

Report No:

AN128

Title:

In-System Programming (ISP) of NXP LPC2xxx ARM7 FLASH microcontroller family using the JTAG Programming Interface

Author:

John Marriott

Date:

8th August 2011

Version Number:

1.04

All rights are reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner. The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without prior notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent or other industrial or intellectual property rights

Contents

1.0 Introduction	6
1.1 Features	6
1.2 Programmers supporting NXP LPC2xxx devices	8
1.3 Programmer firmware versions for NXP LPC2xxx support.....	9
1.4 Device Support.....	10
1.4.1 Overview	10
1.4.2 NXP LPC214x Family.....	10
1.4.3 NXP LPC213x Family.....	10
1.4.4 NXP LPC210x Family.....	10
1.5 Upgrading your Equinox Programmer to support NXP LPCxxx JTAG Programming	11
1.5.1 Overview	11
1.5.2 Purchasing an NXP LPC2XXX License.....	11
1.5.3 How do I enable the programmer for NXP LPC2XXX JTAG?.....	11
1.5.4 Upgrading an Epsilon5, FS2003 or FS2009 Programmer to support NXP LPC2xxx JTAG programming.....	12
1.5.5 Upgrading a PPM3-MK2 or PPM4-MK1 Programmer to support NXP LPC2xxx JTAG...13	
1.5.6 Entering the License String to upgrade your programmer	14
2.0 NXP LPC2xxx - JTAG Programming Interface	16
2.1 Overview	16
2.2 JTAG Features	17
2.3 JTAG single-chip In-System Programming (ISP) Schematic.....	18
2.4 JTAG Pins considerations	19
2.5 TRST (JTAG TEST Reset) pin.....	19
2.6 RTCK pin.....	19
2.7 JTAG-in-a-chain In-System Programming (ISP) Schematic.....	21
2.8 ARM JTAG Debug Interface (20-way)	22
3.0 Creating an EDS (Development) Project	24
3.1 Overview	24
3.2 Information required to create a NXP LPC2xxx Project.....	24
3.3 Creating an EDS (Development project).....	25
3.4 Selecting the correct Target Device	26
3.4.1 Device selection	26
3.4.2 Silicon revision and Chip ID / Signature.....	27
3.5 Target Oscillator Settings	28
3.6 Target System – Power Supply Settings.....	30
3.7 Specifying the FLASH (Code) File	31
3.8 Launching EDS at the end of the EDS Wizard.....	32
4.0 Testing a Project in Development (EDS) Mode	33
4.1 Introduction to EDS	33
4.2 EDS - Default settings for JTAG, statemachine etc.....	34
4.3 JTAG Settings	35
4.3.1 JTAG frequency	35
4.3.2 JTAG Chain settings	36
4.4 Checking the NXP LPC2xxx – Target Voltage	37
4.5 Testing JTAG communication with the NXP LPC2xxx device	38
4.5.1 Overview	38
4.5.2 Diagnostic Info	39
4.5.3 Possible failure messages.....	39
4.5.4 Error 3044 / 3039 – Failed to enter programming mode	40
4.5.5 Error 41 or 3041 – Signature FAIL message	42
4.6 Erasing the FLASH area.....	43
4.6.1 Overview	43
4.6.2 Erasing the FLASH area	43

4.7 Programming the FLASH Area.....	44
4.11 Exporting an EDS Project to a Standalone Project.....	45
5.0 Creating a Standalone Programming Project.....	46
5.1 Overview.....	46
5.2 Creating a Standalone Project from EDS (Development Mode)	46
5.3 Add Project File to a new Project Collection.....	47
5.4 Uploading a Project to a programmer.....	48
6.5 Re-testing a Project in EDS (Development mode).....	49
Appendix 1 – ARM ISP Cable for Epsilon5, FS2003 and FS2009 programmers.....	50
1.0 Overview.....	50
1.1 Features.....	50
1.2 Programmer compatibility.....	50
1.3 Power Supply - Selection Jumper	51
1.4 Cable Installation Instructions	51
1.5 Getting Started.....	52
1.6 ARM JTAG Debug Header.....	52
Appendix 2 – CONMOD Module + NXP LPCxxx	53
1.0 Overview.....	53

1.0 Introduction

Equinox Technologies manufacture a comprehensive range of programmers suitable for high-speed In-System Programming (ISP) of NXP LPC2xxx FLASH microcontrollers via the '**ARM JTAG Debug Interface**'. This application note describes how to develop and implement **In-System Programming (ISP)** support for the NXP LPC2xxx FLASH microcontroller families using the '**ARM JTAG Debug Interface**'. The document details how to make a JTAG '**Programming Project**' which will operate on any Equinox ISP programmer including a full description of how to implement JTAG In-System Programming (ISP) of the NXP LPC2xxx devices.

1.1 Features

The Equinox programming range includes solutions for development, low / mid / high volume production and field programming of **NXP LPC2xxx** microcontrollers.

General features.....

- High-speed In-System Programming (ISP) support of **NXP LPC2xxx** microcontrollers
- Programming solutions for development, low / mid / high volume production and field programming **of NXP LPC2xxx** microcontrollers
- Programs the on-chip FLASH Memory of NXP LPC2XXX microcontrollers
- Uses industry standard '**ARM JTAG Debug Interface**' port as the ISP interface
- Very high-speed programming due to fast JTAG 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.
- Optimised Erase operations - blanks the on-chip FLASH in less than 200ms
- Supports programming of NXP **LPC2xxx** devices when placed in a '**JTAG Chain**' (JTAG daisy-chain mode)
- 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 **NXP LPC2xxx** device can be controlled from **EDS**
- Program and read back the LPC 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 NXP LPC2xxx '**Standalone Programming Projects**' can be stored inside an FS2003, FS2009, PPM3-MK2, PPM4-MK1, ISPjuno or 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**'

In 'Field Programming Mode'.....

- The Epsilon5, FS2009 and ISPjuno programmers can all be used to re-program an **NXP LPC2xxx** microcontroller based Target System in the field.
- All programmers operate in so-called '**Standalone Mode**' where a PC / laptop is not required after the programmer has been configured.
- Operator uses programmer keypad / Display to select the required '**Standalone Programming Project**' to execute
- Single-key '**Autoprogram button**' – allows the same project to be executed repeatedly
- Ideal for field-service personnel for re-programming vending machines and any other equipment / plant in the field.
- The FS2009 and ISPjuno programmers can store up to 64 x '**Standalone Programming Projects**' so the field operator can select which product firmware to program.

1.2 Programmers supporting NXP LPC2xxx devices

Most Equinox ISP Programmers can be upgraded to support high-speed programming of NXP LPC2xxx microcontrollers via the '**ARM JTAG Debug Interface**'. The table below lists all the Equinox ISP programmers which are capable of programming NXP LPC2XXX microcontrollers.

Programmer	NXP LPC2xxx Support	Requirements	Upgrade Order Code
EPSILON5(UN)	Upgrade	License upgrade + CAB-ARM7-20W ISP Cable	EPSILON5-UPG27
FS2003(UN)	Upgrade	License upgrade + CAB-ARM7-20W ISP Cable	FS2003-UPG27
FS2009(UN)	Upgrade	License upgrade + CAB-ARM7-20W ISP Cable	FS2009-UPG27
PPM3-MK2	Upgrade	License upgrade + EQ-IO-CON-10 Connector Module + EQ-SFM-MAX-V1.3 High-speed I/O Driver Module	PPM3A1-UPG27
PPM4-MK1	Upgrade	License upgrade + EQ-IO-CON-10 Connector Module + EQ-SFM-MAX-V1.3 High-speed I/O Driver Module	PPM4MK1-UPG27
ISPnano Series 1	Upgrade	License upgrade	ISPNano-UPG27
ISPnano Series 2	Upgrade	License upgrade	ISPNano-UPG27
ISPnano Series 3	Upgrade	License upgrade	ISPNano-UPG27
ISPnano Series 3 ATE	Upgrade	License upgrade	ISPNano-UPG27
ISPnano Series 4	Upgrade	License upgrade	ISPNano-UPG27
ISPnano-MUX8	Upgrade	License upgrade	ISPNano-UPG27
ISPnano-MUX4	Upgrade	License upgrade	ISPNano-UPG27
ISPnano-MUX2	Upgrade	License upgrade	ISPNano-UPG27

Please note:

- A chargeable '**License Upgrade**' is required to enable the **NXP LPC2xxx** support on any of these programmers.
- A suitable '**NXP LPC2XXX ISP Connector Cable**' is also required to connect from the programmer to the **NXP LPC2xxx** Target Board. This is provided with the **License Upgrade**.
- The **programmer firmware** will probably also need to be upgraded in order to support **NXP LPC2xxx** device programming – see section 1.3.
- It is also recommended that EQTools build 1156 or above is used when programming **NXP LPC2xxx** devices.

1.3 Programmer firmware versions for NXP LPC2xxx support

Most Equinox ISP Programmers can be upgraded to support high-speed programming of NXP LPC2xxx microcontrollers via the **'ARM JTAG Debug Interface'**. The table below lists all the Equinox ISP programmers which are capable of programming NXP LPC2xxx microcontrollers. A chargeable **'License Upgrade'** is required to enable the NXP LPC2xxx support on any of these programmers. A suitable **'NXP LPC2XXX ISP Connector Cable'** is also required to connect from the programmer to the NXP LPC2XXX Target Board. This is supplied with the License Upgrade.

Fig. 1.3 Programmer firmware versions for NXP LPC2XXX JTAG In-System Programming Support

Programmer	NXP LPC2XXX Support
EPSILON5	TBD
FS2003	TBD
FS2009	TBD
PPM3-MK2	TBD
PPM4-MK1	TBD
ISPnano Series I/II	TBD
ISPnano Series III	7.01
ISPnano Series IV	Please contact Equinox
ISPnano-MUX 2 / 4 / 8	Please contact Equinox

Please note:

- Due to limited firmware storage space on the FS2003 and PPM3-MK2 programmers, these programmers cannot support both the NXP LPC2xxx firmware and other microcontrollers at the same time.
- It is necessary to upload a special version of firmware for programming NXP LPC2xxx devices which will then not support programming of any other devices e.g. Atmel AVR microcontrollers.
- It is possible to switch between the conventional firmware and **'NXP LPC2xxx'** firmware at any time by simply uploading the relevant **'Firmware upgrade project'** to the programmer.

1.4 Device Support

1.4.1 Overview

There are various different families of NXP LPC2xxx microcontrollers, each with a different performance / peripheral mix. The Device Support is now split into a different section for each different family.

1.4.2 NXP LPC214x Family

The NXP LPC214x Family contains the following devices:

Device	On-chip FLASH Size (kb)	FLASH Page Size (bytes)	Minimum firmware version	Programming Support available	Tested on actual device
LPC2141	32	256	7.00	YES	-
LPC2142	64	256	7.00	YES	-
LPC2144	128	256	7.00	YES	-
LPC2146	256	256	7.00	YES	-
LPC2148	500*	256	7.00	YES	YES

* This device actually features 512 kb of the FLASH but the top 12 kb are used for the Boot Loader and so cannot be used for user code.

1.4.3 NXP LPC213x Family

The NXP LPC213x Family contains the following devices:

Device	On-chip FLASH Size (kb)	FLASH Page Size (bytes)	Minimum firmware version	Programming Support available	Tested on actual device
LPC2131	32	256	7.00	YES	-
LPC2132	64	256	7.00	YES	-
LPC2134	128	256	7.00	YES	-
LPC2136	256	256	7.00	YES	-
LPC2138	500*	256	7.00	YES	YES

* This device actually features 512 kb of the FLASH but the top 12 kb are used for the Boot Loader and so cannot be used for user code.

1.4.4 NXP LPC210x Family

The NXP LPC210x Family contains the following devices:

Device	On-chip FLASH Size (kb)	FLASH Page Size (bytes)	Minimum firmware version	Programming Support available	Tested on actual device
LPC2101	8	256	7.00	YES	-
LPC2102	16	256	7.00	YES	-
LPC2103	32	256	7.00	YES	YES

1.5 Upgrading your Equinox Programmer to support NXP LPCxxx JTAG Programming

1.5.1 Overview

The NXP LPC2xxx JTAG algorithms are not supported as standard on any Equinox programmers. It is necessary to purchase a '**License Upgrade**' for '**NXP LPC2xxx JTAG**' support from Equinox. Equinox will then send you a '**JTAG Upgrade License String**' which will upgrade your programmer to support programming of this device family.

1.5.2 Purchasing an NXP LPC2XXX License

All Equinox ISP programmers require the purchase of a '**License Upgrade**' to enable NXP LPC2xxx JTAG 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 NXP LPC2XXX JTAG?

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

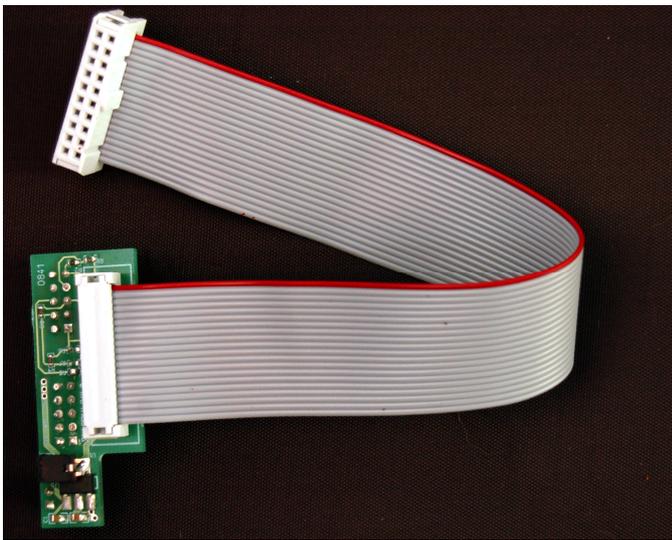
1. **If you purchase the upgrade directly from Equinox**
 - Equinox will email you a 'JTAG License String'.
 - This string can be entered directly into the **<Enter License>** screen in EQTools.

2. **If you purchase the upgrade from a distributor**
 - The distributor will send you the Upgrade Pack by courier.
 - Within the Upgrade Pack you will find an Upgrade Form with a Code String on it.
 - Email this Code String plus your programmer '**Serial Number**' to support@equinox-tech.com
 - Equinox will then send you a '**NXP LPC2xxx JTAG License String**' which is keyed to your programmer Serial Number.
 - This string can be entered directly into the **<Enter License>** screen in EQTools.

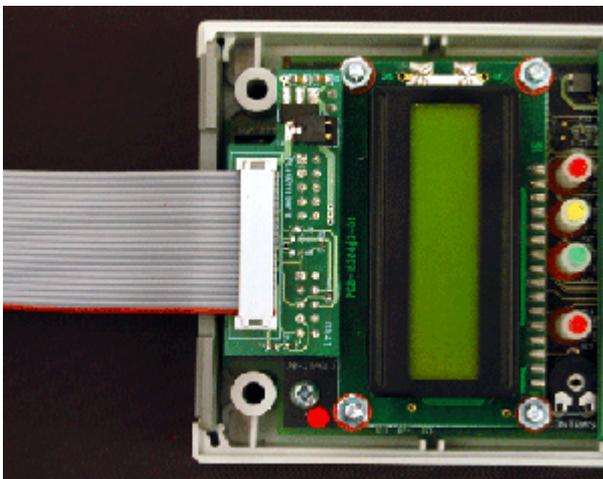
1.5.4 Upgrading an Epsilon5, FS2003 or FS2009 Programmer to support NXP LPC2xxx JTAG programming

To upgrade an Epsilon5, FS2003 or FS2009 programmer to support NXP LPC2XXX JTAG programming, please follow the steps below:

- Order an '**NXP LPC2XXX JTAG License**' from Equinox
- Enter the '**NXP LPC2XXX Upgrade License String**' given to you by Equinox into EQTools
- Make sure you have the required version of '**Programmer Firmware**' to support the device you wish to program – see section 1.2.
- You will also receive a special '**ARM Programming Cable**' with this upgrade.
See Appendix 1 for full details of how to connect this cable to your programmer.



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

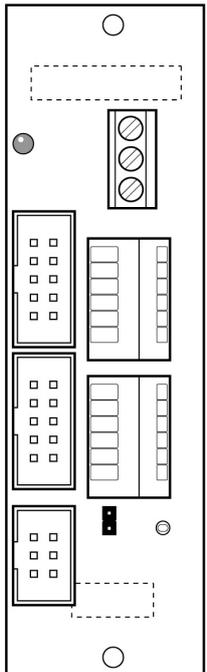


- Plug the 20-way IDC plug end of the cable into the mating IDC socket on your Target System.
- You are now ready to program an NXP LPC2XXX device via JTAG.

1.5.5 Upgrading a PPM3-MK2 or PPM4-MK1 Programmer to support NXP LPC2xxx JTAG

To upgrade a PPM3-MK2 or PPM4-MK1 programmer to support NXP LPC2xxx JTAG, please follow the steps below:

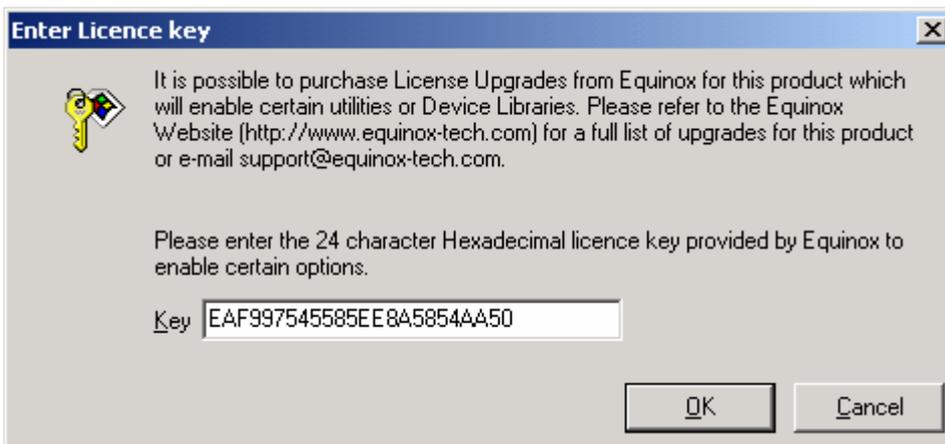
- Order a '**NXP LPC2xxx JTAG upgrade**' from Equinox Technologies
- Enter the '**JTAG Upgrade License String**' given to you by Equinox into EQTools
- The JTAG upgrade also includes a new '**I/O Connector Module**' for the PPM4-MK1 called the '**EQ-IO-CON-10**'. This module features a 20-way JTAG header for connection to the NXP LPC2XXX Target Board. It also includes the High Speed / High Current Special Function Module '**EQ-SFM-MAX-V1.3**'. This significantly reduces the overall programming times for many high capacity devices. For full instructions on fitting this module please refer to **Application Note AN115** provided with your upgrade
- Make sure you have the required version of Programmer Firmware to support the device you wish to program – see section 1.2.
- Plug the '**EQ-IO-CON-10**' module into the PPM3-MK2 / PPM4-MK1 programmer
- Plug the 20-way ISP cable supplied with the programmer into the '**JTAG**' ISP Header on the '**EQ-IO-CON-10**' module.
- Connect the other end of the 20-way ISP cable to the JTAG port on your Target Board
- You are now ready to program a Target Chip via JTAG

 <p>EQ-IOCON-10</p>	<p>I/O Connector Module 10 (NXP LPC2XXX JTAG) – Fast Connect Version I/O connector module for In-System Programming (ISP) of NXP LPC2xxx microcontrollers using JTAG protocol</p> <p>Features:</p> <ul style="list-style-type: none"> • Plugs into suitable Equinox programmer e.g. PPM4-MK1 • Generic 20-way ARM JTAG IDC connector • Single-in-line header with all programmer I/O brought out for wire-wrapping to bed-of-nails probe wires • Screw terminals for power connections • Target Vcc Status LED • Link to connect / isolate the programmer Vcc from the Target Vcc
---	---

1.5.6 Entering the License String to upgrade your programmer

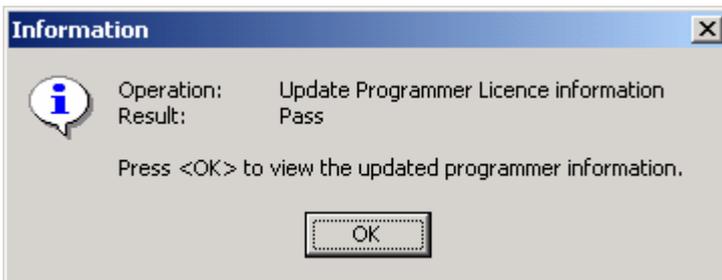
Once you have received the License String from Equinox, please follow the steps below to apply the upgrade to your programmer:

- Launch EQTools → The EQTools 'Welcome Screen' is displayed.
- Close down the EQTools 'Welcome Screen'
- From the top menu bar, select **<Programmer><Programmer Info>**
→ the Programmer Information screen is displayed
- Click the **<Enter License>** button
→ The **<Enter License Key>** screen is displayed.



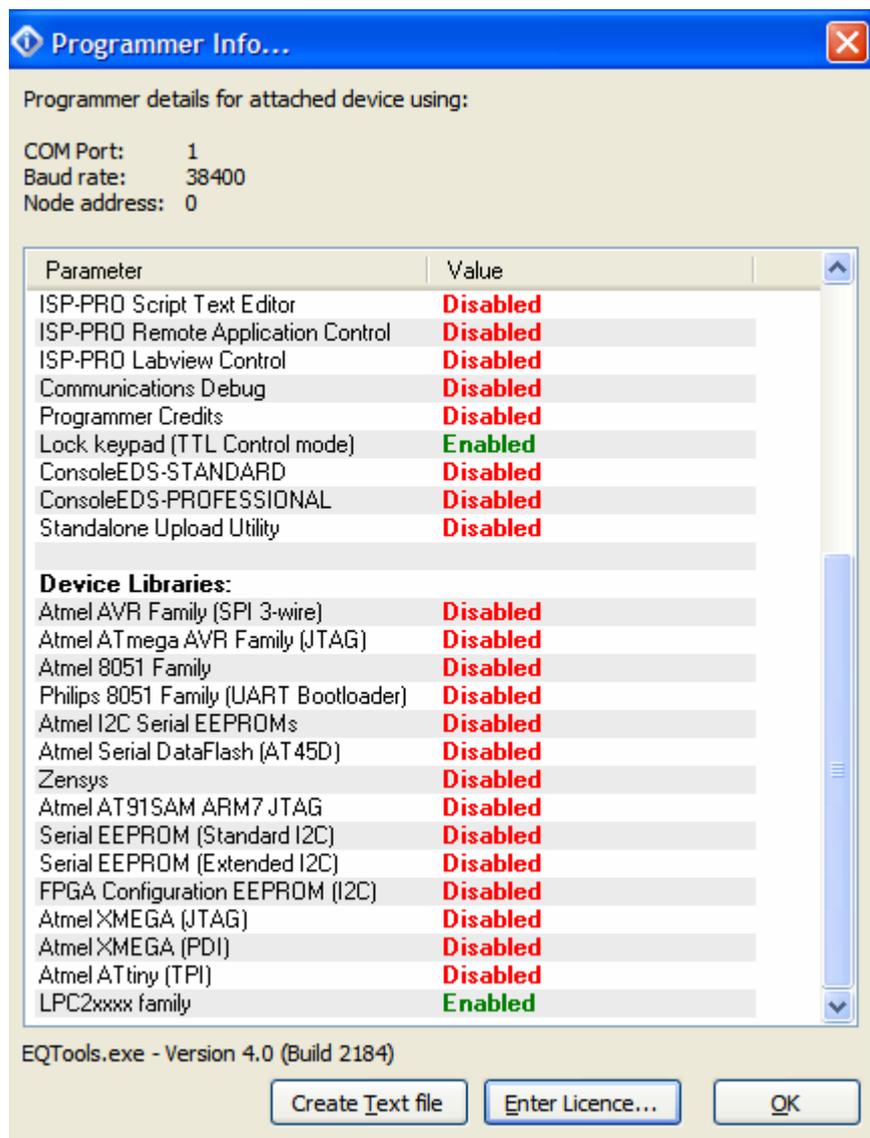
Enter the License String you were sent by Equinox

- 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 **'NXP LPC2XXX ARM7 JTAG'** is now ENABLED.

The Programmer Info screen should display *'LPC2xxx Family'* enabled – see screenshot below.



2.0 NXP LPC2xxx - JTAG Programming Interface

2.1 Overview

There are a number of different methods of programming the on-chip FLASH of an NXP LPC2xxx microcontroller as detailed in the table below.

Interface		
USB	USB Interface	Uses the NXP Boot Loader stored in protected FLASH.
JTAG	ARM Debug Port	Uses the ARM Debug Port for both In-System Debugging and Programming.
UART	UART interface	Uses the NXP Boot Loader stored in protected FLASH to upload firmware via a 2-wire UART interface. This is usually implemented via an RS232 link to a PC.

The Equinox range of programmers use the '**ARM Debug JTAG Interface**' for programming NXP LPC2xxx devices. This interface provides a method for both **In-System Debugging (ISD)** and **In-System Programming (ISP)** of NXP LPC2xxx Microcontrollers. The JTAG Interfaces uses an industry-standard set of signals to provide the connection between the programmer / debugger and the NXP LPC2xxx microcontroller. The actual JTAG Header (connector) used by NXP and Equinox is the standard 20-pin JTAG connector used for most popular ARM7 microcontrollers.

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

In a production environment, the JTAG Interface can be used for high-speed **In-System Programming (ISP)** of the Target Microcontroller. This method of operation requires the use of any Equinox ISP Programmer which has been enabled to support the '**NXP LPC2xxx JTAG**' algorithms.

The Equinox ISP Programmer range supports high-speed **In-System Programming (ISP)** of a single or multiple NXP LPC2xxx microcontrollers on a Target Board using the 4-wire JTAG bus. Support has now been added for programming of any NXP LPC2xxx microcontroller when connected as part of a '**JTAG Chain**'. This mode allows multiple JTAG devices to be in-system programmed using a single JTAG bus.

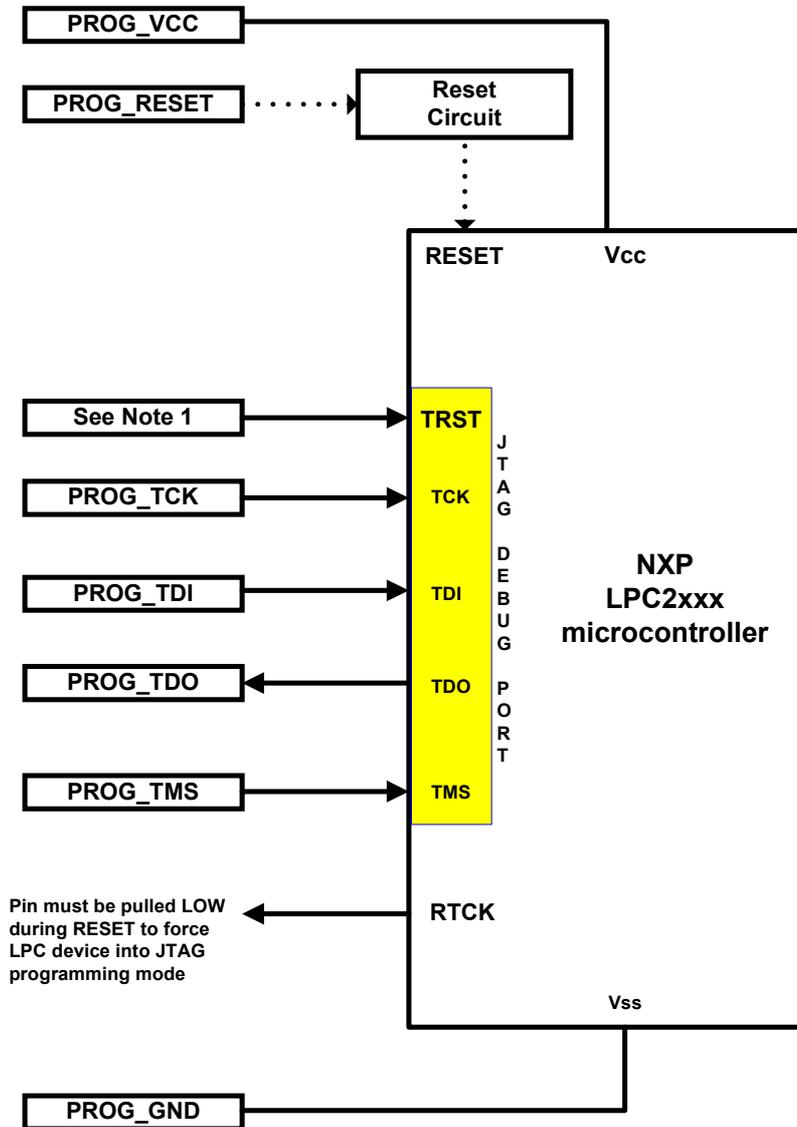
2.2 JTAG Features

- Fast Programming speeds
- Simple 4-wire JTAG bus connection + RESET signal
- JTAG programming does not depend on the NXP LPC2xxx oscillator frequency so JTAG programming will always work
- JTAG interface is compatible with any third party In-System Debugger so same interface can be used for development and production.
- Both '**Single microcontroller**' and '**JTAG-in-a-chain**' implementations are supported

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

The diagram below details the connections required to implement JTAG In-System Programming of a single NXP LPC2xxx microcontroller using an Equinox ISP programmer.

Fig 2.3 –NXP LPC2xxx – JTAG Programming Interface connections



NXP LPC2XXX – JTAG Programming Interface - signal names and directions

Programmer Signal Name	Signal description	Signal direction (from Programmer)	Connect to NXP LPC2XXX Microcontroller Pin	Signal direction (from Microcontroller)
PROG_TCK	Test Clock Pin	Output	TCK	Input
PROG_TDI	Test Data Input	Output	TDI	Input
PROG_TDO	Test Data Output	Input	TDO	Output
PROG_TMS	Test Mode Select	Output	TMS	Input
PROG_RESET	RESET	Output	RESET	Input
See section 2.5	JTAG Test RESET	Pulled HIGH	TRST	See note 1
See section 2.6	RTCK – enable JTAG	Pulled LOW	RTCK	I/O

2.4 JTAG Pins considerations

The JTAG port on an NXP LPC2XXX device is made up of the following pins: TDI, TDO, TCK, TMS, TRST and RESET. On the NXP LPC2xxx Target System, it is recommended that a pull-up resistor to the main microcontroller Vcc (e.g. 3.3V) is placed on the TCK, TMS and TDI pins. The TDO pin can be left floating.

Programmer Signal Name	Signal description	Signal direction (from Programmer)	Pull-up resistor on Programmer Interface Cable	Pull-up resistor on Target System
PROG_TCK	Test Clock Pin	Output	47k ohm	47k – 100k ohm
PROG_TDI	Test Data Input	Output	47k ohm	47k – 100k ohm
PROG_TDO	Test Data Output	Input	47k ohm	Floating
PROG_TMS	Test Mode Select	Output	47k ohm	47k – 100k ohm
See section 2.5	JTAG Test RESET	Pulled HIGH	TRST	Usually floating

Please note:

- The Equinox '**ARM ISP Cable**' which connects the Equinox Epsilon5, FS2003 and FS2009 programmers to the NXP LPC2XXX Target System features 47k ohm pull-up resistors on the TCK, TMS and TDI pins. These resistors can be removed if required.
- The Equinox '**EQ-IOCON-10**' I/O Connector Module which connects the Equinox PPM3-MK2 and PPM4-MK1 programmers to the NXP LPC2XXX Target System features 47k ohm pull-up resistors on the TCK, TMS and TDI pins. These resistors can be removed if required.

2.5 TRST (JTAG TEST Reset) pin

The **TRST** pin is a dedicated JTAG pin used by the programmer to force a hardware reset of the '**JTAG hardware state machine**' inside the target ARM microcontroller. The **TRST** pin is different from the RST (RESET) pin which forces a full reset of the entire LPC2xxx device. The **TRST** pin is used by some debuggers to halt / break the ARM core execution without resetting the entire device.

Please note:

The **TRST** pin is simply tied HIGH with a pull-up resistor on all Equinox programmers. It is not actually driven by the programmer.

2.6 RTCK pin

The **RTCK** pin on an LPC2xxx microcontroller must be pulled LOW when the device comes out of the reset condition (Power-on-RESET or manual reset condition) otherwise an external programmer cannot force the device into '**Debug Mode**' using the '**JTAG Interface**'. It will therefore be impossible to program the device via JTAG.

Equinox programmers do **NOT** actively drive the **RTCK** pin, so it is recommended that the **RTCK** pin is configured so that it can be pulled LOW on the Target Board for programming / debug purposes.

This can be achieved by using one of the following techniques:

- Use a pull-down resistor from the **RTCK** pin to GROUND.
- Use a jumper link which connects the **RTCK** pin to GROUND using a pull-down resistor. The jumper link must be fitted when programming or debugging via the '**JTAG Interface**'.
- Using '**Test Pads**' which connect to two '**Test Probes**' on the programming fixture which then connect the **RTCK** pin to GROUND via a pull-down resistor.

Important notes

1. ARM ISP Cable – remove resistor R6

If you are using the Equinox '**ARM ISP Cable**', please check that the resistor R6 is removed from the circuit board. This resistor is connected to a 47k ohm pull-up to Vcc and will pull the **RTCK** pin HIGH preventing the NXP LPC2xxx device from entering '**Debug Mode**' using the '**JTAG Interface**'.

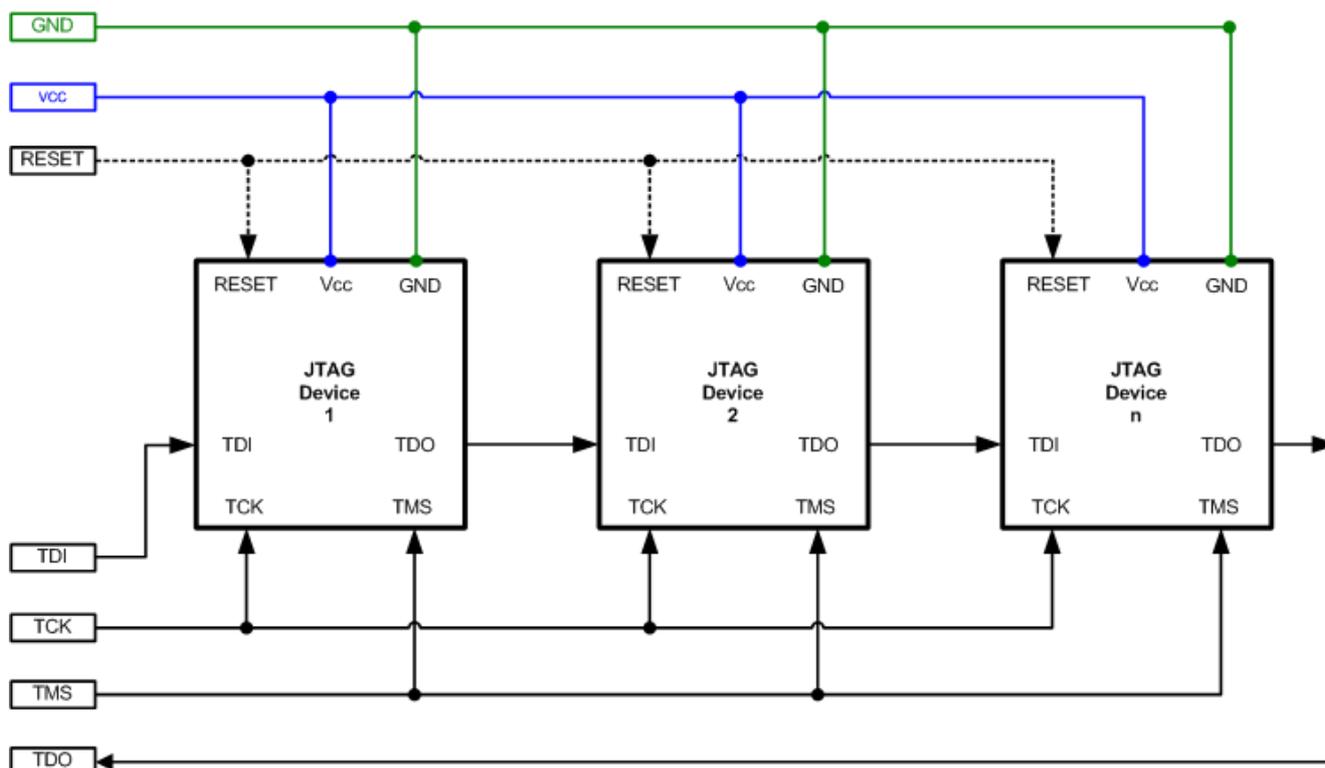
2. ISPnano 'CONMOD' connector board – remove resistor R6

If you are using the Equinox '**ISPnano CONMOD connector board**', please check that the resistor R6 is removed from the circuit board. This resistor is connected to a 47k ohm pull-up to Vcc and will pull the **RTCK** pin HIGH preventing the NXP LPC2xxx device from entering '**Debug Mode**' using the '**JTAG Interface**'.

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

The diagram below details the connections required to implement JTAG In-System Programming of either a single or multiple NXP LPC2xxx Microcontrollers which are connected in a 'JTAG Chain' arrangement.

Fig 2.6 – NXP LPC2XXX – JTAG Programming Interface connections



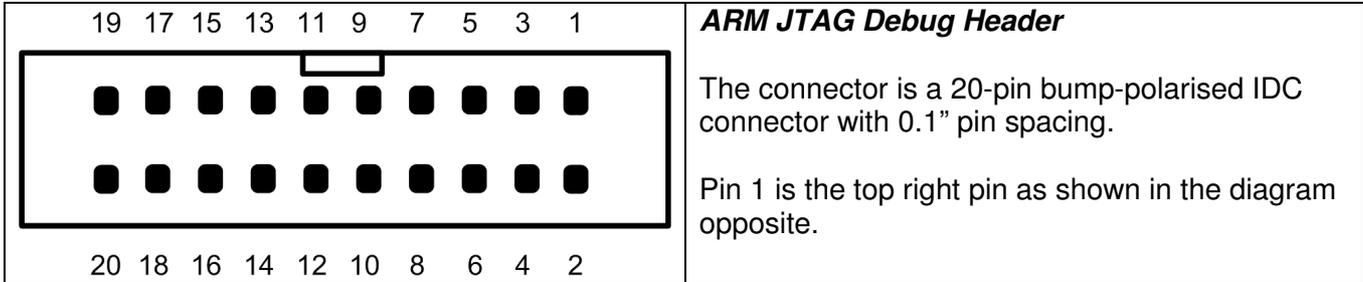
The TDI signal is fed into the TDI input on the first JTAG Device in the chain. The data path then goes through the first device and comes out on the TDO pin. The TDO pin is connected to the TDI pin of the next JTAG Device in the chain.

NXP LPC2XXX – JTAG Programming Interface - signal names and directions

Programmer Signal Name	Signal description	Signal direction (from Programmer)	Connect to NXP LPC2XXX Microcontroller Pin	Signal direction (from Microcontroller)
PROG_TCK	Test Clock Pin	Output	TCK	Input
PROG_TDI	Test Data Input	Output	TDI	Input
PROG_TDO	Test Data Output	Input	TDO	Output
PROG_TMS	Test Mode Select	Output	TMS	Input
PROG_RESET	RESET	Output	RESET	Input

2.8 ARM JTAG Debug Interface (20-way)

All Equinox NXP ARM ISP programmer solutions offer the use of the standard 20-way '**ARM JTAG Debug Header**' to connect to an NXP LPC2xxx Target System. This connector is a 20-way 2.54mm female IDC (Insulation Displacement Connector).



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

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

			Target System.
11	RTCK	No connect	<p>Input Return Test Clock signal from Target System This signal is not used by Equinox programmers. For NXP LPC2xxx devices, this pin should be pulled LOW on the target system during reset to force the device into JTAG programming mode.</p> <p>Please check that the RTCK pin is disconnected from the programmer as follows:</p> <ul style="list-style-type: none"> • ARM ISP Cable – remove resistor R6 • ISPnano 'CONMOD' connector board – remove resistor R6
12	PROG_GND	Passive	<p>Ground Connection Common ground connection between Programmer and Target System.</p>
13	PROG_TDO	Input	<p>JTAG TDO – Test Data Output pin JTAG data output from Target device JTAG port to programmer. Typically connected to TDO pin on NXP LPC2XXX device.</p>
14	PROG_GND	Passive	<p>Ground Connection Common ground connection between Programmer and Target System.</p>
15	PROG_RESET	O	<p>Microcontroller RESET control signal This pin connects to the main RESET pin of the Target NXP LPC2xxx microcontroller or to the NXP LPCxxx RESET circuit on the Target System.</p>
16	PROG_GND	Passive	<p>Ground Connection Common ground connection between Programmer and Target System.</p>
17	-	No connect	No connect to Equinox Programmer
18	PROG_GND	Passive	<p>Ground Connection Common ground connection between Programmer and Target System.</p>
19	-	No connect	No connect to Equinox Programmer
20	PROG_GND	Passive	<p>Ground Connection Common ground connection between Programmer and Target System.</p>

3.0 Creating an EDS (Development) Project

3.1 Overview

This section describes how to make a '*Programming Project*' for an NXP LPC2xxx device.

Please note:

The following versions of EQTools and firmware are required to support NXL LPC2xxx programming:

- EQTools version 4 build 2186 or higher
- Firmware 7.01 or higher

3.2 Information required to create a NXP LPC2xxx Project

The following information is required about the Target Board in order to create an NXP LPC2xxx JTAG Programming Project:

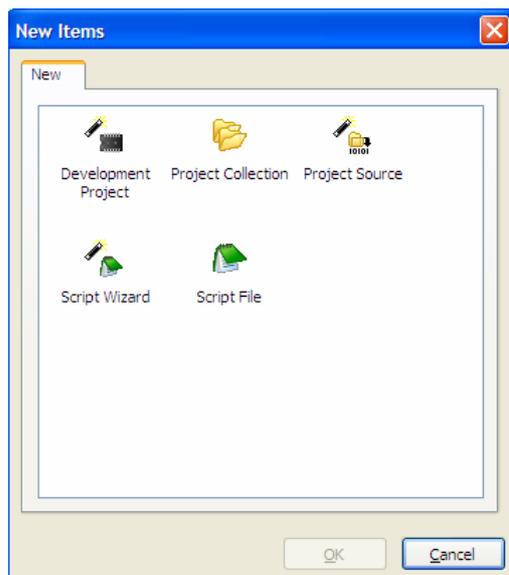
#	Information / data required	Example
1	NXP LPC2XXX Device part number	NXP LPC2XXX
2	JTAG connections / connector on Target board	20-way IDC connector
3	JTAG configuration	i. Single JTAG device or ii. JTAG device is part of a 'JTAG chain'
4	JTAG chain configuration parameters These parameters are required if the device to be programmed is part of a 'JTAG chain'. If a single device is to be programmed via JTAG, then simply set all the 'JTAG Chain' parameters to '0'.	<ul style="list-style-type: none"> • Devices before: 0 • Devices after: 0 • Bits before: 0 • Bits after: 0
5	Target oscillator configuration to be used during programming – see appendix 3	<ul style="list-style-type: none"> • External crystal oscillator • External clock signal
6	Target external device oscillator frequency	e.g. 12 MHz
7	Target System Vcc voltage	e.g. 3.3V
8	Target System maximum current consumption	e.g. 100mA
9	FLASH area 'Program File'	Binary (*.bin) or Intel Hex (*.hex)
12	Reset circuit parameters	e.g. <ul style="list-style-type: none"> • Capacitor / Resistor circuit • Watchdog supervisor circuit • Voltage monitoring circuit

3.3 Creating an EDS (Development project)

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

The steps required to create a project are as follows:

- Click the '**New**' icon on the task bar
- The '**New items**' screen will be displayed...



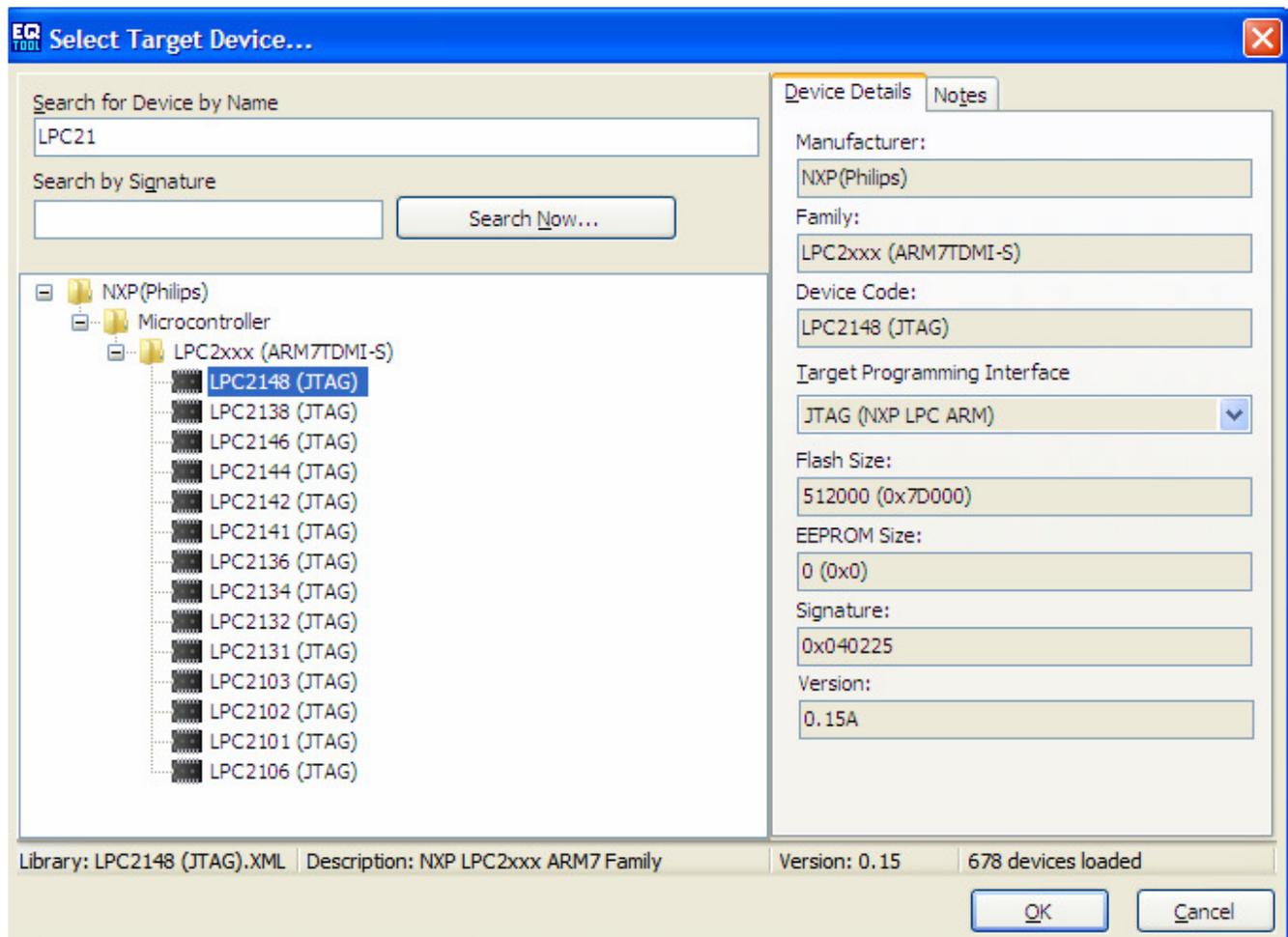
- Select **<Development Project>** and click **<OK>**
- The EDS (Development) Wizard will launch
- Select the relevant 'Programmer' and then click **<Next>**

3.4 Selecting the correct Target Device

It is important to select the correct '**Target Device**' when programming an NXP LPC2XXX device. The part number of the device should be printed on the top of the chip e.g. '**NXP LPC2xxx**'.

3.4.1 Device selection

- Click <Next> → the <Select Target Device> screen will be displayed.
- Type in the '**Device Part Number**' e.g. '**NXP LPC2**' into the '**Search for Device**' field → a list of all matching devices will be displayed in the box underneath.



- Select the required device from the list e.g. '**LPC2148**' and then click <OK> → The device is now selected.

3.4.2 Silicon revision and Chip ID / Signature

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

Target Device

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



Selected Device

Manufacturer:	Family:	Device Code:	
<input type="text" value="NXP(Philips)"/>	<input type="text" value="LPC2xxx (ARM7TDMI-S)"/>	<input type="text" value="LPC2148 (JTAG)"/>	<input type="button" value="Select Device..."/>
Device Algorithm Version:	Silicon Revision:	Signature:	
<input type="text" value="0.15A"/>	<input type="text" value="63"/>	<input type="text" value="0x040225"/>	
Flash Size:	Flash Page:		
<input type="text" value="512000 (0x7D000)"/>	<input type="text" value="256"/>		
Target Programming Interface			
<input type="text" value="JTAG (NXP LPC ARM)"/>			
<input checked="" type="checkbox"/> Check Signature	<input checked="" type="checkbox"/> Read JTAG ID:	Flash Sectors:	Part ID:
<input type="text" value="0x040225"/>	<input type="text" value="0x4F1F0F0F"/>	<input type="text" value="27"/>	<input type="text" value="0x0402FF25"/>

- The project is set to automatically read and validate the '**JTAG ID**' of the Target Device by default.
- The actual '**Chip ID**' for the NXP LPC2xxx device being programmed can be found in the User Manual for the device. Alternatively, it can be read from the target device using EDS.

<input checked="" type="checkbox"/> Check Signature	<input checked="" type="checkbox"/> Read JTAG ID:
<input type="text" value="0x0D0940"/>	<input type="text" value="0x0F0F0F0F"/>

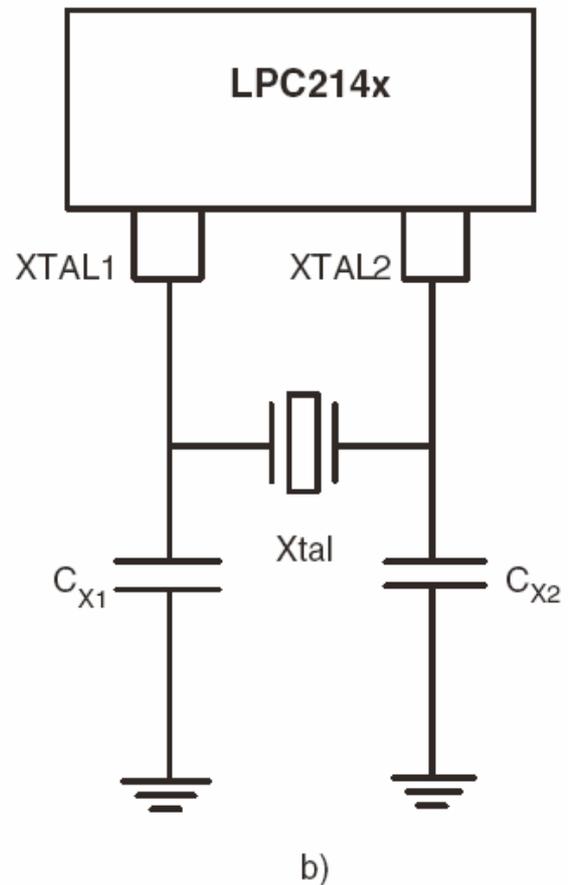
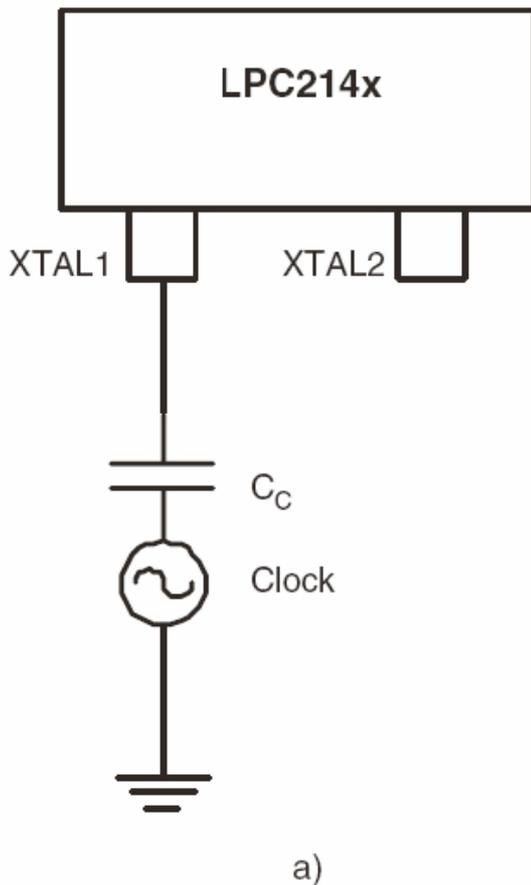
- If you are unsure of the actual '**Chip ID**' for the NXP LPC2xxx device being programmed, select the nearest device for now and then read the actual '**Chip ID**' back using EDS.
- Click **<Next>** to advance to the next screen

3.5 Target Oscillator Settings

When creating a '**Programming Project**' for an NXP LPC2xxx device using the Equinox EQTools software, it is necessary to specify the correct '**Target Oscillator frequency**' in the '**Target Oscillator Settings**' section of the project. The '**Target Oscillator frequency**' is used to calculate the correct '**FLASH timing parameters**' for the NXP LPC2xxx device which are then used during the programming process. The programmer requires the fastest possible oscillator in order to program the FLASH as fast as possible.

Supported oscillator sources:

- External oscillator signal (a)
- External crystal oscillator (b)



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

Target ARM Device - Oscillator Settings 

Select the 'Clock Source' to be used during programming

Target ARM Device - Oscillator Settings

Target Clock Mode

External crystal (XTAL) - specify frequency

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

Target Oscillator Frequency

Please select 

Min	Max
<input type="text" value="1.0 MHz"/>	<input type="text" value="50.0 MHz"/>

**Enter Target
Oscillator Frequency
here**

Instructions

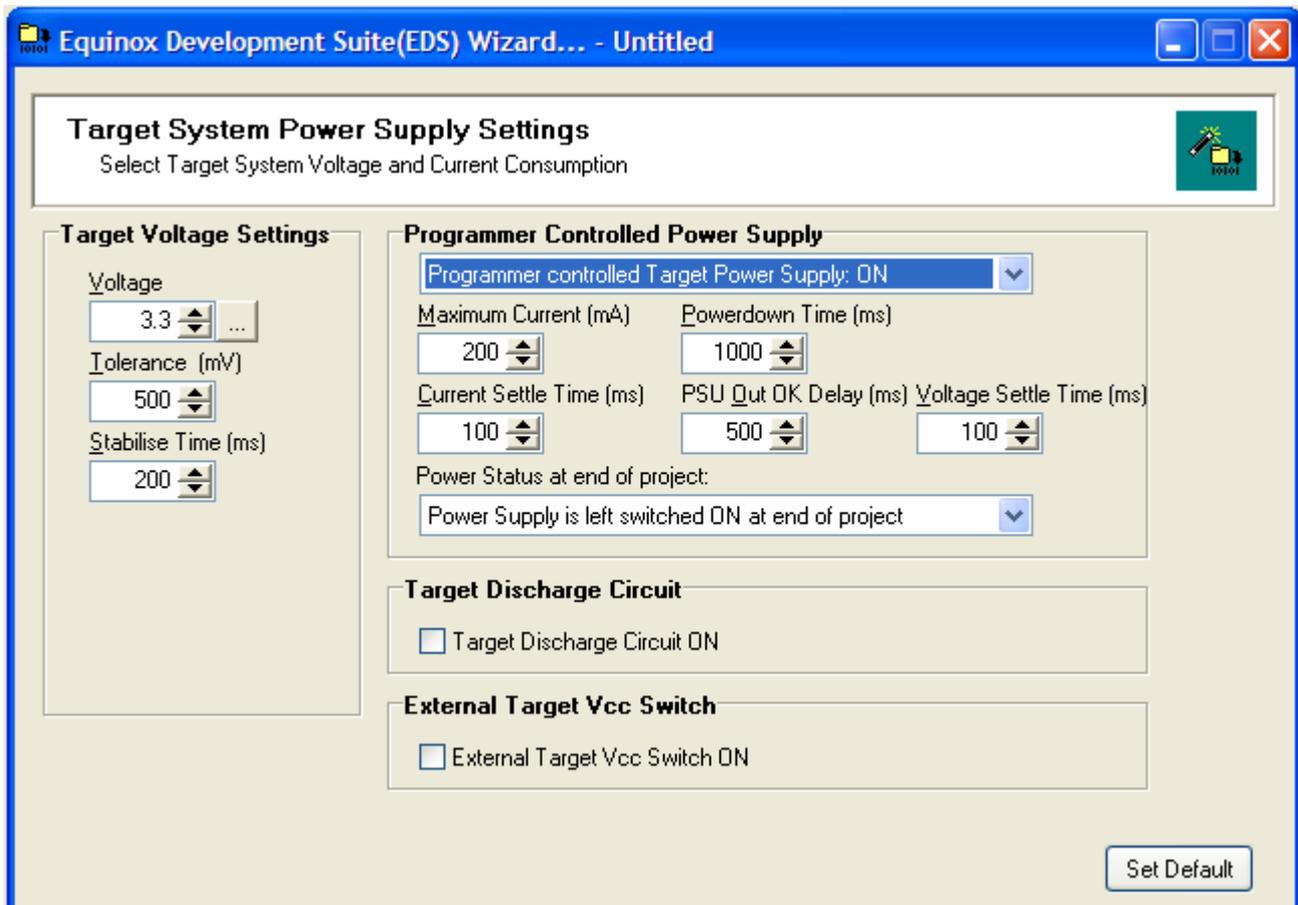
- If your Target Board features a '*crystal oscillator*' connected to the LPC2xxx device, check the frequency value printed on the can of the crystal.
- If your Target Board features an '*external oscillator*' connected to the LPC2xxx device, check the frequency either from the circuit schematic or measure the frequency with a frequency counter / oscilloscope.
- Enter the frequency into the '*Target Oscillator frequency*' field.
- Click **<Next>** to continue to the next screen.

Warning!

Specifying the incorrect '*Target Oscillator frequency*' could damage the FLASH memory of the target LPCxxx device.

3.6 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

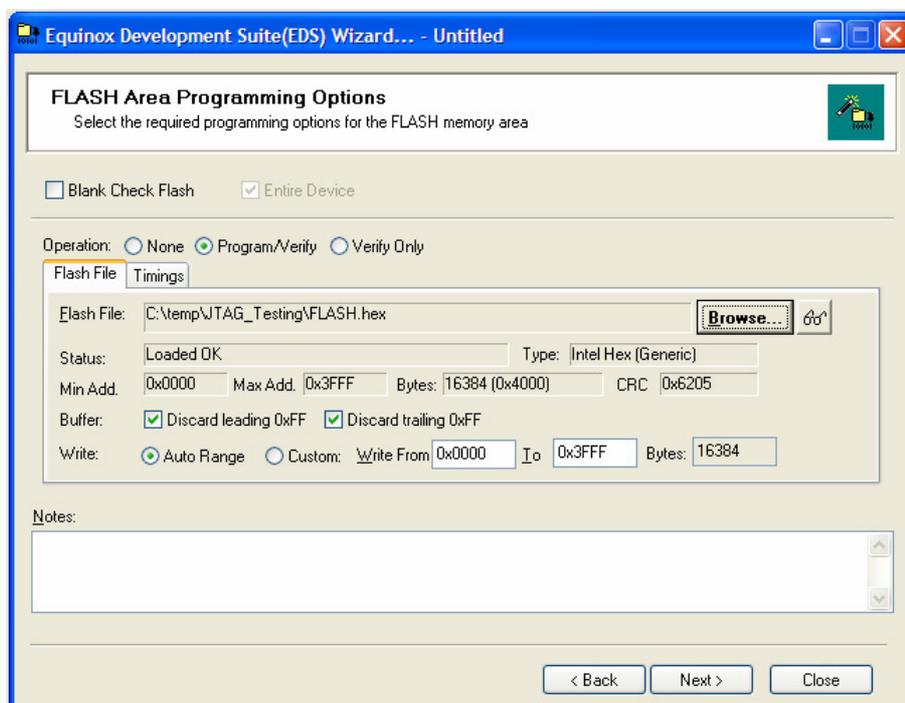
- This should be the voltage at which the Target Device itself e.g. NXP LPC2xxx IC is being powered at during the programming operation. This is usually 3.0 – 3.6V.
- Set the '**Voltage Tolerance**' to be as wide as possible e.g. 500mV to allow for power supply variations. If the programmer is powering the Target System, this will also give a faster power-up time.
- It may also be possible to power the entire Target System by feeding in a higher voltage e.g. +5V into the power supply input on the Target System.

ii. Set up the Target Powering and current parameters

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

3.7 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).



i. Blank Check the FLASH

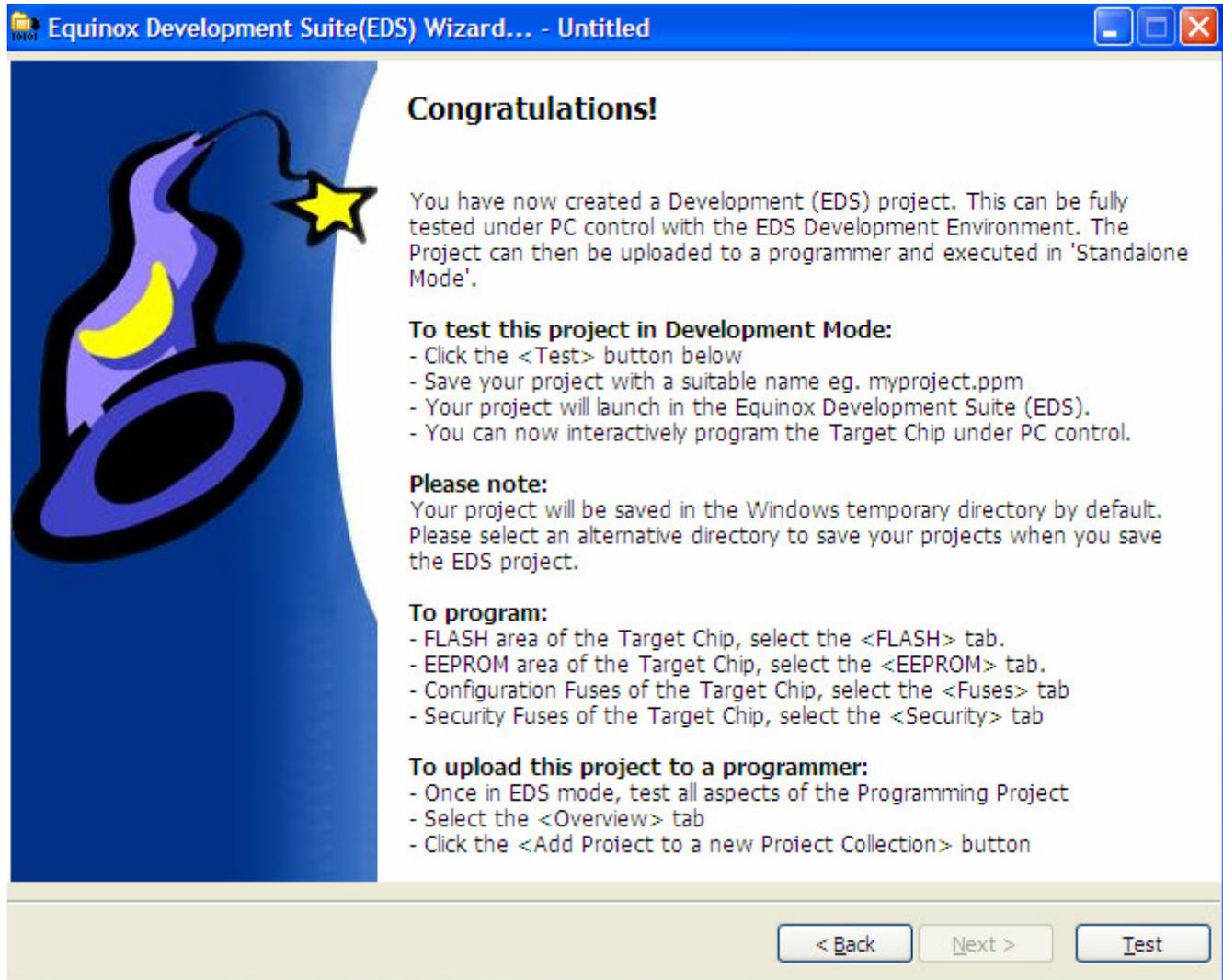
- If the chip has been erased at the start of the programming cycle, then the FLASH should already be blank (i.e. all locations contain the value 0xFF).
- If you want to be absolutely sure the FLASH is blank, you can enable the '**Blank Check Flash**' option. This will perform a full Blank Check of the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

ii. Selecting the FLASH File

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

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).

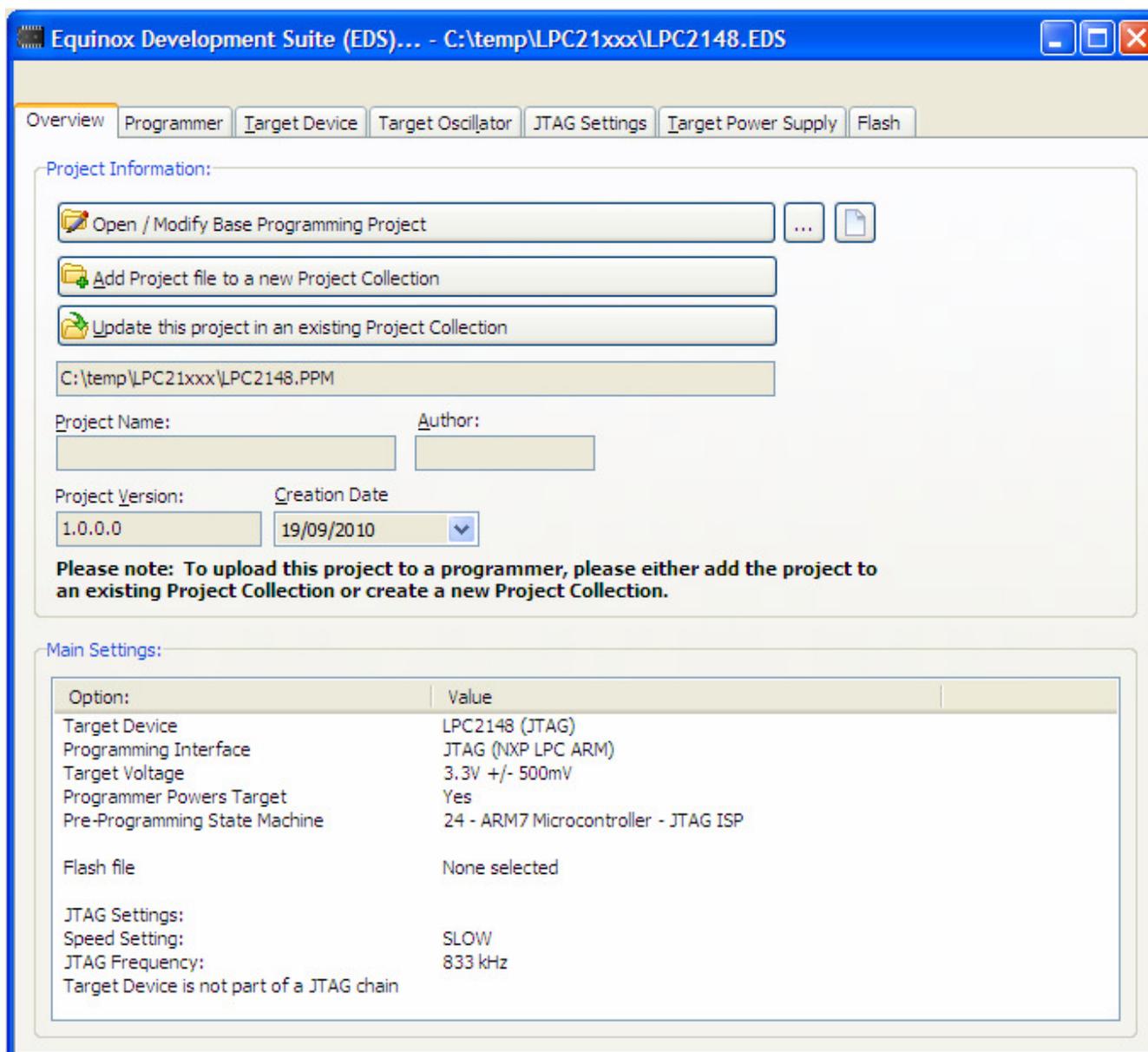


Enter a name for the EDS project e.g. **NXP LPC2xxx** and click the **<Test>** button
→ Your project will now launch in EDS (Development) Mode.

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 JTAG, statemachine etc

The following default settings will be used:

- SLOW JTAG speed at maximum SLOW frequency
- Single JTAG device (no JTAG Chain)
- Target System not powered by programmer (unless enabled during the EDS Wizard)
- The default NXP LPC2xxx - JTAG pre-programming state machine – **Statemachine '24'** will be used.
- The **RESET** pin will **NOT** be driven by the programmer by default. Instead it will be tri-stated at all times.

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

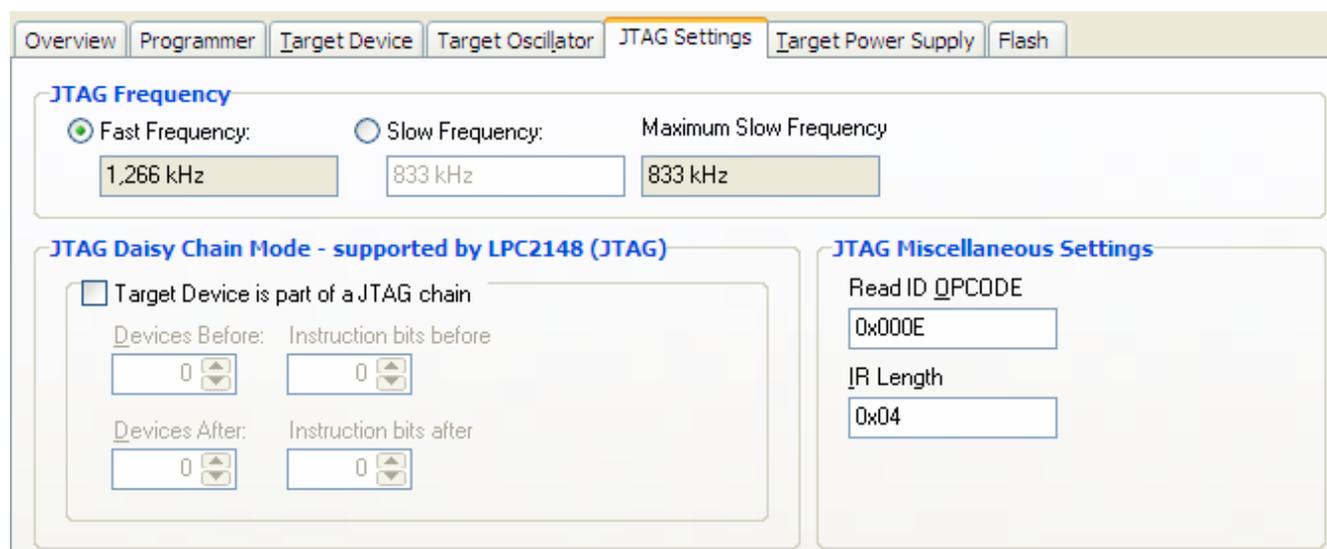
Please follow the instructions in the next sections which explain how to set up the:

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

4.3 JTAG Settings

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

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



The screenshot shows the 'JTAG Settings' tab selected in a software interface. The 'JTAG Frequency' section has 'Fast Frequency' selected with a value of 1,266 kHz, 'Slow Frequency' set to 833 kHz, and 'Maximum Slow Frequency' also set to 833 kHz. The 'JTAG Daisy Chain Mode - supported by LPC2148 (JTAG)' section has a checkbox for 'Target Device is part of a JTAG chain' which is unchecked. Below this are four spinners for 'Devices Before', 'Instruction bits before', 'Devices After', and 'Instruction bits after', all set to 0. The 'JTAG Miscellaneous Settings' section has 'Read ID_OPCODE' set to 0x000E and 'IR Length' set to 0x04.

4.3.1 JTAG frequency

There are two choices of JTAG frequency setting:

i. Slow JTAG (default setting)

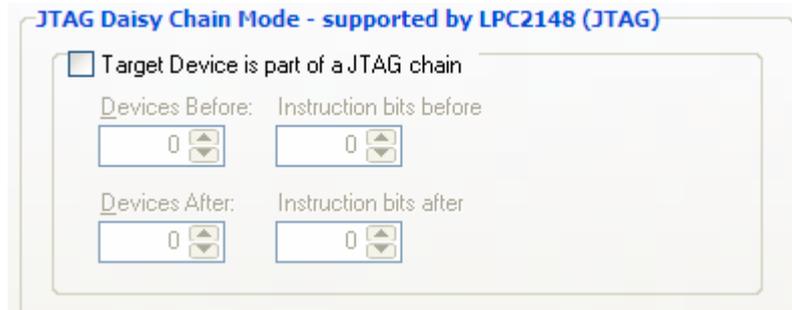
- Selecting the **SLOW JTAG** option allows you to specify a JTAG frequency from 30 kHz up the maximum SLOW JTAG frequency.
- This option should be used if there are reliability problems with JTAG programming using the '**FAST JTAG**' option. If the JTAG frequency is slowed down, the reliability of programming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

ii. FAST JTAG

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

4.3.2 JTAG Chain settings

To set up the position of the Target NXP LPC2xxx Device in a JTAG Chain, select the <JTAG Settings> tab.



JTAG Daisy Chain Mode - supported by LPC2148 (JTAG)

Target Device is part of a JTAG chain

Devices Before: 0 Instruction bits before: 0

Devices After: 0 Instruction bits after: 0

i. Single JTAG Device

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

ii. Device is part of a JTAG Chain

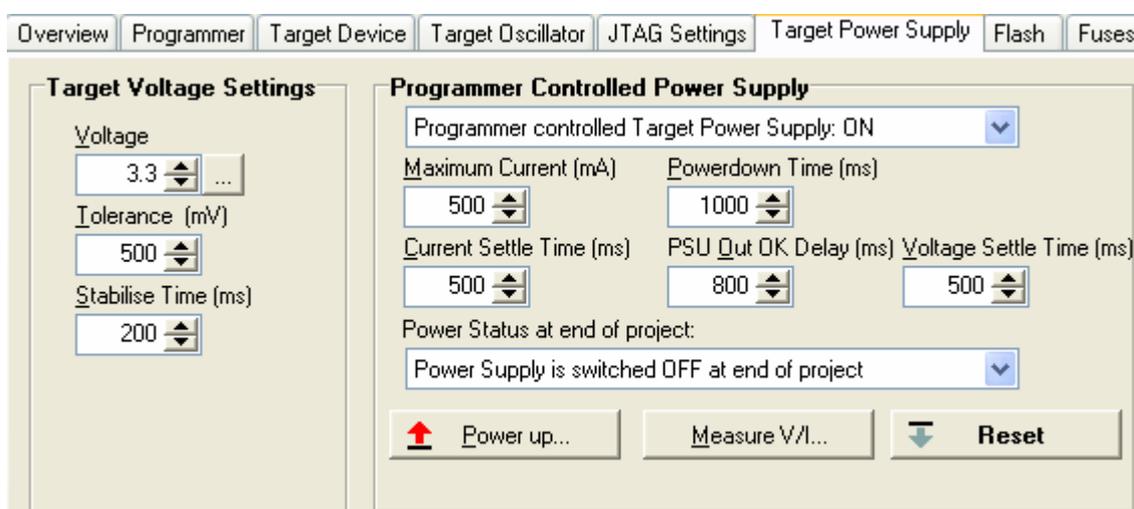
If the Target Device to be programmed is connected so it is part of a JTAG chain, then it is necessary to specify the number of '**JTAG devices**' and '**Instruction Bits**' both BEFORE and AFTER the Target Device in the Chain – see example JTAG Chain below.

4.4 Checking the NXP LPC2xxx – Target Voltage

It is a good idea to check that the target NXP LPC2xxx device is powered at the correct voltage before trying to program it. An NXP LPC2xxx device normally runs at between 3.0 and 3.6V with the core running at 1.8V. The programmer '**Target Vcc**' pin should be connected to the 3.3V rail on the Target System allowing the programmer to measure the Target Voltage (even if the programmer is not powering the Target System).

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

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



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 NXP LPC2xxx device is running at e.g. 3.3V.
- The programmer will then generate JTAG signals which swing between 0V and the '**Target Voltage**'.
- Click the **<Power up>** button to power up the Target System.
- → The programmer will then switch on the programmer controlled power supply and the Target System should power up to the specified voltage.
- The measured '**Target Voltage**'. will be continuously displayed. If it is not, then you can simply click the **<Measure V/I>** button.
- The voltage should be within 3.0 and 3.3V.

If the programmer is **NOT** powering the Target System...

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

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

- The voltage should be within 3.0 and 3.6V.

4.5 Testing JTAG communication with the NXP LPC2xxx device

4.5.1 Overview

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

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

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

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

- The **Device Signature (Part ID)** is displayed e.g. **0x040225**
- The full **JTAG ID** is displayed: e.g. **0x4F1F0F0F**
- The **JTAG ID** is then split into the '**Silicon Revision**', '**Part Number**' and '**Manufacturer**'.
This is not relevant for NXP LPC2xxx devices. The '**Chip ID**' holds this information instead.

This message means that the programmer has established a connection via the JTAG interface to the specified target NXP LPC2xxx device and that the device has the correct '**Part 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.....

Please note:

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

4.5.3 Possible failure messages

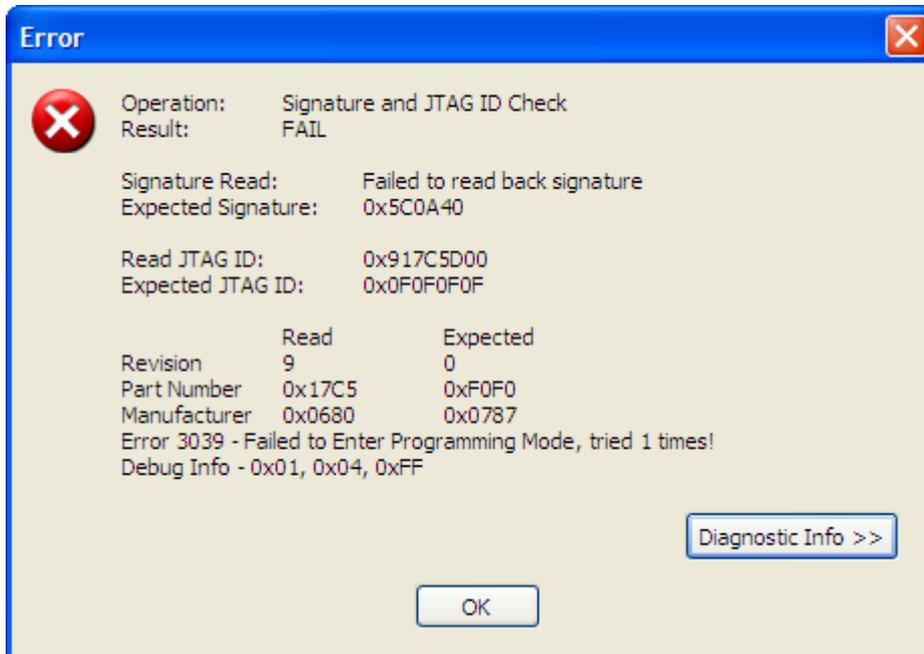
The action of performing a **<Check ID>** can produce any of the following error messages:

- i. Error 3039 / 3044 – Failed to enter programming mode
- ii. Error 44 / 3041 – Signature failure: Read back: **0x??????** Expected: **0x??????**

These errors are discussed in the next two sections.

4.5.4 Error 3044 / 3039 – Failed to enter programming mode

This error message means that the programmer attempted to place the target NXP LPC2xxx device into programming mode, but the operation failed for some reason.



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

If you receive this error, please check the following:

Hardware checks.....

- The JTAG connections between the programmer and the Target System are correct.
- There is definitely power applied to the Target System and to all the JTAG devices if the Target Device is part of a JTAG chain. (Please note – The error message will display the Target Voltage – click the **Diagnostic Info** button.)
- Check that the **RESET** pin of the NXP LPC2XXX device is not being held low by an external reset circuit on the Target Board. This will prevent the device from running the debugger code.
- If there is a **RESET** button or RESET jumper on the Target Board, try resetting the device.
- Try removing power from the Target Board and then switching power back on again.

!!! Important note !!!!

If you are using either of the connector boards / cables detailed below, please check that the **RTCK** pin is disconnected from the programmer as follows:

- **ARM ISP Cable** – remove resistor R6
- ISPnano '**CONMOD**' connector board – remove resistor R6

If the resistor R6 is fitted, it will connect the target LPCxxxx device to the 3.3V supply on the Target Board which will prevent the device from entering debug mode. If in doubt, measure the voltage on the LPCxxxx RTCK pin. It should be 0V in order for programming to work.

EQTools checks.....

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

4.5.5 Error 41 or 3041 – Signature FAIL message

This error message means that the programmer managed to communicate OK with (connect to) the target NXP LPC2xxx device via the JTAG interface, but the **Device Signature (Part ID)** read back does not match the **Part ID** in the device library.



Please note:

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

This message confirms the following:

- The connections between the programmer and NXP LPC2xxx device are OK.
- The programmer is correctly communicating with (connecting to) the NXP LPC2XXX device.

If you receive this error, please check the following:

- The problem is simply that the **'Chip ID' (Signature)** read back from the NXP LPCxxx device does not match the **'Part ID'** in the programming project.
- Check that you have selected the correct NXP LPC2xxx device in your project.
- Check if the **'Chip ID'** read back from the NXP LPC2xxx device matches one of the **'Chip IDs'** from a different silicon revision of the same device. This information can be found in the NXP device datasheet.
- The **'JTAG Chain'** settings are correct for the Target Device being programmed.
- Try slowing down the **'JTAG Frequency'** and then try to check the Device Signature again.
- If the Signature looks like a valid **'Part ID'**, make sure that you have selected the correct JTAG Device in the chain. It is possible that the programmer is actually communicating with a different device by mistake and hence reading the wrong signature.
- For further diagnostics information, click the **[Diagnostic Info>>]** button.

If you still cannot find a device with the correct 'Part ID':

- Try disabling the **'Signature Check'** option on the **Target Device** tab.

4.6 Erasing the FLASH area

4.6.1 Overview

The FLASH area must be erased so that each location contains the value 0xFF before any new data can be programmed into it. It is possible to erase the area of a Target Device by clicking the **<Erase>** button. This sends the '**Chip Erase**' command to the target NXP LPC2xxx device which will then erase the entire FLASH.

4.6.2 Erasing the FLASH area

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

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

4.7 Programming the FLASH Area

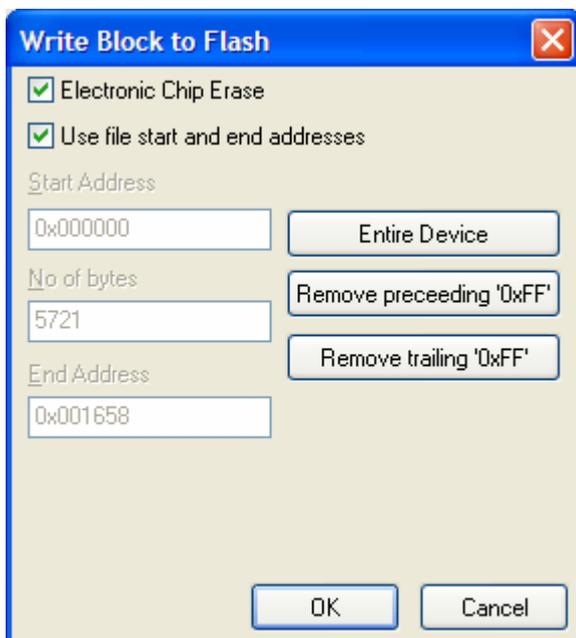
The instructions detailed below describe how to program the contents of a file into the FLASH area of the Target NXP LPC2xxx device:

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

Important note:

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

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



- EDS will automatically perform a **Chip Erase** by default which will erase the entire FLASH before programming any data into it.
- Select the address range you wish to program.
- EDS will automatically use the **'Start'** and **'End'** address of the FLASH input file unless otherwise specified. This reduces the total data actually programmed to the number of bytes in the input file rounded to the end of the nearest FLASH Page.
- If you want to program the entire FLASH range, click the **<Entire Device>** button.
- Click **<OK>** to program the FLASH of the Target Chip.
- The programmer should now start to program the chip.
- The BUSY LED will illuminate on the programmer.
- The programmer will program the contents of the Buffer Window into the FLASH area of the Target Device.
- Each block of data is programmed and then verified so if a failure occurs it will be notified immediately.

- To verify that the data has been programmed correctly, click the <Verify> button.

4.11 Exporting an EDS Project to a Standalone Project

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

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

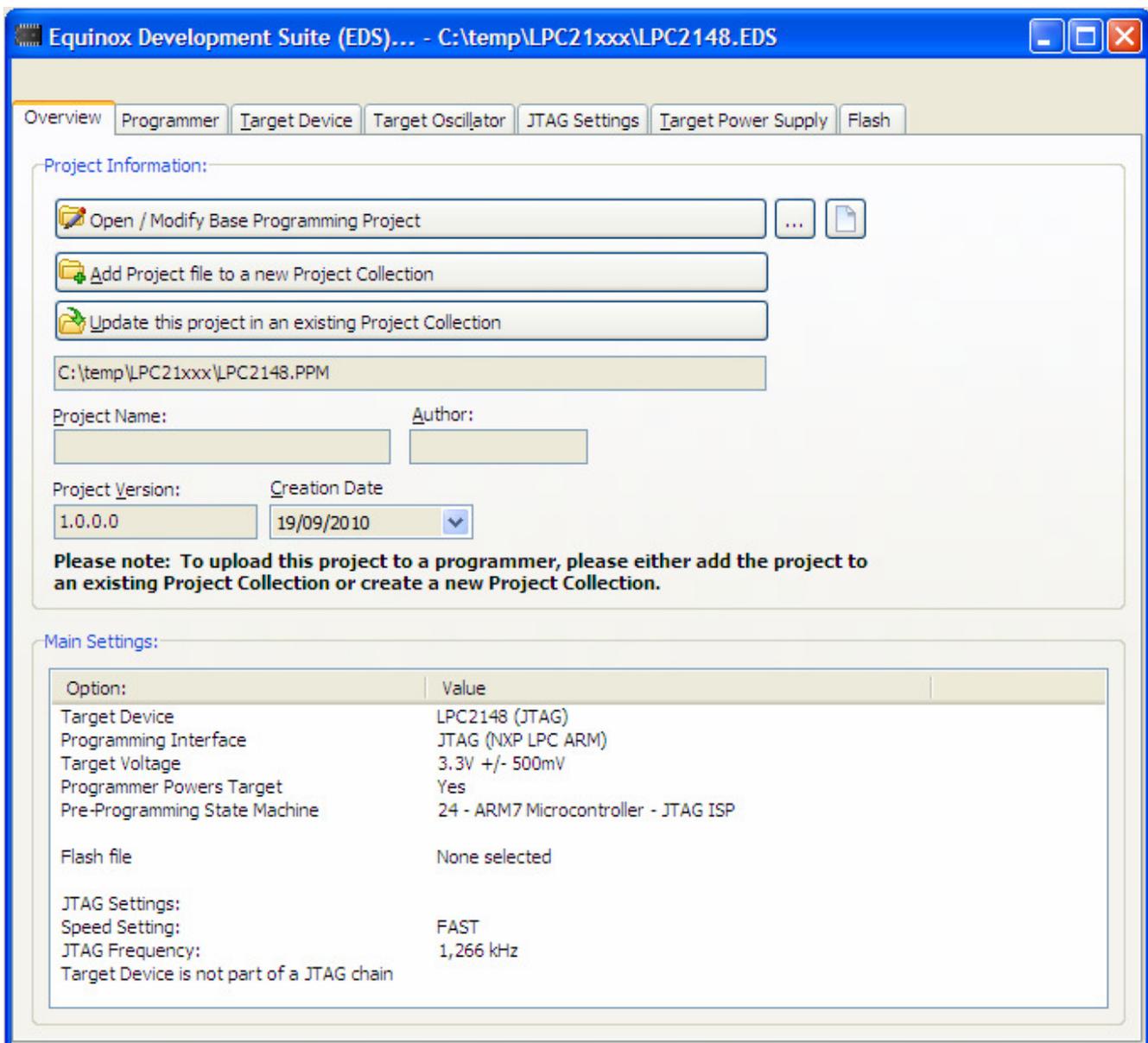
5.0 Creating a Standalone Programming Project

5.1 Overview

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

5.2 Creating a Standalone Project from EDS (Development Mode)

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

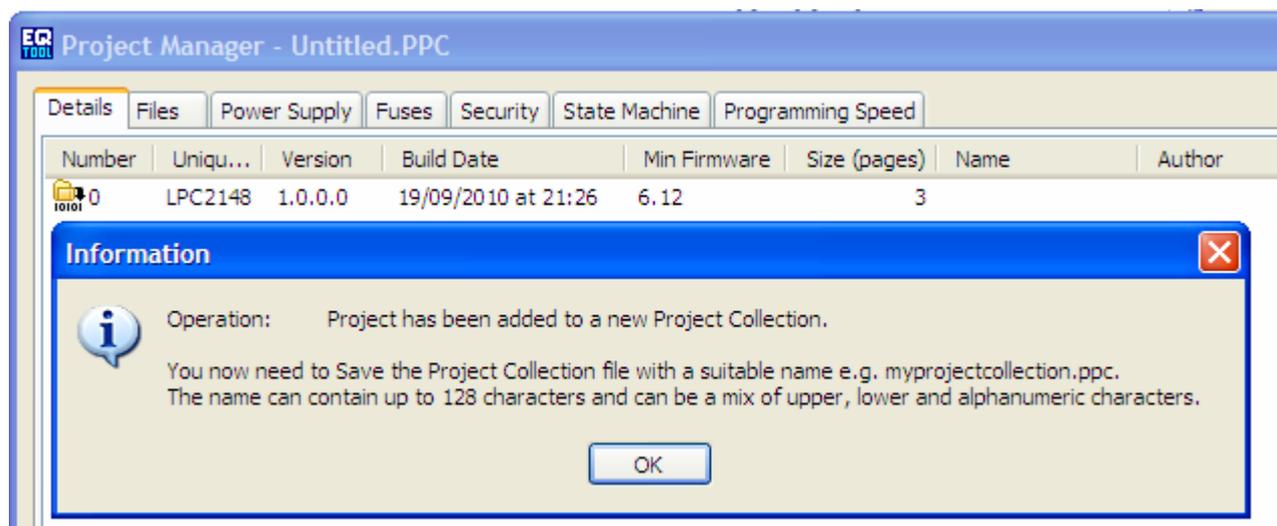


- If this is the first time the EDS Project has been uploaded to a programmer, click the **<Add Project File to a new Project Collection>** button.
- If the EDS Project has already been uploaded to a programmer before, click the **<Update this project in an existing Project Collection>** button.

5.3 Add Project File to a new Project Collection

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

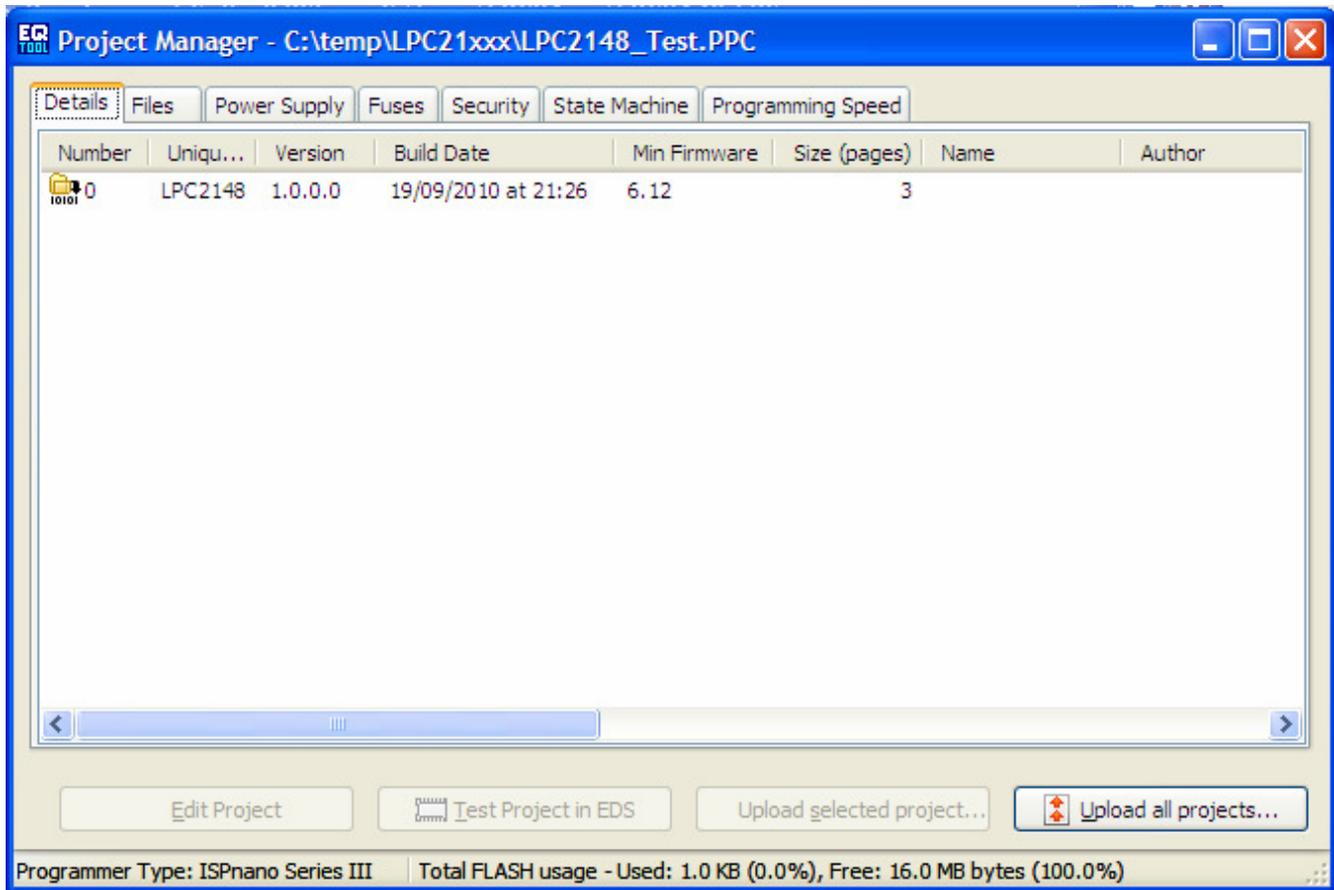
- The EDS Project will appear in a **'Project Manager'** window.



- You will then be prompted to save the **'Project Collection'**.
- Choose a suitable name e.g. **'LPC2148_Test.ppc'** and click the **<Save>** button.
- The Project Manager window is now displayed – see section 5.4.

5.4 Uploading a Project to a programmer

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



In this example we have only one project called '**NXP LPC2148**'.

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

To upload the project to the programmer:

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

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

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

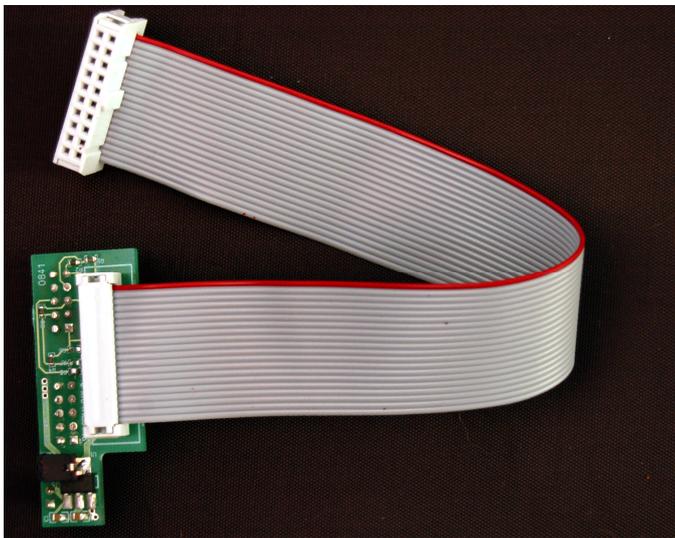
If you want to re-test a Programming Project in EDS (Development Mode), the simplest method to do this is as follows:

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

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

1.0 Overview

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



1.1 Features

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

1.2 Programmer compatibility

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

- Epsilon5 MK2, Epsilon5 MK3 – Portable ISP Programmer
- FS2003 – Portable ISP Programmer
- FS2009 – Portable ISP Programmer

1.3 Power Supply - Selection Jumper

If you have version 'V1.1' of the '**NXP LPC2xxx ISP Cable**' then you will find a '**3-way Jumper Link**' on the circuit board. This jumper is used to configure how the Programmer and Target System are powered. The jumper is labelled '**IN**' and '**OUT**' on the actual PCB. Please refer to the table below to select the correct powering method for your application.

Link pins	Link pos	Powering scenario	Voltage (V)
1-2	IN	Target System powers the Programmer <ul style="list-style-type: none"> The Target System voltage is fed directly to the programmer and will be used to power the programmer. 	3.0 – 5.0V
2-3	OUT	Programmer powers the Target System at 3.3V (default) <ul style="list-style-type: none"> The programmer must be powered from an external power supply via the DC Jack Socket in the range of 3.5 to 5.0V. A 3.3V regulator on the 'NXP LPC2xxx ISP Cable' then feeds 3.3V down to the Target System. 	3.3V

Warning!

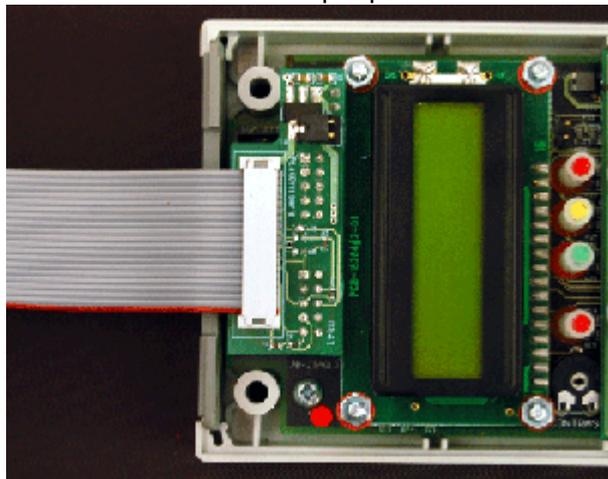
If you select link position '**IN**' and then input +5V into the DC Jack Socket of the programmer, this will feed +5V directly to the Target System. This +5V could damage circuitry on the Target System if it is only designed to run at 3.3V.

1.4 Cable Installation Instructions

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

Instructions:

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



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

!!! Important note !!!

If you are using the Equinox '**ARM ISP Cable**' to program an NXP LPC2xxx microcontroller, please check that the resistor R6 is removed from the circuit board. This resistor is connected to a 47k ohm pull-up to Vcc and will pull the **RTCK** pin HIGH preventing the NXP LPC2xxx device from entering '**Debug Mode**' using the '**JTAG Interface**'.

1.5 Getting Started

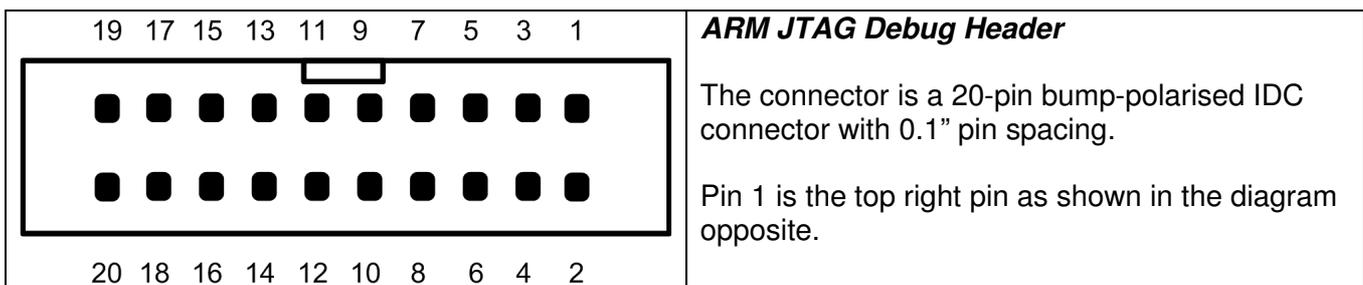
Once you have the installed into the programmer and connected to your Target System, then you are ready to program an ARM device.

Instructions:

- Check that you have selected the correct position for the '**Power Supply – Selection Jumper**' – see section 1.3
- To be on the safe side, measure the voltage on pin 1 of the 20-way IDC connector. If the programmer is powering the Target System, then this voltage should 3.3V.
- If you are using the Equinox '**ARM ISP Cable**' to program an NXP LPC2xxx microcontroller, please check that the resistor R6 is removed from the circuit board.
- Connect the '**ARM Programming Cable**' to the 20-way IDC connector on your LPC2xxxTarget System
- Power up the programmer first → this makes sure that the programmer Line Drivers are powered BEFORE you power up the Target Board.
- Power up the Target Board
- Check that the programmer and Target Board power up OK.

1.6 ARM JTAG Debug Header

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

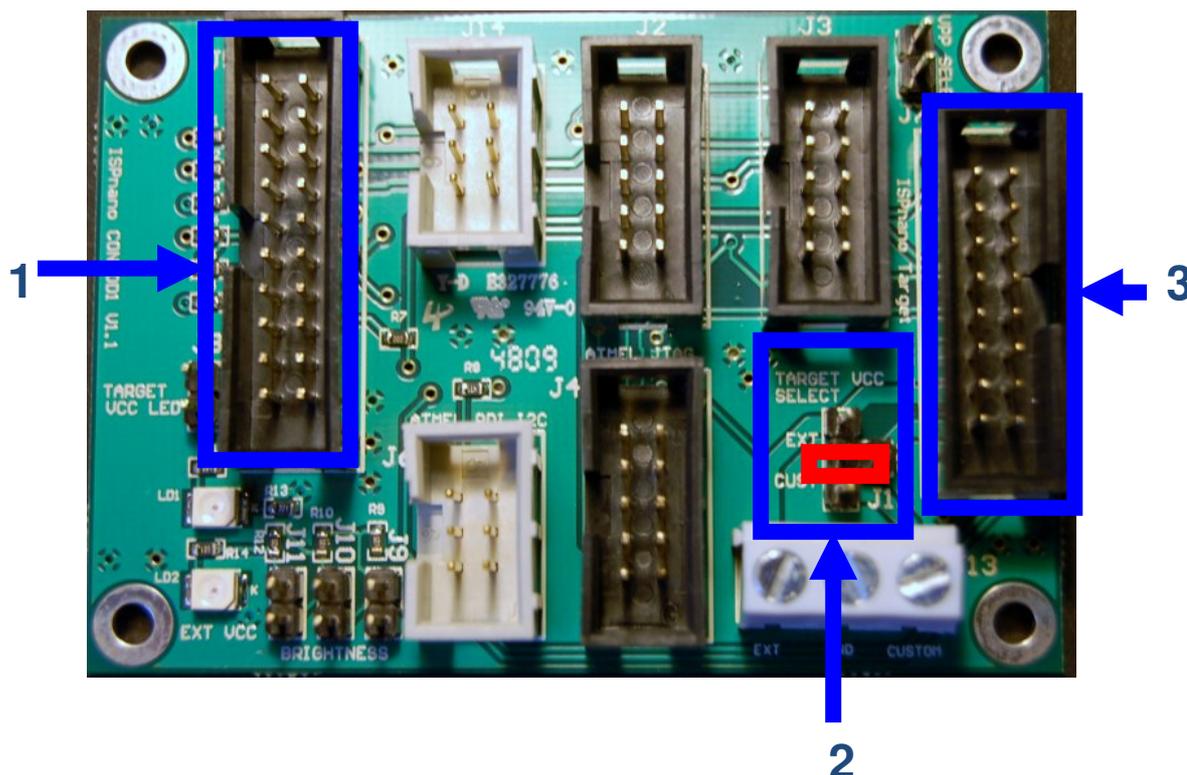


Appendix 2 – CONMOD Module + NXP LPCxxx

1.0 Overview

This appendix describes how to use the '*ISPnano CONMOD Module*' to connect an '*ISPnano Series III*' programmer to an '*NXP LPCxxxx*' device using the JTAG interface.

The '*ISPnano CONMOD Module*' is shown below.....



#	Description	Further information
1	20-way ARM Connector	20-way 0.1" pitch IDC Box Header
2	Target Vcc Selection jumper	3-way Jumper Link
3	ISPnano programmer connector	16-way 0.1" pitch IDC Box Header

Connecting the programmer to the LPC Target Board....

- Referring to the annotated picture above
- Plug the 16-way IDC cable supplied with the programmer between the programmer '**Target ISP Port**' (16-way IDC connector) and the CONMOD Module 16-way header (J7) – see arrow (3).
- Plug one end of the 20-way IDC cable supplied with the programmer into **the '20-way ARM Connector'** (1) on the CONMOD board
- Plug the other end of the 20-way IDC cable into the mating 20-way **IDC 'ARM Connector'** on your LPC21xx Target Board.

- Set up the '**Target Vcc Select**' jumper so that the programmer powers the Target Board – see red box marked (2) in the picture.