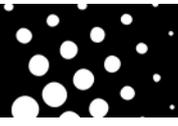
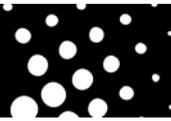




# ISP nano

## Series I / II / III / III ATE & MUX 2 / 4 / 8 Programmer User Manual





# Contents

Copyright Information .....	vii
Equinox Warranty Information .....	viii
Electromagnetic Compatibility (EMC) Compliance.....	x
Technical Support.....	xi
<b>Product Documentation and Software.....</b>	<b>xii</b>
Overview.....	xii
Documentation and software for the ISPnano programmer.....	xiv
Device algorithm - Application notes.....	xiv
Programmer related - Application notes .....	xv
<b>1.0 Introduction .....</b>	<b>1</b>
1.1 ISPnano Series III Programmer .....	1
1.2 ISPnano Series III ATE Programmer .....	2
1.3 ISPnano MUX2 / MUX4 / MUX8 - Multiplexed Programmers .....	3
1.4 ISPnano Programmer Module - Main Features .....	5
1.5 Programmer Selection Guide.....	7
1.5.1 Single channel ISPnano programmers.....	7
1.5.2 Multi-channel multiplexed ISPnano programmers.....	8
1.5.3 Programmer on-board Standalone Project Storage .....	8
1.5.4 Programmer Target I/O Capability.....	9
1.5.5 Supported programming interfaces .....	9
1.5.6 Programmer – Controlled Power Supplies .....	10
1.5.7 Communication / Control Ports.....	10
1.5.8 Programmer Control methods in Standalone Mode .....	11
1.6 Programmer – Target I/O Signals .....	12
1.7 System Contents .....	13
1.7.1 ISPnano Series 3 Kit .....	13
1.7.2 ISPnano-MUX Programming Sytem Kit.....	14
1.8 ISPnano Series I - Rear panel connections .....	15
1.9 ISPnano Series II / III – Rear panel connections .....	16
1.10 ISPnano - Front panel connections.....	17
1.10.1 Overview .....	17
1.10.2 Programmer Remote Control Port.....	17
1.10.3 Programmer ‘Target ISP Connector Port’.....	18
1.11 Programmer Specifications.....	19
1.12 Programmer on-board FLASH Memory Store.....	22
<b>2.0 Getting Started Guide.....</b>	<b>23</b>
2.1 Overview.....	23
2.2 Hardware Installation Procedure.....	24
2.2.1 ESD Precautions .....	24
2.2.2 Unpacking the programmer and accessories .....	24
2.2.3 Selecting the Communications Mode .....	24
2.2.4 Powering the ISPnano programmer .....	24
2.3 Software Overview and Installation.....	26
2.3.1 Software Overview .....	26
2.3.2 Programmer Control Mechanisms.....	27
2.3.3 Software Installation .....	28

2.4 Programmer Operating Modes.....	29
2.5 Programmer Status LEDs .....	31
2.6 ASCII Text Communications Mode .....	32
2.6.1 Overview .....	32
2.6.2 Limitations of this protocol .....	32
2.6.3 Further Information .....	32
2.7 Standalone Remote I/O Programmer Control .....	33
2.7.1 Overview .....	33
2.7.2 TTL 4-Wire Remote Control – Signal Connections.....	34
2.7.3 Control diagram for Remote I/O 4-wire Control .....	35
2.8 Target System Connect / Disconnect detection .....	36
2.9 Script Mode (ISP-PRO).....	36
2.10 ConsoleEDS – Command Line control .....	37
<b>3.0 Serial Communications Ports .....</b>	<b>39</b>
3.1 Overview of Serial Ports.....	39
3.2 ISPnano Series I – Serial Port connections .....	40
3.2.1 Overview .....	40
3.2.2 ISPnano Series I - Connecting to the RS232-1 Serial Port.....	41
3.2.3 ISPnano Series I - connecting to the RS232 Serial Port (RJ11).....	42
3.3 ISPnano Series II + III (ATE) + MUX – Serial Port connections .....	43
3.3.1 Overview .....	43
3.3.2 ISPnano Series 3 - Comparison of communications modes.....	44
3.3.3 ISPnano Series I / II / III - connecting to the RS232 Serial Port (RJ11).....	44
3.4 USB Port.....	46
3.4.1 Overview of USB connection.....	46
3.4.2 USB Driver - Installation Instructions .....	46
3.4.3 Selecting the USB – Virtual COM Port (VCP) in EQTools .....	47
3.5 RS485 Port .....	48
3.5.1 Overview .....	48
3.5.2 Equipment required .....	48
3.5.3 Instructions .....	49
3.5.4 RS485 FCC68 to 5-pin Molex Converter Module .....	49
3.6 Detecting attached programmers .....	50
3.6.1 Overview .....	50
3.6.2 Using the ‘Detect Programmer(s)’ utility .....	50
3.6.3 Resolving Programmer Detection issues.....	51
3.7 Setting the programmer ‘Communications Node Address’ .....	52
3.8 Connecting the Remote Display / Keypad Module.....	55
<b>4.0 Programmer / Target System Power Supply Scenarios .....</b>	<b>57</b>
4.1 Overview of programmer Power Supplies.....	57
4.2 Typical applications of each power supply .....	58
4.3 ISPnano - Power Supply schematic diagram.....	59
4.4 Programmer Controlled ‘Target Vcc (TVCC) Power Supply’ .....	60
4.5 Programmer controlled ‘Target Vpp (TVPP) Power Supply’ .....	61
4.6 External VCC – programmer controlled switch .....	62
4.7 Programmer controlled Target Discharge Circuit.....	64
4.2 DC Power Connector overview .....	67
4.4 Powering the programmer.....	68



4.4.1 Overview .....	68
4.4.2 Powering the programmer using the AC Wall Adaptor .....	68
4.4.3 Powering the programmer from an external power supply .....	69
4.5 Switching EXTERNAL VCC to the Target System .....	70
4.5.1 Overview .....	70
4.5.2 How the programmer measures the Target Vcc.....	71
4.5.3 Connection instructions .....	73
4.5.4 Setting up a project in EDS to externally power the Target System .....	74
4.5.5 Measuring the External Target Vcc Voltage .....	75
<b>Appendix 1 – ISPnano Series I / II - Target ISP Connector Port.....</b>	<b>77</b>
1.1 Programmer I/O Signals .....	77
1.2 Target ISP Port – connector pin-out .....	79
<b>Appendix 2 – ISPnano Series III (ATE) - Target ISP Connector Port .....</b>	<b>83</b>
1.1 Programmer I/O Signals .....	83
1.2 Target ISP Port – connector pin-out .....	84
1.3 Atmel AVR microcontroller - SPI connections.....	87
1.4 Atmel AVR microcontroller - JTAG connections .....	88
1.5 Atmel AT91SAM7 microcontroller - JTAG connections .....	89
1.6 Serial EEPROM – I2C connections .....	90
1.7 Zensys Z-WAVE devices - SPI connections .....	91
1.7.1 Overview .....	91
1.7.2 Using the CONMOD module .....	92
<b>Appendix 3 - Remote Control Port .....</b>	<b>95</b>
1.1 Overview.....	95
1.2 Remote Control Port – connector pin-out.....	95
1.3 Connecting Remote Status LEDs .....	96
<b>Appendix 4 – ISPnano Series 3 ATE .....</b>	<b>97</b>
<b>1.0 Overview.....</b>	<b>97</b>
1.1 Front panel layout.....	98
1.2 Rear Panel IDC Connector Ports and signal routing.....	100
1.3 Target System routed to External ATE System (default) .....	101
1.4 Target System routed to internal programmer (RELAYS ON).....	102
<b>2.0 Power Supply requirements .....</b>	<b>105</b>
2.1 Overview.....	105
2.2 Power Supply connections.....	106
2.3 Programmer Power Supply recommendations .....	106
<b>3.0 Programmer / ATE Signal routing .....</b>	<b>107</b>
3.1 Overview of ports.....	107
3.2 'TARGET ISP' Port .....	108
3.3 'Remote Control' Port .....	108
3.4 ATE Port .....	108
3.5 ATE Port → Target ISP Port – Pin routing (RELAYS OFF).....	110
<b>4.0 Controlling the RELAY switching.....</b>	<b>111</b>
4.1 Overview.....	111
4.2 Controlling the RELAYS from EDS (Development Mode) .....	111
4.3 Setting the Programmer Signal Line Driver voltage .....	112
4.4 'Programmer controlled Target Power supply' setup .....	112
4.5 Target System is externally powered.....	114

4.6 Switching the RELAYs ON.....	115
4.7 Switching the RELAYs OFF .....	116
4.8 Leaving the RELAYs ON at the end of a project.....	116
4.9 Measuring the Target Vcc voltage .....	117
4.10 Switching power via the relays.....	118

## Copyright Information

Information in this document is subject to change without notice and does not represent a commitment on the part of the manufacturer. The software described in this document is furnished under license agreement or nondisclosure agreement and may be used or copied only in accordance with the terms of the agreement.

It is against the law to copy the software on any medium except as specifically allowed in the license or nondisclosure agreement.

The purchaser may make one copy of the software for backup purposes. No part of this manual may be reproduced or transmitted in any form or by any means, electronic, mechanical, including photocopying, recording, or information retrieval systems, for any purpose other than for the purchaser's personal use, without written permission.

**© 2000 - 2011 Copyright Equinox Technologies UK Limited. All rights reserved.**

Atmel™ and AVR™ are trademarks of the Atmel Corporation

Microsoft, MS-DOS, Windows™, Windows 95™, Windows 98™, Windows XP™ and Windows NT4™ are registered trademarks of the Microsoft Corporation

IBM, PC and PS/2 are registered trademarks of International Business Machines Corporation

Intel, MCS 51, ASM-51 and PL/M-51 are registered trademarks of the Intel Corporation

Every effort was made to ensure accuracy in this manual and to give appropriate credit to persons, companies and trademarks referenced herein.

Equinox guarantees that its products will be free from defects of material and workmanship under normal use and service, and these products will perform to current specifications in accordance with, and subject to, the Company's standard warranty which is detailed in Equinox's Purchase Order Acknowledgment.

## Equinox Warranty Information

This product is guaranteed by Equinox Technologies UK Limited for a period of 12 months (1 year) after the date of purchase against defects due to faulty workmanship or materials. One guarantee covers both parts and labour. Service under the guarantee is only provided upon presentation of reasonable evidence that the date of the claim is within the guarantee period (e.g. completed registration/guarantee card or a purchase receipt).

The guarantee is not valid if the defect is due to accidental damage, misuse or neglect and in the case of alterations or repair carried out by unauthorised persons. A number of exceptions to the warranty are listed in the 'Exceptions to warranty' section below. Service (during and after guarantee period) is available in all countries where the product is distributed by Equinox Technologies UK Limited.

### Exceptions to warranty

#### Over-voltage damage

This warranty does not cover damage to the programmer due to voltages beyond the specified voltage limits being applied to the 'DC Power Input' (CON1) or any of the ISP Headers. The user must ensure that sufficient care is taken to avoid over-voltage and static conditions on any of the 'ISP Header' I/O pins.

#### Over-current damage

This warranty does not cover damage to the programmer due to excessive current being drawn from the programmer power supply. The user must ensure that there is sufficient over-current protection within the test fixture to protect against short circuit loads.

#### Short-circuit damage

This warranty does not cover damage to the programmer due to short-circuit loads being placed across programmer I/O lines.

#### Warning!

Any damage caused to the programmer by Electrostatic Discharge (ESD) through inadequate earthing is not covered under the warranty of the product.

## Disclaimer

Whilst every effort has been made to ensure that programming algorithms are correct at the time of their release, it is always possible that programming problems may be encountered, especially when new devices and their associated algorithms are initially released. It is Equinox's Company Policy to endeavour to rectify any programming issues as quickly as possible after a validated fault report is received.

It is recommended that high-volume users always validate that a sample of a devices has been programmed correctly, before programming a large batch. Equinox Technologies UK Ltd. can not be held responsible for any third party claims which arise out of the use of this programmer including 'consequential loss' and 'loss of profit'.

Equinox Technologies UK Ltd. cannot be held responsible for any programming problems which are 'out of our control'. This type of problem is usually listed in the 'Errata Sheet' for the particular device being programmed and is available from the silicon vendor.

Information contained in this manual is for guidance purposes only and is subject to change. E&OE.

# Electromagnetic Compatibility (EMC) Compliance

The 'ISPnano Programming Module' is a CE Approved Product. It is designed for use in an ESD controlled environment i.e. development or production. This means, therefore, that the user must ensure that there is no possibility of damage from electrostatic discharge (ESD). Since the devices and equipment to which this product is likely to be connected may well themselves be susceptible to ESD, this should not pose any difficulty.

For example, if you are handling microcontrollers and EEPROMS etc. then you will already be used to appropriate precautions, such as the use of anti-static mats, wrist straps and so on. You should treat your 'ISPnano Programming Module' with the same care as you would these types of devices. Always ensure that you are not carrying a static charge yourself before handling the product. Wearing an earthed anti-static wrist strap is recommended.

Equinox has taken great care in designing this product to be compliant with the European EMC directive. When using the equipment be sure to follow the instructions provided. Although RF emissions are within prescribed limits, care should be taken if you are using the product near to sensitive apparatus. If you experience any difficulty please refer to Equinox Technical Support.



## ESD Points to remember

- Work in a static-free environment.
- Wear an earthed wrist strap when handling either the programmer and/or any programmable device.
- Ensure that the PC, programmer and Target system are connected to the same EARTH (0V) potential.
- Do NOT plug the ISP cable of the programmer into a Target System when the Target power is ON.

### Warning!

Any damage caused to the programmer by Electrostatic Discharge (ESD) through inadequate earthing is not covered under the warranty of the product.

## Technical Support

It is often the case that users experience problems when installing or using a product for the first time.

If you have a technical support problem, please consult the following list for help:

▶ **User Manual**

▶ **On-line help**

Press <F1> for help at any time when running EQTools or ISP-PRO.

The help system is context-sensitive. Simply press <F1> on any error message and the possible causes of the error should be listed. This help system is updated on a regular basis. Please see software update details for information on keeping up-to-date with software revisions.

▶ **Internet Web Site**

The support / download page for the '*ISPnano programmer range*' can be found at:

<http://www.equinox-tech.com/products/details.asp?ID=1440&displ=t>

▶ **E-mail**

Please e-mail any technical support questions about this product to:

[support@equinox-tech.com](mailto:support@equinox-tech.com)

▶ **Fax**

Please fax any technical support questions about this product to: +44 (0) 1942 844181

Equinox will try our best to answer your questions about this product as quickly as possible. However, we cannot promise an immediate reply. Please consult our web site for new software updates as the problem that you are enquiring about may have already been fixed in a new version.

# Product Documentation and Software

## Overview

This manual provides an overview of the contents of the *'ISPnano Programming Range'* plus associated hardware and software. References may be made to other hardware and software products which are not covered in detail in this manual.

Please refer to the table below for a list of sources of documentation and/or browse to

<http://www.equinox-tech.com/products/details.asp?ID=1440&displ=tl>

Software:	
	<p><b>EQTools Script Builder – Manual</b></p> <p>This software is used to create and upload 'Programming Projects' to the programmer.</p> <p>The following sources of documentation are available for this software:</p> <p>Installation and Getting Started Guide</p> <p>Help file</p>
	<p><b>ASCII Text Communications Protocol – Application Note</b></p> <p>This protocol can be used to control the programmer from an external controller via RS-232.</p> <p>The following sources of documentation are available for this protocol:</p> <ul style="list-style-type: none"> <li>• Application Note – AN110</li> </ul>
	<p><b>ISP Pro – Manual</b></p> <p>This software is used to control the programmer in a production environment. It is not supplied as standard with this programmer.</p> <p>The following sources of documentation are available for this software:</p> <p>Installation and User Manual</p> <p>Help File</p>

 <p>Upload Wizard</p>	<p><b>Upload Wizard - Standalone Project Upload Utility</b></p> <p>This software utility is used to upload Programming Projects to any Equinox programmer. These projects can then be used in Standalone Mode, i.e. without a PC.</p> <ul style="list-style-type: none"> <li>• Please follow the on-screen instructions within the Upload Wizard utility itself.</li> </ul> <p>Application Note – AN117</p>
	<p><b>Labview – Remote Application Control – Application Note</b></p> <p>This upgrade allows a production facility to control a single programmer from a ‘Labview for Windows’ application. The Application note describes how to control the programmer using a custom Labview (from National Instruments) application.</p> <p>The following sources of documentation are available for this software:</p> <ul style="list-style-type: none"> <li>• Application Note – AN109</li> </ul>
	<p><b>Remote Application Control – Application Note</b></p> <p>Describes how to control the programmer using a custom Remote Application written in e.g. Visual Basic, C++, C Builder, Delphi etc.</p> <p>The following sources of documentation are available for this software:</p> <ul style="list-style-type: none"> <li>• Application Note – AN109</li> </ul>
<p>ConsoleEDS</p>	<p><b>ConsoleEDS Pro – Application note</b></p> <p>This software utility allows any Equinox programmer to be controlled via simple Command Line instructions from a Command Window within Windows.</p> <p>The following sources of documentation are available for this software:</p> <ul style="list-style-type: none"> <li>• Application Note - AN111</li> </ul>
	<p><b>JTAG In-System (ISP) Upgrade – Application Note</b></p> <p>This license upgrade enables the programmer to support high-speed In-System Programming (ISP) of the Atmel ATmega microcontroller family using the JTAG algorithm. Support is offered for both single and multiple JTAG devices in a JTAG Chain.</p> <p>The following sources of documentation are available for this software:</p> <ul style="list-style-type: none"> <li>• Application Note – AN105</li> </ul>

## Documentation and software for the ISPnano programmer

In line with our policy of continuous improvement, the software and associated documentation for this product are updated on a regular basis. You can download the latest software, firmware, User Manuals and application notes for the ISPnano programmer from the following page on the Equinox website:

<http://www.equinox-tech.com/products/details.asp?ID=1440&displ=tl>

You may be asked to register / log in to download some of these files.

## Device algorithm - Application notes

The table below lists the Application Notes available for helping to create '*Programming Projects*' for different device families.

Application Note	Device Family	Programming Interface
AN100	Atmel - AT89Sxxxx FLASH microcontrollers	SPI
AN101	Atmel - AVR FLASH microcontrollers via the SPI Interface	SPI
AN105	Atmel - AVR FLASH microcontrollers via the JTAG Interface	JTAG
AN118	Generic I2C 24xxx Serial EEPROM memories	I2C
AN122	Atmel - AT91SAM7 ARM7 FLASH microcontrollers	JTAG
AN127	Atmel – XMEGA AVR FLASH microcontrollers via the 2-wire PDI interface	PDI
AN128	NXP – LPCxxx ARM7 FLASH microcontrollers	JTAG
AN130	Zensys – ZWxxx – Z-WAVE Series	SPI
AN132	Atmel ATtiny AVR microcontrollers via the TPI interface	TPI
AN133	Atmel AT45D Serial DataFlash programming	SPI

These application notes can be found in PDF format on the CD-ROM which was supplied with the programmer. You can also find the very latest versions on the "*ISPnano Download Page*" on the Equinox website.

## Programmer related - Application notes

The table below lists the Application Notes available for the ISPnano programmer range which describe the USB driver installation, the different control methods available, firmware update procedure and '**AVR Oscillator Calibration**' procedure.

Application Note	Description
AN109	Remote Application Control of Equinox ISP Programmers using ISP-PRO Utility
AN110	ASCII Text Control (ATC) Protocol for Remote Control of Equinox Programmers
AN111	ConsoleEDS Protocol for Remote Control of Equinox Programmers
AN112	Firmware Update instructions for Equinox ISP Programmers
AN114	Accurate on-chip Oscillator Calibration for Atmel AVR microcontrollers
AN121	Equinox EQTools – Release Notes
AN122	In-System Programming (ISP) of the Atmel AT91SAM7 FLASH Microcontroller Families using the JTAG Programming Interface
AN123	Controlling an Equinox ISP Programmer from a Remote System via the Remote 4-wire TTL Port
AN126	USB Driver Installation instructions for PPM4-MK1 and ISPnano programmers

These application notes can be found in PDF format on the CD-ROM which was supplied with the programmer. You can also find the very latest versions on the "**ISPnano Download Page**" on the Equinox website.



## 1.0 Introduction

The ISPnano series of ISP programmers are designed for high-speed In-System Programming (ISP) of FLASH Microcontrollers and Serial Memory devices. Their compact size and abundance of programming ports makes them ideal for integration into any Test Fixture or ATE System. They supports programming of devices by most interfaces including SPI, JTAG, JTAG chain, SCI, I2C (2-wire), XMEGA PDI, ATtiny TPI and UART Boot Loader. The fastest possible programming times are guaranteed due to a combination of highly optimised algorithms, local storage of Project Data and high slew rate Line Driver Circuitry.

The ISPnano series of programmers represents a scaleable ISP programming solution capable of programming multiple PCBs on a 'PCB Panel'. This is made possible by networking up to 32 x ISPnano programmers to a single PC. All programmers can then be configured to start simultaneously under PC control.

### 1.1 ISPnano Series III Programmer

The '*ISPnano Series III*' is a state-of-the-art Production ISP Programmer designed for high-speed In-System Programming (ISP) of FLASH Microcontrollers and Serial Memory devices. Its compact size and abundance of programming ports makes it ideal for integration into any Test Fixture or ATE System. It supports programming of devices by most interfaces including SPI, JTAG, JTAG chain, SCI, I2C (2-wire) and UART Boot Loader. The fastest possible programming times are guaranteed due to a combination of highly optimised algorithms, local storage of Project Data and high slew rate Line Driver Circuitry.

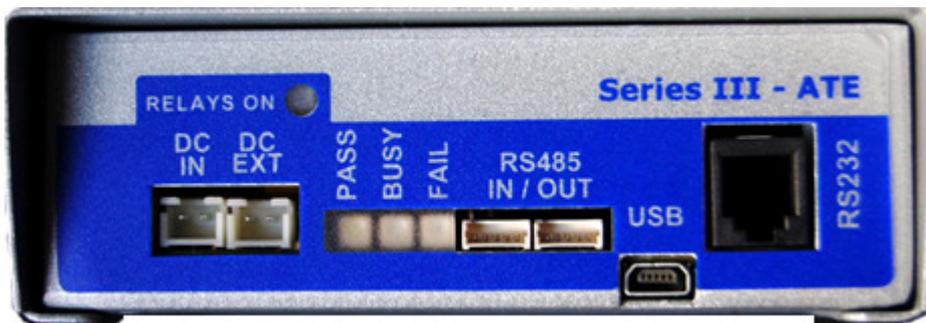


The '*ISPnano Series III*' is capable of operating in '**Standalone Mode**' where any one of 64 independent '**Programming Projects**' can be executed via a remote command. The programmer also features a '**4-wire TTL Remote Control interface**' making it ideal for interfacing to any ATE or In-Circuit Tester (ICT). A special '**Start Signal**' can be used to detect when a Test Fixture lid has been closed and thereby automatically start a programming operation.

The '*ISPnano Series III*' is a scaleable ISP programming solution capable of programming multiple PCBs on a 'PCB Panel'. This is made possible by networking up to 32 x ISPnano programmers to a single PC. All programmers can then be configured to start simultaneously under PC control.

## 1.2 ISPnano Series III ATE Programmer

The '*ISPnano Series III ATE*' is a state-of-the-art Production ISP Programmer designed for high-speed In-System Programming (ISP) of FLASH Microcontrollers and Serial Memory devices. It has been specially developed to allow simple integration with ATE Systems and In-Circuit Testers (ICTs) used in high-volume production environments. The programmer supports automated relay switching of all Target System signal and power signal lines between an external ATE / ICT system and the internal programmer electronics. This allows the programmer to be completely out-of-circuit during normal ATE / ICT testing operation. A dedicated '*ATE Interface*' allows the ATE / ICT to control the operation of the programmer in '*Standalone Mode*'.



The compact size and abundance of programming ports makes the '*ISPnano Series III ATE*' ideal for integration into any Test Fixture or ATE System. It supports programming of devices by most interfaces including SPI, JTAG, JTAG chain, SCI, I2C (2-wire), XMEGA 2-wire PDI, ATtiny HV (+12V Vpp), ATtiny TPI and UART Boot Loader. The fastest possible programming times are guaranteed due to a combination of highly optimised algorithms, local storage of Project Data and high slew rate Line Driver Circuitry.

The '*ISPnano Series III ATE*' is capable of operating in '*Standalone Mode*' where any one of 64 independent '*Programming Projects*' can be executed via a remote command.

### Please note:

The '*ISPnano Series III ATE*' programmer features a standard '*ISPnano Series III*' programming module and a '*relay switching module*' integrated into a single enclosure. The '*ATE*' version will run exactly the same projects as the standard version with the exception that the '*EXT-VCC*' supply is used to power / control the relay coils.

### 1.3 ISPnano MUX2 / MUX4 / MUX8 - Multiplexed Programmers

The '*ISPnano-MUX*' family of ISP programming systems have been specially developed for medium to high-volume production programming applications where there is a requirement to program multiple devices mounted on a '*PCB Panel*'. The '*ISPnano-MUX*' programmers use a single integrated '*ISPnano Series 3*' programmer which is then multiplexed sequentially to each UUT (Unit Under Test) in turn on the PCB panel. This allows the system to sequentially program up to 8 individual UUTs, one after the other.

#### Main Features...

- Available as 2 / 4 or 8 channel multiplexed programming system
- Supports sequential programming of 2 / 4 or 8 UUTs (depending on programmer model)
- A single '*ISPnano Series 3*' programmer is multiplexed sequentially to 2 / 4 / 8 individual Target Boards (UUTs)
- All programming signals are multiplexed to each UUT using high-quality analogue switches
- All programming signals are fully ESD and over-voltage protected
- All passive / power signals (Target VCC, Target Vpp and EXT-VCC power signals) are switched via individual high-quality relays to each UUT
- Power is always switched off to the target UUT before switching to the next UUT thus providing very long relay contact life
- Each programming channel features its own interchangeable 'I/O Connector Module' with relays and any custom target connectors
- Status indicators include - current channel being programmed and PASS / BUSY / FAIL
- Controllable via a PC using the Equinox EDS, ConsoleEDS or ISP-PRO software utilities
- Channel multiplexer control can be controlled either via the PC or a remote system

#### ISPnano-MUX Programmer range....

There are currently three programmers in the '*ISPnano-MUX*' family as detailed in the table below....

Programmer version:	ISPnano-MUX2	ISPnano-MUX4	ISPnano-MUX8
Number of multiplexed programming channels	2	4	8
No. of programming channels (RS485 networked)	up to 64 channels	up to 128 channels	up to 256 channels
Intended use / application	Programming 2 devices on the same PCB e.g. microcontroller + memory device	Programming 4 x PCBs on a multi-PCB panel	Programming 8 x PCBs on a multi-PCB panel
Availability	Q2 2011	Q2 2011	Available now

The picture below shows the '*ISPnano-MUX8*' programmer...



### Sequential multiplexing...

The sequential multiplexing approach supports programming of only one UUT at a time which means that the programming time for a panel of e.g. 2 / 4 / 8 PCBs which will be significantly longer than if an individual programmer was used for each channel. However, for many programming applications, the programming time is very short anyway or is not the most important factor in the system, so this multiplexed approach would work very well.

### Multiplexing of programming signals...

The programming signals for each programming channel are multiplexed using high-quality analogue switches to each UUT in turn. The power signals (TVCC, VPP and EXT-VCC) are all individually switched to each UUT using relays. This approach allows the power to each UUT to be individually controlled and measured and means that only a single UUT is then powered on at any point in time making it possible to detect short circuits and other faults for an individual UUT.

### Operating the programmer in 'Standalone Mode'....

The *ISPnano-MUX* programmers are capable of operating in '*Standalone Mode*' where any one of 64 independent '*Programming Projects*' can be executed via a remote command. The programmer also features a 4-wire TTL Remote Control interface making it ideal for interfacing to any ATE or In-Circuit Tester (ICT). A special '*Start Signal*' can be used to detect when a Test Fixture lid has been closed and thereby automatically start a programming operation.

## 1.4 ISPnano Programmer Module - Main Features

The main features of the '*ISPnano Programming Module*' are detailed in this section. This module is found in the ISPnano Programmer, ISPnano-ATE programmer and also the ISPnano-MUX2 / MUX4 / MUX programming Systems.

### Wide ranging Device Support capability

- Supports In-System Programming (ISP) of many popular FLASH Microcontrollers, Serial EEPROM and serial FLASH Memories

### High-speed Programming

- Optimised algorithms, on-board project data storage and high-speed line-driver circuitry delivers the fastest possible programming times

### Supports most ISP Protocols

- SPI, JTAG, I2C (2-wire), SCI, BDM,
- Series 3 programmers only – Atmel XMEGA 2-wire PDI, ATtiny TPI(LV), ATtiny TPI(HV)

### High-speed JTAG port

- Supports high-speed JTAG programming of Atmel AVR and AT91SAM7 microcontrollers

### High-speed SPI port

- Supports full range of SPI speeds from 10 Hz up to 4 MHz

### Dedicated 2-wire I2C serial port with independent drive circuitry

- Supports high-speed I2C programming of Serial EEPROM and F-RAM memories.

### Fully ESD and Over-voltage protected I/O

- All Target I/O pins feature both ESD and over-voltage protection

### Supports programming at Target Voltages down to 1.8V

- An optimised driver circuit delivers fast clean programming waveforms from 1.8 to 5.0V.

### Excellent Host Control connectivity

#### ISPnano Series 1

- 2 x RS232 Serial Ports
- 1 x RS485 In/OUT port – supports up to 32 programmers on an RS485 network
- 4-wire TTL Remote Control Port
- Remote Keypad / Display (ideal for standalone control in a Test Fixture)

#### ISPnano Series 2

- 1 x USB Port
- 1 x RS232 Serial Port
- 1 x RS485 In/OUT port – supports up to 32 programmers on an RS485 network
- 4-wire TTL Remote Control Port

- Remote Keypad / Display (ideal for standalone control in a Test Fixture)

### **Gang Programming support**

- Up to 32 programmers can be controlled from a single PC using the RS485 Bus

### **Standalone Operation**

- Programmer can operate in '**Standalone Mode**' i.e. without PC Control.
- Programmer can be controlled via [START] button, 4-wire TTL Remote control Port, Target Detection or Lid switch detection control methods or Remote Keypad / Display.

### **Supports up to 64 Independent 'Programming Projects'**

- Each project supports programming of a complete device inc. FLASH, EEPROM, Fuses etc.

### **High accuracy Programmer Controlled Target Vcc Power Supply**

- The programmer can switch 1.8 to 5.0V @ 300mA to the Target System.
- Accuracy +/- 20 mV

### **Programmer controlled Target Discharge Circuit**

- The programmer can automatically discharge any residual charge on the Target System to guarantee that the Target Vcc reaches 0V.

### **High accuracy Programmer Controlled Target Vpp Power Supply**

- The programmer can switch 6.5V to 13.5V @ 100mA to the Vpp pin of a Target IC.

### **User-swappable I/O Connector Modules (optional)**

- Many different I/O Connector Modules are available to suit any required connection system.

### **Firmware upgradeable**

- New algorithms and features can be added via a simple firmware upgrade

### **Compact physical size ideal for integration into Test Fixtures**

- Programmer measures only 9cm x 6cm x 6cm and has ergonomic design to supports the maximum packing density.

### **Special ATE programmer version with RELAY isolation available (option)**

- All programmer ISP and power signals are galvanically isolated via relays.

### **Special multiplexed programmer version available**

- A single ISPnano programmer is multiplexed sequentially to either 2, 4 or 8 Target Boards (UUTs) in turn.

## 1.5 Programmer Selection Guide

### 1.5.1 Single channel ISPnano programmers

This section describes the range of 'single channel' ISP programmers which are available.

Programmer version:	Series I	Series II	Series III	Series III ATE
Number of programming Channels (standard)	1	1	1	1
No. of programming Channels (networked)	up to 32 channels			
Intended use / application	Programming / Test fixtures, multi-PCB panel programming	Programming / Test fixtures, multi-PCB panel programming	Programming / Test fixtures, multi-PCB panel programming	Programming / ICT fixtures requiring relay isolation of ICT
Availability	Discontinued	Spares only	Available	Available

Please note:

- The '**Series I**' and '**Series II**' programmers have both now been discontinued. Some spares are still available on request.
- The '**Series III ATE**' programmer is an '**Series III**' programmer with an integrated '**Relay card**' which galvanically isolates all the programmer I/O and power signals.
- All of the above programmers can be networked via the RS485 bus to make a parallel programming network capable of concurrently programming up to 32 Target Boards (UUTs).

### 1.5.2 Multi-channel multiplexed ISPnano programmers

The *ISPnano-MUX* programmer range supports multiplexing of a single '*ISPnano Series 3*' programmer to 2 / 4 or 8 individual Target Boards (UUTs).

Programmer version:	ISPnano-MUX2	ISPnano-MUX4	ISPnano-MUX8
Number of multiplexed programming channels (standard)	2	4	8
Programming type	Single programmer is multiplexed to 2 x UUTs	Single programmer is multiplexed to 4 x UUTs	Single programmer is multiplexed to 8 x UUTs
No. of programming Channels (networked)	up to 64 channels	up to 128 channels	up to 256 channels
Intended use / application	Programming / Test fixtures, multi-PCB panel programming	Programming / Test fixtures, multi-PCB panel programming	Programming / Test fixtures, multi-PCB panel programming
Availability	Q2 2011	Q2 2011	Available

### 1.5.3 Programmer on-board Standalone Project Storage

Project Storage	Series I	Series II	Series III	Series III ATE
Number of standalone projects supported	64	64	64	64
On-board FLASH Project Storage (Mbytes)	16	16	16	16

### 1.5.4 Programmer Target I/O Capability

Target I/O Capability	Series I	Series II	Series III	Series III ATE
Programmable Target I/O	5 x I/O 1 x O			
I/O Voltage Range	1.8 – 5.0V	1.8 – 5.0V	1.8 – 5.0V	1.8 – 5.0V
I/O Isolation	-	-	-	Galvanic Relay isolation per I/O signal, power & 0V
ESD protection on Target I/O lines	IEC61000-4-2, air discharge: 15 kV  IEC61000-4-2, contact discharge: 8kV	IEC61000-4-2, air discharge: 15 kV  IEC61000-4-2, contact discharge: 8kV	IEC61000-4-2, air discharge: 15 kV  IEC61000-4-2, contact discharge: 8kV	IEC61000-4-2, air discharge: 15 kV  IEC61000-4-2, contact discharge: 8kV
Over-voltage protection on Target I/O lines	YES Voltages over 9V are clamped.			

### 1.5.5 Supported programming interfaces

Programming Interface	Series I	Series II	Series III	Series III ATE
SPI Port	Yes - multiplexed	Yes - multiplexed	Yes - multiplexed	Yes - multiplexed
JTAG Port	Yes - multiplexed	Yes - multiplexed	Yes - multiplexed	Yes - multiplexed
I2C Port	Yes - Dedicated	Yes - Dedicated	Yes – multiplexed with PDI port	Yes – multiplexed with PDI port
UART Port	Yes - multiplexed	Yes - multiplexed	Yes - multiplexed	Yes - multiplexed
XMEGA PDI Port	-	-	Yes – multiplexed with I2C port	Yes – multiplexed with I2C port
ATtiny TPI Port (LV)	-	-	Yes – multiplexed with I2C port	Yes – multiplexed with I2C port
ATtiny TPI Port (HV)	-	-	Yes – multiplexed with I2C port. +12V VPP on RESET pin	Yes – multiplexed with I2C port. +12V VPP on RESET pin
Programmable clock output	Yes 32 kHz	Yes 32 kHz	Yes 32 kHz	Yes 32 kHz

**Please note:**

- The ‘SPI Port’, ‘JTAG Port’ and ‘UART’ port signals all share the same physical programmer I/O lines.
- This means that the programmer can only be connected via one of these programming interfaces at any one time.

### 1.5.6 Programmer – Controlled Power Supplies

Programmer Power Supplies	Series I	Series II	Series III	Series III ATE
Programmable Vcc voltage (TVCC)	Yes 1.8–5.0V @480mA Tolerance: +/- 1%	Yes 1.8–5.0V @480mA Tolerance: +/- 1%	Yes 1.8 – 5.0V@480mA Tolerance: +/- 1%	Yes 1.8 – 5.0V@480mA Tolerance: +/- 1%
Programmable Vpp Voltage (TVPP)	Yes - 5.8 – 14.0V @ 100mA	Yes – 5.8 – 14.0V @ 100mA	Yes – 5.8 – 14.0V @ 100mA	Yes – 5.8 – 14.0V @ 100mA
External Switched Vcc supply	Yes – 1.0 – 24V	Yes – 1.0 – 24V	Yes – 1.0 – 24V	Yes – 12V only Used to power relay coils
Analogue voltage measurement	Yes Target Vcc (TVCC), Target Vpp (TVPP)	Yes Target Vcc (TVCC) Target Vpp (TVPP) Ext_Vcc IN/OUT	Yes Target Vcc (TVCC) Target Vpp (TVPP) Ext_Vcc IN/OUT	Yes Target (TVCC) Target (TVPP) Ext_Vcc IN/OUT
Target current measurement	Yes on TVCC only Accuracy: +/- 8%	Yes on TVCC only Accuracy: +/- 8%	Yes on TVCC only Accuracy: +/- 5%	Yes on TVCC only Accuracy: +/- 5%
Target over-current detection	YES on TVCC only	YES on TVCC only	YES on TVCC only	YES on TVCC only
Target controlled discharge circuit	Yes	Yes	Yes	Yes

### 1.5.7 Communication / Control Ports

Communications ports	Series I	Series II	Series III	Series III ATE
RS232 port	2	1	1	1
RS485 (In/Out)	1	1	1	1
USB port	-	1	1	1
ATE Control Port / TTL Remote Control	Y	Y	Y	Y

### 1.5.8 Programmer Control methods in Standalone Mode

It is possible to control an ISP nano programmer in '**Standalone Mode**' (no PC control) using many different '**hardware control interfaces**' as detailed in the table below.

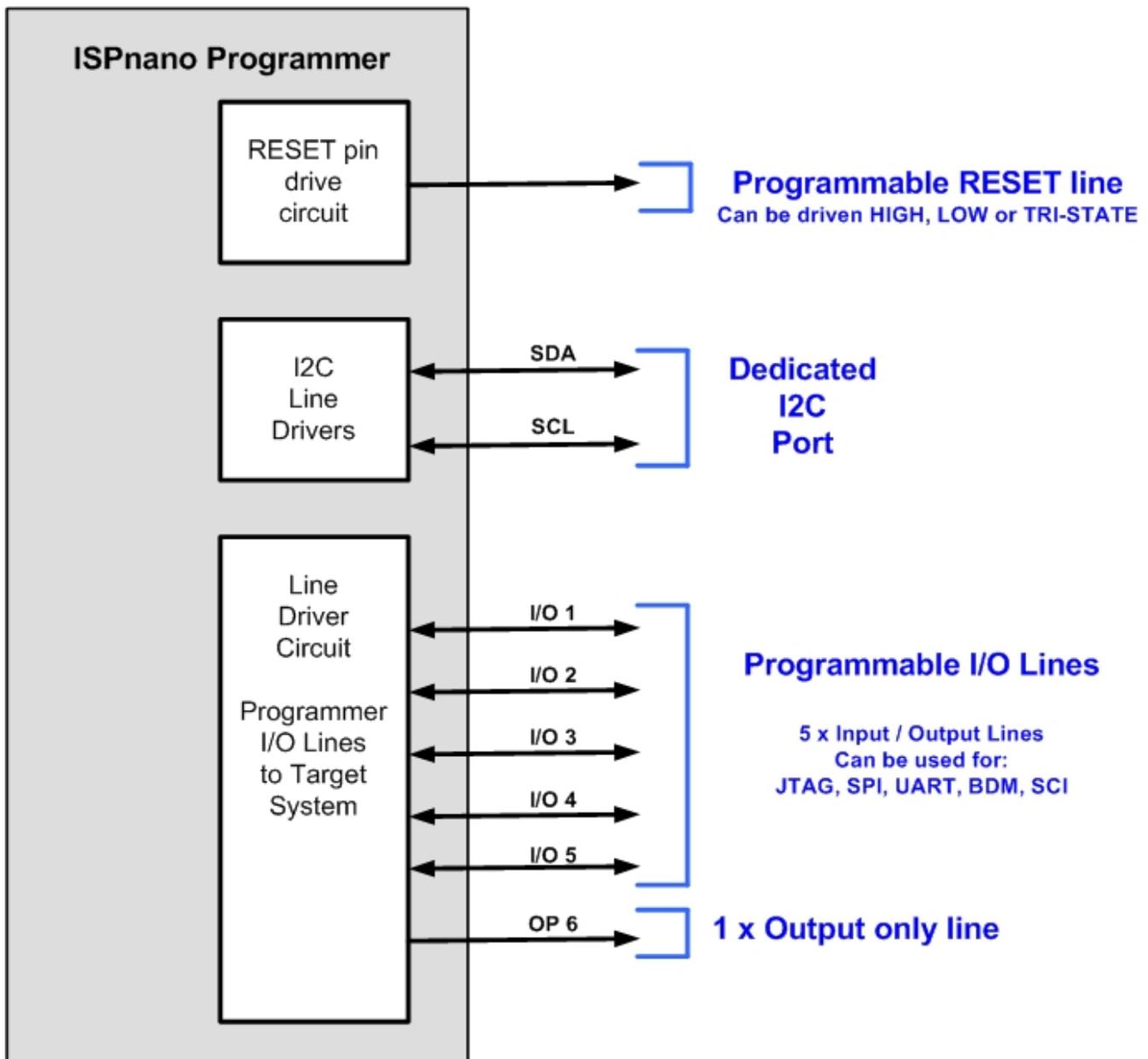
Control Methods	Series I	Series II	Series III	Series III ATE
ATE Control Port / TTL Remote Control	Y	Y	Y	Y
Target Sense Circuit	Y	Y	Y	Y
Fixture Lid START Switch	Y	Y	Y	Y
START Button	Y	Y	Y	Y
Remote Keypad / Display	Optional	Optional	Optional	Optional

## 1.6 Programmer – Target I/O Signals

The ISPnano features the following programmable signal lines which interface the programmer and the Target IC(s) to be programmed on the Target System:

- 5 x Programmable **Input / Output** lines
- 1 x Programmable **Output only** line
- 1 x Dedicated **I2C (Two Wire Interface)** port
- 1 x Dedicated **RESET** pin

The diagram below shows the available programmer I/O lines:



**Please note:**

- These I/O signals are available on the 16-way IDC Connector labelled "**ISP PORT**".

- See Appendix 1 for the pin-out details of this port.

## 1.7 System Contents

### 1.7.1 ISPnano Series 3 Kit

The ISPnano programmer is available as the ISPnano **'Programmer Module'** only or as a **'Programmer Kit'** which includes the Programmer Module along with a power supply, cables and software.

The full contents list of the **'ISPnano Programmer Kit'** (Order code: **ISPnano-S3KIT**) is detailed below.

#### Hardware

- ISPnano Programming Module (ISPnano-S3 or ISPnano-S3ATE version)
- Mains Power Supply Wall Adaptor (9V DC regulated @ 1.7A fitted with 2-pin JST connector)
- ISPnano – CONMOD Connector Module (ISPnano-CONMOD)
- ISPnano – EVAL Module (ISPnano-EVAL)

#### Cables

- DC Power Cable (2-pin JST to bare wires)
- RS232 Serial Cable (RJ11 to 9-way D connector)
- RS232 Extension Cable (9-way Female to 9-way Male Serial Cable)
- 6-way IDC ribbon cable
- 10-way IDC ribbon cable
- 16-way IDC ribbon cable
- 20-way IDC ribbon cable

#### Software

- EQTools (Project Management Utility for Equinox Production ISP Programmers)
- Upload Wizard (used to upload Standalone Projects to the programmer)

#### Miscellaneous

- 4 x Rubber mounting feet (separate packet)

### 1.7.2 ISPnano-MUX Programming System Kit

The ISPnano-MUX Programming Systems are available as a kit which includes the following items:

#### Hardware

- 1 x ISPnano-MUX Programming Module (ISPnano-MUX2 or ISPnano-MUX4 or ISPnano-MUX8)
- 1 x Mains Power Supply Wall Adaptor (9V DC regulated @ 1.7A fitted with 2-pin JST connector)
- 2 / 4 / 8 x IO-MOD1 ISPnano plug-in Connector Modules
- 1 x ISPnano – EVAL Module (ISPnano-EVAL)

#### Cables

- DC Power Cable (2-pin JST to bare wires)
- RS232 Serial Cable (RJ11 to 9-way D connector)
- RS232 Extension Cable (9-way Female to 9-way Male Serial Cable)
- 6-way IDC ribbon cable
- 10-way IDC ribbon cable
- 16-way IDC ribbon cable
- 20-way IDC ribbon cable

#### Software

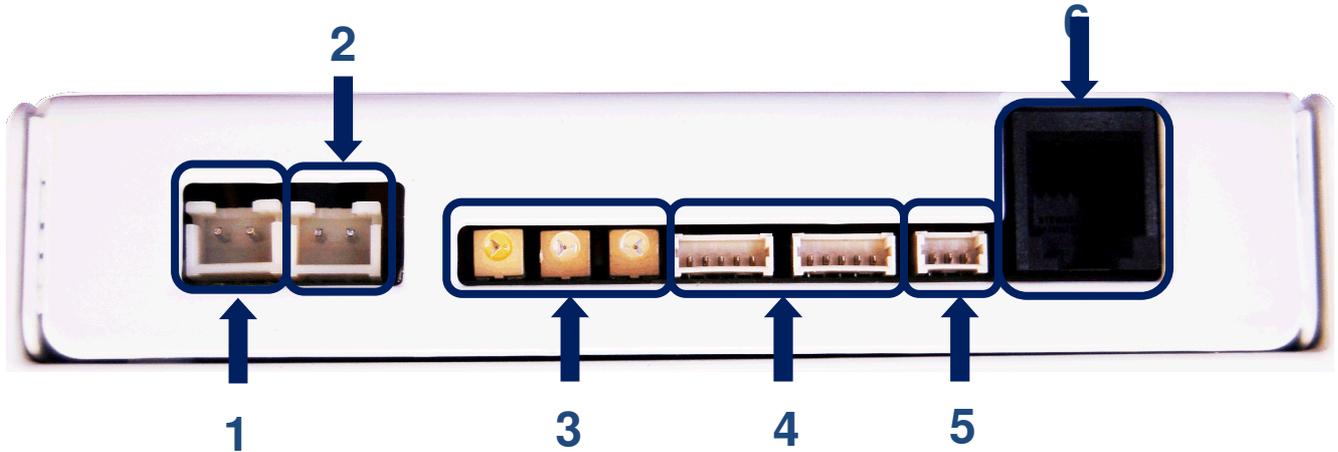
- EQTools (Project Management Utility for Equinox Production ISP Programmers)
- Upload Wizard (used to upload Standalone Projects to the programmer)

#### Miscellaneous

- 4 x Rubber mounting feet (separate packet)
- 4 x metal chassis mounting feet (to secure enclosure into a Test Fixture)

## 1.8 ISPnano Series I - Rear panel connections

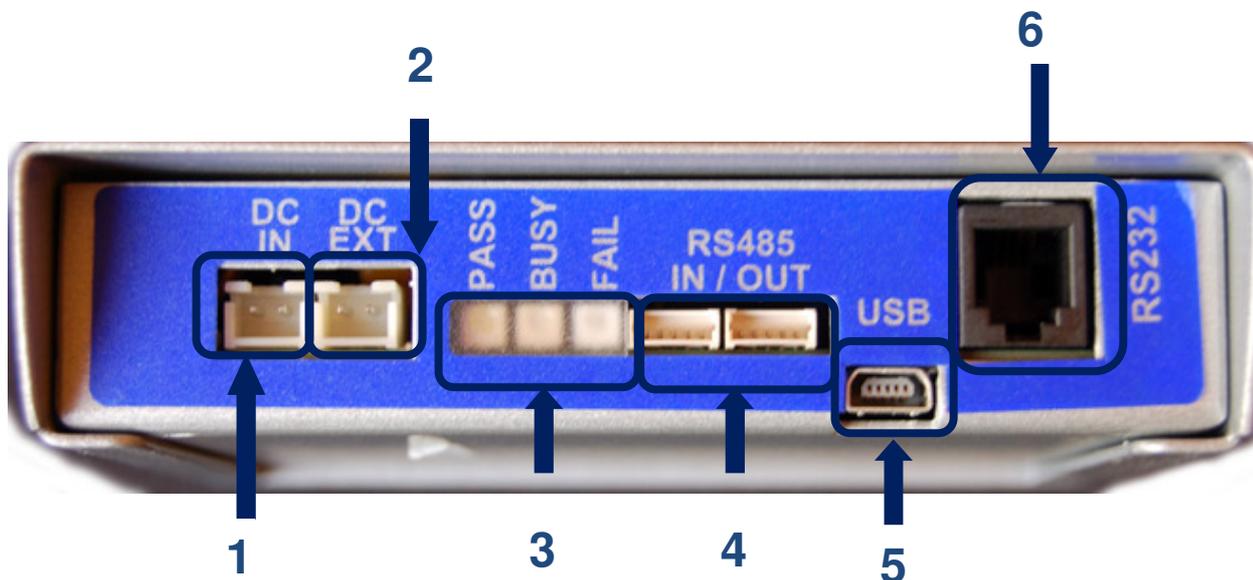
The illustration below details the function of each of the connectors on the rear panel of the programmer.



#	Connector Name	Description / comment
1	DC Power Input	<b>DC Power Input to power the programmer</b> <ul style="list-style-type: none"> <li>Voltage range: 9.0 – 24.0 V DC (right-hand pin positive)</li> </ul>
2	EXTERNAL TARGET VCC INPUT	<b>EXTERNAL TARGET VCC INPUT</b> <ul style="list-style-type: none"> <li>DC Power Input to power the programmer</li> <li>Voltage range: 3.0 – 24.0 V DC (right-hand pin positive)</li> </ul>
3	Programmer STATUS LEDs	<b>Programmer 'Status' LEDs</b> <ul style="list-style-type: none"> <li>From Left to right: <ul style="list-style-type: none"> <li>PASS (GREEN)</li> <li>BUSY (YELLOW)</li> <li>FAIL (RED)</li> </ul> </li> </ul>
4	RS485 Ports	<b>RS485 Ports (1) + (2) - Serial Communications Ports</b> <ul style="list-style-type: none"> <li>These port can be used to control the programmer via an RS485 link from a PC. They are connected in parallel internally so it does not matter which is used as the RS485 INPUT or OUTPUT.</li> <li>The connector is a 5-pin 2.54mm Molex.</li> </ul>
5	RS232 Port (1)	<b>RS232 Port (1) - Serial Communications Port</b> <ul style="list-style-type: none"> <li>This port can be used to control the programmer via an RS232 link from a PC or other Test Equipment.</li> <li>The connector is a 3-pin 2.5mm Molex.</li> </ul>
6	RS232 Port (2) or Remote Display Keypad	<b>RS232 Port (2) - Serial Communications Port</b> <ul style="list-style-type: none"> <li>This port can be used to control the programmer via an RS232 link from a PC or other Test Equipment.</li> <li>A '<b>Remote Display / Keypad Module</b>' can also be connected to this port allowing the programmer to be remote controlled.</li> <li>The connector is a 4-pin RJ11 connector.</li> </ul>

## 1.9 ISPnano Series II / III – Rear panel connections

The illustration below details the function of each of the connectors on the rear panel of the programmer.



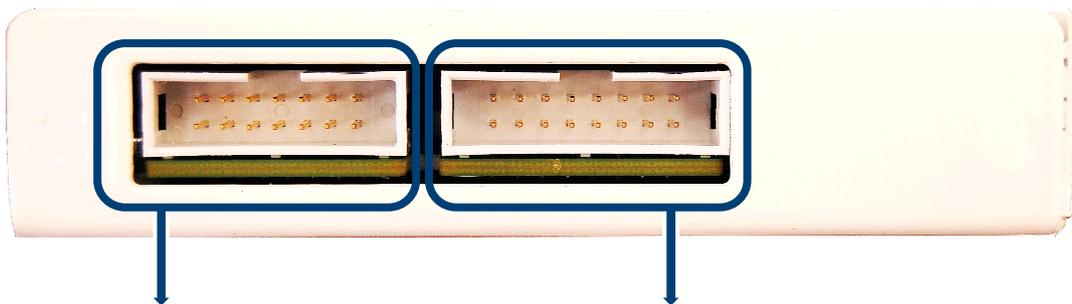
#	Connector Name	Description / comment
1	DC Power Input	<b>DC Power Input to power the programmer</b> <ul style="list-style-type: none"> <li>Voltage range: 9.0 – 24.0 V DC (right-hand pin positive)</li> </ul>
2	EXTERNAL TARGET VCC INPUT	<b>EXTERNAL TARGET VCC INPUT</b> <ul style="list-style-type: none"> <li>DC Power Input to power the programmer</li> <li>Voltage range: 3.0 – 24.0 V DC (right-hand pin positive)</li> </ul>
3	Programmer STATUS LEDs	<b>Programmer ‘Status’ LEDs</b> <ul style="list-style-type: none"> <li>From Left to right:                             <ul style="list-style-type: none"> <li>PASS (GREEN)</li> <li>BUSY (YELLOW)</li> <li>FAIL (RED)</li> </ul> </li> </ul>
4	RS485 Ports	<b>RS485 Ports (1) + (2) - Serial Communications Ports</b> <ul style="list-style-type: none"> <li>These port can be used to control the programmer via an RS485 link from a PC. They are connected in parallel internally so it does not matter which is used as the RS485 INPUT or OUTPUT.</li> <li>The connector is a 5-pin 2.54mm Molex.</li> </ul>
5	USB Port	<b>USB Port</b> <ul style="list-style-type: none"> <li>This port can be used to control the programmer via a USB link from a PC or other Test Equipment.</li> </ul>
6	RS232 Port (2) or Remote Display Keypad	<b>RS232 Port (2) - Serial Communications Port</b> <ul style="list-style-type: none"> <li>This port can be used to control the programmer via an RS232 link from a PC or other Test Equipment.</li> </ul>

- A '**Remote Display / Keypad Module**' can also be connected to this port allowing the programmer to be remote controlled.
- The connector is a 4-pin RJ11 connector.

## 1.10 ISPnano - Front panel connections

### 1.10.1 Overview

The illustration below details the function of each of the connectors on the rear panel of the programmer.



#### Programmer Remote Control Port

- This port is used to control the programmer from e.g. an **ATE** via a "**4-wire TTL Interface**".
- The "**Remote Status LEDs**" PASS, BUSY, FAIL are also on this port. This allows LEDs to be mounted on the lid of the Test Fixture for easier visibility.

#### Target ISP Connection Port

- This Port features all the Target I/O Signals."
- 5 x Programmer controlled I/O lines Multiplexed for SPI, JTAG, UART, BDM, PDI
- 1 x Programmer Output line (e.g. Relay control)
- Dedicated 2-wire I2C Port
- Controlled Target Vcc Supply

### 1.10.2 Programmer Remote Control Port

The '**Remote Control**' port features all the signals required to implement the following functionality:

#### i. Remote Control of the programmer via 4 TTL signals: START, BUSY, PASS, FAIL

This allows the programmer to be controlled by an external system using 4 TTL control signals such as a PLC.

#### ii. Remote Programmer Status LEDs

This allows the programmer Status LEDs to be mounted remotely from the programmer. e.g. on the Test Fixture panel itself.

#### iii. Remote LID Start Switch

This input allows can be connected to the "**Lid Switch**" on a Test Fixture and allows the programmer to automatically commence programming when the Test Fixture lid is closed.

Please see Appendix 2 for the pin-out of this connector.

### **1.10.3 Programmer ‘Target ISP Connector Port’**

The ‘**Target ISP Connector**’ port features all the signals required to implement In-System Programming of a Target IC using SPI, JTAG, I2C or UART interface. This connector also features a programmable “**Target Vcc**” and “**Target Vpp**” supply.

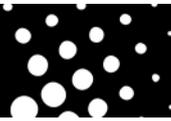
Please see Appendix 1 for the pin-out of this connector.

## 1.11 Programmer Specifications

The table below details the hardware specifications for the programmer. Please refer to the stated section for further information about any specific parameter.

#	Parameter	Description / comment	Refer to section
1	<b>Target Device Support</b>	See Device Support List (separate document)	
2	<b>Target Device Programming Interfaces</b>	<p>The ISPnano Series I and II programmers supports the following programming interfaces:</p> <ul style="list-style-type: none"> <li>• Atmel Low Voltage SPI</li> <li>• Atmel High Voltage SPI (+12V Vpp)</li> <li>• Atmel AVR JTAG (chargeable update)</li> <li>• Atmel (formerly Temic) 8051 Boot loader ISP</li> <li>• NXP (Philips) 8051 - Boot Loader ISP</li> <li>• Generic I2C (SDA + SCL)</li> <li>• ARM7 JTAG Debug Interface</li> </ul> <p>The ISPnano Series III programmer supports all of the above programming interfaces plus the following additional interfaces:</p> <ul style="list-style-type: none"> <li>• Atmel XMEGA 2-wire PDI interface</li> <li>• Atmel ATtiny 2-wire TPI interface</li> </ul>	
3	<b>Operating modes</b>	<p>The programmer supports the following operating modes:</p> <ul style="list-style-type: none"> <li>• Development Mode (PC controlled)</li> <li>• Project Upload Mode (PC controlled)</li> <li>• Standalone Mode – Single Start button mode</li> <li>• Standalone Mode – Automatic Target Connect / Disconnect Sensing</li> <li>• Standalone Mode - Remote Controlled via 'ASCII Text Communications' RS-232 Serial Protocol</li> <li>• Standalone Mode - Remote Controlled via 4-wire TTL Remote Control Port</li> <li>• PC controlled via ISP-PRO software (chargeable upgrade)</li> <li>• PC controlled via ConsoleEDS utility (chargeable upgrade)</li> </ul>	
4	<b>On-board FLASH Memory Store</b>	<ul style="list-style-type: none"> <li>• 128 Mbits (16 MBytes) FLASH Memory for Project Storage</li> </ul>	
5	<b>Project storage in Memory Store</b>	64 x Programming Projects	
6	<b>Keypad entry</b>	1 x <START> button	
7	<b>Status LED's</b>	<ol style="list-style-type: none"> <li>i. 3 x Status LED's located on the programmer: &lt;PASS&gt;, &lt;BUSY&gt;, &lt;FAIL&gt;</li> <li>ii. 3 x remote Status LED's can also be connected off-board on a Test Fixture via the 'Remote System Status' port.</li> <li>iii. 1 x Bi-coloured status LED</li> </ol>	
10	<b>PC Control Software</b>	<p>The programmer can be controlled using:</p> <ul style="list-style-type: none"> <li>• EQTools (as standard)</li> </ul>	

		<ul style="list-style-type: none"> <li>• ASCII Text Communications Mode (as standard)</li> <li>• ConsoleEDS (chargeable upgrade)</li> <li>• ISP-PRO (chargeable upgrade)</li> </ul>	
11	<b>EQTools / ISP-PRO PC requirements</b>	PC running Windows 95 / 98 / 2000 / ME / NT4 / XP / Windows 7	
12	<b>Power Supply Options</b>	<p>There are 2 possible modes for powering the Programmer / Target System:</p> <ul style="list-style-type: none"> <li>• Programmer controls power to the Target System</li> <li>• Programmer and Target System are independently powered</li> </ul>	
13	<b>Voltage range</b>	<ul style="list-style-type: none"> <li>• Device Programming voltage range: 1.8 – 5.0V</li> <li>• Programmer Controlled Power Supply range: 1.8 – 5.0V</li> <li>• Possible Target System voltage range: 1.8 – 5.0V</li> </ul>	
14	<b>Vpp Voltage Generator</b>	<ul style="list-style-type: none"> <li>• User-programmable 6.5V – 13.5V Vpp Voltage Generator</li> </ul>	
15	<b>DC Power Connector (2-pin JST connector)</b>	<p>DC Power Connector for powering the programmer</p> <ul style="list-style-type: none"> <li>• 2-pin JST connector</li> <li>• 9.0V – 24.0V DC regulated @ 500mA (worst case)</li> <li>• Mates with Power Supply Adaptor supplied with programmer</li> <li>• DC Power Lead fitted with 2-pin JST connector is also supplied with programmer if a custom power supply is to be used.</li> </ul>	
16	<b>External DC Power Connector (2-pin JST connector)</b>	<p>External DC Power Connector for externally powering the Target System</p> <ul style="list-style-type: none"> <li>• 2-pin JST connector</li> <li>• 3.0V – 24.0V DC regulated @ 500mA (worst case)</li> <li>• This supply can be used to switch an external power supply to the Target System under programmer control.</li> </ul>	
18	<b>Target SPI Frequency</b>	<p><b>Software SPI algorithm:</b></p> <ul style="list-style-type: none"> <li>• SLOW SPI: 0 – 490.2 kHz (user selectable)</li> <li>• MEDIUM SPI: 0 – 490.2 kHz (user selectable)</li> </ul> <p>( SPI speeds are estimations only due to uneven mark/space ratio and non-continuous waveforms)</p> <p><b>Hardware SPI algorithm</b></p> <p>SLOW SPI: 115.2 kHz to 3.6864 MHz (user selectable)</p> <ul style="list-style-type: none"> <li>• FAST SPI: 115.2 kHz to 3.6864 MHz (user selectable)</li> </ul>	



19	<b>Target JTAG Frequency</b>	<p>The programmer supports a user-defined JTAG frequency:</p> <ul style="list-style-type: none"> <li>• SLOW JTAG: 0 – 833 kHz</li> <li>• FAST JTAG: approx 1.4 MHz</li> </ul>	
20	<b>JTAG Programming Modes</b>	<p><b>The programmer supports the following JTAG modes:</b></p> <ul style="list-style-type: none"> <li>• Single JTAG device connected to programmer</li> <li>• Multiple JTAG devices connected in a JTAG Chain</li> <li>• JTAG ID can be verified for both Atmel AVR, AT91SAM7 and non-Atmel JTAG devices</li> <li>• JTAG chain validation is also supported</li> </ul>	
21	<b>I2C Frequency</b>	<p>The programmer supports a user-defined I2C frequency:</p> <ul style="list-style-type: none"> <li>• SLOW I2C: 0 – 400 kHz</li> <li>• Supports ISP programming of both 100 kHz and 400 kHz I2C Serial EEPROMs</li> </ul>	
22	<b>Target UART Speed</b>	<ul style="list-style-type: none"> <li>• This is the Communications Speed (BAUD rate) from the programmer to the Target Device when communicating via a Boot Loader.</li> <li>• The BAUD rate is configurable from 1,200 to 115 kBaud.</li> <li>• This settings is completely independent from the BAUD rate settings for the PC communicating with the programmer.</li> </ul>	
23	<b>ISP Headers</b>	<p>The programmer can support any ISP header by simply plugging in the relevant 'ISPnano Pin-out Converter Module'.</p> <p>The currently supported ISP Headers are as follows:</p> <ul style="list-style-type: none"> <li>• Atmel 10-way (SPI)</li> <li>• Atmel 6-way (SPI)</li> <li>• Atmel 10-way (JTAG)</li> <li>• Atmel 20-way (JTAG) for AT91SAM7 microcontrollers</li> <li>• Equinox 10-way (SPI+UART)</li> <li>• Fast-connect wire connectors</li> </ul>	
24	<b>Frequency generator output (SCK2)</b>	<p>Programmer can output a configurable square wave frequency on the SCK2 pin. This is used to clock the Target Device in the absence of a Target Oscillator.</p> <ul style="list-style-type: none"> <li>• Frequency range: 167kHz – 1.8432 MHz</li> </ul>	
25	<b>AVR On-chip Oscillator Calibration</b>	<p>The programmer can generate a suitable square wave for use with 'AVR on-chip oscillator calibration'.</p>	

25	PC Connection	<ul style="list-style-type: none"><li>i. <b>RS232-1 – RS232 Serial Channel 1</b><ul style="list-style-type: none"><li>• 3-pin Molex connector</li></ul></li><li>ii. <b>RS232-2 – RS232 Serial Channel 2 or Remote Keypad</b><ul style="list-style-type: none"><li>• RJ11 connector</li></ul></li><li>iii. <b>RS-485 Multiple Programmer Channel Control</b><ul style="list-style-type: none"><li>• Programmers are daisy chained via RS-485 leads</li><li>• An RS-232 to RS-485 converter is required at the PC end of the chain.</li></ul></li></ul>	
26	Temperature range	0 to 50 deg. C	

## 1.12 Programmer on-board FLASH Memory Store

The ISPnano programmers feature 128 MBits (16 Mbytes) of on-board non-volatile FLASH storage. This memory area can be used to permanently store up to 64 '**Standalone Programming Projects**'. The projects are fully CRC checked and are retained in FLASH memory once uploaded.

## 2.0 Getting Started Guide

### 2.1 Overview

This section gives a '**Quick Start**' Guide to using the programmer. Please refer to the table below for help installing the programmer hardware and software and also for the operating instructions for the programmer.

Description	Refer to section
Hardware Installation Procedure	2.2
Software Installation Overview	2.3
Programmer Operating Modes	2.4
Development Mode (EDS)	2.5
Standalone - Keypad Mode	2.6
Standalone – ASCII Text Communications Control	2.7
Standalone – 4-Wire TTL Control	2.8
Target System connect / Disconnect Detection	2.9
Script Mode	2.10
ConsoleEDS	2.11

Further help and instructions can be found in the following documents:

1. '**EQTools – Getting Started Guide**' (pdf manual) supplied on CD-ROM which came with the programmer (also available on-line)
2. '**EQTools – On-line Help File**' (Press <F1> within EQTools).

## 2.2 Hardware Installation Procedure

Please follow the instructions below to set up the programmer hardware:

### 2.2.1 ESD Precautions



- Ensure you are wearing an ESD strap and/or you are working in an ESD protected environment.
- The programmer I/O lines are ESD protected, but it always good practice to take proper ESD precautions. This can help to avoid ESD damage to either the programmer or Target System.

### 2.2.2 Unpacking the programmer and accessories

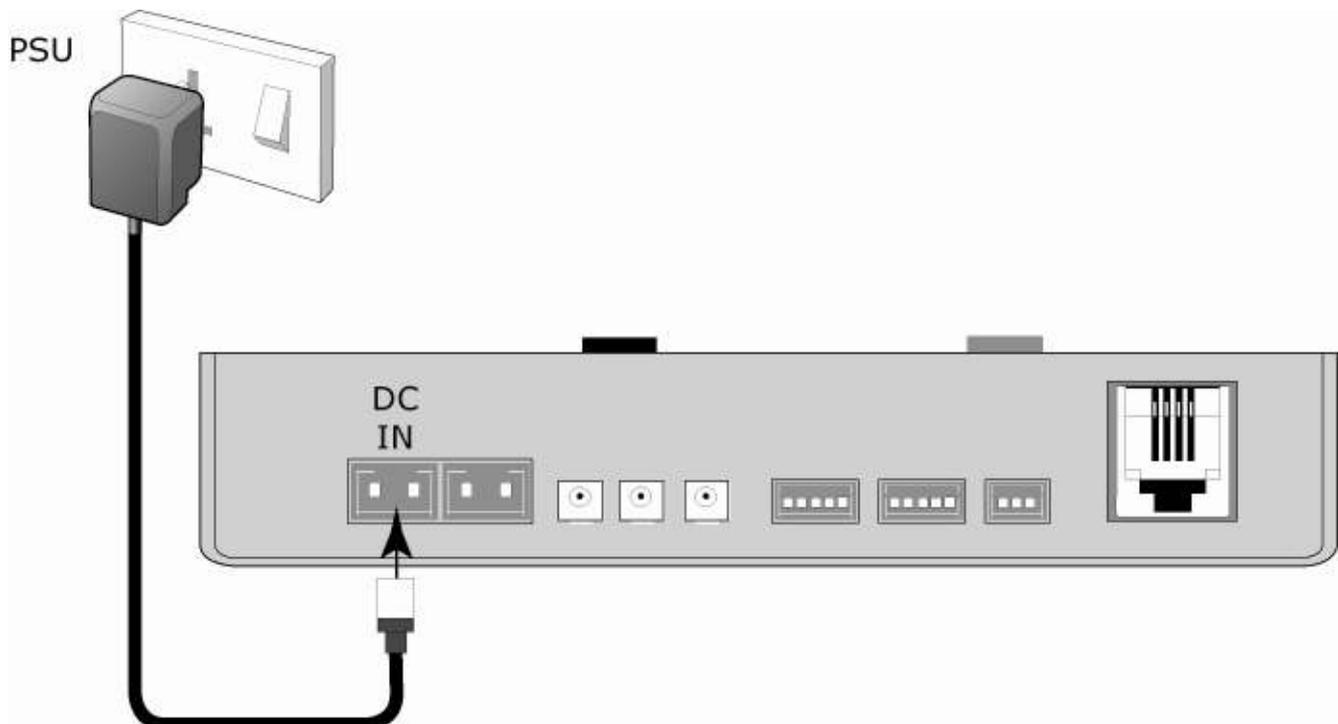
The ISPnano '**programmer kit**' comes complete with the IPSnano programmer, CONMOD Module, RS232 serial cable, USB cable and various ISP cables. Please see the '**System Contents**' section.

### 2.2.3 Selecting the Communications Mode

The ISPnano programmer range can be controlled via RS232, RS485 or USB. Please see section 3 for a full explanation of each communications mode.

### 2.2.4 Powering the ISPnano programmer

The ISPnano programmer requires a power supply of 9.0 – 24.0V to be connected to the '**DC IN**' 2-pin JST connector on the rear panel of the programmer – see illustration below.



The ISPnano kit comes complete with a 9 volt DC universal power supply fitted with a 2-pin JST connector.

- Plug the 2-pin JST lead into '**DC IN**' connector on the rear panel of the programmer.
- Plug the universal power supply adaptor into a suitable mains socket and switch the power on.
- → The programmer should now power up and the '**Status**' LED should illuminate.

## 2.3 Software Overview and Installation

### 2.3.1 Software Overview

The ISPnano programmer is supplied with a comprehensive Integrated Development Environment (IDE) called the '**EQTools Software Suite**'. This software supports creation and testing of so-called '**Standalone Programming Projects**' which can then be uploaded to the programmer for use in '**Standalone Mode**'. EQTools contains many separate utilities most of which are enabled as standard for this programmer. Certain options such as compiling and running of '**Programming Scripts**' must be purchased from Equinox as a '**License upgrade**'.

The table below details the various EQTools components and whether they are enabled as standard.

Fig. 2.3.1 Software / Control Options for the ISPnano Programmer

EQTools Software	Enabled as standard
• EQTools - Project Builder	YES
• EQTools - Project Management Utility	YES
• EQTools - Project Upload/Download Utility (Upload Wizard)	YES
• EQTools - Development Suite (EDS)	YES
• EQTools - Script Builder	U
• EQTools - Script Editor	U
• EQTools – Incremental Repository (Serial number / MAC address generator)	U
• EQTools - Development Mode (PC Controlled)	YES
<b>Programmer Control mechanisms</b>	
• START button (Standalone operation)	YES
• START signal (used to start sequence using switch in Fixture lid)	YES
• Remote System Control (4-wire)	YES
• ASCII Text Mode Control	YES
<b>ConsoleEDS – command-line utility</b>	
• ConsoleEDS – Evaluation version	YES
• ConsoleEDS – Standard version	U
• ConsoleEDS – PRO version	U
<b>ISP-PRO – production programming utility</b>	U
• ISP-PRO - Network Management Utility	U
• Interface Database (Diagnostics Log)	U
• Labview Control (PC)	U
• Remote Application Control (PC)	U

#### Key

YES – feature available as standard

U – a license upgrade must be purchased from Equinox to enable this feature.

### 2.3.2 Programmer Control Mechanisms

The programmer can be controlled by many different methods including EQTools, EDS, from a Remote System via the Equinox ISP-PRO software or via the ConsoleEDS command-line software. An overview of the available control mechanisms is given below.

#### **Standalone Operation (Keypad)**

The programmer can be operated using the '**START**' button without being connected to a PC or other control system. This will execute the first project found in the programmer memory.

#### **Remote System Control (4-wire)**

This indicates that the programmer can be controlled from a remote electrical system e.g. In Circuit Tester (ICT) or PLC using a 4-wire TTL protocol (START, ACTIVE, PASS, FAIL). This method only supports execution of 'Standalone Programming Projects' which have been previously uploaded to the programmer.

#### **Test Fixture - Jig Switch Control (1-wire)**

A switch on the Test Fixture lid can be used to automatically start a '**Standalone Programming Project**' in the programmer.

#### **ASCII Text Mode Control**

This indicates that the programmer can be controlled using a simple '**ASCII Serial Communications Protocol**'. This is ideal for applications where a Remote System wishes to control the programmer using a serial communications link e.g. RS-232. This method only supports execution of 'Standalone Programming Projects' which have been previously uploaded to the programmer.

#### **ISP-PRO Software**

This software suite is capable of controlling up to 32 x Equinox Production ISP programmers on an RS-485 network at the same time. All programming operations are controlled via '**Programming Scripts**' and '**Programming Projects**' which have been created using EQTools. The application minimises user intervention during programming and logs all programming operations to an Interface Database.

#### **Labview Control (PC)**

This allows an Equinox programmer to be controlled by a remote Labview Application. The Labview application can launch 'Programming Scripts' and read/write unique data such as serial numbers and calibration data.

#### **Remote Application Control (PC)**

This allows an Equinox programmer to be controlled by a Remote Application written in e.g. Visual Basic, C++ etc. The Remote Application can launch '**Programming Scripts**' and read/write unique data such as serial numbers and calibration data.

#### **ConsoleEDS**

This is a powerful utility which allows any Equinox ISP Programmer to be controlled from a PC application by executing simple command-line instructions. This method of control is ideal for interfacing the programmer to any Windows application including Visual Basic, Labview etc.

### 2.3.3 Software Installation

The ISPnano programmer comes with the software suite called '**EQTools**' as standard. The latest version of this software which was available at the time of shipping is supplied on CD-ROM with the programmer. However, this software version is likely to have been superseded by a newer version.

Please check the Equinox website for the latest version of this software...

► **Internet Web Site**

The support / download page for the '**ISPnano programmer range**' can be found at:

<http://www.equinox-tech.com/products/details.asp?ID=1440&displ=tl>

EQTools versions are quoted as e.g. **EQTools V4 Build 2661**

The filename for this version would be '**eqtools-V4-b2661.exe**'

For further information, please locate the '**EQTools – Getting Started Guide**' PDF document either on the CD-ROM supplied with the programmer or from the Equinox Web Site.

Please refer to section 2 of the guide for detailed EQTools installation instructions.

## 2.4 Programmer Operating Modes

The ISPnano programmer can be controlled in various different ways as detailed in the table below:

#	Programmer Mode	Control Software	U/S	Functional description	Refer to section
1	<b>Development Mode</b>	EQTools - EDS	S	In this mode, the programmer is controlled by EQTools - Equinox Development Suite (EDS). EDS allows you to read/write the target device FLASH / EEPROM / FUSE / LOCK bits interactively under PC control without uploading a 'Programming Project' to the programmer.	2.5
2	<b>Standalone - Keypad Mode</b>	None	S	In this mode, the programmer is not connected to a PC and can program a single ' <b>Programming Project</b> ' into the Target Device when the <b>&lt;START&gt;</b> key is pressed. (A valid project must have been previously uploaded to the programmer using EQTools – Upload Wizard.)	2.6
3	<b>Standalone - ASCII Text Communications Control</b>	Any Terminal Emulation Software	S	This protocol can be used to control an Equinox programmer from a Remote Application or Remote System using a simple set of ASCII serial commands. This protocol is ideal for controlling Equinox programmers from any Remote System which features an RS-232 serial port and which can operate at a fixed baud rate of 38,400.	2.7
4	<b>Standalone - 4-Wire TTL Control</b>	None	S	The programmer can be controlled using a 4-wire TTL signalling protocol as shown in the diagram below. This control method is ideal for interfacing the programmer In-Circuit Testers (ICT's) or other production equipment which does not have an RS-232 port.	2.8
5	<b>Target System auto-detect mode</b>	None	S	The programmer is capable of automatically detecting when a Target System is connected to / disconnected from the programmer. This can be used to trigger the execution of a Programming Project in Standalone Mode.	2.9

6	<b>Script Mode</b>	EQTools – Script Debugger or ISP-PRO	U	In this mode, the programmer is controlled from a 'Programming Script' running within either the EQTools or ISP-PRO applications on a PC.	2.10
7	<b>ConsoleEDS</b>	ConsoleEDS	U	In this mode, the programmer is controlled via the ConsoleEDS utility which allows control via simple Command Line commands and arguments	2.11 See App Note AN111

**Key**

S – feature available as standard

U – a license upgrade must be purchased from Equinox to enable this feature.

## 2.5 Programmer Status LEDs

The current status of the programmer is displayed on the programmer Status LED's as detailed in the table below.

Fig. 2.6.5 Programmer Status LED's – state descriptions

Status LED Display	State	State Description
	<b>WAITING</b>	<ul style="list-style-type: none"> <li>Programmer is now waiting to start an 'Autoprogram' operation.</li> </ul>
	<b>BUSY</b>	<ul style="list-style-type: none"> <li>Programmer is 'BUSY' performing a programming operation.</li> <li>If the programmer is controlled from EDS, the BUSY LED will remain on after a programming operation until a 'RESET programmer' command is executed.</li> </ul>
	<b>FAIL</b>	<ul style="list-style-type: none"> <li>Programming operation has FAILED.</li> <li>See programmer LCD for diagnostics</li> <li>Programmer will automatically switch off target power if it is controlling power.</li> </ul>
	<b>PASS</b>	<ul style="list-style-type: none"> <li>Programming operation was successful.</li> </ul>

### Status LED key:

ON FLASHING OFF

### Please note:

- There are also two special modes called '**Waiting Target Connection**' or '**Waiting Target Disconnection**' where the yellow BUSY LED will FLASH.
- These modes are enabled when '**Target Connection Sense**' is enabled in the Programming Project.

## 2.6 ASCII Text Communications Mode

### 2.6.1 Overview

The '**ASCII Text Control**' communications protocol or '**ATC**' is designed as a simple way for a Remote Test System to control the basic programming operations of an Equinox programmer via an RS-232 serial link. It is an 'ASCII' protocol with simple commands to initiate programming of pre-loaded '**Standalone Programming Projects**' which are stored in the on-board '**FLASH Memory Store**' of the programmer. The protocol supports up to 64 independent Programming Projects per programmer, but is limited to one programmer.

Implementation of this protocol is straightforward due to the limited number of commands and restricted functionality. As this is an ASCII protocol, it can be quickly evaluated using any Terminal Emulation Software Utility such as Hyperterminal.

### 2.6.2 Limitations of this protocol

This protocol has been designed as a simple method to allow a Remote Application such as an In-Circuit Tester to control the basic operations of an Equinox programmer. The programmer only supports the programming of entire 'Programming Projects' which are already uploaded to the programmer on-board FLASH memory. This protocol does not support writing / reading of individual bytes or blocks of data to / from a Target Device.

The limitations of this protocol are as follows:

1. The protocol only supports programming of Programming Projects which are pre-loaded into the Programmer on-board FLASH Memory Store.
2. The protocol does not support writing / reading of individual bytes or blocks of data to / from a Target Device.
3. The protocol does not support uploading of '**Standalone Programming Projects**' to the programmer. This must be performed using the '**EQTools – Project Upload Wizard**'.
4. Only very limited diagnostics are available using this protocol ie. FAIL + Error Number.

### 2.6.3 Further Information

A full description of this protocol can be found in the following Application Note:

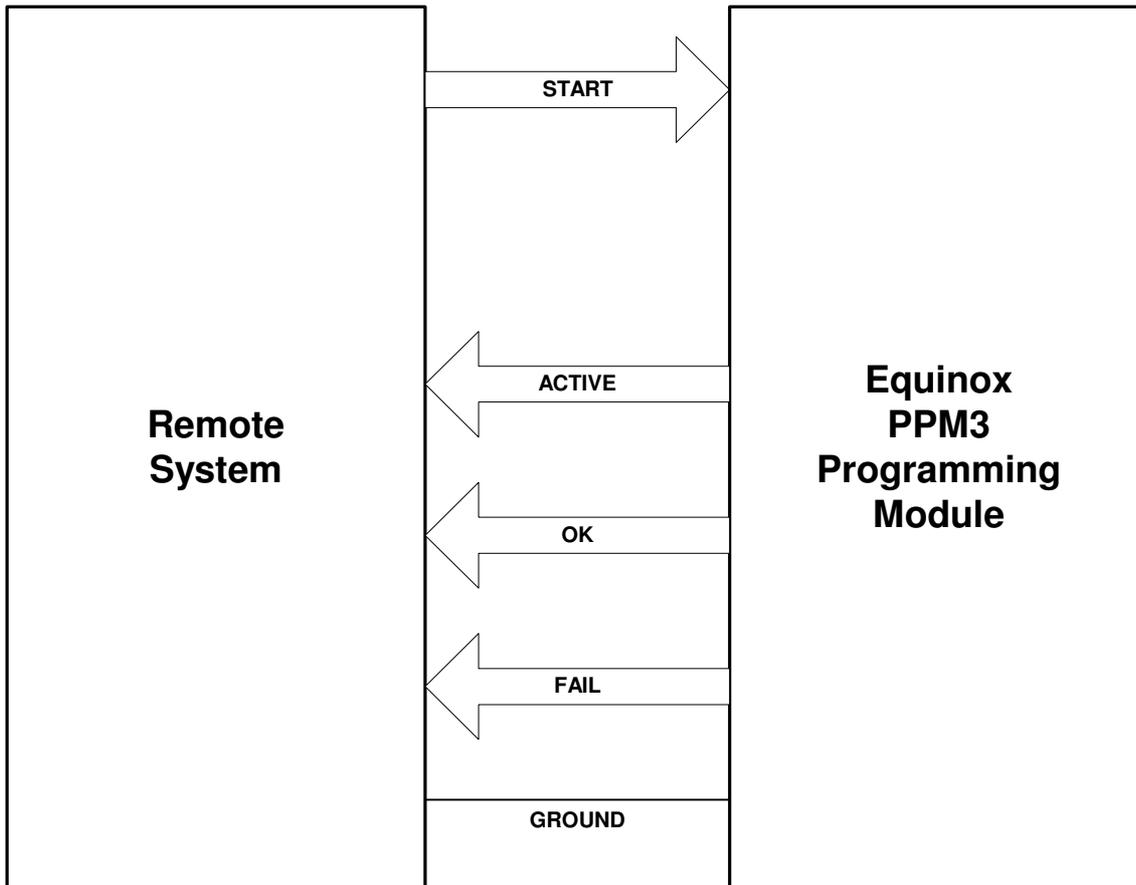
**'AN110 - ASCII Text Control (ATC) Protocol for Remote Control of Equinox Programmers'**.

## 2.7 Standalone Remote I/O Programmer Control

### 2.7.1 Overview

The ISPnano programmer can be controlled using a simple 4-wire Remote I/O (TTL) signalling protocol as shown in the diagram below. This control method is ideal for interfacing the programmer to In-Circuit Testers (ICT's) or other production equipment which does not have an RS-232 port.

Fig. 2.8.1 Remote System Control of Programmer



The Remote System simply asserts the programmer **<START>** signal to initiate the execution of a 'Programming Project'. The programmer will then assert the **<ACTIVE>** signal to indicate that it has commenced programming. At the end of the Programming operation the programmer asserts either the **<OK>** or **<FAIL>** signal depending on the outcome. The Remote System must then de-assert the **<Start>** signal to allow the programmer to reset ready to program the next device.

Please refer to section 2.8.3 for an example of the Remote I/O control signal waveforms for a typical programming iteration.

## 2.7.2 TTL 4-Wire Remote Control – Signal Connections

The Remote System connects to the programmer via the '**Remote Control Port**' which is the 14-way IDC connector. The signal names and descriptions are detailed in fig. 2.8.2.

Fig. 2.8.2 Remote System Control Port – signal names and descriptions

Pin No	Programmer Pin name	Input / Output	Notes
7	REMOTE START SIGNAL	I	<ul style="list-style-type: none"> <li>This is the TTL-4-wire Remote Control Port '<b>START</b>' signal.</li> <li>Connect to '<b>START</b>' output of the Remote System</li> </ul>
9	REMOTE FAIL SIGNAL	O	<ul style="list-style-type: none"> <li>This is the TTL-4-wire Remote Control Port '<b>FAIL</b>' signal.</li> <li>Connect to '<b>FAIL</b>' input on Remote System</li> </ul>
11	REMOTE OK SIGNAL	O	<ul style="list-style-type: none"> <li>This is the TTL-4-wire Remote Control Port '<b>OK</b>' signal</li> <li>Connect to 'OK / PASS' input on Remote System</li> </ul>
13	REMOTE BUSY SIGNAL	I	<ul style="list-style-type: none"> <li>This is the TTL-4-wire Remote Control Port '<b>OK</b>' signal</li> <li>Connect to '<b>BUSY</b>' input on Remote System</li> </ul>
10, 12, 14	GROUND (0V)	P	<ul style="list-style-type: none"> <li>Connect to GROUND on Remote System</li> </ul>

O - Output from programmer

I - Input to programmer

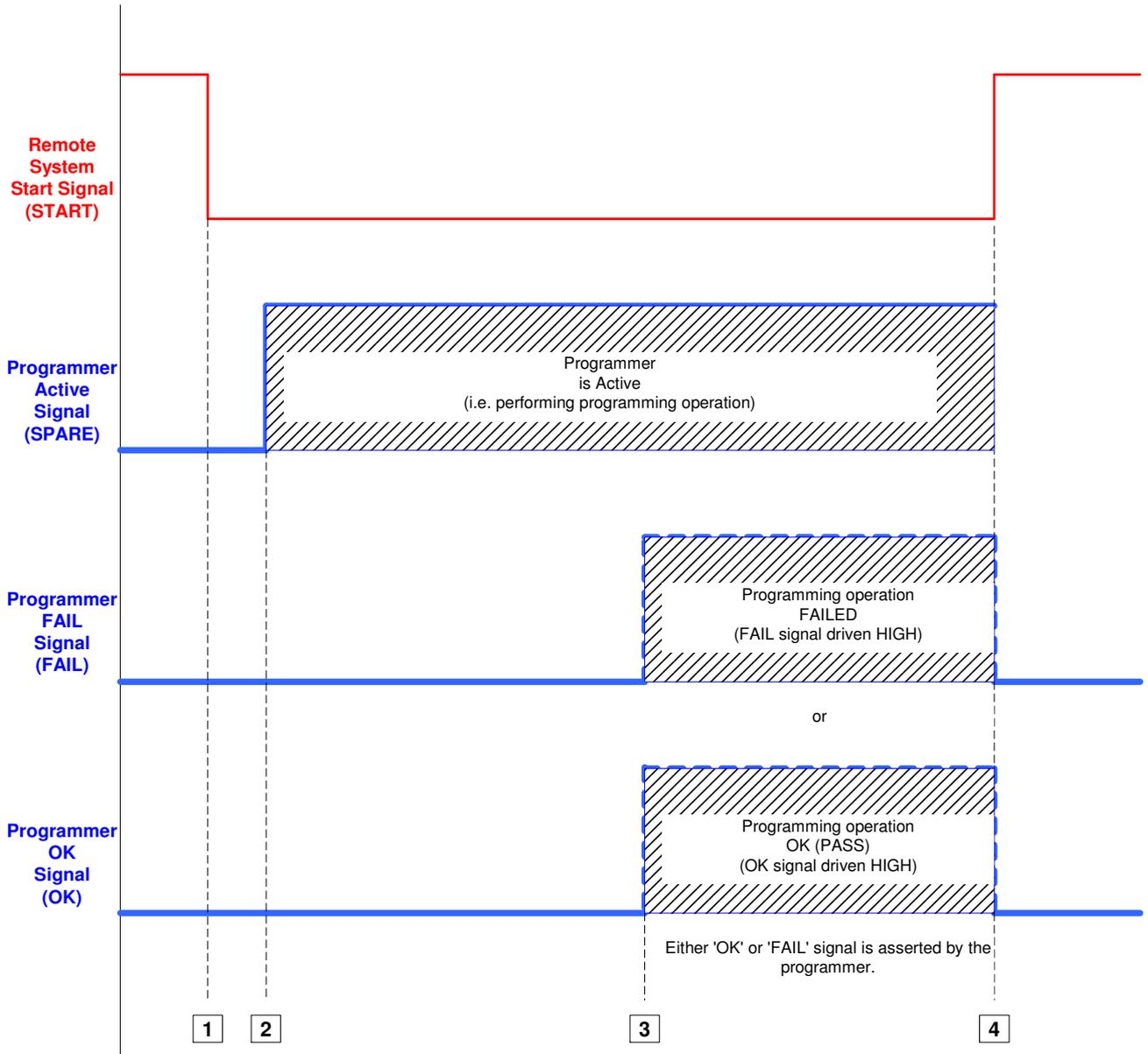
P - Passive e.g. GROUND and power rails

N/C - Not connected

### 2.7.3 Control diagram for Remote I/O 4-wire Control

The control sequence for the '4-wire Control' method is detailed below.

Fig. 2.8.2 Control diagram for 4-wire Control



Key	Programmer / Remote System Action
1	The Remote System asserts the programmer <START> signal to initiate the execution of a 'Programming Project'.
2	The programmer will then assert the <ACTIVE> signal to indicate that it has commenced programming.
3	At the end of the Programming operation the programmer asserts either the <OK> or <FAIL> signal depending on the outcome.
4	The Remote System must then de-assert the <Start> signal to allow the programmer to reset ready to program the next device.

## 2.8 Target System Connect / Disconnect detection

The ISPnano programmer has a special mode where it can automatically detect when a Target System (UUT) is connected to or disconnected from the programmer. A load sensing technique is used to detect the presence of the UUT. This method of UUT detection can be used to trigger a **'Standalone Programming Project'** or even a sequence of chained projects to start.

The typical sequence of events is as follows:

- Programmer waits for UUT to be connected (flashing BUSY LED)
- Operator connects UUT to programmer
- Programmer detects connection of UUT
- Programmer automatically executes a sequence of Standalone Programming Projects
- At the end of the programming sequence, the programmer will FLASH either the PASS or FAIL LED
- Programmer then goes into 'Target Disconnection Sensing' mode (flashing BUSY LED)
- Operator removes the UUT
- Programmer resets itself ready for the next UUT

Please see Application Note – AN113 for further instructions on how to set up your programmer to operate in this mode.

## 2.9 Script Mode (ISP-PRO)

**'Script Mode'** is designed for production users who require logging of all programmer operations to a database. This mode utilises the Equinox ISP-PRO software which allows execution of Programming Scripts. It is possible to write scripts to program unique data such as serial numbers, calibration data etc. It is also possible to control the programmer from a Remote Application written in eg. Labview, Visual Basic, C++ etc.

In order to use 'Script Mode' with your programmer, it is necessary to purchase the relevant license upgrade for ISP-PRO. Please see the Equinox Website for further details.

## 2.10 ConsoleEDS – Command Line control

ConsoleEDS is a powerful software utility which allows any Equinox ISP Programmer to be controlled from a custom Remote Application running under Windows.

The main features of ConsoleEDS are:

- Allows an Equinox ISP Programmer to be remote controlled via simple Command Line commands.
- Suitable for interfacing to any application which executes under Windows
- Simple Command Line interface makes even complex programming operations simple to implement
- Supports writing of a block of data from a file to the FLASH or EEPROM of a Target Chip
- Supports reading of a block of data from the FLASH or EEPROM of the Target Chip to a file on the PC hard disk
- Supports uploading of pre-compiled standalone 'Programming Projects' to a target programmer without requiring EQTools or the Project Upload Wizard.
- Supports running of a specified '**Standalone Programming Project**' which has already been uploaded into the programmer '**FLASH Memory Store**'
- Supports programming of the '**Configuration Fuses**' of the Target Chip
- Supports programming of the '**Security Fuses**' of the Target Chip
- Supports automatic generation of and programming of unique information such as serial numbers and calibration data
- Supports reading / writing of Atmel AVR on-chip Oscillator Calibration Byte

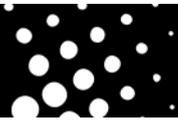
**Please note:**

Many of the above features are only accessible when using '**ConsoleEDS PRO**'.

Typical ConsoleEDS example:

```
ConsoleEDS BaseProject.prj /AUTOPROGRAM=MainFirmware  
/EEPROMWRITE=CalData.hex,0,50
```

For further information about ConsoleEDS, please refer to Application Note 111.



## 3.0 Serial Communications Ports

### 3.1 Overview of Serial Ports

The ISPnano series of programmers can be controlled from any external device which supports RS232, RS-485 or USB connectivity. The “**Series I**” programmers feature 2 x RS232 ports and an RS485 port but do not have a USB port. The “**Series II / III**” programmers feature a single RS232 port, an RS485 port and a USB port.

The table below details the “**Communications Ports**” available on each variant of the ISPnano programmer.

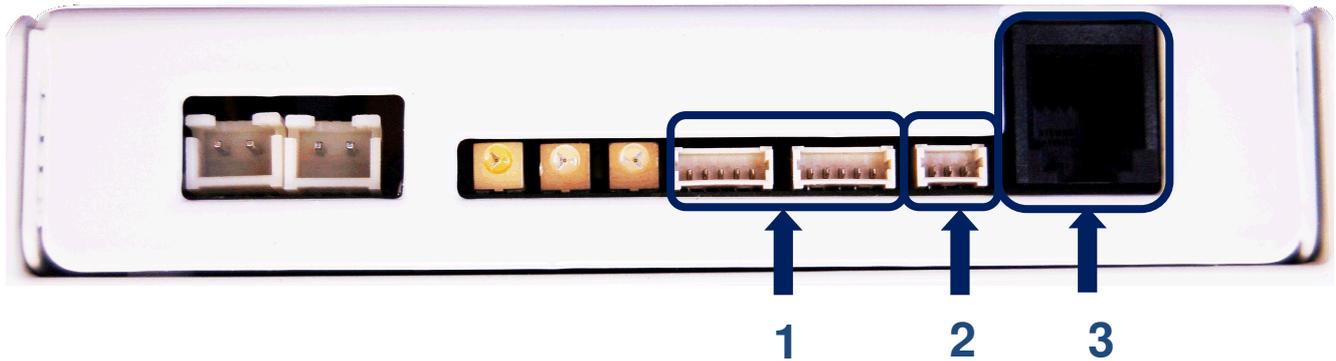
Serial Port	Series I	Series II	Series III	Series III ATE	Series III MUX2 / MUX4 / MUX8
RS232-1 Port	Yes Labelled “RS232-1”	No	No	No	No
RS232 / Powered Remote Keypad + Display Port	Yes Labelled “RS232-2”	Yes Labelled “RS232”	Yes Labelled “RS232”	Yes Labelled “RS232”	Yes Labelled “RS232”
RS485 (In/Out) Ports	Yes	Yes	Yes	Yes	Yes
USB Port	-	Yes	Yes	Yes	Yes

## 3.2 ISPnano Series I – Serial Port connections

### 3.2.1 Overview

The ISPnano Series I programmer features the following Serial Port connections:

- 2 x RS232 Ports (labelled '**RS232-1**' and '**RS232-2**')
- 1 x '**Remote Display / Keyboard**' Port (This is the same physical port as the '**RS232-2**' port.)
- 1 x RS485 Port (labelled '**RS485 IN / OUT**')



Key No	Function	Further description
1	RS-485 Input (& output)	RS-485 - Input/output. Connector: 5-pin 2.5mm Molex
2	RS-232 Communications Port 1	RS-232 - Input/output Connector: 3-pin 2.5mm Molex
3	RS-232 Communications Port 2 or Remote Display / Keypad	RS232 + LCD power. Connector: RJ11

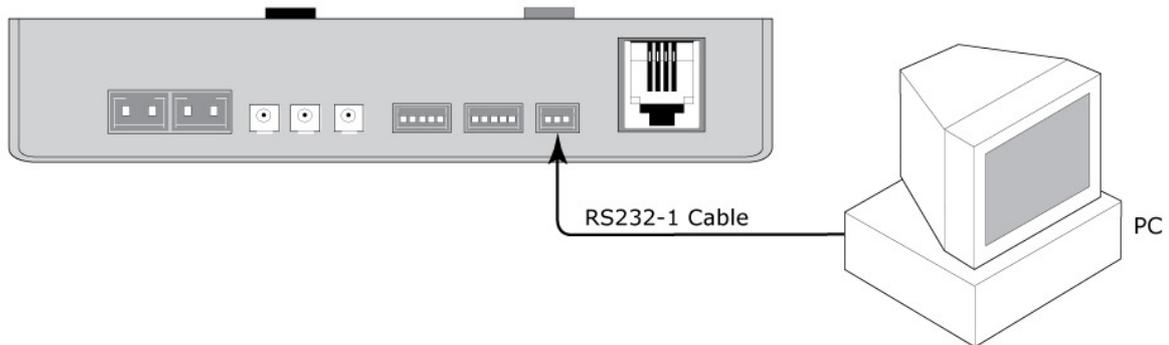
**Please note:**

- The programmer will automatically detect communications on the RS485 or either RS232 port.
- Either of the RS-232 ports can be used to communicate with the programmer.
- The RS232-2 (RJ11) port is the preferred port as the connector is more robust.

Communications Port	Max Nodes	Maximum Baud Rate	Connector	Comment
RS232-1	1	115,200	3-pin 2.5mm Molex	Standard RS232 Port
RS232-2	1	115,200	RJ11	Standard RS232 Port Can also be used as a ' <b>Remote Display / Keypad</b> ' port
RS485 In/Out	32	230,400	5-pin 2.5mm Molex	2 RS485 ports (In + Out) Ports wired in parallel.

### 3.2.2 ISPnano Series I - Connecting to the RS232-1 Serial Port

If the programmer is being used inside a Test Fixture where it is unlikely that the RS232 cable will be regularly plugged then the '**RS232-1 Serial Port**' can be used. This port features a miniature 3-pin Molex connector.



**Please note:**

The '**RS232-1 Serial Port**' is only available on the ISPnano Series I programmer. It has been replaced with a USB Port on the Series II and Series III programmers.

**Instructions**

- Plug the '**3-pin Molex to 9-way D-connector**' into the **RS232-1** port on the programmer.
- Connect the other end of the cable to the '**9-way D-connector Serial Cable**'
- Connect the other end of the '**9-way D-connector Serial Cable**' to a spare COM port on your PC.

*ISPnano – RS232-1 – 3-pin Molex to 9-way female D-connector pin-out*

ISPnano RS232-1 connector	ISPnano signal description	9-way female D-connector pin	9-way female D-connector signal description
1	TX from programmer	2	RX into PC or Remote System
2	RX to programmer	3	TX from PC or Remote System
3	0V (GROUND)	5	0V (GROUND)

**Please note:**

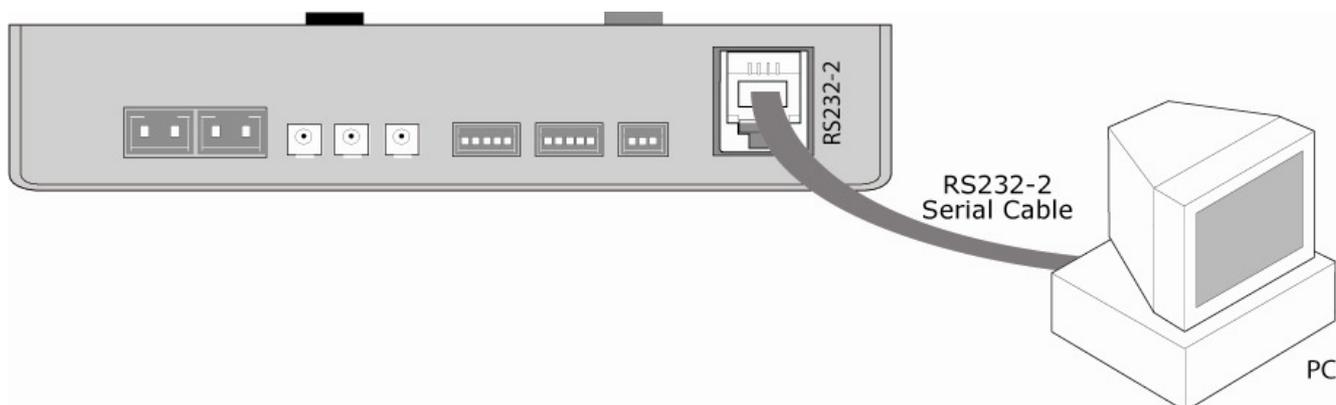
- Pin 1 of the 3-pin Molex connector is the left-hand pin.

### 3.2.3 ISPnano Series I - connecting to the RS232 Serial Port (RJ11)

If the programmer is being used in an environment where the RS232 cable will be regularly plugged in and unplugged, then it is safer to use the **'RS232-2 Serial Port'**. This port features a robust RJ11 connector which can withstand many insertions and de-insertions. The programmer comes with an **'RJ11 to 9-way D-connector'** Serial Cable and also a **'9-way serial extension cable'**. The combination of these two cables is used to connect the programmer to the PC COM port.

#### Instructions

- Plug the **'RJ11 to 9-way D-connector'** into the RJ11 connector port (**RS232-2**) on the programmer (see diagram below).
- Connect the other end of the cable to the **'9-way D-connector Serial Cable'**
- Connect the other end of the **'9-way D-connector Serial Cable'** to a spare COM port on your PC.



ISPnano – RS232-2 – RJ11 to 9-way female D-connector pin-out

ISPnano RJ11 connector	ISPnano signal description	9-way female D-connector pin	9-way female D-connector signal description
1	Switched +5.2V from programmer	No connect	No connect
2	TX from programmer	2	RX into PC or Remote System
3	RX to programmer	3	TX from PC or Remote System
4	0V (GROUND)	5	0V (GROUND)

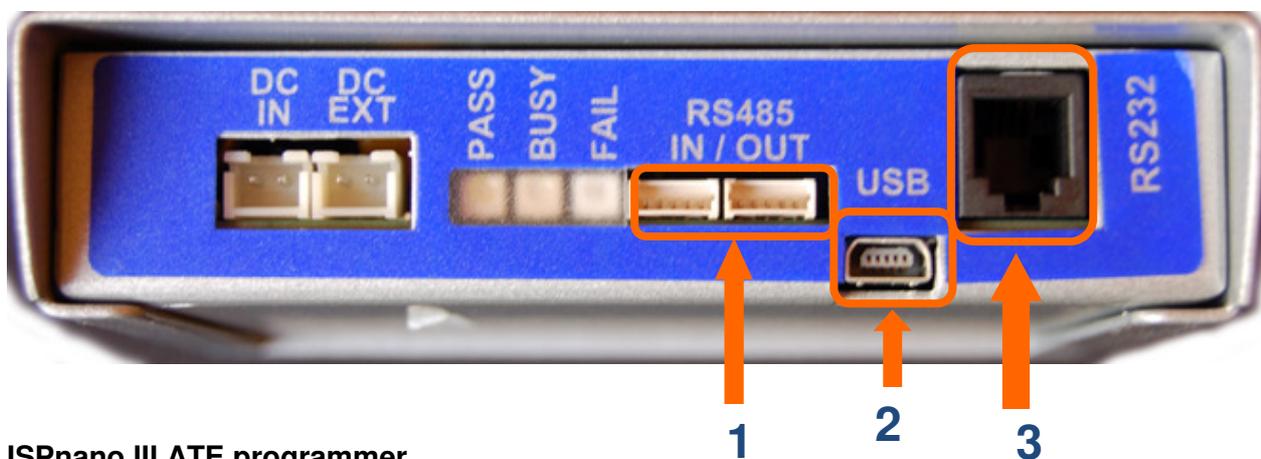
### 3.3 ISPnano Series II + III (ATE) + MUX – Serial Port connections

#### 3.3.1 Overview

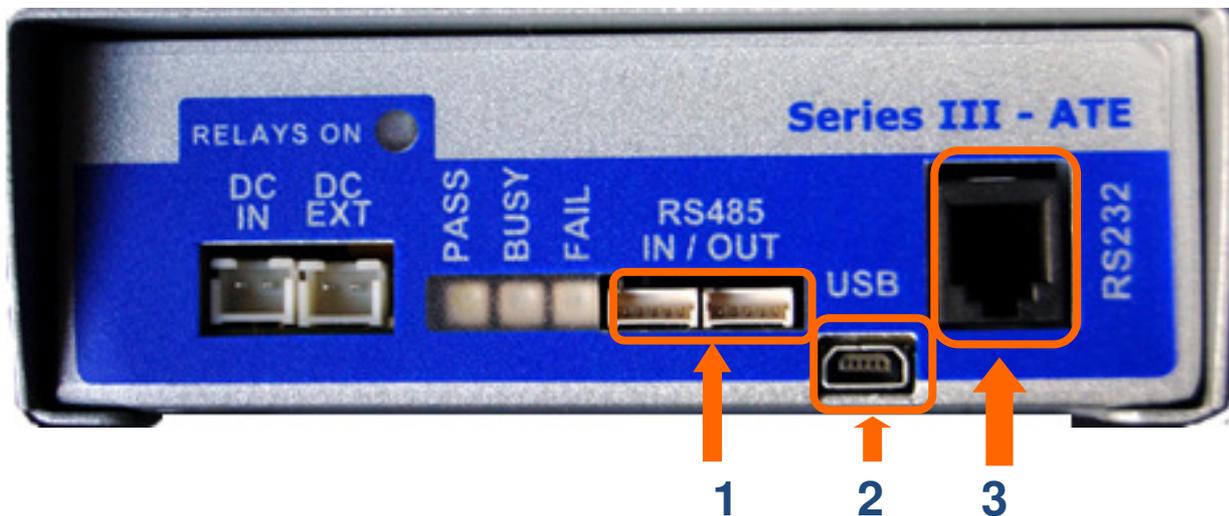
The ISPnano Series II, Series III, Series III(ATE) and ISPnano-MUX programmers all features the following Serial Port connections:

- 1 x RS485 Port (labelled '**RS485 IN / OUT**')
- 1 x High-speed USB Port
- 1 x RS232 Port (labelled '**RS232**')
- 1 x '**Remote Display / Keyboard**' Port (This is the same physical port as the '**RS232**' port.)

#### ISPnano Series II or III programmer



#### ISPnano III ATE programmer



Key No	Function	Further description
1	RS-485 Input (& output)	RS-485 - Input/output. Connector: 5-pin 2.5mm Molex
2	USB Port	USB - Input/output Connector: miniUSB connector
3	RS-232 Communications Port or Remote Display / Keypad	RS232 + LCD +5V power. Connector: RJ11

### 3.3.2 ISPnano Series 3 - Comparison of communications modes

The table below provides a comparison between the available communications modes.

Communications Port	Max Nodes	Maximum Baud Rate	Connector	Comment
RS232	1	115,200	RJ11	Standard RS232 Port Can also be used as a ' <b>Remote Display / Keypad</b> ' port
RS485 In/Out	32	230,400	5-pin 2.5mm Molex	2 x RS485 ports (In + Out) Ports wired in parallel.
USB	1 ?	230,400	Mini-USB connector	High-speed USB Port

**Please note:**

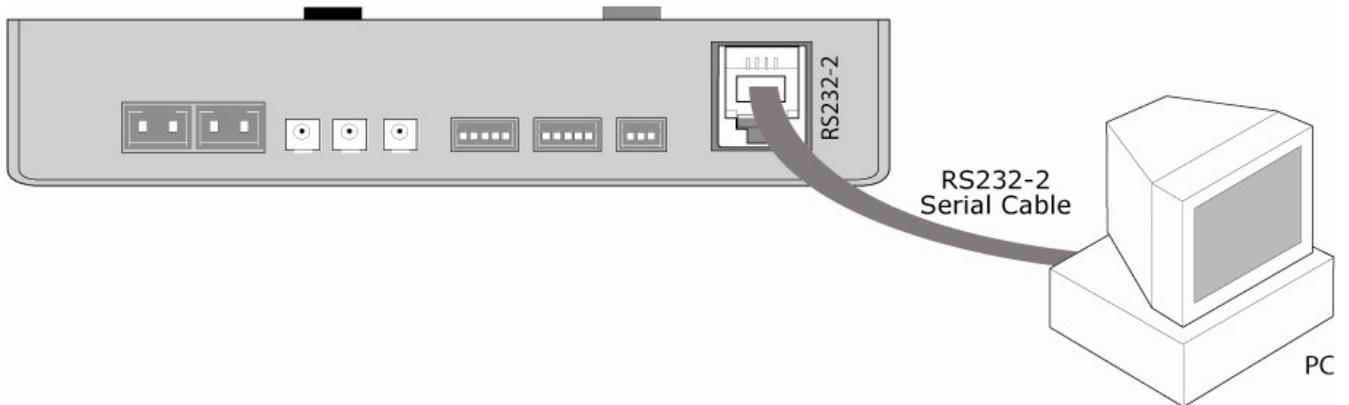
- The programmer will automatically detect communications on the RS232, USB or RS485 ports.
- The RS232 (RJ11) port is a dual purpose port. It can be used as a standard RS232 port to control the programmer from the PC. It can also be used with a '**Remote Display / Keypad Module**' which can then control the programmer. This port also features a regulated +5V supply to power the '**Remote Display / Keypad Module**'
- The '**ASCII Text communications Protocol (ATC)**' is only supported using the '**RS232**' port. It will not work with a '**Virtual COM Port**' over USB.

### 3.3.3 ISPnano Series I / II / III - connecting to the RS232 Serial Port (RJ11)

The '**RS232 Serial Port**' can be used to control the programmer from either a PC RS232 (COM) port or some other RS232 enabled control device. The RS232 port features a robust RJ11 connector which can withstand many insertions and de-insertions.

**Instructions**

- Plug the '**RJ11 to 9-way D-connector**' into the RJ11 connector port (**RS232-2**) on the programmer (see diagram below).
- Connect the other end of the cable to the '**9-way D-connector Serial Cable**'
- Connect the other end of the '**9-way D-connector Serial Cable**' to a spare COM port on your PC.



The programmer comes with an '**RJ11 to 9-way D-connector**' Serial Cable and also a '**9-way serial extension cable**'. The combination of these two cables is used to connect the programmer to the PC COM port.

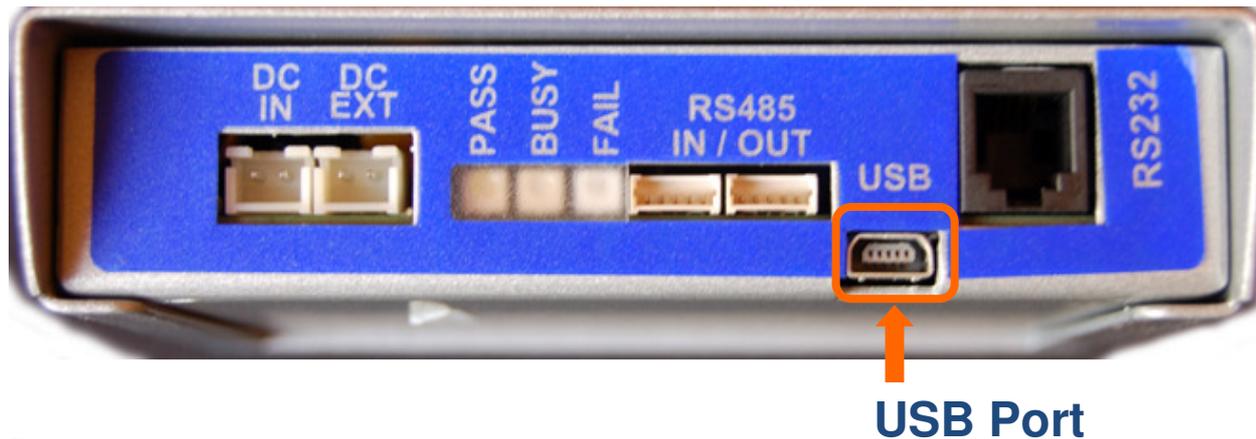
*ISP nano – RS232 – RJ11 to 9-way female D-connector pin-out*

ISP nano RJ11 connector	ISP nano signal description	9-way female D-connector pin	9-way female D-connector signal description
1	Switched +5.2V from programmer	No connect	No connect
2	TX from programmer	2	RX into PC or Remote System
3	RX to programmer	3	TX from PC or Remote System
4	0V (GROUND)	5	0V (GROUND)

## 3.4 USB Port

### 3.4.1 Overview of USB connection

It is possible to connect a single '*ISPnano Series III*' programmer to a controlling PC using the high-speed USB interface. The USB interface provides a high-speed method of uploading programming projects to the programmer. It works by installing a '*Virtual COM Port (VCP)*' on the host PC which our EQTools / ISP-PRO / ConsoleEDS utilities then communicate through.



Please note:

- The USB connector on the programmer is a '*mini-USB*' connector.
- It is **NOT** possible to power the programmer via the USB port.

### 3.4.2 USB Driver - Installation Instructions

1. Make sure the USB cable is **NOT** connected between the PC and programmer
2. Install the '*USB Driver*' – see Application Note AN126 for further instructions.  
The '*USB Driver*' can be found as follows....

#### i. On the CD-ROM which came with the programmer

The driver should be in the following sub-folder on the CD:

`\USBDrivers\PPM4-MK1_ISPnano_SeriesII`

#### ii. On the Equinox website

The latest USB Driver can always be found on the Equinox Website. This version may be more up-to-date than the one on the CD-ROM so it is usually a good idea to check the website version first.

To download the USB Driver, please browse to the following web page:

<http://www.equinox-tech.com/products/downloads.asp?details=286>

\*\*\* Make sure the USB Driver is installed before going to step (3) \*\*\*

3. Connect the supplied USB cable between the USB Port on the rear of the programmer and a spare USB port either on the PC or a USB Powered Hub.

→ The PC should display a message “**Found new hardware.....**”

- Follow the on-screen instructions to install the USB Driver.
- Please refer to Application Note AN126 for further instructions.

### **3.4.3 Selecting the USB – Virtual COM Port (VCP) in EQTools**

- Make sure the USB cable is connected between the programmer and PC
- Power up the programmer
- In EQTools, select **<Programmer> <Communications and scripting settings>**
- Select the COM port which corresponds to the VCP which the programmer is connected to.
- Click **<Test>** to test if the PC can communicate to the programmer via USB

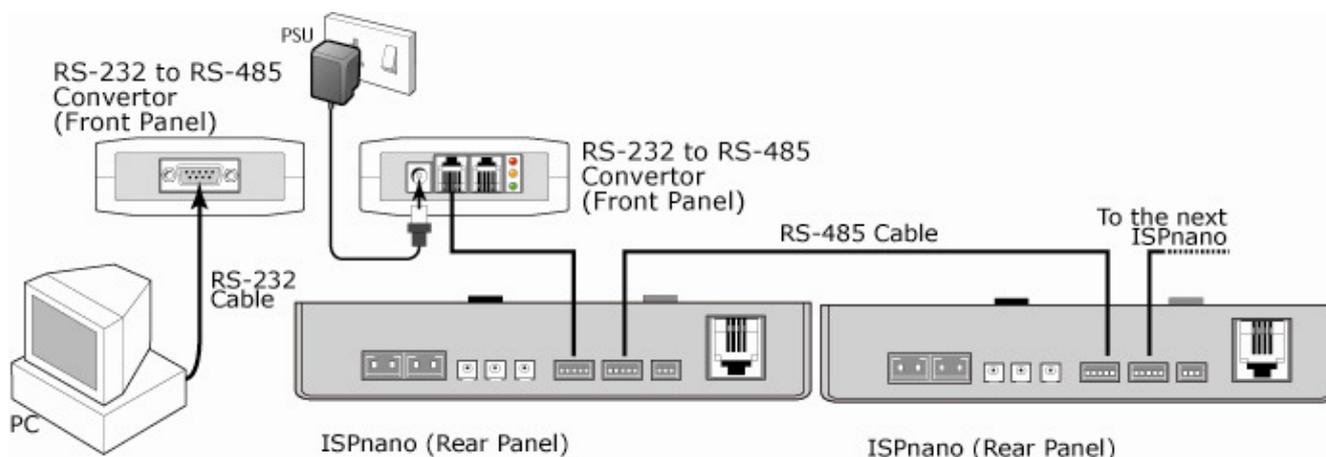
Please refer to Application Note AN126 for further instructions on setting up and testing the USB ‘**Virtual COM Port**’.

## 3.5 RS485 Port

### 3.5.1 Overview

It is possible to connect up to 32 x ISPnano programmers (Series I, II or III) to a single PC COM port using the '**RS485 Network**'. The programmers are daisy-chained together on an RS485 network using special '**5-pin Molex RS485 Cables**' which connect to the '**RS485 In / Out**' connectors on the ISPnano programmer.

The diagram below shows a typical RS485 programmer network with 2 x ISPnano programmers daisy-chained together.



The RS485 converter should be placed as close to the PC as possible so as to minimise the length of the RS232 cable. The cable from the RS485 Converter to the first ISPnano programmer must be terminated using the '**RS485 FCC68 to 5-pin Molex Converter Module**' as the ISPnano uses a smaller 5-pin Molex connector.

### 3.5.2 Equipment required

To operate the programmers in multi-channel RS485 mode, you will need the following equipment:

- 1 x PC with spare 9-way COM Port
- 1 x Equinox '**RS232 to RS485 Converter**' (Order code: RS485-C2)
- 1 x '**RS485 FCC68 to 5-pin Molex Converter Module**' (Converts RJ11 to 5-pin Molex ISPnano RS485 connector)
- 1 x RJ11 RS485 Flat Cable (pin-out crossed)
- n x '**5-pin Molex RS485 Cable**' (for connecting between ISPnano programmers – where n is the number of programmers)

### 3.5.3 Instructions

To set up a multi-channel RS485 programmer network, please follow the instructions detailed below...

- Place the '**RS485 Converter**' as close to the PC COM port as possible.
- Connect the 9-way RS232 cable between the '**RS485 Converter**' RS232 port and a spare PC COM Port.
- Connect an '**RS485 Cable**' flat cable (must be an '**crossed cable**') from either of the '**RS485 Connectors**' on the Converter to the '**RS485 FCC68 to 5-pin Molex Converter Module**'
- Connect a '**5-pin Molex RS485 Cable**' to the '**RS485 In**' connector on the first programmer in the series.
- Connect a '**5-pin Molex RS485 Cable**' from the '**RS485 Out**' connector on the first programmer to the '**RS485 In**' connector on the second programmer in the chain.
- Connect a '**5-pin Molex RS485 Cable**' from the '**RS485 Out**' connector of the second programmer to the connector of the third programmer.....and so on until you reach the last programmer in the chain.

#### Important notes:

- The RS232 cable between the '**RS485 Converter**' and the PC COM Port should be kept as short as possible i.e. no longer than 2m.
- The RS485 Converter should be placed as close to the PC COM port as possible.
- An RS485 terminator is NOT required on the last programmer.
- The '**RS485 In**' or '**RS485 Out**' connectors are connected in parallel inside the programmer so it does not matter whether the '**In**' or '**Out**' connector is used.
- Each programmer on the RS485 network must have a unique '**Node Address**'. The address is set by connecting to either RS232 port and then using the '**Set Node Address**' utility in EQTools.
- This mode of operation requires either the ISP-PRO or ConsoleEDS software utilities to be running on the PC

### 3.5.4 RS485 FCC68 to 5-pin Molex Converter Module

The cable from the RS485 Converter to the first ISPnano programmer '**RS485 In**' connector must be terminated using the '**RS485 FCC68 to 5-pin Molex Converter Module**' as the ISPnano uses a smaller 5-pin Molex connector. The '**Converter Module**' is a small circuit board which converts the RJ11 connector from the RS485 Converter to the miniature 5-pin Molex connector used for RS485 connections on the ISPnano programmer. The flat cable from the RS485 Converter plugs in one end. A '**5-pin Molex RS485 Cable**' plugs in the other end and then connects to the first ISPnano programmer in the chain.

## 3.6 Detecting attached programmers

### 3.6.1 Overview

An ISPnano programmer can communicate to a controlling PC via RS232, RS485 or USB. It is factory configured to have the '**Communications Node Address**' set to '**0**'. If you know that the attached ISPnano programmer is configured as '**Address 0**', then you simply need to make sure that the '**Communications Address**' in EQTools is set to '**0**' and then EQTools should be able to communicate with the programmer. This is the standard setting for EQTools and an ISPnano programmer.

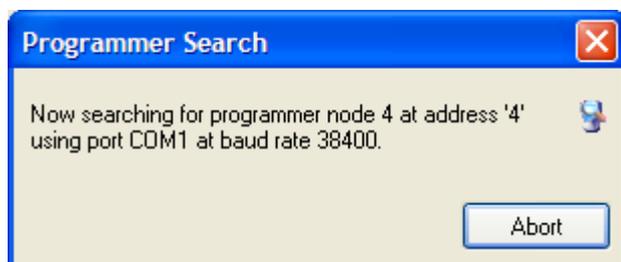
If the attached programmer has been configured with an address which is not '**0**' or you have multiple programmers attached on the RS485 bus, then the simplest way to find out which programmers are attached is to use the '**Detect Programmer(s)**' utility in EQTools.

### 3.6.2 Using the '**Detect Programmer(s)**' utility

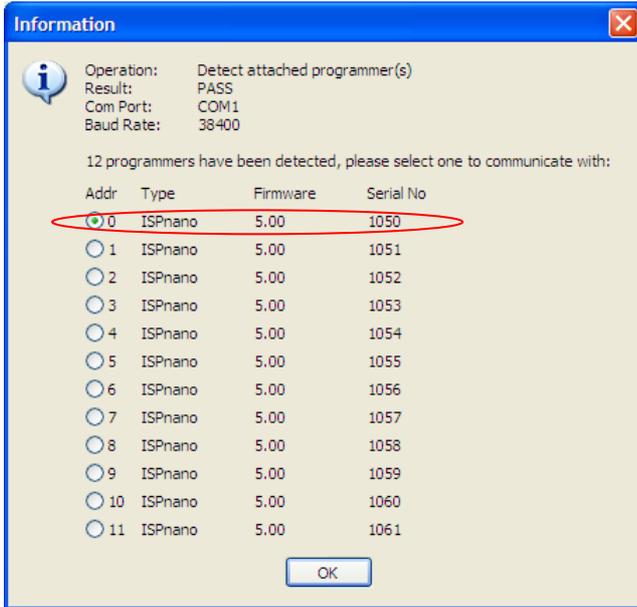
The instructions below detail how to detect one or more programmers which are connected to the PC.

#### Instructions

- Launch EQTools
- Select **<Programmer> <Detect Programmer(s)>**
- EQTools will now scan addresses 0 to 31 for attached programmers.



- A list of the attached programmers which have been detected will then be displayed. In the example below 12 x ISPnano programmers have been detected with '**Node Addresses**' ranging from 0 to 11.



- To set EQTools to communicate with a particular programmer, simply click the radio button next to the desired programmer in the list and then click **<OK>**
- EQTools will now be set up to communicate with the attached programmer at the specified address.

### 3.6.3 Resolving Programmer Detection issues

If no programmers are detected, please check the following:

- The serial cable is connected between the PC COM port and the programmer or the RS485 Converter.
- All programmers are powered up.
- All programmers have a unique '**Node Address**'.
- The '**RS485 Converter**' (if used) is powered on.

If multiple programmers are detected but some are not detected, please check the following:

- The RS485 cable is connected between the last detected programmer and the first non-detected programmer.
- Try swapping the RS485 cable for a known working cable just in case the cable is faulty.
- Make sure each programmer has a unique '**Node Address**'. If more than one programmer has the same address then this may cause 'Communications Errors'.

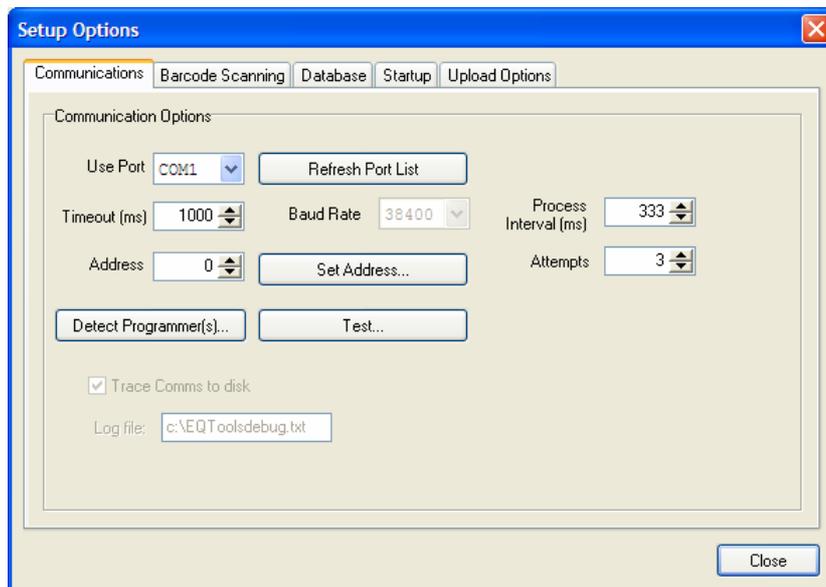
### 3.7 Setting the programmer 'Communications Node Address'

It is possible to have up to 32 x ISPnano programmers networked together on an RS485 bus. Each programmer must have a unique '**Communications Node Address**' so that the PC knows which programmer it is communicating with. This address must be in the range of 0 to 31. The address is set by default to address '0' (zero).

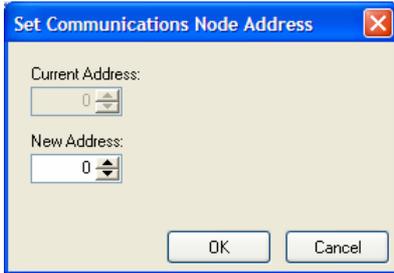
On the ISPnano programmer, the '**Communications Node Address**' is stored in a location in EEPROM and is therefore permanently remembered by the programmer even if the programmer is power cycled. It is possible to set the '**Communications Node Address**' to any address between 0 and 31 using the '**Programmer and Scripting Setup**' screen in EQTools. Setting of the address can only be done when connected to the programmer via RS232.

To set '**Communications Node Address**' :

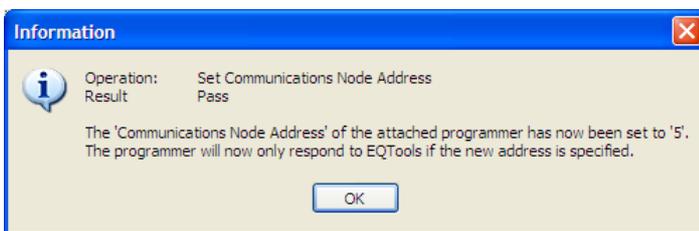
- Select either the '**RS232-1**' or '**RS232-2**' serial ports on the programmer.
- Connect a serial cable from either of these serial ports to a spare PC COM port on your PC.
- Launch EQTools
- Select **<Programmer> <Programmer and Scripting Setup>**
- The following screen is displayed...



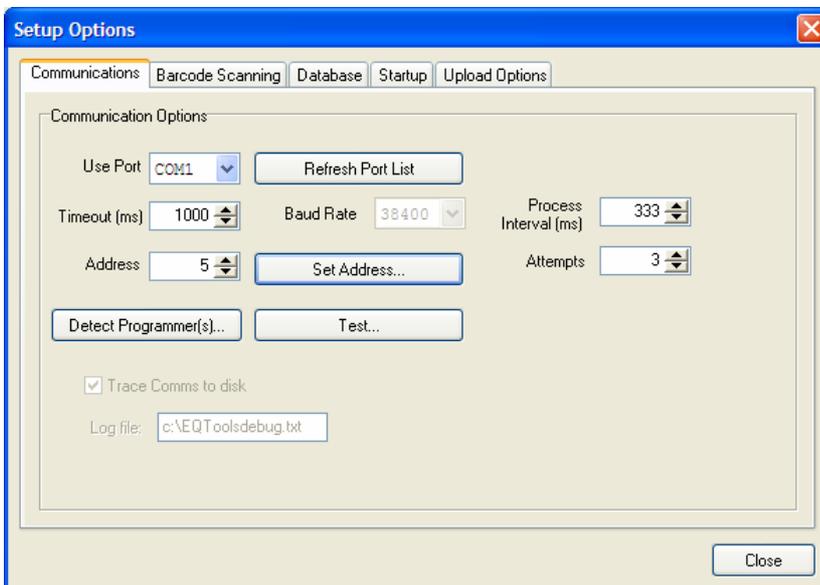
- Select either the COM port to which the programmer is connected e.g. COM1
- Click the **<Set address>** button
- → The following screen will be displayed:



- Change the 'New Address' field to be the address you want to give the attached programmer e.g. address 5
  - Click **<OK>**
- The programmer should now confirm that the correct address has been stored.



- Click **<OK>**
- → The 'Address' in EQTools is now also updated to your new address e.g. address 5 – see screenshot below.



- Click **<Close>**
- The attached programmer will now only respond to EQTools if the '**Communications Address**' in EQTools matches the '**Node address**' you stored in the programmer.

**Please note:**

- The '**Communications Node Address**' can only be set by connecting to the programmer using either of the RS232 ports. It cannot be set when connected to the programmer via the RS485 port.

- The default address (factory setting) is '0' (zero)
- EQTools can only communicate with the attached programmer if the '**Communications Address**' is set to the same address in EQTools.

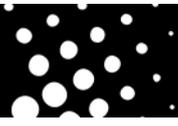
### 3.8 Connecting the Remote Display / Keypad Module

It is possible to control the programmer via a '**Remote Display / Keypad Module**'. This module plugs into the **RS232-2** port on the programmer. An RJ11 cable wired 1-1 is used to connect the ISPnano to the '**Remote Display / Keypad Module**'. The **RS232-2** port features both the RS232 signals and a switched +5.2V supply to power the remote display.

The table below details the connections between the ISPnano and the '**Remote Display / Keypad Module**'.

Fig. 3.5 ISPnano – RS232 to Remote Display / Keypad Module pin-out

ISPnano RJ11 connector pin	ISPnano signal description	Remote Display / keypad pin	Remote Display / keypad signal description
1	Switched +5.2V from programmer	1	Switched +5.2V from programmer
2	TX from programmer	2	RX into Display / Keypad module
3	RX to programmer	3	TX from Display / Keypad module
4	0V (GROUND)	4	0V (GROUND)



## 4.0 Programmer / Target System Power Supply Scenarios

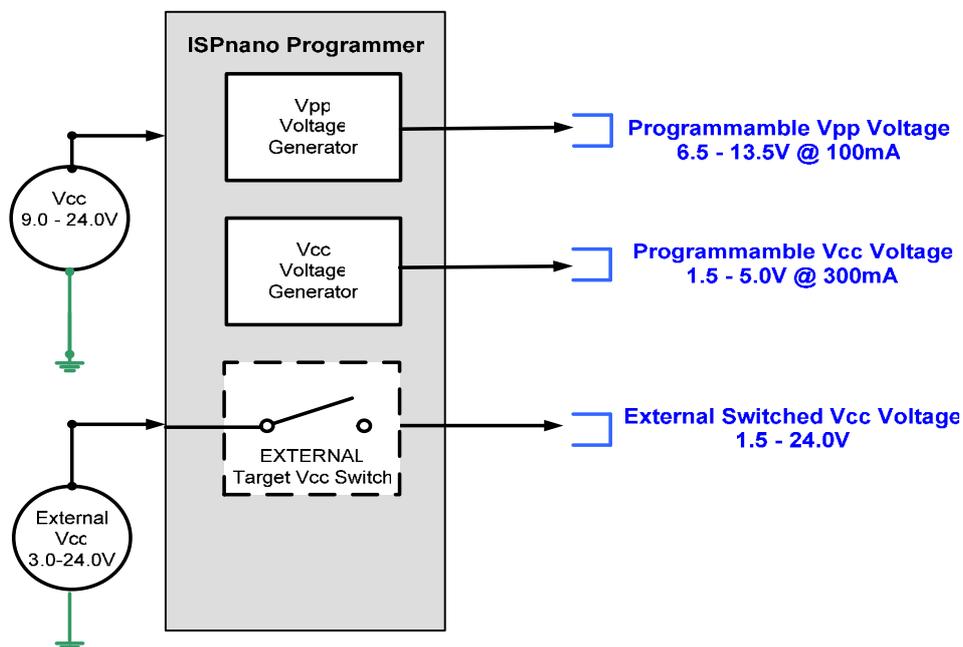
### 4.1 Overview of programmer Power Supplies

The ISPnano programming module family has been specially developed for use within a Programming Fixture or Test Fixture. The ISPnano programmer is not just a programmer. It also features two very accurate user-programmable power supplies which can be used to power the Target System (UUT).

The ISPnano programmer family features the following internal power supplies....

#	Power Supply	Voltage range (V DC)	Voltage accuracy	Max current (mA)	Current monitor / limit	ISPnano Output Pin
1	Programmer Controlled Target Vcc (TVCC) Power Supply	1.8 – 5.0	+/- 1%	480	Yes 5% accuracy	TVCC
2	Programmer Controlled Target Vpp (TVPP) Power Supply	6.5 – 13.5	+/- 3%	100	No	TVPP
3	Programmer controlled EXTERNAL VCC Switch	1.2 – 30.0	N/A	1000	No	EXT-VCC

The illustration below shows how these power supplies are implemented....



## 4.2 Typical applications of each power supply

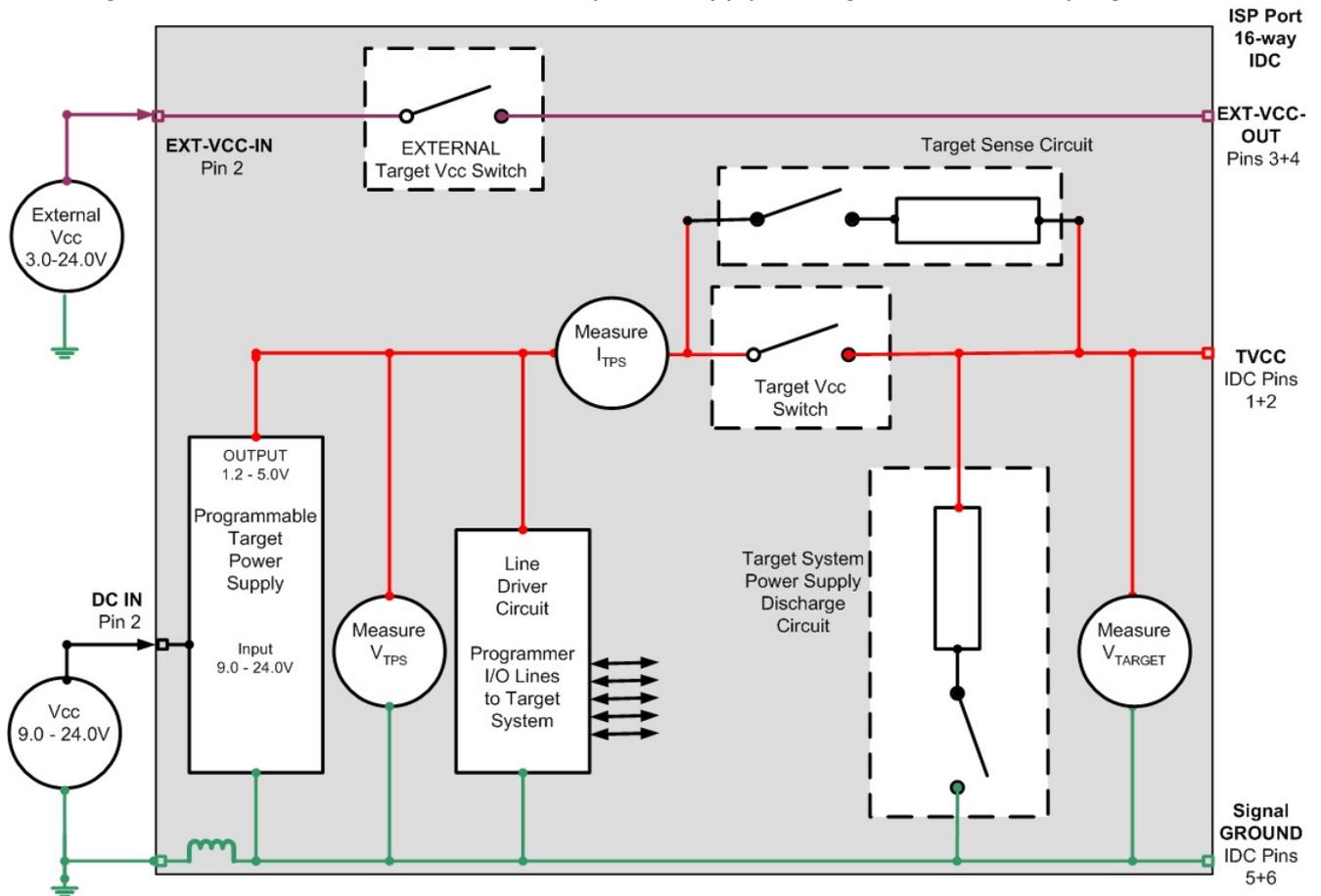
The table below gives an overview of the typical applications of the different ISPnano programmer power supplies.....

#	Power Supply	Voltage range (V DC)	Features / typical application of power supply
1	<b>Programmer Controlled Target Vcc (TVCC) Power Supply</b>	1.8 – 5.0	<ul style="list-style-type: none"> <li>Used to power the internal <b>'Programmer Line Driver Circuitry'</b> which sets the signal levels for the <b>'Programmer Target I/O'</b> pins.</li> <li>Used to apply controlled power to any Target System (UUT) which can be powered at 1.8 to 5.0V.</li> <li>Can also be used to power up the UUT at a specified voltage for target testing and / or voltage calibration purposes.</li> <li>This power supply also features current / over-current monitoring.</li> </ul>
2	<b>Programmer Controlled Target Vpp (TVPP) Power Supply</b>	6.5 – 13.5	<ul style="list-style-type: none"> <li>This power supply should only be used for devices which require a <b>'Vpp'</b> programming voltage to force the device into <b>'programming mode'</b>.</li> <li>The voltage is defined by the programming algorithm and is not user-definable.</li> </ul>
3	<b>Programmer controlled EXTERNAL VCC Switch</b>	1.2 – 30.0	<ul style="list-style-type: none"> <li>Allows an <b>'External power supply'</b> to be switched under programmer control to a Target System (UUT)</li> <li>Typically used to power a UUT which requires an input voltage of &gt;5.0V</li> <li>Can also be used if the <b>'Target I/O Signal Voltage'</b> needs to be different to the actual <b>'Target Voltage'</b> e.g. an IC is powered at 3.3V but the main UUT circuitry must be powered at e.g. 5.0V.</li> <li>Used to power the relay coils at +12V on the <b>'ISPnano Series 3 ATE'</b> programmer</li> </ul>

The following sections describe the various power supplies in more detail.

### 4.3 ISPnano - Power Supply schematic diagram

The diagram below shows an overview of the power supply routing of the ISPnano programmer.



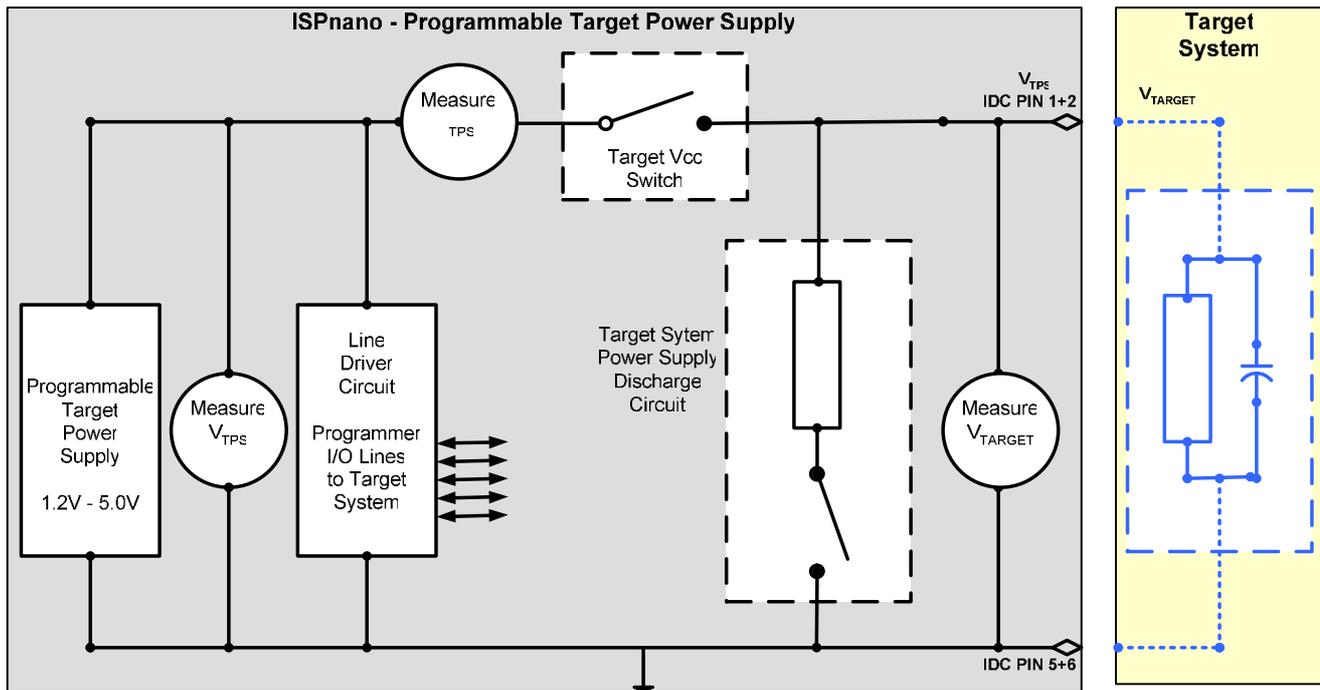
The function of each signal is described in the table below.

Signal name	Function	Voltage range (V DC)	Connector	Description
DC IN	DC Power Input	9.0 – 24.0V	DC-IN – pin 2	Power supply input for programmer
EXT-VCC-IN	External Switched DC Power Input	1.8 – 24.0V	EXT-VCC-IN – pin 2	<b>'EXTERNAL VCC'</b> power supply input
EXT-VCC-OUT	Switched output of <b>'External VCC'</b> supply	1.8 – 24.0V	<b>'Target ISP Port'</b> - pins 3+4	Switched output of <b>'EXTERNAL VCC'</b> supply
TVCC	Target Vcc Supply / measurement	1.2 – 5.0 V	<b>'Target ISP Port'</b> - pins 1+2	Powers Target System or Measures Target Vcc
SIGNAL_GROUND	0V Signal GROUND	0V	<b>'Target ISP Port'</b> - pins 5+6	Signal GROUND between programmer and Target System.

The **'Target ISP Port'** is the 16-way IDC connector on the ISPnano programmer – see appendix 2.

## 4.4 Programmer Controlled 'Target Vcc (TVCC) Power Supply'

The '**Programmer controlled Target Vcc Power Supply**' is a very accurate user-programmable power supply which has been designed to be able to provide controlled power to most Target Systems (UUTs) which require a voltage between 1.8 and 5.0V. This power supply also features '**target current monitoring**' so a current-limit can be set up to automatically shut down the power supply in the event of a short-circuit.



### Power supply features:

- User programmable power supply – user can set up both Voltage and Current settings
- The power supply can switch 1.2 to 5.0V @ 480mA to the Target System (UUT)
- The power supply also generates the '**Programmer Line Driver Voltage**' which sets the Target I/O signal levels.
- The voltage is output on the '**TVCC**' pin of the programmer
- Voltage accuracy: +/- 1% using factory calibrated voltages
- Voltage accuracy: +/- 20 mV is possible at user-defined '**spot calibrated**' voltages
- Fast voltage slew-rate on switch-on
- Power supply '**Voltage measurement**' and '**Target Voltage validation**' (checks that target voltage is within pre-set limits)
- User-definable '**voltage stabilisation**' settings before / after target switch on
- Target Voltage can also be measured if the programmer is not supplying power
- Power supply by source up to 2 Amps continuously (before going into current limit) allowing it to charge up the '**target system capacitance**' at high speed
- Target current monitoring with over-current detection at switch on
- User-definable current stabilisation settings before / after target switch on

- Current measurement accuracy +/- 6%

## 4.5 Programmer controlled 'Target Vpp (TVPP) Power Supply'

The '*Programmer controlled Target Vpp Power Supply*' is designed to apply a so-called '*Programming voltage*' or '*Vpp voltage*' to a Target IC in order to force Target IC into programming mode. The ICs which require this '*Vpp voltage*' do not require much current, so this power supply is only rated at 100mA maximum current.

### Power supply features:

- The **Vpp** voltage is used to force certain ICs into 'programming mode'
- The **Vpp** voltage is automatically set up by the selected device algorithm (not user-adjustable)
- The programmer can switch 6.5 to 13.5 V @ 100 mA to the Target System (UUT)
- Voltage accuracy: +/- 3% using factory calibrated voltages

### Device algorithms which require this Vpp voltage include:

- Atmel - ATtiny TPI interface – High-Voltage Mode (+12V Vpp)
- Atmel – ATtiny 4-wire interface – High-Voltage Mode (+12V Vpp)
- TI – MSP430 – JTAG mode – Vpp required to blow the 'Security fuse' (+13.0V Vpp)

### Warning!

**As the programmer is capable of applying a voltage higher than the normal 'Target Vcc' voltage, this higher voltage could cause damage to the Target Board circuitry. Please ensure that the Target Board has been designed to withstand the VPP voltage being applied during programming.**

## 4.6 External VCC – programmer controlled switch

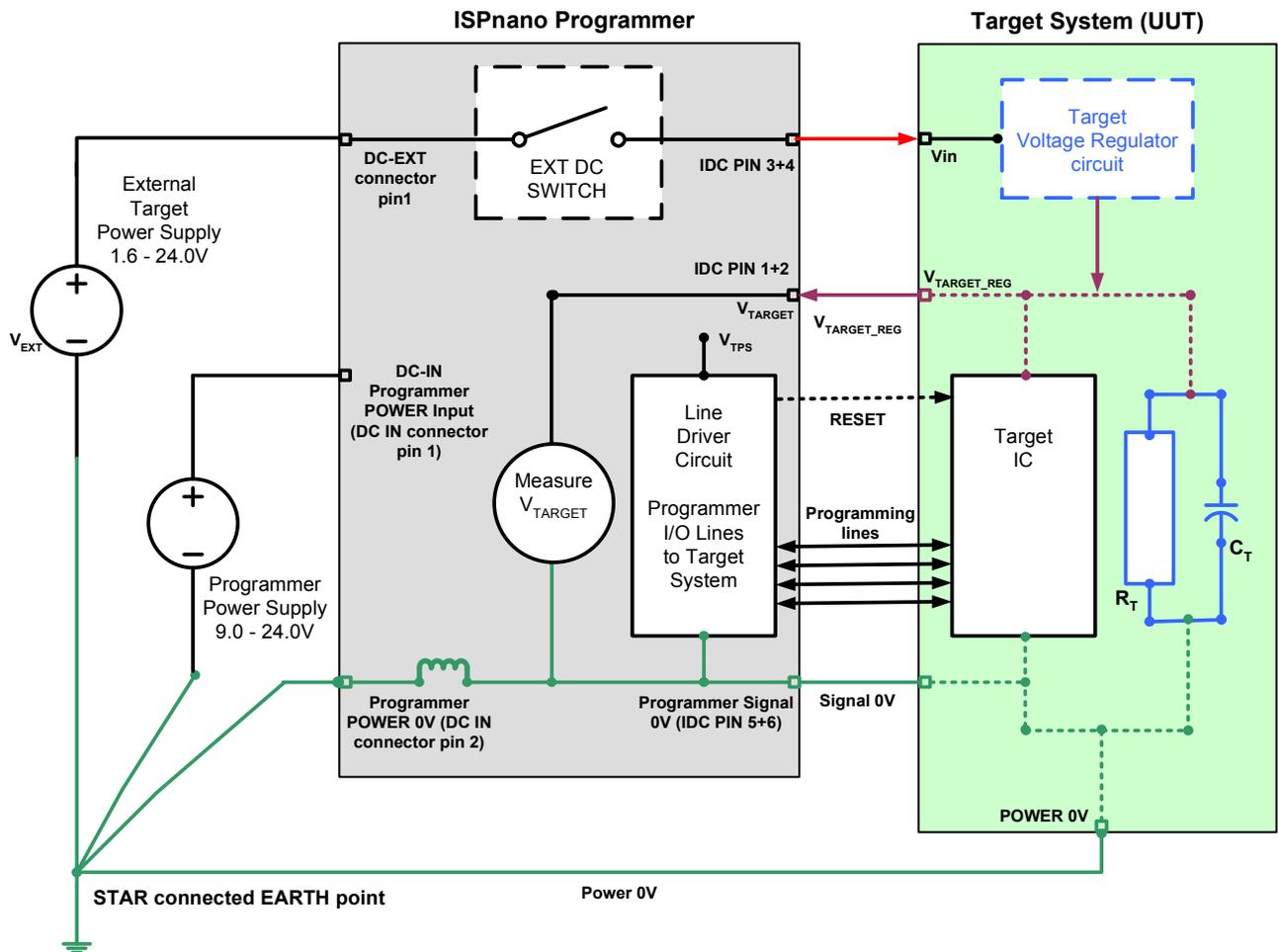
The ISPnano programmer features a special '**External VCC Switch**' circuit which allows an '**External Voltage (V\_EXT)**' in the range of 1.6V – 24V to be switched through the programmer to the Target System under project control. This feature is useful when the Target System features a '**Voltage Regulator Circuit**' which needs to be powered from a higher voltage than +5V. In order to power many Target Systems, it is necessary to apply a voltage of eg. 7 – 24V to the input of the Target System regulator circuit.

### Power supply features:

- Allows an '**External power supply**' to be switched under programmer control to a Target System (UUT)
- Any voltage between 1.2 and 30V can be switched via the '**External VCC Switch**' circuit.
- The '**External VCC Switch**' circuit is rated at 1 Amp.
- Internal self-resetting fuse for protection against short-circuit conditions
- Both the input and output '**External Voltage (V\_EXT)**' can be measured by the programmer

The diagram below shows a typical example where an external power supply is switched to the Target System (UUT) through the programmer.

**Fig. 4.6 ISPnano – Powering Target System using External Switched Voltage**



### Typical scenario:

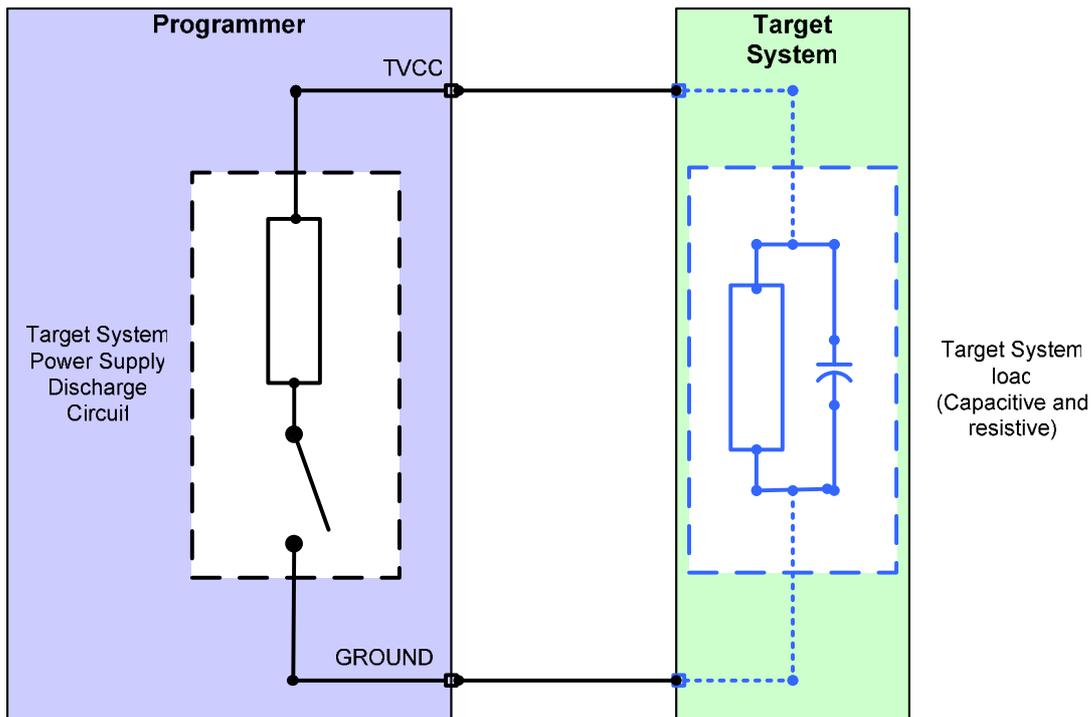
- An '**External**' voltage '**V<sub>EXT</sub>**' in the range of 1.6 – 24.0V is fed into the **EXT-VCC-IN** connector of the programmer from an external power supply.
- This voltage is then switched by the '**External Vcc Switch**' inside the programmer to the **EXT-VCC-OUT** pins on the '**Target ISP Port**' of the programmer.

### Typical applications:

- Used to control power to a Target System (UUT) which requires an input voltage of > 5.0V. This is typical of many designs where the power supply / regulator circuit on a Target System may require e.g. 9 – 24V which is then regulated down to eg. 3.3V or 5.0V
- Used on the '**ISPnano Series III (ATE)**' programmer to switch +12V power to the relay coils

## 4.7 Programmer controlled Target Discharge Circuit

The ISPnano programmer features a programmer controlled '**Target Discharge Circuit**' which can be used to automatically discharge any residual charge stored on the Target System (UUT) after the target power has been switched off. This feature can be fundamental in preventing ESD damage to both the UUT and programmer if a UUT is removed from a Programming Fixture while it still has residual charge on it.



As soon as the target power supply has been switched off, the programmer automatically switches a resistive load across the Target System (UUT) which discharges any energy stored in reservoir capacitors on the UUT. The programmer constantly monitors the '**Target Voltage**' on the **TVCC** pin until the voltage has decayed to a specified voltage e.g. <math><200\text{ mV}</math> at which point the programmer signals that the UUT is safe to remove from the programming fixture. If the '**Target Voltage**' does not decay to the specified voltage in the user-programmable '**discharge timeout**' period, then an error condition is generated.

### Features:

- Programmer controlled '**Target Discharge Circuit**'
- Switches a '**resistive load**' (120 ohm resistor) across the '**Target Load**' (TVCC and GROUND) to discharge the UUT
- Used to automatically discharge any energy stored in reservoir capacitors on the Target System (UUT) after the UUT power has been switched off
- Prevents ESD damage to the UUT by ensuring there is no energy left in the UUT when it is disconnected from the programmer

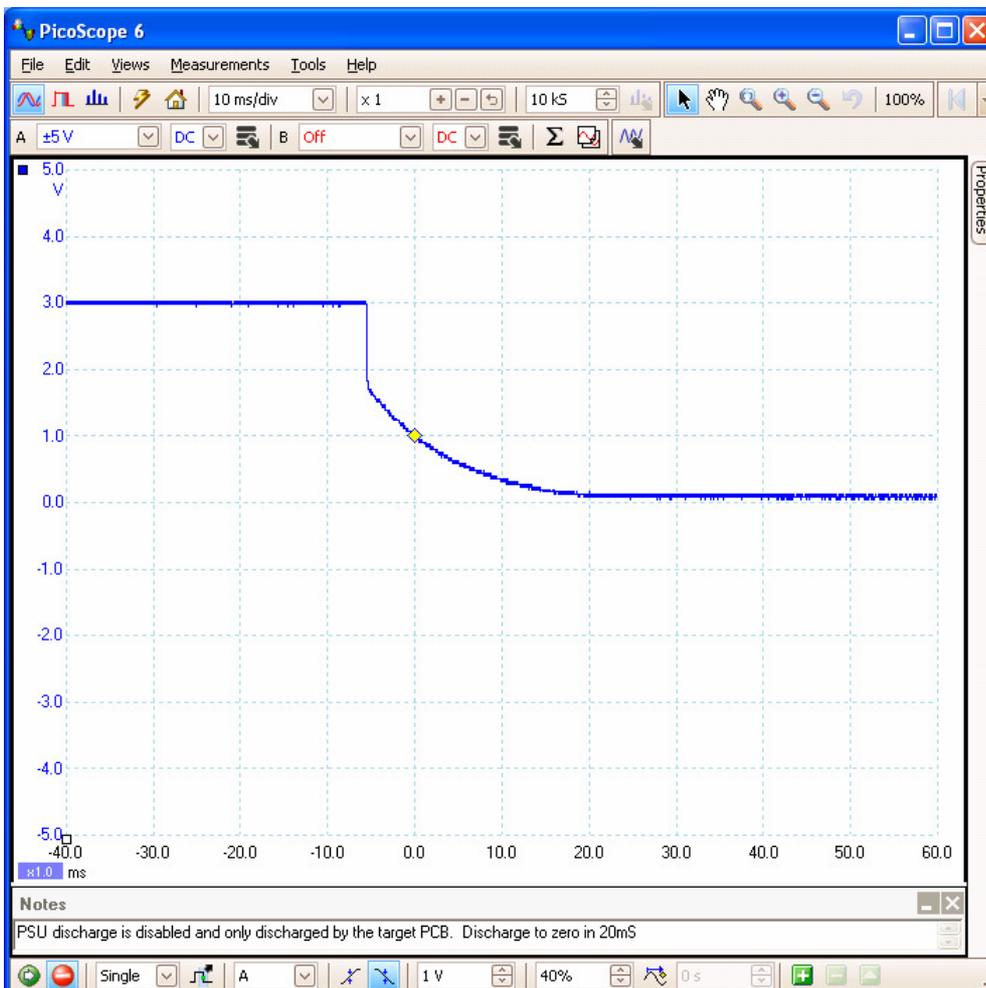
- Discharging the Target Vcc to <200 mV also forces most target microcontrollers to perform a 'Power-on-RESET' which they would not perform if power had not been fully removed.
- User-programmable '**discharge timeout**' and '**discharge voltage**'

**Please note:**

Many microcontroller based target systems feature low-power microcontrollers and do not have a resistive path to ground capable of discharging the reservoir capacitors on the board. This effectively means that after the power is removed from the Target System, the Target Vcc rail may not decay for minutes, hours and sometimes days. This has the effect of keeping power on the microcontroller so it may keep running even after the power has been removed. It also means that there is a potential ESD problem when the Target System is removed from the programmer.

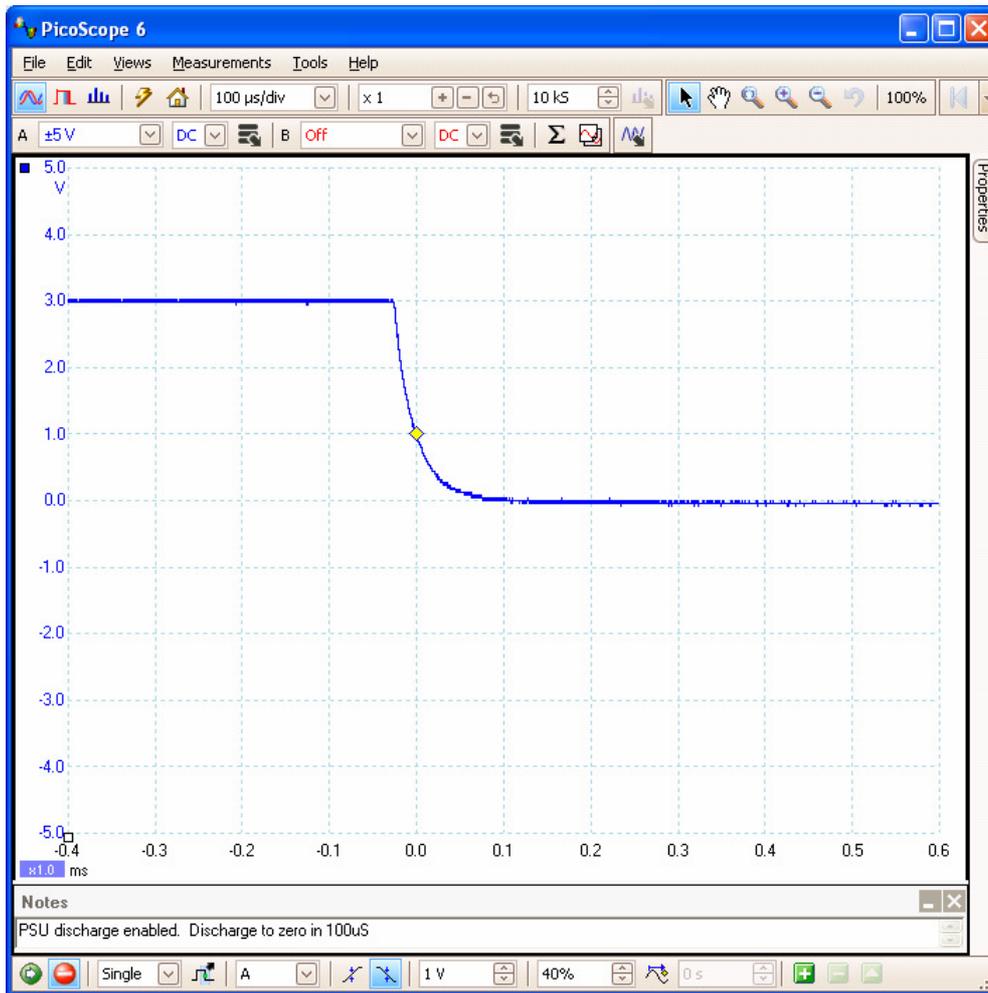
For many target microcontrollers, the target voltage must drop below e.g. 400 mV before the on-chip reset circuitry detects that power has been removed. If the power is not completely removed, then the target device can either remain in programming mode after the next power-up or it may end up getting stuck in a brown-out state where it never comes of reset. Using the '**Target Discharge Circuit**' should prevent these problems from happening.

**Scope trace showing the voltage decay with the 'Target Discharge Circuit' DISABLED....**



The '**Target Voltage**' decays from 3.0V to 0V in approximately 20ms.

Scope trace showing the voltage decay with the 'Target Discharge Circuit' ENABLED....

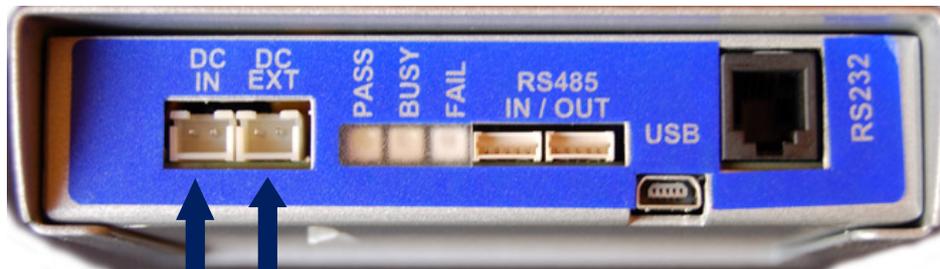


The '**Target Voltage**' decays from 3.0V to 0V in approximately 100us when the 'Target Discharge Circuit' is enabled. This is obviously much faster than leaving the voltage to decay on its own.

## 4.2 DC Power Connector overview

The ISPnano range of programmers each feature two '**DC Power**' connectors as shown in the illustrations below. These connectors are in the same position on all ISPnano programmers.

Fig. 4.2.1 ISPnano Series 2 / 3 – Power Connectors



- 2. DC-EXT – External Switched VCC Connector
- 1. DC-IN - DC Power INPUT Connector

Fig. 4.2.2 ISPnano Series 3 ATE – Power Connectors



- 2. DC-EXT - +12V DC input for ATE relays
- 1. DC-IN - DC Power INPUT Connector

Key No	Function	Voltage range	Further description
1	<b>DC-IN</b> DC Power Input	9.0 – 24.0V DC	<ul style="list-style-type: none"> <li>• Used to power the programmer internal circuitry and the '<b>Target Controlled Power Supply</b>'</li> <li>• Pin 1 (left pin) = 0V (GROUND)</li> <li>• Pin 2 (right pin) = +ve DC supply voltage</li> </ul>
2	<b>DC-EXT(IN)</b> External Switched DC Power Input	Series I / II / III 1.8 – 24.0V DC  Series III ATE 12.0V DC only	<ul style="list-style-type: none"> <li>• The '<b>External Supply voltage</b>' applied to this connector can be switched via the programmer to pins 3+4 of the '<b>Target ISP Connector</b>'</li> <li>• The 'ISPnano Series 3 ATE' programmer required to be fed with +12V on this connector to power the relay coils.</li> <li>• Pin 1 (left pin) = 0V (GROUND)</li> <li>• Pin 2 (right pin) = +ve DC EXTERNAL voltage</li> </ul>

## 4.4 Powering the programmer

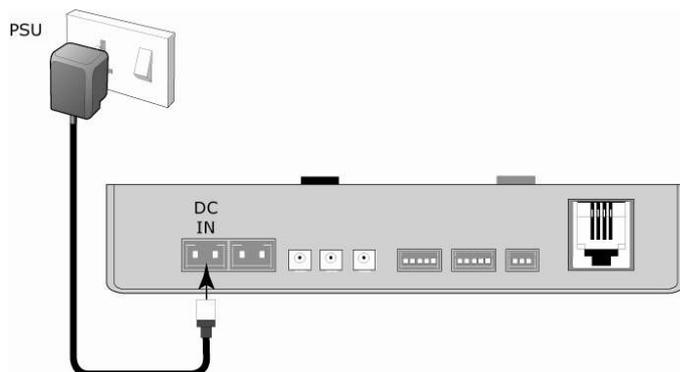
### 4.4.1 Overview

The ISPnano programmer can be powered by any of the following methods:

#	Function	Voltage range	Further description
1	Using AC – DC power supply adaptor	9.0V	<ul style="list-style-type: none"> <li>This is the 9V DC power supply which comes with the ISPnano kit.</li> </ul>
2	Using an external power supply	9.0 – 24.0V DC	<ul style="list-style-type: none"> <li>This is an external power supply.</li> <li>A '<b>DC Power Cable</b>' is required to connect to the '<b>DC IN</b>' connector.</li> </ul>

### 4.4.2 Powering the programmer using the AC Wall Adaptor

The ISPnano programmer kit comes with an 'AC Mains Adaptor' which converts 110 – 240V AC mains to a 9V DC regulated supply suitable for use with the programmer. This can be used to power the programmer – see connection diagram below.



#### Instructions:

- Plug the 2-pin JST connector on the end of the power supply cable into the 2-pin socket on the ISPnano programmer labelled "**DC IN**"
- Select the correct mains connector for your country and fit this to the Mains Adaptor.
- Plug the Mains Adaptor into a power socket and switch on the power.
- The ISPnano programmer should power up and the 'STATUS LED' should illuminate.

### ***4.4.3 Powering the programmer from an external power supply***

The ISPnano programmer can also be powered from an external power supply with any voltage between 9.0V and 24.0V using the “**DC IN**” connector. The ISPnano kit comes with a ‘**DC Power Cable**’ which has a 2-pin JST connector on one end and bare wires at the other end. This cable can be used to connect an external power supply to the programmer “**DC IN**” connector.

**Please note:**

- Pin 1 (left pin) = 0V (GROUND)
- Pin 2 (right pin) = +ve DC EXTERNAL voltage
- The +ve wire is indicated by a ridge on the side of the wire and also a red piece of sleeving.

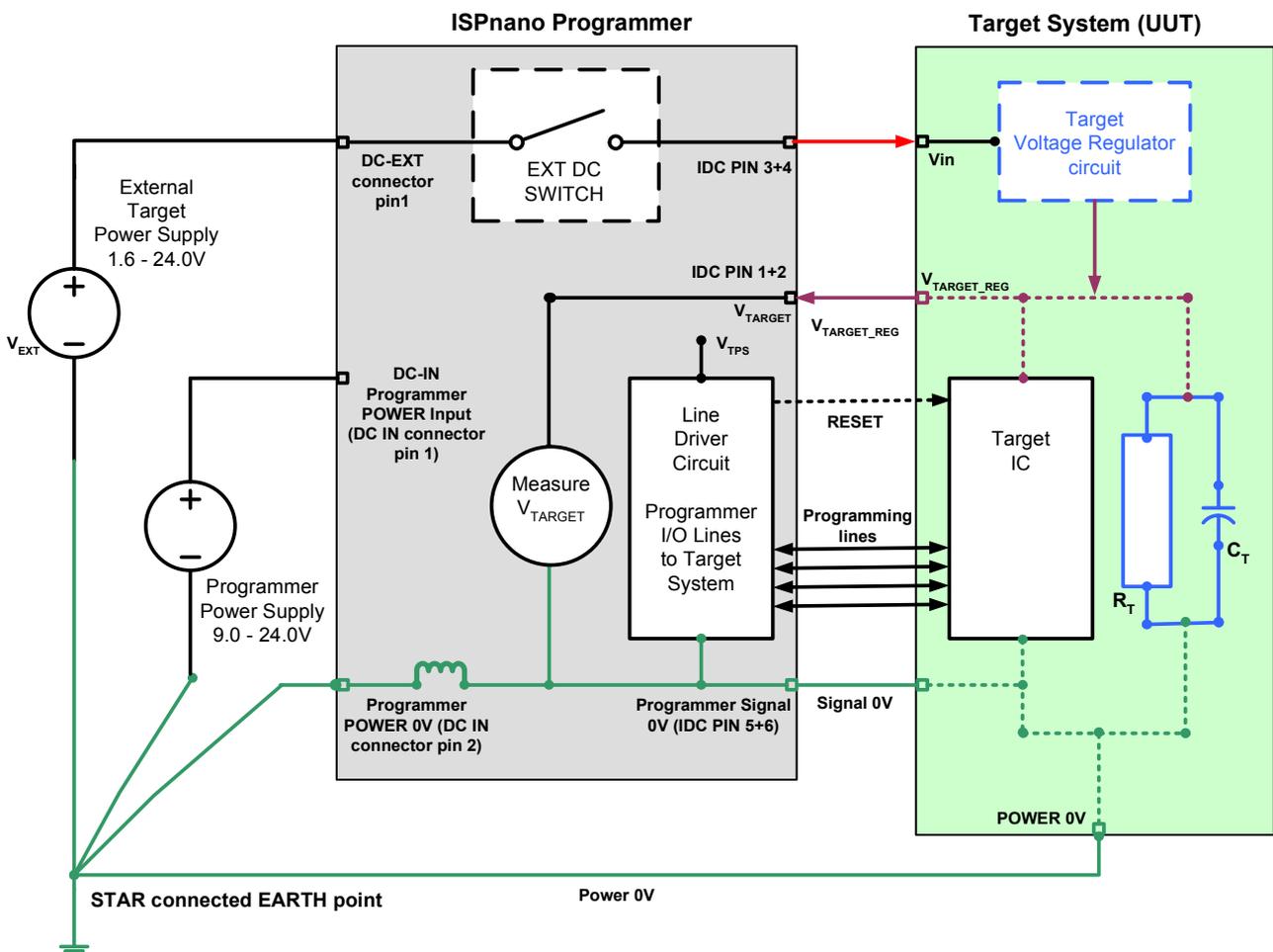
### 4.5 Switching EXTERNAL VCC to the Target System

#### 4.5.1 Overview

The ISPnano programmer features a special ‘**External VCC Switch**’ circuit which allows an ‘**External Voltage ( $V_{EXT}$ )**’ in the range of 1.6V – 24V to be switched through the programmer to the Target System under project control. This feature is useful when the Target System features a ‘**Voltage Regulator Circuit**’ which needs to be powered from a higher voltage than +5V. In order to power many Target Systems, it is necessary to apply a voltage of eg. 7 – 24V to the input of the Target System regulator circuit.

The diagram below shows a typical example where an external power supply is switched to the Target System through the programmer.

**Fig. 4.5.1. ISPnano – Powering Target System using External Switched Voltage**



- An ‘**External**’ voltage ‘ **$V_{EXT}$** ’ in the range of 1.6 – 24.0V is fed into the **EXT-VCC-IN** connector of the programmer from an external power supply.
- This voltage is then switched by the ‘**External Vcc Switch**’ inside the programmer to the **EXT-VCC-OUT** pins on the ‘**Target ISP Port**’ of the programmer.

- The voltage is then applied to the **input** of the '**Voltage Regulator Circuit**' on the Target System.
- The '**Voltage Regulator Circuit**' on the Target System then generates the supply rail(s) voltage(s) for the device(s) to be programmed – **V\_TARGET\_REG** in the above diagram.
- This generated voltage is the voltage that the device(s) to be programmed are actually running at and is known as the '**Target Vcc (TVCC)**'. The voltage is usually in the range of 1.6 – 5.0V for most programmable devices.
- The programmer can measure the '**Target Vcc (TVCC)**' to make sure that this voltage is within the limits specified for the Target Board.
- The Target System requires that a '**POWER\_GROUND**' connection is made from the GROUND of the Target System back to the star-connected GROUND connection of the power supply / test fixture.

**Important note:**

If the programmer is switching an '**External power supply**' to the Target System and then measuring the Target Voltage (on the TVCC pin) then the programmer must wait for the '**Target Voltage**' to stabilise at its final value after switching the '**External power supply**' on before measuring the voltage.

#### 4.5.2 How the programmer measures the Target Vcc

When the programmer is NOT controlling Target Power but instead is switching an '**External Supply**' to the Target System, it is still possible for the programmer to measure and check the 'Target Voltage'.

This is how the programmer handles switching on of the '**External Vcc**'....

- The operator must manually switch on the **EXTERNAL VOLTAGE (V\_EXT)** to the '**DC-EXT**' connector.
- Start the programming project or press **<Power up>** in EDS mode
  - Programmer switches on the '**External VCC Switch**' and sets an internal timer to t=0.
  - The regulator / power supply on the Target System will then switch on and start to charge on any on-board capacitance such as reservoir capacitors.
- The programmer then waits for the '**Voltage Settle time**' to elapse which gives time for the any capacitance to charge up and the '**Target Voltage**' to settle to its final value.
- The programmer then starts to check that the '**Target Voltage**' measured on the TVCC pin is within the limits specified in the project. If it is not, then an '**Error 64 – Target voltage is outside limits**' is immediately generated.
- If the voltage is within limits, the programmer will continue to check '**Target Voltage**' is within limits until the '**PSU\_OUT delay**' has elapsed. This ensures that the voltage is definitely stable (ie not fluctuating) before the programming operation commences.

**Example:**

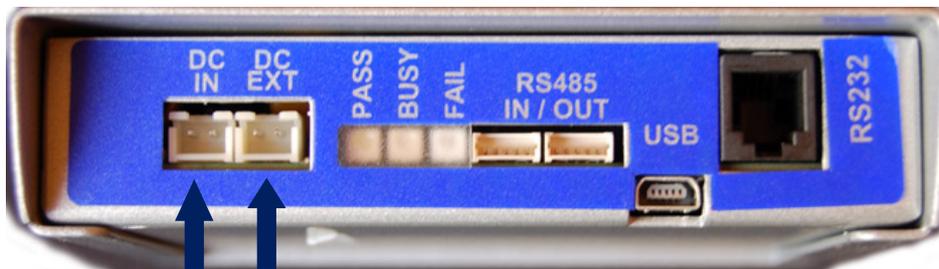
Let's say the power supply on the Target System has a 24V input voltage and a 3.3V output voltage. The time taken for the 3.3V supply to reach 3.3V is approximately 600ms after the 24V supply has

been switched on. This delay could be due to the 24V supply having to charge up the capacitance on the Target System. If this 24V supply is current limited to a sensible value eg. 500mA, then it will take a finite time for the 3.3V rail to reach 3.3V.

### 4.5.3 Connection instructions

If you wish to power the Target System using the **'External Switched Supply'**, please follow the connection instructions detailed in the table below.

Fig. 4.5.1 ISPnano Series 2 / 3 – Power Connectors



- 2. DC-EXT – Connect external 1.8V – 30.0V supply
- 1. DC-IN – Connect 9.0V – 24.0V DC to power programmer

Programmer Signal name	Function	Voltage range (V DC)	Programmer Connector	Connect to.... ....on the Target System
DC IN	DC Power Input	9.0 – 24.0V	DC-IN – pin 2	Power supply input for programmer
EXT-VCC-IN	External Switched DC Power Input	1.8 – 24.0V	EXT-VCC-IN – pin 2	Connect to <b>'EXTERNAL'</b> power supply source.
EXT-VCC-OUT	Switched output of <b>'External VCC'</b> supply	1.8 – 24.0V	<b>'Target ISP Port'</b> - pins 3+4	Connect to the <b>INPUT</b> of the Target System regulator circuit.
TVCC	Target Vcc Supply / measurement	1.2 – 5.0 V	<b>'Target ISP Port'</b> - pins 1+2	Connect to the <b>OUTPUT</b> of the Target System regulator circuit or the <b>Vcc rail</b> for the IC to be programmed.
SIGNAL_GROUND	0V Signal GROUND	0V	<b>'Target ISP Port'</b> - pins 5+6	Signal GROUND between programmer and Target System.
POWER_GROUND	0V Power GROUND	0V	<b>Do not connect!!!</b>	Connect GROUND wire from 0V of Target System to star-connected 0V of main power supply.

Please refer to fig 4.5.1 for a schematic showing the connections and pin numbers.

#### 4.5.4 Setting up a project in EDS to externally power the Target System

The best way to test that your hardware set up is correct is using the EDS – Development Mode.

##### Instructions:

- Launch EQTools
- Create a new EDS – Development Project
- Select your target device
- On the '**Target Power Supply**' screen, select the following settings:

The screenshot displays the 'Target Power Supply' configuration window in the EDS software. It is divided into several sections:

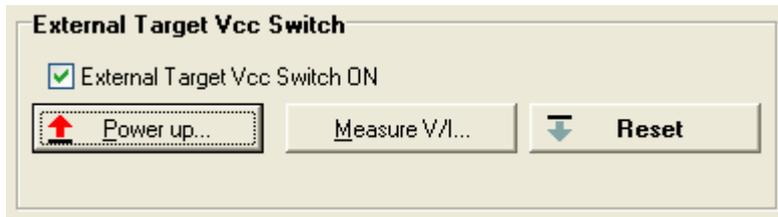
- Target Voltage Settings:** Includes input fields for Voltage (3.3V), Tolerance (500 mV), and Stabilise Time (200 ms). A checkbox for 'Check Target Voltage' is checked.
- Programmer Controlled Power Supply:** A dropdown menu is set to 'OFF'. Below it are several timing parameters: Maximum Current (200 mA), Powerdown Time (1000 ms), Current Settle Time (100 ms), PSU Out OK Delay (500 ms), and Voltage Settle Time (100 ms). A dropdown for 'Power Status at end of project' is set to 'Programmer Line Driver voltage is permanently enabled'. Buttons for 'Power up...', 'Measure V/I...', and 'Reset' are present.
- Target Discharge Circuit:** A checkbox 'Target Discharge Circuit ON' is checked.
- External Target Vcc Switch:** A checkbox 'External Target Vcc Switch ON' is checked. It also includes 'Power up...', 'Measure V/I...', and 'Reset' buttons.

##### Notes:

- The '**Programmer controlled Power Supply**' must be switched **OFF** so that it does not clash with the power supply on the Target System.
- The '**EXTERNAL Switched Vcc Supply**' must be switched **ON** in the Programming Project.
- The 'Target Voltage settings' are used to power the programmer '**Line Drivers**' and so this voltage must be set in the '**Programming Project**' to the same value as the '**Target Voltage**' which the Target IC is running from. So, if the Target IC actually runs from 3.3V then the voltage should be set to 3.3V in the project. The programmer I/O lines will then swing between 0V and 3.3V.
- The '**Target Voltage (TVCC)**' can still be measured by the programmer.
- A volt drop of 50 to 150mV across the FET switch inside the programmer may cause the '**EXTERNAL VCC OUTPUT Voltage**' to be slightly lower than the '**Input Voltage**'.

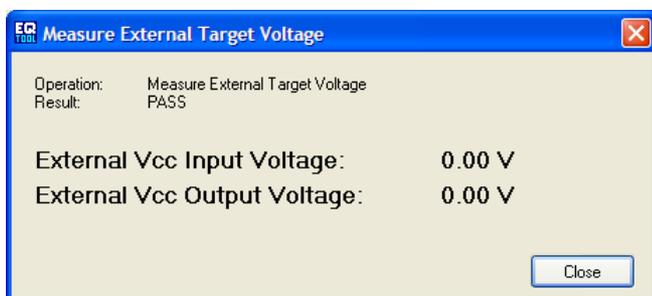
### 4.5.5 Measuring the External Target Vcc Voltage

The panel which controls the switching on/off of the '*External Target Vcc*' voltage is shown below....



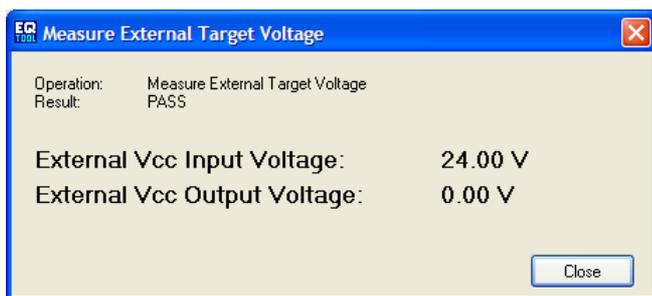
#### Instructions:

- Ensure that the '*EXTERNAL VCC OUTPUT Voltage*' is switched **OFF** at the supply.
- Disconnect any cables connected to the '*Target ISP Connector*'. This will prevent accidental damage in case of a wiring error!
- Click the *<Measure V/I>* button to display the voltage measured....



→ Both the '*External Target*' INPUT and OUTPUT voltages should read 0.0V.

- Switch the '*External Target Vcc*' supply ON (by manually switching on the external power supply).
- Click the *<Measure V/I>* button to display the voltage measured....

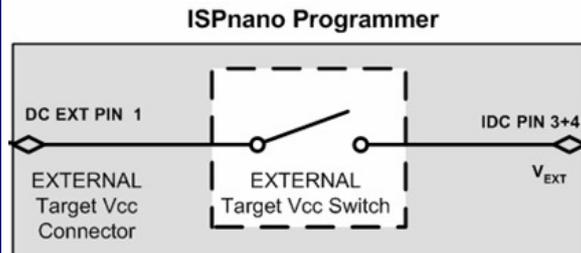
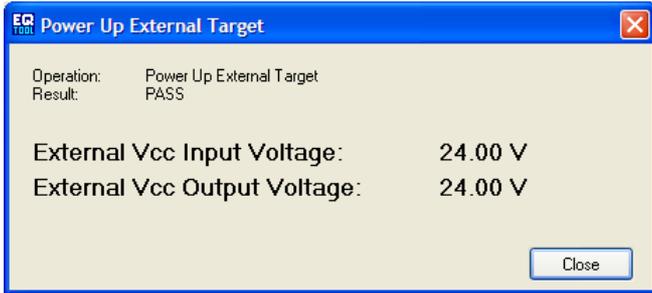


→ The programmer now measures 24.0V on the '*External Target Vcc*' input but 0.0V on the output as the '*External Target Vcc switch*' is OFF.

Click the *<Power up..>* button to switch the '*External Target Vcc switch*' ON.

→ The '*BUSY*' LED on the programmer will illuminate.

→ The '*External Target Vcc switch*' is switched ON and so 24.0V should be measured at both input and the **EXT-VCC-OUT** output (pins 3+4 of the '*Target ISP Port*').



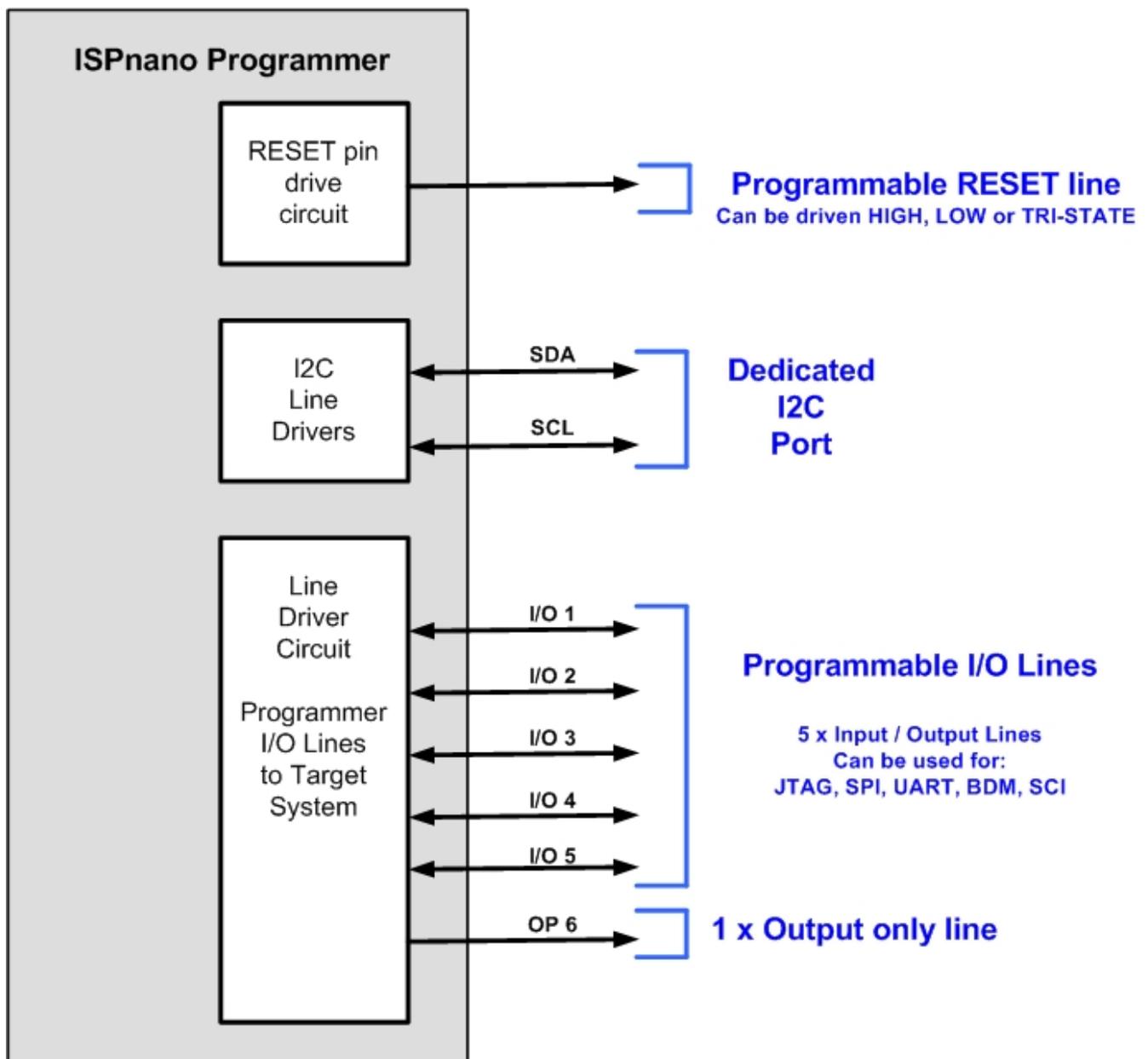
# Appendix 1 – ISPnano Series I / II - Target ISP Connector Port

## 1.1 Programmer I/O Signals

The ISPnano programmer features the following programmable signal lines which interface the programmer and the Target IC(s) to be programmed on the Target System:

- 5 x Programmable **Input / Output** lines
- 1 x Programmable **Output only** line
- 1 x Dedicated **I2C (Two Wire Interface)** port
- 1 x Dedicated **RESET** pin

The diagram below shows the available programmer I/O lines:



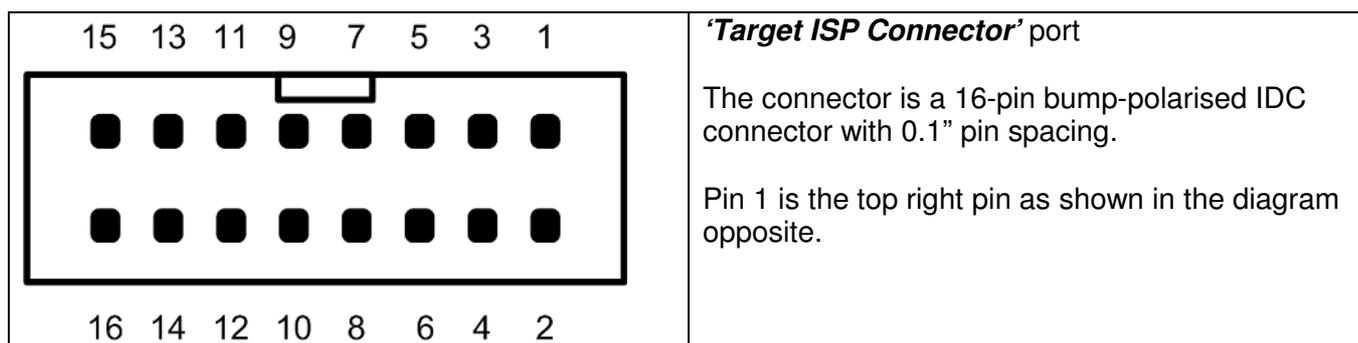
**Please note:**

These I/O signals are available on the 16-way IDC Connector labelled “**ISP PORT**” on the ISPnano Series 1 and 2 programmers and labelled “**Target ISP PORT**” on the ‘**Series 3 ATE**’ programmer.

## 1.2 Target ISP Port – connector pin-out

The **‘Target ISP Connector’** port features all the signals required to implement In-System Programming (ISP) of a Target IC using the SPI, JTAG, I2C or UART interface. This connector also features the programmable **“Target Vcc”** and **“Target Vpp”** voltages plus a switched **“EXTERNAL Vcc”** supply.

The illustration below shows the location of the **‘Target ISP Connector’** port on the rear panel of the programmer.



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

Pin No	Programmer Pin name	Programmer Input / Output	Connect to pin on Target System	Notes
1 + 2	TARGET_VCC	P	TARGET_VCC	<b>Target VCC</b> This pin should be connected to the Target System Vcc.
3 + 4	TARGET_EXT_VCC	P	See notes.	<b>Target External VCC</b> This pin should be connected to the Target System Vcc.  Only use this pin if the <b>“External Target Vcc”</b> is to be switched to the <b>“Target Vcc”</b>
5 + 6	PROG_GND	P	<b>Signal GROUND (0V)</b>	<b>Signal Ground Connection (1)</b> 0V to which the programmer JTAG, SPI, I2C signal lines are referenced to.
7	I2C_SCL	I/O	I2C SCL	<b>I2C SCL clock signal</b> This is the I2C SCL clock signal.
8	I2C_SDA	I/O	I2C SDA	<b>I2C SDA data signal</b> This is the I2C SDA data signal.
9	OP6_SPARE	O	Spare Output	<b>Spare Output</b> This is a spare output pin. It is designed to control the activation of the relay coils on the Relay Board.

10	Programmer I/O5	I/O	Spare I/O	
11	Programmer I/O4	I/O	<ul style="list-style-type: none"> <li>JTAG – TMS</li> </ul>	This pin is used for JTAG Target Systems only.
12	Programmer I/O3	I/O	<ul style="list-style-type: none"> <li>JTAG – TCK</li> <li>SPI - SCK</li> </ul>	This pin can be used for JTAG or SPI Target Systems.
13	Programmer I/O2	I/O	<ul style="list-style-type: none"> <li>JTAG – TDO</li> <li>SPI – MISO</li> <li>UART - RXD</li> </ul>	This pin can be used for JTAG, SPI or UART Target Systems.
14	Programmer I/O1	I/O	<ul style="list-style-type: none"> <li>JTAG – TDI</li> <li>SPI – MOSI</li> <li>UART - TXD</li> </ul>	This pin can be used for JTAG, SPI or UART Target Systems.
15	PROG_VPP	P	See note	<b>Vpp Voltage</b> The programmer can output a “Vpp” voltage on this pin between 6.5V and 13.8V. This pin should not be connected unless a Vpp voltage is required by the Target IC.
16	PROG_RESET	O	<b>RESET</b>	<b>Target RESET control pin</b> This pin controls the Target Device RESET pin. It will driven HIGH/LOW according to the device type and settings in the <Pre-program State Machine> tab in the EQtools project.

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

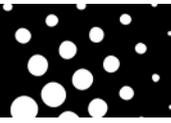
**Please note:**

**Signal GROUND (0V)**

The ‘**Signal GROUND (0V)**’ connection is the 0V to which the programming lines (JTAG, SPI, I2C, UART) are referenced to. This 0V line should be connected directly to the Target System (UUT) 0V connection via the shortest possible wire link.

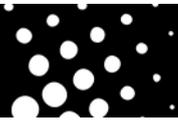
**Target Vcc**

The ‘**Target Vcc**’ is the output of the ‘Controlled Power Supply’ from the programmer. The power supply has a range of 1.8 to 5.0V and can be switched ON / OFF to the Target System under programmer control.



### ***External Target Vcc***

The '***External Target Vcc***' is the voltage which is applied to the '***DC EXT***' connector of the programmer. It is electronically switched to pins 3+4 inside the programmer. This allows an external voltage from 1.8 – 24.0V to be switched to the Target System.



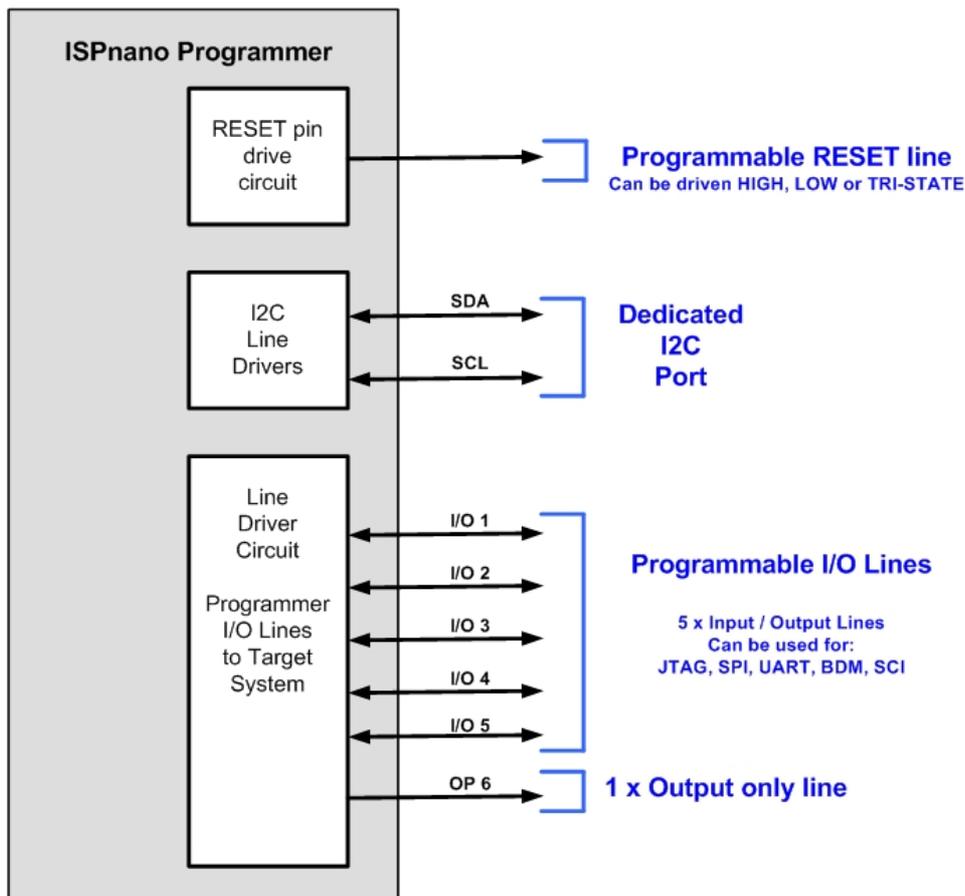
## Appendix 2 – ISPnano Series III (ATE) - Target ISP Connector Port

### 1.1 Programmer I/O Signals

The ISPnano Series III / Series III ATE programmers feature the following programmable signal lines which interface the programmer and the Target IC(s) to be programmed on the Target System:

- 5 x Programmable **Input / Output** lines
- 1 x Programmable **Output only** line
- 1 x 2-wire multi-function port supporting the following interfaces:
  - I2C (Two Wire Interface)
  - Atmel XMEGA 2-wire PDI port
  - Atmel ATtiny 2-wire TPI port
- 1 x Dedicated **RESET** pin

The diagram below shows the available programmer I/O lines:



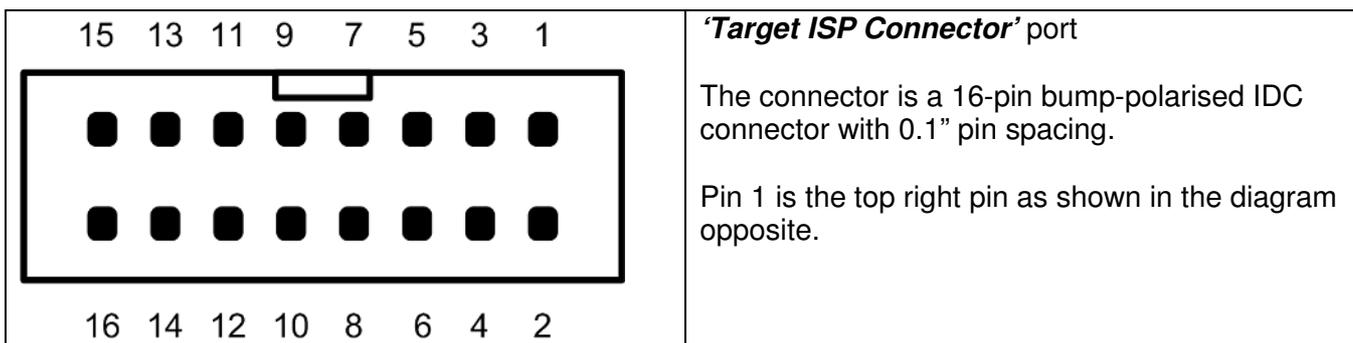
**Please note:**

These I/O signals are available on the 16-way IDC Connector labelled “**ISP PORT**” on the ISPnano Series 1 and 2 programmers and labelled “**Target ISP PORT**” on the ‘**Series 3 ATE**’ programmer.

### 1.2 Target ISP Port – connector pin-out

The **‘Target ISP Connector’** port features all the signals required to implement In-System Programming (ISP) of a Target IC using the SPI, JTAG, I2C, XMEGA PDI, ATtiny TPI or UART interface. This connector also features the programmable **“Target Vcc”** and **“Target Vpp”** voltages plus a switched **“EXTERNAL Vcc”** supply.

The illustration below shows the location of the **‘Target ISP Connector’** port on the rear panel of the programmer.



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

Pin No	Programmer Pin name	I/O	Connect to pin on Target System	Notes
1 + 2	TARGET_VCC	P	TARGET_VCC	<b>Target VCC</b> This pin should be connected to the Target System Vcc.
3 + 4	TARGET_EXT_VCC	P	See notes.	<b>Target External VCC</b> This pin should be connected to the Target System Vcc.  Only use this pin if the <b>“External Target Vcc”</b> is to be switched to the <b>“Target Vcc”</b>
5 + 6	PROG_GND	P	<b>Signal GROUND (0V)</b>	<b>Signal Ground Connection (1)</b> 0V to which the programmer JTAG, SPI, I2C signal lines are referenced to.
7a 7b 7c	I2C_SCL XMEGA_PDI_CLK ATTINY_TPI_CLK	I/O O O	<ul style="list-style-type: none"> <li>I2C_SCL</li> <li>RESET</li> <li>TPI_CLK</li> </ul>	<ul style="list-style-type: none"> <li>I2C_SCL clock signal</li> <li>XMEGA CLOCK Signal</li> <li>ATtiny CLOCK Signal</li> </ul>
8a 8b 8c	I2C_SDA XMEGA_PDI_DATA ATTINY_TPI_DATA	I/O I/O I/O	<ul style="list-style-type: none"> <li>I2C_SDA</li> <li>TEST (PDI_DATA)</li> <li>TPI_DATA</li> </ul>	<ul style="list-style-type: none"> <li>I2C_SDA data signal</li> <li>XMEGA DATA Signal</li> <li>ATtiny DATA Signal</li> </ul>

9	OP6	O	Algorithm specific	<b>Spare Output</b>
10	Programmer I/O5	I/O	<ul style="list-style-type: none"> <li>• Spare I/O</li> </ul>	
11	Programmer I/O4	I/O	<ul style="list-style-type: none"> <li>• JTAG – TMS</li> </ul>	This pin is used for JTAG Target Systems only.
12	Programmer I/O3	I/O	<ul style="list-style-type: none"> <li>• JTAG – TCK</li> <li>• SPI - SCK</li> </ul>	This pin can be used for JTAG or SPI Target Systems.
13	Programmer I/O2	I/O	<ul style="list-style-type: none"> <li>• JTAG – TDO</li> <li>• SPI – MISO</li> <li>• UART – RXD</li> <li>• PDI_RXD</li> </ul>	This pin can be used for JTAG, SPI, UART, XMEGA PDI and ATtiny TPI Target Systems.
14	Programmer I/O1	I/O	<ul style="list-style-type: none"> <li>• JTAG – TDI</li> <li>• SPI – MOSI</li> <li>• UART - TXD</li> </ul>	This pin can be used for JTAG, SPI or UART Target Systems.
15	PROG_VPP	P	See note	<b>Vpp Voltage</b> The programmer can output a “Vpp” voltage on this pin between 6.5V and 13.8V. This pin should not be connected unless a Vpp voltage is required by the Target IC.
16	PROG_RESET	O	<b>RESET</b>	<b>Target RESET control pin</b> This pin controls the Target Device RESET pin. It will driven HIGH/LOW according to the device type and settings in the <Pre-program State Machine> tab in the EQtools project.

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

**Please note:**

**Signal GROUND (0V)**

The ‘**Signal GROUND (0V)**’ connection is the 0V to which the programming lines (JTAG, SPI, I2C, UART, PDI, TPI) are referenced to. This 0V line should be connected directly to the Target System (UUT) 0V connection via the shortest possible wire link.

**Target Vcc**

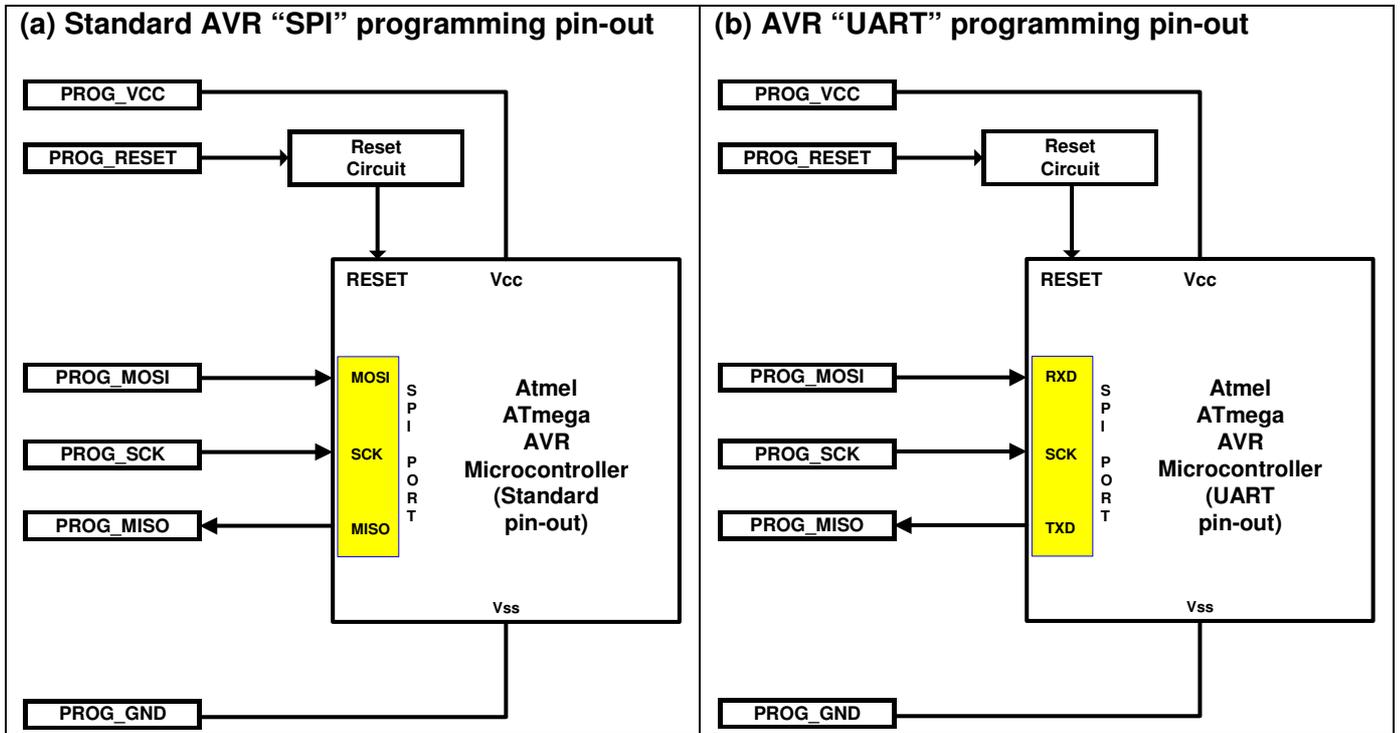
The ‘**Target Vcc**’ is the output of the ‘**Controlled Power Supply**’ from the programmer. The power supply has a range of 1.8 to 5.0V and can be switched ON / OFF to the Target System under programmer control.

***External Target Vcc***

The '***External Target Vcc***' is the voltage which is applied to the '***DC EXT***' connector of the programmer. It is electronically switched to pins 3+4 inside the programmer. This allows an external voltage from 1.8 – 24.0V to be switched to the Target System.

### 1.3 Atmel AVR microcontroller - SPI connections

Most Atmel AVR microcontrollers use the standard “SPI” pins for In-System Programming – see diagram (a) below. However, certain ATmega devices actually use the UART pins (RXD / TXD) for In-System Programming – see diagram (b) below. Please refer to Application Note AN101 for detailed instructions of how to program an Atmel AVR microcontroller via the SPI interface.



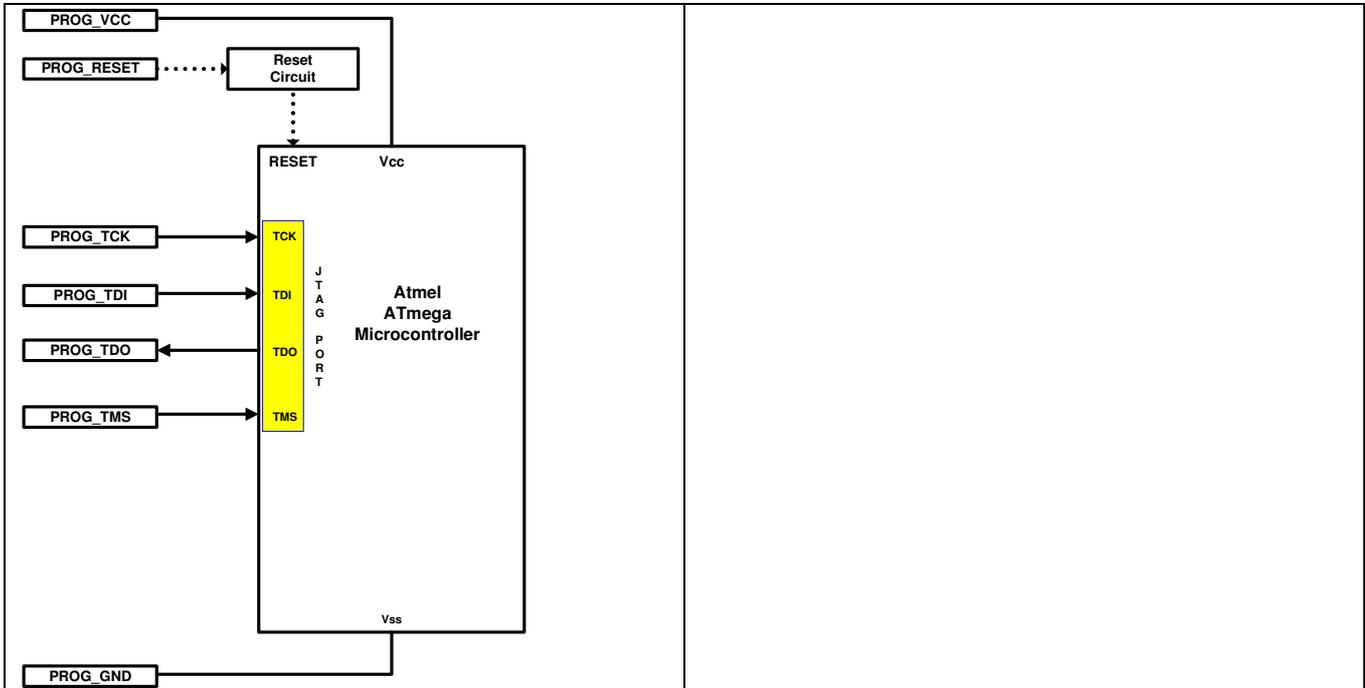
The table below details the connections for programming Atmel AVR microcontrollers via the SPI Interface.

Pin No	Programmer Pin name	Programmer Input / Output	Connect to pin on Target System	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	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

### 1.4 Atmel AVR microcontroller - JTAG connections

The table below details the connections for programming Atmel AVR microcontrollers via the JTAG Interface.

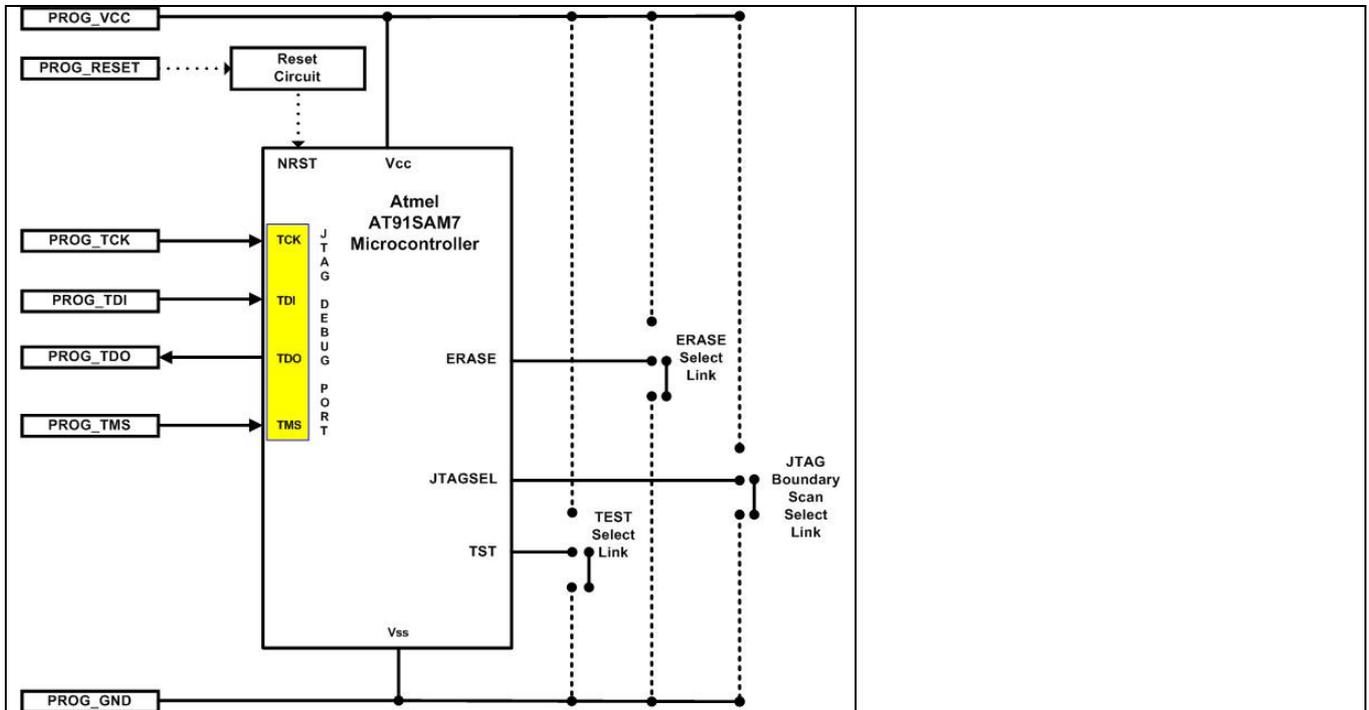


Pin No	Programmer Pin name	Programmer Input / Output	Connect to pin on Target System	Notes
1 + 2	TARGET_VCC	P	TARGET_VCC	Target VCC
3 + 4	TARGET_EXT_VCC	P	See notes.	Target External VCC
5 + 6	PROG_GND	P	Signal GROUND (0V)	Signal Ground Connection
10	Programmer I/O5	I/O	Spare I/O	
11	Programmer I/O4	I/O	JTAG – TMS	
12	Programmer I/O3	I/O	JTAG - TCK	
13	Programmer I/O2	I/O	JTAG - TDO	
14	Programmer I/O1	I/O	JTAG - TDI	
16	PROG_RESET	O	RESET	Target RESET control pin

Please refer to Application Note AN105 for detailed instructions of how to program an Atmel AVR microcontroller via the JTAG interface.

## 1.5 Atmel AT91SAM7 microcontroller - JTAG connections

The schematic below shows the connections required to In-System Program (ISP) an Atmel AT91SAM7 ARM7 microcontroller via the JTAG Interface.



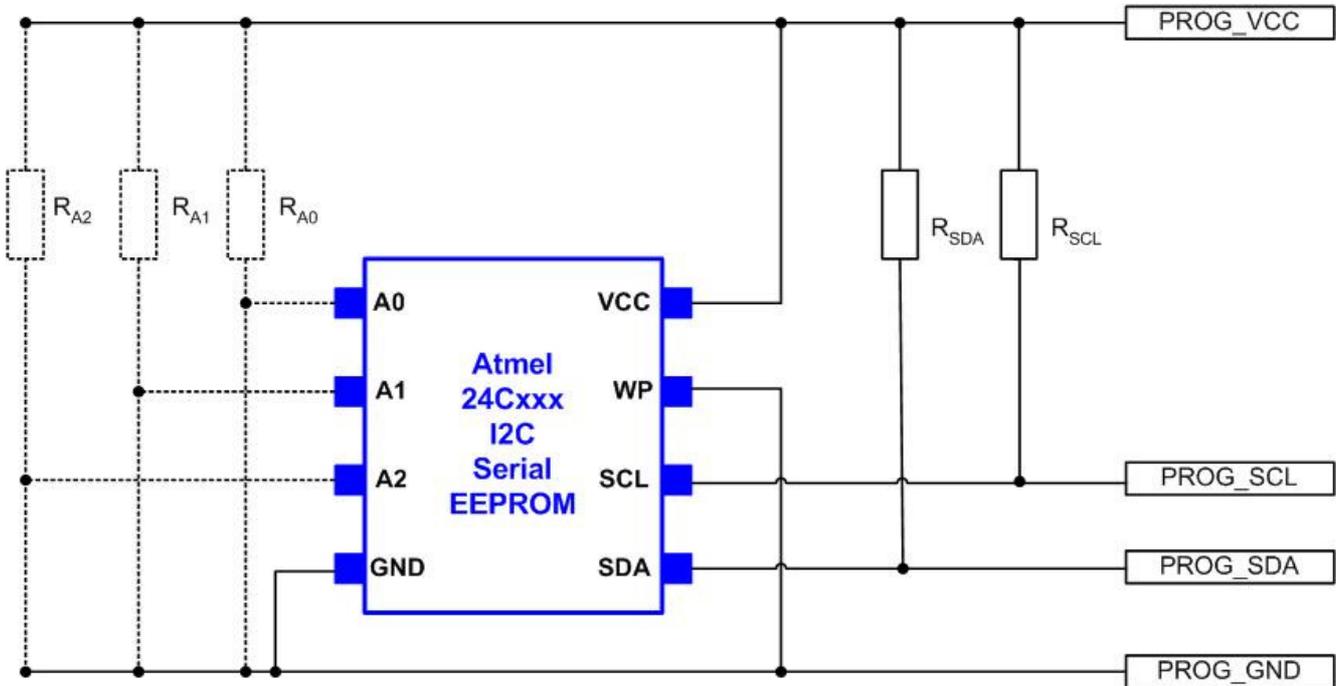
The table below details the connections for programming Atmel AVR microcontrollers via the JTAG Interface.

Pin No	Programmer Pin name	Programmer Input / Output	Connect to pin on Target System	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
10	Programmer I/O5	I/O	Spare I/O	
11	Programmer I/O4	I/O	JTAG – TMS	
12	Programmer I/O3	I/O	JTAG - TCK	
13	Programmer I/O2	I/O	JTAG - TDO	
14	Programmer I/O1	I/O	JTAG - TDI	
16	PROG_RESET	O	RESET	Target RESET control pin

Please refer to Application Note AN122 for detailed instructions of how to program an Atmel AT91SAM7 microcontroller.

### 1.6 Serial EEPROM – I2C connections

The schematic below shows the connections required to In-System Program (ISP) a single 24xxx Serial EEPROM Memory device using the ISPnano programmer.



The table below details the connections for programming an I2C Serial EEPROM device via the I2C Interface.

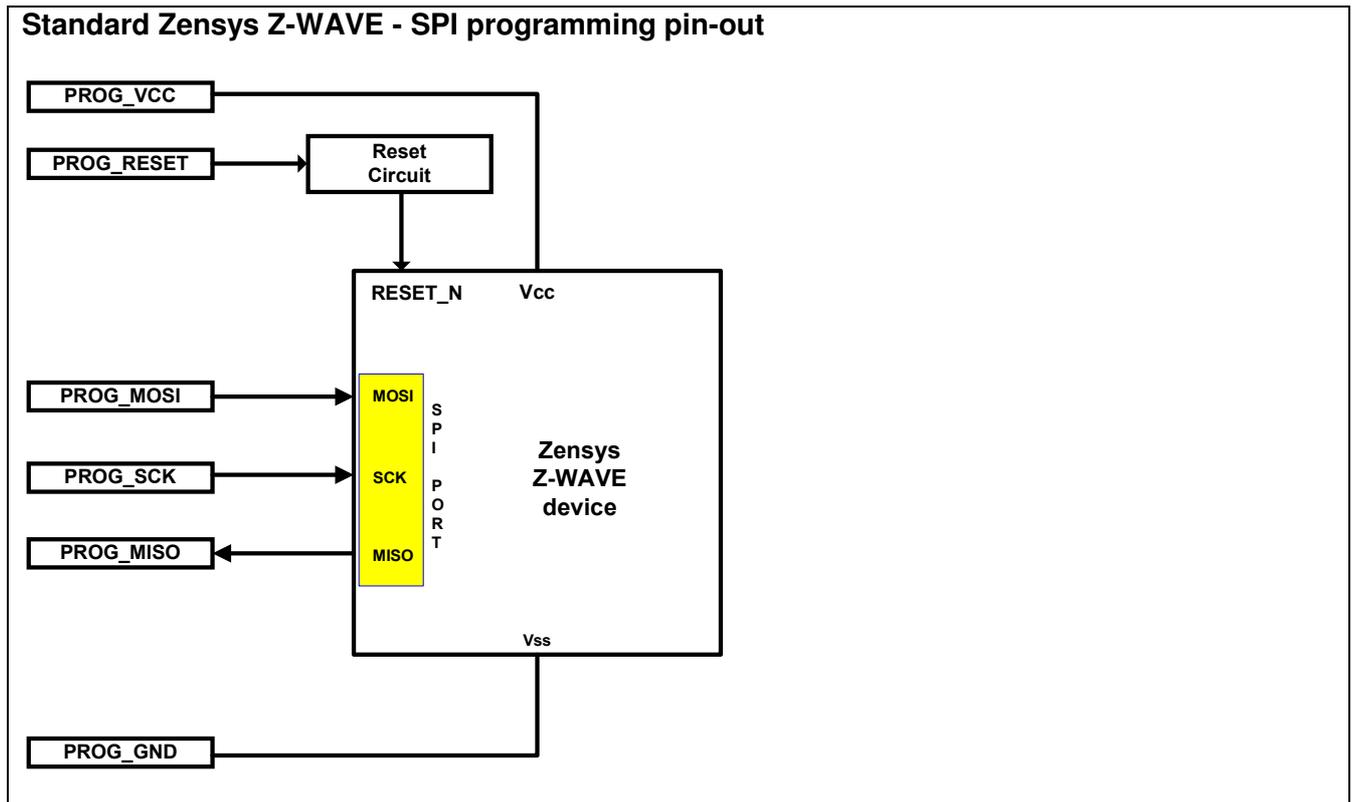
Pin No	Programmer Pin name	Programmer Input / Output	Connect to pin on Target System	Notes
1 + 2	TARGET_VCC	P	TARGET_VCC	<b>Target VCC</b> This pin should be connected to the Target System Vcc.
5 + 6	PROG_GND	P	<b>Signal GROUND (0V)</b>	<b>Signal Ground Connection (1)</b> 0V to which the programmer JTAG, SPI, I2C signal lines are referenced to.
7	I2C_SCL	I/O	I2C SCL	<b>I2C SCL clock signal</b> This is the I2C SCL clock signal.
8	I2C_SDA	I/O	I2C SDA	<b>I2C SDA data signal</b> This is the I2C SDA data signal.

Please refer to Application Note AN118 for detailed instructions of how to program Serial EEPROM memory device via the I2C interface.

## 1.7 Zensys Z-WAVE devices - SPI connections

### 1.7.1 Overview

The *Zensys 100 / 200 / 300 series Z-WAVE* devices use a standard **'SPI interface'** for In-System Programming – see diagram below. A connection to the **RESET\_N** pin is also required to force the device into programming mode. The diagram below shows the connections required between the programmer and the Target IC in order to program these devices.



The table below details the connections for programming Zensys Z-WAVE devices via the SPI Interface.

Pin No	Programmer Pin name	Programmer Input / Output	Connect to pin on Target System	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

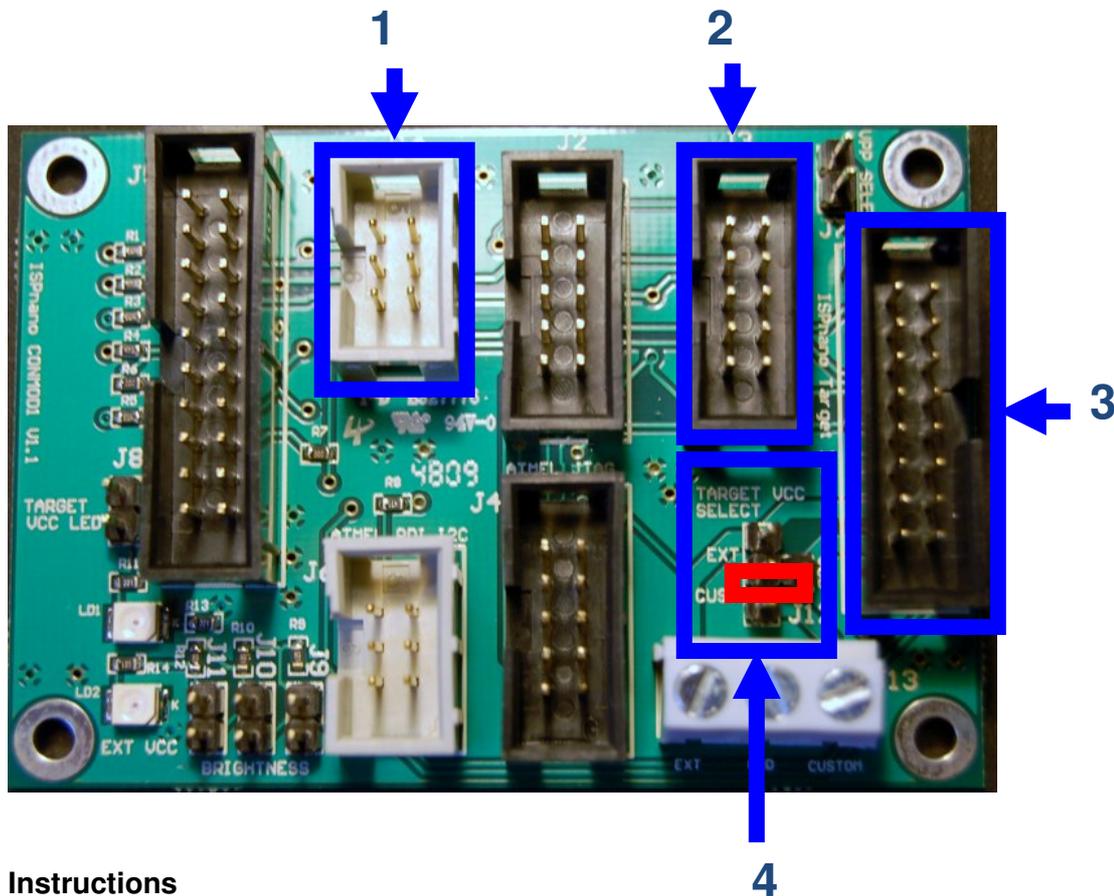
O - Output from programmer to Target Device  
 I - Input to programmer from Target Device  
 P - Passive e.g. GROUND and power rails  
 N/C - Not connected

It is possible to connect the Zensys Target Board directly to the programmer '*Target ISP Connector*' or you can use the '*CONMOD – ISPnano Connector Module*'.

### 1.7.2 Using the CONMOD module

It is possible to use the '*CONMOD – ISPnano Connector Module*' to connect to a Zensys Target Board. The '*CONMOD*' board features the standard '*Equinox 10-way IDC*' header which Zensys have adopted for use on many of their evaluation boards. It can therefore be used to connect directly to any suitable Zensys evaluation board.

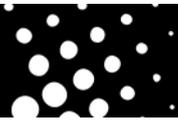
The relevant connectors for Zensys Z-WAVE device programming are shown in the illustrated picture of the '*CONMOD – ISPnano Connector Module*' below...



#### Instructions

- 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).

- If you wish to use the '**Equinox 10-way SPI header**', connect your Zensys Target System to the 10-way header labelled '**Equinox ISP**' on the CONMOD module – see (2) on above diagram.
- If you wish to use the '**Equinox 6-way SPI header**', connect your Zensys Target System to the 6-way header labelled '**Atmel ISP**' on the CONMOD module – see (1) on above diagram.
- Set up the '**Target Vcc Select**' jumper so that the programmer powers the Target Board – see red box marked (2) in the picture.



## Appendix 3 - Remote Control Port

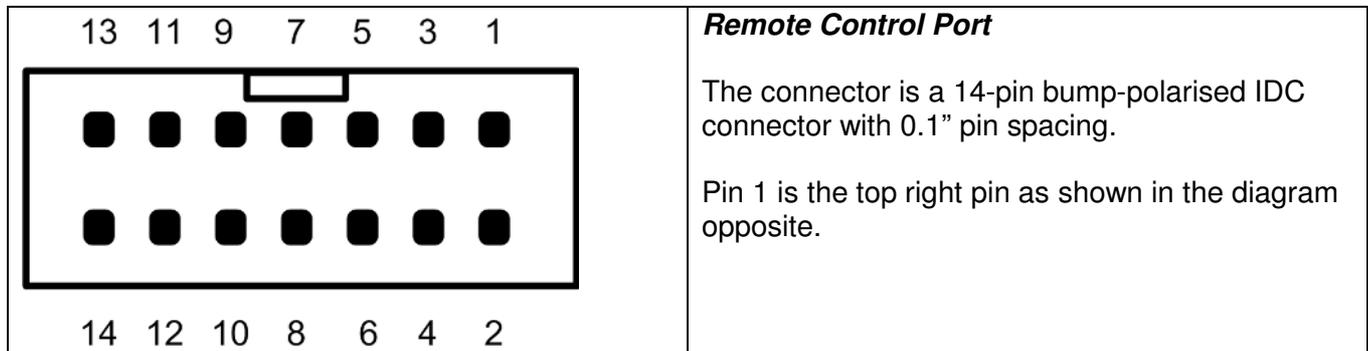
### 1.1 Overview

The '**Remote Control Port**' features all the connections required to control the programmer via a '**4-wire TTL control**' protocol or via a single '**Start Switch**'. It also features connections for '**Remote Status LEDs**' which allows the Status LEDs to be mounted remotely on a Test Fixture.

The functions of the '**Remote Control Port**' are as follows:

- 3 x Remote Status LEDs
- 1 x Remote 4-wire TTL Control Port (START, BUSY, PASS, FAIL)
- 1 x Remote START signal (for Test Fixture Lid Switch)

### 1.2 Remote Control Port – connector pin-out



The pin-out for the '**Remote Control Port**' is detailed in the table below.

Pin No	Programmer Pin name	Input / Output	Notes
1 + 2	REMOTE PASS LED	P	Connect to Remote ' <b>PASS</b> ' LED: <ul style="list-style-type: none"> <li>• pin 1 - LED anode (+ve)</li> <li>• pin 2 - LED cathode (-ve)</li> </ul>
3 + 4	REMOTE BUSY LED	P	Connect to Remote ' <b>BUSY</b> ' LED: <ul style="list-style-type: none"> <li>• pin 3 – LED anode (+ve)</li> <li>• pin 4 – LED cathode (-ve)</li> </ul>
5 + 6	REMOTE FAIL LED	P	Connect to Remote ' <b>FAIL</b> ' LED: <ul style="list-style-type: none"> <li>• pin 5 - LED anode (+ve)</li> <li>• pin 6 – LED cathode (-ve)</li> </ul>
7	REMOTE START SIGNAL	I	<ul style="list-style-type: none"> <li>• This is the TTL-4-wire Remote Control Port '<b>START</b>' signal.</li> <li>• Connect to '<b>START</b>' output of the Remote System</li> </ul>
8	REMOTE JIG SWITCH	I	<ul style="list-style-type: none"> <li>• Connect to switch (normally open) on Test Fixture lid.</li> <li>• When the lid is closed this signal can be used to automatically start a programming operation.</li> </ul>

9	REMOTE FAIL SIGNAL	O	<ul style="list-style-type: none"> <li>This is the TTL-4-wire Remote Control Port '<b>FAIL</b>' signal.</li> <li>Connect to '<b>FAIL</b>' input on Remote System</li> </ul>
11	REMOTE OK SIGNAL	O	<ul style="list-style-type: none"> <li>This is the TTL-4-wire Remote Control Port '<b>OK</b>' signal</li> <li>Connect to 'OK / PASS' input on Remote System</li> </ul>
13	REMOTE BUSY SIGNAL	I	<ul style="list-style-type: none"> <li>This is the TTL-4-wire Remote Control Port '<b>OK</b>' signal</li> <li>Connect to '<b>BUSY</b>' input on Remote System</li> </ul>
10, 12, 14	GROUND (0V)	P	<ul style="list-style-type: none"> <li>Connect to GROUND on Remote System</li> </ul>

O - Output from programmer

I - Input to programmer

P - Passive e.g. GROUND and power rails

N/C - Not connected

### 1.3 Connecting Remote Status LEDs

If the programmer is to be mounted inside a Test Fixture, then it is likely that the programmer 'Status LEDs' will not be visible. It is possible to mount a set of so-called 'Remote Status LEDs' on the outside of the Test Fixture so that they are visible to the operator. These LEDs should be wired to the relevant pins of the '**Remote Control Port**' as detailed in the table below.

Pin No	Programmer Pin name	Input / Output	Notes
1 + 2	REMOTE PASS LED	P	Connect to Remote ' <b>PASS</b> ' LED: <ul style="list-style-type: none"> <li>pin 1 - LED anode (+ve)</li> <li>pin 2 - LED cathode (-ve)</li> </ul>
3 + 4	REMOTE BUSY LED	P	Connect to Remote ' <b>BUSY</b> ' LED: <ul style="list-style-type: none"> <li>pin 3 - LED anode (+ve)</li> <li>pin 4 - LED cathode (-ve)</li> </ul>
5 + 6	REMOTE FAIL LED	P	Connect to Remote ' <b>FAIL</b> ' LED: <ul style="list-style-type: none"> <li>pin 5 - LED anode (+ve)</li> <li>pin 6 - LED cathode (-ve)</li> </ul>

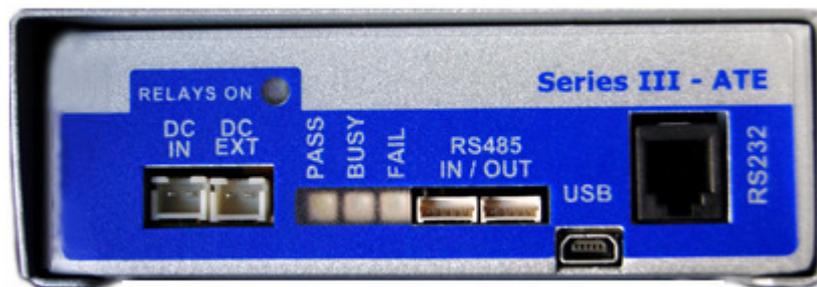
**Please note:**

The Remote LEDs can be directly connected across pins 1+2, 3+4 and 5+6 of the without using any series resistor. The programmer has a 330 ohm resistor in the anode and cathode leg on the main programmer circuit board.

## Appendix 4 – ISPnano Series 3 ATE

### 1.0 Overview

The "*ISPnano Series III ATE*" is a state-of-the-art ISP Programmer designed for high-speed In-System Programming (ISP) of FLASH Microcontrollers and Serial Memory devices in a production environment. It has been specially developed to allow simple integration with Automatic Test Equipment (ATE) systems and In-Circuit Testers (ICTs) used in high-volume production systems.



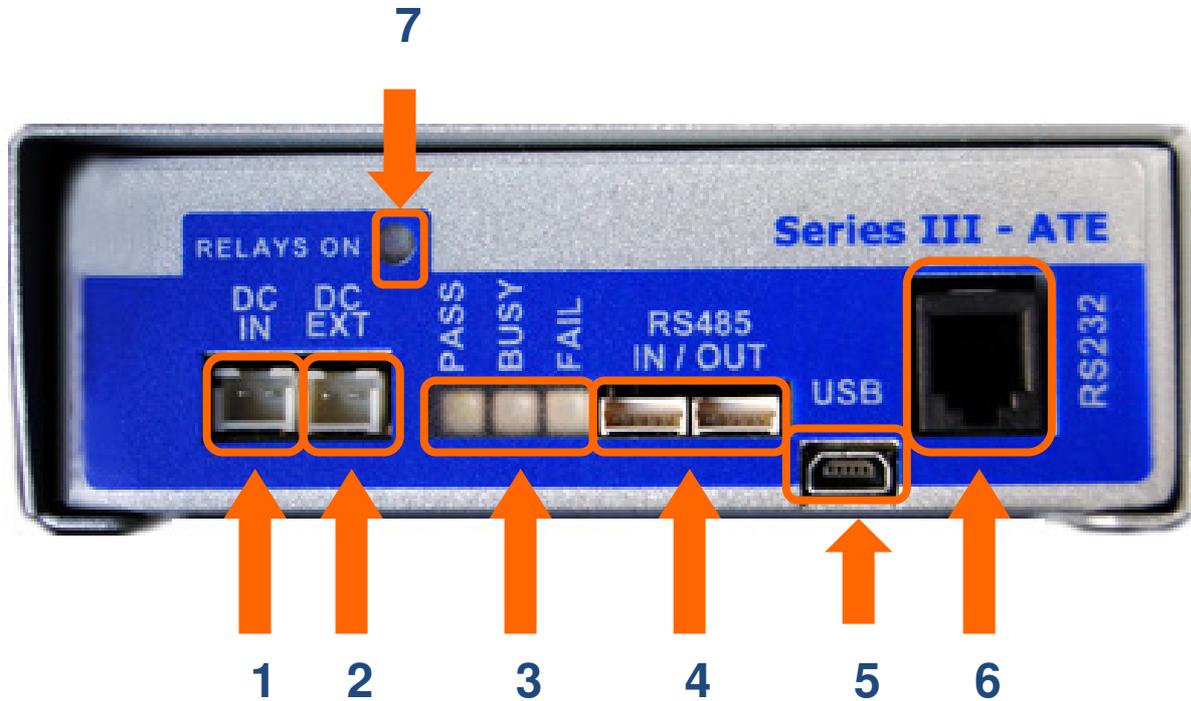
The programmer supports automated relay switching of all Target System programming and power signal lines between an external ATE / ICT system and the programmer. This allows it to be completely out-of-circuit during normal ATE testing operation. A dedicated '**ATE Interface**' also allows the ATE / ICT to control the operation of the programmer in 'Standalone Mode' (no PC required).

#### Features

- Integrated '*ISPnano Series 3*' programmer (see separate features for this module)
- Compact enclosure featuring integrated programmer and relay switching module
- All programmer I/O signal lines are isolated from both the Target System (UUT) and ATE / ICT via relays
- All power lines (Target Vcc, Target Vpp and GROUND) are independently switched via relays.
- The ATE / ICT is routed to the Target System (UUT) by default so the programmer is completely isolated from the UUT.
- When a programming operation starts, the Target System (UUT) ISP I/O and power signals are automatically switched to the programmer and the ATE / ICT is then completely isolated from the programmer.
- At the end of a programming operation, the Target System (UUT) is automatically re-connected to the ATE / ICT.
- Simple integration with ATE / ICT as the programmer controls the relay switching
- '**RELAYS ON**' indicator signals when the programmer is connected to the Target System (UUT)
- An external +12V supply is required to power the relay coils (Connect to EXT-VCC connector on the programmer)
- Supports signal routing of an external '**JTAG Boundary Scan Tester**' via the ATE Port to the Target System (UUT).

## 1.1 Front panel layout

The layout of the front panel of the programmer is shown below.



#	Connector Name	Description / comment
1	DC Power Input	<b>DC Power Input to power the programmer</b> <ul style="list-style-type: none"> <li>Voltage range: 9.0 – 24.0 V DC (right-hand pin positive)</li> </ul>
2	EXTERNAL TARGET VCC INPUT	<b>EXTERNAL TARGET VCC INPUT</b> <ul style="list-style-type: none"> <li>DC Power Input to power relay coils</li> </ul>
3	Programmer STATUS LEDs	<b>Programmer 'Status' LEDs</b> <ul style="list-style-type: none"> <li>PASS (GREEN), BUSY (YELLOW), FAIL (RED)</li> </ul>
4	RS485 Ports	<b>RS485 Ports (1) + (2) - Serial Communications Ports</b> <ul style="list-style-type: none"> <li>These ports can be used to control the programmer via an RS485 link from a PC.</li> <li>Up to 32 x ISPnano programmers can be connected on the RS485 bus.</li> <li>The connectors are connected in parallel internally so it does not matter which is used as the RS485 INPUT or OUTPUT.</li> <li>The connector is a 5-pin 2.54mm Molex.</li> </ul>
5	USB Port	<b>High-speed USB Port</b> <ul style="list-style-type: none"> <li>This port can be used to control the programmer via a USB link from a PC.</li> </ul>
6	RS232 Port or	<b>RS232 Port - Serial Communications Port</b> <ul style="list-style-type: none"> <li>This port can be used to control the programmer via an RS232</li> </ul>



	<p><b>Remote Display Keypad Port</b></p>	<p>link from a PC or other Test Equipment.</p> <ul style="list-style-type: none"> <li>• A '<b>Remote Display / Keypad Module</b>' can also be connected to this port allowing the programmer to be remote controlled.</li> <li>• The connector is a 4-pin RJ11 connector.</li> </ul>
<p>7</p>	<p><b>RELAYS ON</b></p>	<p><b>'RELAYS ON' indicator</b> When illuminated, this indicates that the relays coils are energised so the '<b>ATE port</b>' is routed to the '<b>Target System (UUT)</b>'</p>

### 1.2 Rear Panel IDC Connector Ports and signal routing

The 'ISPnano Series 3 ATE' programmer features three 'IDC Ports' on the back of the programmer as detailed in the illustration below.



The purpose of each connector is described in the table below.

Connector name	Description	Connector type
(1) 'Target ISP Port'	This port connects to the Target Device (UUT) on the Target System.	16-way IDC
(2) 'ATE ISP Port'	This port connects to the external 'ATE System' or any other control electronics which requires to be normally connected to the Target Device on the Target System (UUT).	14-way IDC
(3) 'Remote Control Port'	This port is used to remotely control the programmer from an external ATE / ICT or PLC.	14-way IDC

#### Instructions:

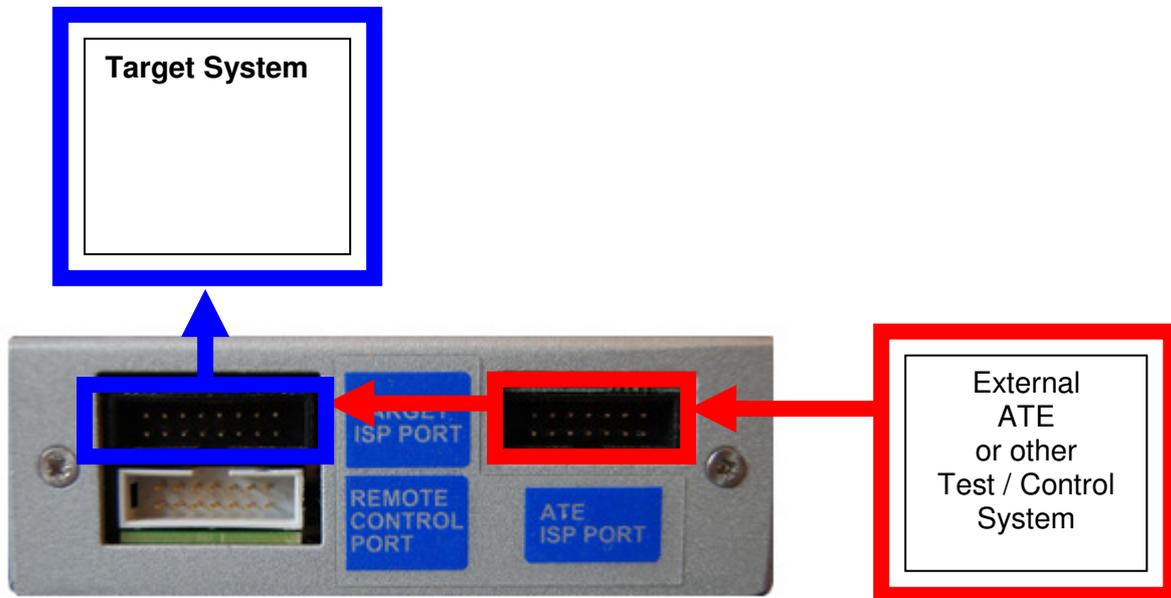
- Connect the 'Target System' (UUT) to the 16-way 'Target ISP Port'.
- Connect the external 'ATE System' or other control system to the 'ATE ISP Port'.
- If you plan to remote control the programmer, connect the relevant remote control signals to the 'Remote Control Port'.

The signal routing depends on whether the relays are energised (ON) or de-energised (OFF) – see table below.

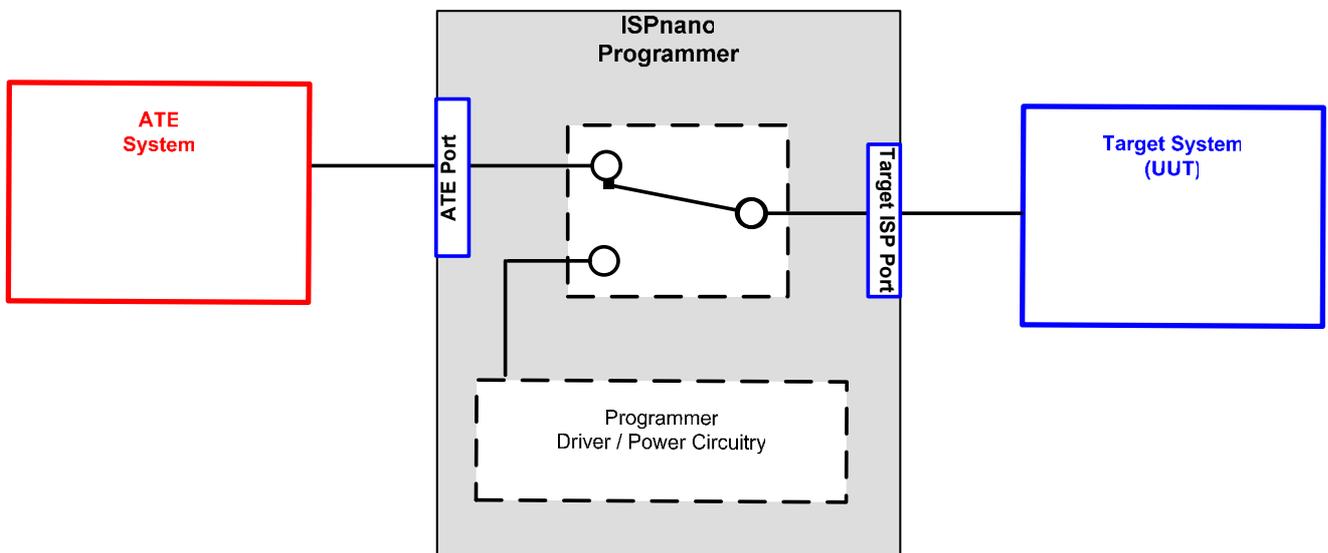
Relay position	Relays ON Indicator	Signal routing
OFF (default)	OFF	'Target ISP Port' signals are routed to the 'ATE ISP Port'
ON	ON	'Target ISP Port' signals are routed to the 'Internal Programmer Power and I/O Driver Circuitry'

### 1.3 Target System routed to External ATE System (default)

When the programmer is powered up but not running a *'Programming Project'*, the relay coils are **not** energised so the *'Target System (UUT)'* connections are routed via the *'Target ISP Port'*, through the relays to the *'Target ISP Port'* where they then connect to the *'External ATE System'*. The signal direction depends on the particular signal I/O line, GROUND or power line being routed.



The path of a single signal or passive power / GROUND line is shown in the schematic below.

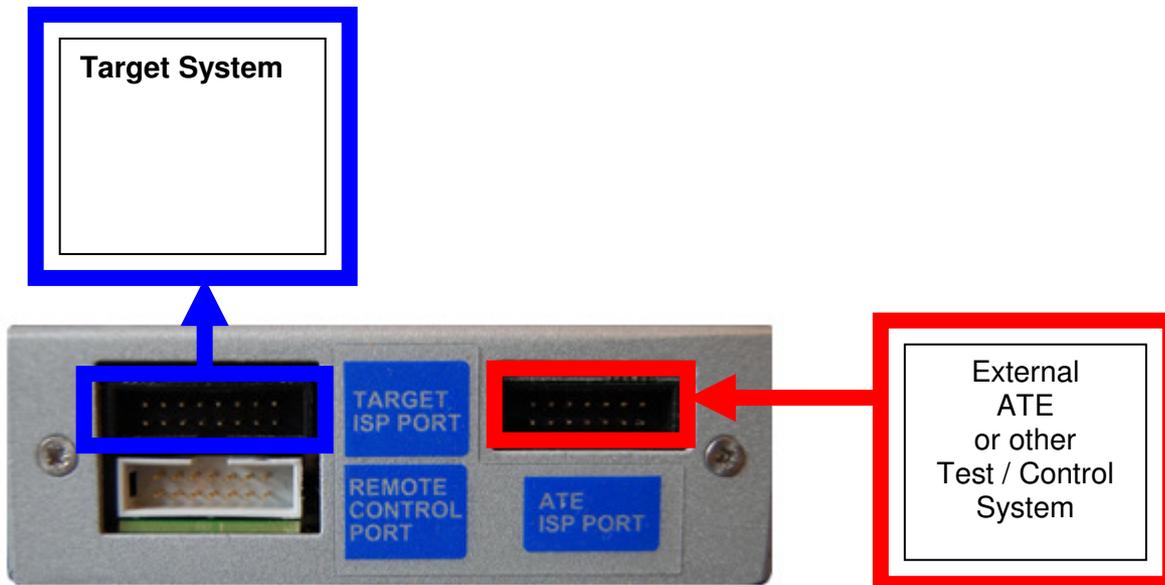


**Please note:**

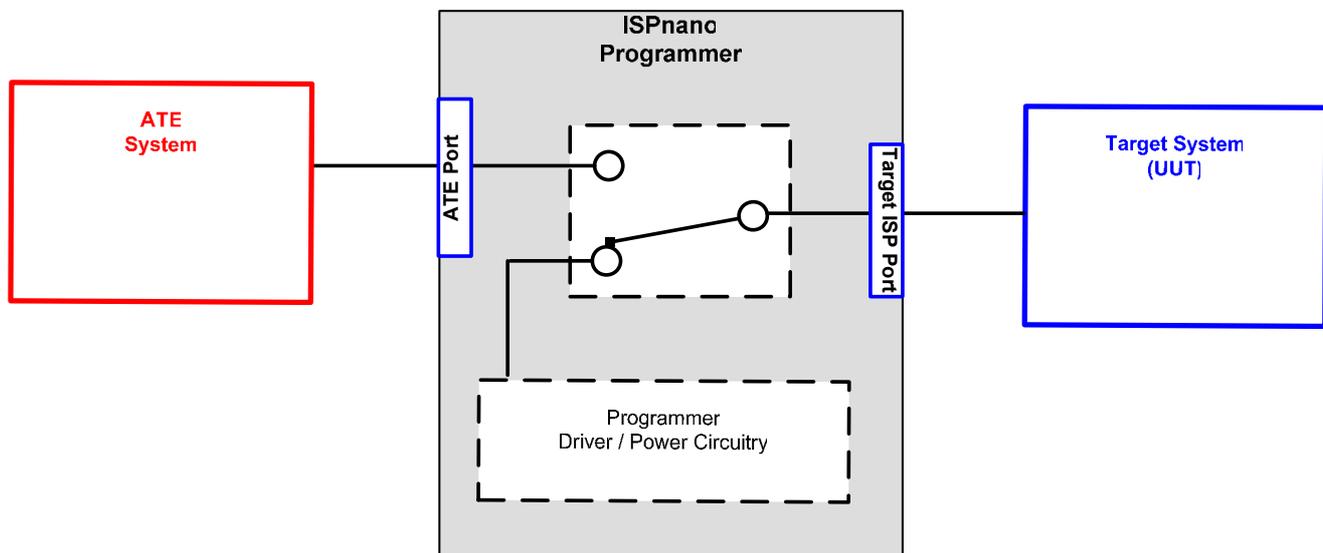
- When the relays are not energised, the programmer electronics are completely isolated from both the *'Target System'* and the *'External ATE System'*.
- This is the default state when the programmer is powered OFF or when it is powered ON and the *'EXT-VCC'* switch is OFF.

## 1.4 Target System routed to internal programmer (RELAYs ON)

When the relay coils are energised, the *'Target System (UUT)'* connections are routed to the *'Internal Programmer'*. The *'External ATE / ICT System'* is at this point completely disconnected (out of circuit). This allows the programmer to power up and program the *'Target Device'* without any problems related to the external system loading the power or programmer signal lines.

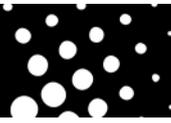


The path of a single signal or passive power / GROUND line is shown in the schematic below.

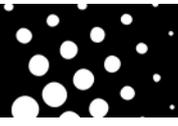


### Please note:

- When the relays are in this state, the *'External ATE System'* is completely isolated from both the *'Target System'* and the *'Internal Programmer electronics'*.
- All *'Target System'* I/O signals, power and GROUND are now routed to the internal *'Programmer Driver / Power Circuitry'*.



- The relays are energised (switched ON) by enabling the '**EXT-VCC**' switch in a Programming Project.



## 2.0 Power Supply requirements

### 2.1 Overview

The 'ISPnano Series 3 ATE' programmer requires **two** independent power supplies, one to power the '*programmer electronics / Target System*' and the other to power the '*relay coils*'.



Please refer to the table below for the details of each supply.

Designator	Description	Voltage range (Volts)	Current Requirement (mA)
(1) 'DC IN'	<b>DC Input to power the programmer</b> <ul style="list-style-type: none"> <li>Powers programmer internal electronics</li> <li>Also Powers 'Target System' (UUT) if 'Programmer controlled Power Supply' is enabled.</li> </ul>	9.0 – 24.0	150mA minimum (depends on Target System current)
(2) 'DC EXT'	<b>External DC power supply input</b> <ul style="list-style-type: none"> <li>The +12V is used to energise the ISPnano relay coils.</li> <li>This supply is switched under programmer control to the relay coils.</li> </ul>	11.5 – 12.5	250mA minimum
(3) RELAYS ON LED	<b>RELAYS ON Indicator</b> This LED illuminates when the relay coils are energised.		

## 2.2 Power Supply connections

The '*ISPnano Series 3 ATE*' programmer requires a separate +12V power supply to power the RELAY coils.



**Programmer Power Supply**  
+9.0V to +24.0V @ 150mA  
min

**RELAY Coil Power Supply**  
+12.0V @ 250mA min

### Instructions:

- Connect a +9.0V to +24.0V power supply to the '**DC IN**' connector.
- Connect a +12.0V regulated power supply to the '**DC EXT**' connector.

## 2.3 Programmer Power Supply recommendations

The '**DC-IN**' connector is used to supply power to the internal programmer electronics. If the programmer is not powering the Target System, then the programmer will only draw around 150mA @ 9.0V. However, if the '**Programmer controlled Target Power Supply**' is enabled, then the programmer will also supply power to the Target System which could take a significant amount of current. It is therefore recommended that a suitable power supply capable of supplying at least 500mA on load is used.

Although the '**DC-IN**' connector is rated for any voltage from 9.0V to 24.0V, it is recommended where possible to use 12.0V instead of 24.0V. In the event of a short circuit or other problem, the programmer over-voltage protection will better protect the internal electronics if the supply voltage is only 12.0V.

## 3.0 Programmer / ATE Signal routing

### 3.1 Overview of ports

The '*ISPnano Series 3 ATE*' programmer features three '*IDC Ports*' on the back of the programmer as detailed in the illustration below.



The purpose of each connector is described in the table below.

Connector name	Description	Connector type
(1) 'Target ISP Port'	This port connects to the Target Device (UUT) on the Target System.	16-way IDC
(2) 'ATE ISP Port'	This port connects to the external ' <i>ATE System</i> ' or any other control electronics which needs to be normally connected to the Target Device on the Target System.	14-way IDC
(3) 'Remote Control Port'		14-way IDC

#### Instructions:

- Connect the '*Target System*' (UUT) to the 16-way '**Target ISP Port**'.
- Connect the external '*ATE System*' or other control system to the '**ATE ISP Port**'.
- If you plan to remote control the programmer, connect the relevant remote control signals to the '**Remote Control Port**'.

The signal routing depends on whether the relays are energised (ON) or de-energised (OFF) – see table below.

Relay position	Relays ON Indicator	Signal routing
OFF (default)	OFF	' <i>Target ISP Port</i> ' signals are routed to the ' <b>ATE ISP Port</b> '
ON	ON	' <i>Target ISP Port</i> ' signals are routed to the ' <b>Internal Programmer Power and I/O Driver Circuitry</b> '

### 3.2 'TARGET ISP' Port

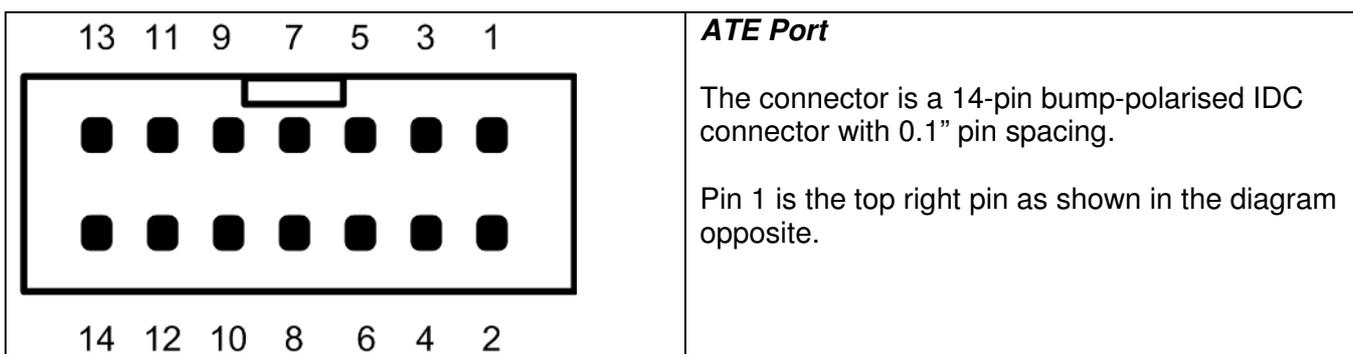
The '**TARGET ISP Port**' connects directly to the Target IC to be programmed on the Target System. It is a 16-way 0.1" pitch IDC connector. It has the same pin-out as the standard ISPnano programmers. Please see Appendix 2 for full pin-out details.

### 3.3 'Remote Control' Port

The '**Remote Control Port**' provides all the necessary signals for remote controlling the programmer. It is a 16-way 0.1" pitch IDC connector. It has the same pin-out as the standard ISPnano programmer '**Remote Control Port**'. Please see Appendix 3 for full pin-out details.

### 3.4 ATE Port

The '**ATE Port**' is only found on the '**ISPnano Series 3 (ATE)**' version of the programmer. The signals which would normally be connected from an ATE / ICT to the Target IC / Target System (UUT) are instead connected to the '**ATE Port**'. These signals are then routed via relays under programmer control to the '**TARGET ISP Port**'. The connector is a 14-pin IDC connector with 0.1" pin spacing. The pin-out of this connector is detailed below.



Pin No	Programmer Pin name	I/O	Connect to pin on Target System	Notes
1	Programmer I/O1	I/O	<ul style="list-style-type: none"> <li>JTAG – TDI</li> <li>SPI – MOSI</li> <li>UART - TXD</li> </ul>	This pin can be used for JTAG, SPI or UART Target Systems.
2	Programmer I/O2	I/O	<ul style="list-style-type: none"> <li>JTAG – TDO</li> <li>SPI – MISO</li> <li>UART – RXD</li> <li>PDI_RXD</li> </ul>	This pin can be used for JTAG, SPI, UART, XMEGA PDI and ATtiny TPI Target Systems.
3	Programmer I/O3	I/O	<ul style="list-style-type: none"> <li>JTAG – TCK</li> <li>SPI - SCK</li> </ul>	This pin can be used for JTAG or SPI Target Systems.
4	Programmer I/O4	I/O	<ul style="list-style-type: none"> <li>JTAG – TMS</li> </ul>	This pin is used for JTAG Target Systems only.

5	Programmer I/O5	I/O	<ul style="list-style-type: none"> <li>Spare I/O</li> </ul>	
6	OP6	O	Algorithm specific <ul style="list-style-type: none"> <li>PDI_BREAK</li> </ul>	<b>DO NOT USE</b>
7a	I2C_SDA	I/O	<ul style="list-style-type: none"> <li>I2C SDA</li> </ul>	<ul style="list-style-type: none"> <li>I2C SDA data signal</li> </ul>
7b	XMEGA_PDI_DATA	I/O	<ul style="list-style-type: none"> <li>TEST (PDI_DATA)</li> </ul>	<ul style="list-style-type: none"> <li>XMEGA DATA Signal</li> </ul>
7c	ATTINY_TPI_DATA	I/O	<ul style="list-style-type: none"> <li>TPI_DATA</li> </ul>	<ul style="list-style-type: none"> <li>ATtiny DATA Signal</li> </ul>
8a	I2C_SCL	I/O	<ul style="list-style-type: none"> <li>I2C SCL</li> </ul>	<ul style="list-style-type: none"> <li>I2C SCL clock signal</li> </ul>
8b	XMEGA_PDI_CLK	O	<ul style="list-style-type: none"> <li>RESET</li> </ul>	<ul style="list-style-type: none"> <li>XMEGA CLOCK Signal</li> </ul>
8c	ATTINY_TPI_CLK	O	<ul style="list-style-type: none"> <li>TPI_CLK</li> </ul>	<ul style="list-style-type: none"> <li>ATtiny CLOCK Signal</li> </ul>
9	PROG_RESET	O	<b>RESET</b>	<b>Target RESET control pin</b> This pin controls the Target Device RESET pin. It will be driven HIGH / LOW according to the device type and settings in the <Pre-program State Machine> tab in the EQtools project.
10	PROG_VPP	P	See note	<b>Vpp Voltage</b> The programmer can output a “Vpp” voltage on this pin between 6.5V and 13.8V. This pin should not be connected unless a Vpp voltage is required by the Target IC.
11 + 12	PROG_GND	P	<b>Signal GROUND (0V)</b>	<b>Signal Ground Connection (1)</b> 0V to which the programmer JTAG, SPI, I2C, PDI and TPI signal lines are referenced to.
13 + 14	TARGET_VCC	P	TARGET_VCC	<b>Target VCC</b> This pin should be connected to the Target System Vcc.

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

### 3.5 ATE Port → Target ISP Port – Pin routing (RELAYS OFF)

By default the RELAYs are **OFF** and the '**ATE Port**' signals are routed to the '**Target ISP Port**' pins as detailed in the table below.

Target ISP Port Pin Number	ATE Port Pin Number	Programmer Pin name	Notes
1 + 2	13 + 14	TARGET_VCC	<b>Target VCC</b> This pin should be connected to the Target System Vcc.
3 + 4	No connection	EXT_VCC	<b>Target External VCC</b> No connection required as EXT-VCC is being used to control the RELAY coils.
5 + 6	11 + 12	PROG_GND	<b>Signal Ground Connection (1)</b> 0V to which the programmer JTAG, SPI, I2C, PDI, TPI signal lines are referenced to.
7a 7b 7c	8a 8b 8c	<ul style="list-style-type: none"> <li>• I2C_SCL</li> <li>• PDI_CLK</li> <li>• TPI_CLK</li> </ul>	<ul style="list-style-type: none"> <li>• I2C_SCL clock signal</li> <li>• XMEGA PDI CLOCK Signal</li> <li>• ATtiny TPI CLOCK Signal</li> </ul>
8a 8b 8c	7a 7b 7c	<ul style="list-style-type: none"> <li>• I2C_SDA</li> <li>• XMEGA_PDI_DATA</li> <li>• ATTINY_TPI_DATA</li> </ul>	<ul style="list-style-type: none"> <li>• I2C_SDA data signal</li> <li>• XMEGA PDI DATA Signal</li> <li>• ATtiny TPI DATA Signal</li> </ul>
9	6	OP6	<b>Spare Output (used for PDI / TPI)</b>
10	5	Programmer I/O5	Spare I/O pin
11	4	Programmer I/O4	This pin is used for JTAG Target Systems only.
12	3	Programmer I/O3	This pin can be used for JTAG or SPI Target Systems.
13	2	Programmer I/O2	This pin can be used for JTAG, SPI, UART, XMEGA PDI and ATtiny TPI Target Systems.
14	1	Programmer I/O1	This pin can be used for JTAG, SPI or UART Target Systems.
15	10	PROG_VPP	<b>Vpp Voltage</b>
16	9	PROG_RESET	<b>Target RESET control pin</b>

## 4.0 Controlling the RELAY switching

### 4.1 Overview

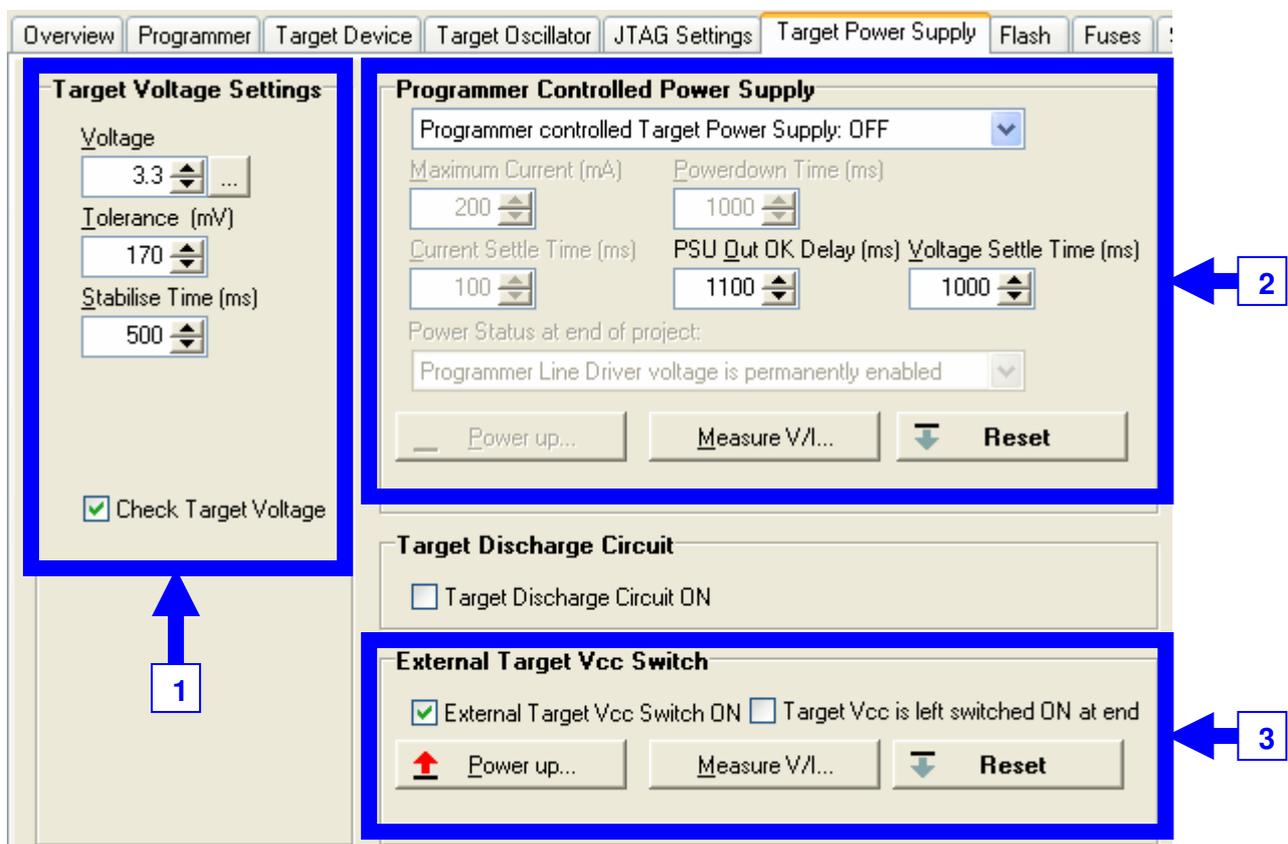
The programmer controls the +12V supply to the relay coils by enabling the electronic ‘**EXT-VCC**’ switch inside the programmer. This switch applies the voltage from the ‘**EXT-VCC**’ input (+12V) to the relay coils. This causes the relays to switch so that the internal programmer I/O and power signals are now connected to the Target System (UUT).

### 4.2 Controlling the RELAYs from EDS (Development Mode)

The simplest way to familiarise yourself with the operation of the RELAY switching is to make an ‘**EDS (Development)**’ project.

#### Instructions:

- Launch EQTools
- Select “**Create a new EDS (Development Project)**”
- Select the programmer ‘**ISPnano Series III**’
- Go through the EDS Wizard and select <Save> at the end of the wizard to save the project.
- The EDS project will now launch.
- Click on the <Target Power Supply> tab → a screen similar to the screenshot below is displayed.



The functions of the different sections of this screen are outlined in the table below.

#	Function	Further information
1	Line Driver / Target Vcc Voltage setup	<ul style="list-style-type: none"> <li>• Sets up the voltage of the programmer I/O signal lines ie. SPI, JTAG, PDI, TPI lines.</li> <li>• This voltage should be set so it is the same as the voltage at which the Target IC is powered at.</li> </ul>
2	'Programmer controlled Target Power supply' setup	<ul style="list-style-type: none"> <li>• These are the controls for the '<b>Programmer controlled Target Power supply</b>'.</li> <li>• This power supply should only be enabled if the programmer is to power the Target System (UUT).</li> </ul>
3	External Target Vcc Switch controls	<ul style="list-style-type: none"> <li>• These controls supporting switching of the '<b>EXT-VCC</b>' supply which in turn switches the relay coils ON / OFF.</li> </ul>

### 4.3 Setting the Programmer Signal Line Driver voltage

The '**Programmer Signal – Line Driver Voltage**' is the voltage level of logic '1' signal on any of the programmer I/O pins.

**Please note:**

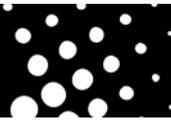
- This voltage should usually be set to the same voltage that the Target IC is actually running at.
- If the programmer is **NOT** powering the Target System but you still wish to measure / validate the Target Voltage, tick the '**Check Target Voltage**' box.

### 4.4 'Programmer controlled Target Power supply' setup

The '**Programmer controlled Target Power supply**' is a fully programmable power supply which can be used to power the Target System. It is capable of generating any voltage from 1.2 to 5.0V @ 300mA.

- Please see section 4.0 of the main User Manual for full details of how to set up this power supply.
- This power supply is routed through a relay to the '**TVCC (Target Vcc)**' pin of the '**Target ISP Port**' so it is **DISCONNECTED** from the Target System until the RELAYs are switched ON.
- When powering up the Target System using this power supply, the programmer will automatically switch the relays coils ON, wait for 50ms for the contacts to switch properly (debounce delay) and then switch the '**Programmer controlled Target Power supply**' ON. This debounce delay helps to extend the life of the relay contacts by not switching the contacts when the power is on.
- The '**PSU\_OUT\_DELAY**' is the overall time allowed for the power supply voltage and current to settle. This parameter should always be set to at least 10ms greater than the 'Voltage Settle time' and 'Current Settle time'.

**Please note:**



- This power supply does NOT have to be enabled to allow the '**EXT-VCC**' relay switch to work.

## 4.5 Target System is externally powered

If the Target System (UUT) is powered from an external power supply or via the ICT / ATE i.e. not from the programmer, then special care should be observed to prevent arcing of the contacts on the TVCC relay. As the external power supply will be switched through the '**TVCC**' pin, through a relay to either the programmer or ATE / ICT system, there may be a significant current flowing when the relay is switched on. This could cause arcing of the contacts which will reduce the lifetime of the relay.

### To prevent relay contact arcing....

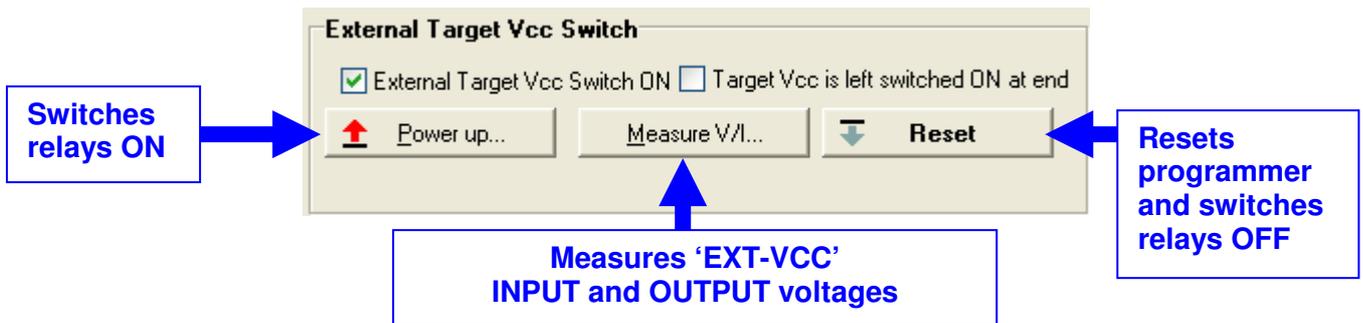
- When switching the relays ON or OFF, it is recommended that the external power supply which is powering the Target System (UUT) is switched OFF.
- If the external power supply originates from the ATE / ICT, then instruct the ATE / ICT to switch off the power supply.
- If the external power supply originates from the Target System side, use a separate relay or electronic switch to switch off the power supply.

## 4.6 Switching the RELAYs ON

The RELAYs can be switched ON by performing a **'Power-up'** operation with the **'External Target Vcc Switch'** set to ON.

### Instructions:

- Set the **'External Target Vcc Switch'** to ON.

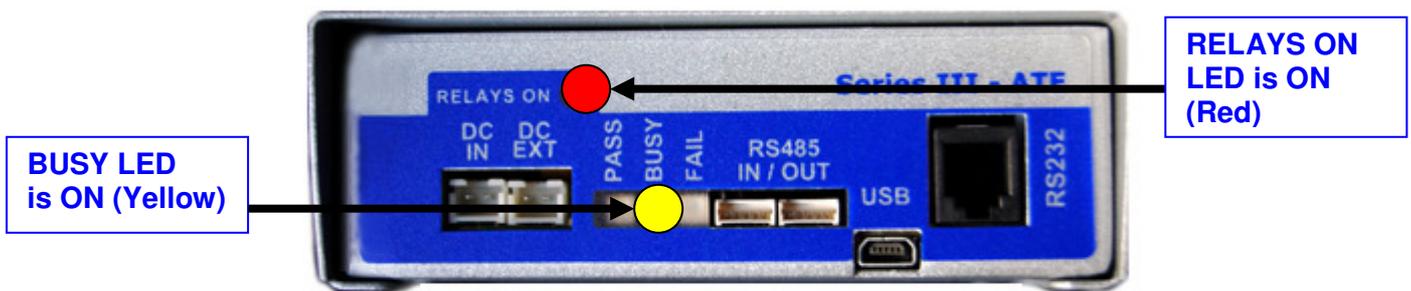


- Switch on the **'Programmer supply'** (connected to **'DC IN'** connector)
- Switch on the **'Relay coil supply'** (+12V connected to **'DC EXT'** connector)
- The programmer should power up but all **'Status LEDs'** should be OFF.
- The **'RELAYS ON'** LED should be OFF.

- Click the  button in the **'External Target Vcc Switch'** pane

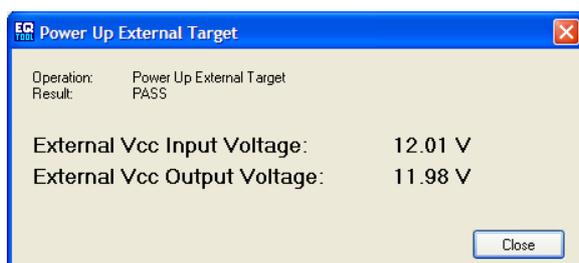
→ The +12V supply should now be connected to the relay coils.

→ The **'BUSY'** and **'RELAYS ON'** indicators should switch ON – see illustration below.



→ You should hear a click as the relay coils energise and the contacts move.

→ The following screen should be displayed.



### Please note:

- The '**External Vcc Input Voltage**' is the voltage applied to the '**DC EXT**' connector.
- The '**External Vcc Output Voltage**' is the voltage applied to the relay coils which will be slightly lower than the '**External Vcc Input Voltage**' due to the volt drop across the FET switch.
- Now that the relay coils are energised, the '**Target ISP Port**' connections are routed to the '**Internal Programmer Power / Line Driver Circuitry**' and the external ATE System is now disconnected.
- It should now be possible to power up and program a device on the Target System (UUT).

## 4.7 Switching the RELAYs OFF

The RELAYs can be switched OFF by simply performing a 'Programmer Reset'.

### Instructions:



- Click the button
- The '**EXT-VCC**' switch is switched OFF and so the relays switch OFF.
- The signals on '**TARGET ISP Port**' are now connected back to the '**ATE Port**'.
- The programmer circuitry is now completely isolated from both the ATE and Target System.

## 4.8 Leaving the RELAYs ON at the end of a project

If your programming sequence involves multiple programming steps then you probably do not want the relays to switch OFF and then ON again after each step in the sequence. This would waste time waiting for the '**Relay Debounce delay**' and would also wear the relays out quicker. To around this problem, it is possible to force the relays to stay ON at the end of '**Programming Project**' so that the next programming action does not have to switch them on again.

### To force the RELAYs to stay ON at the end of a project:

- ENABLE the '**External Target Vcc Switch**'.
- ENABLE the '**Target Vcc is left switched ON at end**' option – see illustration below.



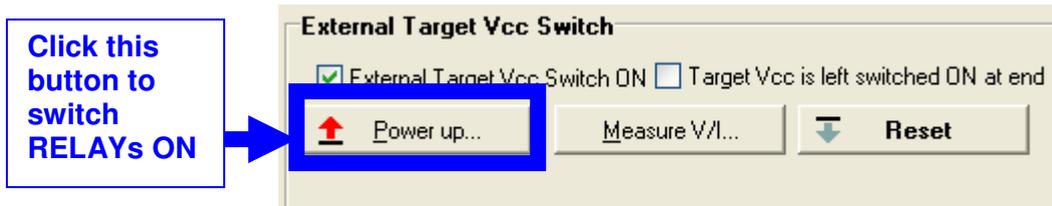
- When the project is executed either in '**EDS Mode**' or '**Standalone Mode**', the RELAYs will remain **ON** at the end of the project and will not switch OFF until a 'RESET Programmer' command is sent.

## 4.9 Measuring the Target Vcc voltage

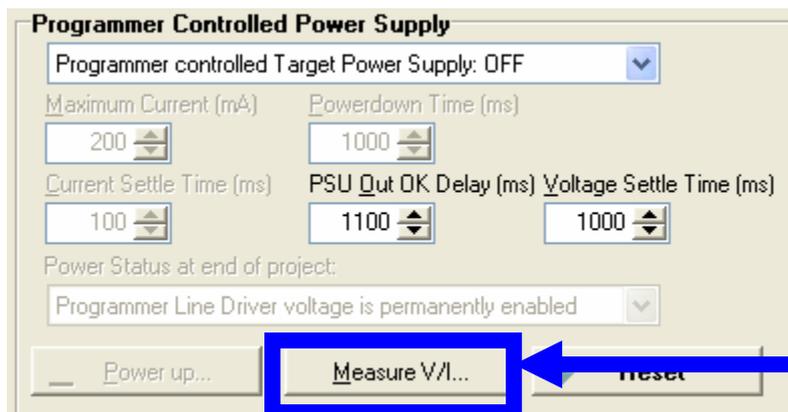
It is a good idea to measure the '**Target Vcc**' voltage using EDS as this checks that the wiring and project settings are correct.

### Instructions:

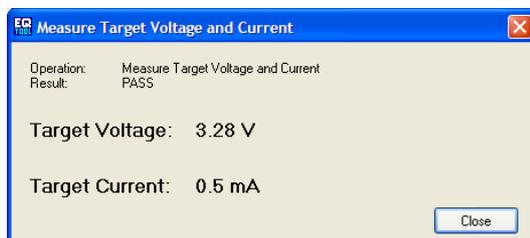
- Follow the instructions in section 4.6 to set up the RELAY control etc.



- Click the  button in the '**External Target Vcc Switch**' pane
- Click the  button to measure the '**Target Vcc**' voltage – see screenshot below.

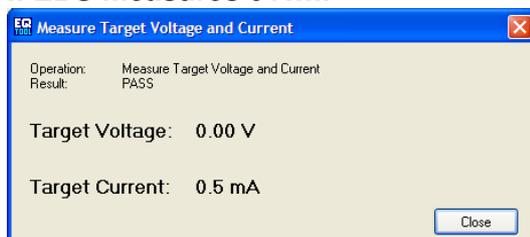


→ EDS should now report a valid voltage for '**Target Vcc**'....



- The programmer measures the voltage connected to the 'TVCC' pin on the 'Target ISP Port' connector.
- This should read back the voltage at which the Target IC is running at.

### If EDS measures 0V.....



The programmer may measure 0.00V in the following conditions:

- The RELAYs are OFF
- No Target System is connected
- There is no power on the Target System
- The 'Target Vcc' is not connected

## 4.10 Switching power via the relays

The relays on the *'Relay Module'* are rated at 100,000 switching operations at rated voltage and current.

The relays are used to switch the following power signals:

- **TARGET\_VCC** (Target Supply from programmer / ATE to Target System)
- **TARGET\_VPP** (Vpp supply from programmer to Target System)

To avoid premature wearing of the relay contacts due to contact arcing, it is recommended that all power is removed from both the ATE System and the Target System (UUT) **BEFORE** the relays in the programmer are switched ON or OFF. This prevents the relay contacts from arcing which can gradually erode the contacts.

The following procedure is recommended when switching the relays ON or OFF:

### Switching the RELAYs ON

- Make sure the external ATE / ICT has removed all power to the Target System (UUT)
- If you are using any other external power supply to power the Target System, make sure this power supply is also switched OFF.
- Instruct the programmer to commence programming.
  - The relays contacts will close.
  - The 'Programmer' is now connected to the Target System (UUT).
  - After a pre-defined delay the power will be safely applied to the Target System (UUT) from the *'Programmer controlled power supply'* (if enabled).

### Switching the RELAYs OFF

- The programmer will automatically switch the *'Programmer controlled power supply'* OFF and then wait a Debounce delay before switching the relays coils OFF.
- If you are using any other external power supply to power the Target System, make sure this power supply is switched OFF BEFORE you instruct the programmer to switch the relays OFF.
- Instruct the programmer to switch the relays OFF. (This is usually done automatically as part of a project.)
  - The relays contacts will open
  - The ATE / ICT system is now connected to the Target System (UUT) again.