Application Note

Report No:

AN145

Title:

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

Author:	Date:	Version Number:
John Marriott	11th September 2015	0.80

Contents

1.0 Introduction 7
1.1 Features 7
1.2 Programmers supporting Z-WAVE 500 series devices 9
1.3 Calibration overview 10
1.3.1 Overview 10
1.3.2 Crystal (XTAL) calibration 10
1.3.3 TX calibration 11
1.3.4 Calibration recovery after accidental NVR erasure / corruption 11
1.3.5 Programmer selection guide for Z-Wave calibration 12
1.4 Device Support 13
1.4.1 Overview 13
1.5 Upgrading your Equinox Programmer to support Sigma 500 series Z-WAVE device programming 14
1.5.1 Overview 14
1.5.2 Purchasing a Sigma Z-WAVE 500 series License 14
1.5.3 How do I enable the programmer for Z-WAVE programming? 14
1.5.4 Entering the License String to upgrade your programmer 15
1.6 Programmer firmware versions for Sigma 500 series support 16
2.0 Z-Wave - Programming Interfaces 17
2.1 Overview 17
2.2 SPI - Programming Interface - Features 17
2.3 Z-WAVE single-chip In-System Programming (ISP) Schematic 18
2.4 ISPnano programmer - Target ISP Port - SPI connections 19
2.4 Z-WAVE UART ISP Schematic 20
3.0 Creating an EDS (Development) Project 23
3.1 Overview 23
3.2 Information required to create an EDS Development project 23
3.3 Creating an EDS (Development project) 24
3.4 Selecting the correct Target Device 25
3.4.1 Device selection 25
3.4.2 Device Chip ID / Signature 26
3.5 Target System – Power Supply Settings 27
3.6 Erase options 28
3.7 Specifying the FLASH (Code) File 29
3.8 Launching EDS at the end of the EDS Wizard 31
4.0 Testing a Project in Development (EDS) Mode 33
4.1 Introduction to EDS 33
4.2 EDS - Default settings for SPI, statemachine etc 34
4.3 SPI - speed settings 35
4.4 Checking the Target Voltage 35
4.5 Testing SPI communication with the 500 series device 36
4.5.1 Overview 36
4.5.2 Diagnostic Info 36
4.5.3 Possible failure messages 37
4.6 NVR Area - reading / writing in EDS mode 38
4.6.1 Overview 38
4.6.2 NVR Area tab - explanation 38
4.6.3 Reading the 'NVR parameters' from a device 39
4.6.4 Exporting specified 'NVR parameters' to an NVR parameter file 40
4.6.5 Importing an 'NVR parameter file' back into EDS 41
Appendix 1 – Standalone Programming Mode setup 43
1.0 Overview 43
1.1 Information required to create a Standalone Programming Project 44

1.2 Software versions for Standalone Programming mode 44
2.0 Explanation of 'standalone mode projects' 45
3.0 Running a 'Standalone Programming Project' 46
4.0 Creating a 'Standalone Programming Project' 47
4.1 Getting started 47
4.2 Header Information screen 47
4.3 Programmer and Project Type options 48
4.4 Select the 'Target Device' 49
4.5 Select the 'SPI frequency' for programming 50
4.6 Setting up the 'Chip Erase' / 'Amend NVR parameters' options 51
4.7 FLASH Area - programming options 52
4.8 Final 'Congratulations' screen - saving your project 53
4.9 Importing an 'NVR parameter file' into a standalone project 54
4.10 Compiling your standalone project 55
4.11 Adding your standalone project to a Project Collection 56
4.12 Uploading your standalone project to a programmer 56
Appendix 2 - ISP-PRO script - how to set up the PRODUCTION standalone project(s) 57
1.0 Overview 57
2.0 Explanation of 'standalone projects' 57
2.1 Opening the Sigma Project Collection 59
2.2 Changing the 'Product firmware' file 60
2.3 Uploading the new Project Collection to the programmer 66
Appendix 3 - Setting up the 'Tx Power' parameters 67
1.0 Overview 67
1.1 Why do I get a 'TX Power parameter' error? 67
1.2 Where do I find the 'Tx Power' settings in Sigma's SDK 68
1.3 'Tx Power' parameters – overview of merging process 69
1.4 'Tx Power' parameters – merging into the FLASH hex file 70
Appendix 4 - Configuring the Z-Wave 'External Non-volatile memory (NVM)' parameters 73
1.0 Overview 73
1.1 NVM memory – configuration parameters overview 73
2.0 No external NVM (memory) device fitted on target board 74
3.0 Custom NVM (memory) device fitted on target board 74
3.1 How to work out the NVM (memory) device parameter values 75
3.2 Configuring the script file to program custom NVM (memory) device parameter values 75
3.3 Configuring individual NVM (memory) device parameters 77
3.4 Testing the custom NVM (memory) device parameters 78
Appendix 5 – Restore 'NVR Area' script 79
1.0 Overview 79
1.1 Modifying the default 'Restore NVR' script 79
1.2 Testing the 'Restore NVR' script in ISP-PRO 81
1.3 Checking the final 'NVR settings' are correct 81
Appendix 6 - ISP-PRO - Quick Start Guide 83
1.0 Overview 83
2.0 Installing the Sigma scripts and projects 83
2.1 Overview 83
2.2 Overview of files required for ISP-PRO software 84
2.3 Setting up ISP-PRO to run the Programming Script(s) 85
2.4 Start the ISP-PRO application and log in 85
2.5 Setting up the correct COM port 86
2.6 Detecting the attached programmer(s) 87
2.7 Selecting a Script File to run 87
2.8 Uploading your Project Collection to the programmer(s) 89
2.9 Running the programming Script(s) 92
2.10 Executing the programming / calibration sequence 93
2.11 Programming sequence - PASS 94

2.12 Programming sequence - FAIL 95
3.0 Setting up an Incremental Serial Number 96
3.1 Overview 96
3.2 Importing a Serial Number from file 96
Appendix 7 - FLASH File - CRC32 Checksum 99
1.0 Overview 99
1.1 Using EQTools - EDS to update the CRC32 checksum 100
1.2 Why do I get an invalid FLASH CRC32 checksum error? 100
1.3 Loading standard Z-Wave Firmware releases from Sigma 101
1.4 Loading a custom Z-Wave Firmware release 101
1.5 How to correct / add a valid CRC32 checksum to an input file 101
1.6 Opening the project in EDS (Development mode) 102
1.7 Calculating the CRC32 checksum of the input file 102
1.8 Saving the revised file with the CRC32 checksum 104
Appendix 8 - Sigma SD3502 Evaluation Module 105
1.0 Overview 105
1.1 Equipment required 105
1.2 Connecting the programmer to the Sigma eval module 106
Appendix 9 – Sigma connector definitions 107
1.0 Sigma - ISP Header Selection 107
Appendix 10 – Connecting the Z-Wave target board to an ISPnano programmer 109
1.1 IOMOD10 module - connections to Z-Wave target board (SPI interface) 109
1.2 IOMOD7 module - connections to Z-Wave target board (SPI interface) 111
1.3 IOMOD4 connector module - connections to Z-Wave target board (SPI interface) 112
1.4 IOMOD4 connector module + Clock Buffer - connections to Z-Wave target board (SPI interface) 113

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

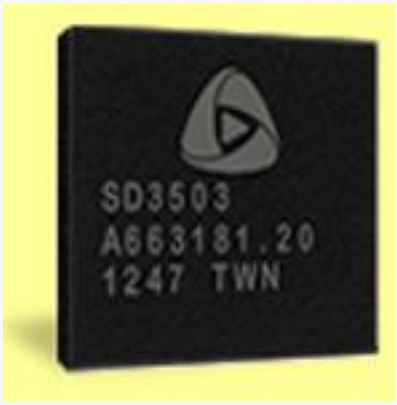
The table below details which Equinox programmers are capable of performing the ***'Crystal (XTAL) calibration'*** and ***'TX calibration'*** procedures.

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.

1.5.4 Entering the License String to upgrade your programmer

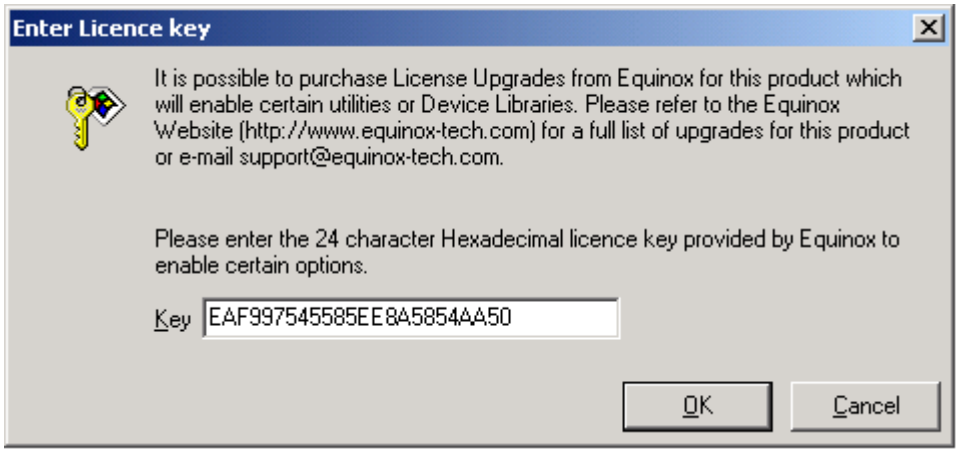
Once you have received the License String from Equinox, please follow the steps below to apply the upgrade to your programmer:

- 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

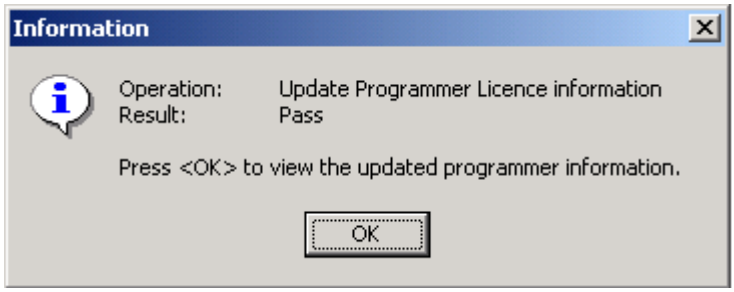
→ The ***<Enter License Key>*** screen is displayed.



Enter the License String you were sent by Equinox

- Click ***<OK>***

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



- 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	6.28
ISPnano Series IV	6.28
ISPnano-MUX 2 / 4 / 8	6.28

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 Z-Wave - Programming Interfaces

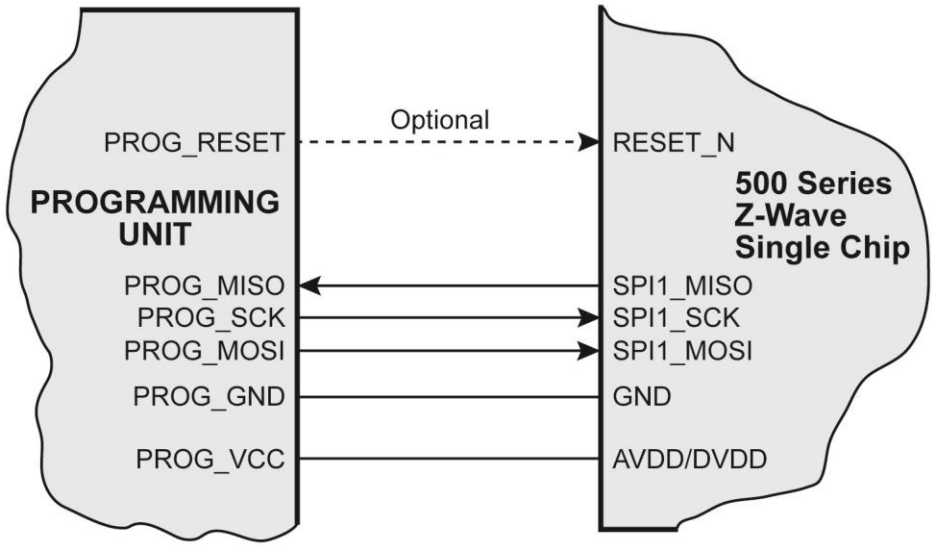
2.1 Overview

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

Interface		Comment
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 speed (compared to UART interface)
- 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 (ISP) of a single ***'Z-WAVE 500 series'*** device via the 'SPI programming interface' using an Equinox ISP programmer.

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

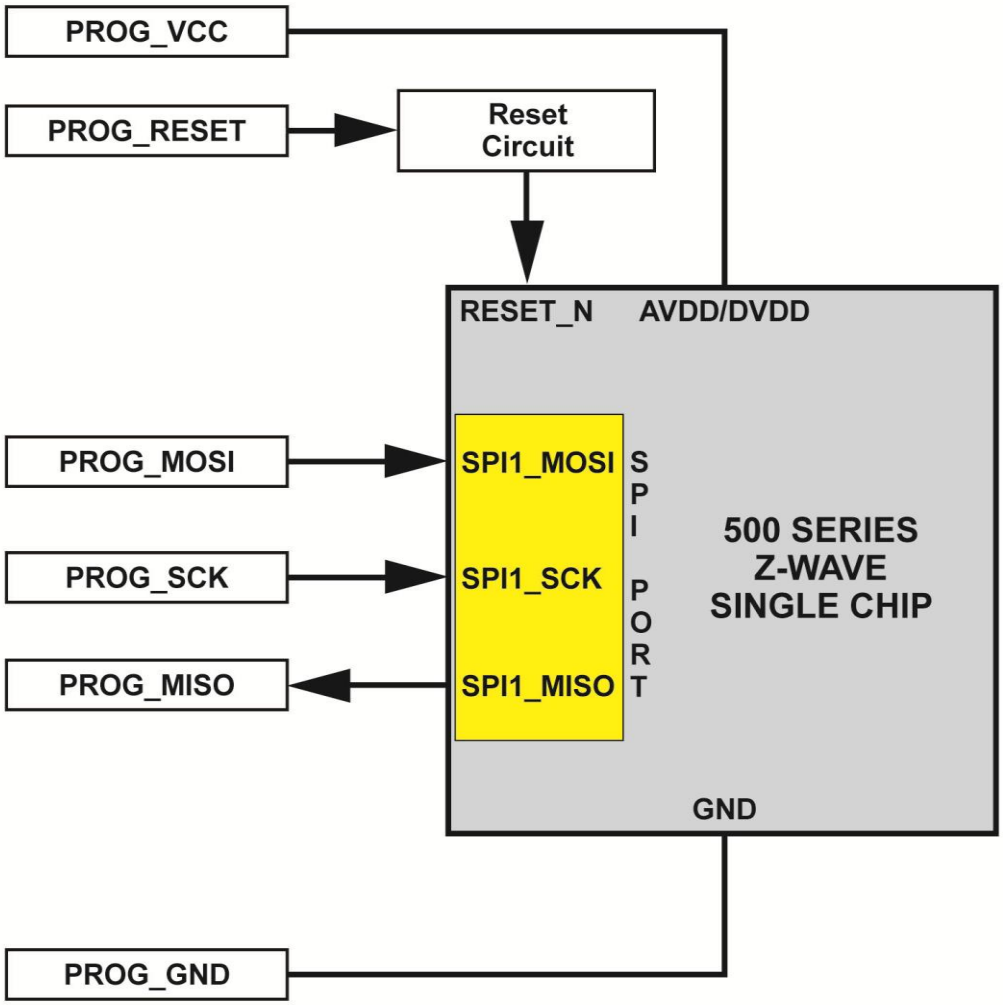
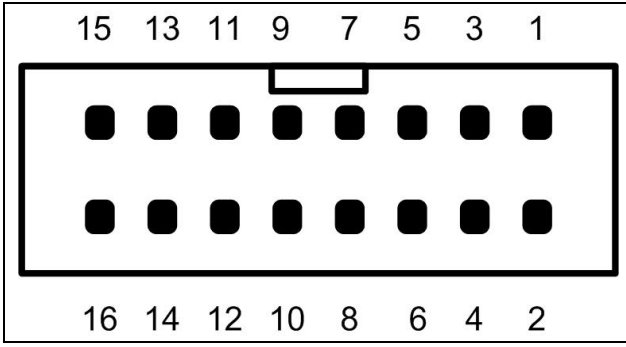


Fig 2.3.b – Sigma Z-wave 500 series device - SPI Signal names and directions

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 ISPnano programmer - Target ISP Port - SPI connections

The table below details the connections for programming a Z-WAVE 500 series device via the ***'SPI Interface'*** using the ***'Target ISP Port'*** on the ***'ISPnano Series 3'***, or ***'ISPnano Series 4'*** programmer..



'Target ISP Connector' port

The connector is a 16-pin bump-polarised IDC connector with 0.1" pin spacing.

Pin 1 is the top right pin as shown in the diagram opposite.

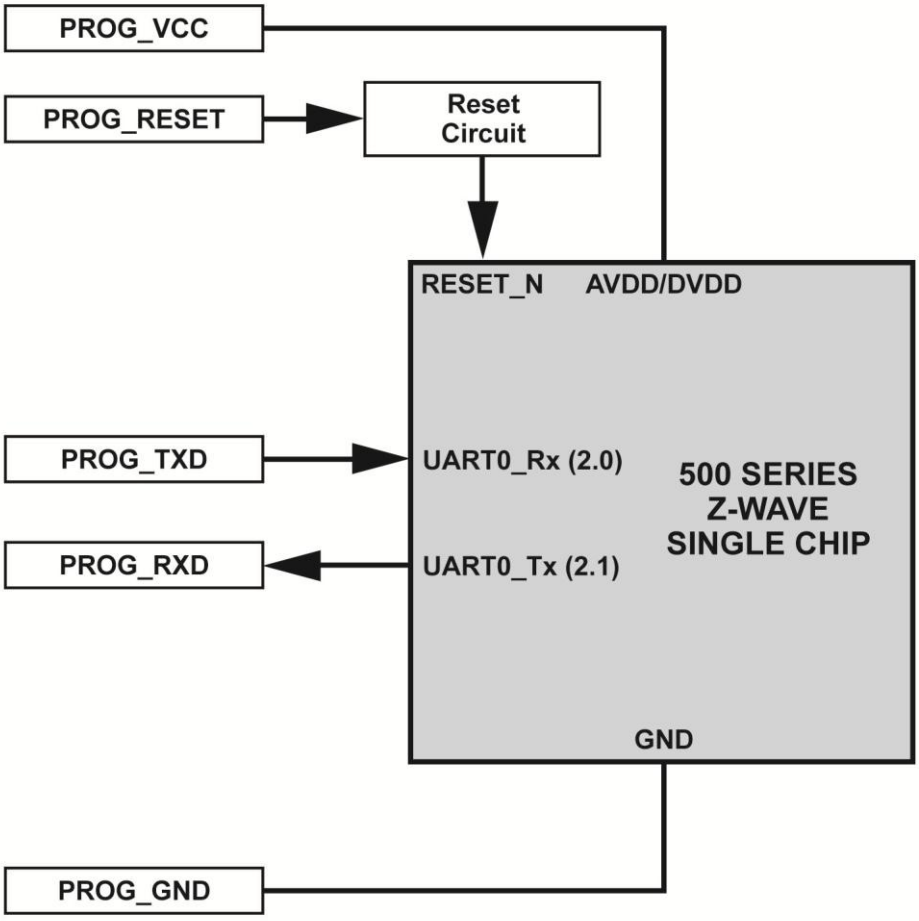
Pin No	Programmer Pin name	Programmer Input / Output	Connect to pin on target Z-Wave device	Notes
1 + 2	TARGET_VCC	P	TARGET_VCC	Target VCC
3 + 4	TARGET_EXT_VCC	P	See notes.	Target External VCC
5 + 6	PROG_GND	P	Signal GROUND (0V)	Signal Ground Connection
12	Programmer I/O3	I/O	SPI - SCK	SPI - Serial Clock Signal
13	Programmer I/O2	I/O	SPI - MISO	SPI – Master In Slave Out
14	Programmer I/O1	I/O	SPI - MOSI	SPI – Master Out Slave In
16	PROG_RESET	O	RESET_N	Target RESET control pin

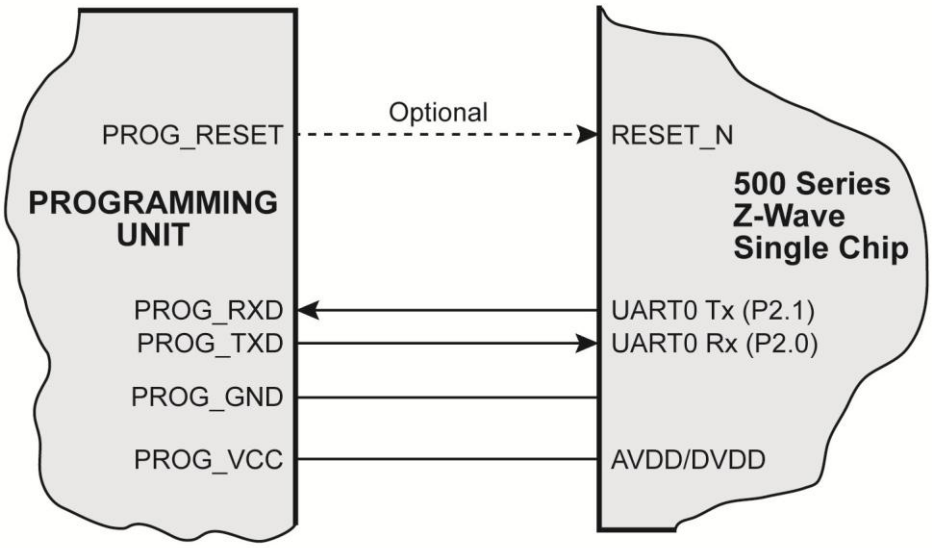
O - Output from programmer to Target Device
I - Input to programmer from Target Device
P - Passive e.g. GROUND and power rails
N/C - Not connected

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





UART Interface to a 3.3V UART Programming Equipment

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 an EDS Development project

The following information is required about your Target Board / application in order to create a development project for a ***'Z-WAVE 500 series'*** device.

#	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.

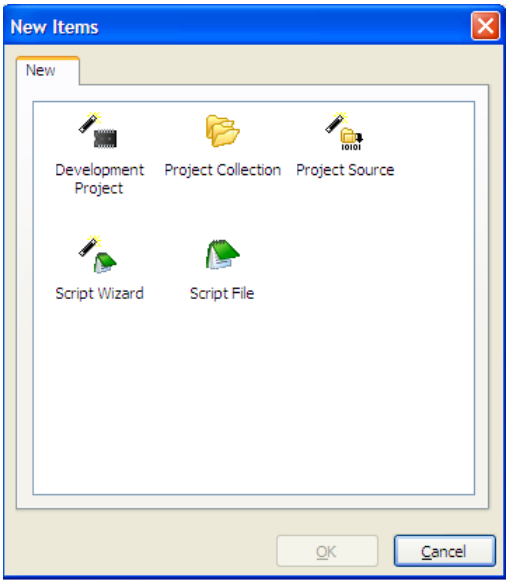
3.3 Creating an EDS (Development project)

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

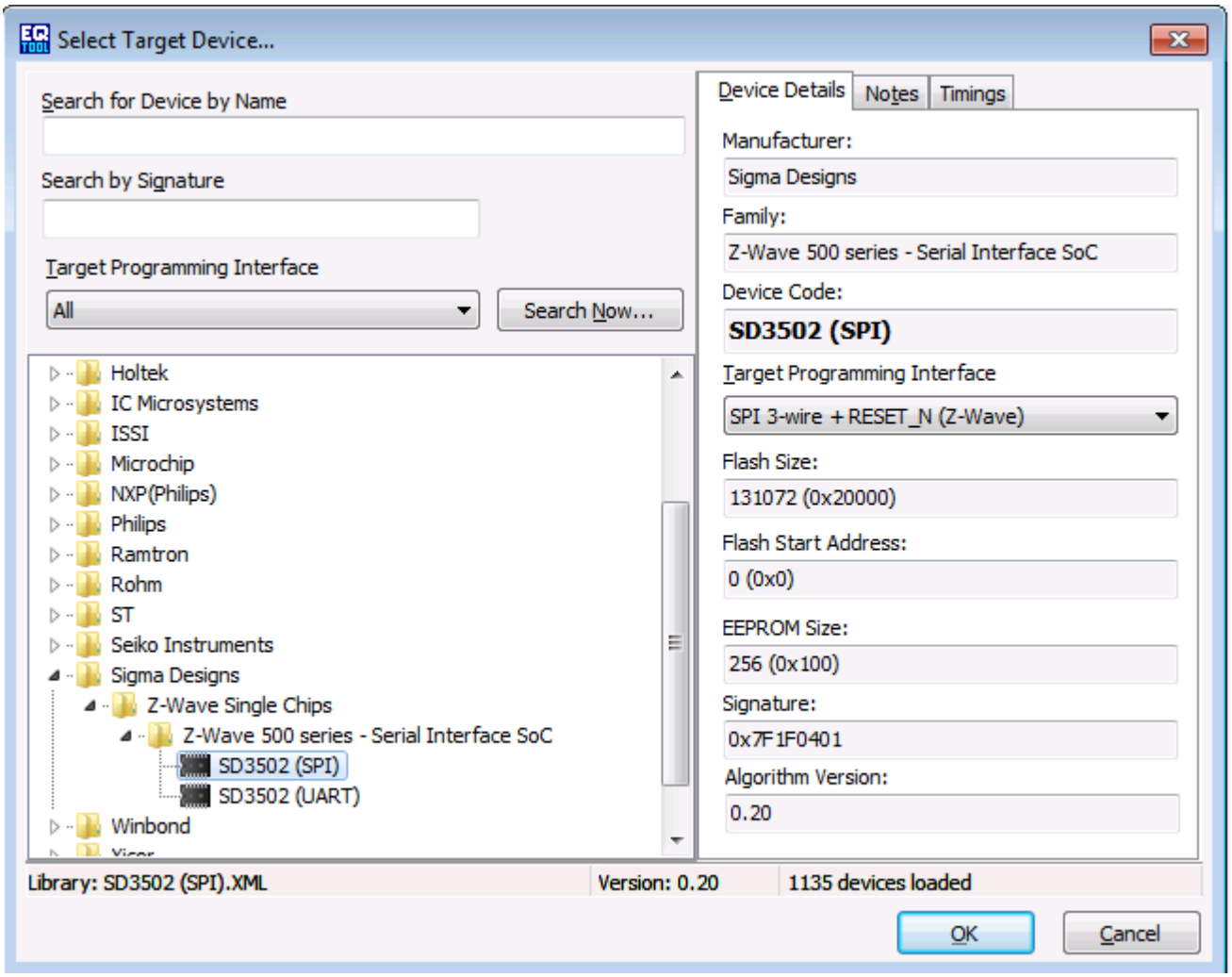
- Select the relevant 'Programmer' and then click ***<Next>***

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.

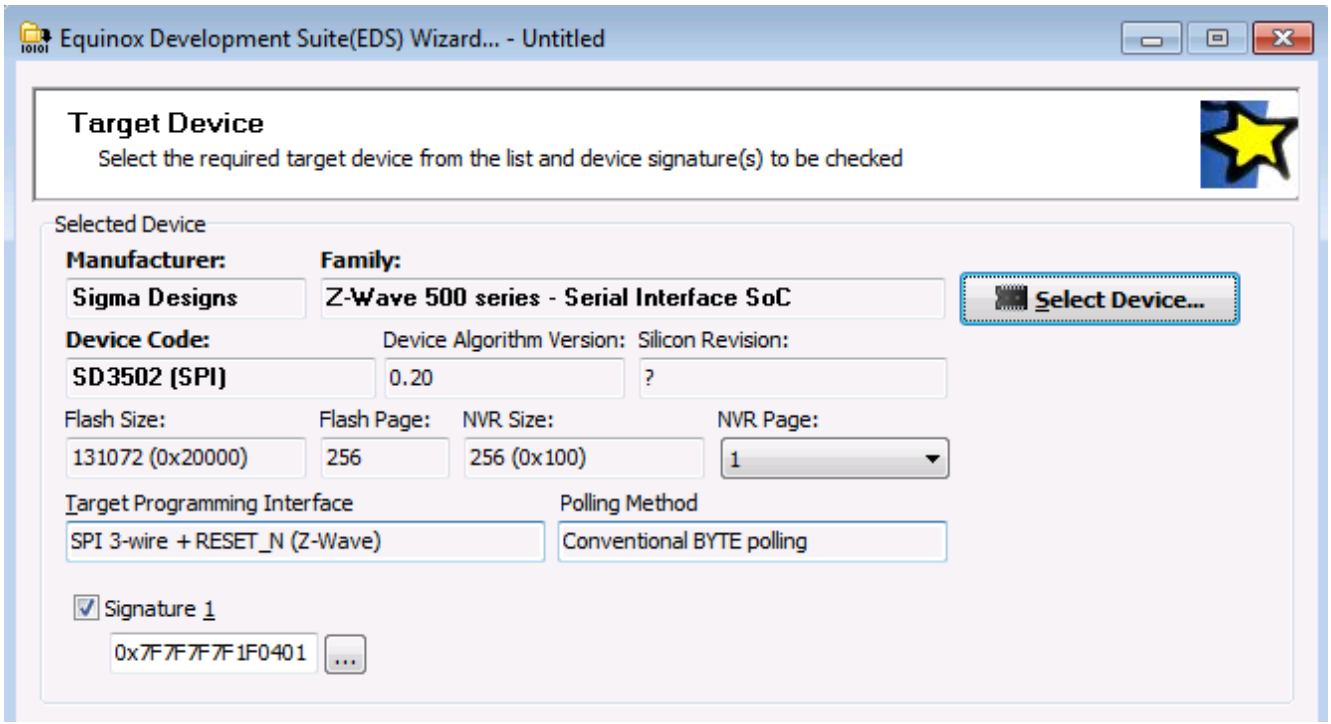


- As the ***Z-wave 500 series*** devices can be programmed via different 'programming interfaces', the device list shows the available interfaces eg. SPI or UART interfaces for the SD3502 device.
- Select the required device / programming interface from the list e.g. ***'SD3502 (SPI)'*** and then click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

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

3.4.2 Device Chip ID / Signature

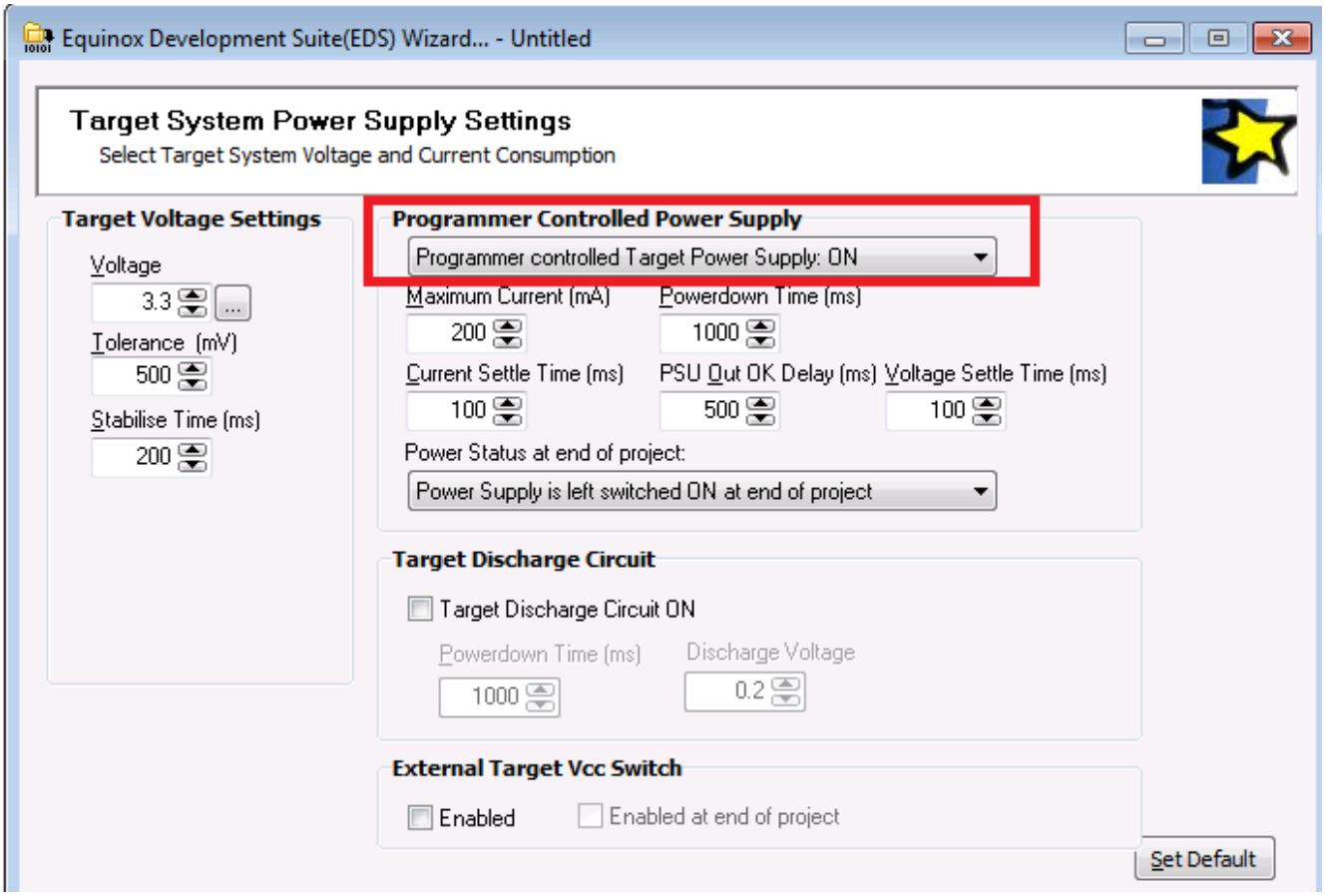
- On the next screen, check that the device selection and all other device parameters are correct



- 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.



i. Select the Target Voltage

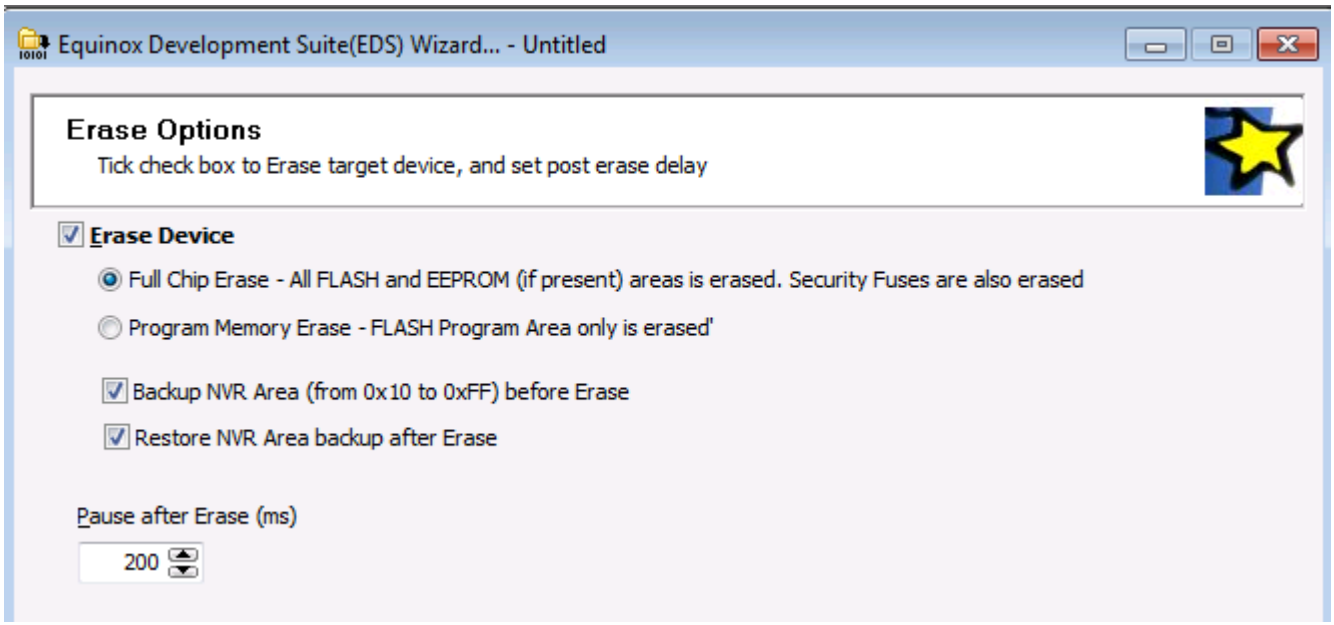
- This should be the voltage at which the Target Z-Wave device itself is being powered at during the programming operation. This is usually 3.0 – 3.6V.
- If the target board is being externally powered, then the programmer voltage should be set so that it matches the I/O voltage of the Z-Wave device.
- 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....



Backup NVR Area before Erase

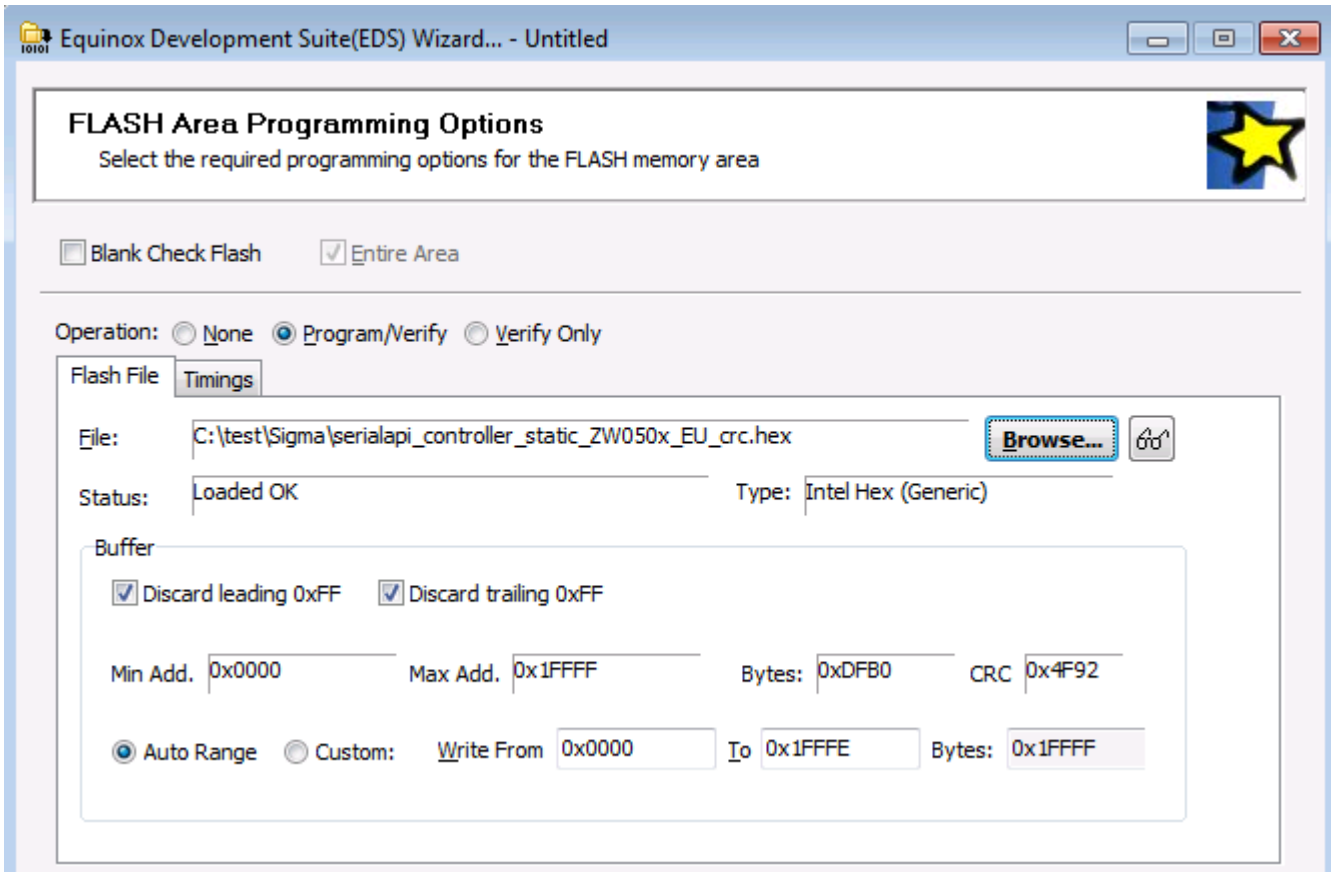
If this option is selected, the programmer will automatically read back the ***'NVR data area'*** from the target device before a ***'Chip Erase'*** operation is performed.

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

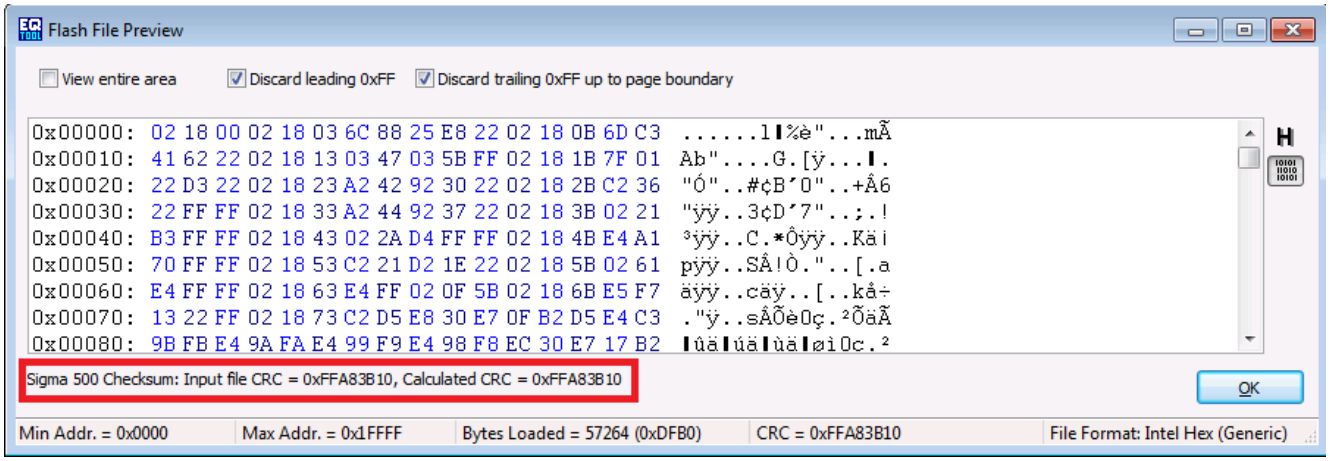
This screen allows you to specify the ***Code (firmware***) file which is to be programmed into the FLASH area of the Target Device.



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

Selecting the FLASH File

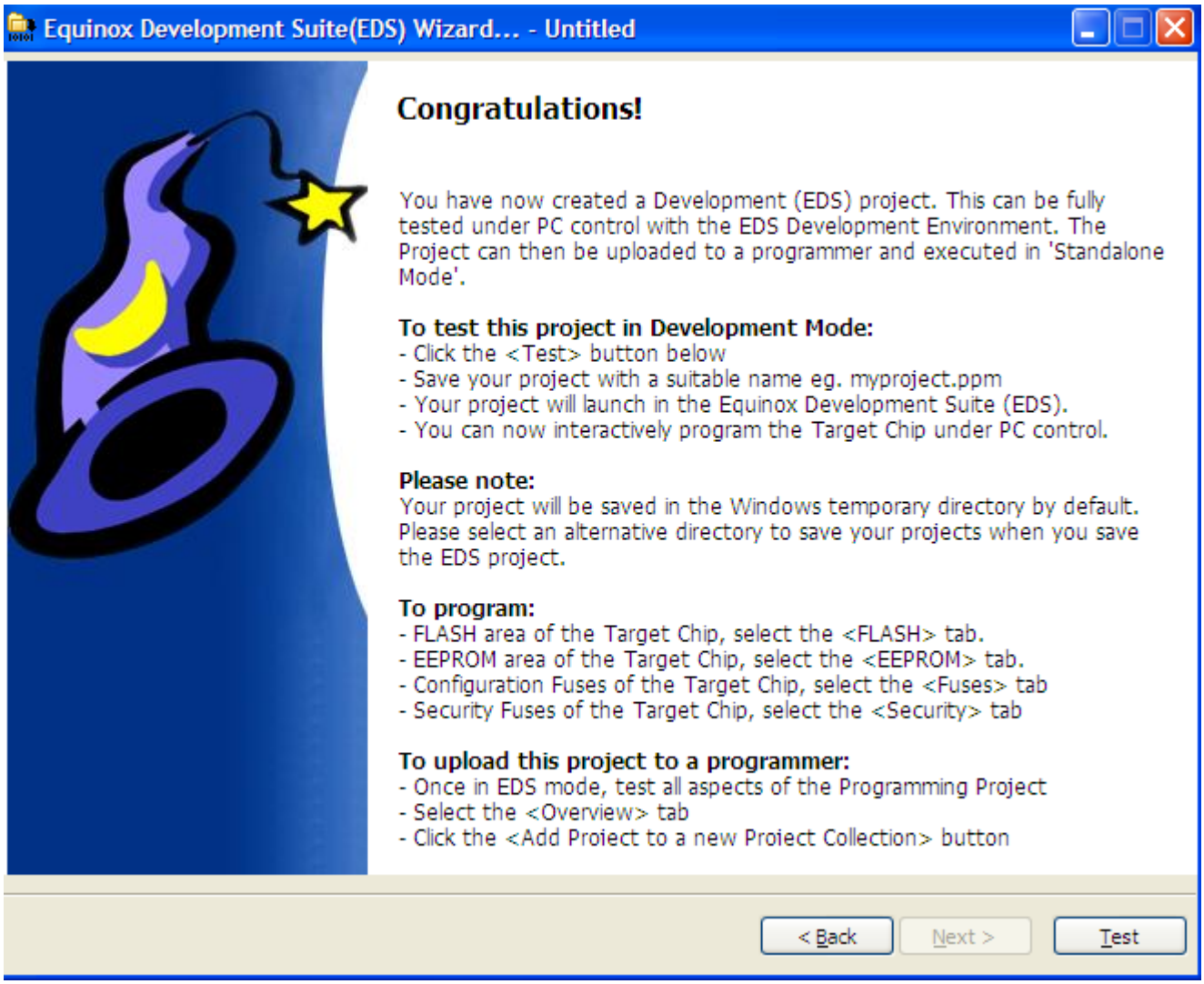
- Click the ***<Browse>*** button
- Browse to and select the file you wish to load and then select ***<OK>***
- --> The file will be automatically loaded into the 'FLASH File Preview' window - see below...



- 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).

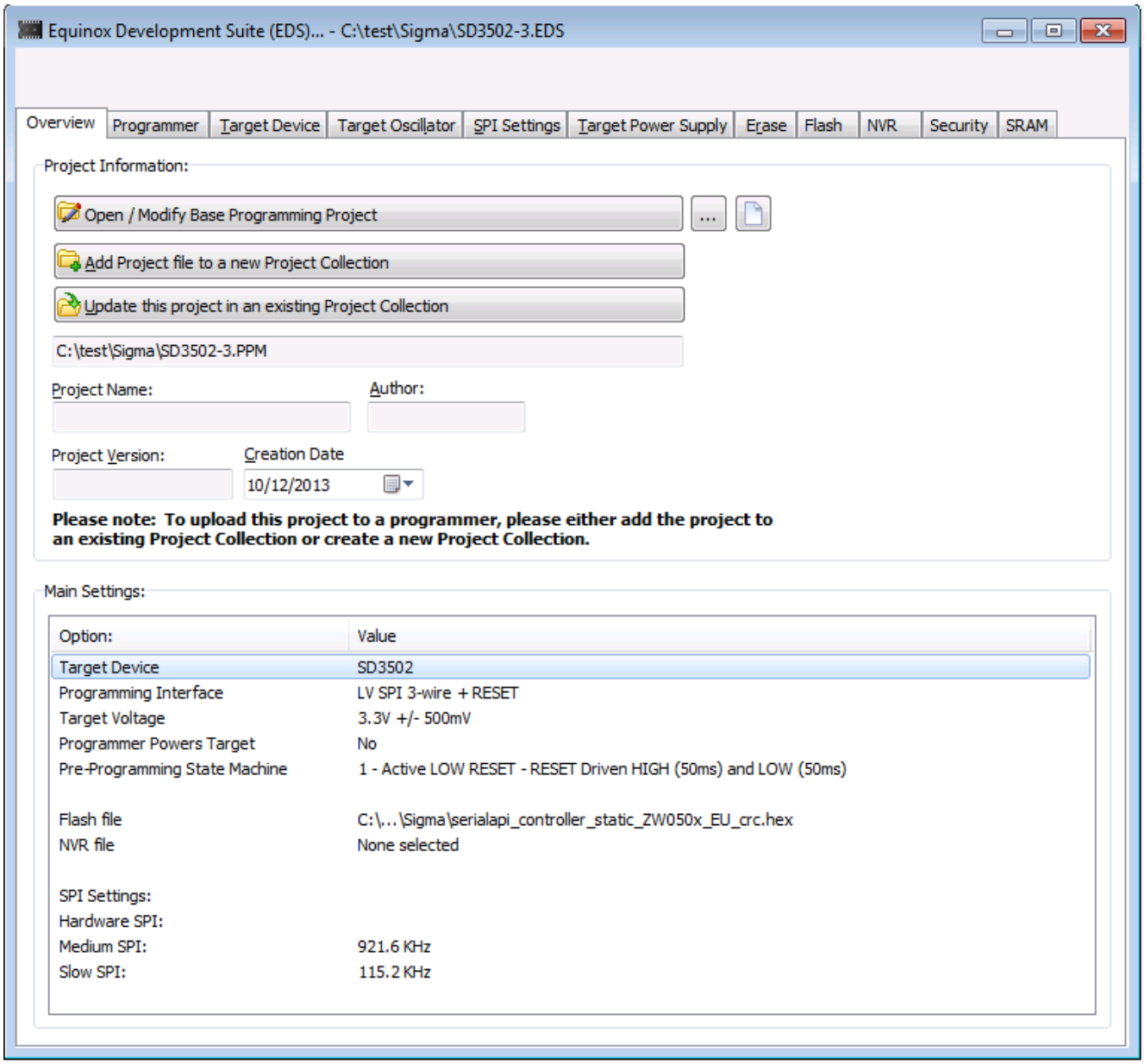


Enter a name for the EDS project e.g. ***SD3502*** and click the ***<Test>*** button
→ Your project will now launch in EDS (Development) Mode.

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.



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

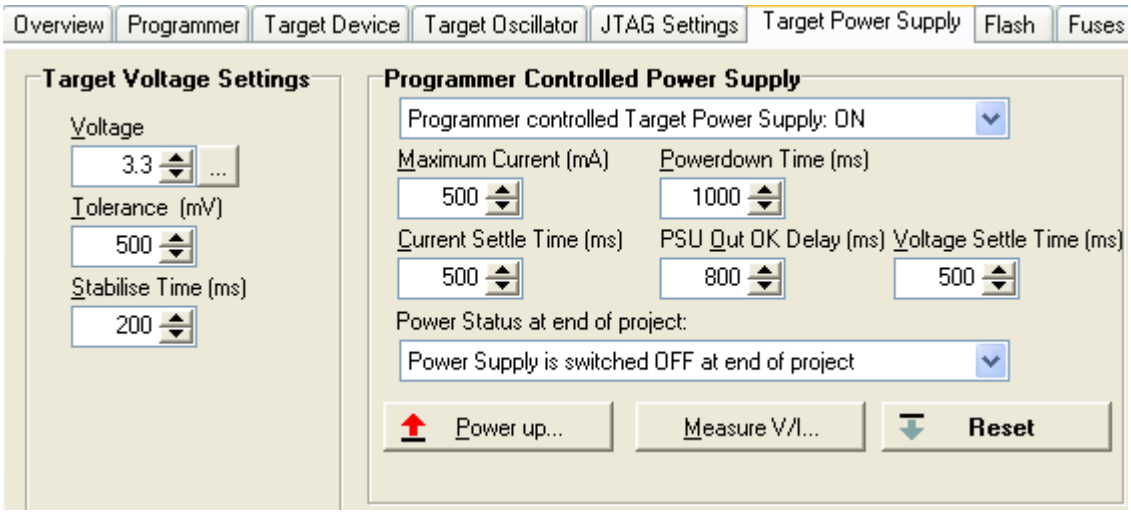
The ***'SPI speed'*** should be set up before any programming operation can take place.

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



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 <u>NOT</u> powering the Target System…

Switch on the independent power supply which is connected to the Target System.

Click the ***<Measure>*** button to measure the ***'Target Voltage'***.

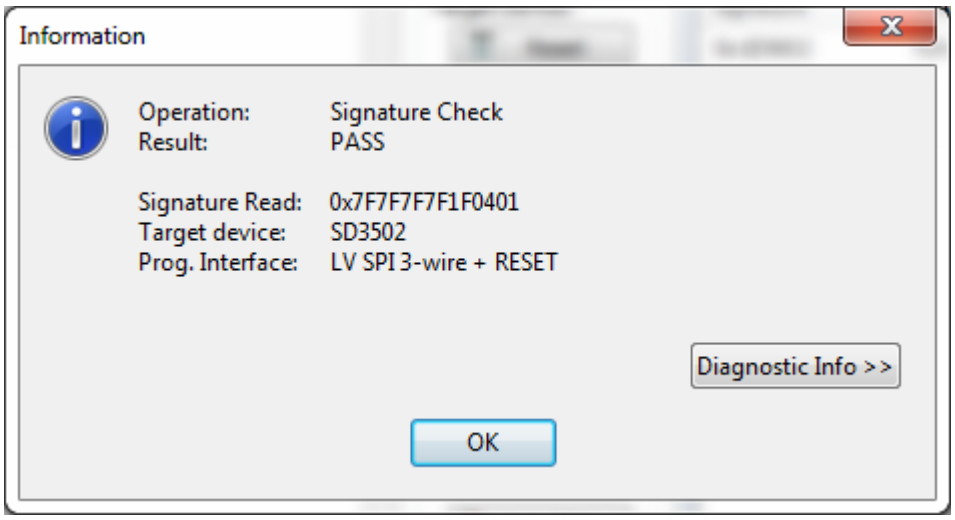
- The voltage should be within 3.0 and 3.6V.

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'***.



- 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.

4.6 NVR Area - reading / writing in EDS mode

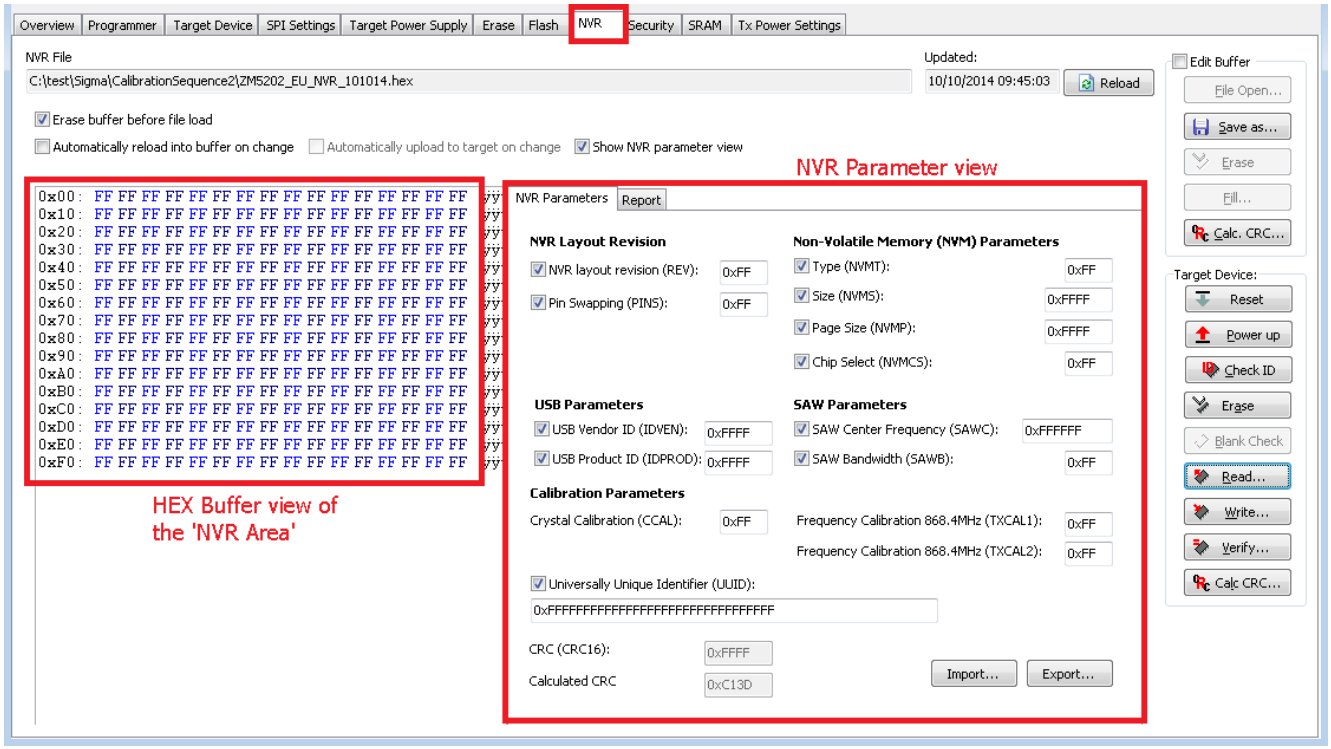
4.6.1 Overview

The ***'NVR Area'*** of a Z-Wave device contains all the ***'factory parameters'*** which are programmed into the device when the device is calibrated in the factory by Sigma. This data is fundamental to the operation of the Z-Wave module so it is very important that this data is never erased or corrupted. The ***'NVR Area'*** contains calibration coefficients and also some parameters to which allow the specification of external components to the Z-Wave IC including a Non-Volatile Memory (NVM) and 'Saw filter' components.

Using the ***EQTools - EDS*** tool, it is possible to read, backup, modify and write data in the Z-Wave ***'NVR Area'***. It is also possible to export a list of ***'NVR parameters'*** with specified values which can then be imported both into a ***'Standalone Programming Project'*** or an ***'ISP-PRO programming script'***.

4.6.2 NVR Area tab - explanation

To read or write the ***'NVR Area'*** of a Z-Wave device, select the ***'NVR'*** tab of the EDS session.
A sample of a typical ***'NVR'*** screen with all data set to 0xFF is shown below....



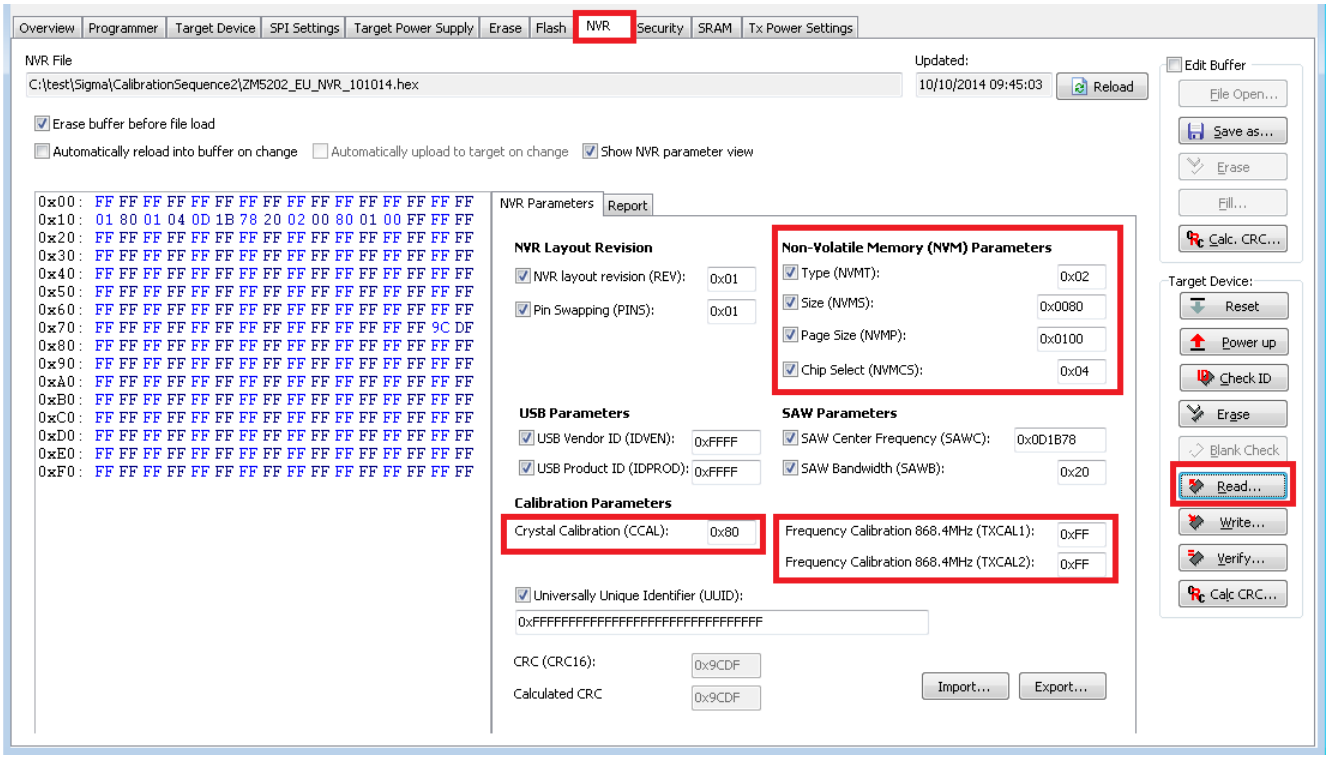
- The ***'HEX Buffer'*** view of the ***'NVR Area'*** on the left shows a hex representation of the entire 256 bytes of the ***'NVR Area'***.
- The ***'NVR Parameter'*** view of the ***'NVR Area'*** on the right lists the main ***'NVR Parameters'*** and their associated values.

4.6.3 Reading the 'NVR parameters' from a device

The ***'NVR parameters'*** can be read from the ***'NVR Area'*** of a Z-Wave device and displayed on screen.

Instructions:

- Select the EDS ***'NVR'*** tab
- Click the ***'Read'*** button on the right-hand side of the screen
- EDS will now prompt you for the address range. Click ***'OK'*** to read the entire 256 bytes back from the target device ***'NVR Area'***.
- The read back ***'NVR data'*** will now be displayed both in the ***'HEX Buffer'*** view and the ***'NVR Parameter'*** view - see screenshot below.....



The most important parameters to check are:

1. Crystal Calibration (CCAL)

If this field is 0x80, then the device is not XTAL (crystal) calibrated.

2. Frequency calibration (TXCAL1 / TXCAL2)

If either parameter is 0xFF, then the device has not been 'TX calibrated'.

3. Non-volatile (NVM) parameters

These parameters set up the type of external ***'Non-volatile memory (NVM)'*** device attached to the Z-Wave device or module.

4.6.4 Exporting specified 'NVR parameters' to an NVR parameter file

If you need to fix certain ***'NVR parameters'*** to specified values during the programming process, then this can be achieved by creating an ***'NVR parameter file (*.NEF)'***. This file contains a list of specified parameters with their required 'fixed' values.

A ***'NVR parameter file (*.NEF)'*** can then be imported either into a ***'Standalone programming project'*** or an ***'ISP-PRO programming script'*** and will then 'fix' the values of the specified parameters.

Instructions:

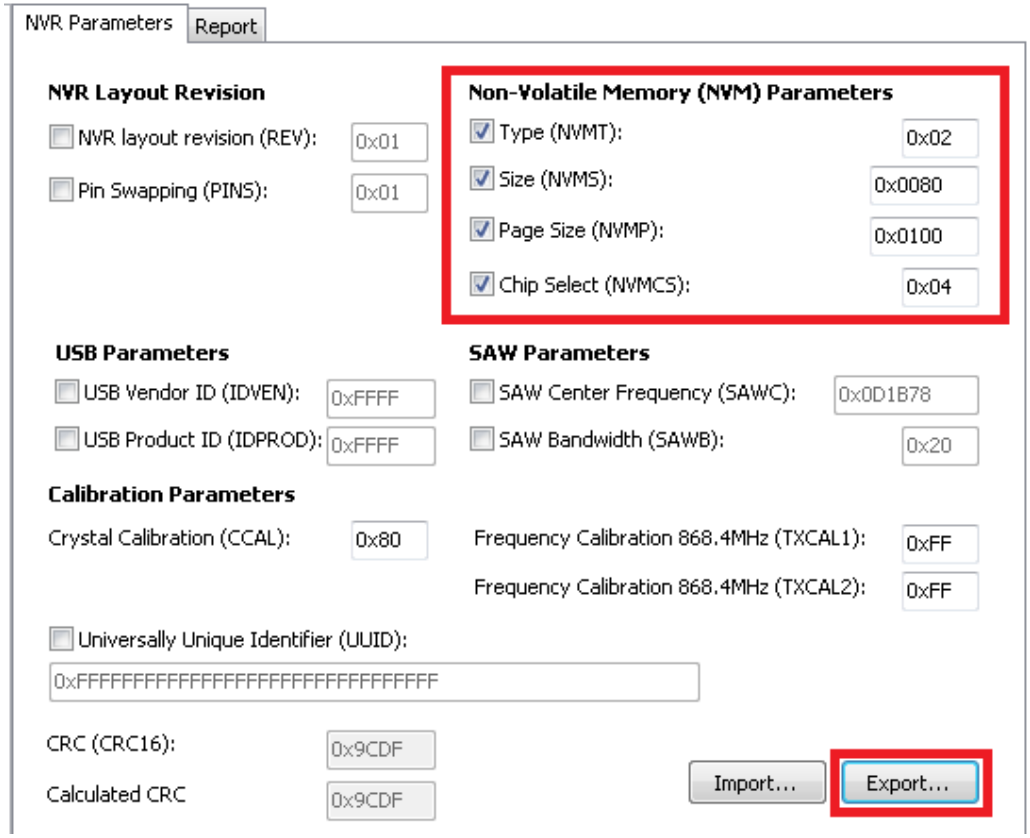
- Select the EDS ***'NVR'*** tab
- Read the ***'NVR area'*** back from a golden sample device which has all the correct values already programmed into it.

or

- Manually enter the required values into the relevant ***'NVR Parameter'*** fields.

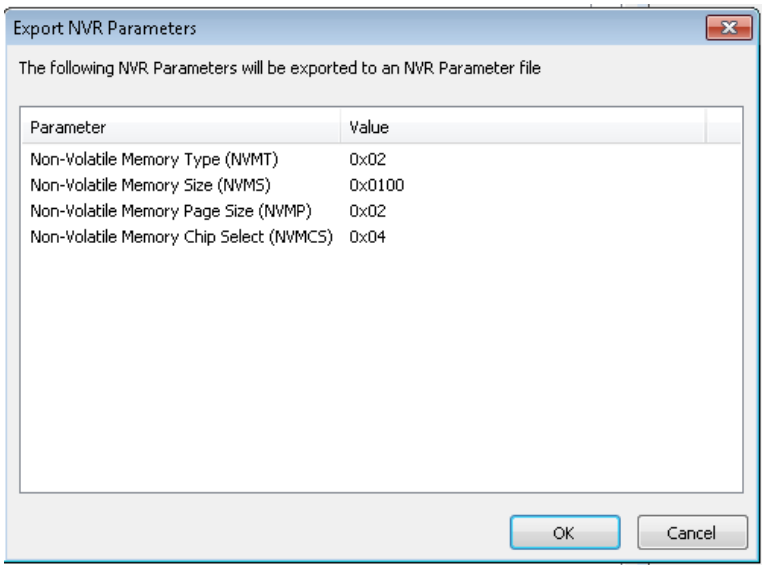
In the ***'NVR Parameter'*** view....

- Untick all the parameters which you do NOT wish to export to the ***'NVR parameter file (*.NEF)'***.
- Tick all the parameters which you wish to export to the ***'NVR parameter file (*.NEF)'***.
- Check that the values of all the parameters you want to export are the correct value.
- In the example screenshot shown below, only the external ***'Non-volatile memory (NVM)'*** parameters are selected for export.....



To export the selected ***'NVR Parameter'*** fields....

- Click the ***<Export>*** button
- The ***'Export NVR Parameters'*** dialog box will now be displayed.....



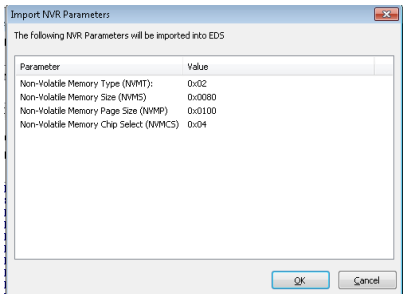
- The parameters you selected should now be displayed with the 'fixed values' which you want to program into the target device.
- Check that all the required parameters have been exported and that the values are correct.
- Once you are happy the exported parameter list is correct, click the <OK> button
- It is now possible to save the ***'NVR Parameter list'*** to a file.
- Enter a suitable file name e.g. ***NVR_Exported_parameters.NEF*** and then click 'Save' to save the parameter list to a ****.NEF*** file
- The ***'NVR Parameter list'*** is now available as a ****.NEF*** file. This ****.NEF*** file can be imported into a ***'Standalone programming project'*** or an ***'ISP-PRO programming script'***.

4.6.5 Importing an 'NVR parameter file' back into EDS

If you need to check the parameters and parameter values contained in an ***'NVR Parameter file'*** file, then the simplest way to achieve this is to import the file back into EDS.

To import an ***'NVR Parameter file'*** file back into EDS....

- Select the EDS ***'NVR'*** tab
- Click the ***<Import>*** button
- Select the relevant ***'NVR Parameter file (*.NEF)'*** and click ***<Open>*** to import this file.
- The following screen will now be displayed....



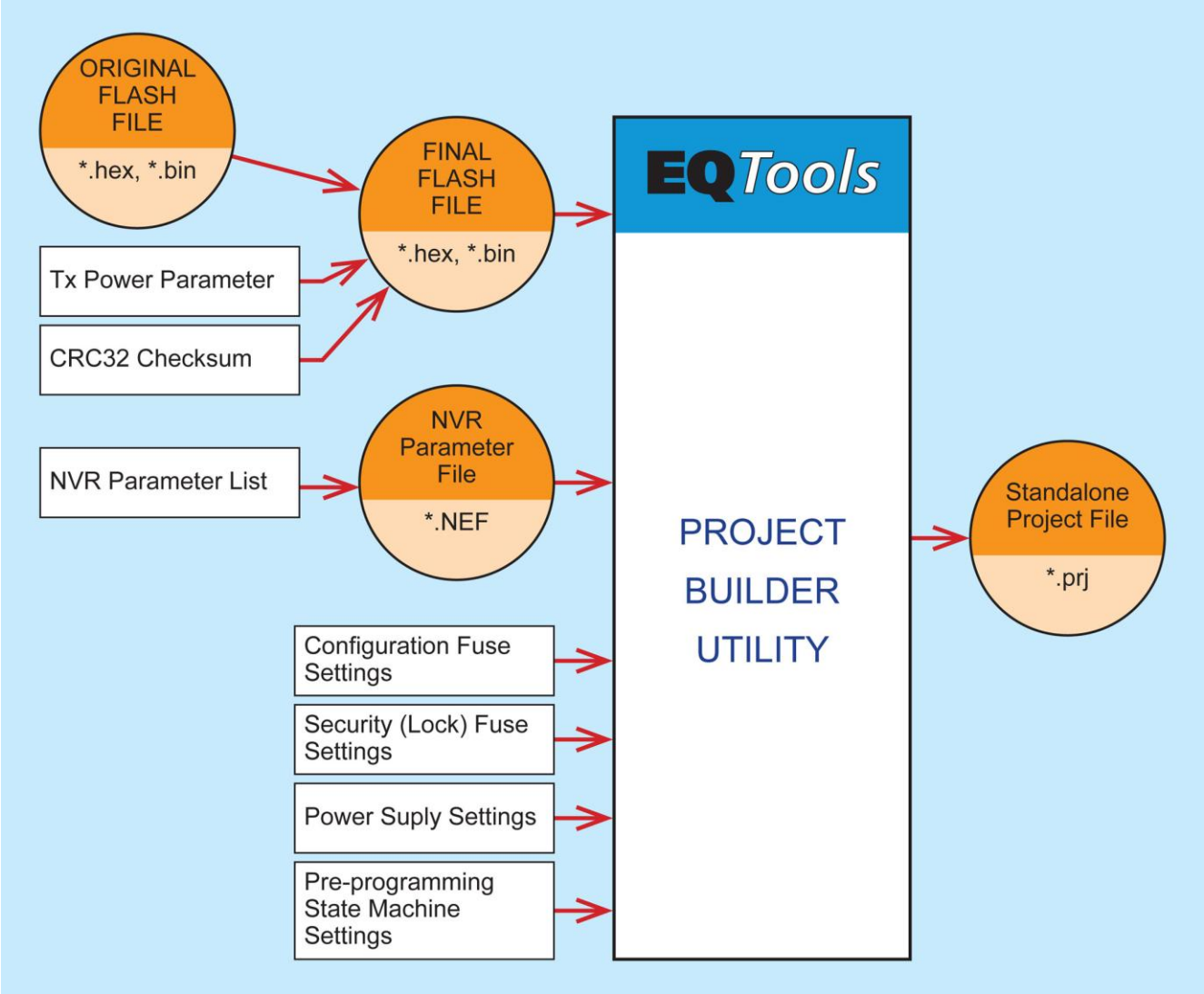
- Click ***<OK>*** to load the parameters / values from the ***'Parameter list'*** into the EDS ***'NVR'*** page.
- The specified parameters / values from the file should now be merged into the existing ***'NVR Area'*** data displayed on the ***'NVR'*** tab.

Appendix 1 – Standalone Programming Mode setup

1.0 Overview

This section describes how to make a ***'Standalone Programming Project'*** for an Equinox programmer. A ***'Standalone Programming Project'*** is a single file (*.prj) which contains all programming actions and user data required to program a target Z-Wave device including the FLASH CODE File, ***'NVR Parameter file'***, ***'Configuration Fuse'*** settings and ***'Security Fuse'*** settings.

A ***'Standalone Programming Project'*** is generated using the ***'EQTools – Project Builder'*** utility. The illustration below shows the process of creating a project.



For most Equinox programmers, it is possible to have up to 64 x ***'Standalone Programming Projects'*** resident in the programmer memory at any point in time. A ***'Standalone Programming Project'*** can be executed by an Equinox programmer without requiring a PC to control the programmer. This makes it possible for the programming sequence to be triggered by an ASCII

command via the RS232 / USB port, via the ***'START'*** button on the programmer, via the 4-wire ***'Remote Control Port'*** or via a ***'Jig switch'*** on the programming fixture lid. As there is no requirement for a PC to control the programmer, this 'Standalone mode' operation is ideal for interfacing the programmer to systems such as ICTs, PLCs etc.

1.1 Information required to create a Standalone Programming Project

The following information is required about your Target Board / application in order to create a ***'Standalone Programming Project'*** for a ***'Z-WAVE 500 series'*** device.....

#	Information / data required	Example
1	Sigma Z-WAVE device part number	SD3502
2	Programming interface	SPI or UART
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	NVR Parameter File	e.g. NVR_parameters.NEF (*.NEF)
8	Tx Power parameters	The ***'Tx Power parameters'*** should be obtained by from RF testing of your final product and then merged into your '***Final FLASH hex file'***.
9	Configuration Fuse settings	See Z-wave datasheet for details (if unsure, leave as default settings)
10	Security Fuse settings	See Z-wave datasheet for details (if unsure, leave as default settings)

1.2 Software versions for Standalone Programming mode

The functionality of being able to program a Z-Wave 500 series device in ***'Standalone Programming Mode'*** requires the following versions of both EQTools and programmer firmware to be installed....

#	Software / firmware	Version required
1	EQTools - programmer configuration software	Version 4 build 3660 or above
2	Programmer firmware version	6.28 or above

Important note:
If you use an earlier version of firmware, the ***'NVR parameter list'*** will not be programmed into the target device. Please ensure you are using the more up-to-date firmware version.

2.0 Explanation of 'standalone mode projects'

A ***'Standalone Programming Project'*** is used for high-speed programming of a Z-Wave device where no unique data per DUT such as a serial number needs to be programmed.

A ***'Standalone Programming Project'*** supports the following functionality:

- Applies power to the DUT
- Enters programming mode
- Checks that the ***'Device ID / Signature'*** of the target device is correct.
- Erases the ***'FLASH area'*** and ***'NVR Area'*** of a Z-Wave device
- Reads back and then validates the existing ***'factory calibration data'*** stored in the ***'NVR Area'*** of the Z-Wave device.
- Validates that the ***'factory calibration data'*** stored in the ***'NVR Area'*** is not corrupted (checks REV and CRC16 checksum)
- Validates that the ***'TX Calibration'*** parameters are within limits (i.e. the device is correctly TX calibrated)
- Validates that the ***'XTAL Calibration'*** parameters are within limits (i.e. the device is correctly XTAL calibrated)
- Validates that the ***'Non-Volatile Memory (NVM)'*** parameters are not erroneous
- Backs up the existing ***'factory calibration data'*** stored in the ***'NVR Area'*** of the Z-Wave device before a ***'Chip Erase'*** operation and automatically restoring this data after the ***'Chip Erase'***.
- Programs fixed value ***'NVR parameters'*** into the ***'NVR Area'*** of a Z-Wave device. e.g. ***'Non-volatile memory (NVM)'*** fixed values.
- Programs a fixed ***'FLASH Firmware'*** hex or binary file into the ***'FLASH area'*** of the Z-Wave device
- Programs the ***'Configuration Fuses'*** of the Z-Wave device
- Checks that the data programmed into the ***'FLASH Area'*** is correct by validating the CRC32 FLASH checksum
- Locks the device (setting the ***'Lock bytes'***)
- Power down / discharge the DUT

A ***'Standalone Programming Project'*** does **NOT** support the following functionality:

- Re-calibrating the ***'TX Calibration'*** parameters if the Z-Wave device is not calibrated correctly.
- Re-calibrating the ***'XTAL Calibration'*** parameters if the Z-Wave device is not calibrated correctly.
- Programming unique data e.g serial numbers, MAC addresses etc into the ***'NVR Area'***
- Recovering the data in the ***'NVR Area'*** if it has been accidentally erased or corrupted.

3.0 Running a 'Standalone Programming Project'

Once a ***'Standalone programming project'*** has been created, it must be uploaded to the programmer using the ***'Upload Wizard'*** utility (part of EQTools) first before it can be executed.

It is then possible to execute a ***'Standalone programming project'*** using any of the following control methods:

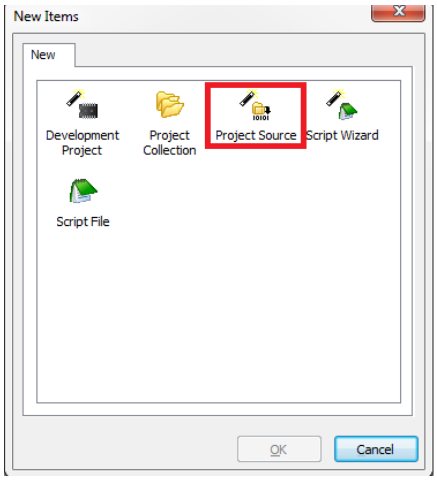
Control method	Control method overview	Number of standalone projects supported
START button	Pressing the ***'START'*** button on the programmer	1
Lid switch	Closing the ***'Fixture Lid Switch'*** contacts on the programmer	1
4-wire TTL port	Remote System e.g. ICT controls programmer execution via 4 x TTL control signals	1
ASCII protocol	Simple ASCII serial protocol	64

4.0 Creating a 'Standalone Programming Project'

4.1 Getting started

To make a 'Standalone programming project' please follow the instructions detailed below...

- Start EQTools
- From the top menu bar, select ***'File - New'***
- Select the ***'Project source'*** icon and click ***<OK>***

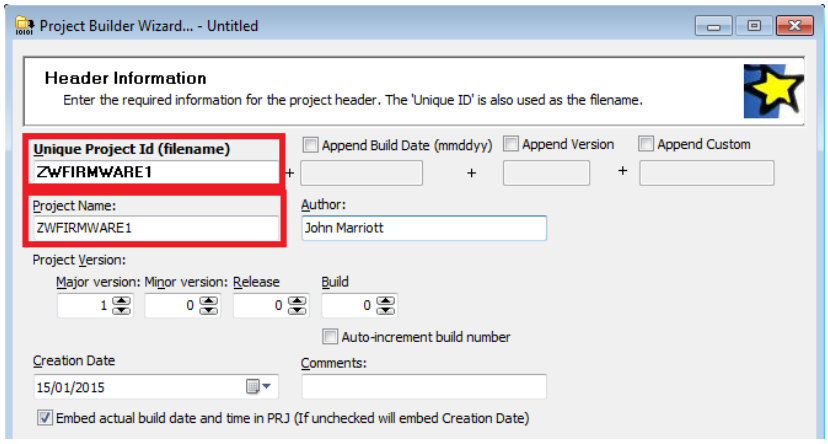


- Select the ***'Project source'*** icon and click <OK>
- The ***'Project Builder Wizard'*** will now be displayed.
- Do NOT change any settings on this screen. Click the ***<Next>*** button --> the ***'Task options'*** screen will be displayed
- Click the ***<Next>*** button --> the ***'Header Information'*** screen will be displayed.

4.2 Header Information screen

The ***'Header Information'*** screen is used to set up the following:

- ***Project name (UniqueID)*** - This is used to reference projects once they have been uploaded to the programmer.
- ***Project File Name*** - This is the actual file name of the project.
- ***Project Version Control information*** - This information allows the project to be given a unique and traceable version control string.



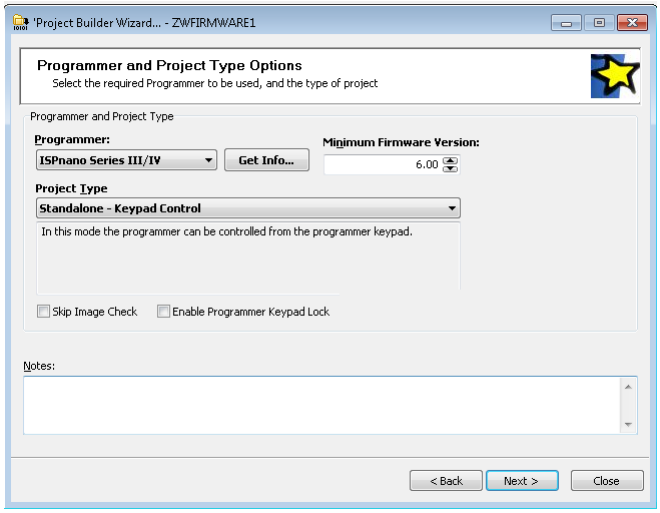
Instructions:

- Type in a suitable ***'Project name / file name'*** in the ***'Unique Project Id (filename)'*** field. e.g. ***'ZWFIRMWARE1'***
- This name will be used for both the project ***'UniqueID'*** reference and also as the ***'Project file name'***.

4.3 Programmer and Project Type options

The ***' Programmer and Project Type options'*** screen is used to set up the following:

- The programmer which the project is to be used with.
- The ***'Project Type'*** which defines how the ***'standalone project'*** will be controlled.



Instructions:

1. Select the ***'Programmer'*** you want the project to work with....
   - Select the correct programmer from the drop-down list e.g. ***ISPnano Series III / IV***.

   or
   - Click the ***<Get info>*** to detect the attached programmer automatically (The programmer must be already connected to the PC COM / USB port).

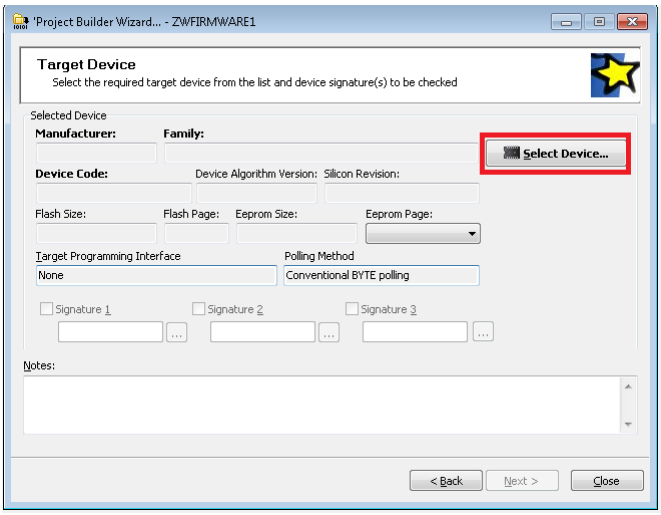
2. Set the ***'Project Type'*** to match your required control method for the project.
   - It is possible to leave the ***'Project Type'*** set to ***'Standalone - keypad control'*** for now.
   - The ***'Project Type'*** setting can be changed globally for all projects via the ***'Global options'*** in EQTools overriding the setting in any individual project.

3. Leave all other settings as the default values

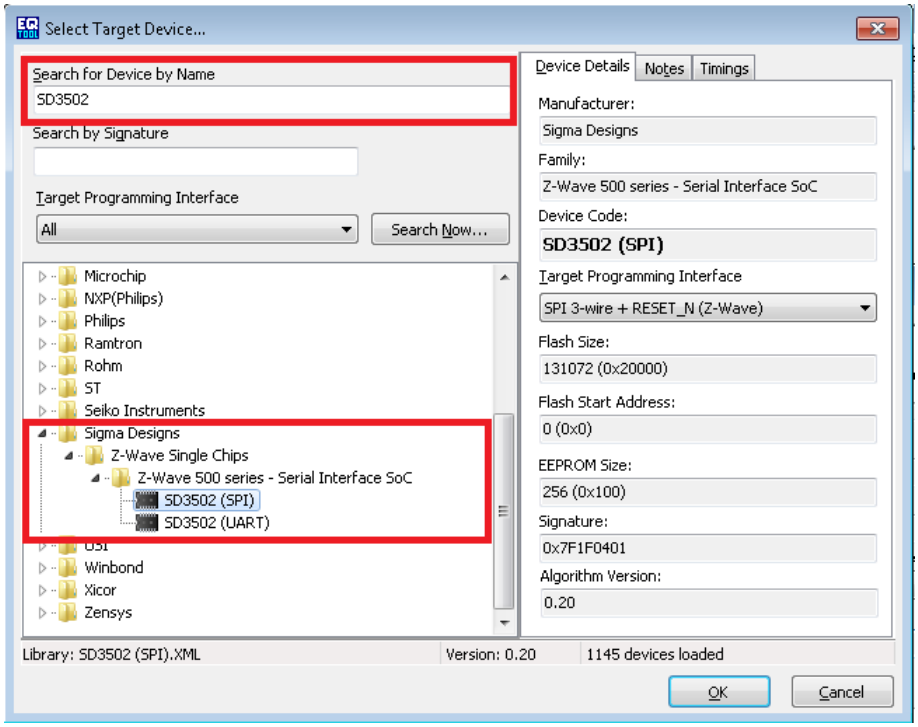
4. Click the ***<Next>*** button to go to the next screen.

4.4 Select the 'Target Device'

The ***'Target Device'*** tab is used to select the device / IC to be programmed and also the ***'programming interface'*** which is being used to connect the programmer to the target device.



Click the ***<Select Device>*** button



To select the device you wish to program:

- Type the ***'Device code'*** e.g. ***SD3502*** into the ***'Search for Device by name'*** field and then click the ***<Search>*** button

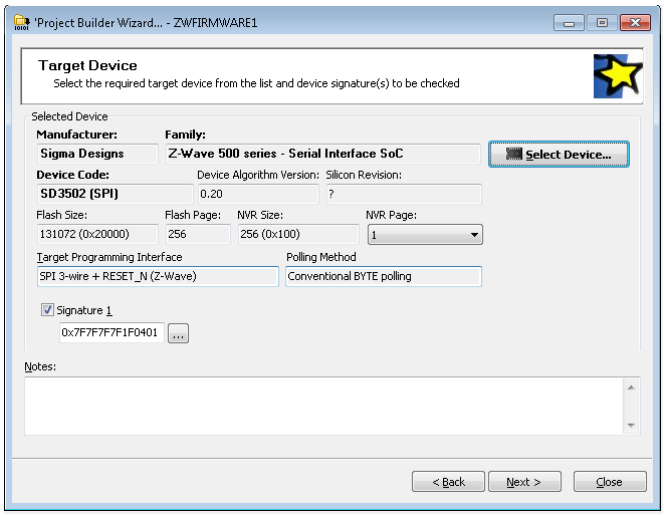
or

- Find ***'Sigma Designs'*** in the tree in the bottom window and then click the triangle symbol to open the tree up to display the ***'Z-Wave 500 series'*** devices.

Select the required device depending on which ***'programming interface'*** is being used:

- **SD3502 (SPI)** - SPI Interface (MOSI, MISO, SCK, RESET_N)
- **SD3502 (UART)** - UART Interface (TXD, RXD, RESET_N)

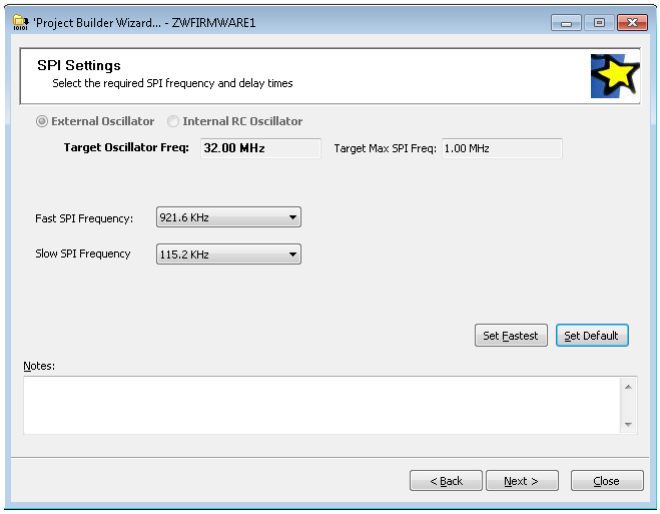
The ***'Target Device'*** tab should now display the selected device and ***'programming interface'*** .....



Click the ***<Next>*** button to go to the next screen.

4.5 Select the 'SPI frequency' for programming

The ***'SPI Settings'*** tab is used to set up the 'SPI frequency' which is used by the programmer to communicate with the target Z-Wave device.



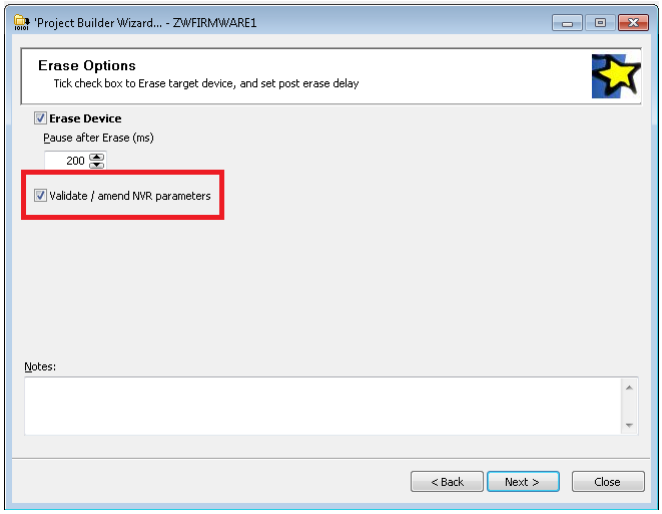
- It is best to leave both the ***'Slow'*** and ***'Fast'*** SPI frequencies at the default settings at this stage.
- If you encounter reliability problems during programming, then try reducing the ***'Fast SPI Frequency'***.
- Click the ***<Next>*** button to go to the next screen.

4.6 Setting up the 'Chip Erase' / 'Amend NVR parameters' options

The ***'ERASE'*** tab is used to configure the following options:

1. Erase Device
When enabled, the programmer will perform a ***'Chip Erase'*** option which will erase both the FLASH and NVR areas

2. Validate / amend NVR parameters
When enabled, the programmer will validate / check the values of the main ***'NVR parameters'*** and it will also merge any customer specific ***'NVR parameters'*** from an ***'NVR parameter file (*.NEF)'***.



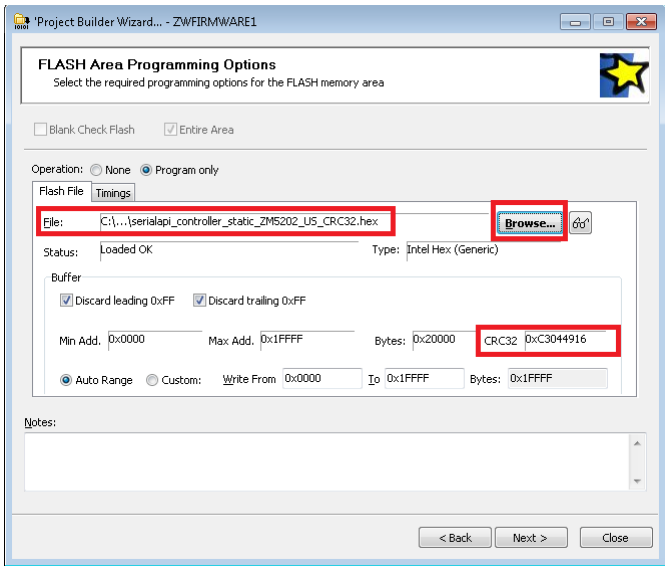
Instructions for ***'standalone programming projects'....***

1. The ***'Erase device'*** option must always be enabled.

2. The ***'Validate / amend NVR parameters'*** option should always be enabled
The programmer will then automatically backup the ***'NVR area'*** before the ***'Chip Erase'***, validate all required factory calibration parameters, merge the ***'Custom NVR parameters'*** from a ***'NVR Parameter file'*** and finally re-program the merged ***'NVR data'*** back into the ***'NVR Area'*** after the ***'Chip Erase'***.

4.7 FLASH Area - programming options

The ***'FLASH Area Programming options'*** tab allows you to specify the 'FLASH firmware' file which is to be programmed into the 'FLASH Area' of the target device.

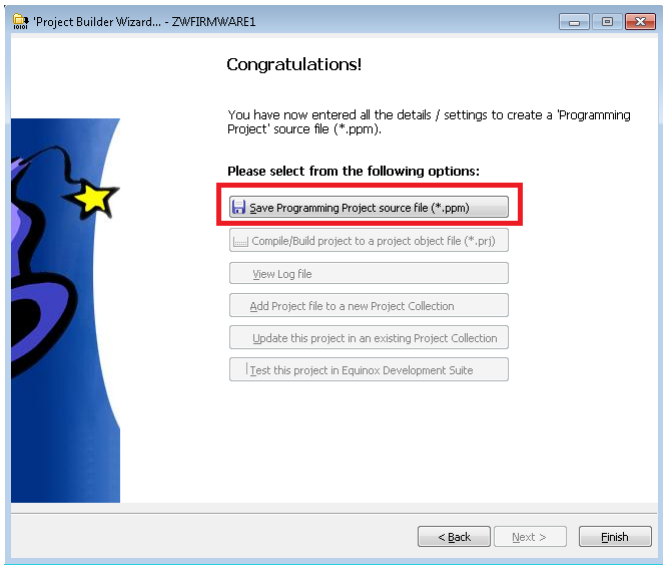


Instructions:

- Check that the ***'Flash file'*** which you plan to load already has the correct ***'CRC32 Checksum'*** set up in the file.
- Check that the ***'Flash file'*** which you plan to load already has the correct ***'Tx Power parameters'*** set up in the file.
- Click the ***<Browse>*** button
- Browse to and select the file which you wish to program into the ***'FLASH area'*** of the device.
- Click ***<OK>*** to select the file
- The ***'File Preview'*** window will now be displayed.
- If a warning about ***'Invalid CRC32 checksum'*** is shown, please consult the relevant section in this document to fix this problem.
- If a warning about ***'Invalid Tx Power parameters'*** is shown, please consult the relevant section in this document to fix this problem.
- Click the ***<Next>*** button to go to the next screen.

4.8 Final 'Congratulations' screen - saving your project

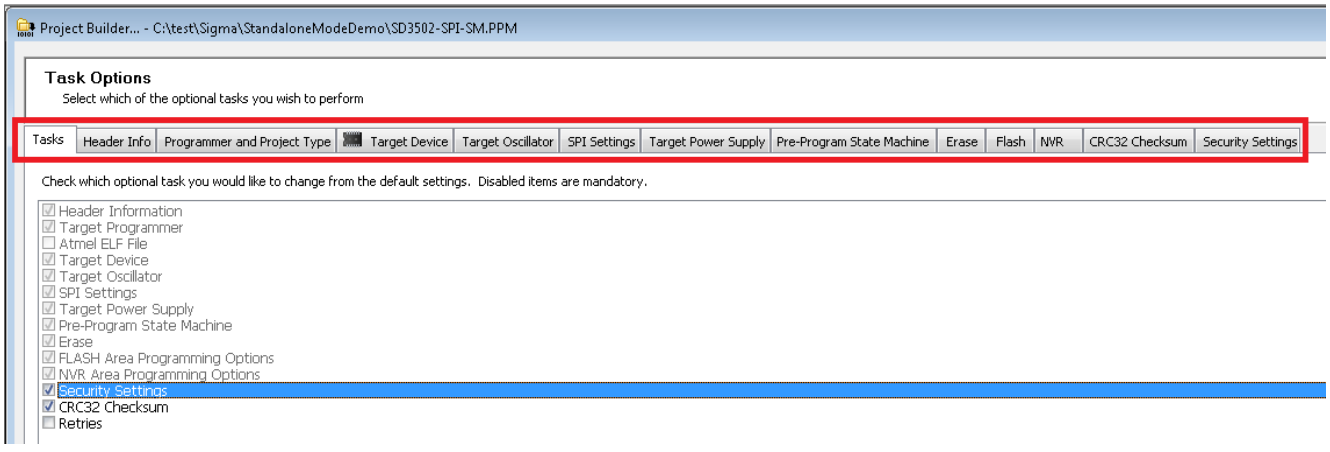
The ***'Congratulations'*** tab allows you to save your project and even change the ***'File name / Unique ID'*** if you wish.



Instructions:

- Click the ***<Finish>*** button to save the project
- You will now be prompted for a file name. The file name you chose at the start of the Wizard will be automatically used unless you choose to change it here.
- Click ***<Save>*** to finally save the project.

The project is now displayed in the ***'Project Builder'*** view as shown below....



- A series of 'tabs' is displayed along the top of the screen.
- Select the ***'Tasks'*** tab
- Make sure the ***'Security Settings'*** and ***'CRC32 Checksum'*** tasks are ticked (enabled)

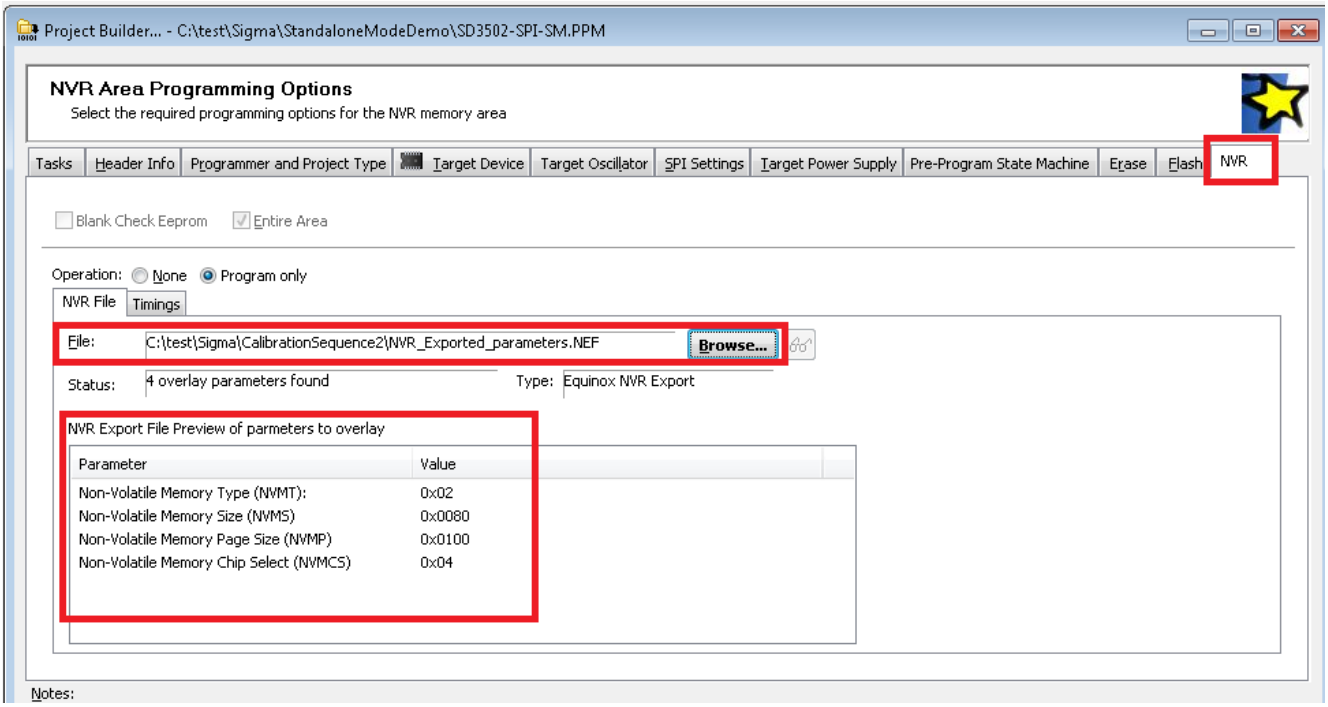
4.9 Importing an 'NVR parameter file' into a standalone project

Once the ***'standalone programming project'*** has been set up, it is then possible to import an ***'NVR Parameter file'*** which will merge customer specified ***'NVR parameter'*** values with the ***'NVR factory data'*** already resident in the ***'NVR Area'*** of the target Z-Wave device.

Please note:
You will need to create an ***'NVR Parameter file (*.NEF)*** using the EQTools - EDS utility to create this file - see section 4.6 of the main application note for further instructions.

Instructions:

- Select the ***'NVR'*** tab of your project
  Click the ***<Browse>*** button
- Browse to and select the ***'NVR Parameter file (*.NEF)'*** which contains the list of 'NVR parameters' which you want to merge into the 'NVR Area data' already resident in the Z-Wave device.
- Click ***<OK>*** to load this file
- A list of the ***'NVR parameters'*** and associated ***'Fixed values'*** should now be displayed.....



- Check that the imported list of ***'NVR parameters'*** is correct.
- EQTools will automatically store all of these parameters in the ***'Standalone programming project'***.

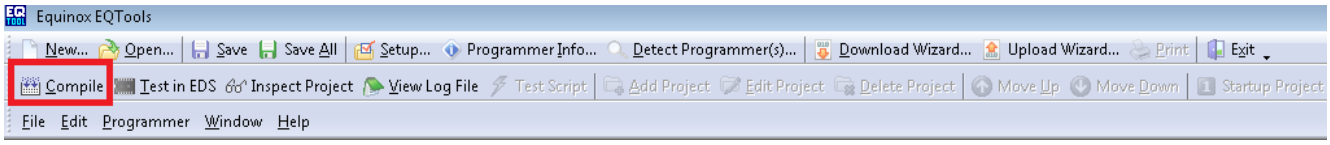
Important note....
When executing the ***'standalone programming project'***, the programmer will automatically merge the ***'NVR parameters'*** specified in the ***'NVR Parameter file (*.NEF)'*** with the ***'Sigma factory programmed data'*** which is already resident in the ***'NVR Area'***. The programmer will NOT change any other parameter values in the ***'NVR Area'***.

4.10 Compiling your standalone project

Once you are happy that all the various settings in your ***'Standalone programming project'*** are correct, it is necessary to compile all these settings into the final ***'Standalone Programming Project'*** file.

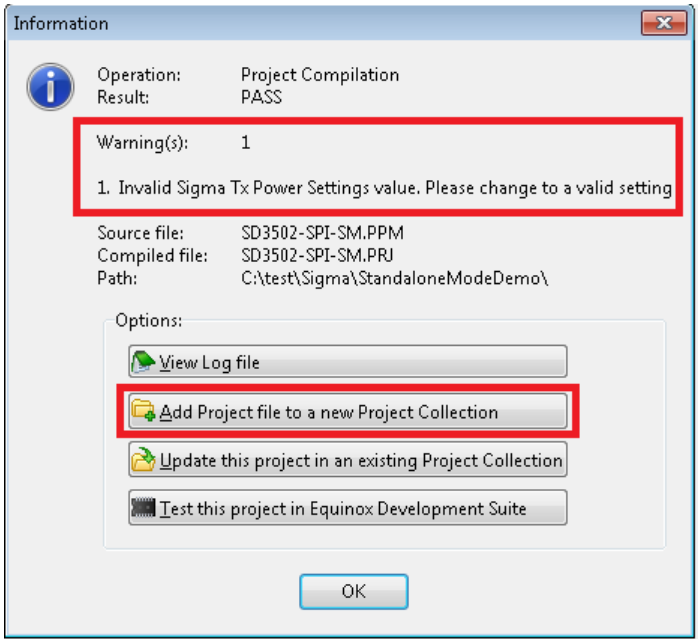
Instructions:

- Make sure you have your ***project file (*.ppm)*** open in EQTools.
- Check that all settings in the project are correct.
- On the top EQTools menu bar, click the ***'Compile'*** icon....



Your project will now be compiled to produce a project file *.prj e.g. ***SD3502-SPI-SM.ppm***.

The following 'Information' screen is now displayed.....



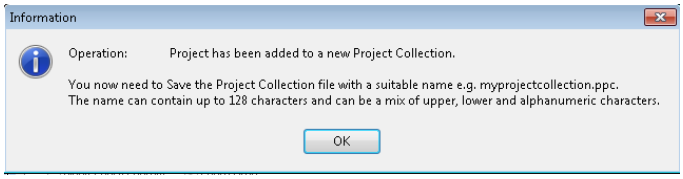
Important note:
The warning shown means that the ***'Tx Power settings'*** have not been set up in the ***'FLASH File'***. To correct this problem, use the ***EQTools – EDS*** utility to re-save your 'FLASH File' with valid ***'Tx Power settings'***.

4.11 Adding your standalone project to a Project Collection

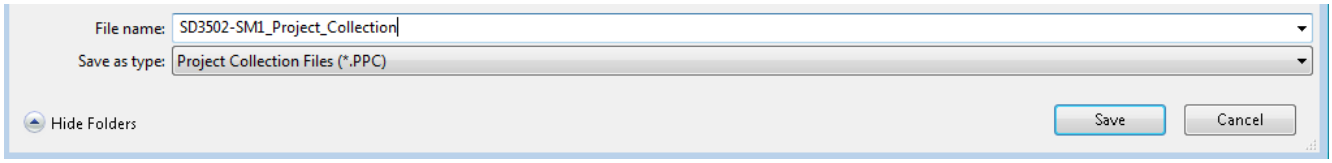
The compiled ***'Project File *.prj'*** must now be added to a ***'Project Collection File (*.ppc)***' so that it can be uploaded to a PC.

To add your project to a new ***'Project Collection'***.....

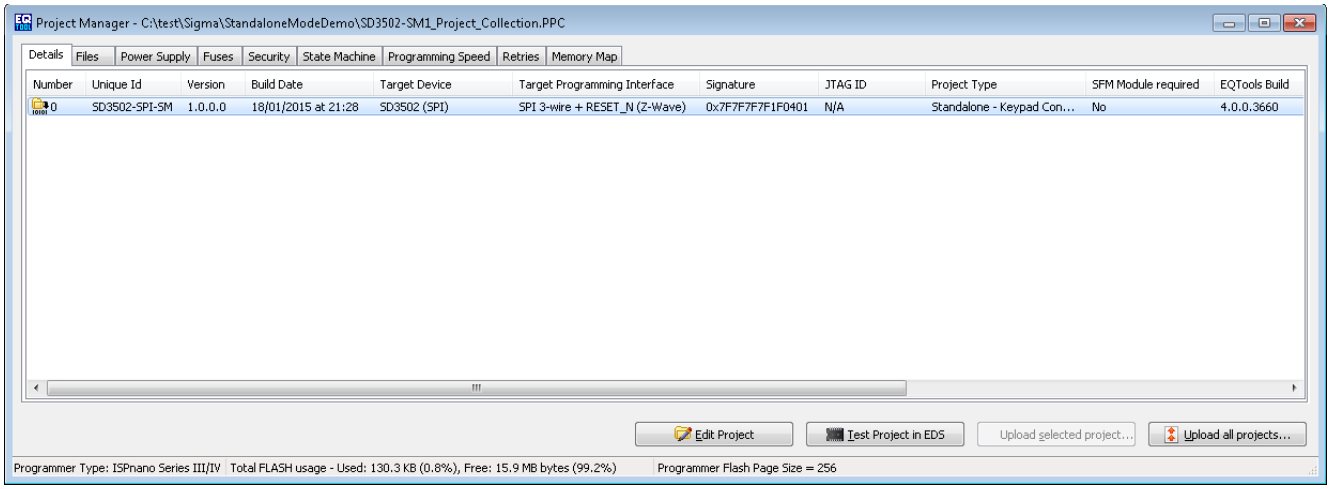
- Click the ***'Add Project file to a new Project Collection'*** button



- Click ***<OK>***
- Enter a suitable name for your new ***'Project Collection'***.....



- Click the ***<Save>*** button
- Your project will now be added to the new Project Collection and the ***'Project Manager'*** window will now be displayed....



4.12 Uploading your standalone project to a programmer

To upload the ***'Project Collection'*** to the attached programmer(s).....

- Click the ***'Upload all projects....'*** button
- The ***'Upload Wizard'*** utility will now start.
- Follow the on-screen instructions to upload the project(s).

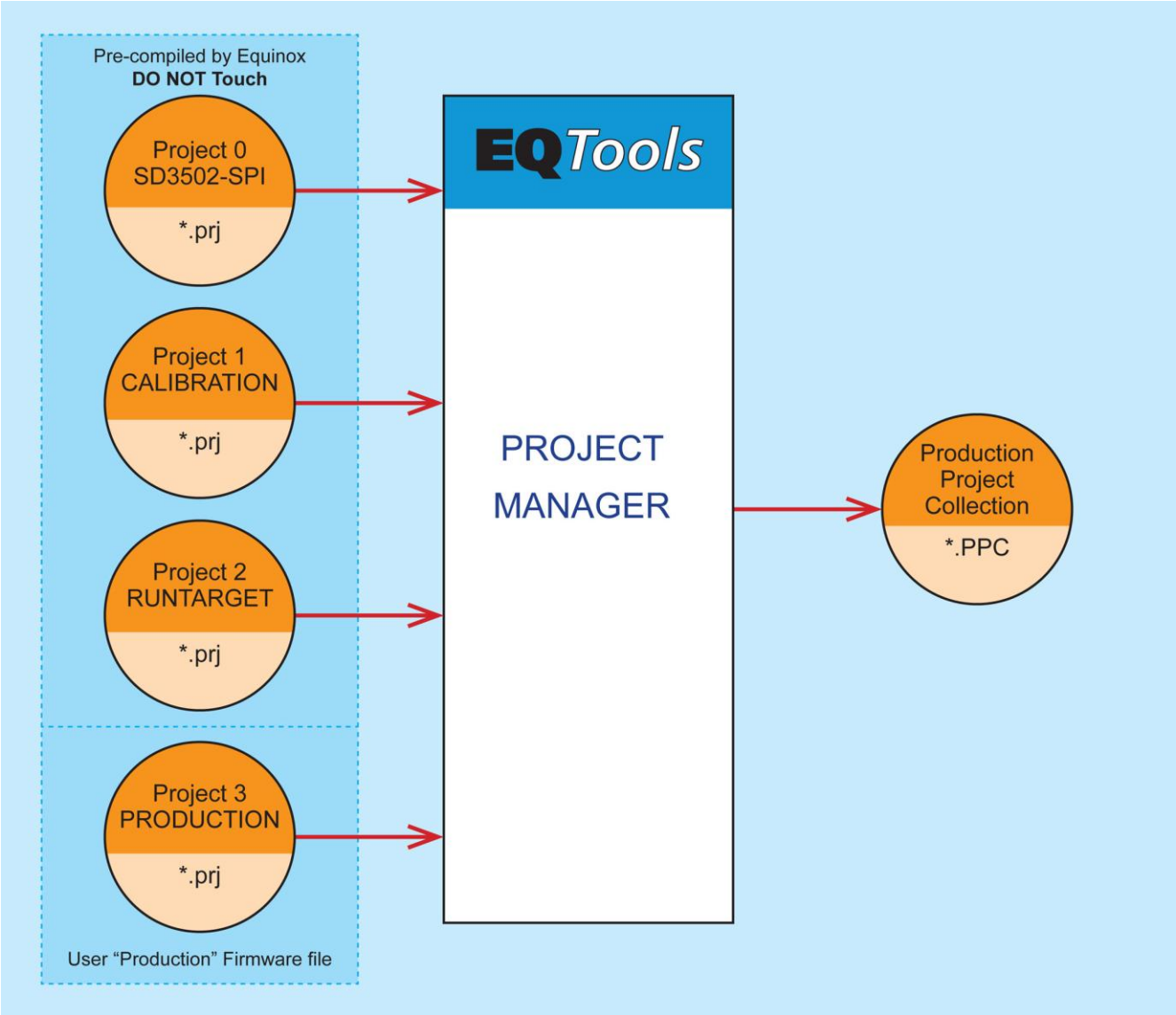
Appendix 2 - ISP-PRO script - how to set up the PRODUCTION standalone project(s)

1.0 Overview

This section describes how to set up the 'standalone programming projects' for use with the Equinox ISP-PRO software. These instructions are only for updating the project collection used with ISP-PRO and an ISP-PRO programming / calibration sequence script for Z-Wave 500 series devices.

2.0 Explanation of 'standalone projects'

The programming / calibration sequence for a ***'Sigma Z-Wave 500 series device'*** when executed within the Equinox ISP-PRO software, is made up of 4 x ***'Standalone Programming Projects'*** as shown in the illustration below....



The projects within this 'Project Collection' are described below....

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.
***** Do NOT edit this project - this is for Equinox use only. *****

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. This project has been pre-prepared by Equinox and you should not need to change this project.
***** Do NOT edit this project - this is for Equinox use only. *****

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. This project has been pre-prepared by Equinox and you should not need to change this project.
***** Do NOT edit this project - this is for Equinox use only. *****

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:

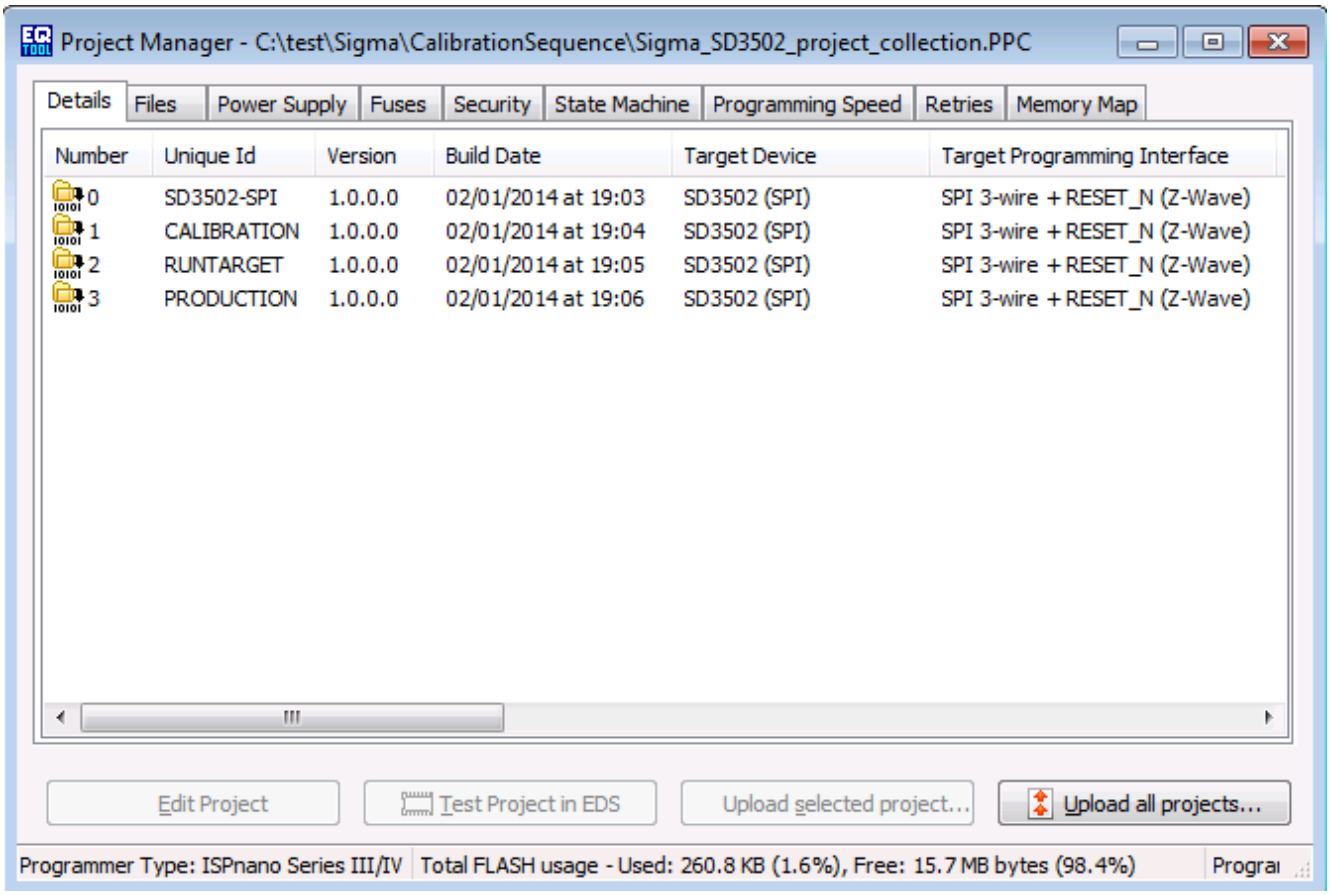
For most applications, the only project which needs to be modified is the ***'PRODUCTION'*** project.
This project programs the ***'Customer firmware file'*** and so will probably be unique to each customer application. The next sections explain how to change the ***'Customer firmware file'*** in the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



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.

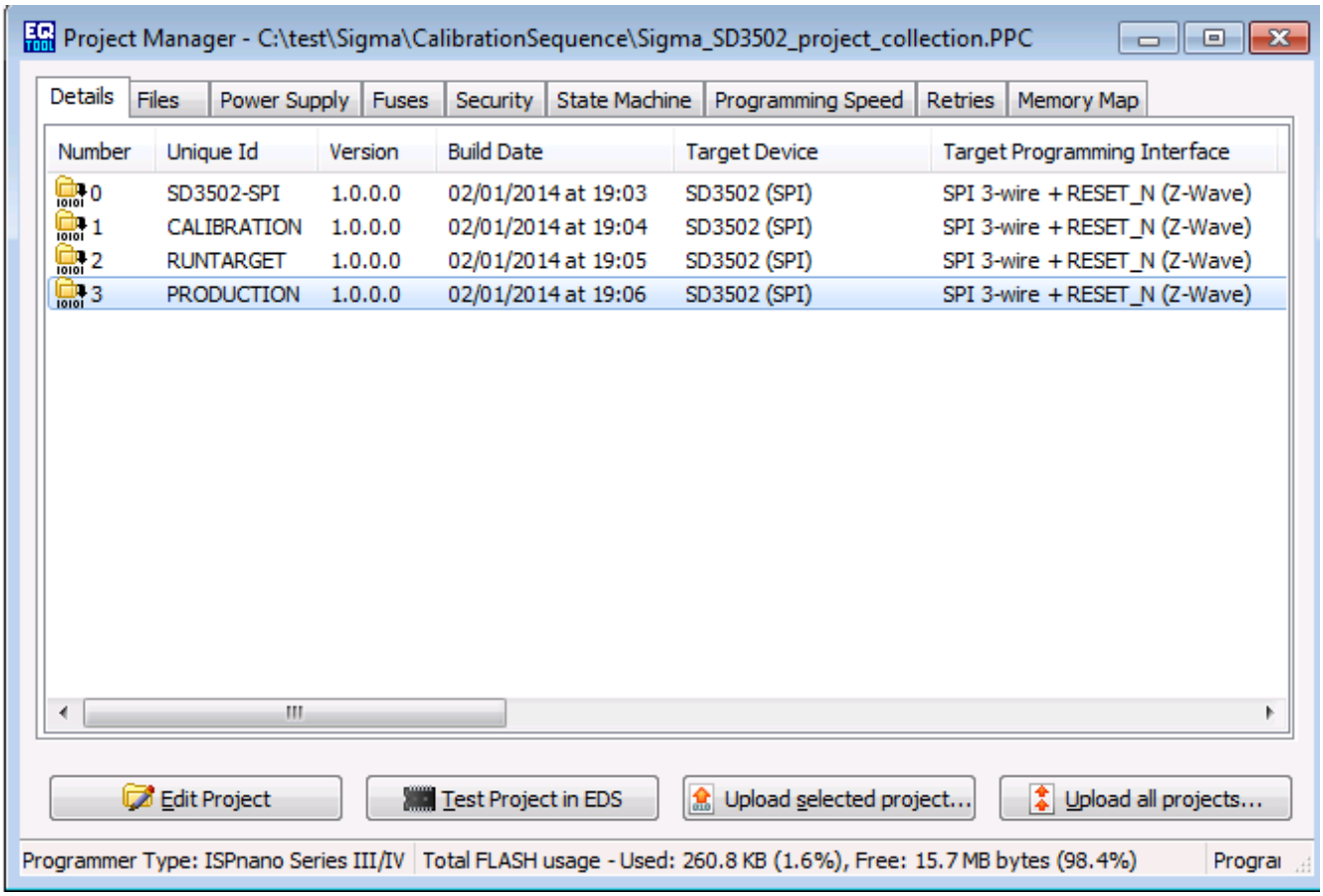
2.2 Changing the 'Product firmware' file

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

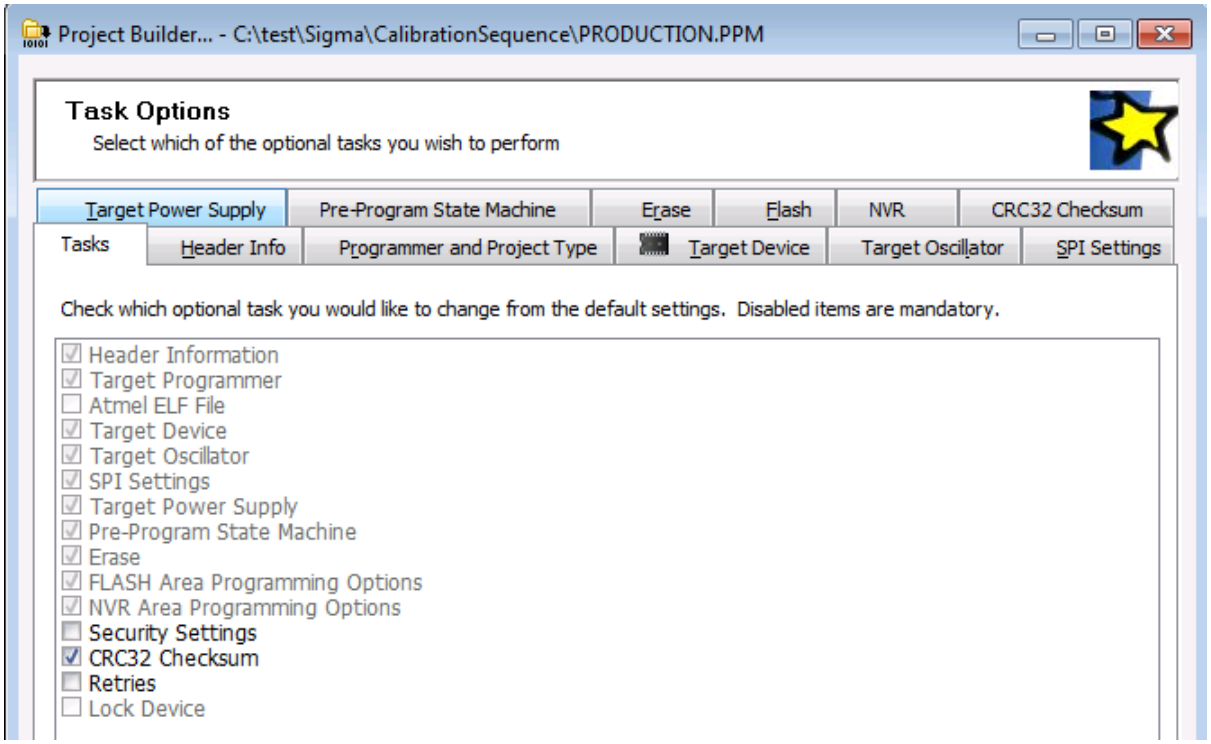
2. Highlight the ***'PRODUCTION'*** project by clicking on the project name once
--> see screenshot below....



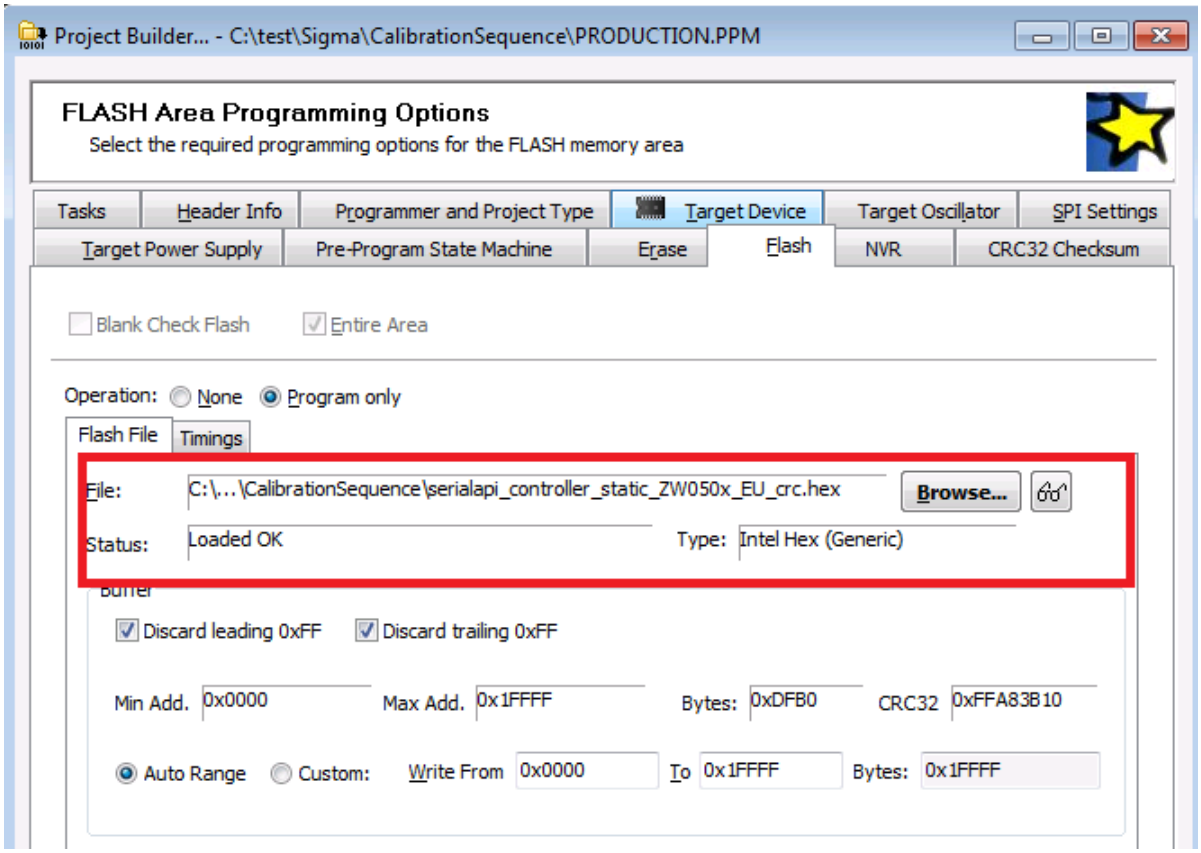
3. To open the ***'PRODUCTION'*** project
- Click the ***'Edit Project'*** button or double-click the ***'PRODUCTION'*** project name in the list.
--> The ***'PRODUCTION'*** project will now open in ***'Project Builder'*** view.

4. Overview of the 'Task options' tab

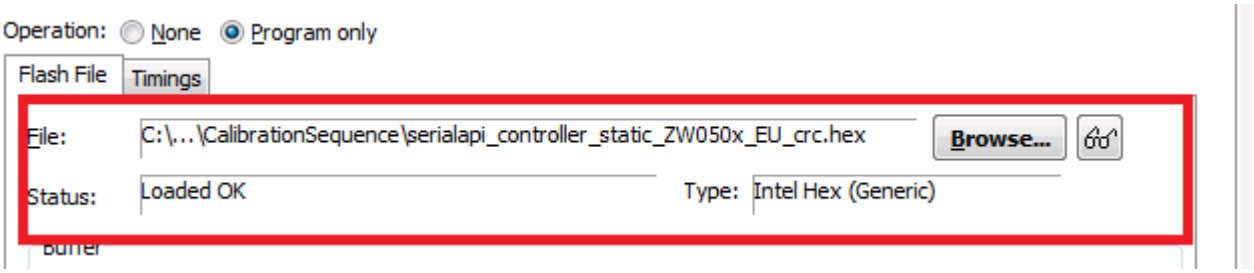
The project will open on the ***'Tasks'*** tab which shows the various ***'Tasks'*** which can be performed by a ***'standalone project'***....



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

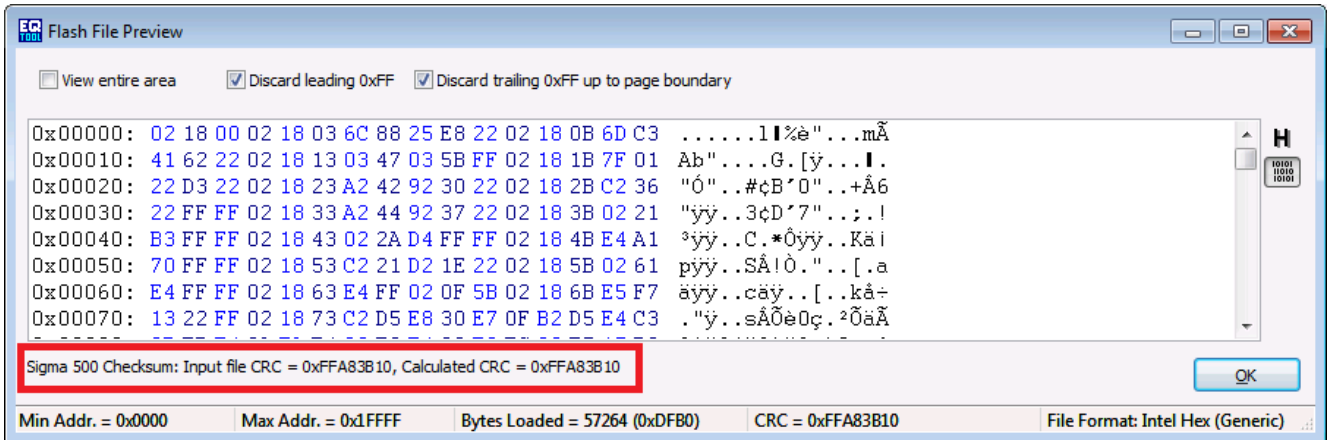


The project is currently setup with an example ***'firmware file'*** from Sigma as follows.....



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.....



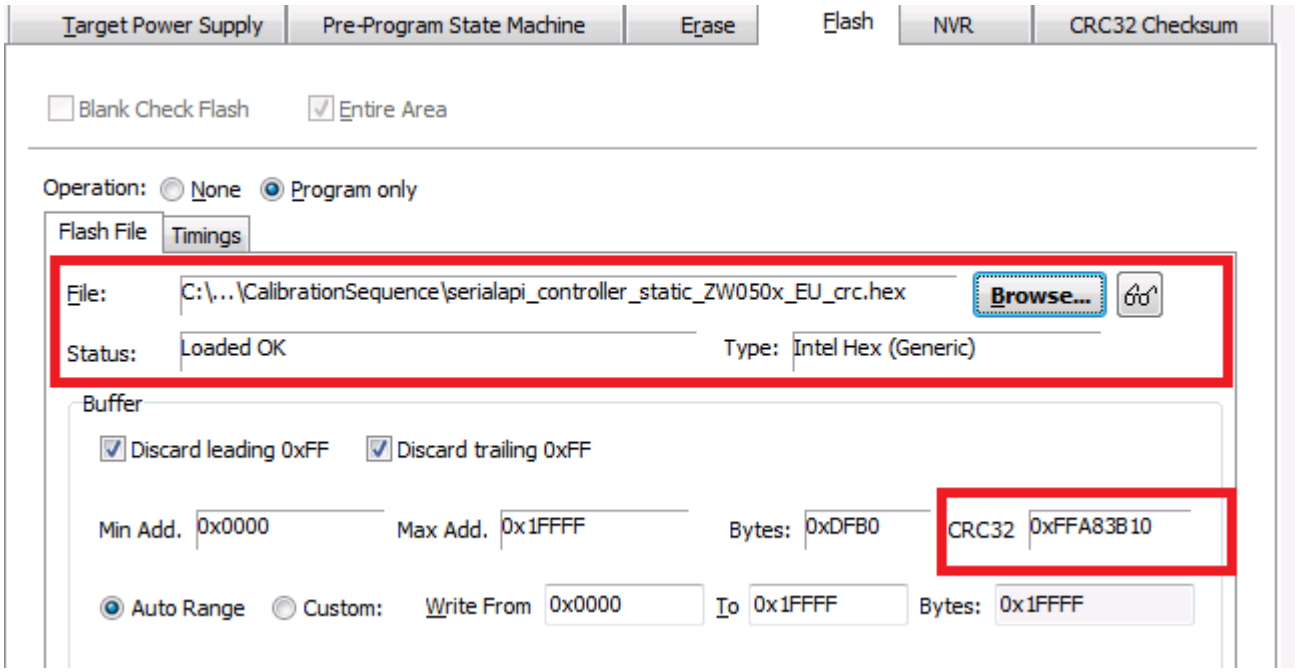
This window displays the following information about the selected input file....

- 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.

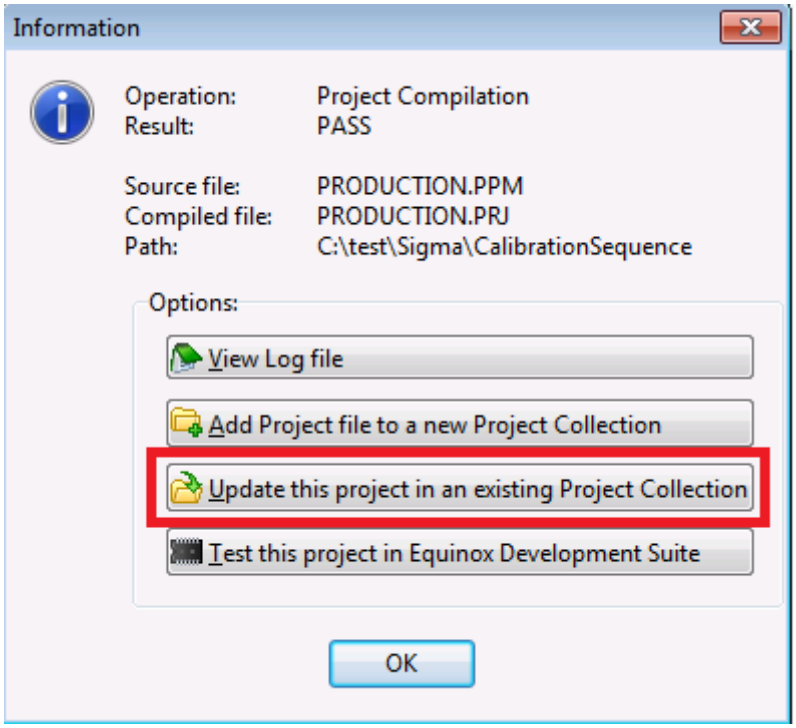
7. The selected ***'Firmware file'*** should now be displayed.
The ***'CRC32 FLASH Checksum'*** calculated by EQTools when the file was loaded is also displayed.
See screenshot below....



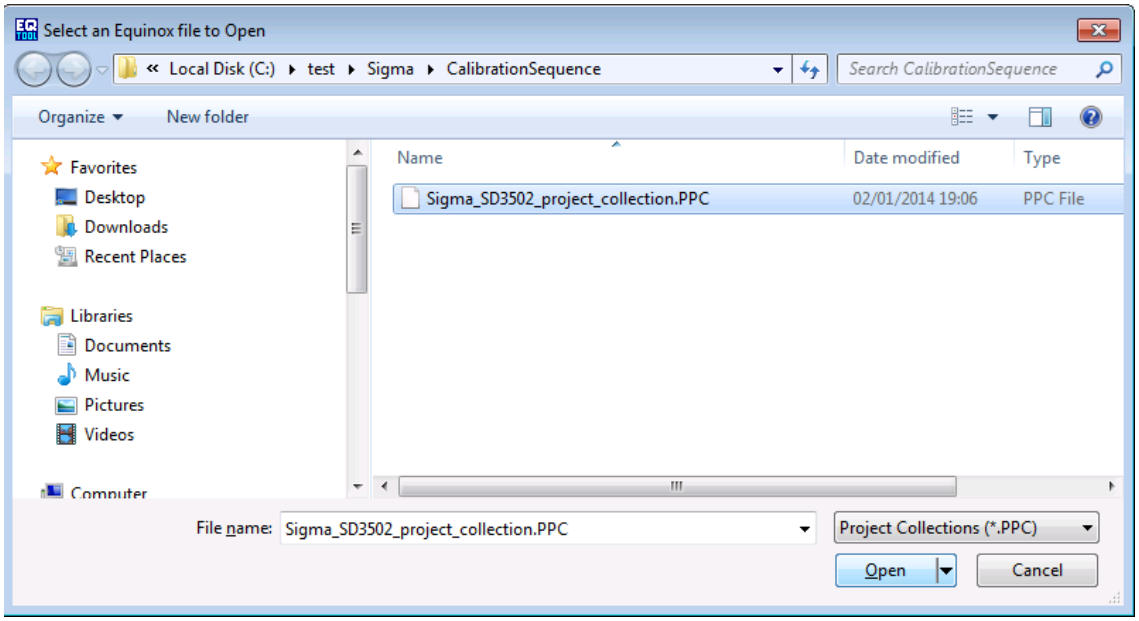
8. Compile the project

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

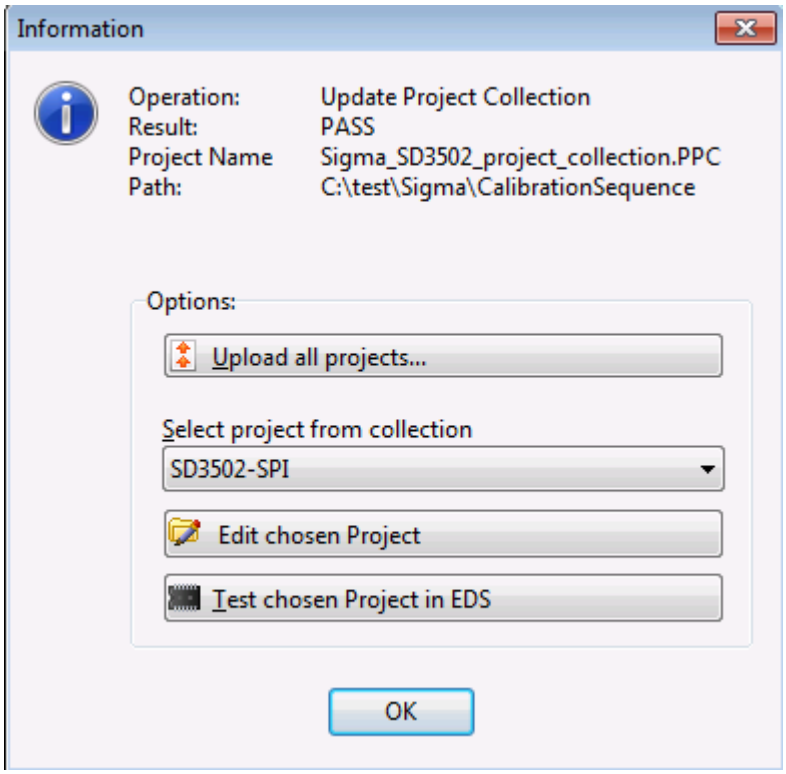
On the top EQTools icon bar, click the ***'Compile'*** icon.
This will compile the project and then display the following Info screen....



- 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...

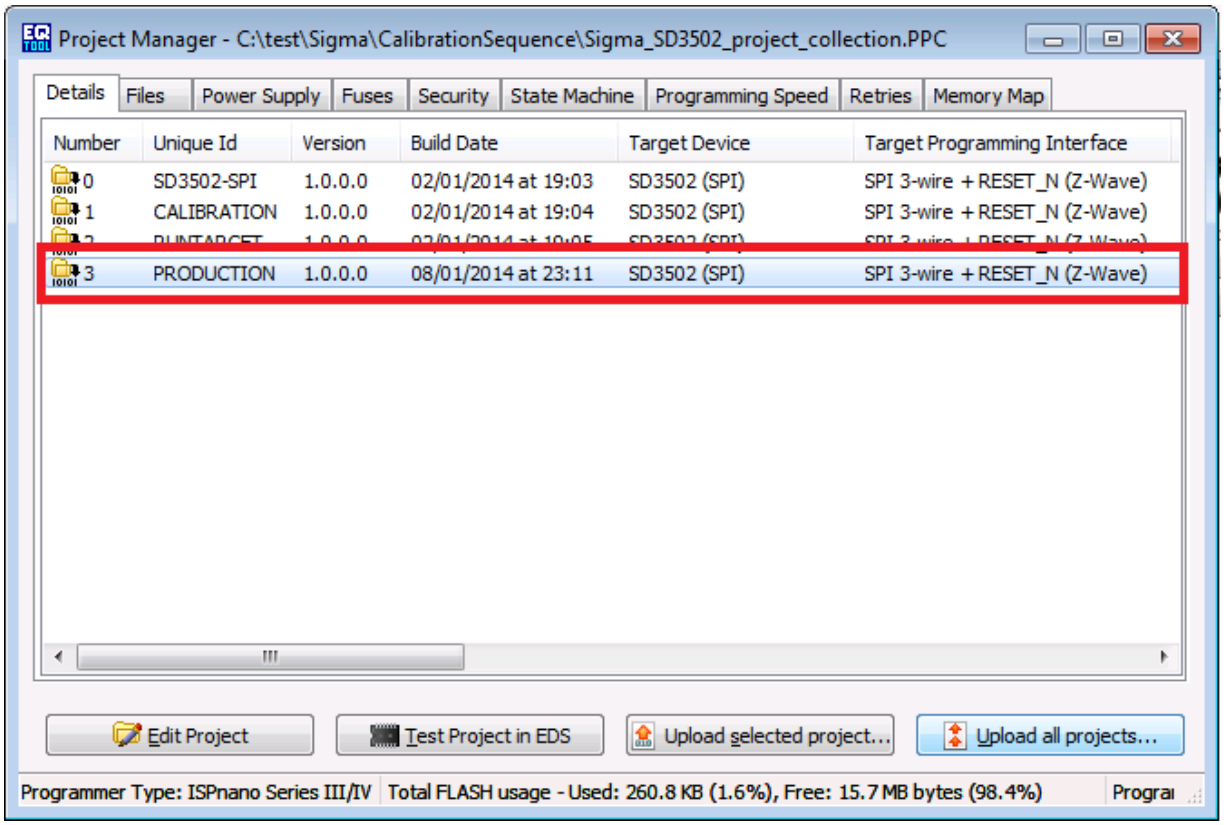


The ***'Update Project Collection'*** process should now report ***'PASS'***.....



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.



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.

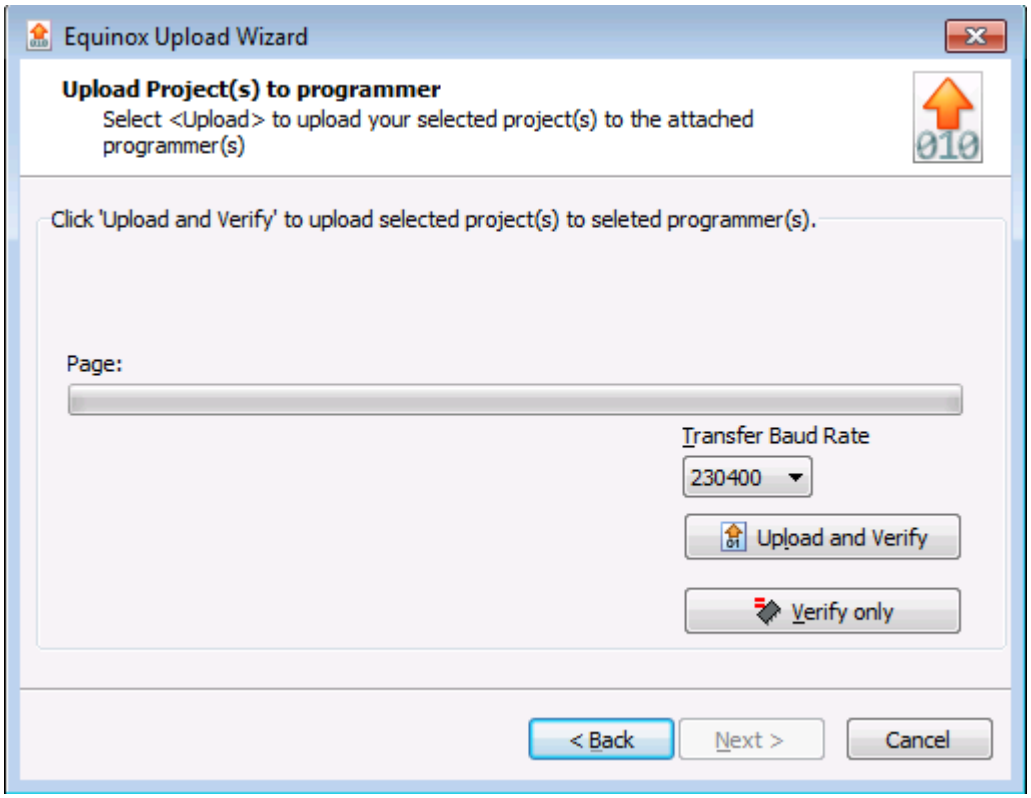
To upload the ***'Project Collection'*** to the programmer....

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.....



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 3 - 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.

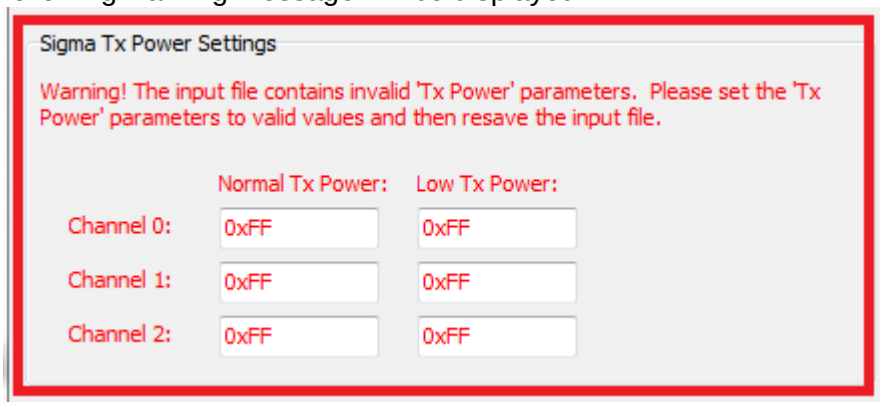
The values for the ***'Tx Power'*** parameters must be derived by experimentation during the development and final R&D testing stages of the customer product. The values are then usually fixed for this product and must be programmed into certain specific locations in the ***'FLASH area'*** of the Z-Wave 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.

1.1 Why do I get a 'TX Power parameter' error?

If you try to load a HEX or BINARY file into the FLASH buffer in the EQTools software, then the file loader automatically checks the input file to make sure that the ***'Tx Power Parameters'*** have been configured in the file.

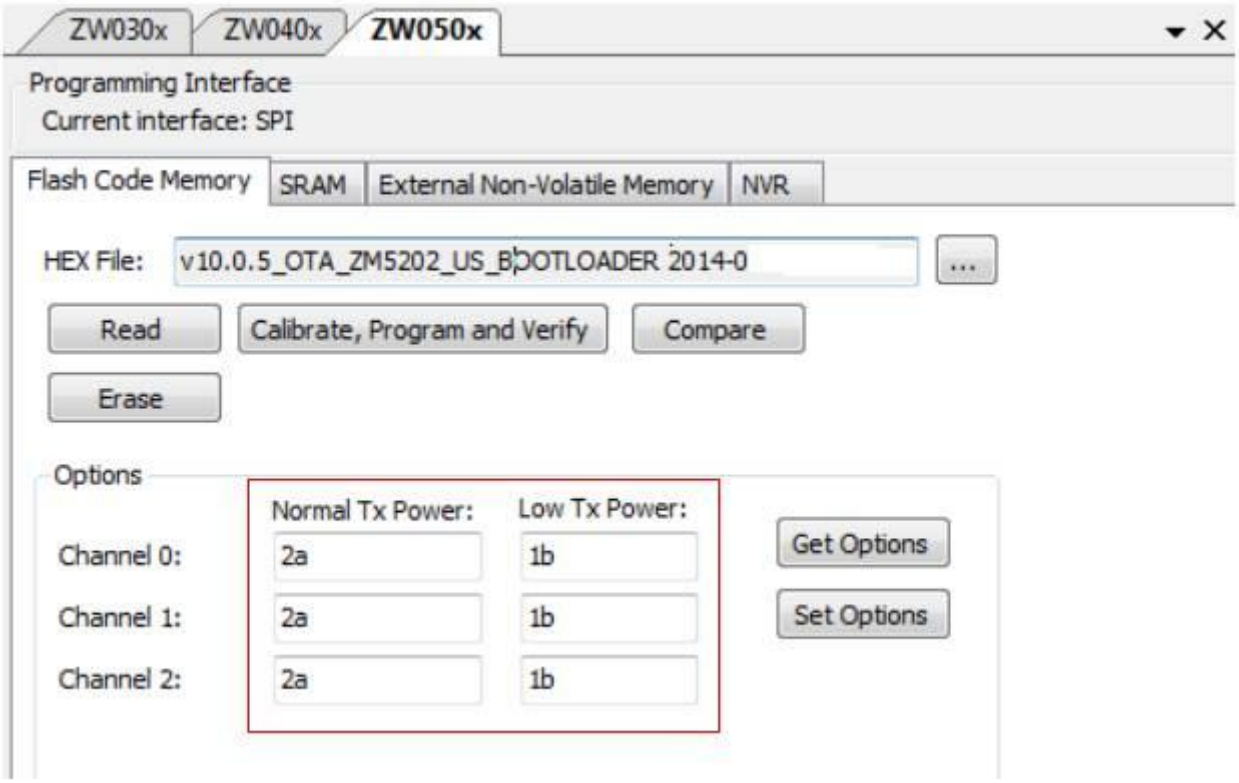
If EQTools detects that the ***'Tx Power Parameters'*** are all set to 0xFF in the input file, then the following warning message will be displayed.....



This error message means that the ***'Tx Power Parameters'*** are invalid. It is therefore necessary to enter valid values for these parameters which match the required power output for your target product. Please refer to the instructions in the next section for further instructions on how to configure the ***'Tx Power Parameters'***.

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…..



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 (supplied by Sigma) for these settings. However, the values used should always be double-checked either with your ‘RF engineer’ or with Sigma Designs technical support service.

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 including the ***'Tx Power'*** parameters.

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

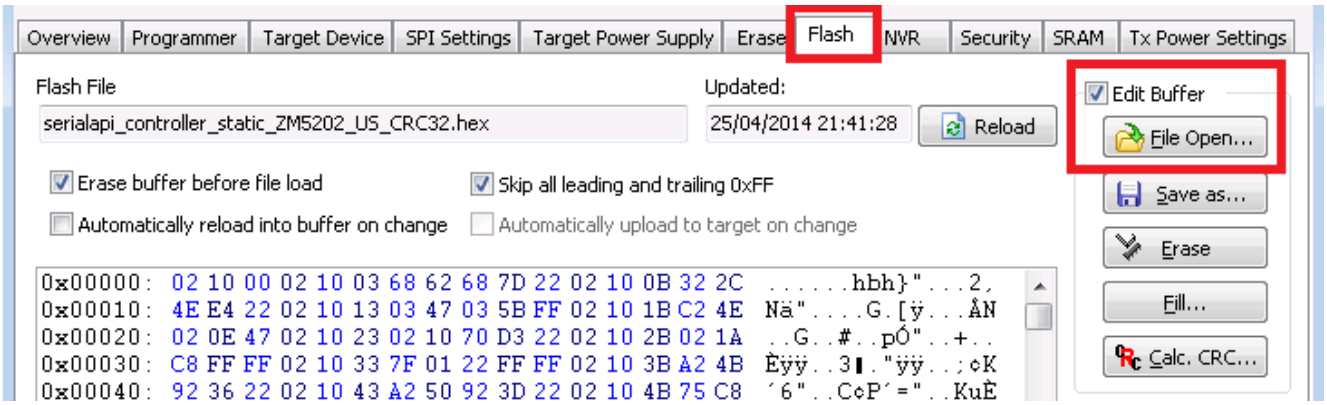
The ***'Tx Power'*** parameters for your product must be merged into the ***'Production FLASH hex file'*** It is possible to perform this task using the EQTools – EDS (Development Mode) utility.

Instructions:

1. Start the Equinox – EQTools software

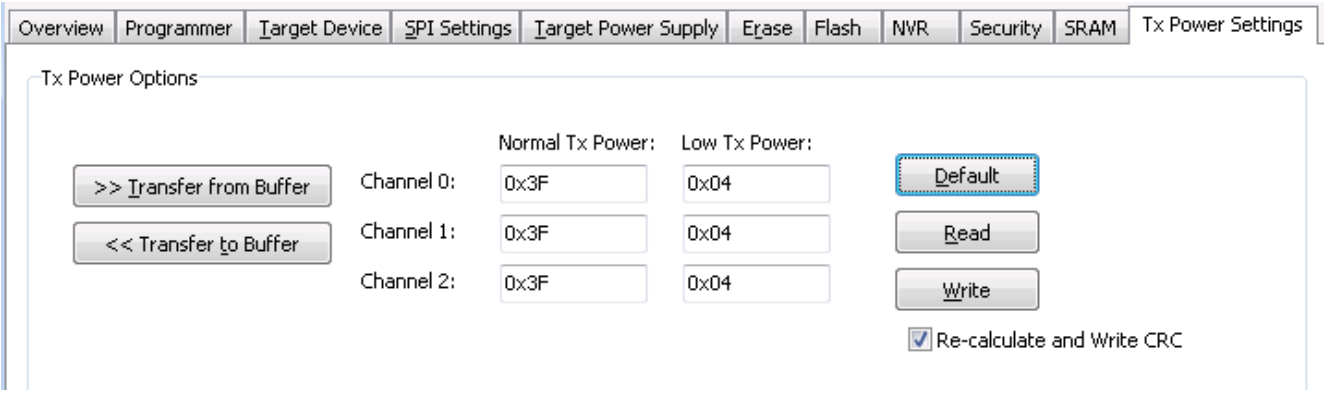
2. Select the option to ***'Create an EDS - Development Project'*** and follow the wizard to create the EDS project
Or
Open an existing Sigma 500 series EDS project (*.eds)

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



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.

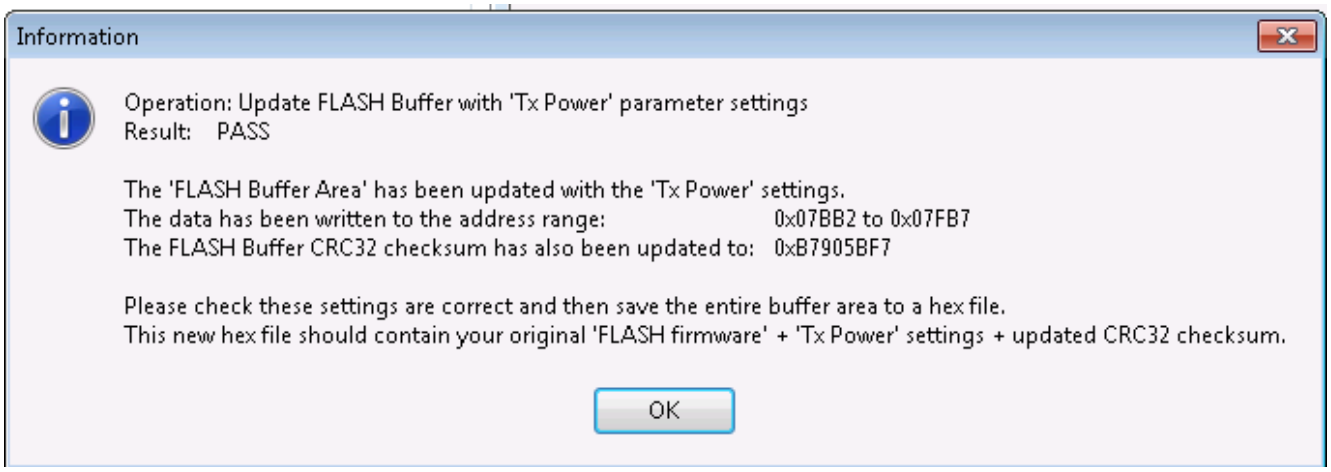
5. Enter the required ***'Tx Power Options'*** for your product as follows…
   - Select the ***'Tx Power Options'*** tab → the following screen should be displayed…



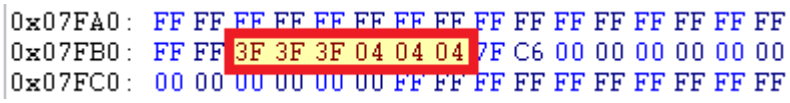
   - Enter the correct values for the ***'Tx Power'*** parameters for your product in the relevant fields

- 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



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.



- 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 4 - 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.

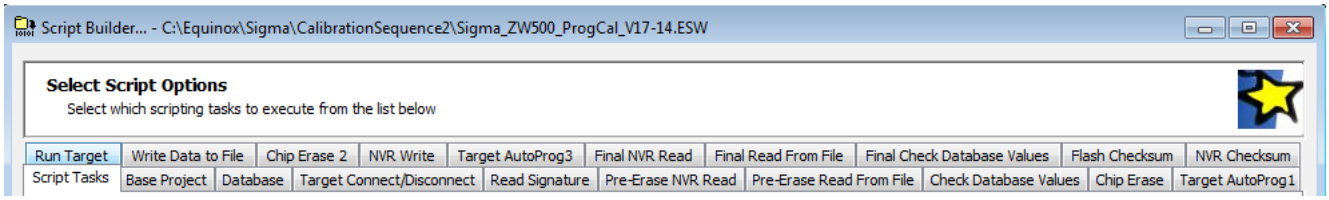
However, if you do not know the relevant values for the ***'External Non-volatile memory (NVM)'*** device parameters, please make a note of the part number of the memory device fitted on your target board and then contact Sigma technical support quoting the memory device part number. They should hopefully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

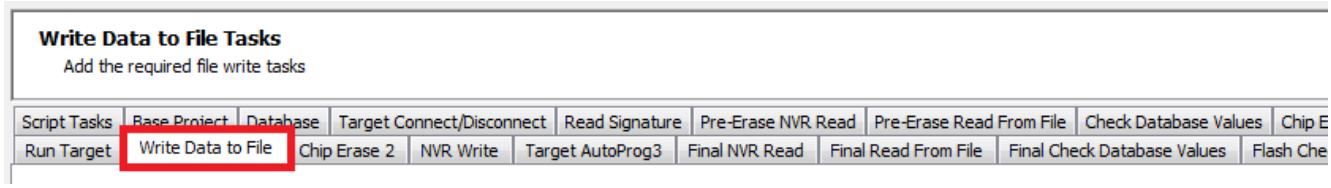
If you need to program custom values for the ***'External Non-volatile memory (NVM)'*** device parameters, you will need to amend the ***'Sigma programming script'***.

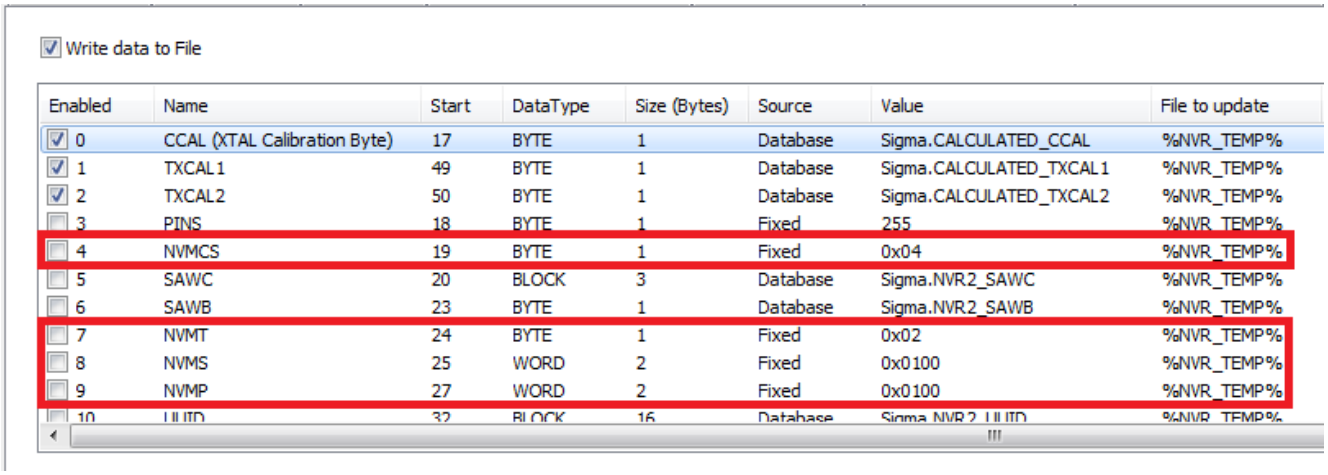
Instructions:

- 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.....



- 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....





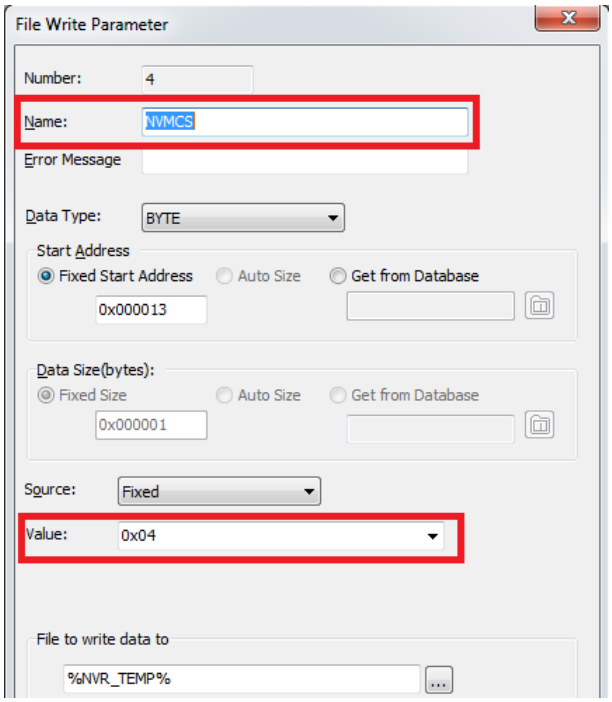
- The ***'NVR parameters'*** which are used to set up the Z-Wave external ***'Non-volatile memory (NVM)'*** are highlighted in the list in red: ***NVMCS, NVMT, NVMS, NVMP***
- By default, the programming script will simply use the value of each parameter (***NVMCS, NVMT, NVMS, NVMP***) which it read 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....

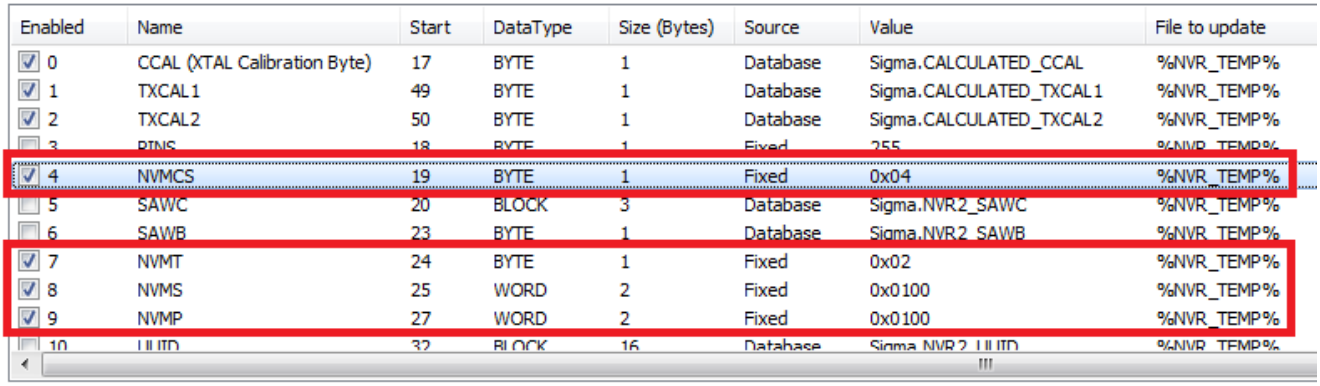


- Set the ***'Value'*** field to the value you wish to program into the ***'NVMCS'*** parameter. e.g. 0x04
- Check the settings match the screenshot above and leave all other settings unchanged!
- Click ***<OK>*** to save your amended settings.
- You will then be returned to the ***'NVR parameters'***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

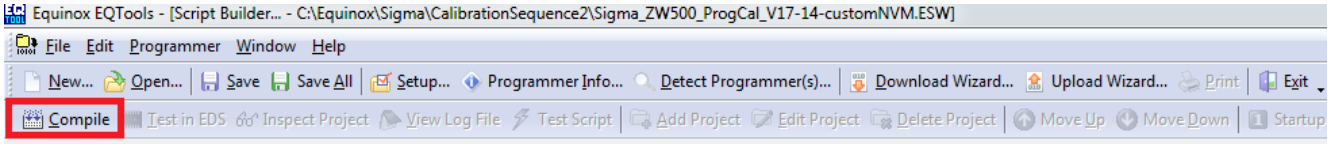
- Repeat the above procedure for each of the ***'Non-volatile memory (NVM)'*** parameters (***NVMCS, NVMT, NVMS, NVMP***)

- 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)***.



- 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 5 – Restore 'NVR Area' script

1.0 Overview

This section describes how to restore the ***'NVR Area'*** of a 500 series Z-Wave device which has a blank (erased to 0xFF) or corrupt ***'NVR Area'***.

The ***'Restore NVR Script'*** is a special version of the Sigma ***'Programming / calibration'*** script which will restore / re-program the ***'NVR parameters'*** back into a Z-Wave device.

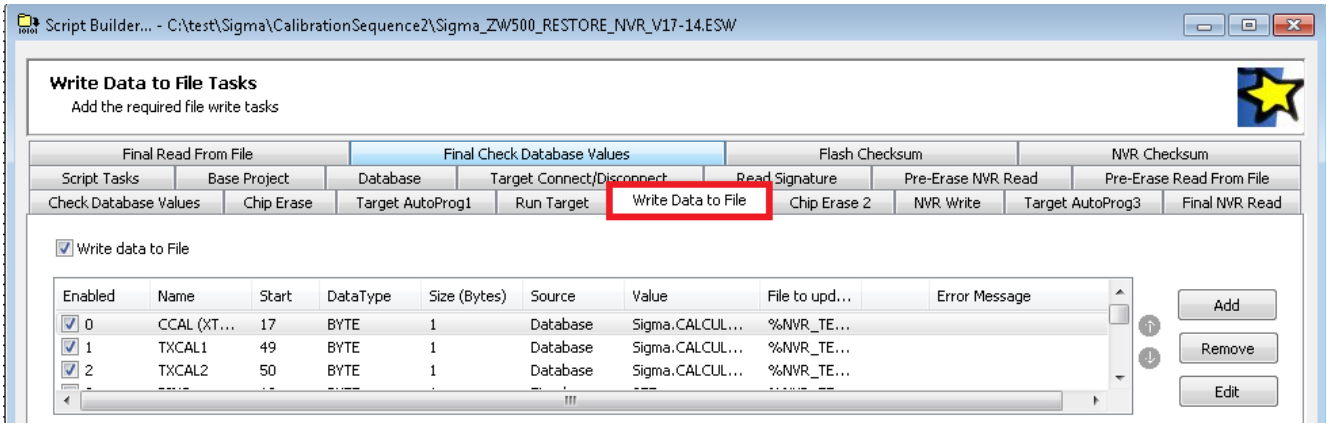
1.1 Modifying the default 'Restore NVR' script

The ***'Restore NVR Script'*** has been set up by Equinox with default values for all the ***'NVR parameters'***. Each customer Z-Wave target board / device will require customised values for the ***'NVR parameters'*** as these parameters will differ from device to module to target board.

If the Z-Wave device has been erased and the entire ***'NVR Area'*** has been blanked (set to 0xFF), then it is necessary to modify the ***'Restore Script'*** to restore / reprogram all the NVR parameters back into the target device. To set up the NVR parameter values, you will need to work out what the values should be.

Instructions:

- Start EQTools
- Select ***File – Open*** and then browse to and select the ***'Restore NVR'*** script e.g. ***Sigma_ZW500_RESTORE_NVR_V17-15.ESW***
- Select the ***'Write to file'*** tab.....



- At the top of this tab, you will see a list of all the available ***'NVR parameters'*** which can be programmed into the target Z-Wave device.
- It is possible to pre-set each ***'NVR parameter'*** to a custom value by simply selecting the relevant parameter and then entering the value you want to set the parameter to.

The following parameters must not be touched.....

- The ***'CCAL'*** and ***'TXCAL1 / TXCAL2'*** parameters are automatically calibrated / calculated when the script executes, so these parameters must not be changed by the user.
- The ***'Calculate NVR CRC16'*** parameter is the CRC16 checksum. This parameter is automatically calculated based on the values of all the other parameters in the list. This parameter must not be changed by the user.

Setting the remaining ***'NVR parameters'*** to your customised values....

- All the remaining parameters must be manually set to either the ***'Sigma factory default values'*** or your custom values which are required for your Z-Wave target board.
- To set the ***'External Non-Volatile Memory (NVR)'*** parameters (***NVMCS, NVMT, NVMS, NVMP***), please refer to the instructions in appendix 3.
- To set the ***'PINS'*** parameter, you will need to find out what the value should be by reading out this parameter from a working Z-Wave module.
- To set the ***'SAWB / SAWC'*** parameters, please refer to the original parameter settings or consult the designer of the ***'Saw filter'*** on your product.
- The ***'UUID'*** parameter is used to store a unique serial number of MAC address. If you are not using this parameter, then the value can simply be set to 0xFF (x16).
- The ***'VID'*** and ***'PID'*** parameters are used to set the USB ***'Vendor ID'*** and ***'Product ID'*** respectively of the device. If you plan to use the Sigma default USB values, then you can simply set both fields to 0xFFFF and 0xFFFF respectively. Otherwise, you will need to enter your custom ***'VID'*** and ***'PID'*** values.

Once you have set up the values for all the ***'NVR parameters'***, the table should look something like the example below....

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	0x00	%NVR_TEMP%
☑ 4	NVMCS	19	BYTE	1	Fixed	0x04	%NVR_TEMP%
☑ 5	SAWC	20	BLOCK	3	Fixed	0xFFFF	%NVR_TEMP%
☑ 6	SAWB	23	BYTE	1	Fixed	FF	%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	Fixed	0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	%NVR_TEMP%
☑ 11	VID	45	WORD	2	Fixed	0xFF	%NVR_TEMP%
☑ 12	PID	47	WORD	2	Fixed	0xFF	%NVR_TEMP%
☑ 13	Calculate NVR CRC16	126	WORD	2	CRC16	CalculateCRC16	%NVR_TEMP%

- All the parameters must have the ***'Enabled'*** option ticked.
- All parameters except ***'CCAL'*** , ***'TXCAL1 / TXCAL2'*** and ***'Calculate NVR CRC16'*** parameters must have the ***'Source'*** set to ***'Fixed'*** and the correct value defined in the ***'Value'*** field.

To save your modified ***'Restore NVR'*** script file.....

- Select ***File – Save*** to save your changes back to the Script File (*.esw).
- Click the ***'Compile'*** icon on the EQTools top icon bar → This will compile your changes into the ISP-PRO script file (*.esf).

1.2 Testing the 'Restore NVR' script in ISP-PRO

The ***'Restore NVR Script'*** can be executed within the ISP-PRO application just like any other script. This script should only be used if the ***'NVR Area'*** of the target board / device being programmed is known to be blank (contains all 0xFF). If this script is run on a target device which has valid NVR settings, then it will fail immediately.

Instructions:

- Start the ISP-PRO application
- Login and then click the ***'Setup'*** icon
- Select the ***'Restore NVR Script'*** e.g. ***Sigma_ZW500_RESTORE_NVR_V17-15.ESW***
- Start the programming network
- ISP-PRO will prompt you for the 'Project Collection' and 'Standalone Programming Project' to be used
- Start execution of the script by clicking the 'Connect' button or using the selection connection method.
- When the script runs, it first checks that the ***'NVR Area'*** of the target Z-Wave device is blank (all parameters are 0xFF).
- If the ***'NVR Area'*** is not blank, then the script will fail immediately. This failure indicates that the target device already has valid 'NVR data'.
- If the ***'NVR Area'*** is blank (all parameters are 0xFF), then the script will program the 'NVR parameters' with the custom values defined in the ***'Write data to*** file' tab in the script
- The ***'Calibration firmware'*** will always be programmed and then executed and the ***CCAL*** and ***TXCAL1 / TXCAL2*** will be calculated and then stored into the ***'NVR Area'***.
- The ***'Production firmware'*** will then be programmed

1.3 Checking the final 'NVR settings' are correct

It is very important to check that the values which are re-programmed into the ***'NVR Area'*** of a target device are actually correct. Once you have executed the ***'Restore NVR Script'***, it is recommended that you then read back the ***'NVR Area'*** using the EQTools – EDS utility and compare the values with a known working sample of the target board.

Appendix 6 - 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 'development' files are not required for ISP-PRO and so do NOT need to be copied to your production PC.

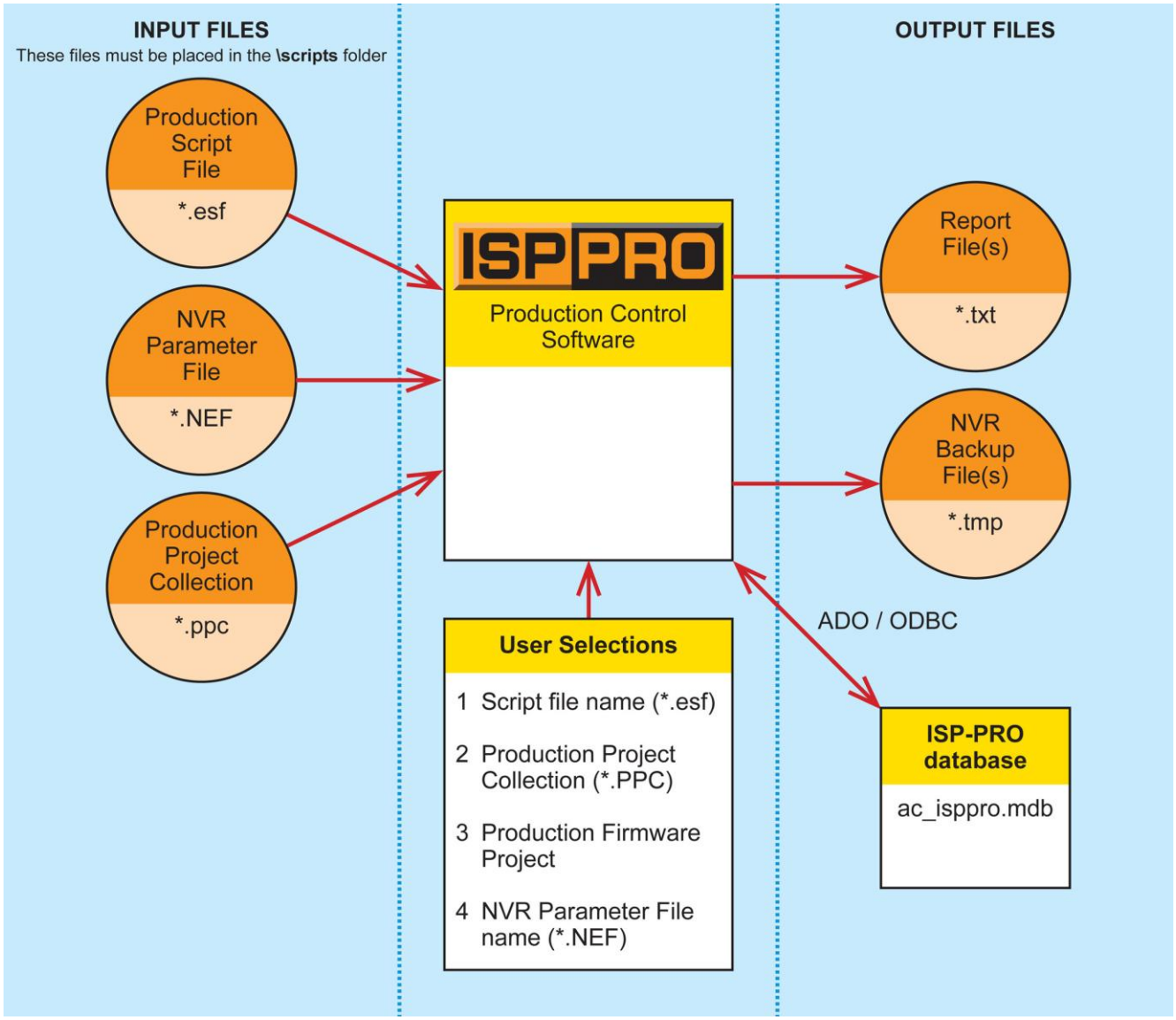
The only files which ISP-PRO actually requires to run the script in ***'production mode'*** are as follows:

- *.esf - Script File(s)
- *.PPC - Project Collection File(s)
- *.prj - Compiled Project File(s) - only the 'Base Project' is required
- *.NEF - NVR Parameter file (if NVR parameters are being merged into the NVR)

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 Overview of files required for ISP-PRO software

The illustration below shows which files are required by ISP-PRO and what files ISP-PRO creates when executing a programming script...



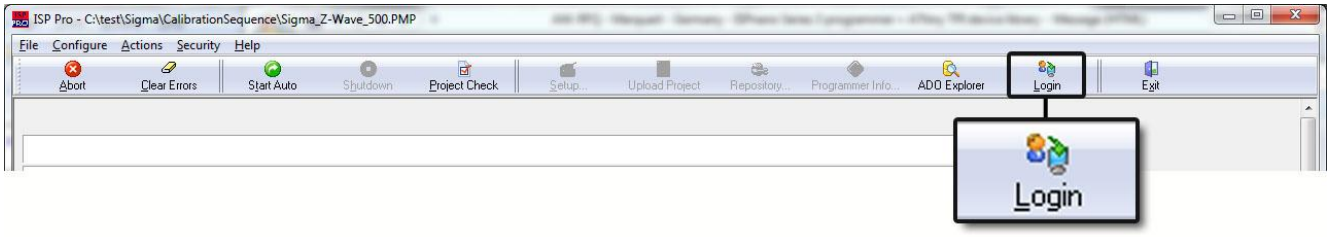
2.3 Setting up ISP-PRO to run the Programming Script(s)

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.

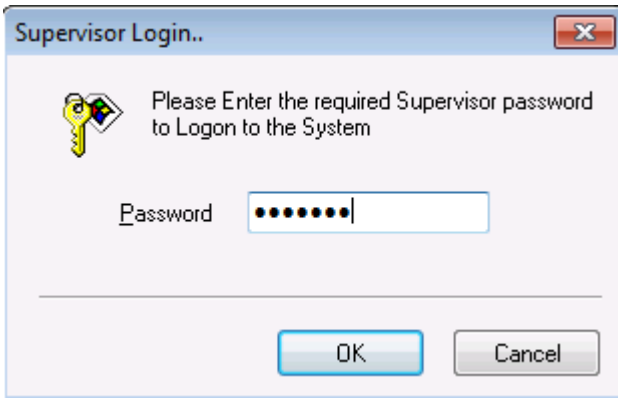
Please follow the steps below before attempting to execute a ***'programming script'***:

2.4 Start the ISP-PRO application and log in

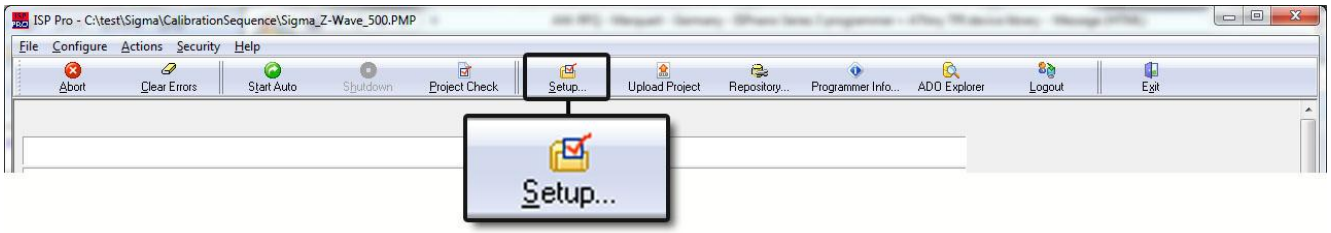
- Select ***<Start><Programs><Equinox><ISPPRO>*** → ISP-PRO application should start up..
- Click the ***<Login>*** button



- You will now be asked to enter your password.

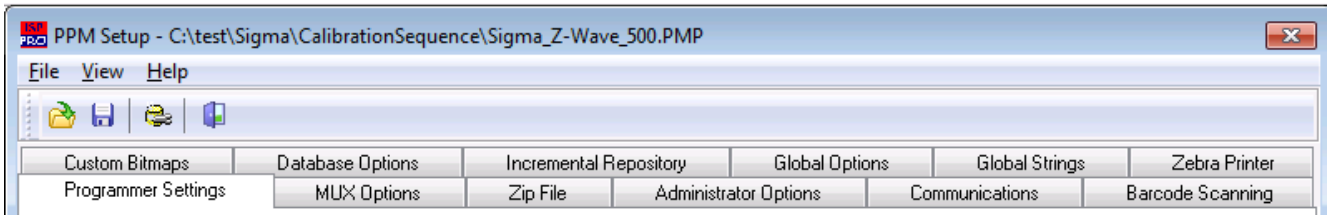


- Type in your password (default password is: ***equinox***)
- The ***<Setup>*** icon should now be selectable.



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

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

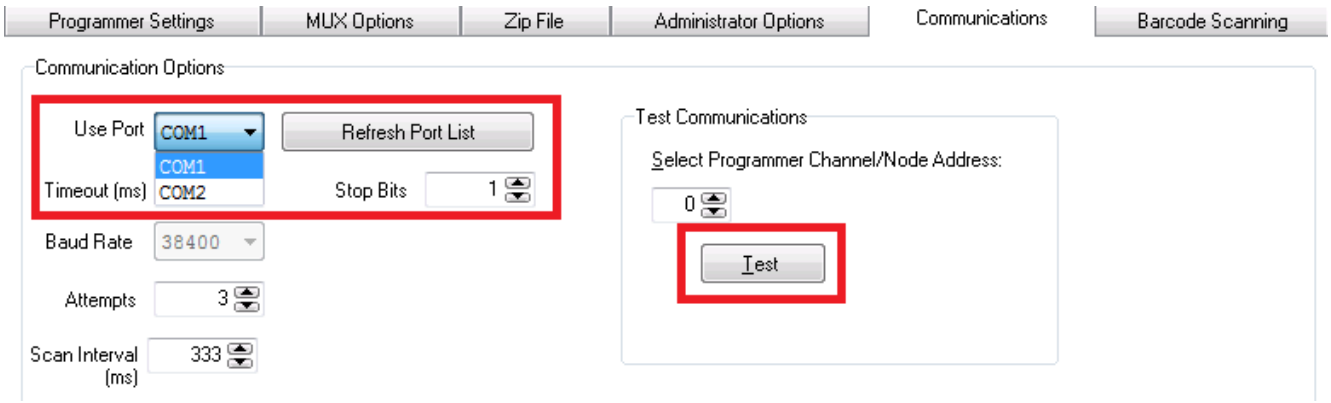


2.5 Setting up the correct COM port

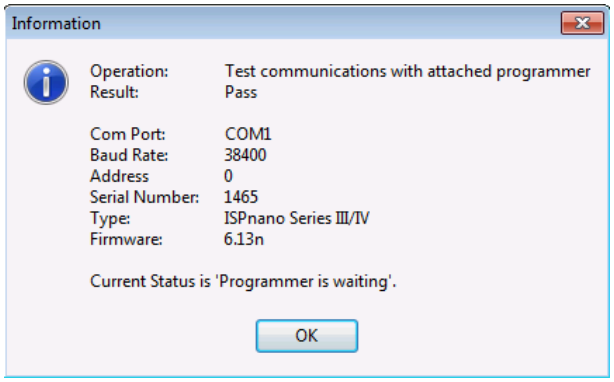
If this is the first time that you have used ISP-PRO, then it is likely that the COM port needs to be configured.

Instructions:

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



- 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.

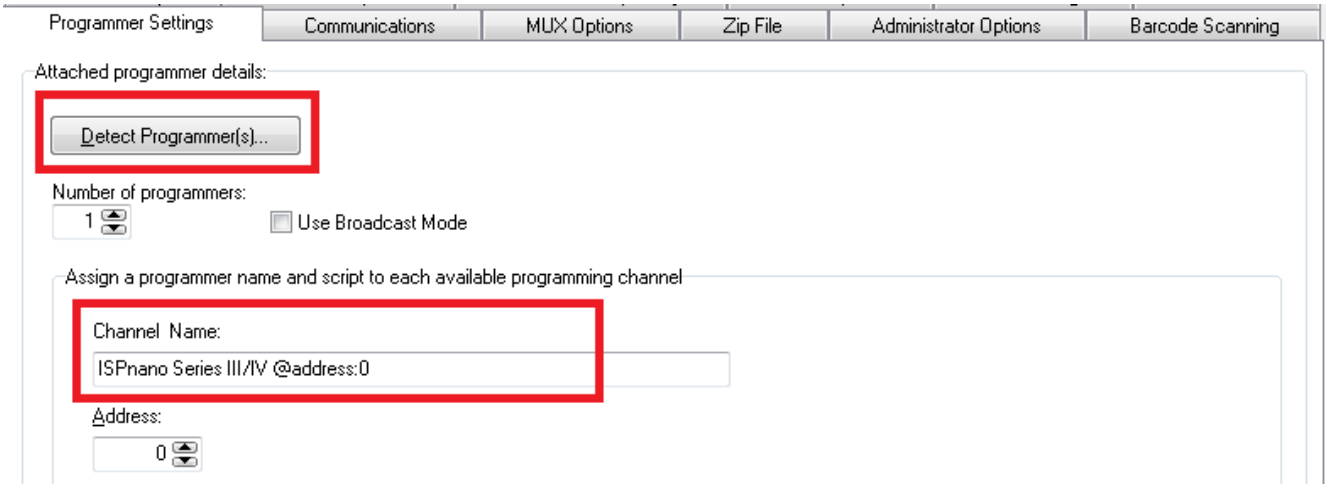


2.6 Detecting the attached programmer(s)

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



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

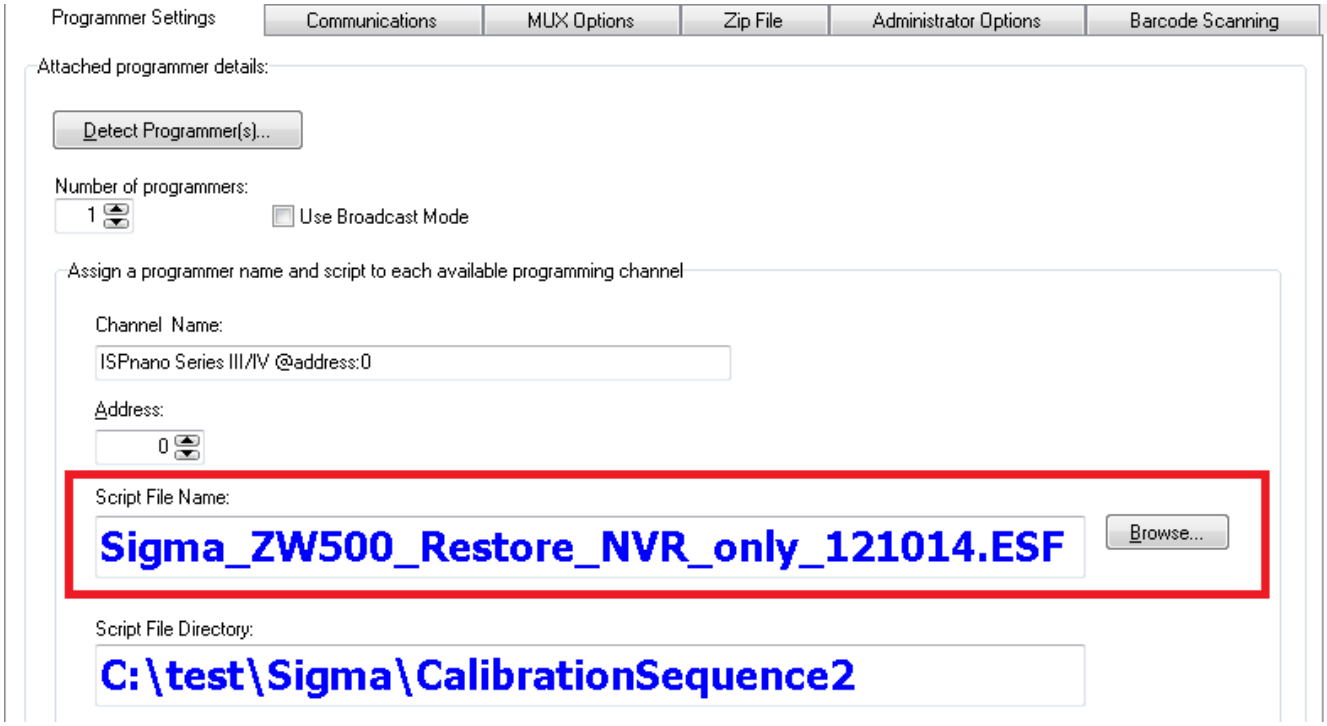
---> The detection process will provide a list of all attached programmers....in this case it has detected a single programmer at node address 0.

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

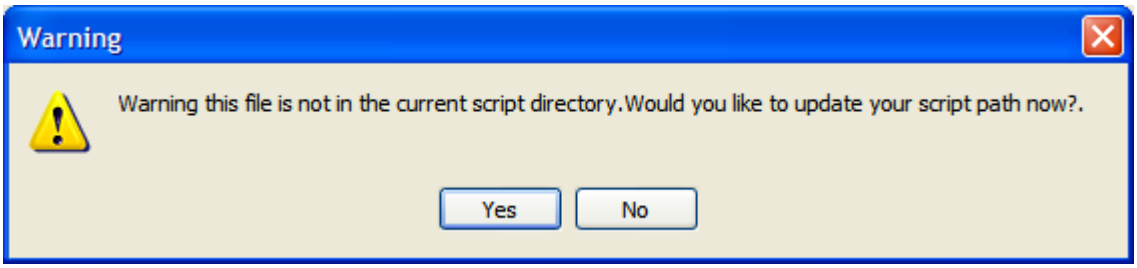
2.7 Selecting a Script File to run

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

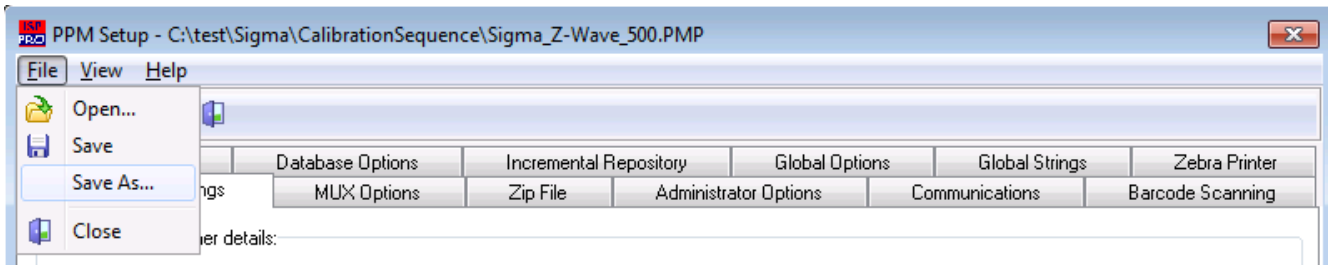
- Click the ***<Setup>*** icon
- Select the ***<Programmer Settings>*** tab



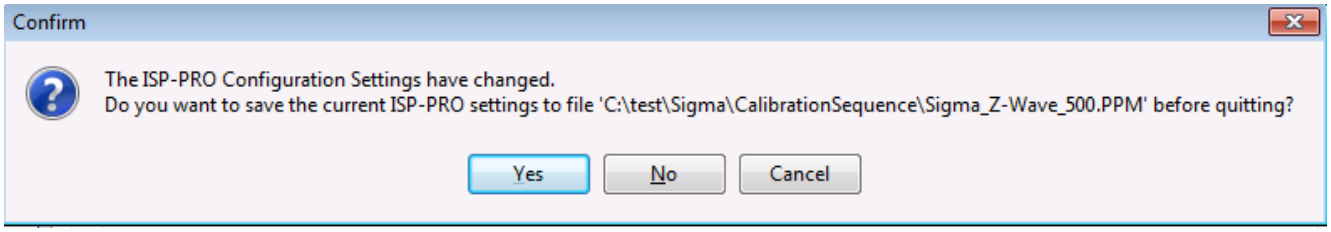
- If you wish to select a ***Script File (*.esf)*** for a single channel, click the ***<Browse>*** button.
- Browse to and select the required ***'Script File'*** which will have the file extension ****.esf***. The ***'Script File'*** should be located in your ***'Scripts'*** directory / folder.
- If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



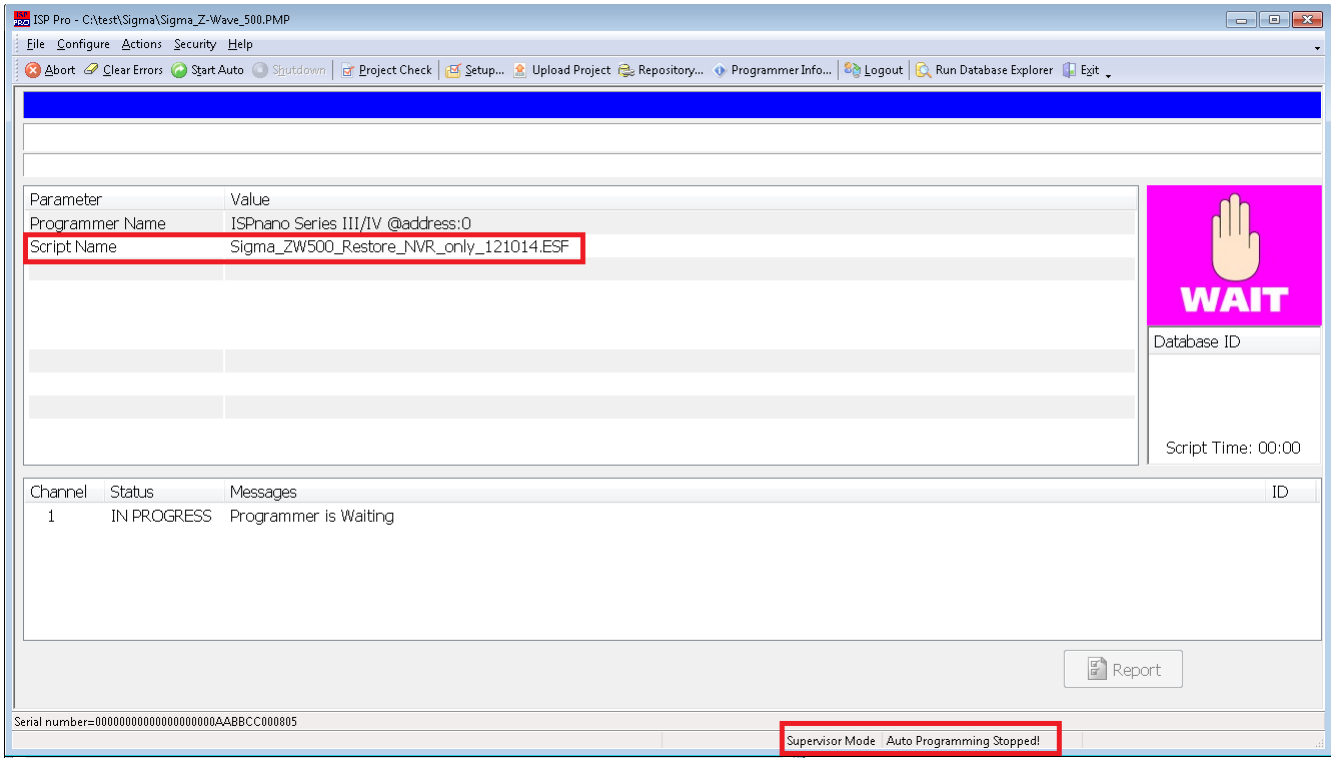
- If you click ***<Yes>*** then ISP-PRO will automatically set the ***'Script File directory'*** to the directory where your selected script is located.
- Once you've selected your ***'Script File'***, then you just need to save your settings by selecting ***'File - Save As...'*** and then specifying 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.....



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



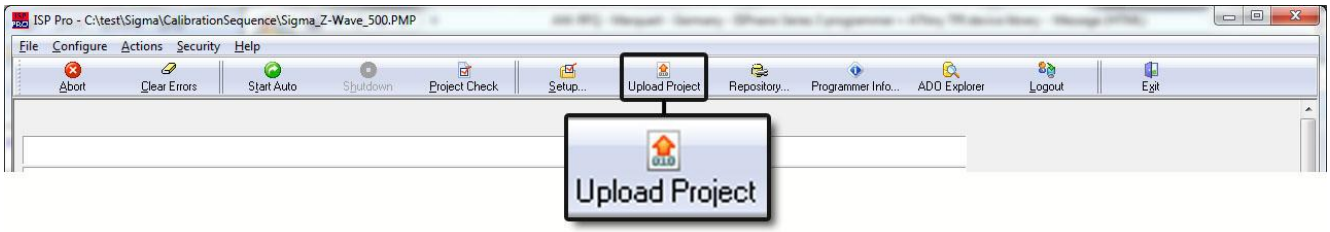
- ISP-PRO is still in ***'Supervisor Mode'*** allowing you to change any other settings before going into ***'Production mode'***.

2.8 Uploading your Project Collection to the programmer(s)

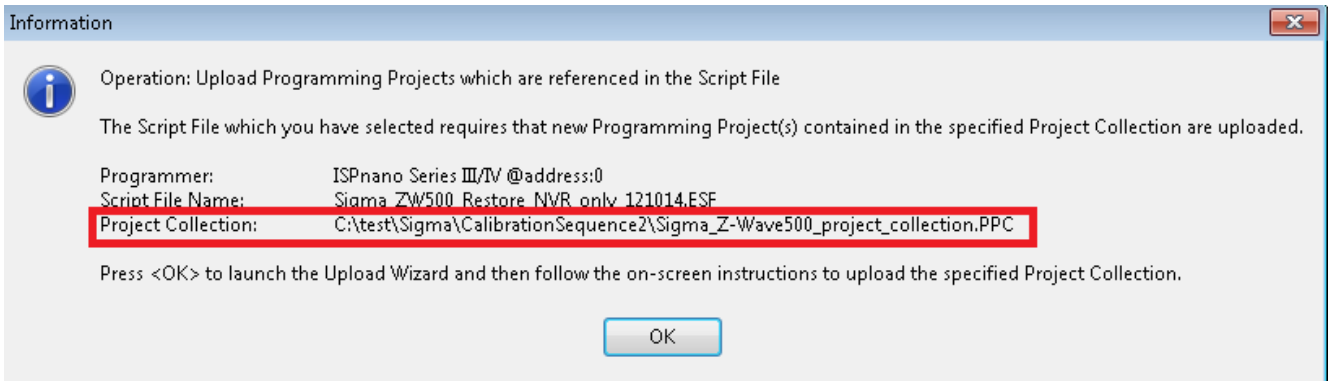
Before running any ***'Script File'*** which uses ***'Standalone programming projects'***, it is necessary to upload these projects to the programmer ***'FLASH Memory Store'***.

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

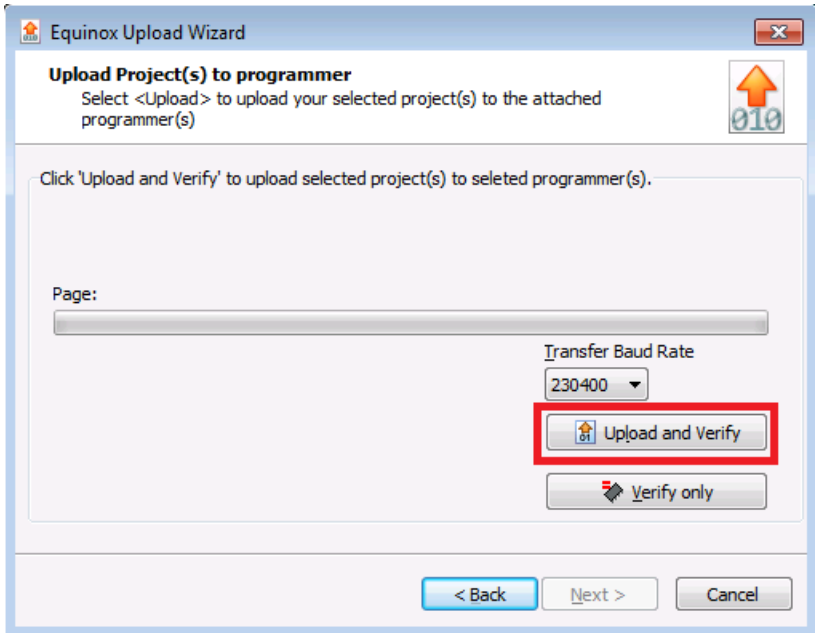
- Click the ***<Upload Project>*** button on the ISP-PRO icon bar.



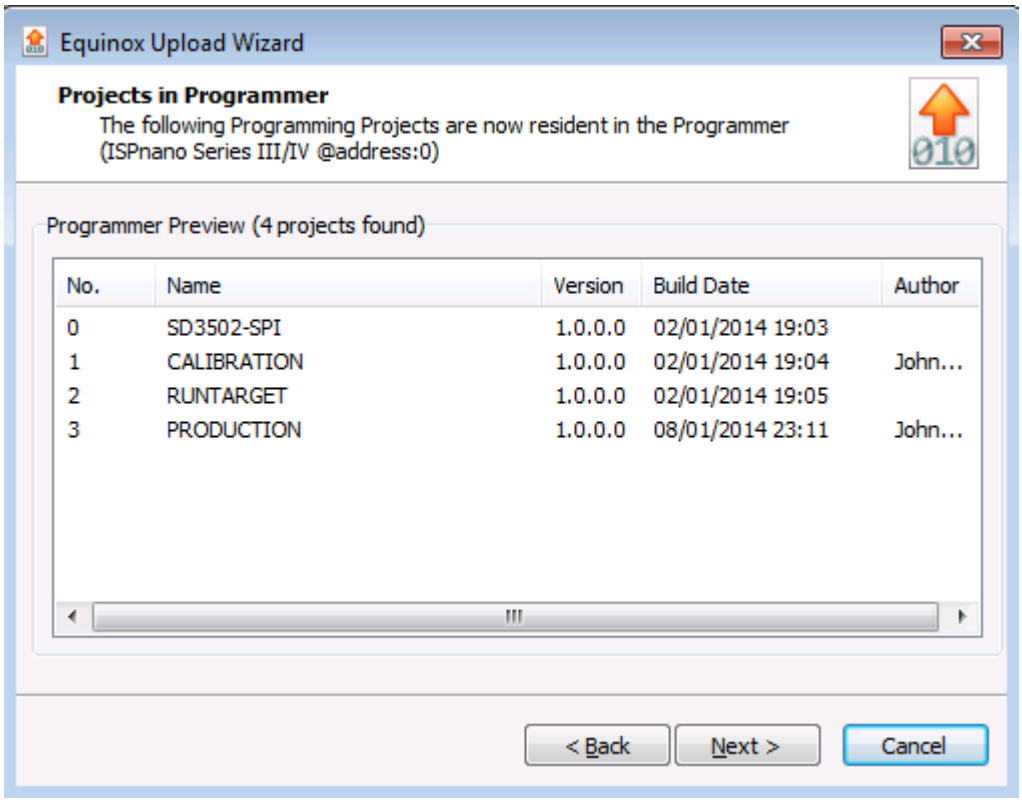
- If you have already selected the script file to execute, then ISP-PRO should automatically notify you of which ***Project Collection (*.ppc)*** file to upload.



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



- 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....

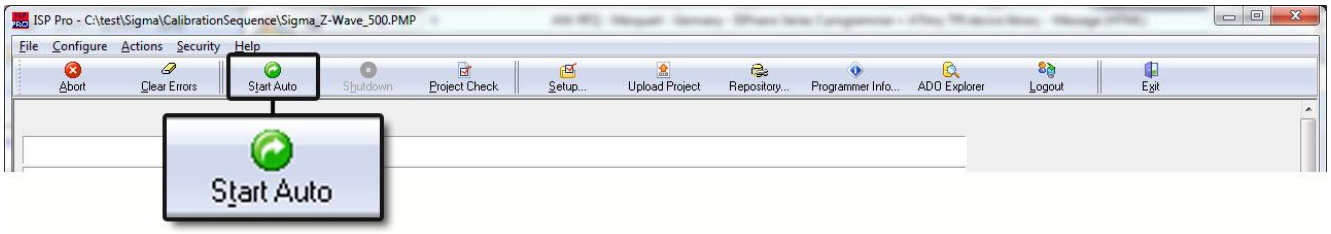


- Click the ***<Next>*** button on this screen and then the ***<Finish>*** button on the next screen to complete the project upload process.
- You will then be taken back to the main ISP-PRO screen.
- ISP-PRO is still in ***'Supervisor mode'*** with the programming network stopped.

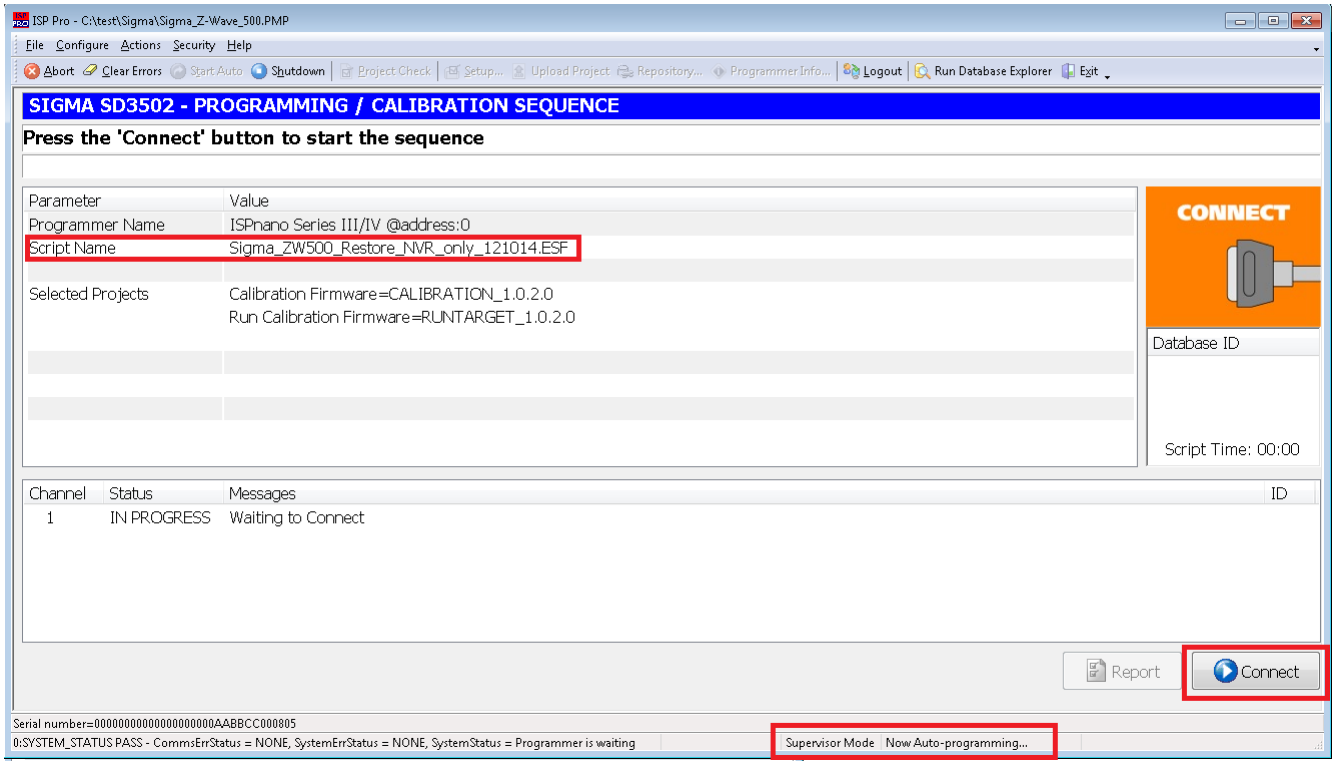
2.9 Running the programming Script(s)

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

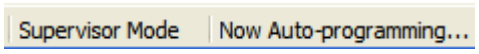
1. Click the **<Start Auto>** icon on the ISP-PRO Icon Bar



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



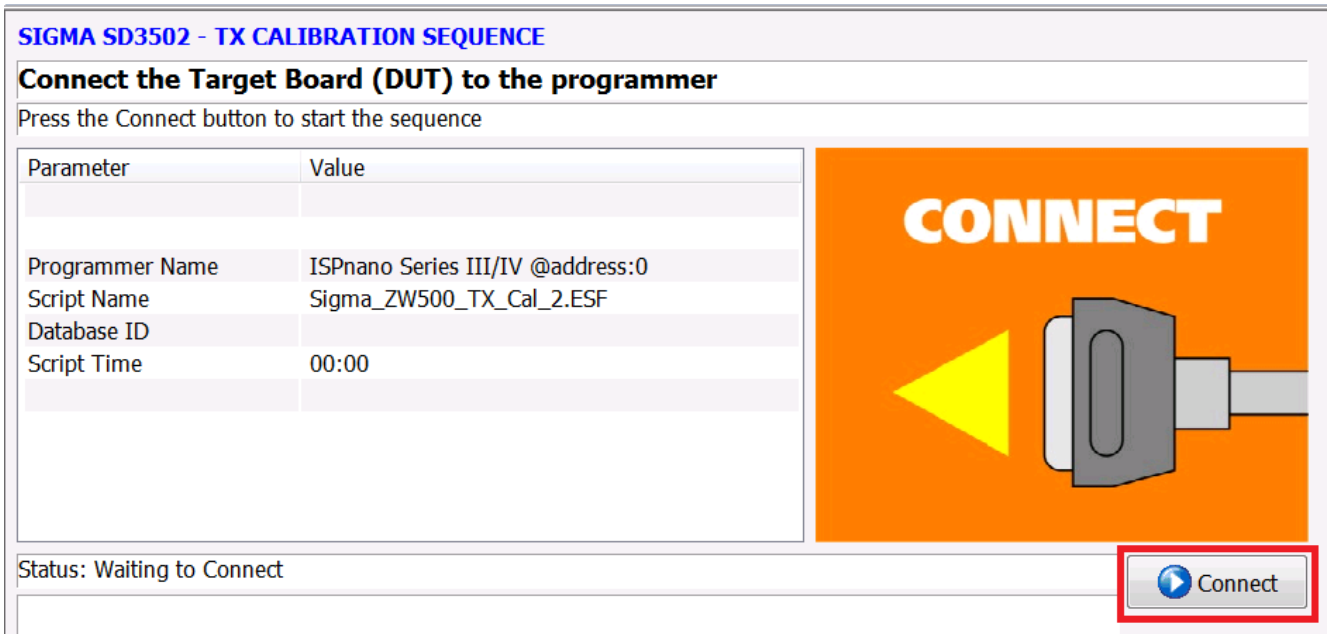
The message ***'Now Autoprogramming'*** should be displayed at the bottom right-hand corner of the ISP-PRO window.



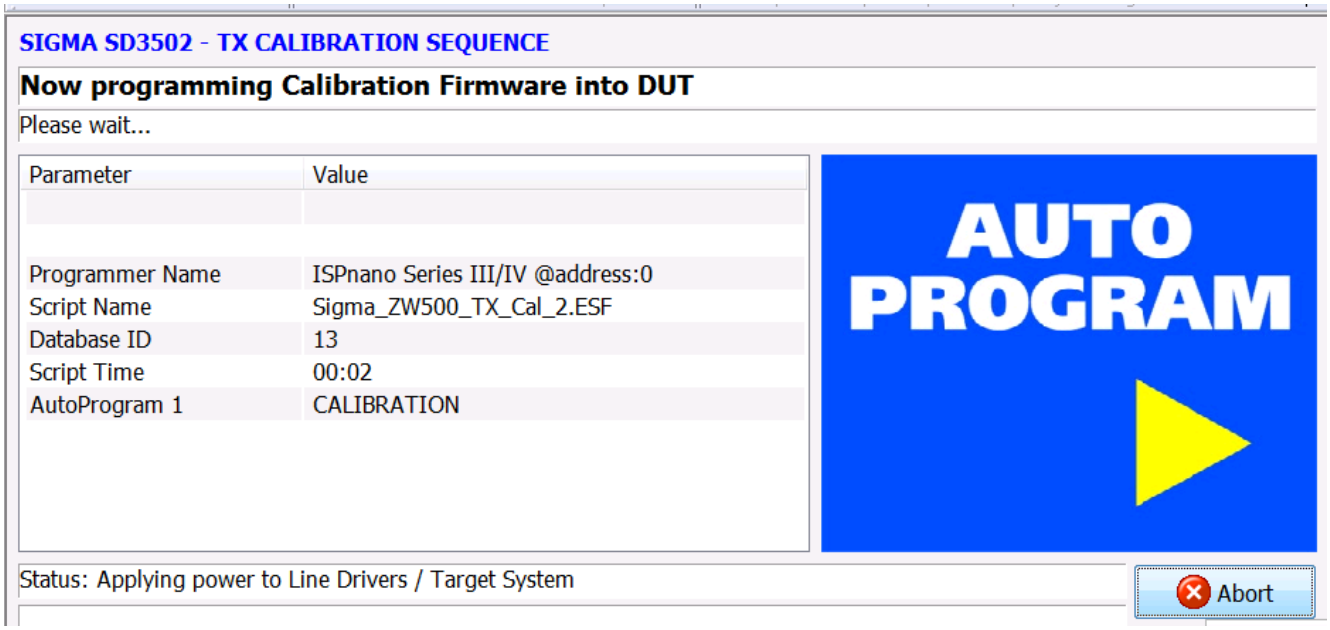
2.10 Executing the programming / calibration sequence

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.



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



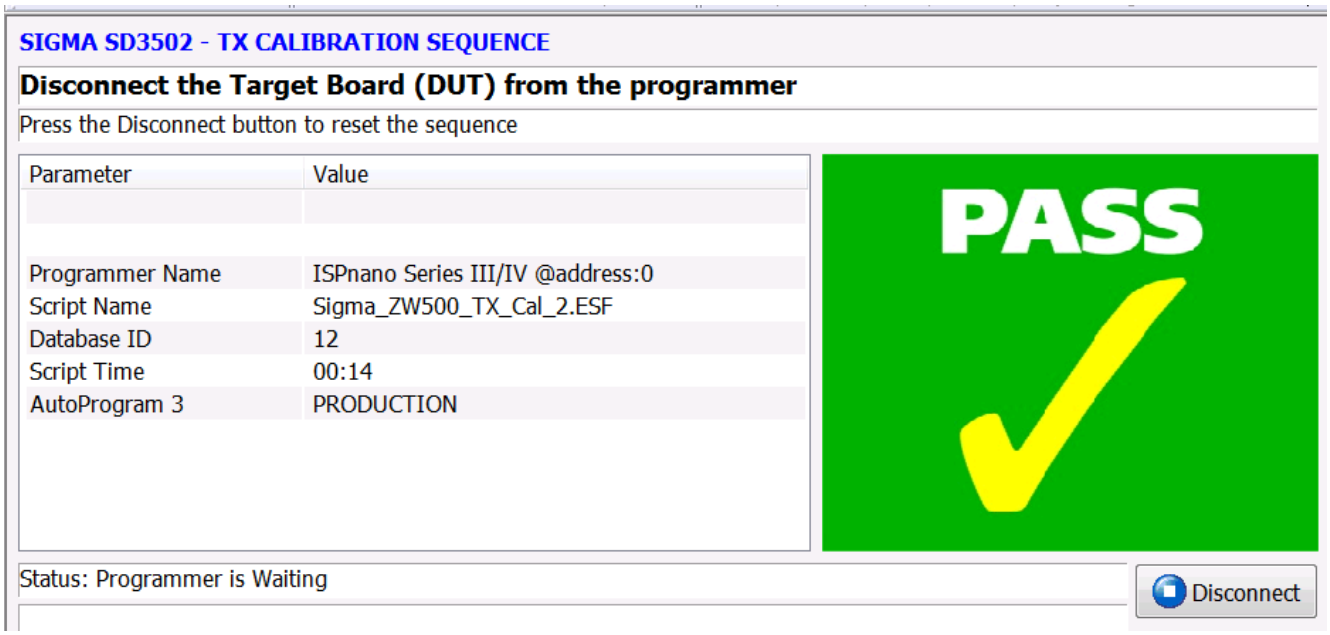
--> The icon will display ***'Auto Program'***
--> The ***'Script Timer'*** will now start timing the execution of the script.

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 !!!

2.11 Programming sequence - PASS

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



- 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)

2.12 Programming sequence - FAIL

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

- 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.

3.0 Setting up an Incremental Serial Number

3.1 Overview

It is possible to setup an ***'Incremental Serial Number'*** within ISP-PRO to generate any format of serial number, RFID, MAC address etc. There is a special utility within ISP-PRO called the ***'Incremental Repository'*** which is used to generate and store any form of 'serial number'. The generated 'Serial Number' can then be programmed into the FLASH or 'NVR Area' of a target device.

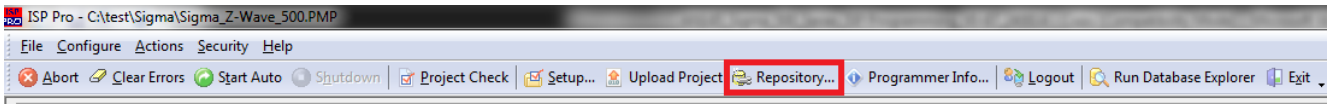
The ***'Sigma calibration / programming'*** scripts and projects are supplied in a single zip file by Equinox.

3.2 Importing a Serial Number from file

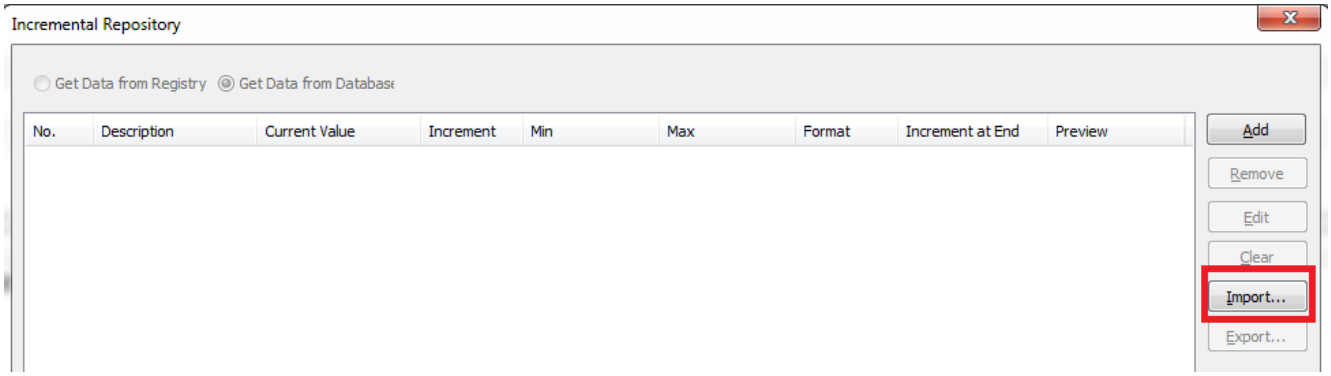
In many instances, a ***'Serial number'*** or ***'MAC address'*** may have already been setup for a particular application. It is possible to import this ***'Serial number'*** from a file directly into the 'Incremental Repository' - see instructions below.

Instructions:

- Start ISP-PRO
- Login
- On the top icon bar, click the ***'Repository'*** button

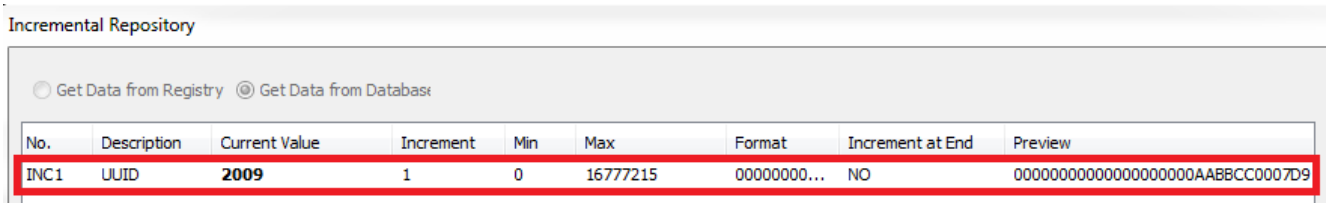


The ***'Incremental Repository'*** screen will then be displayed....



- Click the ***'Import'*** button on the right-hand side
- Browse to and select the required ***'Serial number file'*** eg. ***UUID.ERE***
- This should import the ***'Serial number'*** value, increment, format etc into ***the 'Incremental repository'***.
- In this example, the ***'Serial number'*** is called ***'INC1'*** and is a 16-byte number which is written into the ***'UUID'*** NVR field.

- The ***'Current value'*** is the value which will be programmed next which is represented in the actual display format in the ***'Preview'*** column.....



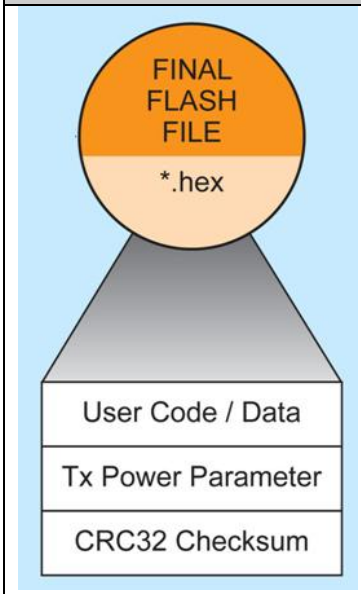
- Once you have setup the ***'Incremental repository'*** then this saves ***'INC1'*** to the database so the value of this ***'Serial number'*** will always be remembered.
- Click ***'Close'*** to save the ***'Incremental repository'*** settings.

Appendix 7 - FLASH File - CRC32 Checksum

1.0 Overview

The Z-Wave 500 series devices feature a 32-bit (4 byte) FLASH checksum ***(CRC32)*** which is used by both the Equinox programmer and the Z-Wave device itself at run-time to validate (verify) that the FLASH contents have been programmed correctly are not corrupt. The ***'CRC32 Checksum'*** is calculated over the entire FLASH address range except for the last 4 bytes. The calculated ***'CRC32 Checksum'*** is stored in the top 4 bytes of the FLASH.

The table below shows the various components which make up the ***'Final FLASH File'*** which is used to program the FLASH area of the Z-Wave device....

	**FLASH File** The ***'Final FLASH File'*** is made up of the data detailed below...
	User Code / Data This is the actual ***'Z-Wave Product Firmware'*** and any other 'user data' which needs to be programmed into the 'FLASH area' of the Z-Wave device.
	Tx Power Parameters The ***'Tx Power Parameters'*** control the output power of the Z-Wave device RF transmitter. The power parameters must be stored at specified locations in the ***'Final FLASH File'***.
	CRC32 FLASH Checksum The ***'Final FLASH File'*** must contain a 4-byte ***'CRC32 Checksum'*** of the entire FLASH image. This checksum must be stored in the last 4 bytes of the file.

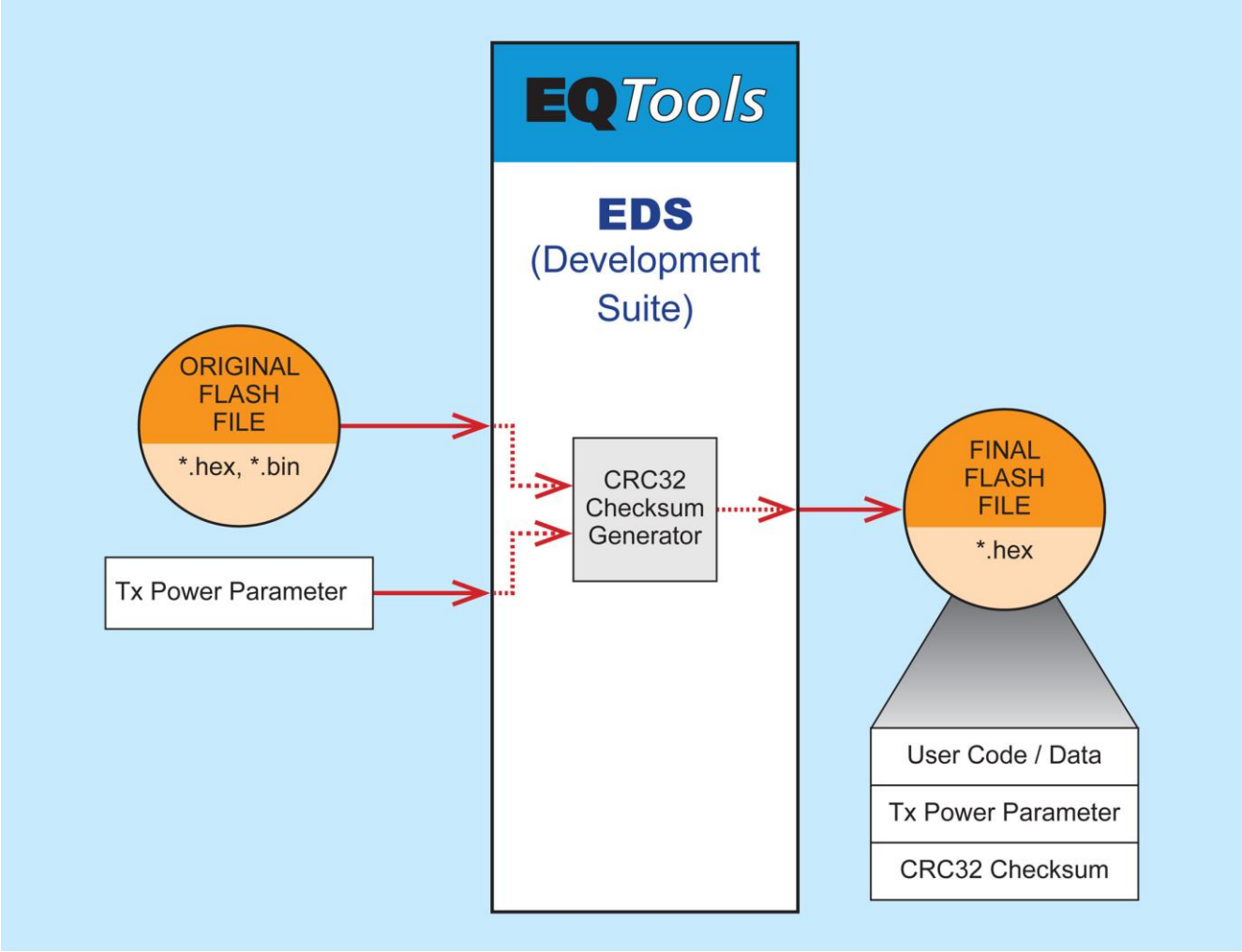
Important notes:

1.The Equinox EQTools configuration software expects that a valid ***'CRC32 Checksum'*** is pre-stored in any input BINARY or HEX file for a Z-Wave 500 series device. If a valid ***'CRC32 Checksum'*** is not detected, then EQTools will display an error message ***'Invalid CRC32 Checksum'***.

2. It is recommended that you update the ***'Tx Power Parameters'*** before you calculate the ***'CRC32 Checksum'*** because the checksum will need to be changes if you update the ***'Tx Power Parameters'***.

1.1 Using EQTools - EDS to update the CRC32 checksum

The ***EQTools - EDS*** utility can be used to update your existing ***'Firmware FLASH file'*** with the correct ***'CRC32 Checksum'*** and ***'Tx Power Parameters'***. EDS takes the ***'Original FLASH file'*** and the ***'Tx Power parameters'*** as input, calculates the ***'CRC32 checksum'*** of the entire FLASH image and then outputs the ***'Final FLASH File'*** - see illustration below.



1.2 Why do I get an invalid FLASH CRC32 checksum error?

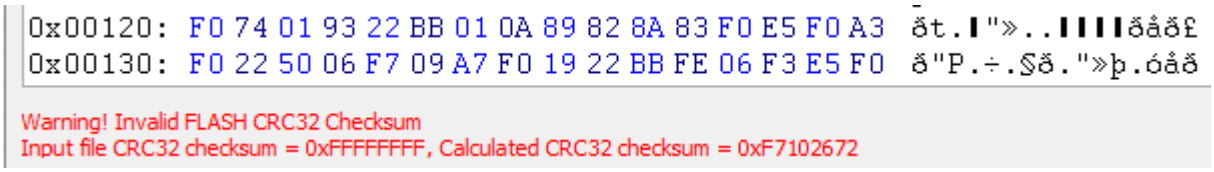
When a BINARY or HEX file is loaded into the ***'FLASH buffer'*** of the Equinox EQTools software, the file loader automatically checks to make sure that a valid ***'CRC32 Checksum'*** for the data in the file is stored in the last 4 bytes of the file data image.

The screenshot below shows a valid ***'CRC32 Checksum'*** stored in last 4 bytes of the ***'FLASH file'*** (address range: 0x1FFFC - 0x1FFFF).....

```
0x1FFE0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x1FFF0: FF FF FF FF FF FF FF FF FF FF FF FF 72 26 10 F7
```

Sigma 500 Checksum: Input file CRC = 0xF7102672, Calculated CRC = 0xF7102672

If the input file does not contain a valid ***'CRC32 Checksum'*** then the following error message will be displayed.....



This error message simply means that the input BINARY or HEX file either has no ***'CRC32 Checksum'*** setup in it as the CRC32 checksum of the file is 0xFFFFFFFF.

1.3 Loading standard Z-Wave Firmware releases from Sigma

If you are trying to load a standard Z-Wave ***'firmware file'*** released by Sigma Designs for a Z-Wave 500 series device, then none of these 'standard' hex files contain a valid ***'CRC32 Checksum'***. It is therefore necessary to use the Equinox ***EQTools - EDS*** development software to resave the hex file with the correct ***'CRC32 Checksum'***.

1.4 Loading a custom Z-Wave Firmware release

If you have created your own BINARY or HEX file to program into the FLASH area of a Z-Wave 500 series device using a third party compiler, then your files will definitely NOT contain the ***'CRC32 checksum'*** as part of the file image. It is therefore necessary to use the Equinox ***EQTools - EDS*** development software to resave the hex file with the correct ***'CRC32 Checksum'***.

1.5 How to correct / add a valid CRC32 checksum to an input file

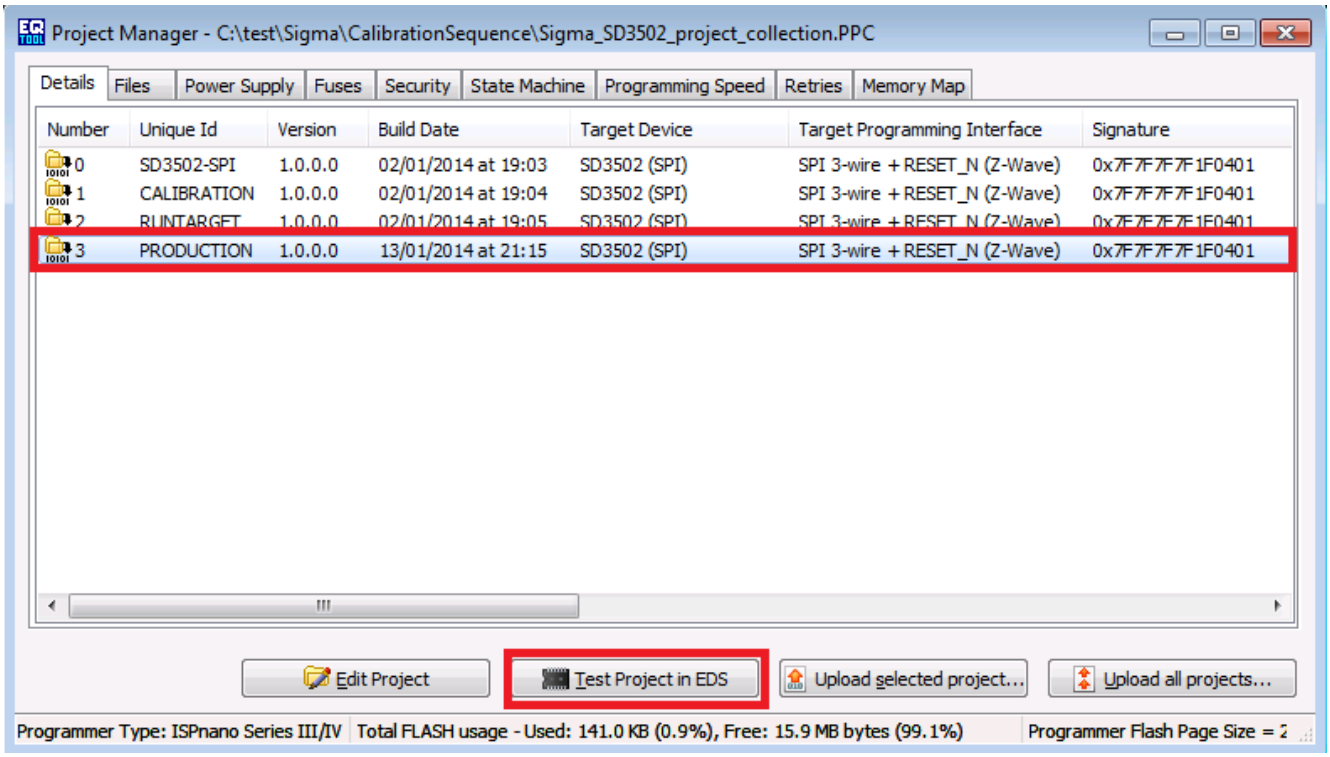
If your input ***'FLASH file'*** does not have a valid ***'CRC32 Checksum'***, then the file cannot be used for programming within EQTools. It is necessary to generate a ***'CRC32 Checksum'*** for the file and then re-save the file with this checksum before the file can be used with EQTools to program a 500 series Z-Wave device.

Please follow the instructions starting in section 1.6 to create a hex file with a valid embedded ***'CRC32 Checksum'***.

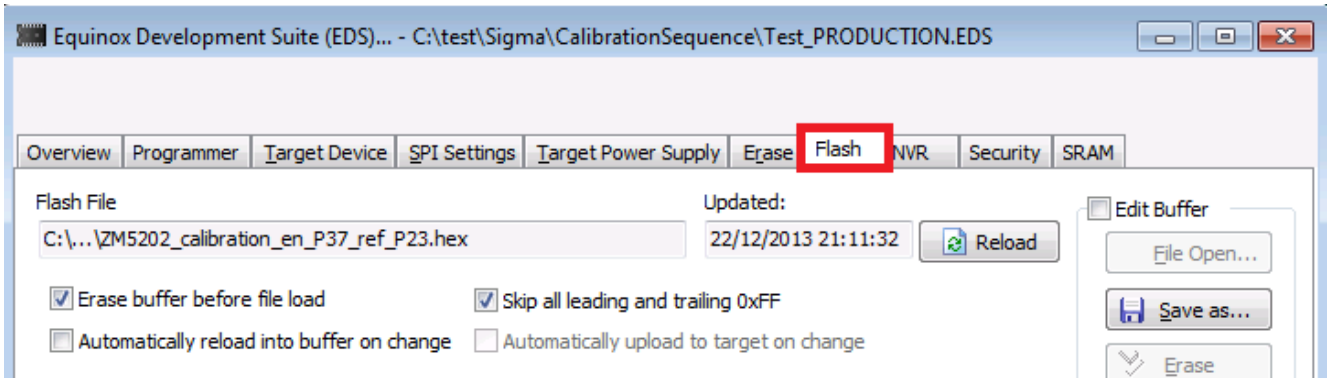
1.6 Opening the project in EDS (Development mode)

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



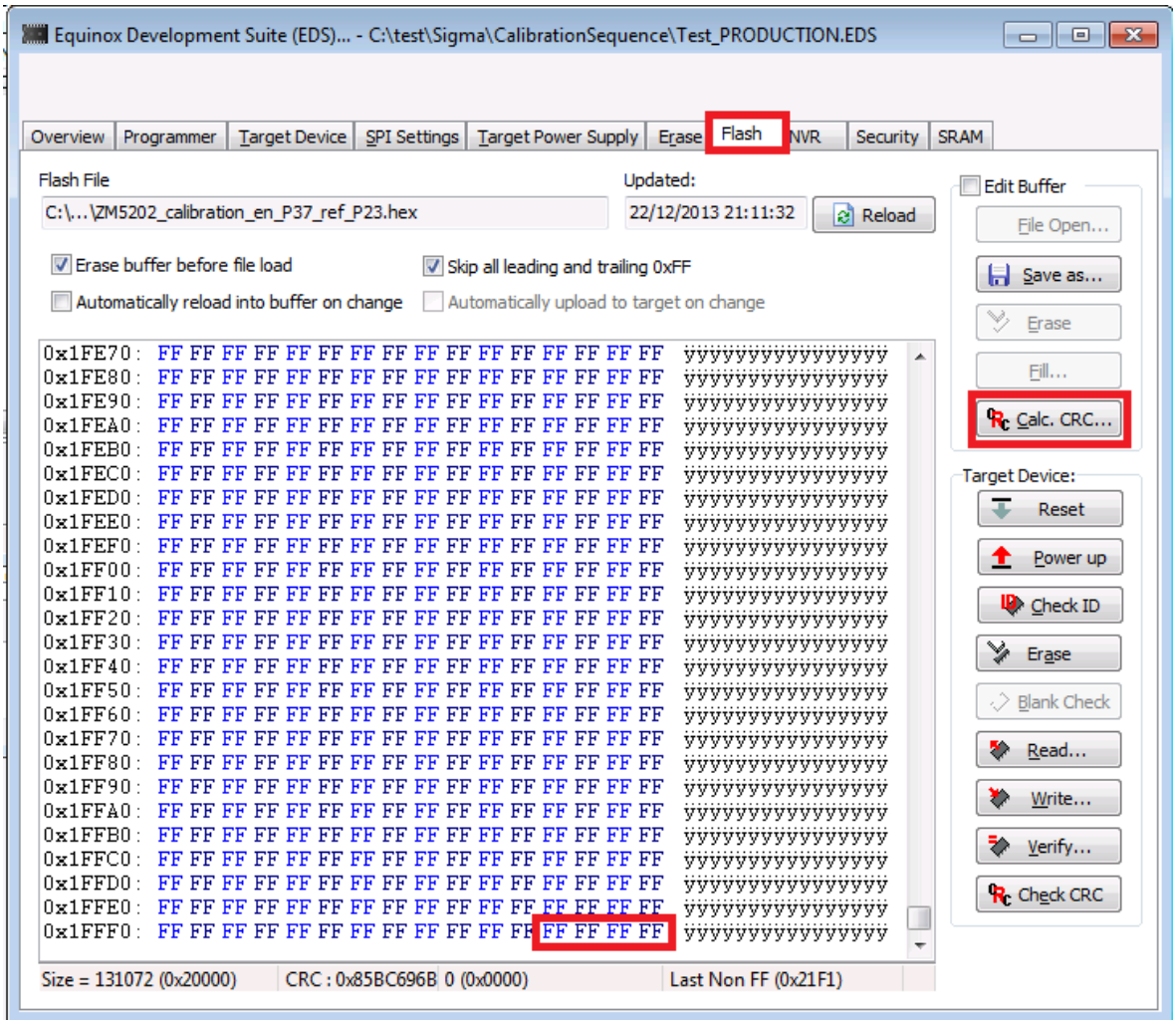
The selected project should then launch in ***'EDS - Development mode'***....



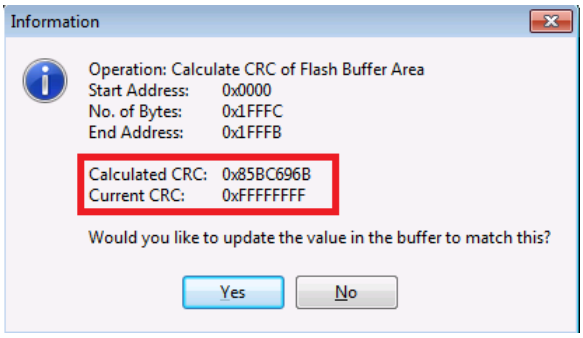
1.7 Calculating the CRC32 checksum of the input file

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

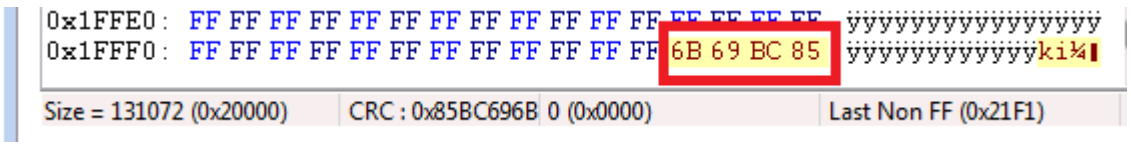
- Select the ***'FLASH'*** tab
- Your selected input file should be displayed in the buffer area
- Use the scroll bar on the right-hand side of the EDS window to scroll down to the end of the target device FLASH area.



- 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.



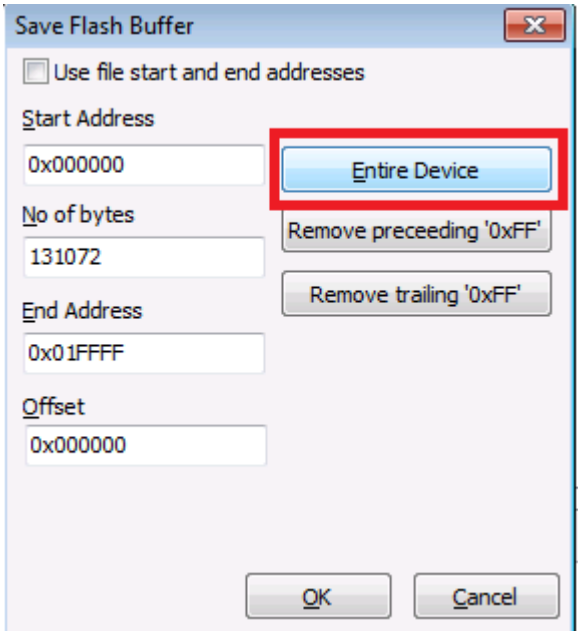
- Click the ***'Yes'*** button to update the ***'Calculated CRC'*** value into the last 4 bytes of the ***'FLASH buffer'***
- If you look at the last 4 bytes of the ***'FLASH buffer'***, they have now been updated with the ***'Calculated CRC checksum value'***....



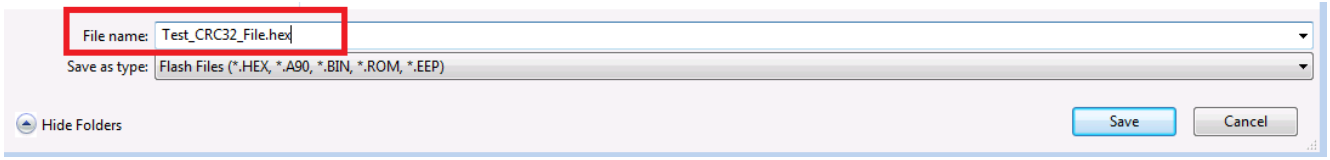
1.8 Saving the revised file with the CRC32 checksum

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.....



- Click the ***'Entire device'*** button --> This selects the entire address range of the device which includes the ***'CRC32 checksum'*** 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.



- 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. You may want to change the file name to append 'CRC32' on the end.
- This file can now be loaded back into the ***'PRODUCTION'*** project as it has a valid ***'CRC32 checksum'***.

Appendix 8 - 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.....

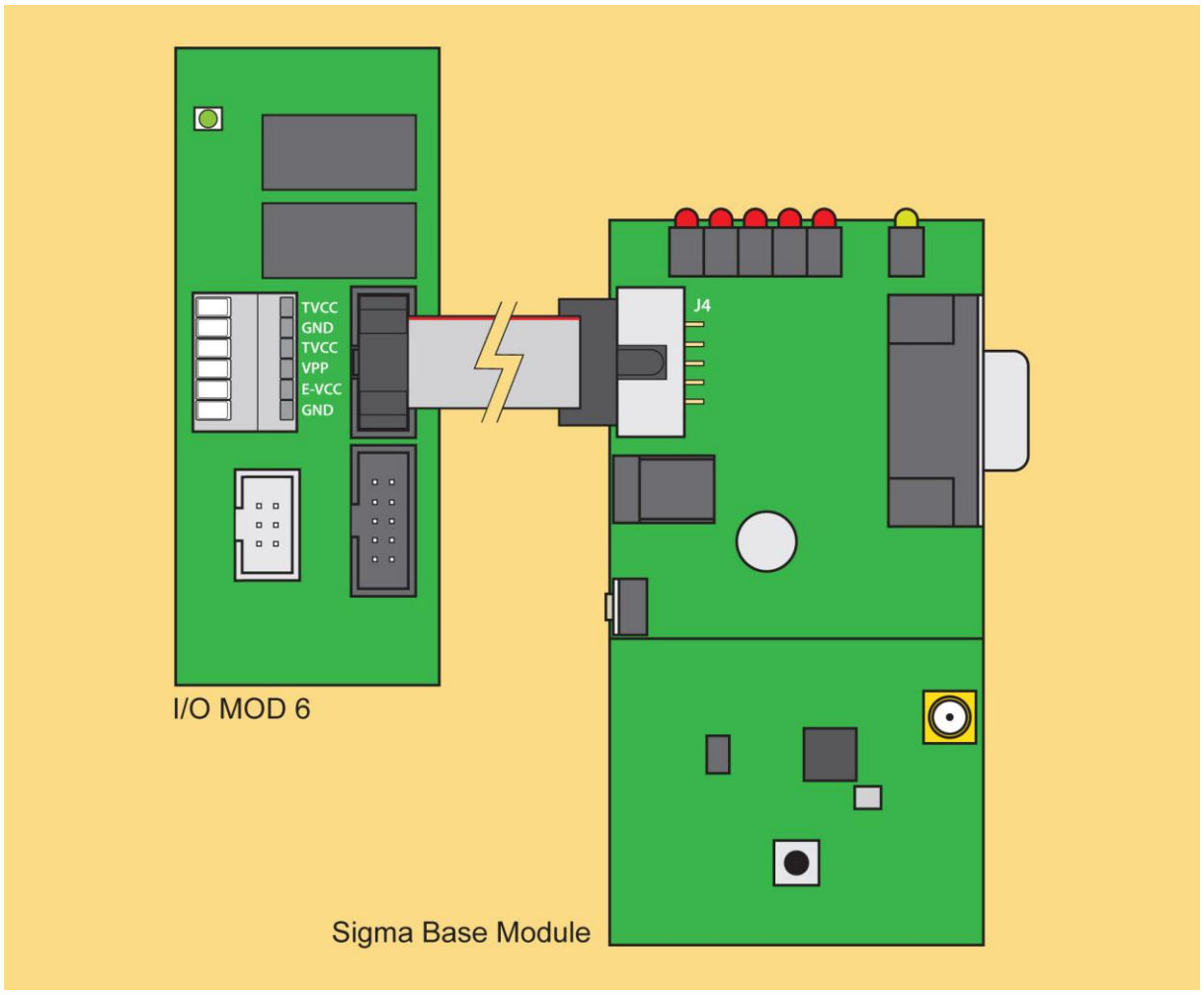
- ISPnano Series IV programmer
- IOMOD6 - I/O Connector Module
- Sigma SD3502 Evaluation Module (available to order from Sigma Designs)
- 10-way IDC ribbon cable

1.2 Connecting the programmer to the Sigma eval module

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

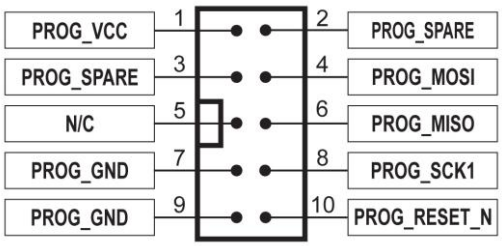
Instructions:

- 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 9 – Sigma connector definitions

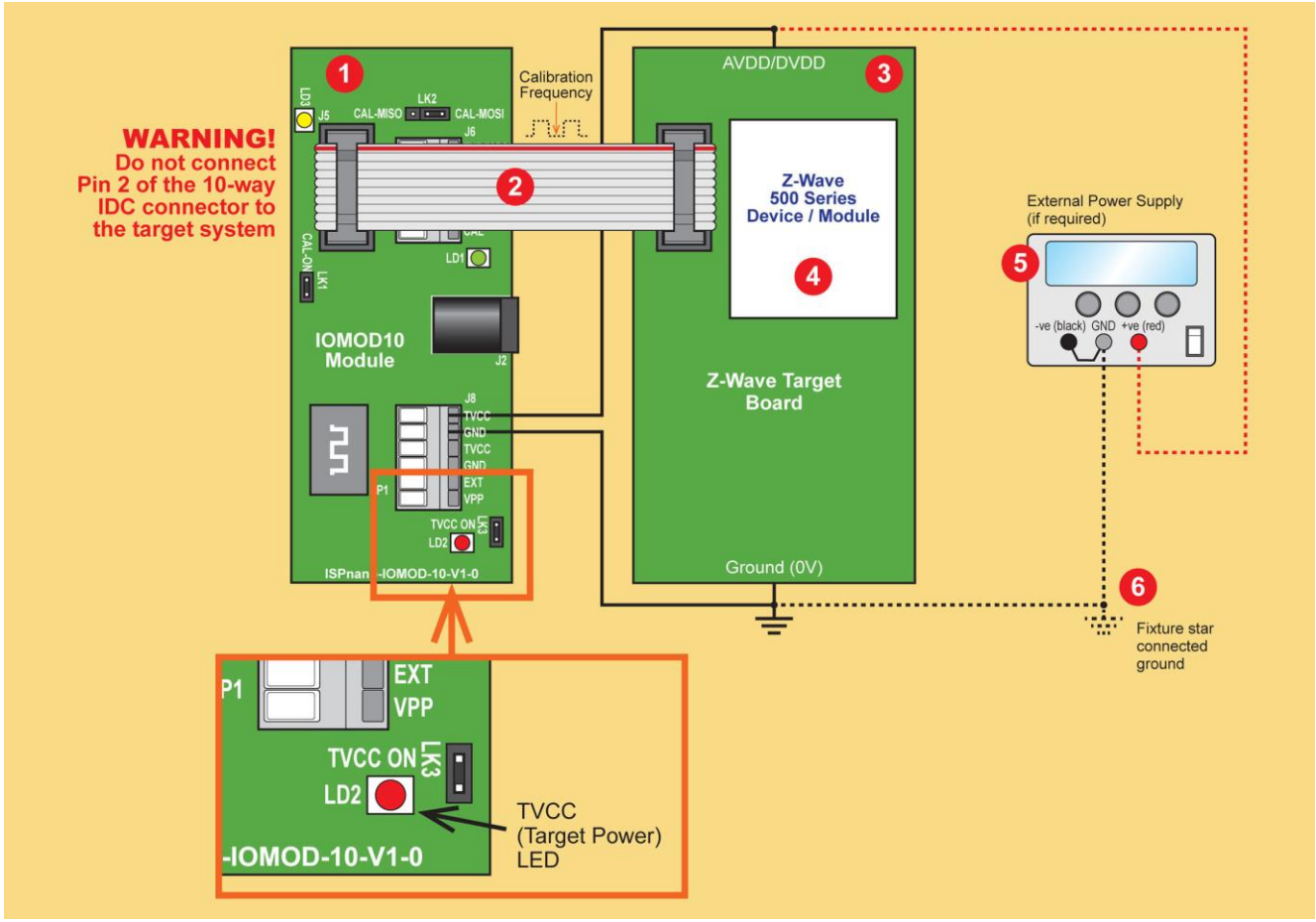
1.0 Sigma - ISP Header Selection

#	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.	

Appendix 10 – Connecting the Z-Wave target board to an ISPnano programmer

1.1 IOMOD10 module - connections to Z-Wave target board (SPI interface)

The diagram below shows the typical connections between the Equinox IOMOD10 connector module and a Z-Wave target board using the SPI programming interface.



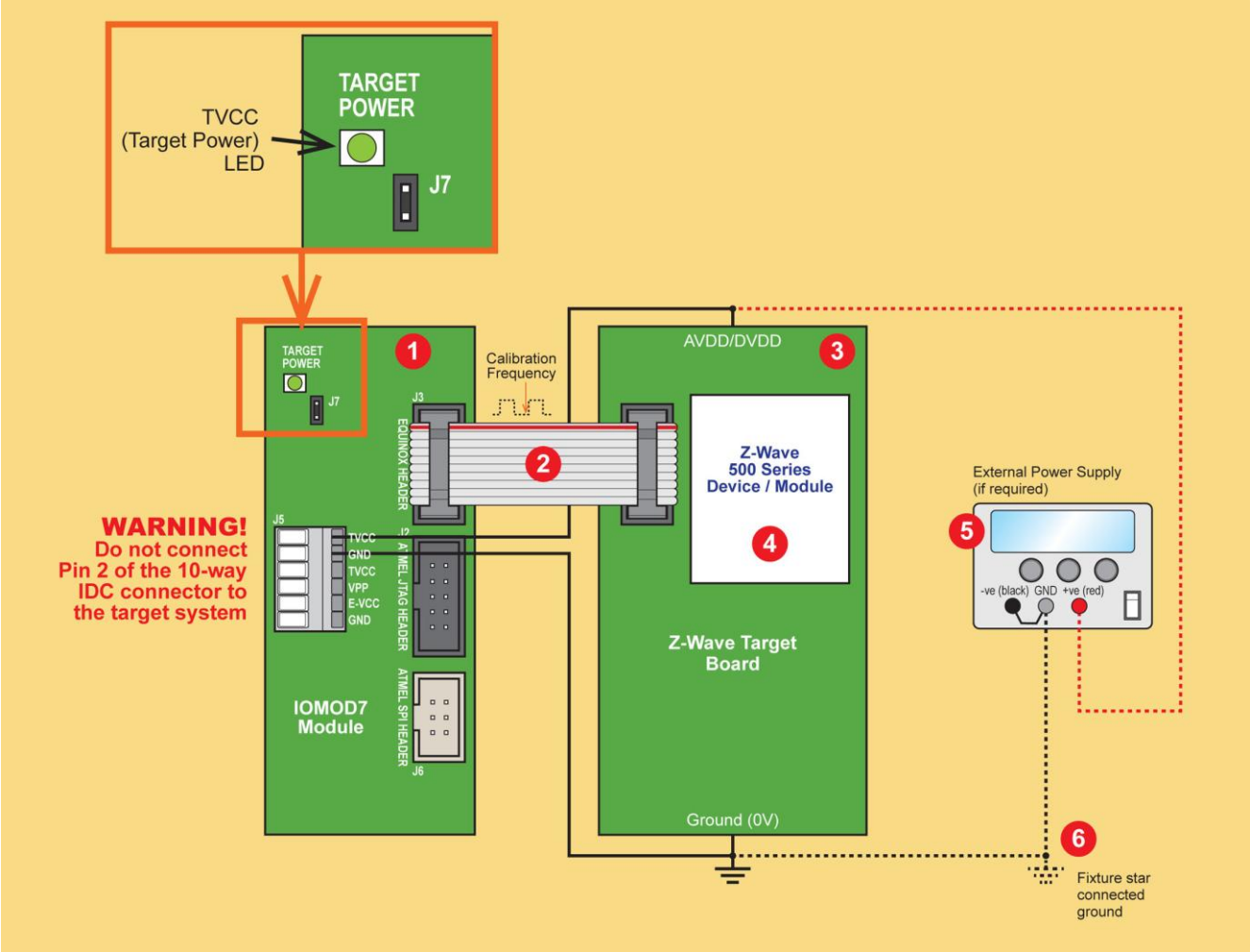
Number	Description
1	IOMOD10 calibration / connector module
2	10-way ISP cable
3	Customer Target Board featuring Z-Wave module or IC
4	Z-wave module or IC
5	External power supply (used to power the Target Board) if the programmer is not supplying power)
6	Fixture main GROUND connection. The Target Board, programmer and External Power Supply should all be star connected to this common GROUND point.

Important notes...

1. It is recommended that separate thick wires are used for the "Power 0V" and "Power VCC" between the programmer and the target board.

2. A separate GROUND connection should be made from the Target Board to the star-connected GROUND point in the fixture.

1.2 IOMOD7 module - connections to Z-Wave target board (SPI interface)



Number	Description
1	IOMOD7 connector module
2	10-way ISP cable
3	Customer Target Board featuring Z-Wave module or IC
4	Z-wave module or IC
5	External power supply (used to power the Target Board) if the programmer is not supplying power)
6	Fixture main GROUND connection. The Target Board, programmer and External Power Supply should all be star connected to this common GROUND point.

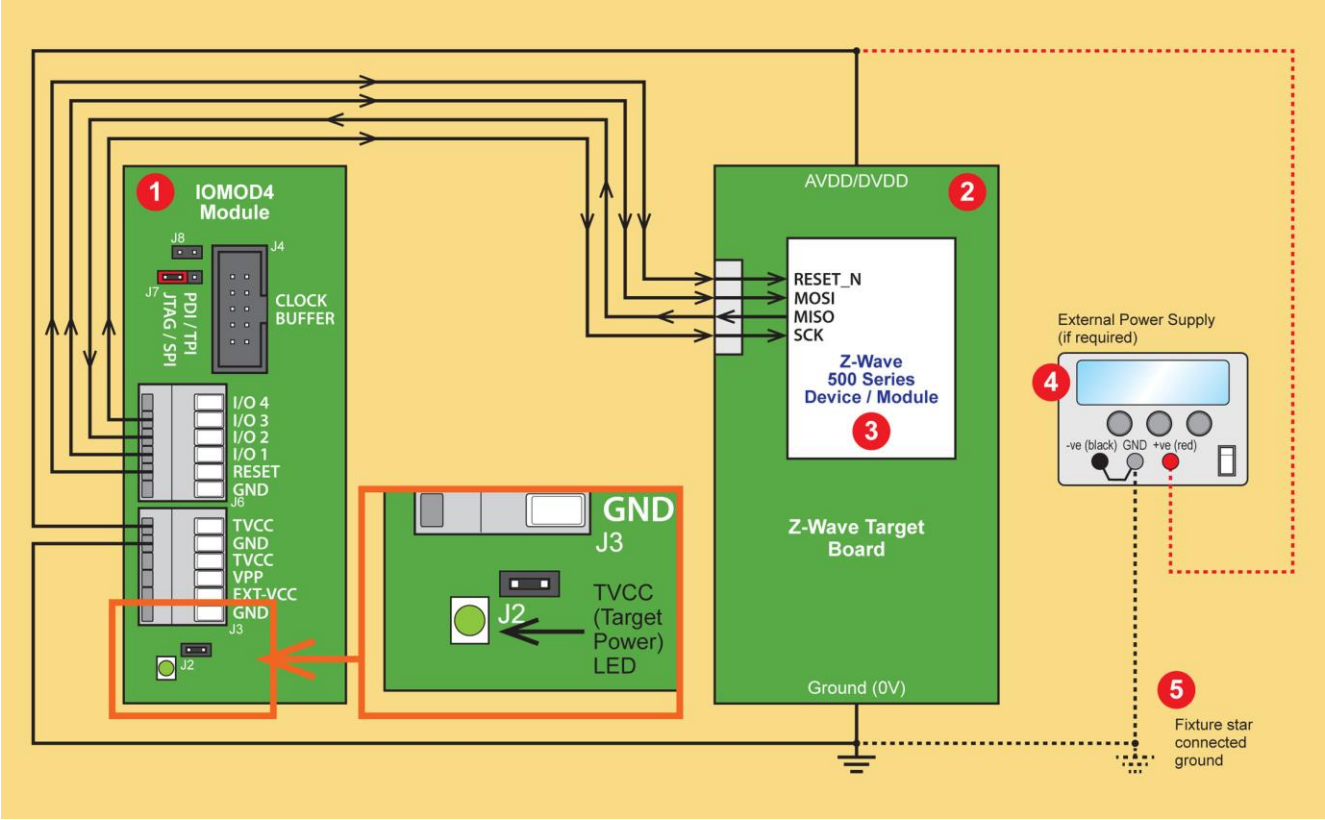
Important notes...

1. It is recommended that separate thick wires are used for the "Power 0V" and "Power VCC" between the programmer and the target board.

2. A separate GROUND connection should be made from the Target Board to the star-connected GROUND point in the fixture.

1.3 IOMOD4 connector module - connections to Z-Wave target board (SPI interface)

The illustration below shows how to connect a 'Z-Wave module / target board' to an ISPnano programmer using an IOMOD4 connector module.



Number	Description
1	IOMOD4 connector module
2	Customer Target Board featuring Z-Wave module or IC
3	Z-wave module or IC
4	External power supply (used to power the Target Board) if the programmer is not supplying power)
5	Fixture main GROUND connection. The Target Board, programmer and External Power Supply should all be star connected to this common GROUND point.

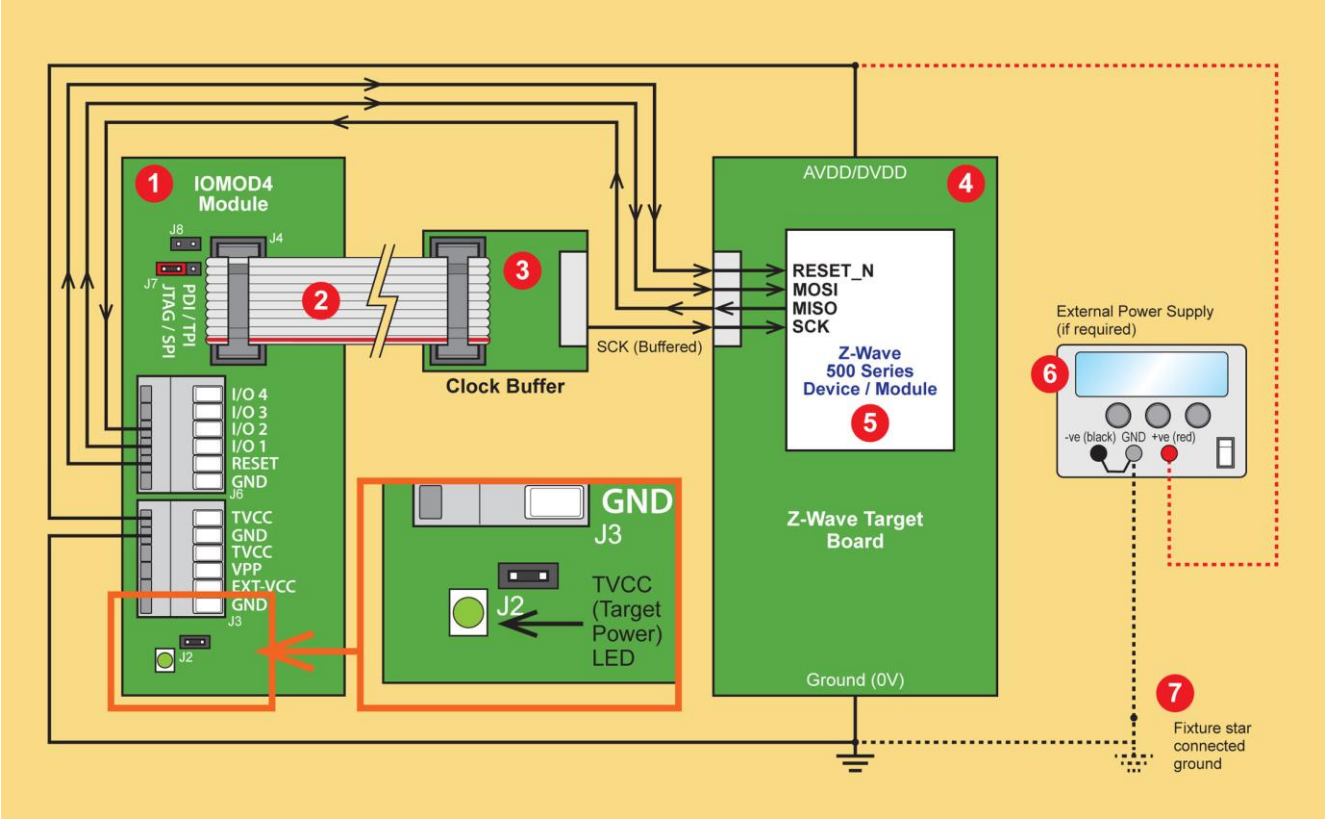
Important notes...

1. It is recommended that separate thick wires are used for the "Power 0V" and "Power VCC" between the programmer and the target board.

2. A separate GROUND connection should be made from the Target Board to the star-connected GROUND point in the fixture.

1.4 IOMOD4 connector module + Clock Buffer - connections to Z-Wave target board (SPI interface)

The illustration below shows how to connect a 'Z-Wave module / target board' to an ISPnano programmer using an IOMOD4 connector module and also a remote 'Clock Buffer'. The 'Clock Buffer' is used to buffer the SPI SCK signal at the target end of the wire to ensure good signal integrity.



Number	Description
1	IOMOD4 connector module
2	10-way ISP cable
3	Clock Buffer
4	Customer Target Board featuring Z-Wave module or IC
5	Z-wave module or IC
6	External power supply (used to power the Target Board) if the programmer is not supplying power)
7	Fixture main GROUND connection. The Target Board, programmer and External Power Supply should all be star connected to this common GROUND point.

Important notes...

1. It is recommended that separate thick wires are used for the "Power 0V" and "Power VCC" between the programmer and the target board.

2. A separate GROUND connection should be made from the Target Board to the star-connected GROUND point in the fixture.