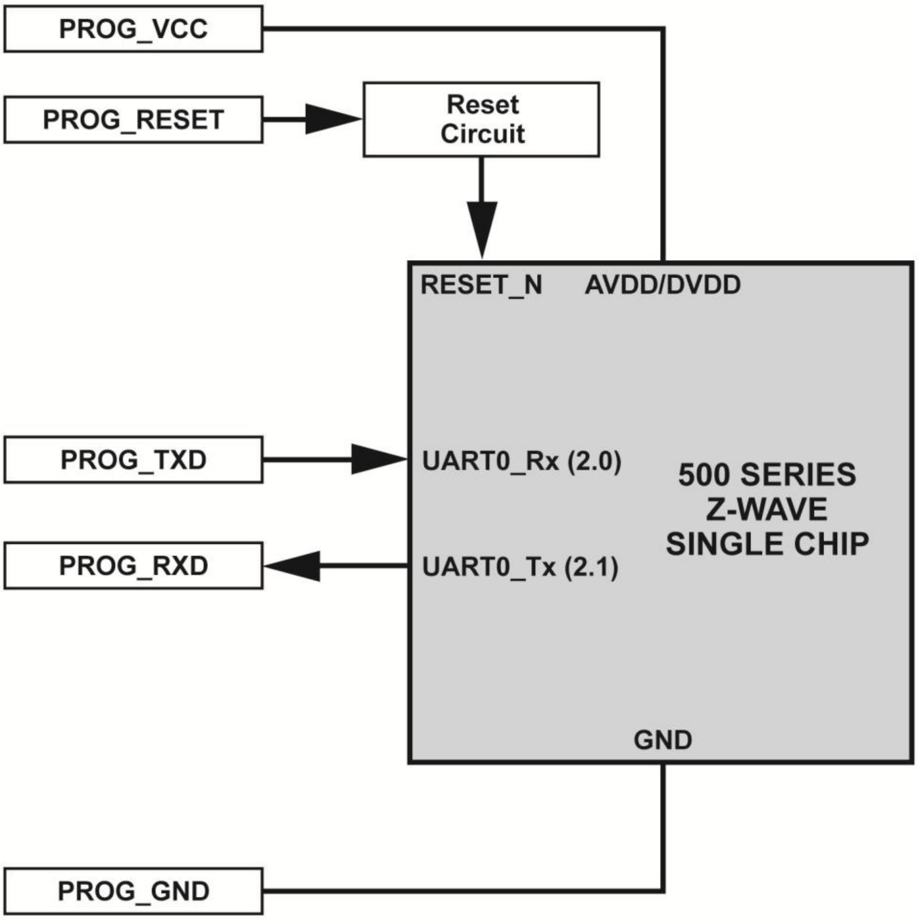
Fig 2.3.b – Sigma Z-wave 500 series device - SPI Signal names and directions

Programmer Signal Name	**Signal description**	**Signal direction (from Programmer)**	**Connect to Z-Wave Pin**	**Signal direction (from Microcontroller)**
PROG_MOSI	Master OUT, Slave In	Output	MOSI	Input
PROG_MISO	Master IN, Slave OUT	Input	MISO	Output
PROG_SCK	Serial Clock	Output	SCK	Input
PROG_RESET	RESET	Output	RESET_N	Input

2.4 Z-WAVE UART ISP Schematic

The diagram below details the connections required to implement In-System Programming of a single ***'Z-WAVE 500 series'*** device using an Equinox ISP programmer via the ***'UART interface'***.

Fig 2.3 – ***'Z-WAVE 500 series'*** device - UART Programming Interface connections



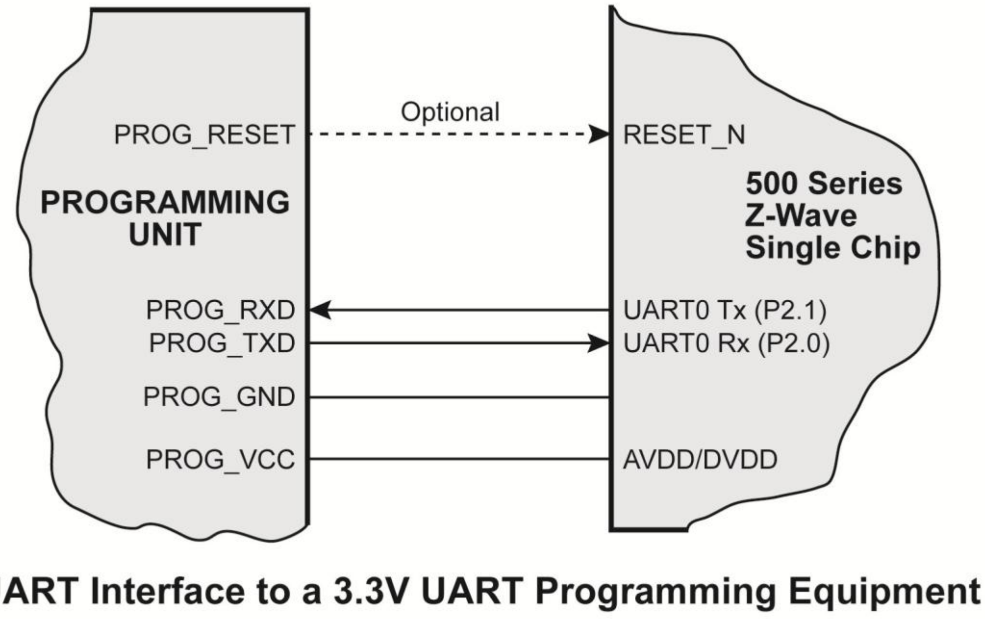


Fig 2.4.b – Sigma Z-wave 500 series device - UART Signal names and directions

Programmer Signal Name	**Signal description**	**Signal direction (from Programmer)**	**Connect to Z-Wave Pin**	**Signal direction (from Z-Wave device)**
PROG_TXD	UART TRANSMIT	Output	UART0 Rx (P2.0)	Input
PROG_RXD	UART RECEIVE	Input	UART0 Tx (P2.1)	Output
PROG_RESET	RESET	Output	RESET_N	Input

3.0 Creating an EDS (Development) Project

3.1 Overview

This section describes how to make a ***'Programming Project'*** for a ***'Z-WAVE 500 series'*** device.

Please note:
The following versions of EQTools and firmware are required to support a ***'Z-WAVE 500 series'*** device programming:

- EQTools version 4 build 3490 or higher
- Firmware 6.11 - please consult Equinox

3.2 Information required to create a Z-WAVE Project

The following information is required about the Target Board in order to create a ***'Z-WAVE 500 series'*** device Programming Project:

#	Information / data required	Example
1	Sigma Z-WAVE device part number	SD3502
2	Connector on Target board	10-way IDC connector (SPI version)
3	Programming interface (SPI or UART)	SPI
4	Target System Vcc voltage	e.g. 3.3V
5	Target System maximum current consumption	e.g. 100mA
6	FLASH area 'Program File'	Binary (*.bin) or Intel Hex (*.hex)
7	Reset circuit parameters	e.g. • Capacitor / Resistor circuit • Watchdog supervisor circuit • Voltage monitoring circuit
8	TX Power parameters	The 'TX Power parameters' should be obtained by from RF testing of your final product.

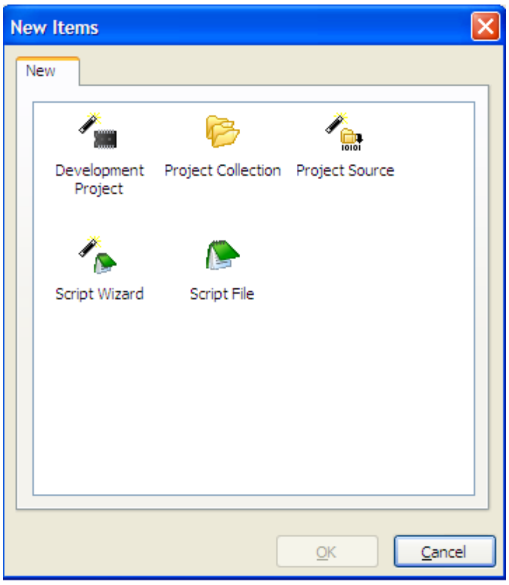
3.3 Creating an EDS (Development project)

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:

- Click the ***'New'*** icon on the task bar

→ The ***'New items'*** screen will be displayed…



- Select **<Development Project**> and click **<*OK*>**

→ The EDS (Development) Wizard will launch

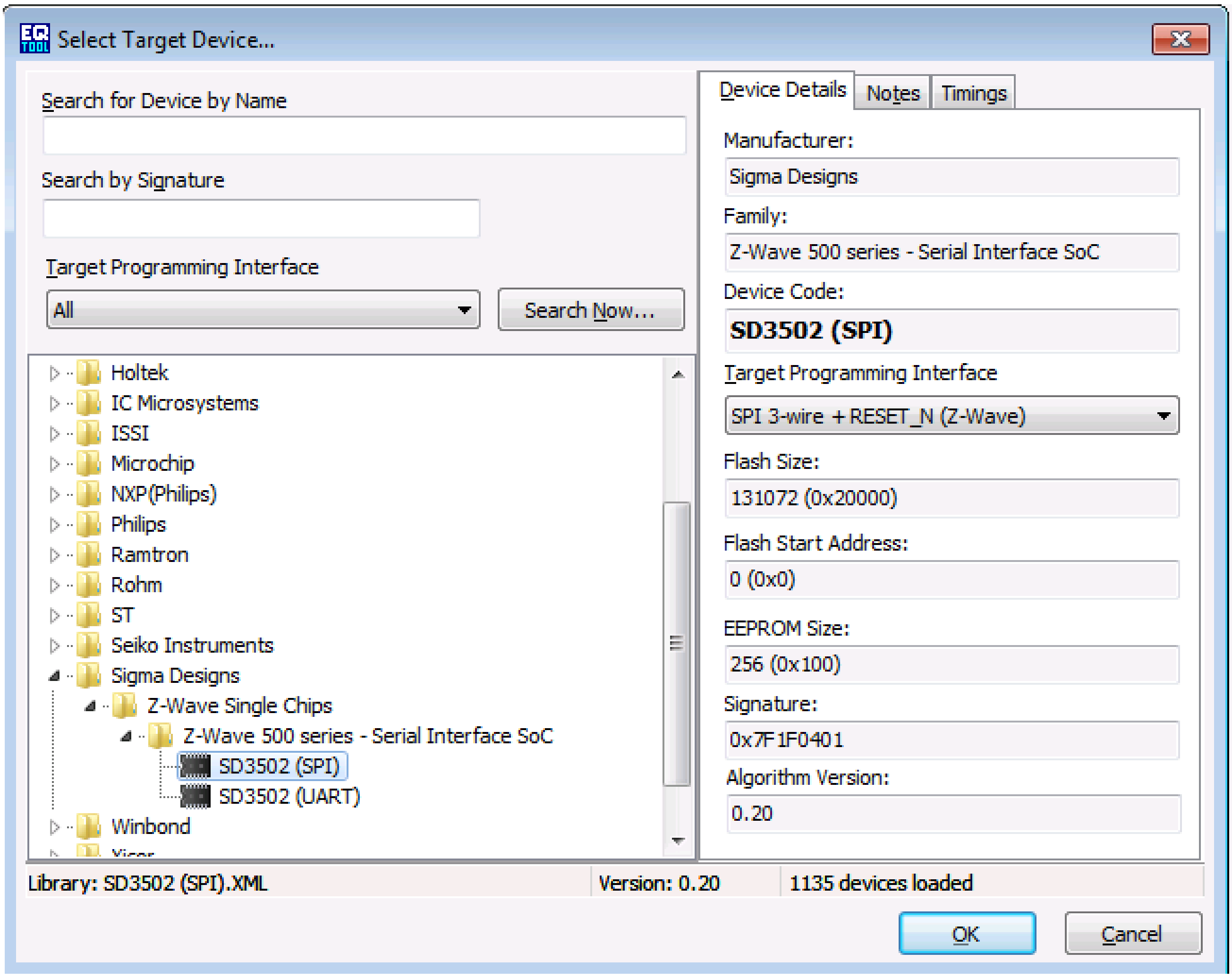
- Select the relevant 'Programmer' and then click **<*Next*>**

3.4 Selecting the correct Target Device

It is important to select the correct ***'Target Device'*** when programming a ***'Z-WAVE 500 series'*** device. The part number of the device should be printed on the top of the chip e.g. ***'SD3502'***.

3.4.1 Device selection

- Click **<Next>** → the **<Select Target Device>** screen will be displayed.
- Type in the ***'Device Part Number'*** e.g. '***SD3502'*** into the ***'Search for Device'*** field → a list of all matching devices will be displayed in the box underneath.



Target Programming Interface: All

Manufacturer: Sigma Designs

Family: Z-Wave 500 series - Serial Interface SoC

Device Code: **SD3502 (SPI)**

Target Programming Interface: SPI 3-wire + RESET_N (Z-Wave)

Flash Size: 131072 (0x20000)

Flash Start Address: 0 (0x0)

EEPROM Size: 256 (0x100)

Signature: 0x7F1F0401

Algorithm Version: 0.20

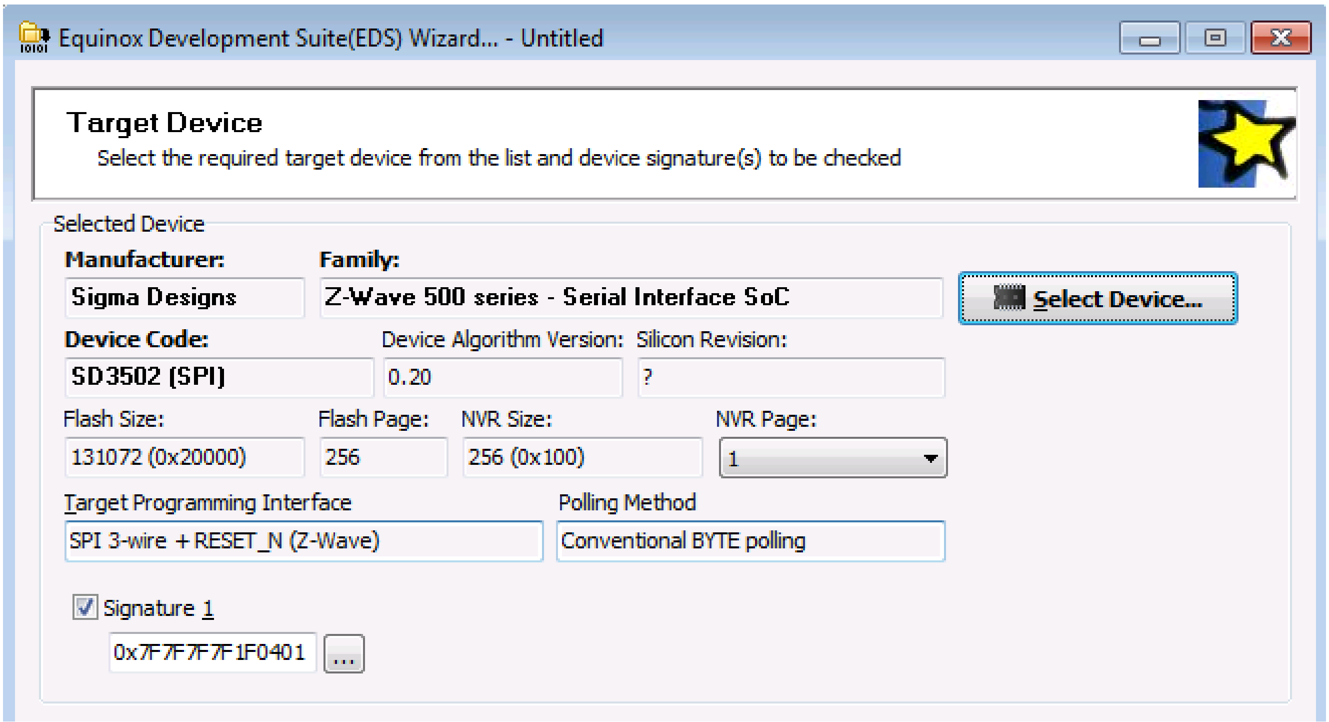
Library: SD3502 (SPI).XML | Version: 0.20 | 1135 devices loaded

- As the ***Z-wave 500 series*** devices can be programmed via different 'programming interfaces', the device list shows the available interfaces eg. SPI or UART interfaces for the SD3502 device.
- Select the required device / programming interface from the list e.g. ***'SD3502 (SPI)'*** and then click **<OK>**

→ The ***SD3502*** device is now selected and will be programmed via the ***'SPI'*** interface..

3.4.2 Device Chip ID / Signature

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



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

Target Device

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

Selected Device

Manufacturer: Sigma Designs

Family: Z-Wave 500 series - Serial Interface SoC

Device Code: SD3502 (SPI)

Device Algorithm Version: 0.20

Silicon Revision: ?

Flash Size: 131072 (0x20000)

Flash Page: 256

NVR Size: 256 (0x100)

NVR Page: 1

Target Programming Interface: SPI 3-wire + RESET_N (Z-Wave)

Polling Method: Conventional BYTE polling

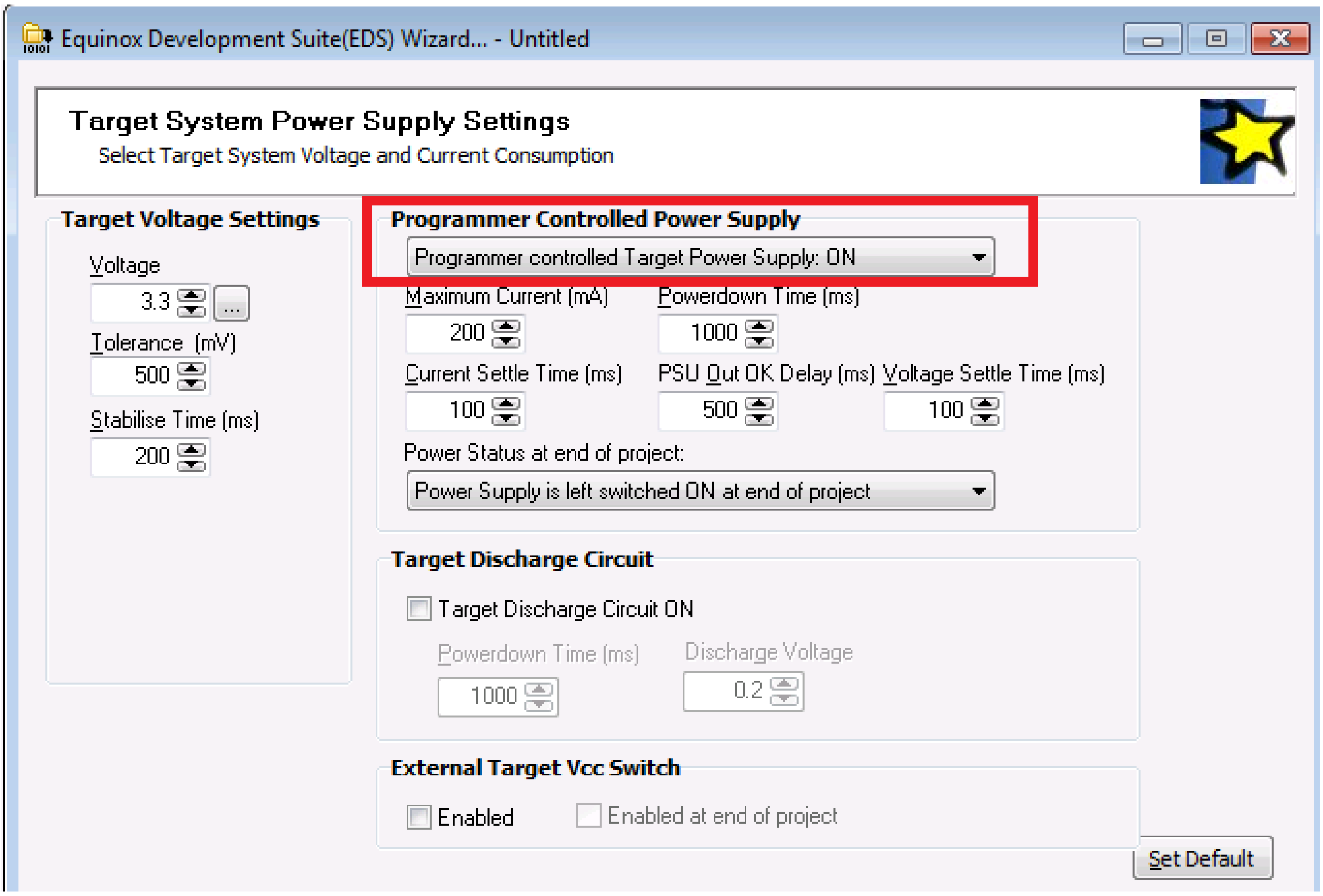
☑ Signature 1

0x7F7F7F7F1F0401

- The project is set to automatically read and validate the ***'Device Signature'*** of the Target Device by default.
- The actual ***'Signature / Chip ID'*** for the device being programmed can be found in the User Manual for the device. Alternatively, it can be read from the target device using EDS.

3.5 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

Powerdown Time (ms): 1000

Discharge Voltage: 0.2

External Target Vcc Switch

□ Enabled

□ Enabled at end of project

i. Select the Target Voltage

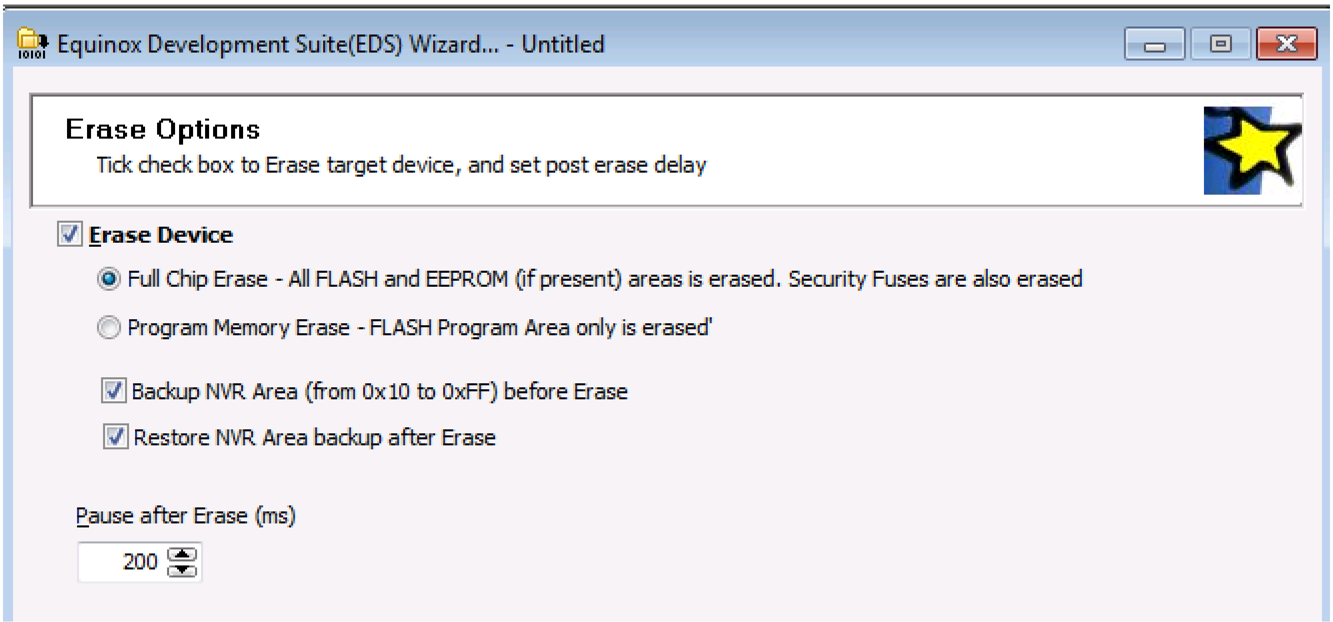
- This should be the voltage at which the Target Device itself is being powered at during the programming operation. This is usually 3.0 – 3.6V.
- Set the ***'Voltage Tolerance'*** to be as wide as possible e.g. 500mV to allow for power supply variations. If the programmer is powering the Target System, 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.

3.6 Erase options

This screen allows you to set up the ***'Erase options'*** for the target device....



Erase Options

Tick check box to Erase target device, and set post erase delay

☑ Erase Device

● Full Chip Erase - All FLASH and EEPROM (if present) areas is erased. Security Fuses are also erased

○ Program Memory Erase - FLASH Program Area only is erased'

☑ Backup NVR Area (from 0x10 to 0xFF) before Erase

☑ Restore NVR Area backup after Erase

Pause after Erase (ms)

200

Backup NVR Area before Erase

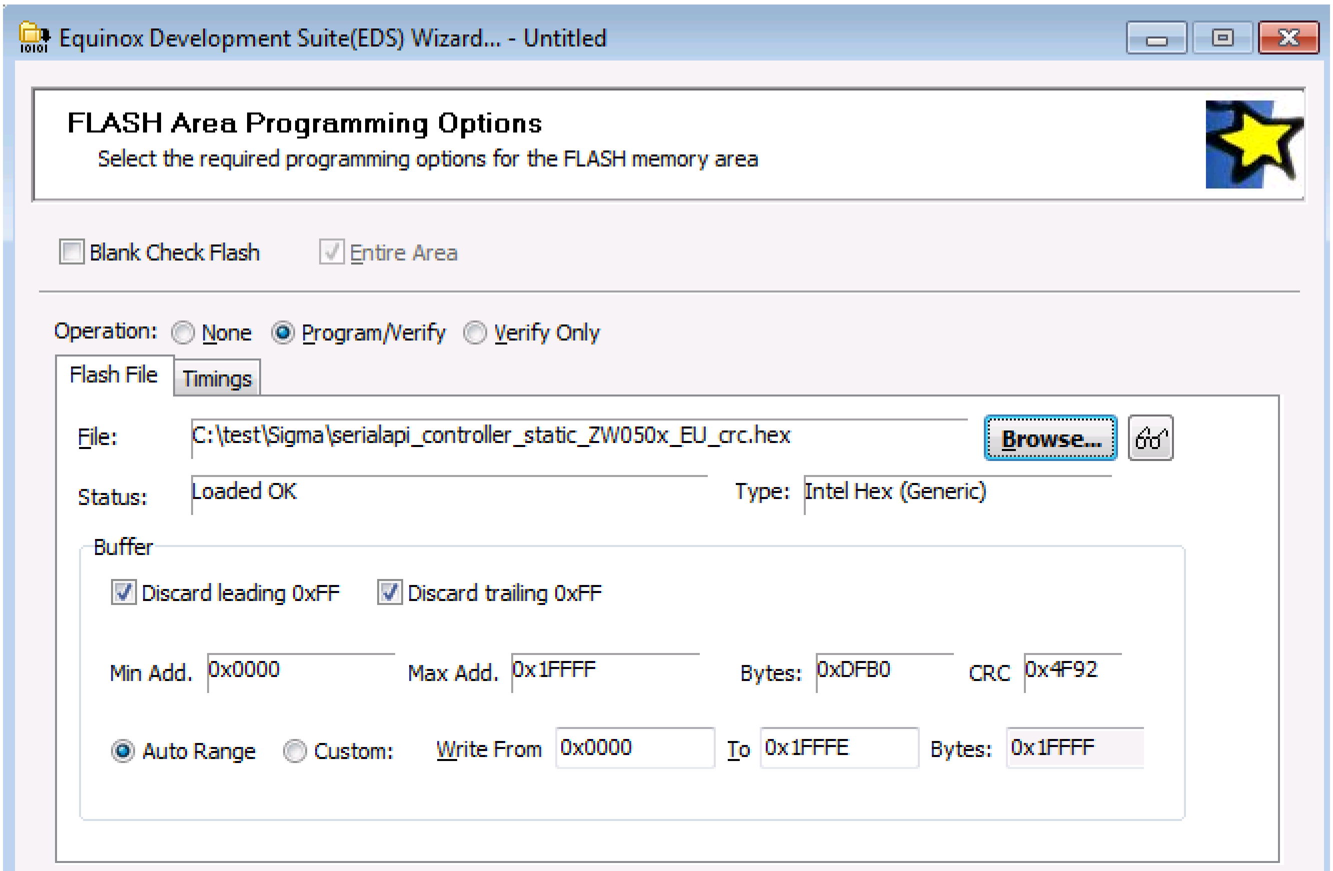
If this option is selected, the programmer will automatically read back the ***'NVR data area'*** from the target device before a ***'Chip Erase'*** operation is performed.

Restore NVR Area after Erase

If this option is selected, the programmer will automatically restore the ***'NVR data'*** contents which were read back from the target device before the ***'Chip Erase'*** operation was performed.

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.



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

FLASH Area Programming Options

Select the required programming options for the FLASH memory area

□ Blank Check Flash ☑ Entire Area

Operation: ○ None ● Program/Verify ○ Verify Only

File: C:\test\Sigma\serialapi_controller_static_ZW050x_EU_crc.hex

Status: Loaded OK Type: Intel Hex (Generic)

Buffer

☑ Discard leading 0xFF ☑ Discard trailing 0xFF

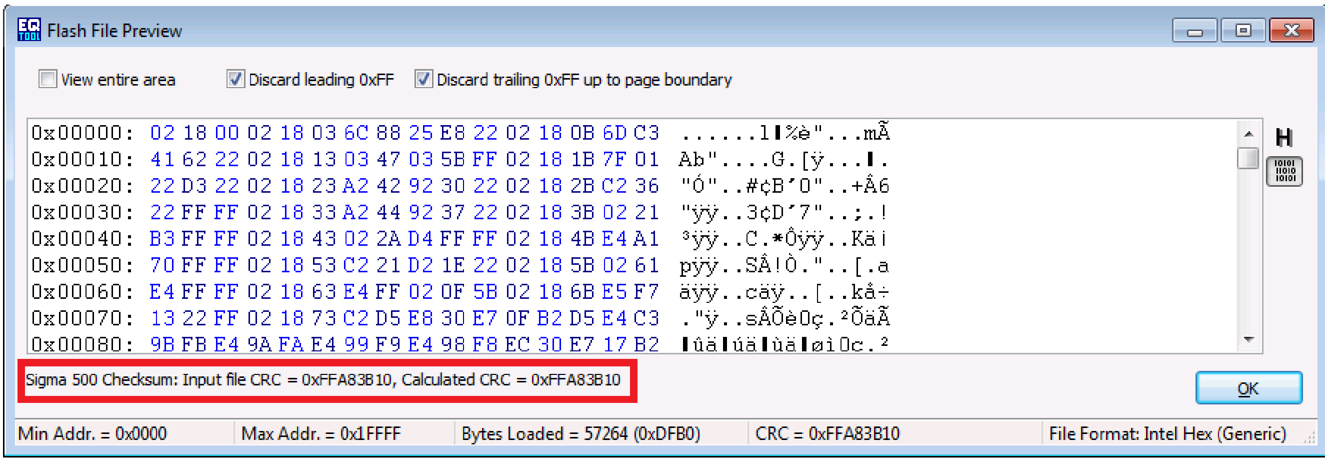
Min Add. 0x0000 Max Add. 0x1FFFF Bytes: 0xDFB0 CRC 0x4F92

● Auto Range ○ Custom: Write From 0x0000 To 0x1FFFE Bytes: 0x1FFFF

This is an optional step – you can also specify the file once you are in the Development Suite (EDS).

Selecting the FLASH File

- Click the ***<Browse>*** button
- Browse to and select the file you wish to load and then select ***<OK>***
- --> The file will be automatically loaded into the 'FLASH File Preview' window - see below...



Flash File Preview

□ View entire area ☑ Discard leading 0xFF ☑ Discard trailing 0xFF up to page boundary

```
0x00000: 02 18 00 02 18 03 6C 88 25 E8 22 02 18 0B 6D C3  ......l▌%è"...mÃ
0x00010: 41 62 22 02 18 13 03 47 03 5B FF 02 18 1B 7F 01  Ab"....G.[ÿ...▌.
0x00020: 22 D3 22 02 18 23 A2 42 92 30 22 02 18 2B C2 36  "Ó"..#¢B'0"..+Â6
0x00030: 22 FF FF 02 18 33 A2 44 92 37 22 02 18 3B 02 21  "ÿÿ..3¢D'7"..;.!
0x00040: B3 FF FF 02 18 43 02 2A D4 FF FF 02 18 4B E4 A1  ³ÿÿ..C.*Ôÿÿ..Kä¡
0x00050: 70 FF FF 02 18 53 C2 21 D2 1E 22 02 18 5B 02 61  pÿÿ..SÂ!Ò."..[.a
0x00060: E4 FF FF 02 18 63 E4 FF 02 0F 5B 02 18 6B E5 F7  äÿÿ..cäÿ..[..kå÷
0x00070: 13 22 FF 02 18 73 C2 D5 E8 30 E7 0F B2 D5 E4 C3  ."ÿ..sÂÕè0ç.²ÕäÃ
0x00080: 9B FB E4 9A FA E4 99 F9 E4 98 F8 EC 30 E7 17 B2  ▌ûä▌úä▌ùä▌øì0ç.²
```

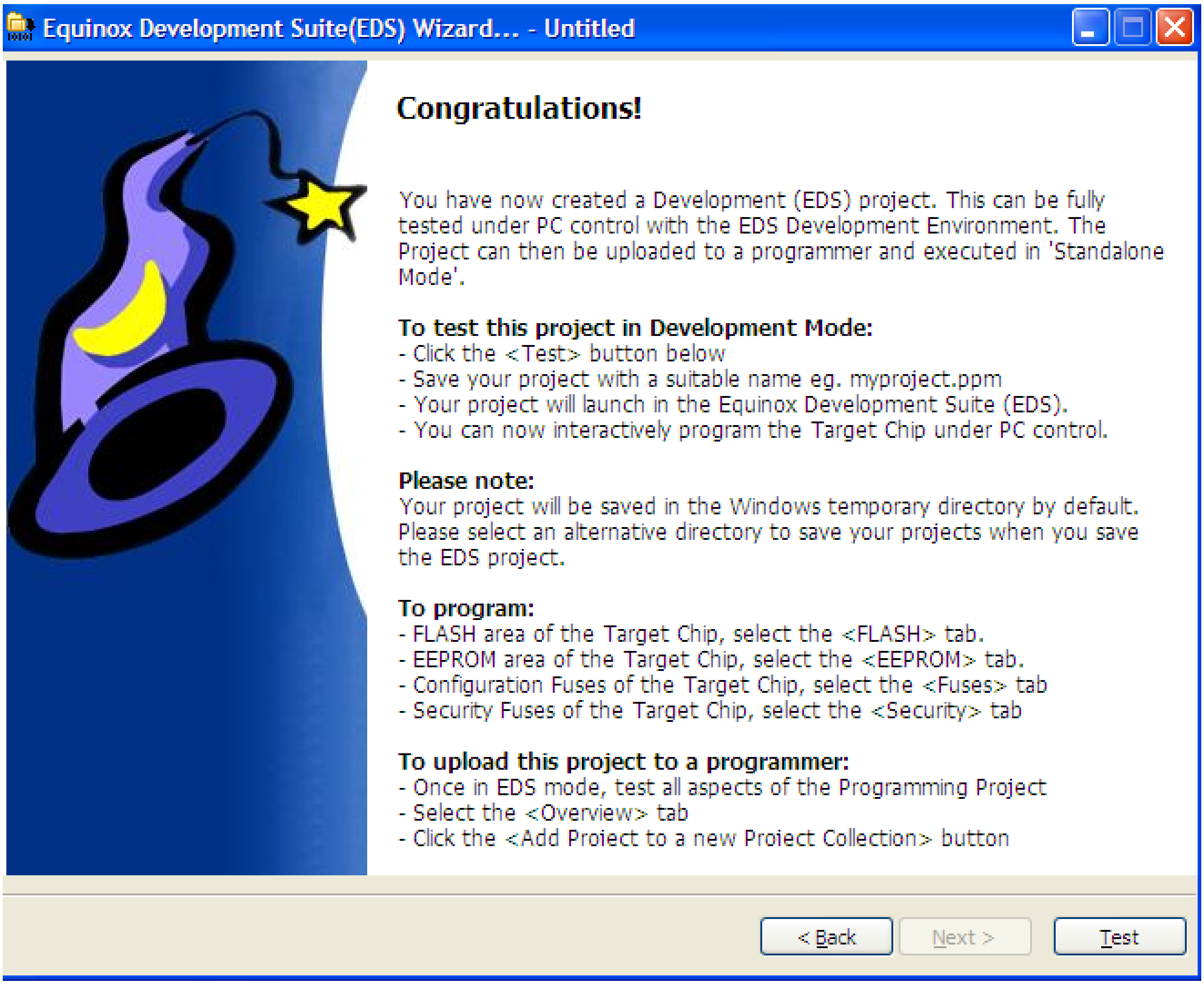
Sigma 500 Checksum: Input file CRC = 0xFFA83B10, Calculated CRC = 0xFFA83B10

Min Addr. = 0x0000 | Max Addr. = 0x1FFFF | Bytes Loaded = 57264 (0xDFB0) | CRC = 0xFFA83B10 | File Format: Intel Hex (Generic)

- 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.

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 Windows temporary 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

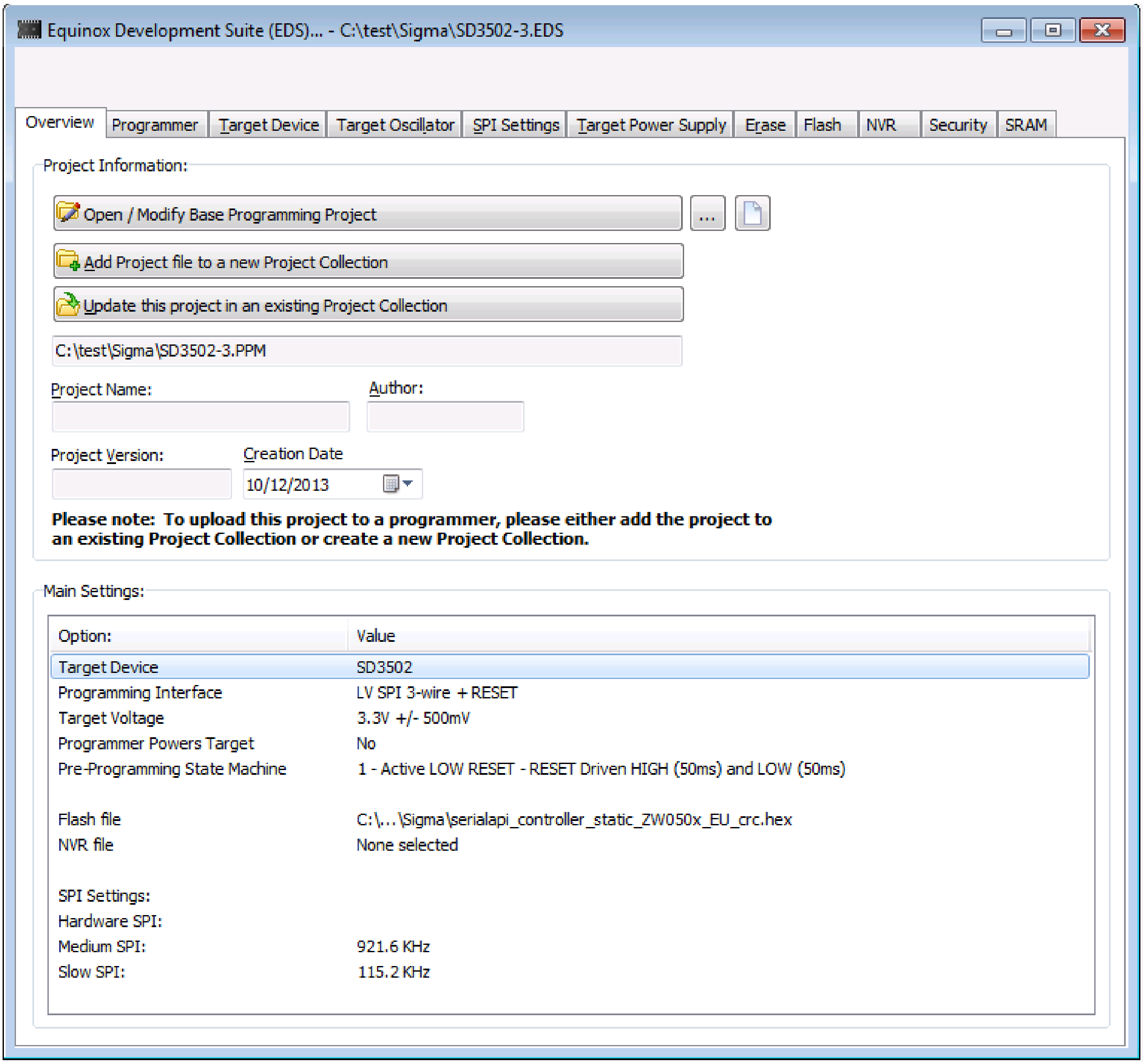
To upload this project to a programmer:
- Once in EDS mode, test all aspects of the Programming Project
- Select the <Overview> tab
- Click the <Add Project to a new Project Collection> button

Enter a name for the EDS project e.g. ***SD3502*** 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:\test\Sigma\SD3502-3.EDS
Overview
Programmer
Target Device
Target Oscillator
SPI Settings
Target Power Supply
Erase
Flash
NVR
Security
SRAM
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:\test\Sigma\SD3502-3.PPM

Project Name:

Author:

Project Version:

Creation Date: 10/12/2013

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	SD3502
Programming Interface	LV SPI 3-wire + RESET
Target Voltage	3.3V +/- 500mV
Programmer Powers Target	No
Pre-Programming State Machine	1 - Active LOW RESET - RESET Driven HIGH (50ms) and LOW (50ms)
Flash file	C:\...\Sigma\serialapi_controller_static_ZW050x_EU_crc.hex
NVR file	None selected
SPI Settings:	
Hardware SPI:	
Medium SPI:	921.6 KHz
Slow SPI:	115.2 KHz

4.2 EDS - Default settings for SPI, statemachine etc

The following default settings will be used:

- ***'Hardware SPI'*** interface

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:

- Test the Target Voltage

4.3 SPI - speed settings

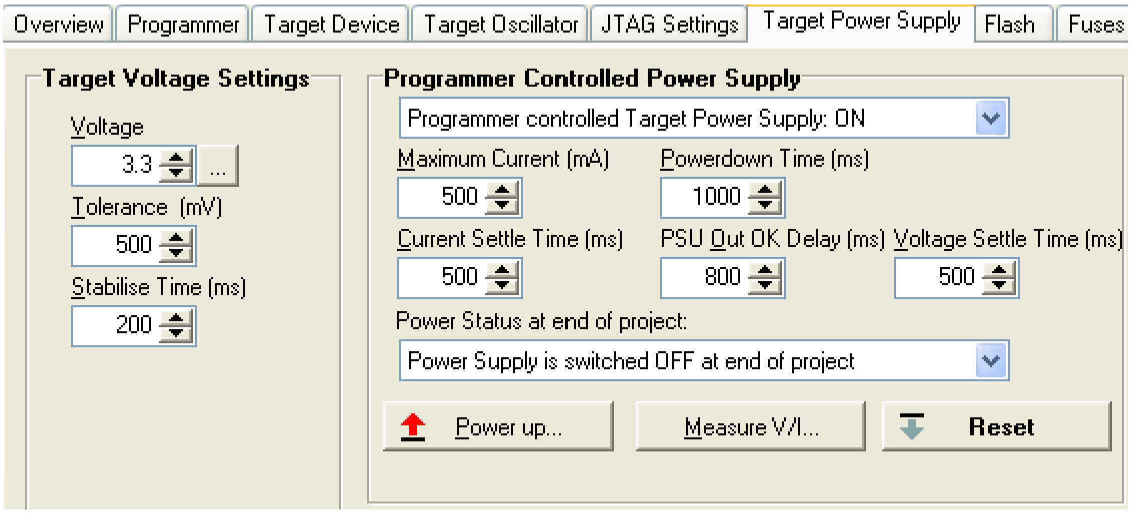
The ***'SPI speed'*** should be set up before any programming operation can take place.

4.4 Checking the Target Voltage

It is a good idea to check that the target device is powered at the correct voltage before trying to program it. A Sigma Z-Wave device normally runs at between 3.0 and 3.6V. 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 Z-Wave 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 NOT 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’***.

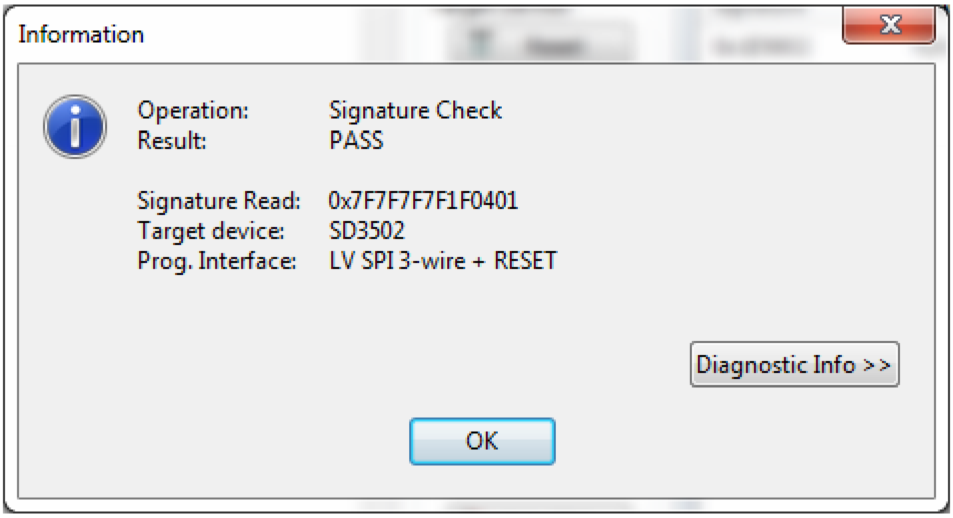
- The voltage should be within 3.0 and 3.6V.

4.5 Testing SPI communication with the 500 series device

4.5.1 Overview

To make sure that the programmer can communicate with the target 500 series device, try reading back the ***Device Signature (Device 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 Check – Result: Pass'***.



Information

Operation: Signature Check
Result: PASS

Signature Read: 0x7F7F7F7F1F0401
Target device: SD3502
Prog. Interface: LV SPI 3-wire + RESET

- The ***Signature (Device ID)*** is displayed e.g. 0x7F7F7F7F1F0401

This message means that the programmer has established a connection via the SPI interface to the specified target device and that the device has the correct ***'Signature / Device 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…..

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.

Appendix 1 - Sigma Programming / Calibration Project setup

1.0 Overview

This section describes how the programming / calibration sequence for a ***'Sigma Z-Wave 500 series device'*** is set up in EQTools.

2.0 Explanation of 'standalone projects'

The programming / calibration sequence for a ***'Sigma Z-Wave 500 series device'***
is made up of 4 x ***'Standalone Programming Projects'*** as follows:

Project 1: SD3502-SPI
The first project called **'SD3502-SPI '** is the so-called ***'Base Project'***.
This project is used by the ***'programming script'*** in ISP-PRO to define the target device, voltage, SPI speed etc.

Project 1: CALIBRATION
This project programs the ***'Calibration firmware'*** into the FLASH area of the target device.
This ***'Calibration firmware'*** firmware must have been compiled for the correct Sigma device and be the correct algorithm for the hardware being calibrated.

Project 2: RUNTARGET
This project programs simply powers up the Target System and then forces the ***'Calibration firmware'*** to run / execute the ***'Calibration firmware'*** by asserting the RESET pin of the DUT.

Project 3: PRODUCTION
This project programs the ***'Production firmware'*** into the FLASH area of the target device.
The ***'Production firmware'*** is the final customer firmware which needs to be programmed into the device before the product leaves the factory.

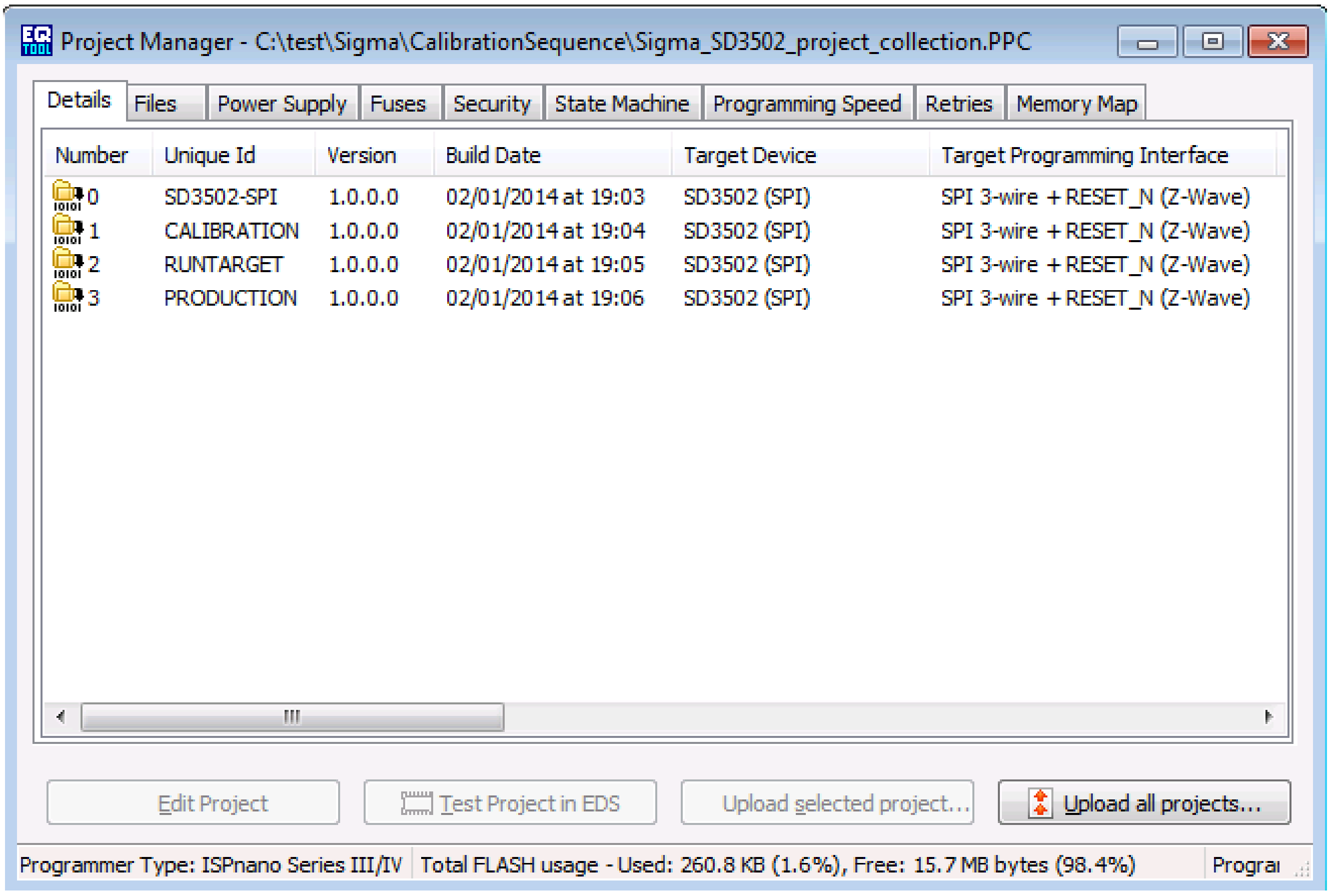
Please note:
For most applications, the only project which needs to be modified is the ***'PRODUCTION'*** project.
This project programs the ***'Customer firmware file'*** and so will probably be unique to each customer application. The next sections explain how to change the ***'Customer firmware file'*** in the ***'PRODUCTION'*** project.

2.1 Opening the Sigma Project Collection

To open the ***'Sigma Project Collection'***....

- Start EQTools
- From the top menu bar, select ***'File - open'*** and browse to and select the ***'Project Collection'*** file ***'Sigma_SD3502_project_collection.PPC'***

➔ The Project Collection should open in ***'Project Manager'*** view - see screenshot below.....



Project Manager - C:\test\Sigma\CalibrationSequence\Sigma_SD3502_project_collection.PPC

Number	Unique Id	Version	Build Date	Target Device	Target Programming Interface
0	SD3502-SPI	1.0.0.0	02/01/2014 at 19:03	SD3502 (SPI)	SPI 3-wire + RESET_N (Z-Wave)
1	CALIBRATION	1.0.0.0	02/01/2014 at 19:04	SD3502 (SPI)	SPI 3-wire + RESET_N (Z-Wave)
2	RUNTARGET	1.0.0.0	02/01/2014 at 19:05	SD3502 (SPI)	SPI 3-wire + RESET_N (Z-Wave)
3	PRODUCTION	1.0.0.0	02/01/2014 at 19:06	SD3502 (SPI)	SPI 3-wire + RESET_N (Z-Wave)

Programmer Type: ISPnano Series III/IV | Total FLASH usage - Used: 260.8 KB (1.6%), Free: 15.7 MB bytes (98.4%) | Prograı

As you can see, the ***'Project Collection'*** comprises of 4 x ***'Standalone Programming Projects'***.

Please see section 2.0 for an explanation of the function of each project.

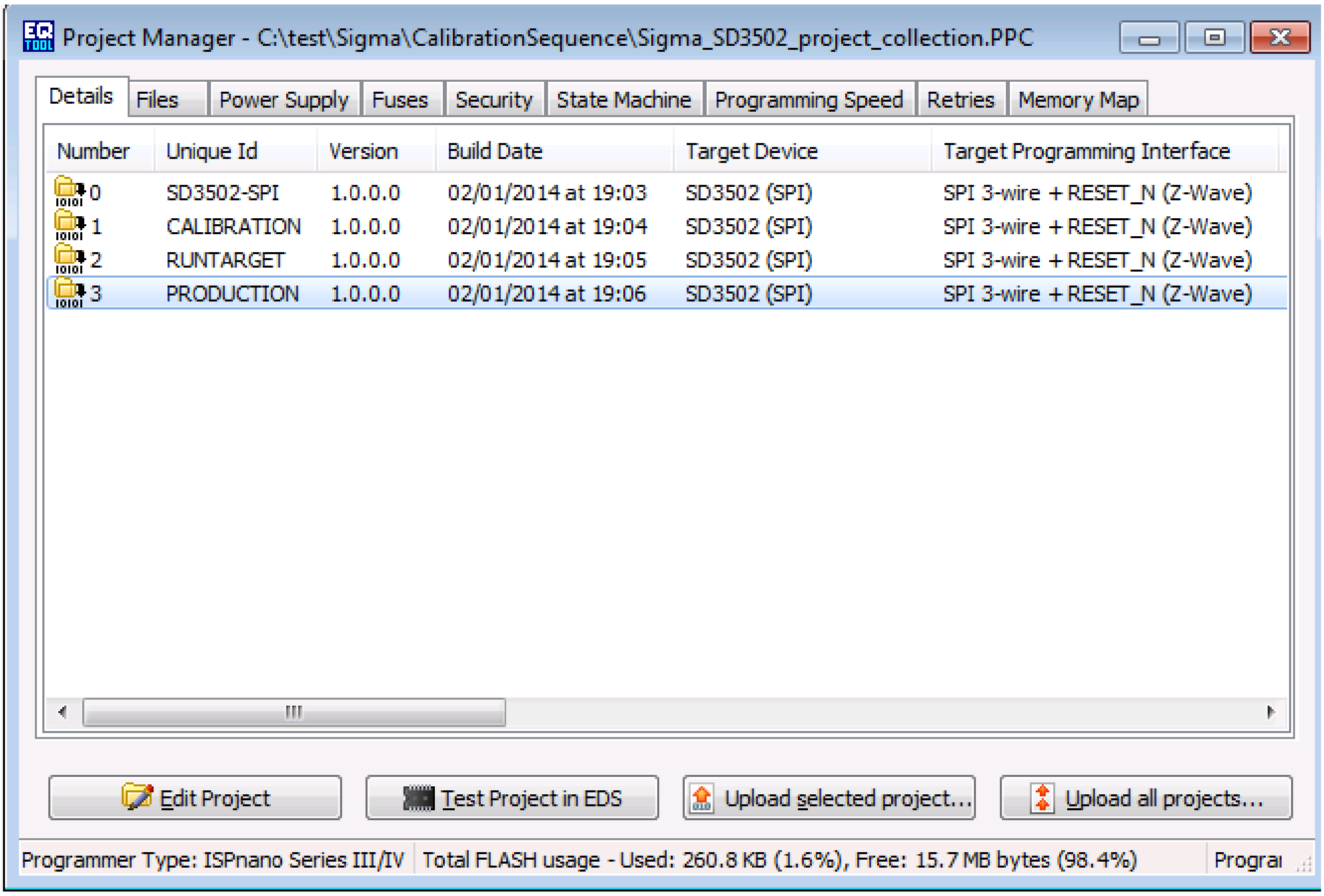
2.2 Changing the 'Product firmware' file

The test ***'Project Collection'*** is shipped with an example ***'Production firmware'*** from Sigma Designs. To program your own ***'Product firmware'*** into the target Z-Wave device, you will need to change the ***'Firmware file'*** specified in the ***'PRODUCTION'*** project to your own ***'Product firmware'*** file.

To change the ***'Product firmware file'***....

1. Open the ***'Sigma Project Collection'*** - see section 2.1 for instructions

2. Highlight the ***'PRODUCTION'*** project by clicking on the project name once
--> see screenshot below....



Project Manager - C:\test\Sigma\CalibrationSequence\Sigma_SD3502_project_collection.PPC

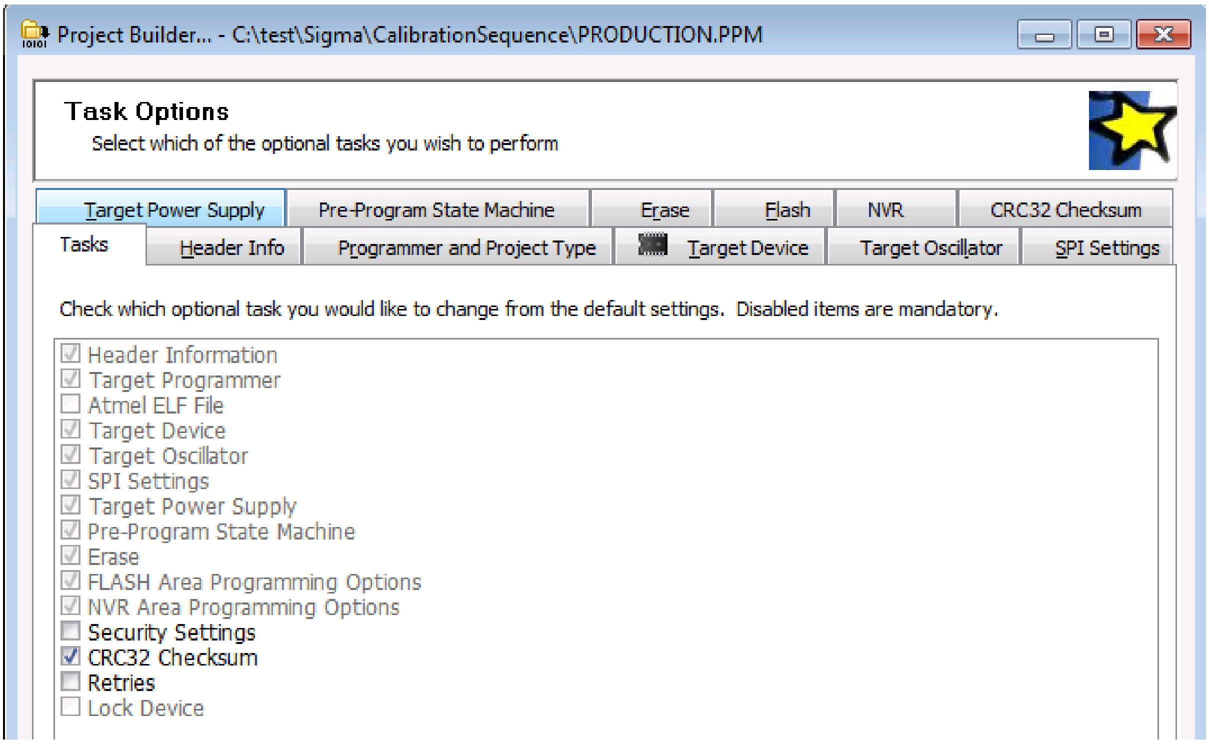
Number	Unique Id	Version	Build Date	Target Device	Target Programming Interface
0	SD3502-SPI	1.0.0.0	02/01/2014 at 19:03	SD3502 (SPI)	SPI 3-wire + RESET_N (Z-Wave)
1	CALIBRATION	1.0.0.0	02/01/2014 at 19:04	SD3502 (SPI)	SPI 3-wire + RESET_N (Z-Wave)
2	RUNTARGET	1.0.0.0	02/01/2014 at 19:05	SD3502 (SPI)	SPI 3-wire + RESET_N (Z-Wave)
3	PRODUCTION	1.0.0.0	02/01/2014 at 19:06	SD3502 (SPI)	SPI 3-wire + RESET_N (Z-Wave)

Programmer Type: ISPnano Series III/IV | Total FLASH usage - Used: 260.8 KB (1.6%), Free: 15.7 MB bytes (98.4%) | Prograi

3. To open the ***'PRODUCTION'*** project
- Click the 'Edit Project' button or double-click the ***'PRODUCTION'*** project name in the list.
--> The ***'PRODUCTION'*** project will now open in ***'Project Builder'*** view.

4. Overview of the 'Task options' tab

The project will open on the ***'Tasks'*** tab which shows the various ***'Tasks'*** which can be performed by a ***'standalone project'***....



Project Builder... - C:\test\Sigma\CalibrationSequence\PRODUCTION.PPM

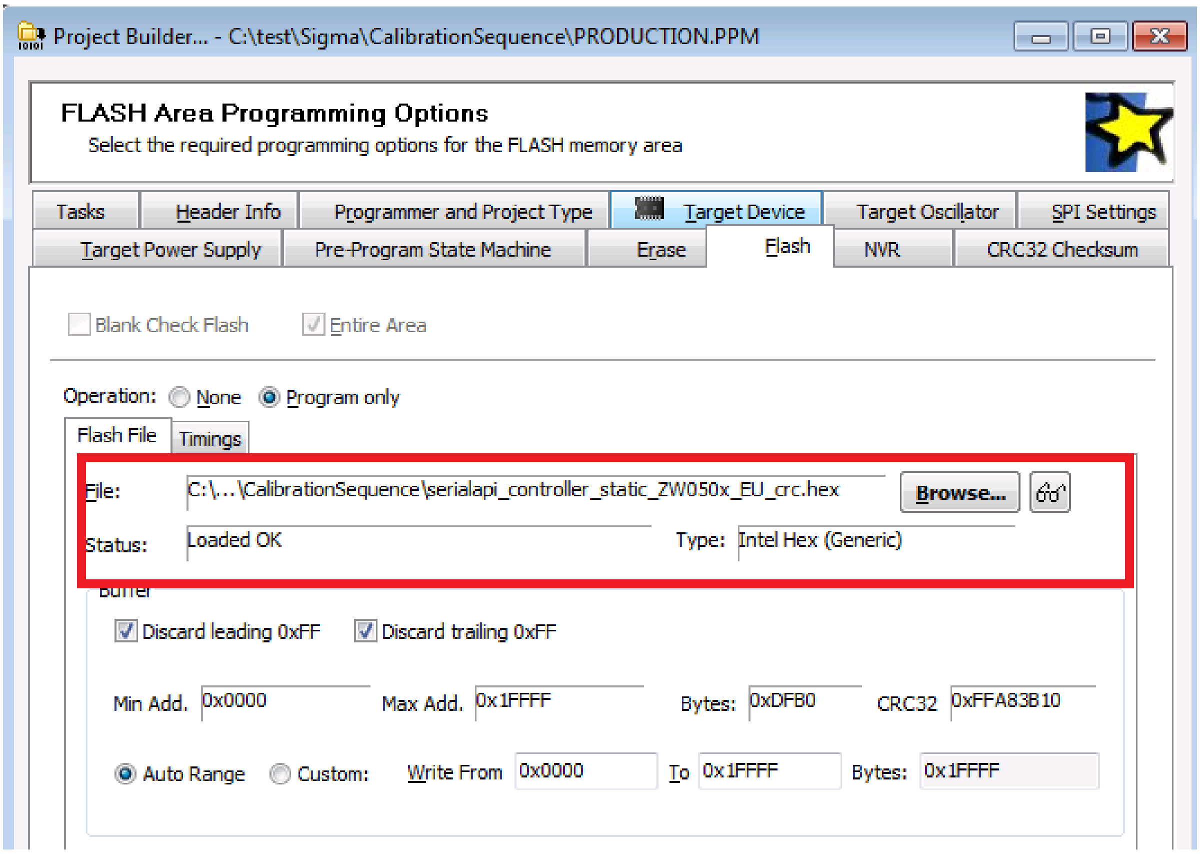
Task Options

Select which of the optional tasks you wish to perform

Check which optional task you would like to change from the default settings. Disabled items are mandatory.

- ☑ Header Information
- ☑ Target Programmer
- ☐ Atmel ELF File
- ☑ Target Device
- ☑ Target Oscillator
- ☑ SPI Settings
- ☑ Target Power Supply
- ☑ Pre-Program State Machine
- ☑ Erase
- ☑ FLASH Area Programming Options
- ☑ NVR Area Programming Options
- ☐ Security Settings
- ☑ CRC32 Checksum
- ☐ Retries
- ☐ Lock Device

5. Select the ***'FLASH'*** tab.



Project Builder... - C:\test\Sigma\CalibrationSequence\PRODUCTION.PPM

FLASH Area Programming Options

Select the required programming options for the FLASH memory area

☐ Blank Check Flash ☑ Entire Area

Operation: ○ None ● Program only

File: C:\...\CalibrationSequence\serialapi_controller_static_ZW050x_EU_crc.hex

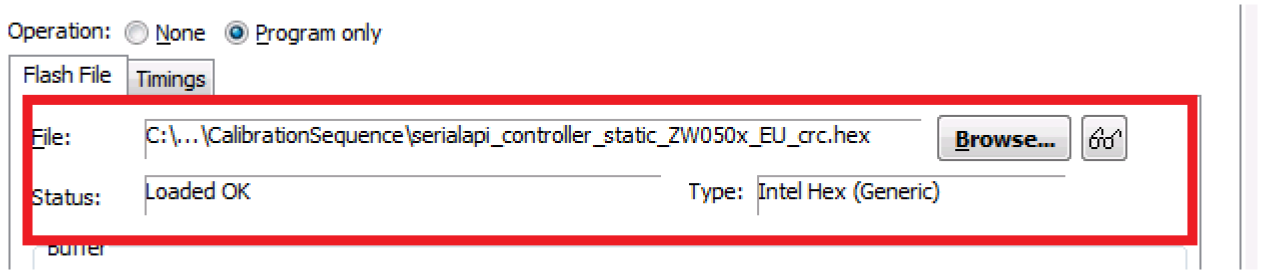
Status: Loaded OK Type: Intel Hex (Generic)

☑ Discard leading 0xFF ☑ Discard trailing 0xFF

Min Add. 0x0000 Max Add. 0x1FFFF Bytes: 0xDFB0 CRC32 0xFFA83B10

● Auto Range ○ Custom: Write From 0x0000 To 0x1FFFF Bytes: 0x1FFFF

The project is currently setup with an example ***'firmware file'*** from Sigma as follows.....



Operation: ○ None ● Program only

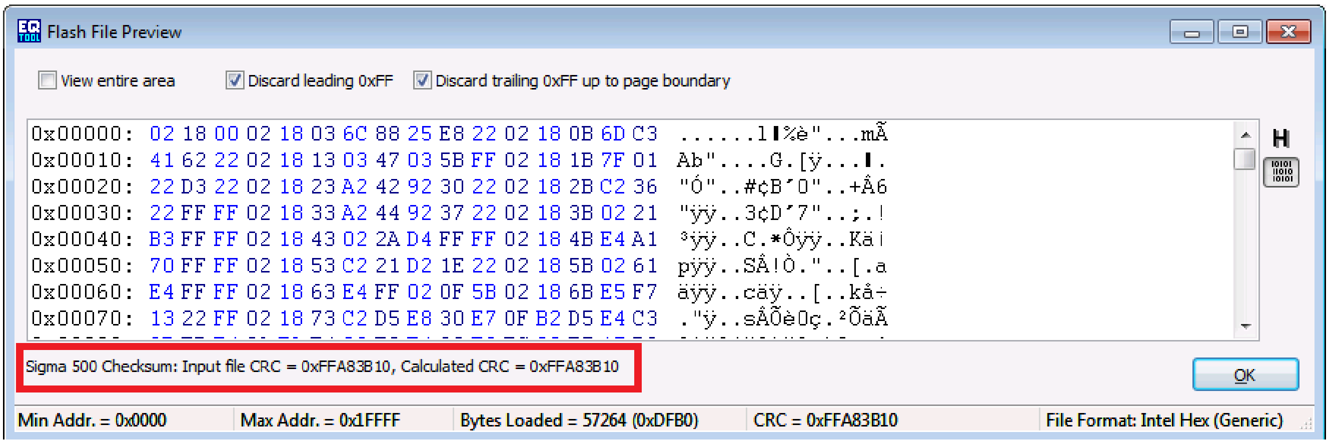
File: C:\...\CalibrationSequence\serialapi_controller_static_ZW050x_EU_crc.hex

Status: Loaded OK

Type: Intel Hex (Generic)

6. To change the ***'Firmware file'***....
Click the ***'Browse'*** button and then browse to and select the file you want to load.
This file can be a binary, Intel Hex or Motorola S-Record format file

--> The ***'FLASH File Preview'*** window is now displayed.....



□ View entire area ☑ Discard leading 0xFF ☑ Discard trailing 0xFF up to page boundary

```
0x00000: 02 18 00 02 18 03 6C 88 25 E8 22 02 18 0B 6D C3  ......l▮%è"...mÃ
0x00010: 41 62 22 02 18 13 03 47 03 5B FF 02 18 1B 7F 01  Ab"....G.[ÿ...▮.
0x00020: 22 D3 22 02 18 23 A2 42 92 30 22 02 18 2B C2 36  "Ó"..#¢B'0"..+Â6
0x00030: 22 FF FF 02 18 33 A2 44 92 37 22 02 18 3B 02 21  "ÿÿ..3¢D'7"..;.!
0x00040: B3 FF FF 02 18 43 02 2A D4 FF FF 02 18 4B E4 A1  ³ÿÿ..C.*Ôÿÿ..Kä¡
0x00050: 70 FF FF 02 18 53 C2 21 D2 1E 22 02 18 5B 02 61  pÿÿ..SÂ!Ò."..[.a
0x00060: E4 FF FF 02 18 63 E4 FF 02 0F 5B 02 18 6B E5 F7  äÿÿ..cäÿ..[..kå÷
0x00070: 13 22 FF 02 18 73 C2 D5 E8 30 E7 0F B2 D5 E4 C3  ."ÿ..sÂÕè0ç.²ÕäÃ
```

Sigma 500 Checksum: Input file CRC = 0xFFA83B10, Calculated CRC = 0xFFA83B10

Min Addr. = 0x0000 | Max Addr. = 0x1FFFF | Bytes Loaded = 57264 (0xDFB0) | CRC = 0xFFA83B10 | File Format: Intel Hex (Generic)

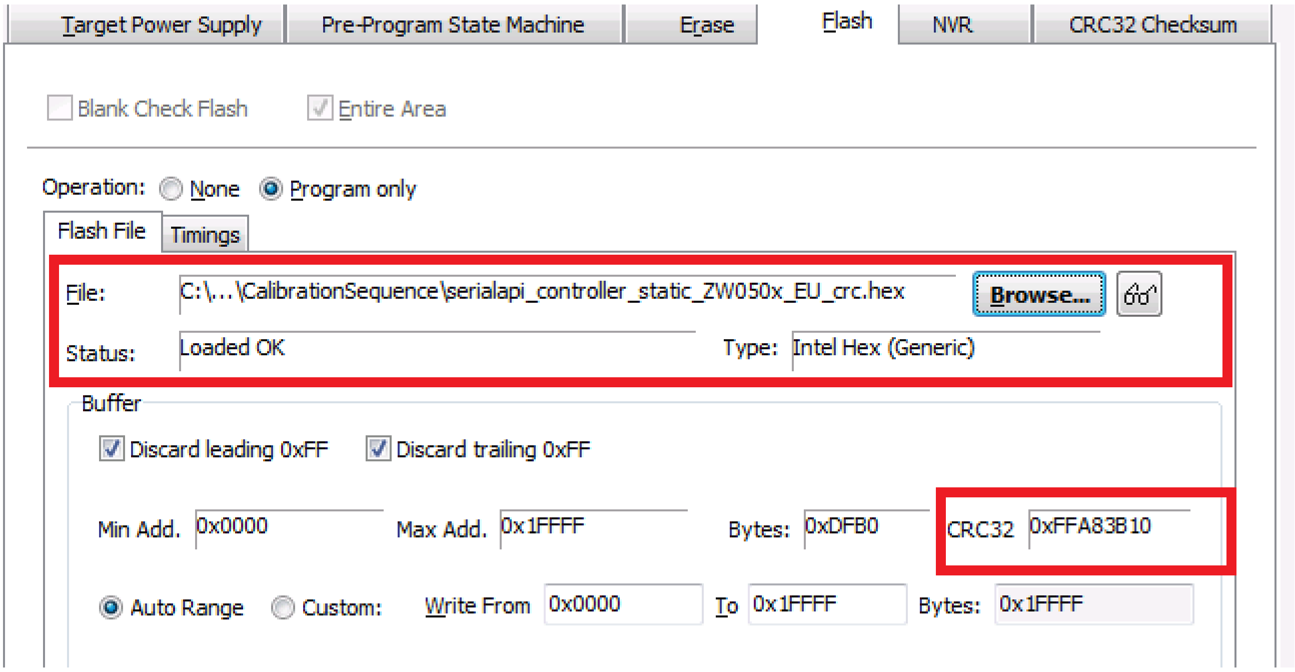
This window displays the following information about the selected input file....

- A preview of the data in the file in both Hex and ASCII format
- The ***'Sigma CRC32 FLASH Checksum'*** value stored in the file (if present)
- The ***'CRC32 FLASH Checksum'*** calculated by EQTools when the file was loaded

!!! Important !!!
i. The input file must have a ***'Sigma CRC32 FLASH Checksum'*** value stored in the last 4 bytes of the file.
ii. The ***'Sigma CRC32 FLASH Checksum'*** value stored in the file must be the same as the ***'CRC32 FLASH Checksum'*** calculated by EQTools when the file was loaded

Once you are happy that the selected input file is OK, click the ***'OK'*** button to load it into the project.

7. The selected ***'Firmware file'*** should now be displayed.
The ***'CRC32 FLASH Checksum'*** calculated by EQTools when the file was loaded is also displayed.
See screenshot below....



□ Blank Check Flash ☑ Entire Area

Operation: ○ None ● Program only

File: C:\...\CalibrationSequence\serialapi_controller_static_ZW050x_EU_crc.hex

Status: Loaded OK Type: Intel Hex (Generic)

☑ Discard leading 0xFF ☑ Discard trailing 0xFF

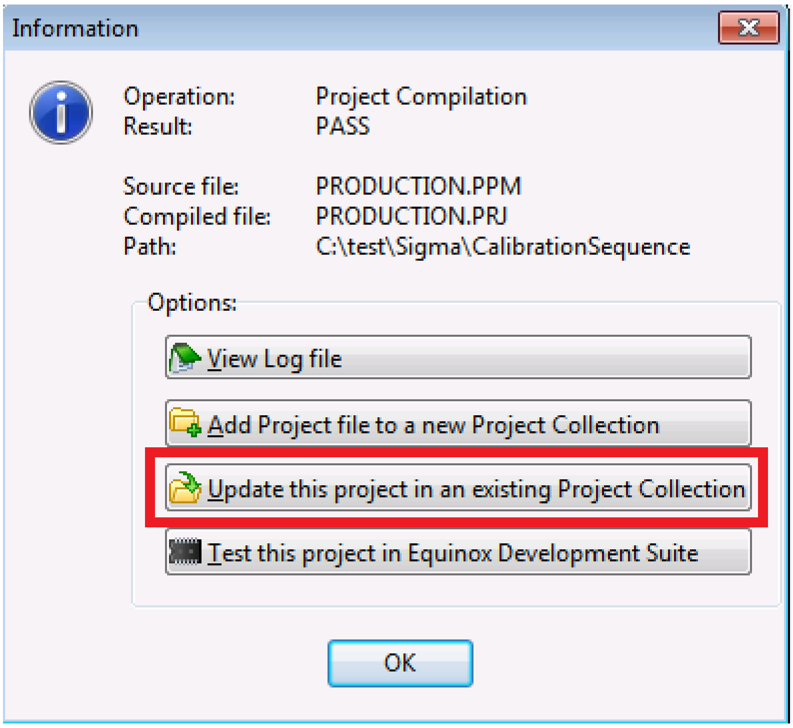
Min Add. 0x0000 Max Add. 0x1FFFF Bytes: 0xDFB0 CRC32 0xFFA83B10

● Auto Range ○ Custom: Write From 0x0000 To 0x1FFFF Bytes: 0x1FFFF

8. Compile the project

The revised ***'PRODUCTION'*** project must be compiled and then updated in the Project Collection before it can be uploaded to the programmer.

On the top EQTools icon bar, click the ***'Compile'*** icon.
This will compile the project and then display the following Info screen....



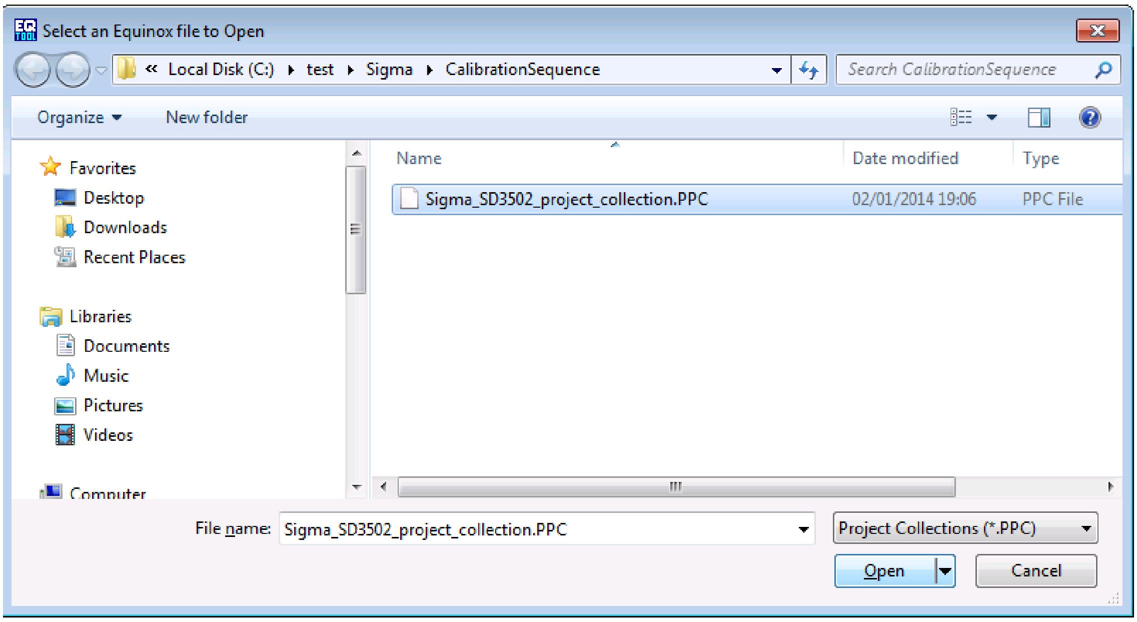
Information

Operation: Project Compilation
Result: PASS

Source file: PRODUCTION.PPM
Compiled file: PRODUCTION.PRJ
Path: C:\test\Sigma\CalibrationSequence

Options:
View Log file
Add Project file to a new Project Collection
Update this project in an existing Project Collection
Test this project in Equinox Development Suite
OK

- Now click the ***'Update this project in an existing Project Collection'*** button.
- Select the ***'Sigma_SD3502_project_collection.PPC'*** file and click the ***'Open'*** button - see screenshot...



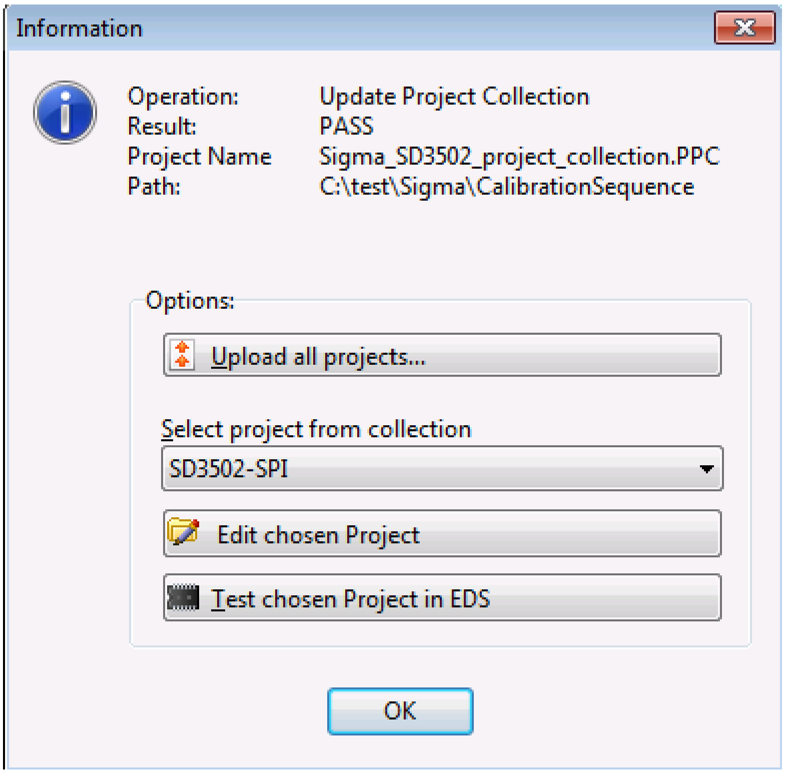
Select an Equinox file to Open

Name	Date modified	Type
Sigma_SD3502_project_collection.PPC	02/01/2014 19:06	PPC File

File name: Sigma_SD3502_project_collection.PPC

Project Collections (*.PPC)

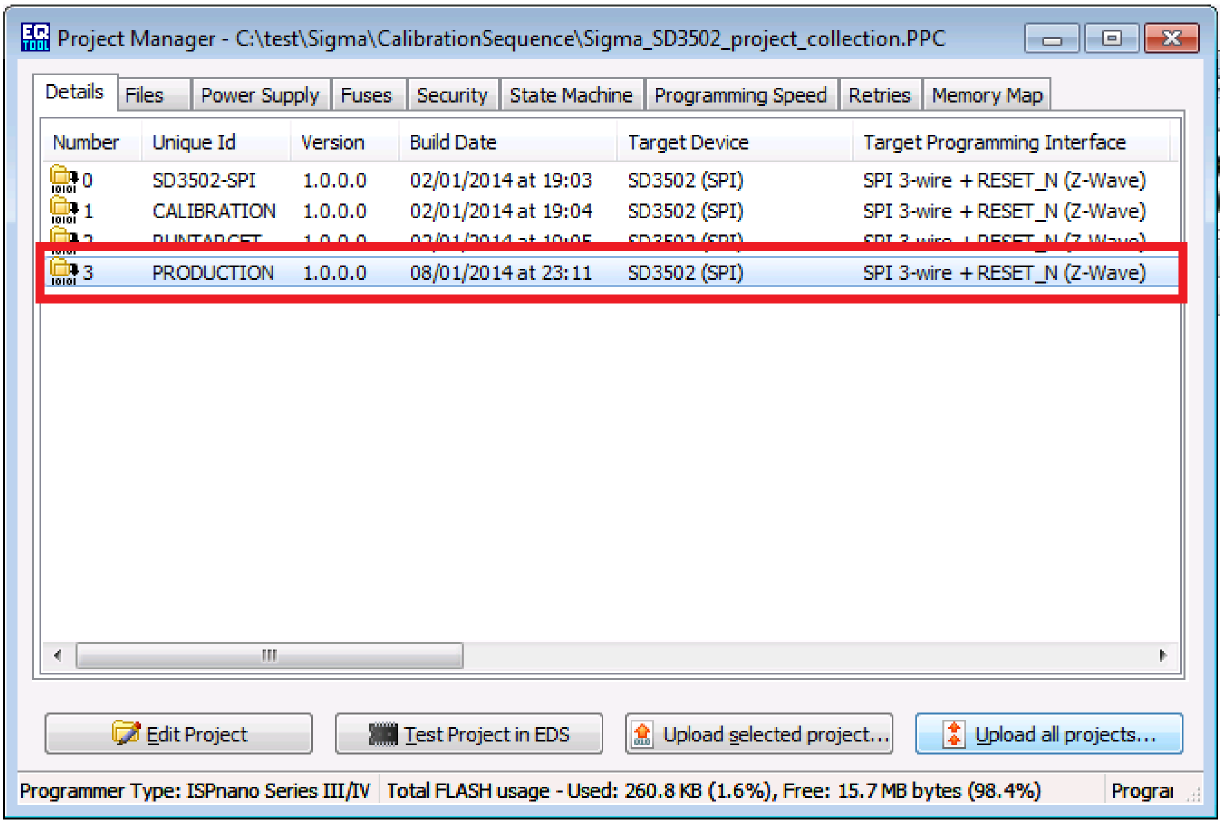
The ***'Update Project Collection'*** process should now report ***'PASS'***........



Operation: Update Project Collection
Result: PASS
Project Name Sigma_SD3502_project_collection.PPC
Path: C:\test\Sigma\CalibrationSequence

Click the ***'OK'*** button to exit this screen.

The ***'PRODUCTION'*** project should now have been updated with the new ***'firmware file'*** and the ***'Build date'*** should show the new date and time of the file.



Project Manager - C:\test\Sigma\CalibrationSequence\Sigma_SD3502_project_collection.PPC
Details
Files
Power Supply
Fuses
Security
State Machine
Programming Speed
Retries
Memory Map

Number	Unique Id	Version	Build Date	Target Device	Target Programming Interface
0	SD3502-SPI	1.0.0.0	02/01/2014 at 19:03	SD3502 (SPI)	SPI 3-wire + RESET_N (Z-Wave)
1	CALIBRATION	1.0.0.0	02/01/2014 at 19:04	SD3502 (SPI)	SPI 3-wire + RESET_N (Z-Wave)
2	RUNTARGET	1.0.0.0	02/01/2014 at 19:05	SD3502 (SPI)	SPI 3-wire + RESET_N (Z-Wave)
3	PRODUCTION	1.0.0.0	08/01/2014 at 23:11	SD3502 (SPI)	SPI 3-wire + RESET_N (Z-Wave)

Programmer Type: ISPnano Series III/IV | Total FLASH usage - Used: 260.8 KB (1.6%), Free: 15.7 MB bytes (98.4%) | Progra

2.3 Uploading the new Project Collection to the programmer

Once you have updated the ***'PRODUCTION'*** project with your ***'firmware file'***, it is then necessary to upload the entire ***'Project Collection'*** to the programmer.

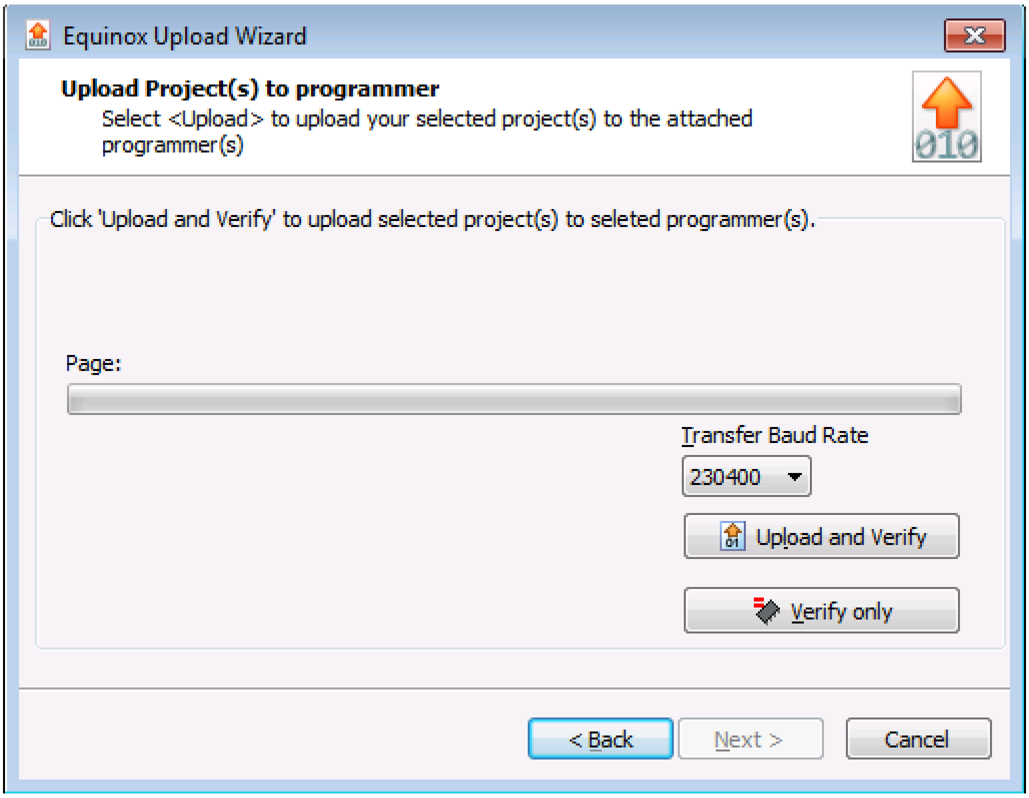
To upload the ***'Project Collection'*** to the programmer....

1. Make sure the ***'Project Collection'*** is already open in EQTools

2. Make sure the programmer is attached to the PC and is powered on

3. Click the ***'Upload all projects'*** button (bottom right of the Project Manager window)

4. If everything is OK, then the ***'Upload Wizard'*** utility will start and the following screen will be displayed.....



Equinox Upload Wizard

Upload Project(s) to programmer

Select <Upload> to upload your selected project(s) to the attached programmer(s)

Click 'Upload and Verify' to upload selected project(s) to seleted programmer(s).

Page:

Transfer Baud Rate: 230400

5. Click the ***'Upload and Verify'*** button to start the upload process

6. Follow the on-screen instructions to upload the Project Collection

7. Once complete, the projects will then be permanently resident in the ***'Programmer FLASH memory Store'***.

Important note:

It is also possible to upload the ***Project Collection*** using ISP-PRO.

In ISP-PRO, select the ***'Programming Script File'*** and then click the ***'Upload Project'*** button.

Appendix 2 - Setting up the 'Tx Power' parameters

1.0 Overview

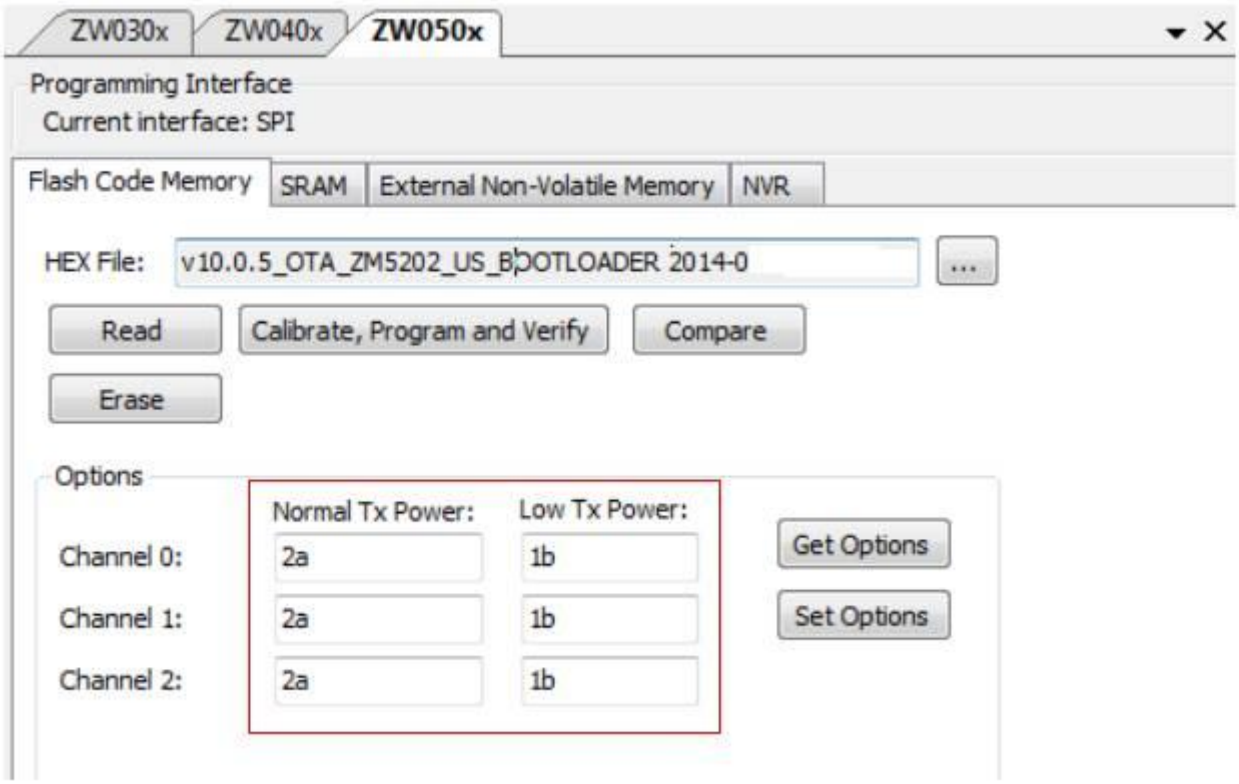
This section describes how to set up the ***'Tx Power'*** parameters for a Z-Wave 500 series module or SOC device.

The values for the ***'Tx Power'*** parameters must be derived by experimentation during the development and final RD testing stages of the customer product. The values are then usually fixed for this product and must be programmed into certain specific locations in the ***'FLASH area'*** of the Z-Wave device. As these parameter values are fixed values, then these values should be placed in the ***'FLASH hex file'*** which is used to program the device in production.

Important note:
It is not possible to over-program the ***'Tx Power'*** parameters after the main FLASH firmware has been programmed. This is because the CRC32 checksum used to validate that the FLASH has been programmed correctly would need to be changed when the ***'Tx Power'*** parameters were programmed.

1.2 Where do I find the ‘Tx Power’ settings in Sigma’s SDK

The ***'Tx Power'*** parameters for a Z-Wave 500 series module or SOC device will usually have already been setup / tested by an RF engineer during the RF testing of the customer product. If the development engineer has used Sigma’s own SDK software, then you should be able to obtain the required ***'Tx Power'*** parameters from the following screen in Sigma’s GUI interface – see screenshot below…..



Programming Interface
Current interface: SPI

HEX File: v10.0.5_OTA_ZM5202_US_BOOTLOADER 2014-0

Options

	Normal Tx Power:	Low Tx Power:
Channel 0:	2a	1b
Channel 1:	2a	1b
Channel 2:	2a	1b

If you have a “golden sample” of your product which already have the correct ***'Tx Power'*** parameters programmed into it, then it is possible to read out the values by clicking the ***‘Get options’*** button on the above screen.

Unfortunately, there is no way to export these settings to the Equinox EQTools software, so please make a note of the values for each parameter. You will need these values to enter into hex file.

For many customer products, it may be possible to simply use the default values for these settings. However, the values used should always be double-checked either with your ‘RF engineer’ or with Sigma technical support.

1.3 'Tx Power' parameters – overview of merging process

The ***'Tx Power'*** parameters for your product must be merged into the ***'Production FLASH hex file'*** so that they are automatically programmed into the target device at the same time as the 'production firmware'.

An overview of the steps to integrate your ***'Tx Power'*** parameters into your final ***'Production FLASH Area hex file'*** is shown below….

1. Obtain the correct values for ***'Tx Power'*** parameters either by experimentation or by reading out the values from a 'golden sample' device which has the correct parameters in it.

2. Overlay these ***'Tx Power'*** parameters into your ***'Production FLASH Area hex file'***
- This can now be done using a utility within EQTools.

3. Recalculate the ***'FLASH CRC32 checksum'*** to take account of the values for the ***'Tx Power'*** parameters.
- This task is performed by EDS.

4. Save the amended hex file which now has the updated ***'Tx Power'*** parameters + updated ***'FLASH CRC32 checksum'***.

5. This updated hex file should now be used as your ***'Production FLASH Area hex file'***.

The ***'Production FLASH Area hex file'*** now contains:

- Your ***'Z-Wave Firmware data'***.
- The correct ***'Tx Power'*** parameters for your end product.
- The correct CRC32 FLASH checksum for the entire file.

1.4 'Tx Power' parameters – merging into the FLASH hex file

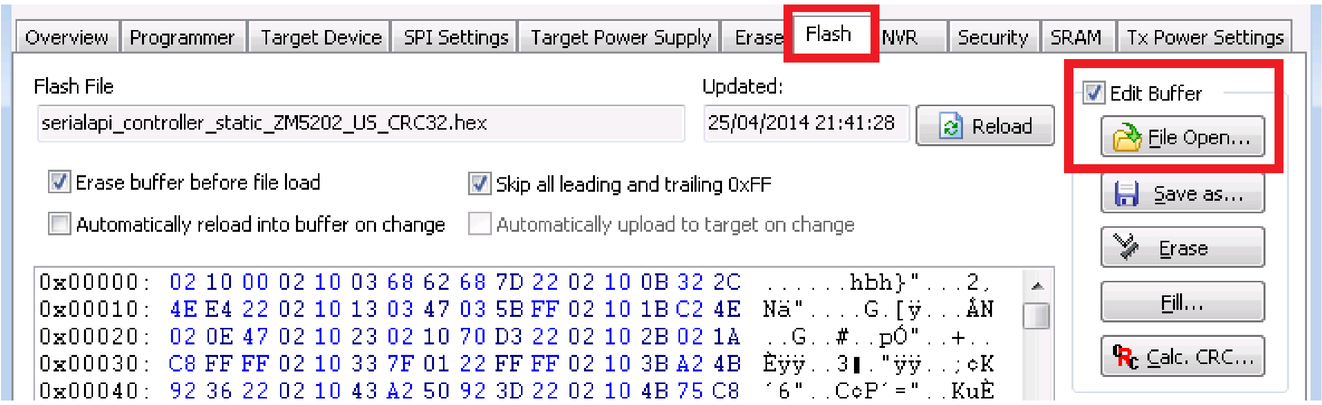
The ***'Tx Power'*** parameters for your product must be merged into the ***'Production FLASH hex file'*** It is possible to perform this task using the EQTools – EDS (Development Mode) utility.

Instructions:

1. Start the Equinox – EQTools software

2. Select the option to ***'Create an EDS - Development Project'*** and follow the wizard to create the EDS project
Or
Open an existing Sigma 500 series EDS project (*.eds)

3. Once the EDS session has started, select the ***'FLASH'*** tab. See screenshot below…..



Flash File: serialapi_controller_static_ZM5202_US_CRC32.hex
Updated: 25/04/2014 21:41:28

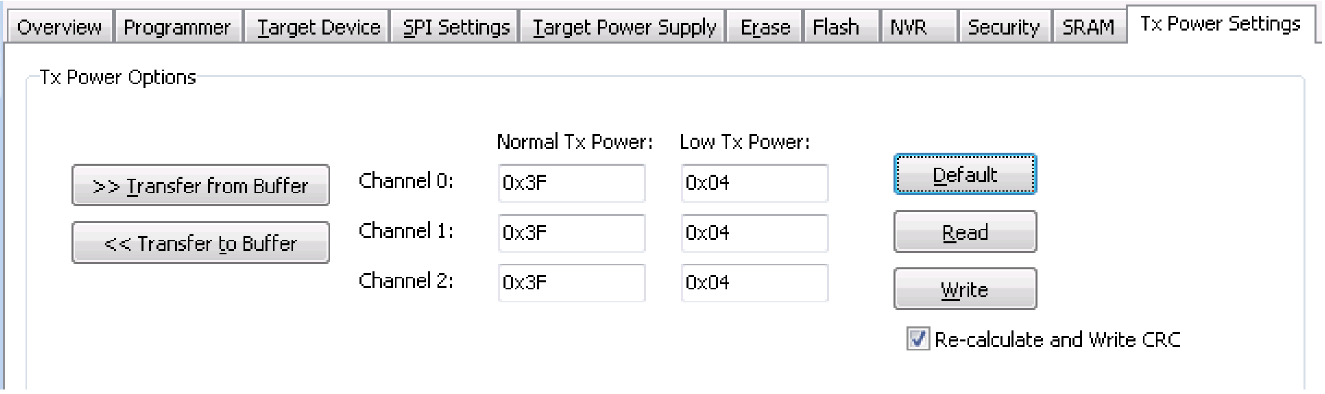
☑ Edit Buffer

☑ Erase buffer before file load
☑ Skip all leading and trailing 0xFF
□ Automatically reload into buffer on change
□ Automatically upload to target on change

```
0x00000:  02 10 00 02 10 03 68 62 68 7D 22 02 10 0B 32 2C  ......hbh}"...2,
0x00010:  4E E4 22 02 10 13 03 47 03 5B FF 02 10 1B C2 4E  Nä"....G.[ÿ...ÅN
0x00020:  02 0E 47 02 10 23 02 10 70 D3 22 02 10 2B 02 1A  ..G..#..pÓ"..+..
0x00030:  C8 FF FF 02 10 33 7F 01 22 FF FF 02 10 3B A2 4B  Èÿÿ..3▮."ÿÿ..;¢K
0x00040:  92 36 22 02 10 43 A2 50 92 3D 22 02 10 4B 75 C8  ´6"..C¢P´="..KuÈ
```

4. Select your ***'Production FLASH hex file'*** as follows…
- Tick the ***'Edit buffer'*** check box on the right-hand side of the screen – see screenshot above
- Click the ***'File open'*** button on the right-hand side of the screen
- Browse to and select the ***'FLASH Firmware file'*** which you want to program into the target device.

5. Enter the required ***'Tx Power Options'*** for your product as follows…
- Select the ***'Tx Power Options'*** tab → the following screen should be displayed…



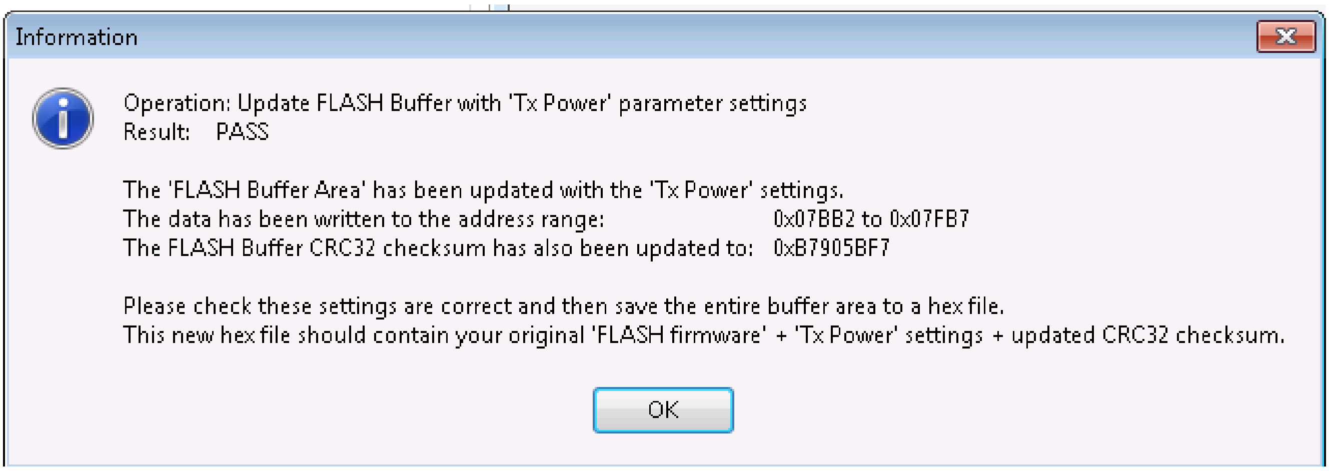
	Normal Tx Power:	Low Tx Power:
Channel 0:	0x3F	0x04
Channel 1:	0x3F	0x04
Channel 2:	0x3F	0x04

☑ Re-calculate and Write CRC

- Enter the correct values for the ***'Tx Power'*** parameters for your product in the relevant fields

- The correct values may be custom to your Z-Wave module or final product so please check the values with your RF engineer or with Sigma's technical support department.

6. Transfer your ***'Tx Power'*** parameter settings to the ***'FLASH Buffer'***
   - Click the ***'Transfer to buffer'*** button to transfer your settings to the ***'FLASH Buffer'.***
   - You should then see the following Information screen

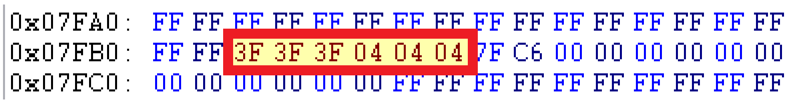


Operation: Update FLASH Buffer with 'Tx Power' parameter settings
Result: PASS

The 'FLASH Buffer Area' has been updated with the 'Tx Power' settings.
The data has been written to the address range: 0x07BB2 to 0x07FB7
The FLASH Buffer CRC32 checksum has also been updated to: 0xB7905BF7

Please check these settings are correct and then save the entire buffer area to a hex file.
This new hex file should contain your original 'FLASH firmware' + 'Tx Power' settings + updated CRC32 checksum.

7. Check the correct values have been transferred to the ***'FLASH Buffer'***....
   - Select the ***'FLASH'*** tab again
   - Go to address 0x7FB2 in the ***'FLASH Buffer'*** (Select CTRL + G + then enter the address: 0x7FB2).
   - You should see the 6 bytes values you entered for the ***'Tx Power'*** parameters now stored at the address range: 0x07FB2 - 0x07FB7 in the FLASH buffer.



```
0x07FA0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x07FB0: FF FF 3F 3F 3F 04 04 04 7F C6 00 00 00 00 00 00
0x07FC0: 00 00 00 00 00 00 FF FF FF FF FF FF FF FF FF FF
```

- The ***'FLASH CRC32 checksum'*** (found in the last 4 bytes of the 'FLASH buffer' will also have been automatically updated by EQTools so it is now correct for the new data you have entered.

8. Save the updated ***'FLASH Buffer'*** back to your ***'FLASH hex file'***......
   - Click the ***'Save as'*** button and then save the entire FLASH area to a new hex file
   - The saved hex file now contains your original FLASH data + ***'Tx Power'*** parameters + updated ***'FLASH CRC32 checksum'.***.
   - This hex file can be used to program the final ***'production firmware'*** into the FLASH area of the target Z-Wave device.

The ***'Production FLASH Area hex file'*** now contains:

- Your ***'Z-Wave Firmware data'***.
- The correct ***'Tx Power'*** parameters for your end product.
- The correct ***'FLASH CRC32 checksum'*** for the entire FLASH file.

Appendix 3 - Configuring the Z-Wave 'External Non-volatile memory (NVM)' parameters

1.0 Overview

It is possible to connect an ***'External Non-volatile memory (NVM)'*** device to a Z-Wave 500 series device. This ***'NVM device'*** is NOT fitted on a Z-Wave module as the module does NOT require the ***'NVM device'*** to operate. Instead, the ***'NVM device'*** can be fitted on the customer's target board and is then connected to the Z-Wave device on the Z-Wave module via the 'SPI1' port of the Z-Wave device. The ***'NVM device'*** is not required for most Z-Wave applications and hence is usually either not catered for on the target board (no footprint provided) or the ***'NVM device'*** is simply not fitted during the assembly process.

However, for some Z-Wave applications, it is necessary to fit the ***'External Non-volatile memory (NVM)'*** device on the customer target board. In this case, it is essential that the relevant ***'External Non-volatile memory (NVM)'*** parameters are configured so that the Z-Wave device knows the relevant settings of the external ***'NVM'*** device.

This section describes how to configure / program the Z-Wave ***'External Non-volatile memory (NVM)'*** parameters to match the configuration of your target Z-Wave board.

1.1 NVM memory – configuration parameters overview

The parameters which are used to configure a Z-Wave 500 series device to interface to an ***'External Non-volatile memory (NVM)'*** device are detailed in the table below.

NVM Parameter	Parameter description	Function of NVM parameter...
NVMT	Non-volatile memory **TYPE**	This parameter defines the type of external memory device fitted to the target board.
NVMS	Non-volatile memory **SIZE**	This parameter defines the physical size in 'kbytes' of the external memory device fitted to the target board.
NVMP	Non-volatile memory **PAGE SIZE**	This parameter defines the physical 'page size' of the external memory device fitted to the target board.
NVMCS	Non-volatile memory **CHIP SELECT**	This parameter configures which pin on the Z-Wave device is used to control the 'Chip Select (CS) signal line of the external memory device fitted to the target board.

2.0 No external NVM (memory) device fitted on target board

If you do not have or plan to fit an ***'External Non-volatile memory (NVM)'*** device on your Z-Wave target board, then there is usually no need to change any of the parameters in the programming script from their default values. If you are programming a Z-Wave ***'module'*** then the memory parameters should already have been factory programmed by Sigma to declare ***'No external NVM fitted'***.

The default factory values of the ***'NVM parameters'*** are shown in the table below....

NVM Parameter	Parameter description	Default value
NVMT	Non-volatile memory **TYPE**	0x00 This indicates that an NVM device is NOT fitted.
NVMS	Non-volatile memory **SIZE**	0xFFFF
NVMP	Non-volatile memory **PAGE SIZE**	0xFFFF
NVMCS	Non-volatile memory **CHIP SELECT**	0x04 ???

Please note:
The default version of the Sigma programming script will simply use the default ***'NVM settings'*** read from the target device and will not change these settings in any way. This plan should work OK for all Z-Wave modules which should have had the ***'NVM settings'*** pre-programmed by Sigma at the factory.

3.0 Custom NVM (memory) device fitted on target board

If you plan to fit an ***'External Non-volatile memory (NVM)'*** device connected to the Z-Wave device on your Z-Wave target board, then it is necessary to configure a set of specific ***'External Non-volatile memory (NVM)'*** parameters in the Z-Wave ***'NVR memory area'*** of the target Z-Wave device. This configuration tells the Z-Wave device what type of ***'External Non-volatile memory (NVM)'*** device is fitted to the device, what the ***'NVM memory size'*** is, what the ***'Page Size'*** is and also which pin on the Z-Wave device should be used to control the ***'Chip Select'*** pin.

Warning!
Failure to declare the settings for the ***'External Non-volatile memory (NVM)'*** device correctly could cause the Z-Wave device to malfunction with newer versions of Z-Wave firmware. It is therefore essential that the ***NVM*** device is correctly configured.

3.1 How to work out the NVM (memory) device parameter values

If you are inheriting a Z-Wave design from an R&D department, then it is likely that the ***'External Non-volatile memory (NVM)'*** device parameters have already been defined for your Z-Wave target board. In this case, you should be able to simply transfer the values you are given by your R&D department directly into the Equinox programming script. If you have been given a working 'golden sample' of your target board, then it may also be possible to read out the relevant ***'External Non-volatile memory (NVM)'*** device parameters from the Z-Wave device on this target board.

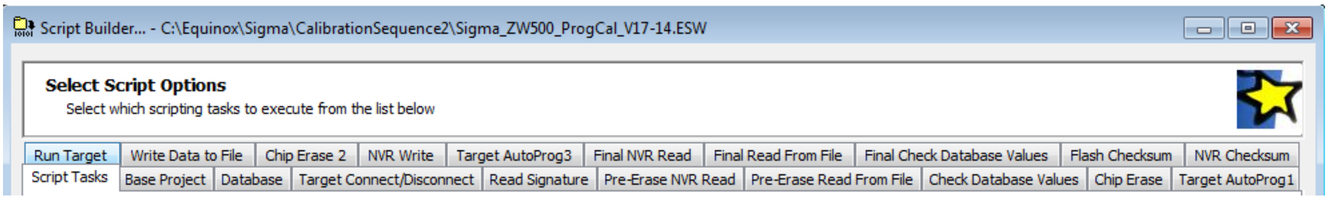
However, if you do not know the relevant values for the ***'External Non-volatile memory (NVM)'*** device parameters, please make a note of the part number of the memory device fitted on your target board and then contact Sigma technical support quoting the memory device part number. They should hopefully be able to tell you what values to use for the configuration.

3.2 Configuring the script file to program custom NVM (memory) device parameter values

If you need to program custom values for the ***'External Non-volatile memory (NVM)'*** device parameters, you will need to amend the ***'Sigma programming script'***.

Instructions:

- Start EQTools
- Select ***File – Open*** and then browse to and open the latest version of the Sigma script source file e.g. ***Sigma_ZW500_ProgCal_V17-14.ESW***
- The ***Script source file (*.esw)*** should now open in the Script Builder utility and the following script related tabs will be displayed.....

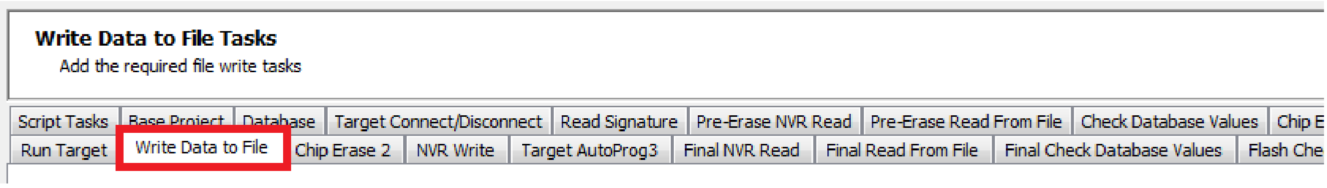


Script Builder... - C:\Equinox\Sigma\CalibrationSequence2\Sigma_ZW500_ProgCal_V17-14.ESW

Select Script Options

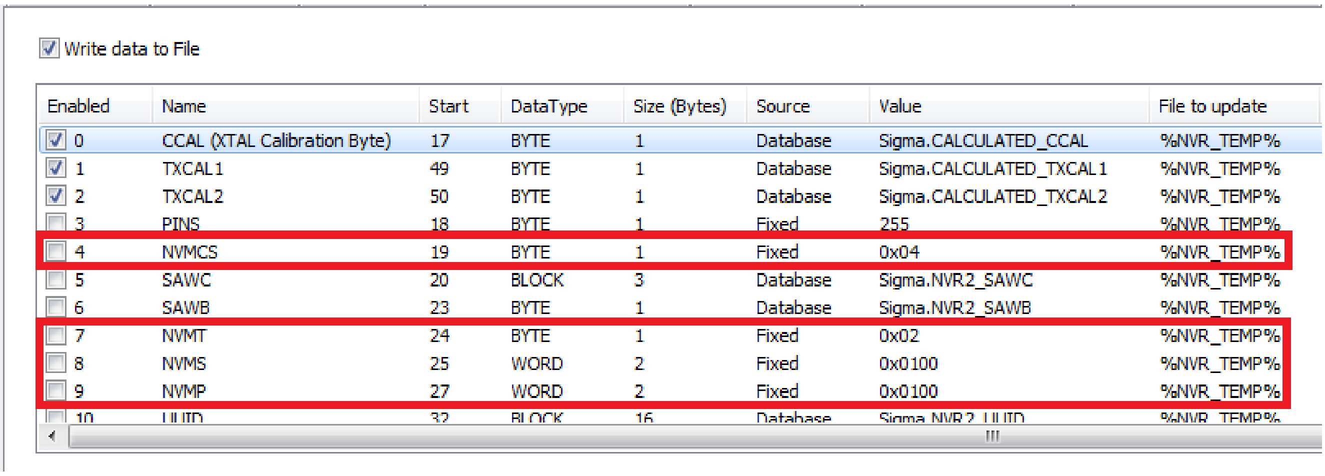
Select which scripting tasks to execute from the list below

- Select the ***'Write data to File'*** tab
- On this tab you should see a list of the available ***'NVR parameters'*** which can be configured – see screenshot below....



Write Data to File Tasks

Add the required file write tasks



☑ Write data to File

Enabled	Name	Start	DataType	Size (Bytes)	Source	Value	File to update
☑ 0	CCAL (XTAL Calibration Byte)	17	BYTE	1	Database	Sigma.CALCULATED_CCAL	%NVR_TEMP%
☑ 1	TXCAL1	49	BYTE	1	Database	Sigma.CALCULATED_TXCAL1	%NVR_TEMP%
☑ 2	TXCAL2	50	BYTE	1	Database	Sigma.CALCULATED_TXCAL2	%NVR_TEMP%
□ 3	PINS	18	BYTE	1	Fixed	255	%NVR_TEMP%
□ 4	NVMCS	19	BYTE	1	Fixed	0x04	%NVR_TEMP%
□ 5	SAWC	20	BLOCK	3	Database	Sigma.NVR2_SAWC	%NVR_TEMP%
□ 6	SAWB	23	BYTE	1	Database	Sigma.NVR2_SAWB	%NVR_TEMP%
□ 7	NVMT	24	BYTE	1	Fixed	0x02	%NVR_TEMP%
□ 8	NVMS	25	WORD	2	Fixed	0x0100	%NVR_TEMP%
□ 9	NVMP	27	WORD	2	Fixed	0x0100	%NVR_TEMP%
□ 10	UUID	32	BLOCK	16	Database	Sigma.NVR2_UUID	%NVR_TEMP%

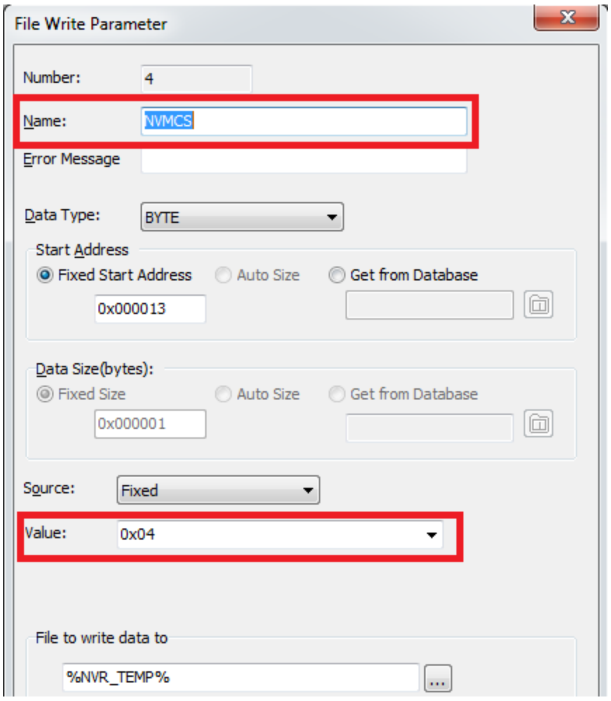
- The ***'NVR parameters'*** which are used to set up the Z-Wave external ***'Non-volatile memory (NVM)'*** are highlighted in the list in red: ***NVMCS, NVMT, NVMS, NVMP***
- By default, the programming script will simply use the value of each parameter (***NVMCS, NVMT, NVMS, NVMP***) which it read from the target device at the start of the script.
- If you wish to configure custom values for the external ***'Non-volatile memory (NVM)'*** parameters to match the hardware configuration of your target board, then please follow the instructions below....

3.3 Configuring individual NVM (memory) device parameters

The instructions in this section describe how to configure a custom fixed value for each ***'Non-volatile memory (NVM)'*** parameter. This will allow you to set up the programming script to program custom values into the following NVR parameters: ***NVMCS, NVMT, NVMS, NVMP.***

Instructions:

- Double-click the ***'NVMCS'*** parameter in the ***'NVR parameter'*** list
- The ***'File Write Parameter'*** screen for the ***'NVMCS'*** parameter should now be displayed....



File Write Parameter

Number: 4

Name: NVMCS

Error Message

Data Type: BYTE

Start Address

● Fixed Start Address ○ Auto Size ○ Get from Database

0x000013

Data Size(bytes):

● Fixed Size ○ Auto Size ○ Get from Database

0x000001

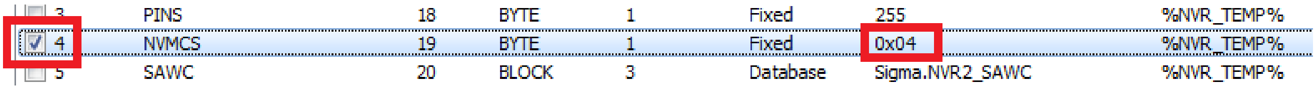
Source: Fixed

Value: 0x04

File to write data to

%NVR_TEMP%

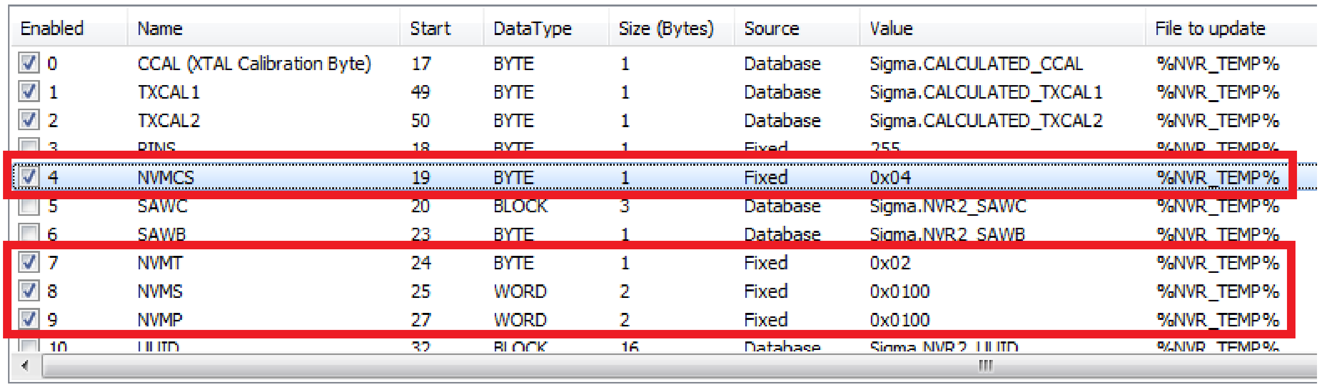
- Set the ***'Value'*** field to the value you wish to program into the ***'NVMCS'*** parameter. e.g. 0x04
- Check the settings match the screenshot above and leave all other settings unchanged!
- Click ***<OK>*** to save your amended settings.
- You will then be returned to the ***'NVR parameters'*** list and the ***'NVMCS'*** parameter should now have been automatically enabled and should show your amended settings. e.g. Value = 0x04.



☐ 3	PINS	18	BYTE	1	Fixed	255	%NVR_TEMP%
☑ 4	NVMCS	19	BYTE	1	Fixed	0x04	%NVR_TEMP%
☐ 5	SAWC	20	BLOCK	3	Database	Sigma.NVR2_SAWC	%NVR_TEMP%

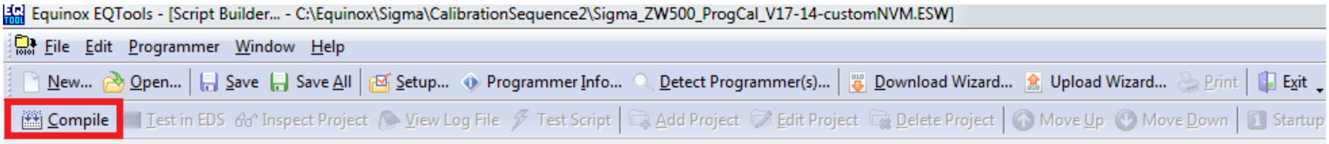
- Repeat the above procedure for each of the ***'Non-volatile memory (NVM)'*** parameters (***NVMCS, NVMT, NVMS, NVMP***)

- Once you have configured all the relevant parameters, the ***'NVR parameters'*** list should then show all these parameters as ***'Enabled'*** and with the correctly configured values....



Enabled	Name	Start	DataType	Size (Bytes)	Source	Value	File to update
☑ 0	CCAL (XTAL Calibration Byte)	17	BYTE	1	Database	Sigma.CALCULATED_CCAL	%NVR_TEMP%
☑ 1	TXCAL1	49	BYTE	1	Database	Sigma.CALCULATED_TXCAL1	%NVR_TEMP%
☑ 2	TXCAL2	50	BYTE	1	Database	Sigma.CALCULATED_TXCAL2	%NVR_TEMP%
□ 3	PINS	18	BYTE	1	Fixed	255	%NVR_TEMP%
☑ 4	NVMCS	19	BYTE	1	Fixed	0x04	%NVR_TEMP%
□ 5	SAWC	20	BLOCK	3	Database	Sigma.NVR2_SAWC	%NVR_TEMP%
□ 6	SAWB	23	BYTE	1	Database	Sigma.NVR2_SAWB	%NVR_TEMP%
☑ 7	NVMT	24	BYTE	1	Fixed	0x02	%NVR_TEMP%
☑ 8	NVMS	25	WORD	2	Fixed	0x0100	%NVR_TEMP%
☑ 9	NVMP	27	WORD	2	Fixed	0x0100	%NVR_TEMP%
□ 10	UUID	32	BLOCK	16	Database	Sigma.NVR2_UUID	%NVR_TEMP%

- The script is now configured to automatically program your custom values for the ***'Non-volatile memory (NVM)'*** parameters into the ***'NVR Area'*** of the target Z-Wave device.
- Click the ***'Compile'*** icon on the top EQTools icon bar to generate the amended ***script file (*.esf)***.



Equinox EQTools - [Script Builder... - C:\Equinox\Sigma\CalibrationSequence2\Sigma_ZW500_ProgCal_V17-14-customNVM.ESW]

- This ***script file (*.esf)*** can now be executed within the ISP-PRO production utility.

3.4 Testing the custom NVM (memory) device parameters

To test whether your custom ***'NVM parameter values'*** have been programmed correctly, you will need to follow the instructions below....

- Execute your customised ***'programming script (*.esf)'*** in the ISP-PRO utility
- Program a virgin Z-Wave target board using your customised ***'programming script (*.esf)'***
- Exit the ISP-PRO utility
- Start the EQTools software utility
- Open or create a new ***'EDS – Development project'***
- Select the ***'NVR'*** tab
- Click the ***'Read'*** button → the current values of the ***'NVR Area'*** will be transferred to the ***'Buffer window'***
- Check that the read back values for the ***'NVM parameter values'*** match the values you declared in your programming script.
- The final test that you have programmed the correct values is to try running Sigma's own firmware on the Z-Wave device and check that it executes correctly. This is not a comprehensive test as some Sigma firmware does not actually require or use the external NVM device!

Appendix 4 - ISP-PRO - Quick Start Guide

1.0 Overview

This section offers an overview / quick-start guide to running a ***'Sigma calibration / programming'*** script with the ISP-PRO application.

ISP-PRO executes ***'Programming Scripts'*** in order to control a target Equinox Programmer. These scripts are created using the ***EQTools – Script Builder*** utility and can be tested / debugged using the ***EQTools – Script Debugger*** utility. Once the scripts have been fully tested using EQTools, they are then ready for executing within the ISP-PRO application.

This section details how to take the files from EQTools and install / execute them within ISP-PRO.

2.0 Installing the Sigma scripts and projects

2.1 Overview

The ***'Sigma calibration / programming'*** scripts and projects are supplied in a single zip file by Equinox.

Instructions:

- Copy the zip file to your PC hard disk
- Unzip the files to a suitable folder on your PC hard disk e.g. ***c:\Equinox***

Important note:
The zip file contains all the 'development' source files which were used to make the projects and scripts. These files are not required for ISP-PRO.

The only files which ISP-PRO actually requires to run the script in ***'production mode'*** are as follows:

- *.esf - Script File(s)
- *.PPC - Project Collection File(s)
- *.prj - Compiled Project File(s) - only the 'Base Project' is required

All other files are only required for maintaining the scripts / projects by the developer or production supervisor and therefore do not have to be copied to the 'production PC'.

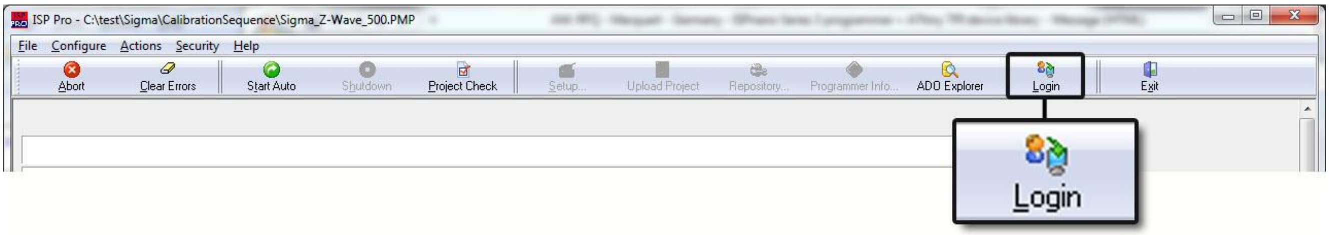
2.2 Setting up ISP-PRO to run the Programming Script(s)

In order to execute the ***'programming scripts'***, it is necessary to install them into your ISP-PRO ***'Scripts folder'*** and also to set up ISP-PRO so that it knows which script(s) to execute.

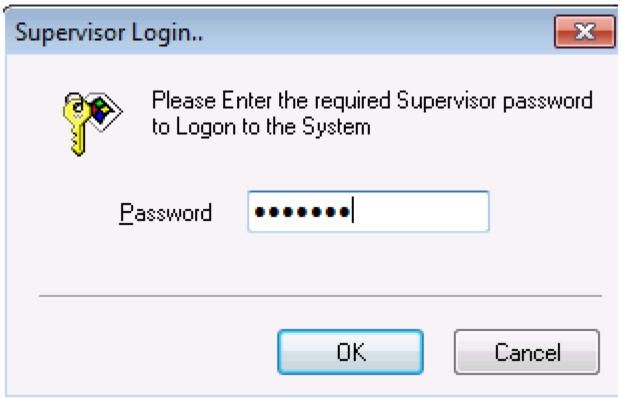
Please follow the steps below before attempting to execute a ***'programming script'***:

2.3 Start the ISP-PRO application and log in

- Select ***<Start><Programs><Equinox><ISPPRO>*** → ISP-PRO application should start up..
- Click the ***<Login>*** button



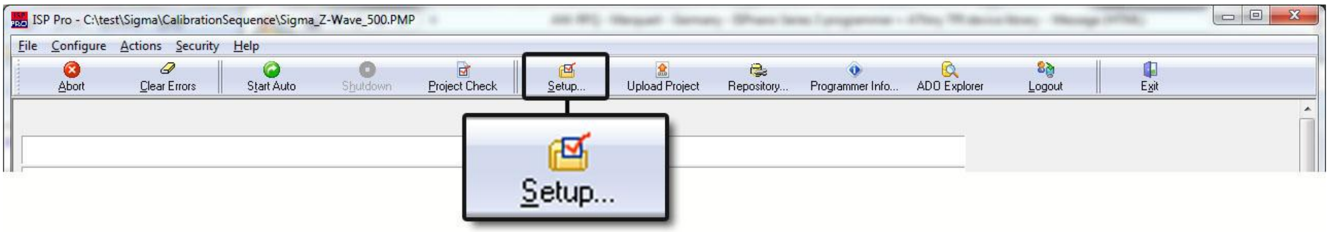
- You will now be asked to enter your password.



Please Enter the required Supervisor password to Logon to the System

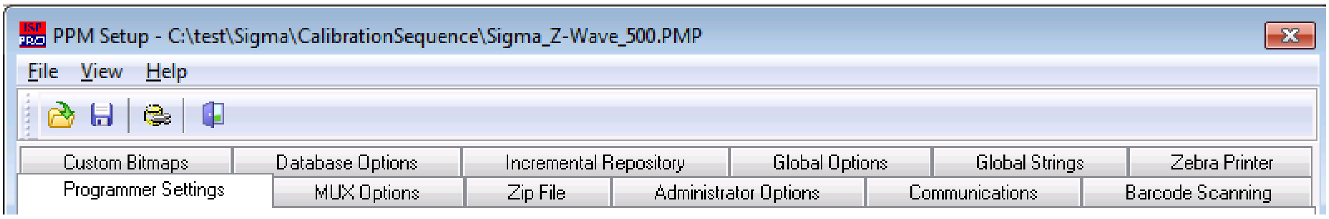
Password •••••••

- Type in your password (default password is: ***equinox***)
- The ***<Setup>*** icon should now be selectable.



- Click the ***<Setup>*** icon

--> The ***'PPM Setup'*** screen will now be displayed with the ***'Programmer settings'*** tab selected....

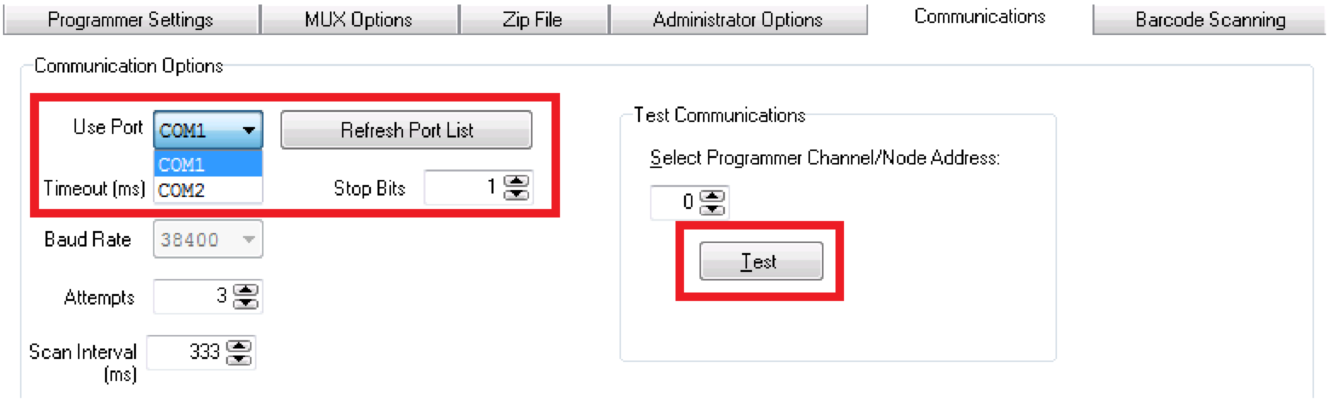


2.4 Setting up the correct COM port

If this is the first time that you have used ISP-PRO, then it is likely that the COM port needs to be configured.

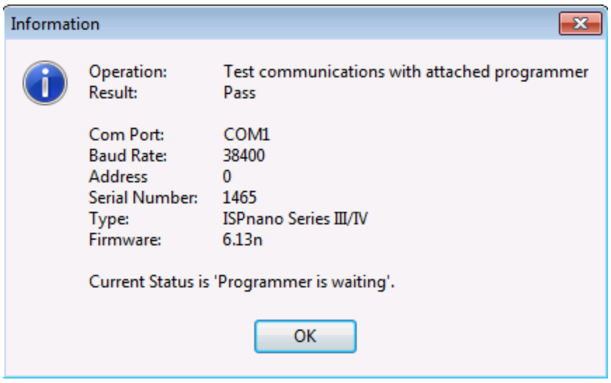
Instructions:

- Click the ***<Setup>*** icon
- Select the ***'Communications'*** tab



Use Port: COM1 (COM1, COM2)
Timeout (ms)
Stop Bits: 1
Baud Rate: 38400
Attempts: 3
Scan Interval (ms): 333
Select Programmer Channel/Node Address: 0

- Click the ***<Refresh Port List>*** button
- A list of the available ***'COM ports'*** on your PC should now be displayed.
- Select the ***COM port*** which the programmer is attached to
- Click the ***<Test>*** button
- ISP-PRO will attempt to communicate with the programmer and should display a message to tell you that it has found the attached programmer.



Operation: Test communications with attached programmer
Result: Pass

Com Port: COM1
Baud Rate: 38400
Address: 0
Serial Number: 1465
Type: ISPnano Series III/IV
Firmware: 6.13n

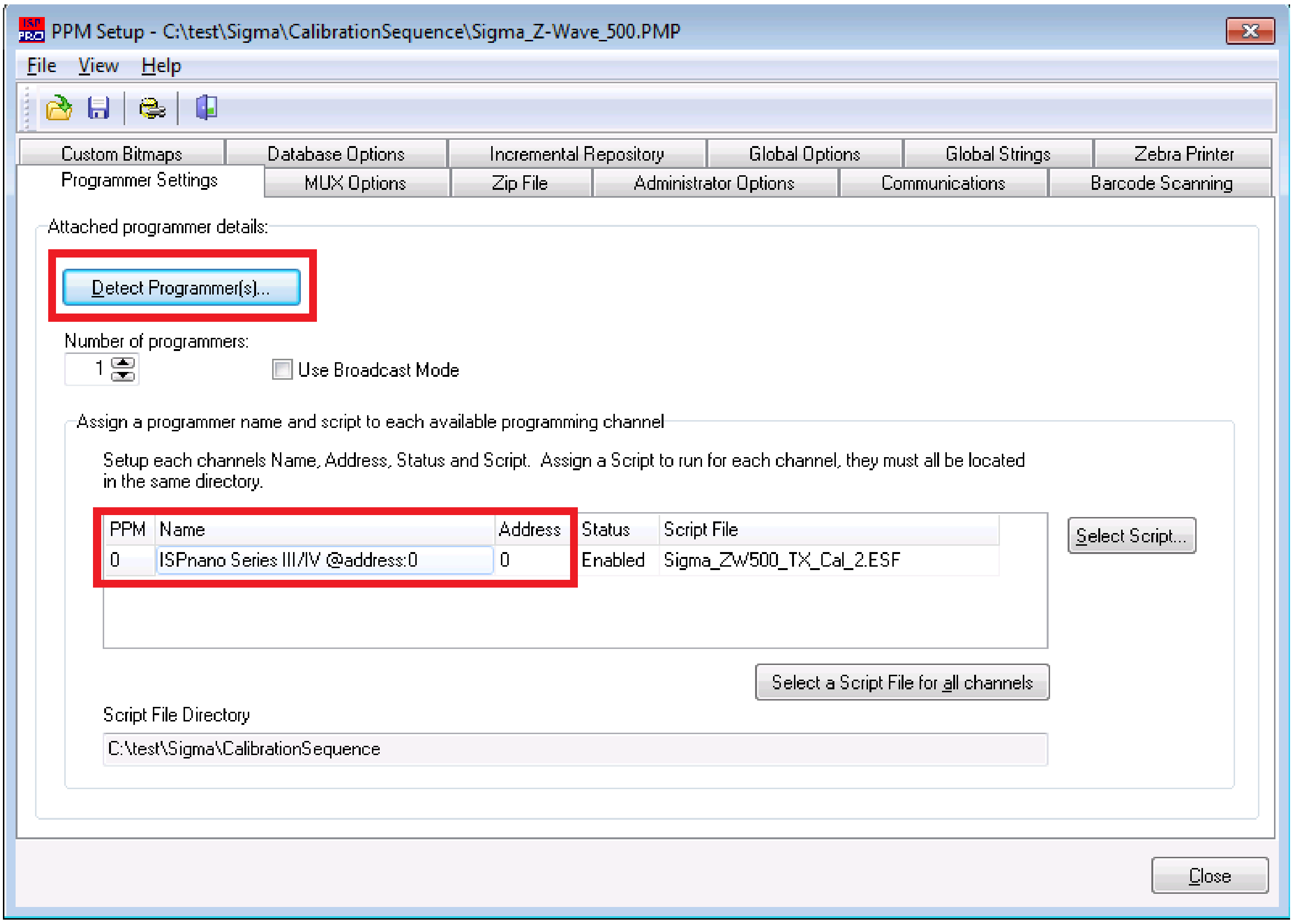
Current Status is 'Programmer is waiting'.

2.5 Detecting the attached programmer(s)

The very first time you run ISP-PRO, it is necessary to detect the attached programmer(s) and also assign ***'programmer names'*** to each programmer.

Please follow the instructions below...

- Click the ***<Setup>*** icon
- Select the ***<Programmer Settings>*** tab



Number of programmers: 1

☐ Use Broadcast Mode

Assign a programmer name and script to each available programming channel

Setup each channels Name, Address, Status and Script. Assign a Script to run for each channel, they must all be located in the same directory.

PPM	Name	Address	Status	Script File
0	ISPnano Series III/IV @address:0	0	Enabled	Sigma_ZW500_TX_Cal_2.ESF

Script File Directory

C:\test\Sigma\CalibrationSequence

- If you have not already detected the attached programmer(s), then click the ***'Detect Programmer(s)'*** button.

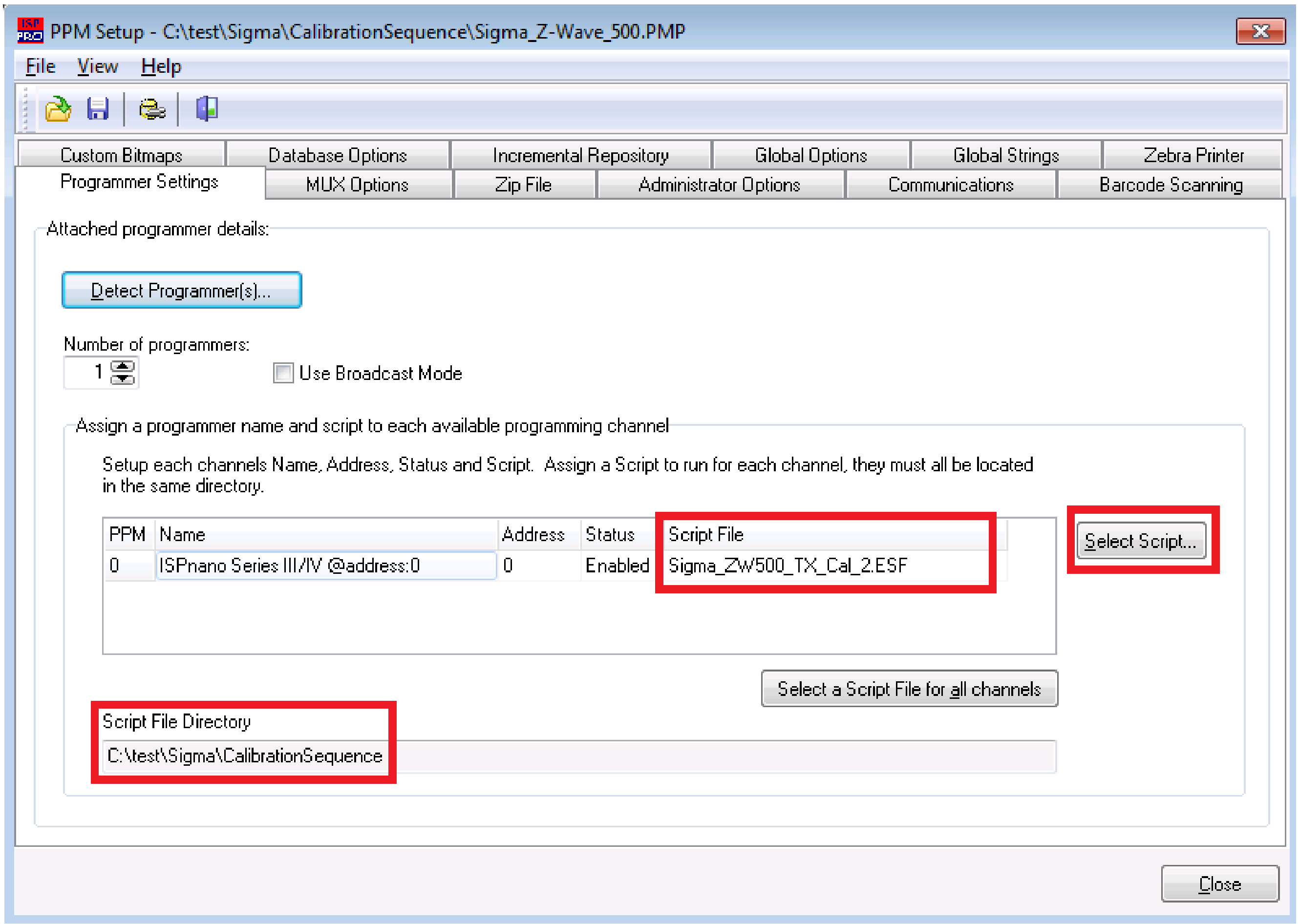
---> The detection process will provide a list of all attached programmers....in this case it has detected a single programmer at node address 0.

- Each attached programmer is automatically given a ***'programmer name'*** e.g. ***'ISPnano Series IIIIV @address:0'***.

2.6 Selecting a Script File to run

To select the required ***'Script File'***, follow the instructions below...

- Click the ***<Setup>*** icon
- Select the ***<Programmer Settings>*** tab

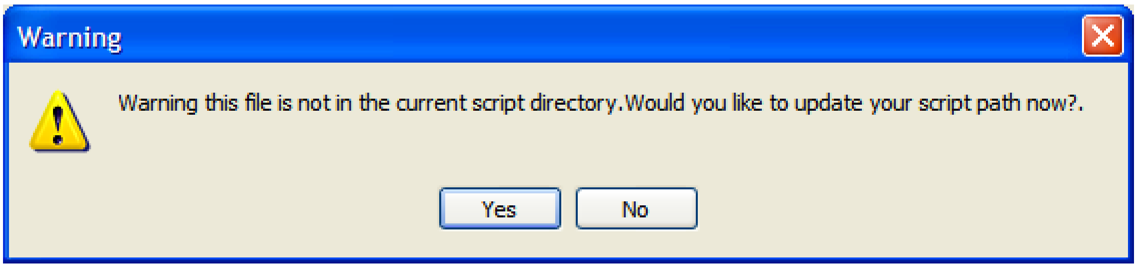


Setup each channels Name, Address, Status and Script. Assign a Script to run for each channel, they must all be located in the same directory.

PPM	Name	Address	Status	Script File
0	ISPnano Series III/IV @address:0	0	Enabled	Sigma_ZW500_TX_Cal_2.ESF

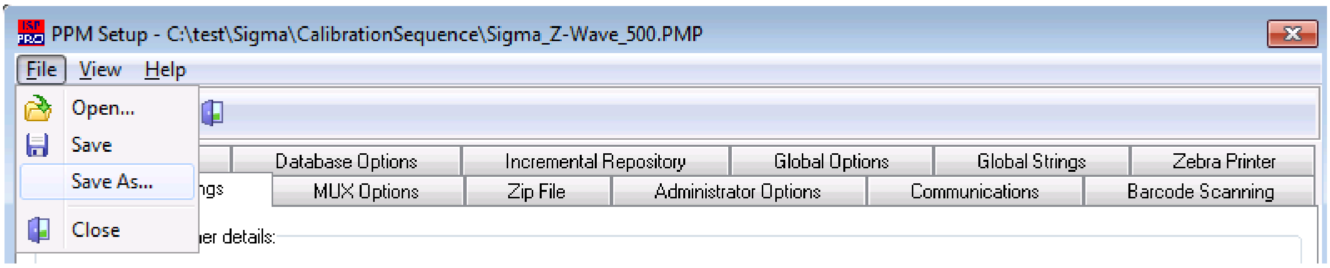
Script File Directory: C:\test\Sigma\CalibrationSequence

- If you wish to select a ***Script File (*.esf)*** for a single channel, click the ***<Select Script file>*** button.
- If you wish to select a ***Script File (*.esf)*** for multiple channels, click the ***<Select a Script file for all channels>*** button.
- Browse to and select the required ***'Script File'*** which will have the file extension ****.esf***. The ***'Script File'*** should be located in your ***'Scripts'*** directory / folder.
- If you selected ***'Script File'*** is not in the ***'Script File directory'*** then you will receive the following warning:

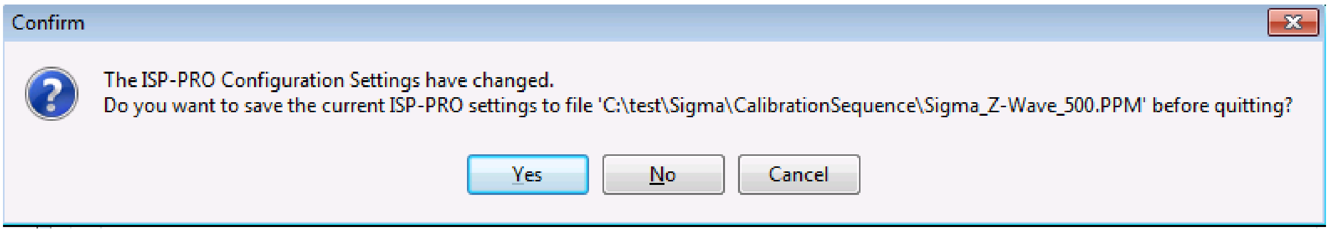


Warning this file is not in the current script directory.Would you like to update your script path now?.

- If you click **<*Yes*>** then ISP-PRO will automatically set the ***'Script File directory'*** to the directory where your selected script is located.
- Once you've selected your ***'Script File'***, then you just need to save your settings by selecting ***'File - Save As...'*** and then specifying a file name. It's a good idea to put this ***'ISP-PRO Settings'*** file in your ***'Script File Directory'***.

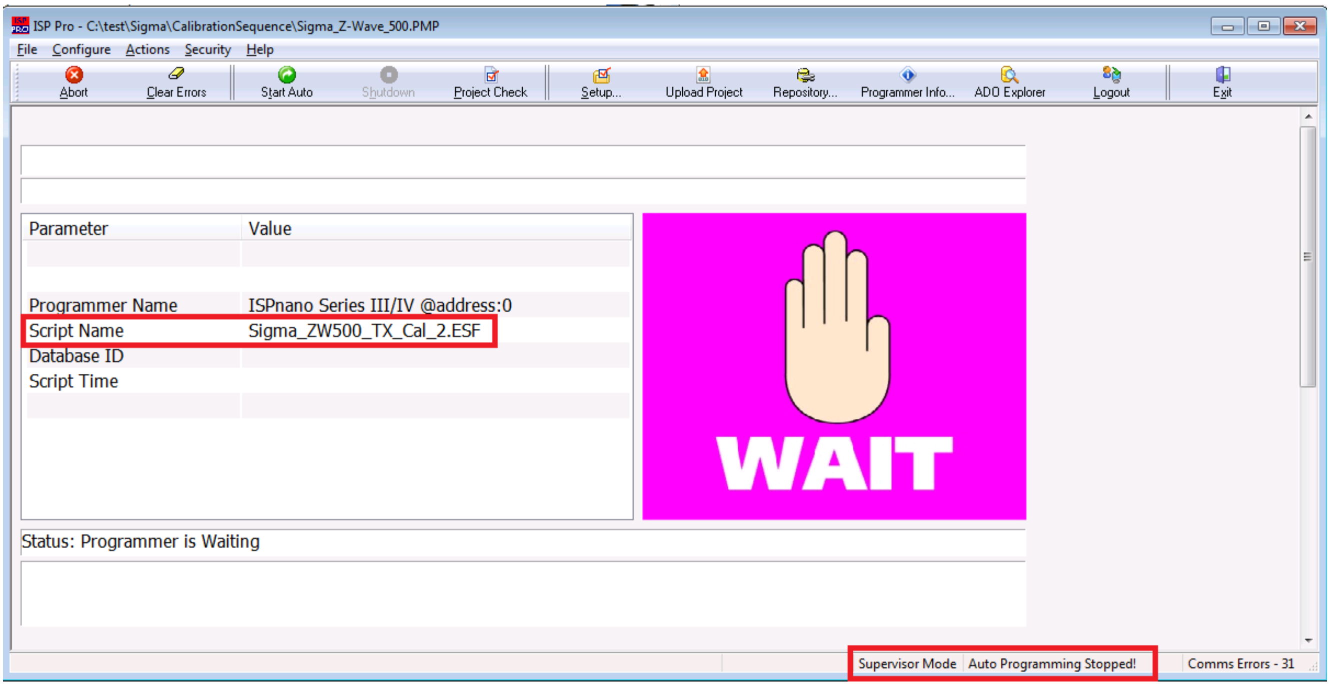


- You should then see a ***'Confirm'*** screen similar to this one.....



The ISP-PRO Configuration Settings have changed.
Do you want to save the current ISP-PRO settings to file 'C:\test\Sigma\CalibrationSequence\Sigma_Z-Wave_500.PPM' before quitting?

- You will then be returned to the main ISP-PRO screen where you should see that your selected ***'Script File'*** is now displayed.



Parameter	Value
Programmer Name	ISPnano Series III/IV @address:0
Script Name	Sigma_ZW500_TX_Cal_2.ESF
Database ID	
Script Time	

Status: Programmer is Waiting

Supervisor Mode | Auto Programming Stopped! | Comms Errors - 31

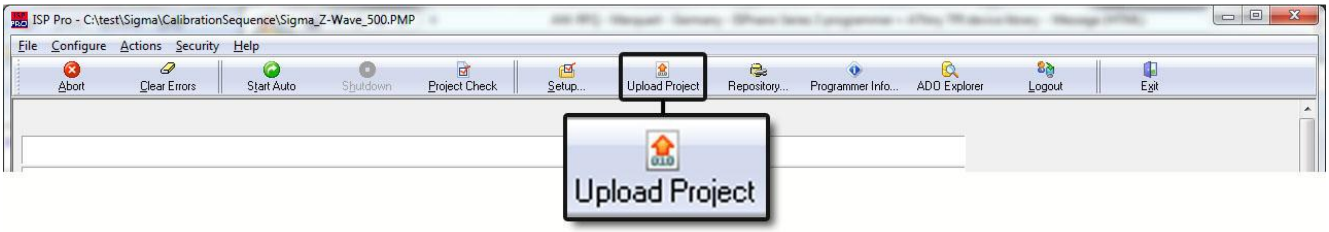
- ISP-PRO is still in ***'Supervisor Mode'*** allowing you to change any other settings before going into ***'Production mode'***.

2.7 Uploading your Project Collection to the programmer(s)

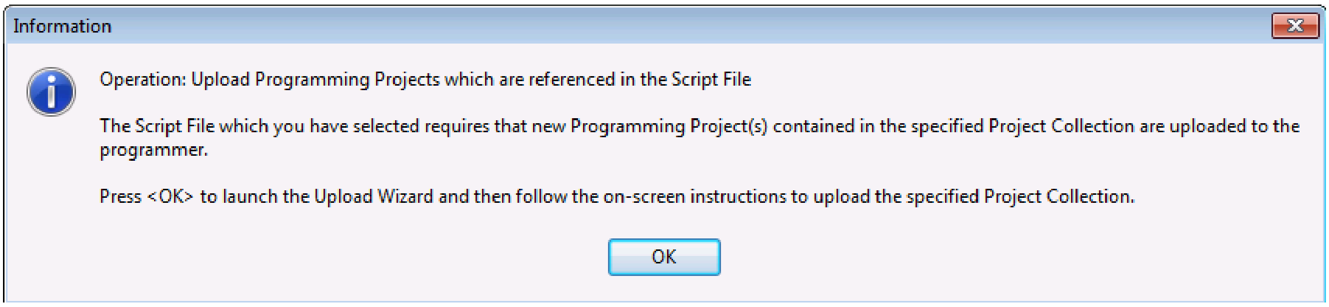
Before running any ***'Script File'*** which uses ***'Standalone programming projects'***, it is necessary to upload these projects to the programmer ***'FLASH Memory Store'***.

To upload a ***'Project Collection'*** to the programmer, follow the instructions below...

- Click the ***<Upload Project>*** button on the ISP-PRO icon bar.



- If you have already selected the script file to execute, then ISP-PRO should automatically notify you of which ***Project Collection (*.ppc)*** file to upload.



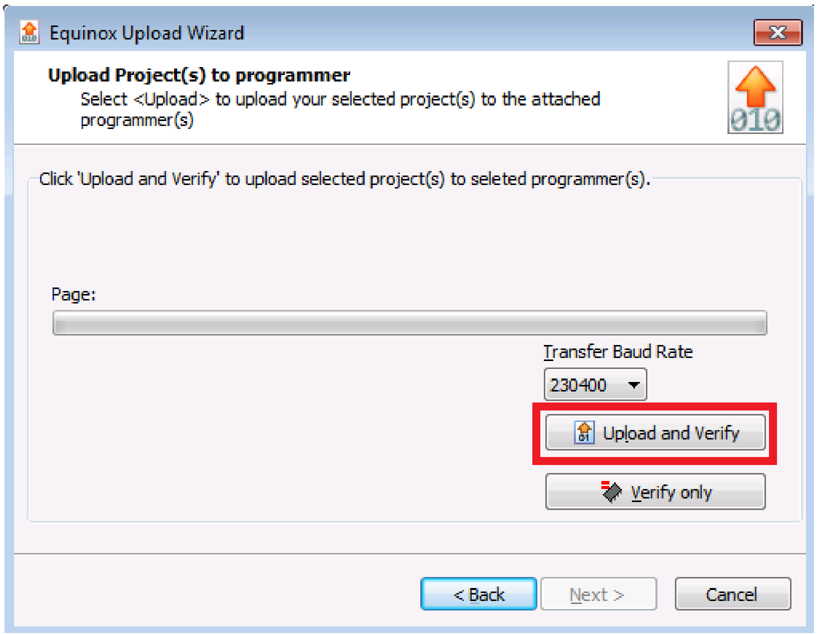
Information

Operation: Upload Programming Projects which are referenced in the Script File

The Script File which you have selected requires that new Programming Project(s) contained in the specified Project Collection are uploaded to the programmer.

Press <OK> to launch the Upload Wizard and then follow the on-screen instructions to upload the specified Project Collection.

- If ISP-PRO detects the attached programmer(s), then the ***'Upload Wizard'*** utility will be automatically started and the following screen will be displayed....



Equinox Upload Wizard

Upload Project(s) to programmer

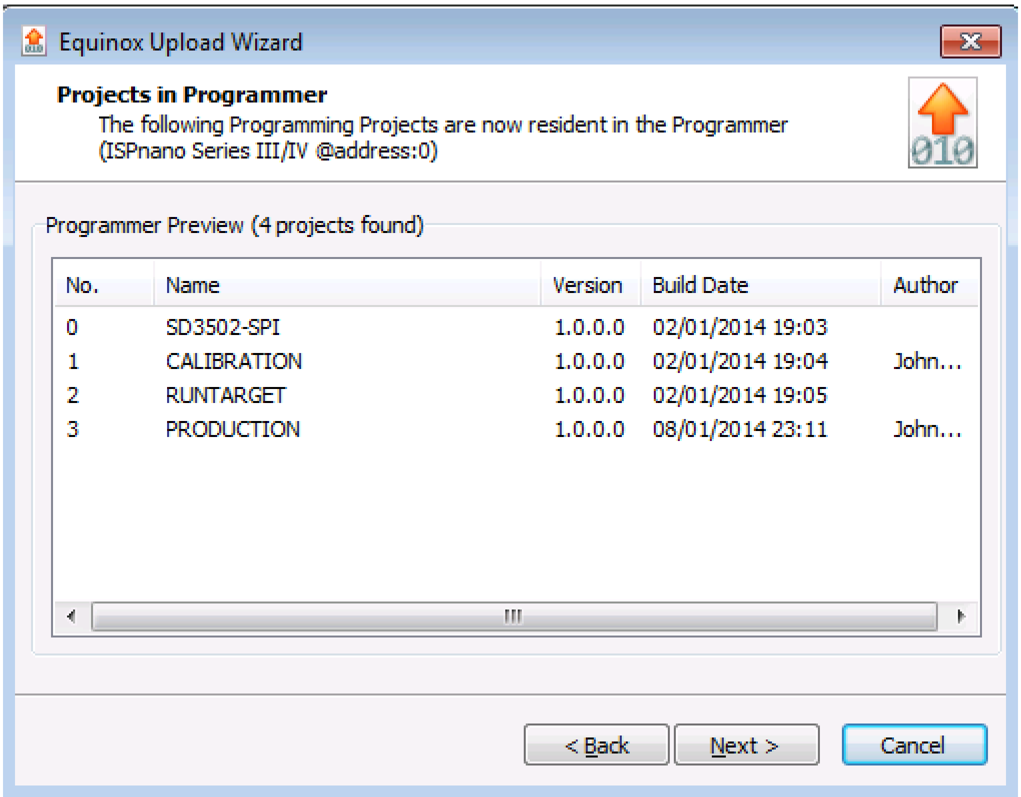
Select <Upload> to upload your selected project(s) to the attached programmer(s)

Click 'Upload and Verify' to upload selected project(s) to seleted programmer(s).

Page:

Transfer Baud Rate: 230400

- Click the ***<Upload and Verify>*** button to upload the ***Project Collection (*.ppc)*** to the attached programmers.
- Once the upload of the projects is complete, the ***'Upload Wizard'*** will display a list of the uploaded projects....



Projects in Programmer

The following Programming Projects are now resident in the Programmer (ISPnano Series III/IV @address:0)

Programmer Preview (4 projects found)

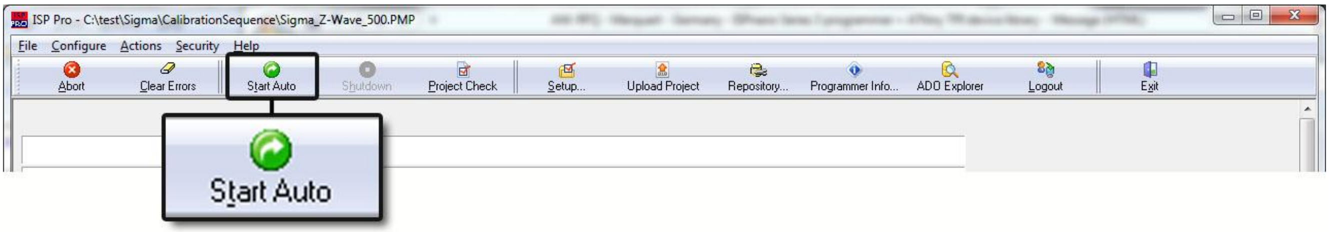
No.	Name	Version	Build Date	Author
0	SD3502-SPI	1.0.0.0	02/01/2014 19:03	
1	CALIBRATION	1.0.0.0	02/01/2014 19:04	John...
2	RUNTARGET	1.0.0.0	02/01/2014 19:05	
3	PRODUCTION	1.0.0.0	08/01/2014 23:11	John...

- Click the ***<Next>*** button on this screen and then the ***<Finish>*** button on the next screen to complete the project upload process.
- You will then be taken back to the main ISP-PRO screen.
- ISP-PRO is still in ***'Supervisor mode'*** with the programming network stopped.

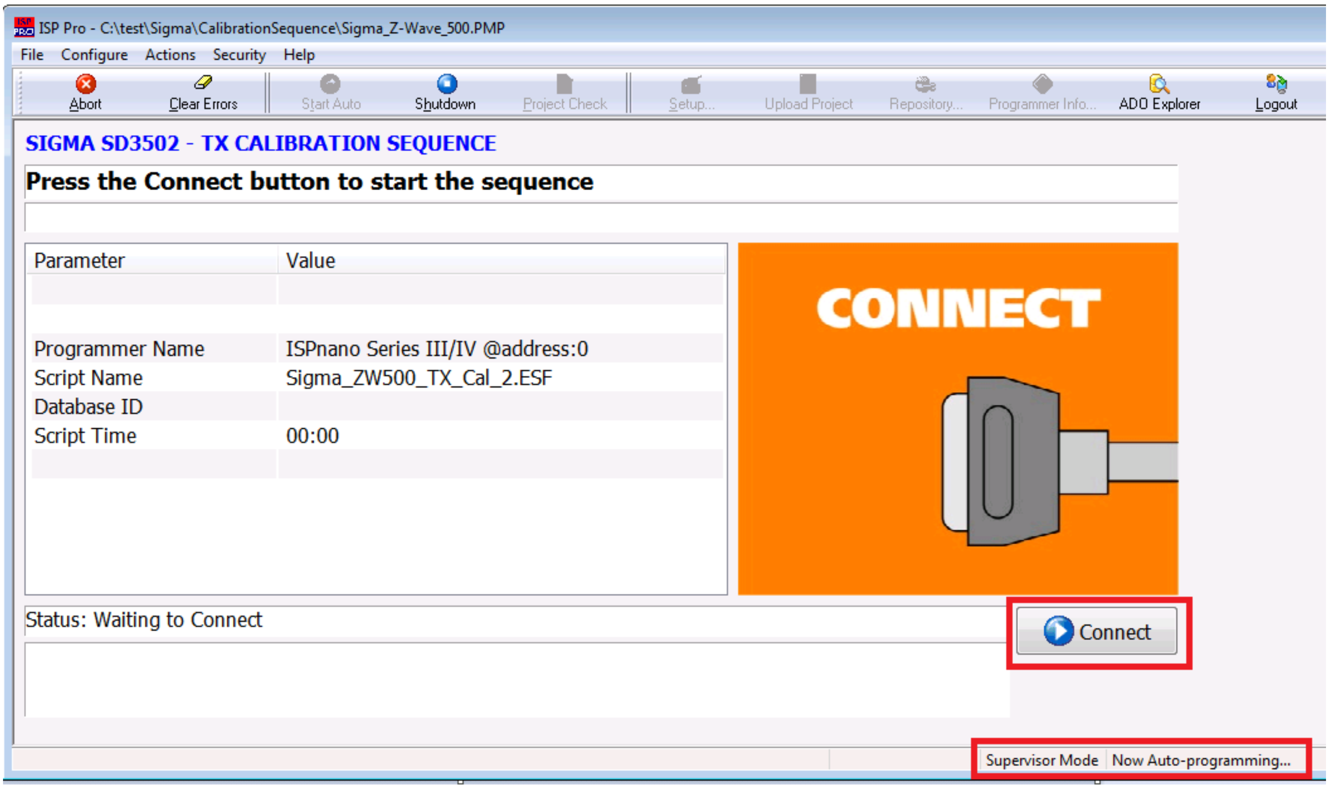
2.8 Running the programming Script(s)

To execute your selected ***Programming Script(s),*** please follow the steps details below:

1. Click the **<Start Auto>** icon on the ISP-PRO Icon Bar



→ All enabled programming channels should now go to the ***<Connect>*** state



SIGMA SD3502 - TX CALIBRATION SEQUENCE

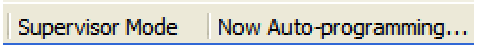
Press the Connect button to start the sequence

Parameter	Value
Programmer Name	ISPnano Series III/IV @address:0
Script Name	Sigma_ZW500_TX_Cal_2.ESF
Database ID	
Script Time	00:00

Status: Waiting to Connect

Supervisor Mode | Now Auto-programming...

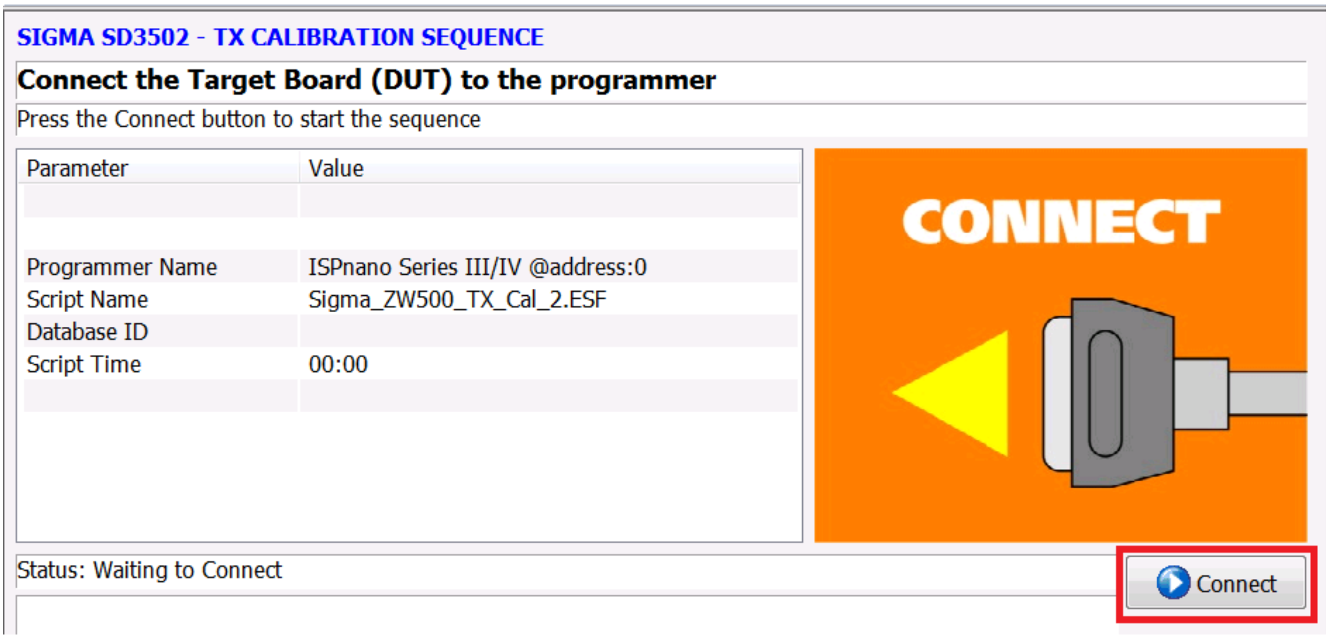
The message ***'Now Autoprogramming'*** should be displayed at the bottom right-hand corner of the ISP-PRO window.


Supervisor Mode | Now Auto-programming...

2.9 Executing the programming / calibration sequence

To execute the programming / calibration sequence on a Target System (DUT)....

1. Connect a Target System (DUT) to the relevant programming channel
2. Click the ***<Connect>*** button on the bottom-right of the channel icon to commence the programming operation on the selected channel.



SIGMA SD3502 - TX CALIBRATION SEQUENCE

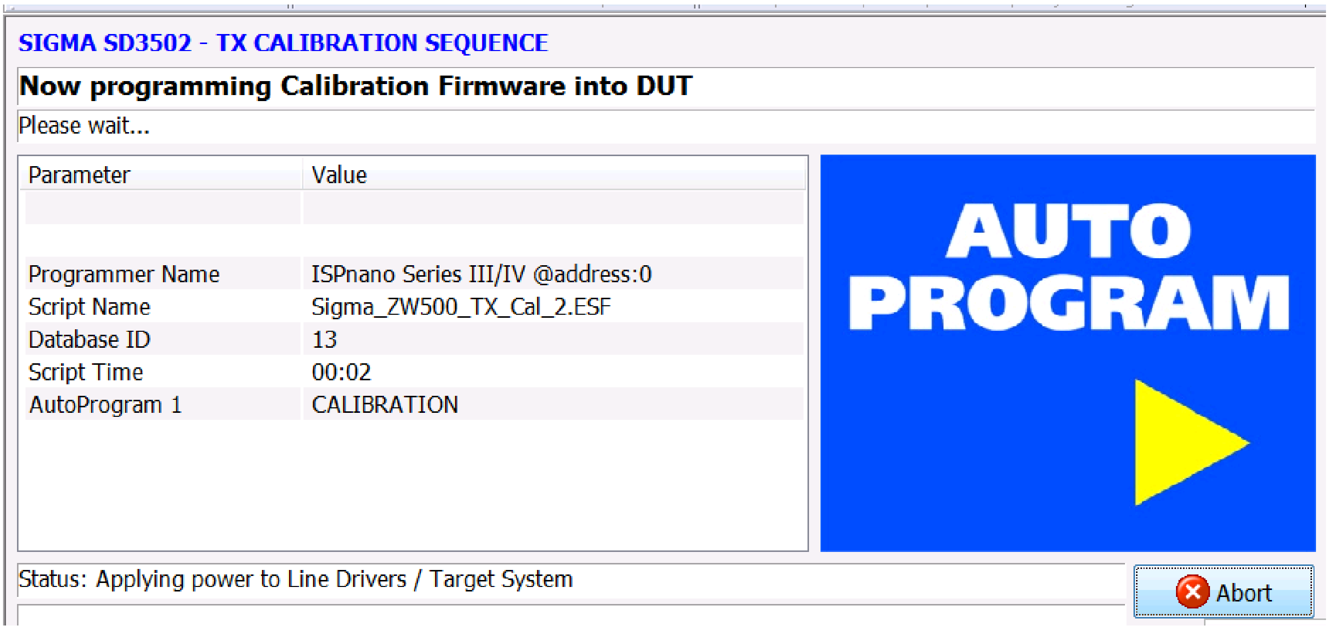
Connect the Target Board (DUT) to the programmer

Press the Connect button to start the sequence

Parameter	Value
Programmer Name	ISPnano Series III/IV @address:0
Script Name	Sigma_ZW500_TX_Cal_2.ESF
Database ID	
Script Time	00:00

Status: Waiting to Connect

2. The script will then start to execute....



SIGMA SD3502 - TX CALIBRATION SEQUENCE

Now programming Calibration Firmware into DUT

Please wait...

Parameter	Value
Programmer Name	ISPnano Series III/IV @address:0
Script Name	Sigma_ZW500_TX_Cal_2.ESF
Database ID	13
Script Time	00:02
AutoProgram 1	CALIBRATION

Status: Applying power to Line Drivers / Target System

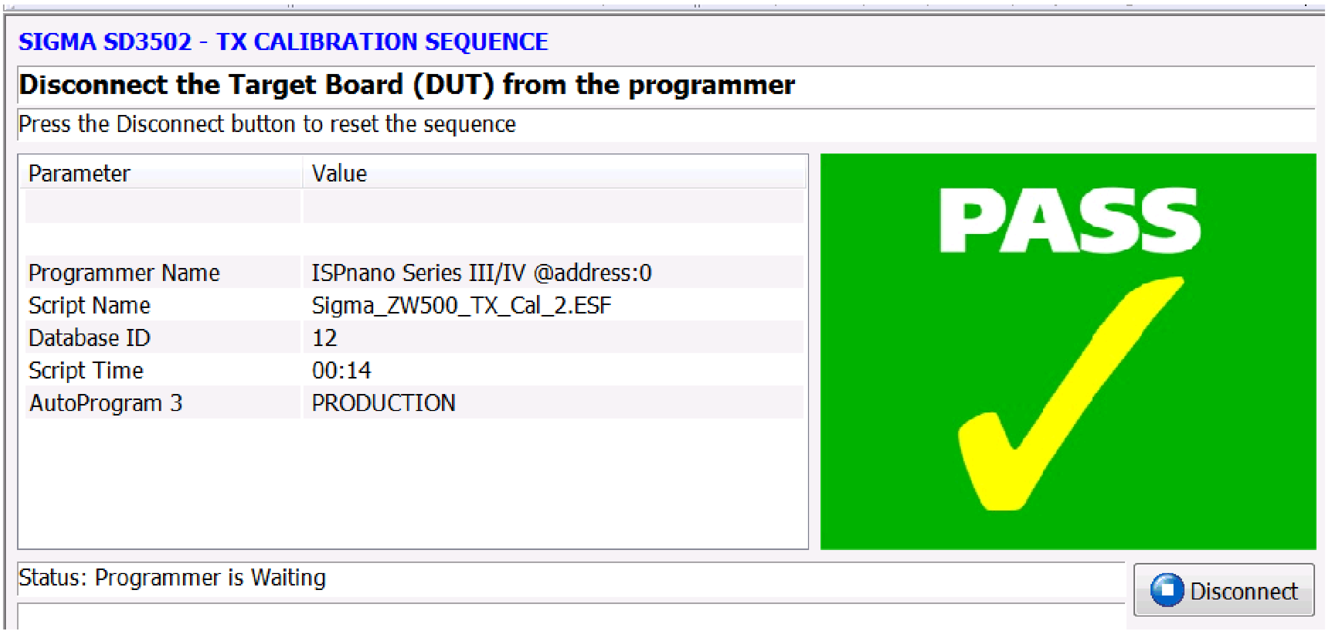
--> The icon will display ***'Auto Program'***
--> The ***'Script Timer'*** will now start timing the execution of the script.

Important note:

- You can abort the execution of the script at any time by pressing the ***'Abort'*** button.
- However, this will leave the ***'NVR area'*** blank so the Target IC is now scrap !!!

2.10 Programming sequence - PASS

If the programming sequence is successful (executes without any errors), then the following screen will be displayed.....



SIGMA SD3502 - TX CALIBRATION SEQUENCE

Disconnect the Target Board (DUT) from the programmer

Press the Disconnect button to reset the sequence

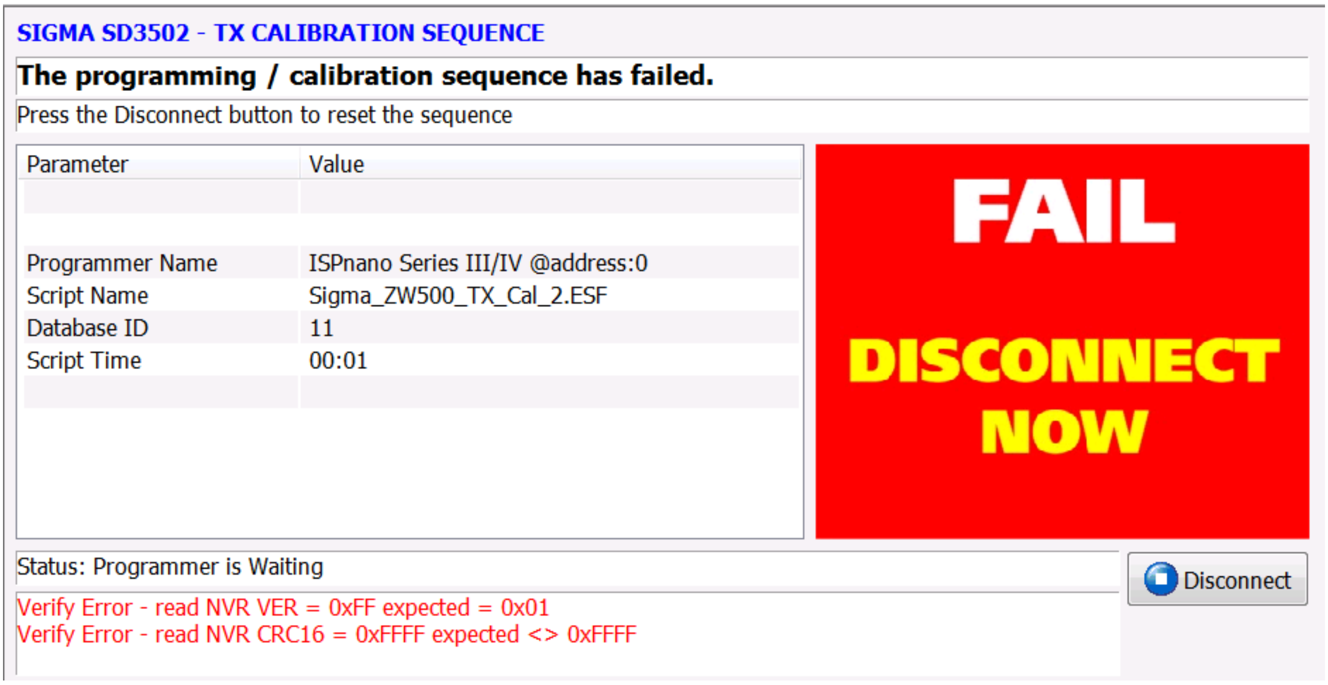
Parameter	Value
Programmer Name	ISPnano Series III/IV @address:0
Script Name	Sigma_ZW500_TX_Cal_2.ESF
Database ID	12
Script Time	00:14
AutoProgram 3	PRODUCTION

Status: Programmer is Waiting

- ISP-PRO will display ***'PASS - Disconnect now'***
- The Target Board (DUT) can now be disconnected from the programmer.
- Press the ***'Disconnect'*** button to reset the sequence ready for the next Target Board (DUT)

2.11 Programming sequence - FAIL

If the programming or calibration fails for any reason, then the following screen will be displayed.....



SIGMA SD3502 - TX CALIBRATION SEQUENCE

The programming / calibration sequence has failed.

Press the Disconnect button to reset the sequence

Parameter	Value
Programmer Name	ISPnano Series III/IV @address:0
Script Name	Sigma_ZW500_TX_Cal_2.ESF
Database ID	11
Script Time	00:01

FAIL

DISCONNECT NOW

Status: Programmer is Waiting

```
Verify Error - read NVR VER = 0xFF expected = 0x01
Verify Error - read NVR CRC16 = 0xFFFF expected <> 0xFFFF
```

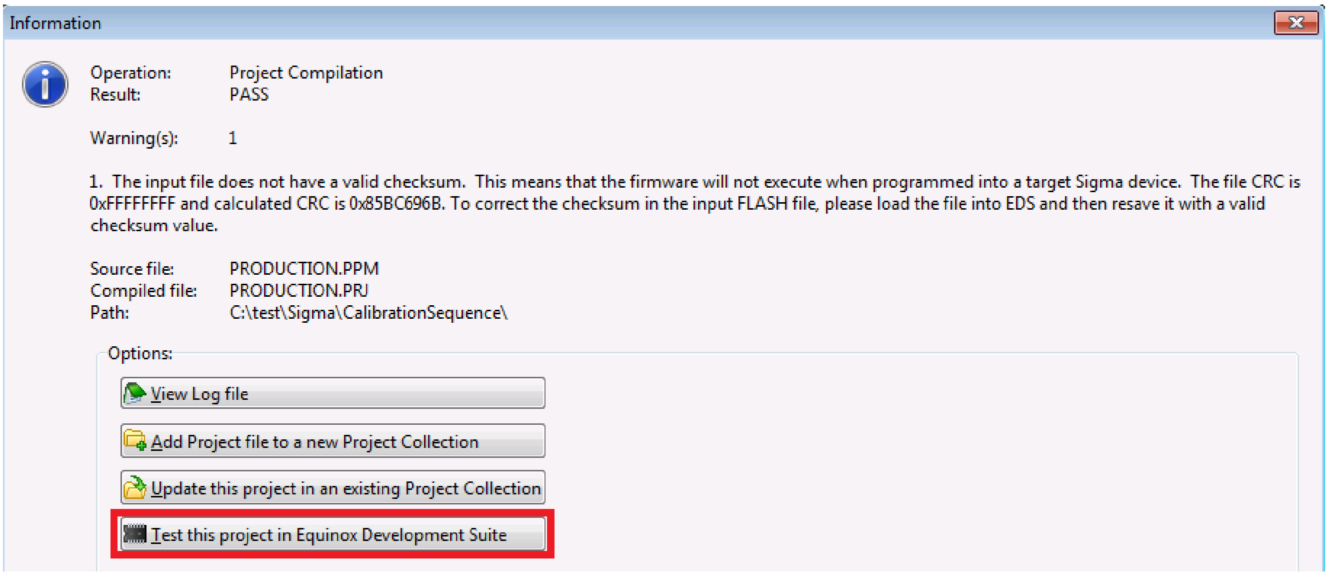
- ISP-PRO will display ***'PASS - Disconnect now'***
- The Target Board (DUT) can now be disconnected from the programmer.
- Press the ***'Disconnect'*** button to reset the sequence ready for the next Target Board (DUT)

Important note:
If the programming fails for any reason, the ***'NVR Area'*** will be left blank (all 0xFF).
This means that the Target IC / Module can never be re-programmed again because the ***'Factory NVR calibration data'*** has now been permanently erased.

Appendix 5 - Sigma CRC32 FLASH Checksum

1.0 Overview

This Z-Wave 500 series devices feature a 32-bit (4 byte) checksum which is used by both the programmer and the Sigma device to validate the FLASH contents are not corrupt.



Information

Operation: Project Compilation
Result: PASS

Warning(s): 1

1. The input file does not have a valid checksum. This means that the firmware will not execute when programmed into a target Sigma device. The file CRC is 0xFFFFFFFF and calculated CRC is 0x85BC696B. To correct the checksum in the input FLASH file, please load the file into EDS and then resave it with a valid checksum value.

Source file: PRODUCTION.PPM
Compiled file: PRODUCTION.PRJ
Path: C:\test\Sigma\CalibrationSequence\

Options:
View Log file
Add Project file to a new Project Collection
Update this project in an existing Project Collection
Test this project in Equinox Development Suite

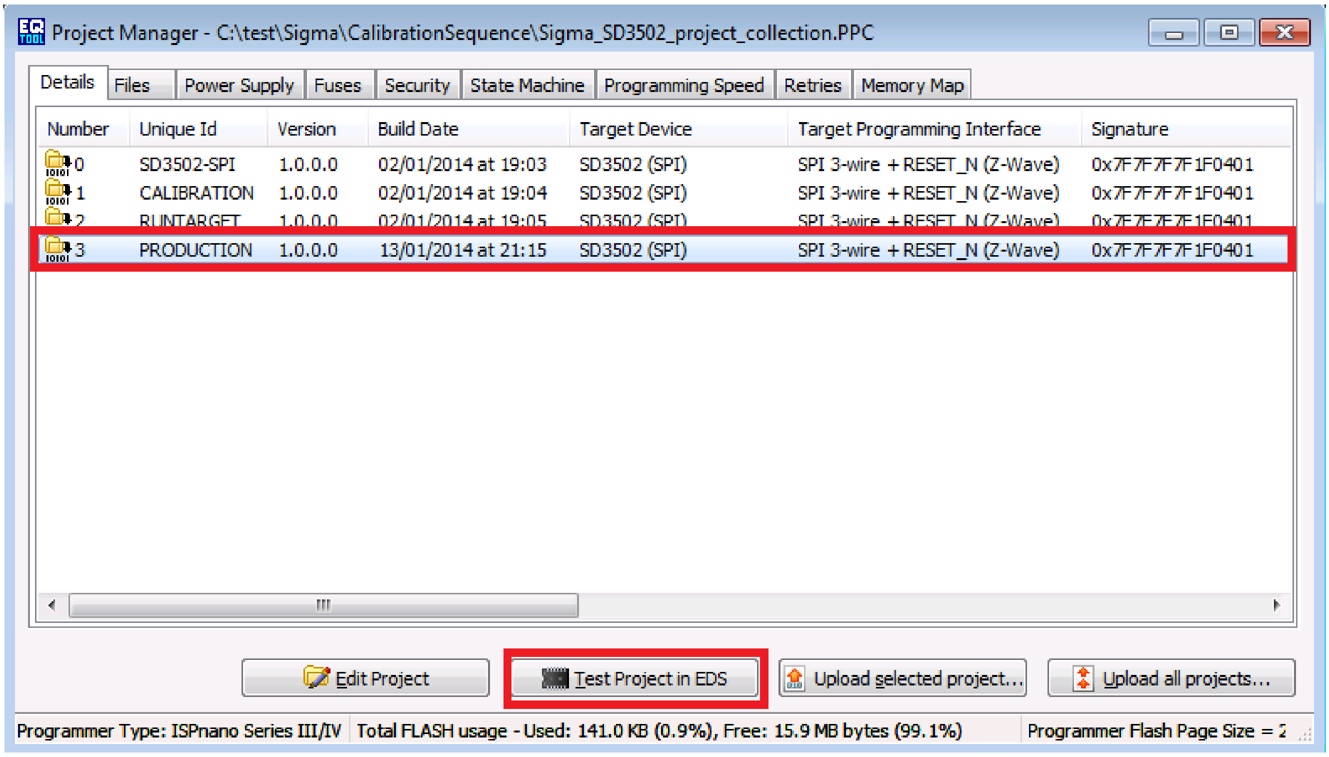
1.2 How to correct / add a valid CRC32 checksum

If your input file does not have a valid ***'CRC32 Checksum'***, then the file cannot be used for programming within EQTools. It is necessary to generate a checksum for the file and then re-save the file with this checksum before the file can be used with EQTools to program a Z-Wave device.

1.3 Opening the project in EDS (Development mode)

A simple way to open the project in EDS (Development mode) is as follows:

- Open the Project Collection
- Click the project you want to test in EDS once so that it is highlighted
- Now click the ***'Check project in EDS mode'*** button

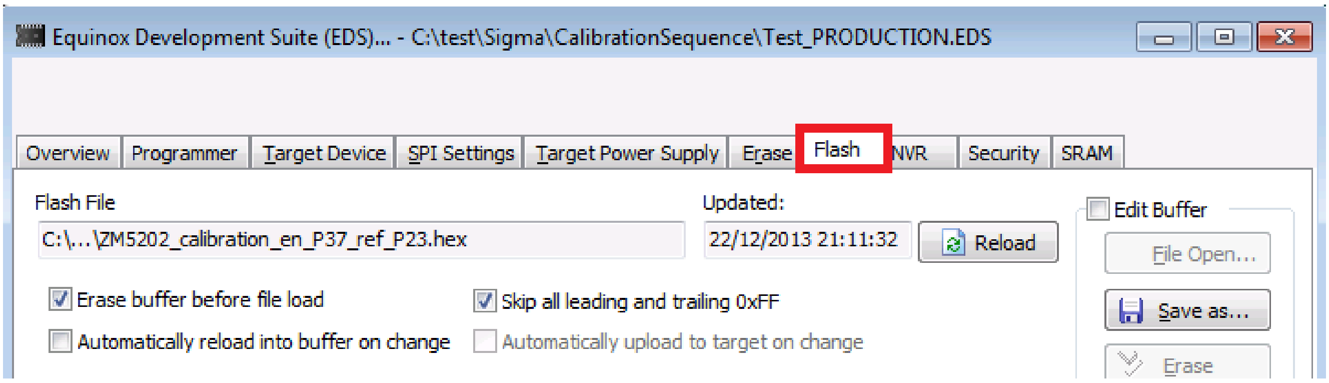


Project Manager - C:\test\Sigma\CalibrationSequence\Sigma_SD3502_project_collection.PPC

Number	Unique Id	Version	Build Date	Target Device	Target Programming Interface	Signature
0	SD3502-SPI	1.0.0.0	02/01/2014 at 19:03	SD3502 (SPI)	SPI 3-wire + RESET_N (Z-Wave)	0x7F7F7F7F1F0401
1	CALIBRATION	1.0.0.0	02/01/2014 at 19:04	SD3502 (SPI)	SPI 3-wire + RESET_N (Z-Wave)	0x7F7F7F7F1F0401
2	RUNTARGET	1.0.0.0	02/01/2014 at 19:05	SD3502 (SPI)	SPI 3-wire + RESET_N (Z-Wave)	0x7F7F7F7F1F0401
3	PRODUCTION	1.0.0.0	13/01/2014 at 21:15	SD3502 (SPI)	SPI 3-wire + RESET_N (Z-Wave)	0x7F7F7F7F1F0401

Programmer Type: ISPnano Series III/IV | Total FLASH usage - Used: 141.0 KB (0.9%), Free: 15.9 MB bytes (99.1%) | Programmer Flash Page Size = 2

The selected project should then launch in ***'EDS - Development mode'***....



Equinox Development Suite (EDS)... - C:\test\Sigma\CalibrationSequence\Test_PRODUCTION.EDS

Flash File: C:\...\ZM5202_calibration_en_P37_ref_P23.hex

Updated: 22/12/2013 21:11:32

☑ Erase buffer before file load

☑ Skip all leading and trailing 0xFF

□ Automatically reload into buffer on change

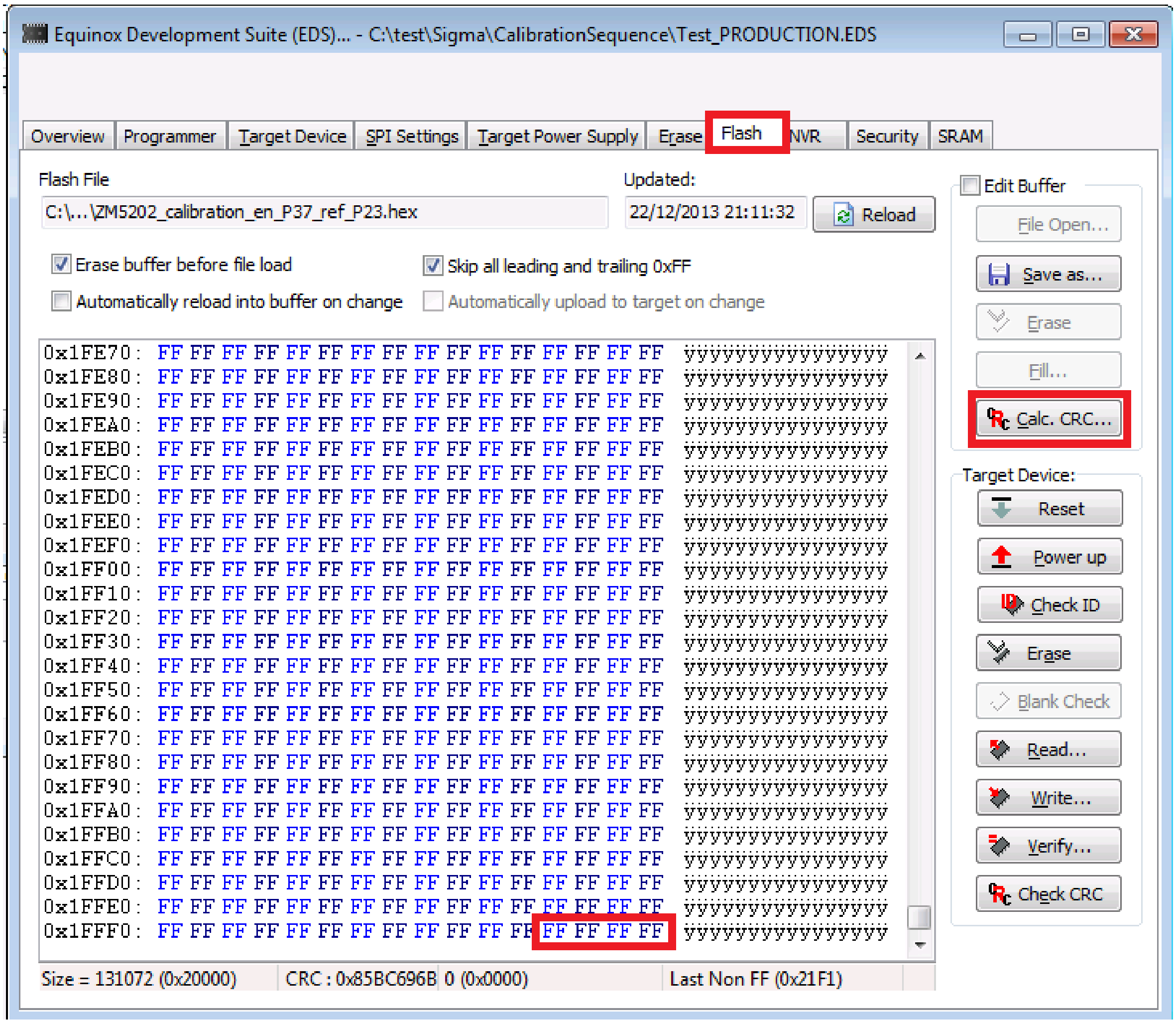
□ Automatically upload to target on change

□ Edit Buffer

1.4 Calculating the CRC32 checksum of the input file

The ***'CRC32 checksum'*** of the input file can be calculated as follows...

- Select the ***'FLASH'*** tab
- Your selected input file should be displayed in the buffer area
- Use the scroll bar on the right-hand side of the EDS window to scroll down to the end of the target device FLASH area.



Flash File: C:\...\ZM5202_calibration_en_P37_ref_P23.hex

Updated: 22/12/2013 21:11:32

☑ Erase buffer before file load

☑ Skip all leading and trailing 0xFF

☐ Automatically reload into buffer on change

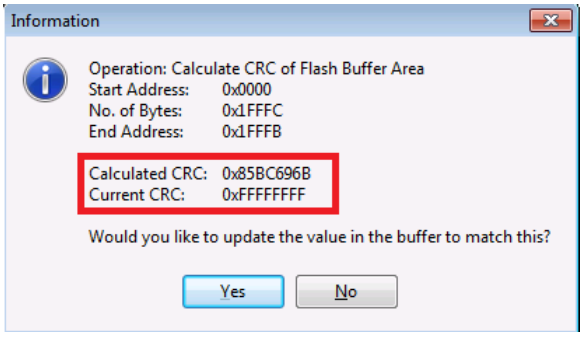
☐ Automatically upload to target on change

☐ Edit Buffer

```
0x1FE70: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x1FE80: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x1FE90: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x1FEA0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x1FEB0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x1FEC0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x1FED0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x1FEE0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x1FEF0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x1FF00: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x1FF10: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x1FF20: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x1FF30: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x1FF40: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x1FF50: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x1FF60: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x1FF70: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x1FF80: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x1FF90: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x1FFA0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x1FFB0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x1FFC0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x1FFD0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x1FFE0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x1FFF0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
```

Size = 131072 (0x20000) | CRC : 0x85BC696B 0 (0x0000) | Last Non FF (0x21F1)

- As you can see, the last 4 bytes of FLASH are set to 0xFFFFFFFF. This means that the ***'CRC32 checksum'*** is invalid.
- Now click the ***'Calc CRC'*** button
- EDS will now calculate the ***'CRC32 checksum'*** for the file loaded into the buffer area.

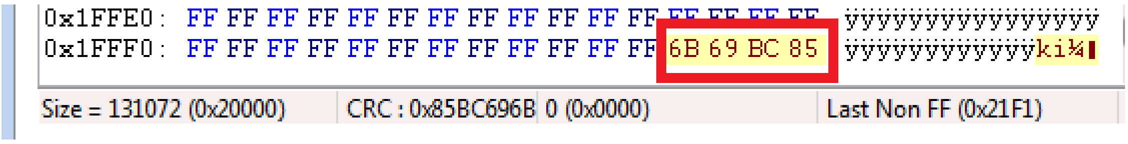


Operation: Calculate CRC of Flash Buffer Area
Start Address: 0x0000
No. of Bytes: 0x1FFFC
End Address: 0x1FFFB

Calculated CRC: 0x85BC696B
Current CRC: 0xFFFFFFFF

Would you like to update the value in the buffer to match this?

- Click the ***'Yes'*** button to update the ***'Calculated CRC'*** value into the last 4 bytes of the ***'FLASH buffer'***
- If you look at the last 4 bytes of the ***'FLASH buffer'***, they have now been updated with the ***'Calculated CRC checksum value'***....

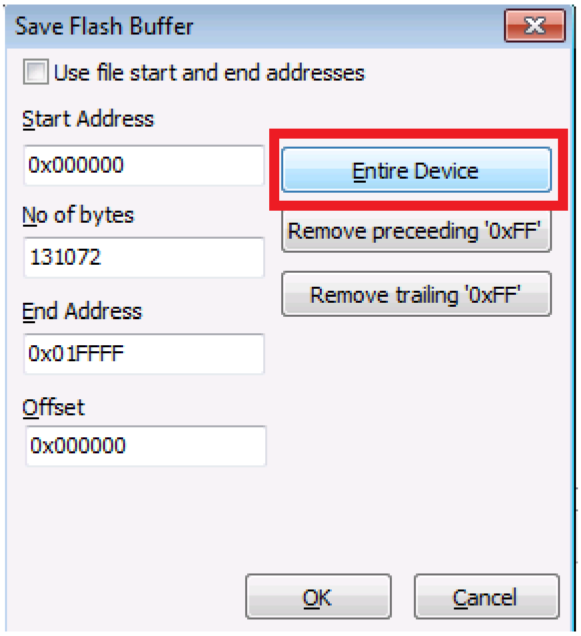


```
0x1FFE0:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x1FFF0:  FF FF FF FF FF FF FF FF FF FF FF FF 6B 69 BC 85  ÿÿÿÿÿÿÿÿÿÿÿÿki¼▌
Size = 131072 (0x20000)    CRC : 0x85BC696B  0 (0x0000)    Last Non FF (0x21F1)
```

1.5 Saving the revised file with the CRC32 checksum

You now just need to save the modified ***'FLASH buffer'*** back to your original hex file.

- Click the ***'Save as'*** button
- The ***'Save FLASH buffer'*** window is now displayed.....



Save Flash Buffer

☐ Use file start and end addresses

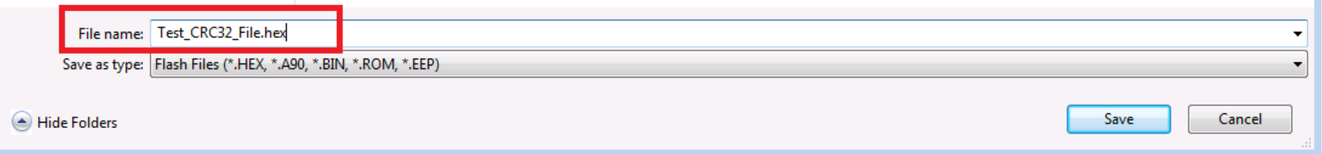
Start Address: 0x000000

No of bytes: 131072

End Address: 0x01FFFF

Offset: 0x000000

- Click the ***'Entire device'*** button --> This selects the entire address range of the device which includes the CRC32 value stored in the last 4 bytes of the FLASH.
- Click ***'OK'*** and then browse to the folder where you want to save the file.



File name: Test_CRC32_File.hex

Save as type: Flash Files (*.HEX, *.A90, *.BIN, *.ROM, *.EEP)

- Click the ***'Save'*** button to save the file.
- The contents of the 'FLASH buffer' including the now valid ***'CRC32 checksum'*** is saved to the specified file name.
- This file can now be loaded back into the ***'PRODUCTION'*** project as it has a valid ***'CRC32 checksum'***.

Appendix 6 - Sigma SD3502 Evaluation Module

1.0 Overview

This section describes how to interface an Equinox ISPnano programmer to a ***'Sigma SD3502 Evaluation Module'*** via the ***'UART'*** programming interface.

1.1 Equipment required

The following equipment is required for the programming evaluation.....

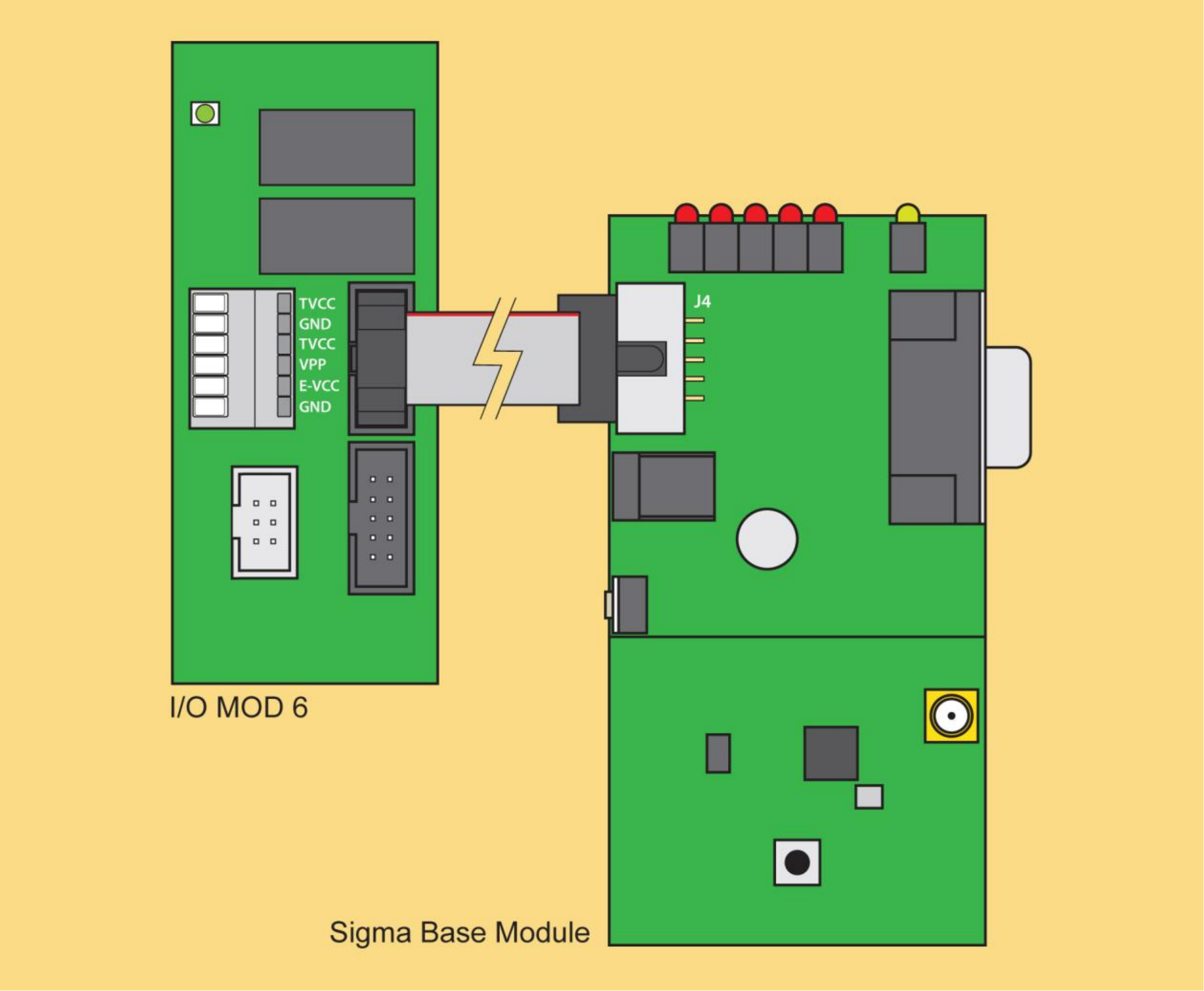
- ISPnano Series IV programmer
- IOMOD6 - I/O Connector Module
- Sigma SD3502 Evaluation Module (available to order from Sigma Designs)
- 10-way IDC ribbon cable

1.2 Connecting the programmer to the Sigma eval module

This section describes how to interface an Equinox ISPnano programmer to a 'Sigma SD3502

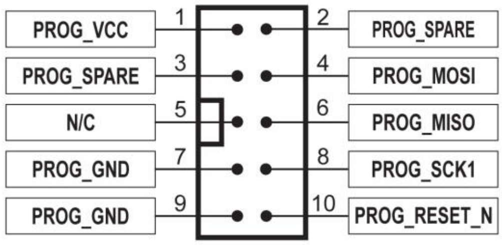
Instructions:

- Insert the **'IOMOD6'** I/O Connector Module into the **'ISPnano Series IV'** programmer
- Plug one end of the 10-way IDC cable into the ***'Equinox header'*** 10-way IDC connector on the 'IOMOD6' module.
- Plug the other end of the 10-way IDC cable into the 10-way IDC connector on the bottom (base) PCB of the ***'Sigma SD3502 Evaluation Module'*** - see illustration below



Appendix 7 – Sigma connector definitions

1.0 Sigma - ISP Header Selection

#	ISP Header	Description / Function	ISP Header Pin-out
1	J6	**Equinox 10-way Header(a)** **Device support:** Sigma Z-Wave 500 series devices via the 'SPI' interface.	 PROG_VCC 1 2 PROG_SPARE PROG_SPARE 3 4 PROG_MOSI N/C 5 6 PROG_MISO PROG_GND 7 8 PROG_SCK1 PROG_GND 9 10 PROG_RESET_N