

# EPSILON

Portable ISP Programmer **MKII**

*Portable ISP Programmer for  
Atmel, Atmel Wireless (Temic) and  
Philips FLASH Microcontrollers*



## User Guide



**The Embedded Solutions Company**

# Contents

<b>Copyright Information .....</b>	<b>iii</b>
<b>Equinox Warranty Information .....</b>	<b>iv</b>
<b>Electromagnetic Compatibility (EMC) Compliance .....</b>	<b>vi</b>
<b>RoHS Compliance .....</b>	<b>vi</b>
<b>Technical Support.....</b>	<b>vii</b>
<b>Product Documentation .....</b>	<b>viii</b>
<b>1.0 Programmer Overview .....</b>	<b>1</b>
1.1 System Contents .....	1
1.2 Hardware Overview (external layout) .....	2
1.3 Hardware Overview (internal layout) .....	3
1.4 Push Button Functions.....	4
1.5 Programmer Specifications .....	5
1.5.1 Programmer Specifications Overview.....	5
1.5.2 Voltage range .....	8
1.5.3 Current Consumption / Parameters .....	9
1.5.4 DC Power Input Connector (CON1).....	10
1.5.5 DC Power Cable Specification.....	10
1.5.6 J5 – RS-232 Communications Port & Serial Cables .....	11
1.6 Device Support .....	14
1.6.1 Devices supported by the programmer.....	14
1.6.2 Programming Interface to Target Device.....	15
<b>2.0 Getting Started Guide.....</b>	<b>17</b>
2.1 Overview .....	18
2.2 Hardware Installation Procedure .....	19
2.3 Software Overview and Installation .....	22
2.3.1 Software Overview .....	22
2.3.2 Programmer Control Mechanisms .....	23
2.3.3 Software Installation.....	24
2.4 Operating the programmer .....	25
2.5 Development Mode (EDS).....	27
2.5.1 Overview .....	27
2.5.2 Creating a new EDS (Development Mode) Project .....	27
2.5.3 Testing an existing Programming Project in a Project Collection in EDS (Development Mode).....	28
2.5.4 Overview of EDS – Development Mode .....	29
2.5.5 Buffer Window - Control Buttons.....	31
2.5.6 Example of using EDS .....	32
2.6 Standalone Mode.....	33
2.6.1 Overview .....	33
2.6.2 Steps required before operating the programmer in Standalone Mode .....	33
2.6.3 How to export your Development (EDS) project to Project Builder .....	33
2.6.4 Executing a Programming Project in Standalone Mode .....	34
2.7 Script Mode.....	35
2.8 ConsoleEDS – Command Line control.....	35

<b>3.0 Programmer / Target System Power Supply Scenarios + Standalone Mode</b>	
<b>Programming Instructions</b>	<b>37</b>
3.1 Overview	37
3.2 Programmer / Target System Power Supply Schematic	38
3.3 Earthing requirements	40
3.3.1 Overview	40
3.3.2 Laptop earthing issues	40
3.3.3 Desktop PC earthing issues	40
3.3.4 Earthing recommendations	41
3.4 Programmer Powers the Target System	42
3.4.1 Overview	42
3.4.2 Hardware Setup - Programmer powers the Target System	42
3.4.3 Circuit Schematic - Programmer powers the Target System	43
3.4.4 Set up instructions - Programmer powers the Target System	44
3.4.5 Programming Instructions (Standalone Mode) - Programmer powers the Target System	46
3.5 Target System powers the Programmer	48
3.5.1 Overview	48
3.5.2 Hardware Setup - Target System powers the Programmer	48
3.5.3 Circuit Schematic - Target System powers the Programmer	49
3.5.4 Setup instructions - Target System powers the Programmer	50
3.5.5 Programming Instructions (Standalone Mode) - Target System powers the programmer	51
3.6 Programmer and Target System are independently powered	53
3.6.1 Overview	53
3.6.2 Hardware Setup - Programmer and Target System independently powered	53
3.6.3 Circuit Schematic - Programmer and Target System independently powered	54
3.6.4 Set up instructions - Programmer and Target System independently powered	55
3.6.5 Programming Instructions	56
<b>4.0 ISP Header Selection</b>	<b>57</b>
4.1 Overview	57
4.2 ISP Header Selection Chart (by header)	58
4.3 ISP Header Selection Chart (by Device)	60
4.4 J3 - Atmel 6-way ISP Header (SPI Interface)	61
4.5 J6(a) - Equinox 10-way Header (Generic SPI Interface)	63
4.6 J6(b) - Equinox 10-way Header (ATtiny11/12/15 HV Interface)	66
4.7 J6(c) - Equinox 10-way Header (UART Boot Loader)	68
4.8 J7 - Atmel 10-way Header (SPI Interface)	70
4.9 J8 - Atmel 10-way JTAG Header (JTAG Interface)	72
4.10 ISP Cable considerations	74

## 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 - 2007 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.

#### Damage to the Programmer Line Driver Circuitry

This warranty does not cover damage to the programmer 'Line Driver Circuitry' due to over-voltage, over-current or short-circuit of any of the programmer I/O lines. It is the responsibility of the user to make sure that sufficient precautions are taken before plugging the ISP Cable into a Target System.

#### 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 'EPSILON5' 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 'EPSILON5' with the same care as you would these types of devices. Always ensure that you are not yourself carrying a static charge before handling the product. Wearing an earthed anti-static wrist strap is recommended.

Equinox have 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.

## RoHS Compliance

The 'EPSILON5' is a RoHS compliant product.



### 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:**

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 page for all Equinox ISP Programmers can be found at:

<http://www.equinox-tech.com/products/downloadsearch.asp>

## **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

This manual provides an overview of the contents of the EPSILON5 Programming System 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/downloadsearch.asp>

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> <ul style="list-style-type: none"> <li>• Installation and Getting Started Guide (pdf manual)</li> <li>• Help file</li> </ul>
	<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><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 for a full description.</li> </ul>
	<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 111 – ConsoleEDS Manual</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> <ul style="list-style-type: none"> <li>• Installation and User Manual</li> <li>• Help File</li> </ul>
	<ul style="list-style-type: none"> <li>• <b>Configit – Firmware Upgrade Utility</b> This utility is used to upgrade the firmware of the programmer. A firmware update may be required to add support for new devices and to correct any firmware issues.</li> </ul>
	<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 - Remote Application Control of ISP-PRO</li> </ul>
	<ul style="list-style-type: none"> <li>• <b>Remote Application Control – Application Note</b> Describes how to control the programmer using a custom Remote Application written in e.g. Visual Basic, C++, C Builder, Delphi etc.</li> </ul>
	<ul style="list-style-type: none"> <li>• <b>Error Message Descriptions</b> This document lists all the possible error messages which can be generated by the EQTools / ISP-PRO applications.</li> </ul>

## Downloading up-to-date documentation and software:

In line with our policy of continuous improvement, the software and associated documentation for this product are updated on a regular basis. Please refer to the 'Downloads Page' for this product on our website at <http://www.equinox-tech.com/products/downloadsearch.asp> for the most up-to-date information for this product.

Browse to <http://www.equinox-tech.com>

- Click / select the <Downloads> tab → the 'Downloads Search Page' will appear.
- Type in the name of the programmer e.g. EPSILON5 into the 'Product Name' field
- Click the <Search> button  
→ a list of all the software and documentation available for this product are displayed.
- Click the <Info> button next to an individual item to see the details of the downloadable file
- Click the <Download> button to download the file

## Please note:

You may be asked to register your programmer before downloading certain files. Equinox use the registration information to send you up-to-date information about new software releases etc.

## 1.0 Programmer Overview

### 1.1 System Contents



#### Hardware

- Epsilon5 MKII Programmer
- 2 x Jumper Links for internal jumpers

#### Cables

- DC Power Cable
- 9-way Female to 9-way Male Serial Cable
- 25-way Female to 9-way Male Serial Adaptor
- ISP Cables
  - 1 x 10-way IDC to 10-way IDC Ribbon Cable
  - 1 x 6-way IDC to 6-way IDC Ribbon Cable

#### Software

(Supplied on an Equinox CD-ROM)

- EQTools  
(Project Management Utility for Equinox  
Production ISP Programmers)

#### Documentation

- Epsilon5 - User Guide



## 1.2 Hardware Overview (external layout)



### Top Panel

#### ISP Cable Slot

ISP cable plugs into relevant ISP Header internally and then exits the case through this slot.



### Front Panel



#### <Target Vcc> LED

- Indicates when 'Target Vcc (Power)' is ON
- Does not indicate 'Programmer Power' unless Jumper J9 is fitted.

#### Status LED's:

- <PASS> LED
- <BUSY> LED
- <FAIL> LED



#### <NO> Button

- Used to cancel a programming operation.



#### <YES> Button

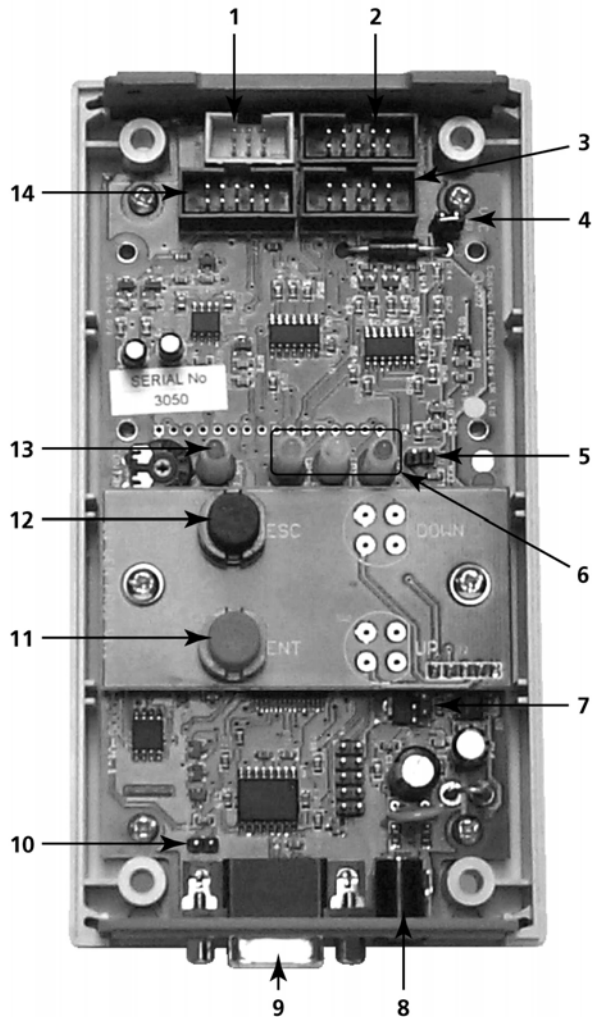
- Used to start an 'autoprogram' operation.



### Bottom Panel

1. RS-232 port (J5)
2. External DC Power Supply Input (CON1)

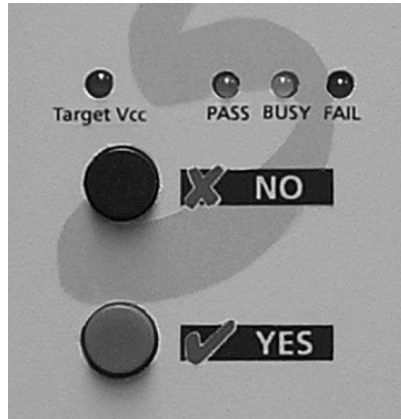
### 1.3 Hardware Overview (internal layout)



#### Hardware


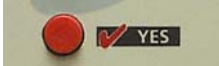
1. Atmel 6-way ISP (SPI) Header
2. Equinox 10-way ISP (SPI / UART) Header
3. Atmel 10-way ISP (SPI) Header
4. Jumper (J9) – Vcc Link
5. Jumper (J10) – Enable Target Power LED
6. Status LED's
7. Firmware Update Jumper 2 (J11)
8. External DC Power Supply Input (CON1)
9. RS-232 port (J5)
10. Firmware Update Jumper 1 (J5)
11. <Yes> button
12. <No> button
13. Target Vcc Present – LED
14. Atmel 10-way JTAG Header

## 1.4 Push Button Functions



The EPSILON5 programmer features two push buttons, **<YES>** and **<NO>**, which are used to control the programmer when it is operating in 'Standalone Mode'.

The function of these buttons is described below.

Button	Function of button
	<p>The <b>&lt;No&gt;</b> button has the following functions:</p> <ul style="list-style-type: none"> <li>i. To cancel a programming action</li> <li>ii. To return to the 'Standby' state</li> </ul>
	<p>The <b>&lt;YES&gt;</b> button has the following functions:</p> <ul style="list-style-type: none"> <li>i. To execute a programming project which has been pre-loaded into the programmer 'FLASH Memory Store'.</li> <li>ii. To return to the 'Standby' state</li> </ul>



## 1.5 Programmer Specifications

### 1.5.1 Programmer Specifications Overview

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	1.6.1
2	<b>Target Device Programming Interfaces</b>	The programmer supports the following programming interfaces: <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> </ul> NXP (Phillips) 8051 - Boot Loader ISP	1.6.2
3	<b>Operating modes</b>	The programmer supports the following operating modes: <p>As standard:</p> <ul style="list-style-type: none"> <li>• Development Mode (PC controlled)</li> <li>• Project Upload Mode using Upload Wizard (PC controlled)</li> <li>• Standalone Mode - Keypad + LCD operation</li> <li>• Standalone Mode – Run Target</li> <li>• ConsoleEDS EVAL – Command Line control</li> </ul> <p>Optional chargeable upgrades:</p> <ul style="list-style-type: none"> <li>• Standalone Mode - Remote Controlled via 'ASCII Text Communications' RS-232 Serial Protocol</li> <li>• PC controlled via ISP-PRO software (chargeable upgrade)</li> </ul> ConsoleEDS PRO – Command Line control	2.4
4	<b>On-board FLASH Memory Store</b>	2 Mbits (256 kBytes) FLASH Memory	
5	<b>Project storage in Memory Store</b>	1 x Programming Project or 2 x 'Chained Projects'	
6	<b>Keypad entry</b>	2 x Push Button - <YES> and <NO>	
7	<b>Status LED's</b>	3 x Status LED's - <PASS>, <BUSY>, <FAIL>	
8	<b>Target Vcc Voltage Detection LED</b>	<ul style="list-style-type: none"> <li>• Target Vcc (POWER) LED will light when Target Vcc is detected.</li> <li>• Threshold detection voltage: approx 3.1V</li> </ul>	



9	<b>PC Control Software</b>	The programmer can be controlled using: <ul style="list-style-type: none"> <li>EQTools (as standard)</li> <li>ISP-PRO (chargeable upgrade)</li> <li>Standalone Upload Wizard (chargeable upgrade)</li> <li>ConsoleEDS Evaluation version</li> <li>ConsoleEDS PRO (chargeable upgrade)</li> <li>ASCII Text Mode (chargeable upgrade)</li> </ul>	2.3
10	<b>PC requirements</b>	PC running Windows 95 / 98 / 2000 or ME / NT4 / XP	
11	<b>Power Supply Options</b>	There are 3 possible modes for powering the Programmer / Target System: <ul style="list-style-type: none"> <li>Programmer powers Target System</li> <li>Target System powers the Programmer</li> <li>Programmer and Target System are independently powered</li> </ul>	3.0
12	<b>Voltage range</b>	See specified section	1.5.2
13	<b>Vpp Voltage Generator</b>	On Board +12.0V Vpp Voltage Generator (Applies +12.0V to RESET pin for ATtiny HV programming)	
14	<b>DC Power Connector (CON1)</b>	<ul style="list-style-type: none"> <li>DC Power Connector</li> <li>2.5 mm Jack Socket – Centre Positive</li> <li>DC Power Lead supplied with programmer – white stripe is +ve.</li> </ul>	1.5.2
15	<b>Programmer Current Consumption</b>	This refers to the current consumption of the actual programmer circuitry. See specified section.	1.5.3
16	<b>Max. current through programmer</b>	This refers to the maximum current which can be supplied through the programmer from CON1 (i.e. external power supply) to the Target System. Max current = 300 mA.	1.5.3
17	<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 (requires firmware 3.04 or greater):</b></p> <ul style="list-style-type: none"> <li>SLOW SPI: 115.2 kHz to 921.6 kHz (user selectable)</li> <li>FAST SPI: 115.2 kHz to 921.6 kHz (user selectable)</li> </ul>	
18	<b>Target UART Speed</b>	This is the Communications Speed (BAUD rate) from the programmer to the Target Device when communicating via a Boot Loader. The BAUD rate is configurable from 1,200 to 115 kBaud. This settings is completely independent from the BAUD rate settings for the PC communicating with the programmer.	
			4.0

19	<b>ISP Headers</b>	4 x ISP Headers (inside enclosure) i. Atmel 10-way (SPI) ii. Atmel 6-way (SPI) iii. Atmel 10-way (JTAG) iv. Equinox 10-way (SPI+UART)	
20	<b>Frequency generator output (SCK2)</b>	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.	
21	<b>PC Connection (J5)</b>	<ul style="list-style-type: none"> <li>• RS232 9-way Female D Connector</li> <li>• Connects to spare PC COM port</li> <li>• All 9 connections must be made in serial cable.</li> <li>• Baud rate: 38,400 (fixed)</li> </ul>	1.5.6
22	<b>Temperature range</b>	0 to 50 deg. C	
23	<b>Size (enclosure)</b>	189H x 80L x 31D mm	
24	<b>Shipped Weight</b>	0.75 kg	

## 1.5.2 Voltage range

The programmer is designed to operate from a supply voltage in the range defined in the table below. Correct operation outside of these limits is not guaranteed. The programmer supply voltage can be derived from either the **CON1 connector** or the **Target Vcc** pin on any of the ISP Headers. The CON1 centre pin and the Target Vcc are physically joined together inside the programmer if the jumper J9 is fitted.

Fig. 1.5.2 Programmer voltage supply range for reliable operation

#	Programmer mode	Min Vcc	Max Vcc	Units
1	Programming a Target Device	3.05	5.00	V
2	Uploading Programming Projects to programmer FLASH	3.10	5.00	V

### Important note:

The programmer electronics should operate reliably between the 'Min Vcc' and 'Max Vcc' voltage levels specified in the table above. This does NOT mean that every Target Device can also be programmed within this voltage range.

To verify that a particular device can be programmed at a certain Vcc voltage:

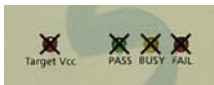



- Check the supply / programming voltage parameters for the actual device (IC) in the manufacturer's datasheet.
- Many devices have two different voltage range versions eg. 2.7 – 6.0V and 4.0 – 6.0V. Please ensure that you are programming the device within the stated voltage range.
- Ensure that you have selected the required Target Vcc Voltage in your EQTools – Programming Project (\*.ppm).  
e.g. if you are programming at +3.3V, you must set up this voltage in your Programming Project. This can be found in the EQTools - Project Builder <Target Vcc> tab.
- For most Atmel AVR and AT89S devices, the FLASH and EEPROM Write Times increase as the Target Voltage decreases. The timings are automatically set for you in your project as long as you have declared the correct Target Vcc voltage in your project.

## 1.5.3 Current Consumption / Parameters

### 1.5.3.1 Current Consumption

The programmer can either be powered from the Target System via one of the ISP Headers or from an independent power supply via CON1. The table below details the programmer current consumption with a supply voltage of 5.0V and 3.3V for both 'Standby' mode and during a typical programming operation when the programmer is independently powered. If the programmer is to be powered from the Target System (typical of field use), then the Target System power supply must be capable of supplying the maximum current that can be drawn by the programmer during a programming operation.

Fig. 1.5.3.1 Programmer voltage supply range for reliable operation (independently powered)

#	Programmer mode	LED Status	CON1 Vcc (V)	Min	Typ	Max	Units
1	Standby Mode		5.0	-	28.0	35.0	mA
2	Standby Mode		3.3	-	28.0	35.0	mA
3	Programming (BUSY)		5.0	-	47.0	60.0	mA
4	Programming (BUSY)		3.3	-	44.0	60.0	mA

#### Status LED Key:

ON FLASHING OFF  


#### Notes:

The above measurements were taken with the programmer and Target System independently powered (i.e. J9 removed). EPSILON BOM – 1.20/1.21. Target device – AT90S8515.

These measurements refer to the current taken by the programmer's internal circuitry.

Standby Mode – All LED's OFF (Target Vcc is OFF so the Target Vcc LED is also OFF)

### 1.5.3.2 Maximum Current handling of programmer

If the programmer is supplying power to the Target System, then the current drawn by the Target System must be below the maximum value stated in the table below.

Fig. 1.5.3.2 Programmer – maximum current ratings

#	Programmer mode	Comments	Typ	Max	Units
1	Programmer supplying power to the Target System	<ul style="list-style-type: none"> <li>Power is supplied from External Power Supply via CON1</li> <li>J9 is fitted connecting the CON1 Vcc to the Target Vcc</li> </ul>	-	300	mA

## 1.5.4 DC Power Input Connector (CON1)

It is possible to power the programmer from an external power supply by plugging the DC Power Cable supplied with the programmer into CON1. This connector is a 2.5mm jack socket.

Fig. 1.5.4 CON1 – pin-out

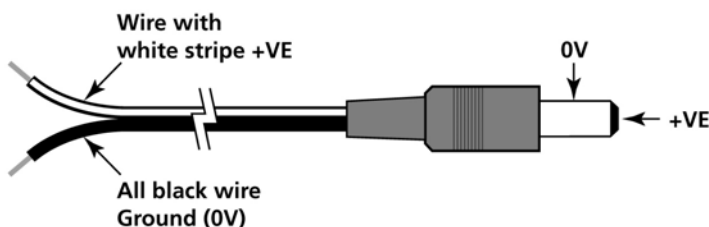
Description	Voltage
Centre pin	+3.1V to +5.0V from external power supply
Outer	0V (Ground connection)

## 1.5.5 DC Power Cable Specification

The programmer is supplied with a DC power cable which plugs into CON1. The other end of this cable is supplied with bare wire ends for connecting to an external power supply. e.g. bench PSU.

### Cable Specification:

- One end features a female 2.5mm DC jack plug
- The other end of the cable is supplied with bare wire ends for connecting to an external power supply. e.g. bench PSU.
- The black lead with the white stripe is the +VE cable.
- The all black lead is the GROUND cable.
- The external voltage applied to CON1 must be between 3.1 and 5.0V !



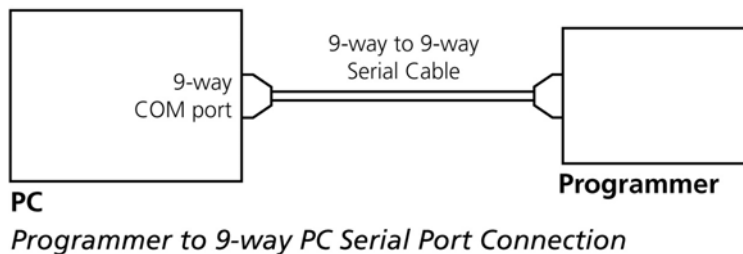
## 1.5.6 J5 – RS-232 Communications Port & Serial Cables

### 1.5.6.1 Connecting the programmer to the PC COM port

The EPSILON5 programmer communicates with a PC via the RS-232 Communications Port (J5). A suitable 9-way to 9-way serial cable is supplied with the programmer.

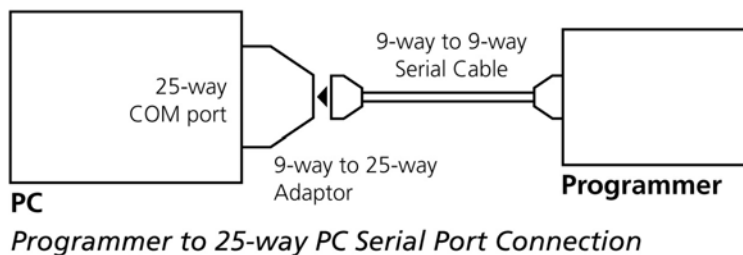
The programmer is supplied with the following PC Serial Cables / Adaptors:

#### 1. 9-way D (female) to 9-way D (Male) Serial Cable



This cable should be connected between the 9-way D-connector on the programmer and a spare 9-way RS-232 COM port on your PC.

#### 2. 9-way D (female) to 25-way D (Female) adaptor



If your PC has a 25-way D connector, you will need to plug the 9-25 way adaptor onto the end of the 9-9 way Serial Cable as shown in the diagram below.

### 1.5.6.2 Serial Cable Length

The serial cable supplied with the programmer is approximately 2m in length. If you wish to make your own longer cable, please ensure that you connect all 9 pins at each end in a straight-through pin-to-pin fashion. See fig. 1.5.6.2 for the pin-out specifications. The programmer features an ESD protected full RS-232 specification driver chipset and so should operate reliably when driving serial cables up to 10m in length. However, it is recommended to keep the serial cable as short as possible to enhance EMI immunity and reduce EMI emissions.

#### **Important note:**

The programmer uses ALL of the NINE pins of the COM port for different purposes. It is therefore imperative that the cable used to connect between the programmer and the PC COM port has all NINE cables connected. Failure to do this will result in the programmer not operating correctly.

## 1.5.6.3 Serial Cable Pin-outs

Fig. 1.5.6.1 Pin-out of J5 - RS-232 Communications Port

Pin No.	RS232 pin	Description
1	DCD	Not Connected
2	RXD	Receive
3	TXD	Transmit
4	DTR	Not Connected
5	0 Volt	0V
6	DSR	Not Connected
7	RTS	Not Connected
8	CTS	Not Connected
9	RING	Not Connected

Fig. 1.5.6.2 Connections for 9-way to 9-way Serial Cable (Straight through cable)

9-way (Male)	9-way (Female)
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9

Fig. 1.5.6.3 Connections for 9-way to 25-way Adaptor

25-way (Female)	9-way (Male)
8	1
3	2
2	3
20	4
7	5
6	6
4	7
5	8
22	9





## 1.6 Device Support

### 1.6.1 Devices supported by the programmer

The programmer supports the following devices:

#	Silicon vendor	Family	Programming algorithm	Device support	
1	Atmel	AT89S	LV SPI	- AT89(L)S8252/53 - AT89(L)S51/52 (including 'L' versions)  - AT89S 2051 / 4051 - AT89S8253	S
2	Atmel	AT90S (AVR)	LV SPI	- AT90(L)S1200(A) / 2313 / 2333 / 4433 / 2323 / 2343 / 4414 / 8515 / 4434 / 8535 (including 'L' versions)	S
3	Atmel	AT90CAN	LV SPI	AT90CAN128 / AT90CAN64 / AT90CAN32	S
4	Atmel	AT90CAN	JTAG (AVR)	AT90CAN128 / AT90CAN64 / AT90CAN32	U
5	Atmel	AT90USB	LV SPI	AT90USB1286/ AT90USB1287/ AT90USB162/ AT90USB646/ AT90USB647/ AT90USB82	S
6	Atmel	AT90USB	JTAG (AVR)	AT90USB1286, AT90USB1287, AT90USB646, AT90USB647	U
7	Atmel	ATmega (AVR)	LV SPI	- ATmega 8 / 16 / 32 / 48 / 88 / 64 / 103 / 128 / 1280 / 1281 / 168 / 169(V) / 169PV / 323 / 325 / 329 / 2560 / 2561 2561V / 8515 / 8535 (including 'L' versions)	S
8	Atmel	ATmega (AVR)	JTAG (AVR)	ATmega 8515 / 8535 / 32 / 64 / 128 / 1280 / 1281 / 161 / 162 / 163 / 169 / 323 / 325 / 329 / 3290(V) 406 / 640(V) / 2560 / 2561 (including 'L' versions)	U
9	Atmel	ATtiny (AVR)	LV SPI	ATtiny 12 / 13 / 15L / 22 / 24* / 25 / 26 / 45 / 85 / 2313 (including 'L' versions)	S
10	Atmel	ATtiny (AVR)	HV SPI (+12V VPP)	ATtiny 11 / 12 / 15L (including 'L' versions – Vcc must be +5V with a +12V Vpp applied to the RESET pin) HV ISP support for the ATtiny13, 25, 45 etc is not currently available.	S
11	Atmel (Temic)	T89C51Rx2	UART Boot Loader	T89C51 RB2 / RC2 / RD2 T89C51 IB2 / IC2 T89C51CC01 (Support for the AT89Cxxxx devices is not currently available)	S
12	NXP (Philips)	T89C51Rx2	UART Boot Loader	P89C51 RB2 / RC2 / RD2 P89C 660 / 662 / 664 / 668	S
13	Zensys	ZW series	LV SPI	ZW0102 / ZW0201 / ZW0301	S

**Key:**

S – Device supported as standard

U – Chargeable license upgrade required

\* – Device support not available at the time of this document going to press. Please contact Equinox for a device support update.

## 1.6.2 Programming Interface to Target Device

The programmer supports the following programming interfaces to the Target Device:

#	Programming Interface	Family	Description / Comment
1	LV SPI	AT89S AT90S AT90USB AT90CAN ATmega ATtiny Zensys	<b>Atmel Low Voltage Serial Programming Mode</b> <ul style="list-style-type: none"> <li>• Atmel 3-wire SPI interface (SCK, MOSI, MISO )</li> <li>• Target Device RESET</li> <li>• Works at target voltage (no Vpp required)</li> </ul>
2	HV SPI (+12V Vpp)	ATtiny	<b>Atmel High Voltage Serial Programming Mode</b> <ul style="list-style-type: none"> <li>• Atmel 4-wire SPI Interface (SCK, MOSI, MISO, MOSI2)</li> <li>• +12V Vpp is applied to the RESET pin to enter High Voltage Serial Programming Mode</li> <li>• Requires different connections to Target Device compared with LV SPI.</li> <li>• Requires all pins of target device to be accessible by the programmer.</li> </ul>
3	UART Boot Loader	T89C51Rx2 P89X51Rx2	<b>Atmel / Philips UART Boot Loader Interface</b> <ul style="list-style-type: none"> <li>• Uses RXD, TXD, PSEN, and Target Device RESET pin</li> <li>• TTL Levels from programmer to Target System</li> </ul>
4	JTAG (Atmel AVR)	ATmega AT90USB AT90CAN	<b>Atmel ATmega AVR – JTAG ISP Interface</b> <ul style="list-style-type: none"> <li>• Uses same connector as Atmel JTAG ICE</li> <li>• A chargeable license upgrade is required to enable the ATmega JTAG ISP libraries.</li> </ul>

Please refer to section 4 for information on which ISP Header to use for the required programming interface.



## 2.0 Getting Started Guide

*EPSILON5 MKII Programmer - User Guide V1.12 – 1<sup>st</sup> August 2007*

## 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 instructions on how to actually use the programmer.

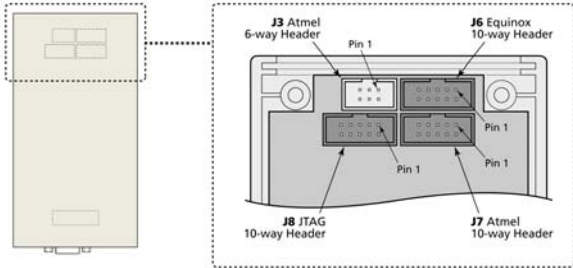
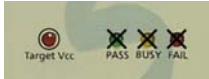

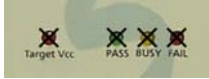
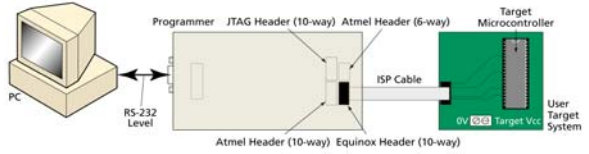
#	Description	Refer to section
1	Hardware Installation Procedure	2.2
2	Software Installation Overview	2.3
3	Operating the programmer	2.4
4	Development Mode (EDS)	2.5
5	Standalone Mode	2.6
6	Script Mode	2.7
7	ConsoleEDS	2.8


Further help and instructions can be found in the 'EQTools – Getting Started Guide' (pdf manual) and also in the '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:

#	Action	Observation
1	Remove programmer from all packaging	<p>You will need the following items:</p> <ul style="list-style-type: none"> <li>• Epsilon5 programmer</li> <li>• 10-way ISP Ribbon Cable</li> <li>• Jumper Link</li> </ul> <p>You will find a 'Jumper Link' and the plastic feet for the programmer inside a small plastic bag.</p>
2	Ensure you are wearing an ESD strap and/or you are working in an ESD protected environment.	
3	Remove all four screws from the back of the programmer	The screws are supplied in a small plastic bag when you open the programmer packaging for the first time.
4	Remove programmer cover	
5	<p>Select Power Supply Scenario</p> <p>Please refer to section 3 for a full explanation of which is the correct way of powering the programmer for your application.</p>	<p>You need to select how the programmer is to be powered. The programmer can be powered from the Target System or from an External Power Supply via the DC Jack Connector. The power supply selection is set up via the Jumper Link J9.</p> <p><b>Where is Jumper Link J9?</b> You can find Jumper Link J9 next to the 'J7 Atmel 10-way ISP Header'</p> <p><b>Jumper J9 Fitted:</b></p> <ul style="list-style-type: none"> <li>• Target Vcc is connected directly to the Programmer CON1 Vcc (same as plugging in the Jack Socket).</li> <li>• In this mode, the programmer will power from the Target system or the Target System can power the Programmer.</li> <li>• The supply voltage must be between 3.0 and 5.0V.</li> </ul> <p><b>Jumper J9 Not Fitted:</b></p> <ul style="list-style-type: none"> <li>• Target Vcc is independent (not connected) to the programmer Vcc.</li> <li>• In this mode you will need to supply power at 3.0 – 5.0V via the DC Jack Connector (CON1)</li> </ul>
6	Check J5 – Firmware Update Jumper - is NOT fitted	<ul style="list-style-type: none"> <li>• This jumper can be found next to the 9-way D connector.</li> <li>• It must NOT be fitted during normal operation of the programmer.</li> <li>• It should only be fitted when the</li> </ul>

		programmer firmware is being updated.
7	<p>Select the required ISP Header and then plug the ISP Cable into this Header.</p> <p>Please refer to section 4.</p>	<p><b>!!! WARNING !!!</b></p> <ul style="list-style-type: none"> <li>If you select the wrong ISP Header then it is possible that you may damage both the programmer and Target System when power is switched on.</li> <li>Please ensure you are using the correct header. If in doubt, please refer to section 4 of this manual and / or contact Equinox Support for assistance.</li> </ul> 
8	<p>Apply power to the programmer</p> <p>Connect 3.0-5.0V to CON1 via supplied lead (white stripe is +ve) or Fit J9 and connect ISP Cable to powered Target System</p> <p>Please refer to section 3 for further details.</p>	<p>→ Programmer circuitry powers up to 3.0V.</p> <p>→ If the Jumper Link J9 is fitted then the 'Target Vcc' LED will illuminate.</p> 
9	Remove power from programmer	<p>→ All LED's go out</p> 
10	Re-fit lid to programmer	The ISP Cable should fit through the slot in the end panel.
11	Re-fit the four screws in the back of the case	
12	Ensure that the PC, programmer and Target System power is switched OFF.	
13	<p>Connect the PC serial cable to the programmer RS-232 port (J5) and connect the other end of the cable to a spare PC COM port.</p> <p>See section 1.5.6 for Serial Cable specifications.</p>	
14	Connect the ISP cable to the ISP Header on your Target System	

15	Switch PC power ON	
16	Switch Programmer / Target System power ON	→ Target Vcc LED illuminates 
17	Hardware installation is now complete.	

### Electrostatic Discharge (ESD) Warning!

There are a number of electrostatic issues which must be taken into account to avoid damage to the programmer and/or the Target System during In-System Programming. Please see section 3.3.



## 2.3 Software Overview and Installation

### 2.3.1 Software Overview

The Epsilon programmer is supplied with a comprehensive Integrated Development Environment (IDE) called 'EQTools Version 2.xx Software Suite'. This software supports creation and testing of so-called 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 Epsilon5 Programmer

EQTools - Project Builder	YES
EQTools - Project Management Utility	YES
EQTools - Project Upload/Download Utility	YES
EQTools - Development Suite (EDS)	YES
EQTools - Script Builder	U
EQTools - Script Editor	U
Development Mode (PC Controlled)	YES
Standalone operation (Keypad – no PC)	YES
ASCII Text Mode Control	U
ConsoleEDS – Evaluation version	YES
ConsoleEDS – PRO version	U
<b>ISP-PRO Software</b>	U
- 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 from EQTools, from a Remote System and via the Equinox ISP-PRO software. An overview of the various control mechanisms is given below.

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

This indicates that the programmer can be operated without being connected to a PC or other control system. In this case an operator can control programming operations using a keypad on the programmer e.g. Epsilon5, FS2003, PPM3-MK2

#### **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.

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

This software suite is capable of controlling up to 16 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.

### 2.3.3 Software Installation

The Epsilon5 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.

Locate the latest version of the EQTools software by browsing to:  
<http://www.equinox-tech.com> and clicking on the <downloads> tab.

EQTools versions are quoted as e.g. **EQTools V2.1.0 Build 412**  
The filename for this version would be '**eqtools210\_412.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. Refer to Section 2 of the guide for detailed EQTools installation instructions.

## 2.4 Operating the programmer

The EPSILON5 programmer can be used in the following Operating Modes as detailed in the table below:

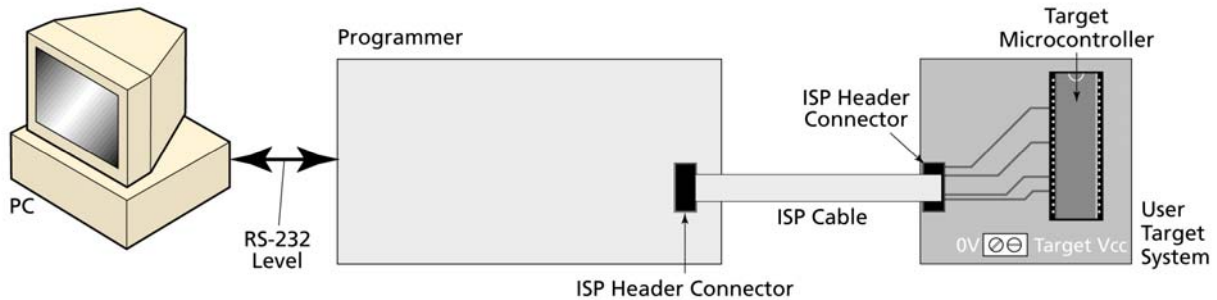
#	Programmer Mode	Connect to PC COM Port	Control Method / Software	Upgrade License Required	Functional description	Refer to section
1	<b>Development Mode</b>	Yes	EQTools - EDS	Standard Feature	In this mode, the programmer is controlled by EQTools - Equinox Development Suite (EDS). EDS allows you to read/write 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 Mode</b>	No	None	Standard Feature	In this mode, the programmer is not connected to a PC and can program a single 'Programming Project' into the Target Device when the <YES> key is pressed. (A valid project must have been previously uploaded to the programmer using EQTools – Upload Wizard.)	2.6
3	<b>Script Mode</b>	Yes	EQTools – Script Debugger or ISP-PRO	Upgrade Required	In this mode, the programmer is controlled from a 'Programming Script' running within either the EQTools or ISP-PRO applications on a PC.	2.7
4	<b>ConsoleEDS</b>	Yes	ConsoleEDS	Upgrade Required	In this mode, the programmer is controlled via the ConsoleEDS utility which allows control via simple Command Line commands and arguments.	See App Note AN111

5	<b>ASCII Control Mode</b>	Yes	ASCII Text Communications	Upgrade Required	In this mode, the programmer is controlled via a simple ASCII Protocol. A simple Terminal Emulator is required to test out this mode or the ASCII commands can be generated by an external controller.	See App Note AN110
---	---------------------------	-----	---------------------------	------------------	--	--------------------

## 2.5 Development Mode (EDS)

### 2.5.1 Overview

In 'Development Mode', the programmer is controlled from the EQTools – EDS (Equinox Development Suite) running on a PC. The Development utility is called Equinox Development Suite or EDS for short. In this mode it is NOT necessary to upload the 'Programming Project' to the programmer as the EDS utility co-ordinates all programming operations.



In EDS Mode, you can perform the following actions under PC control:

- Write data directly to a Target Device from the PC e.g. a HEX file
- Read data from a Target Device to a file on the PC
- Write / Read the Configuration Fuses of a Target Device
- Write / Read the Security Fuses of a Target Device
- Read the Target Vcc Voltage

### 2.5.2 Creating a new EDS (Development Mode) Project

To test a Target Chip in EDS (Development Mode), follow the instructions detailed below:

- Launch EQTools
- From the Welcome Screen, select the '**Create a new Equinox Development Project (EDS)**' option.
- OR
- From the left-hand pane, select the **<Development>** tab and then select '**New Development Project**'.

→ The EDS Wizard will now launch

- Follow the EDS Wizard by filling in each tab and then clicking the **<Next>** button:
  - Select Programmer – click **<Get Info>** button to detect your attached programmer
  - Select Project Type – leave as the default 'Standalone Keypad Operation'
  - Select Device – This is your Target Chip.
  - Select Target Oscillator – If unsure of settings leave all settings as the defaults and click **<Next>**.
  - Select Target Voltage
  - Select FLASH Data File (optional – you can enter the file name at a later stage in EDS)
  - Select EEPROM Data File (optional – you can enter the file name at a later stage in EDS)

- At the end of the EDS Wizard:
  - Click the <Test> button
  - Save the EDS file with an appropriate name eg. ATmega163.eds
  - The EDS Window will now launch – see section 2.5.4

### **2.5.3 Testing an existing Programming Project in a Project Collection in EDS (Development Mode)**

If you have already created a Programming Project and want to test it in EDS (Development Mode), please follow the steps below:

- From the left-hand pane, select **<Project Manager>**
- Select **<Open Collection>**
- Browse to and select a Project Collection (\*.ppc) file
- Project Collection File launches in **Project Manager** view
- Select the Programming Project you wish to test in EDS
- Click the **<Test Project in EDS>** button at the bottom right of the **Project Manager** window
- → The EDS window now launches with all the settings of the select Programming Project – see section 2.5.4

## 2.5.4 Overview of EDS – Development Mode

The EDS Utility consists of a series of tabs. Each tab controls a different area of functionality including FLASH Area Programming, EEPROM Area programming, Configuration Fuse Programming and Security Fuse programming. It is also possible to set up the SPI / JTAG / UART Target Speeds and the Power Supply Settings from EDS.

Fig. 2.5.4.1 EDS – FLASH Tab Overview

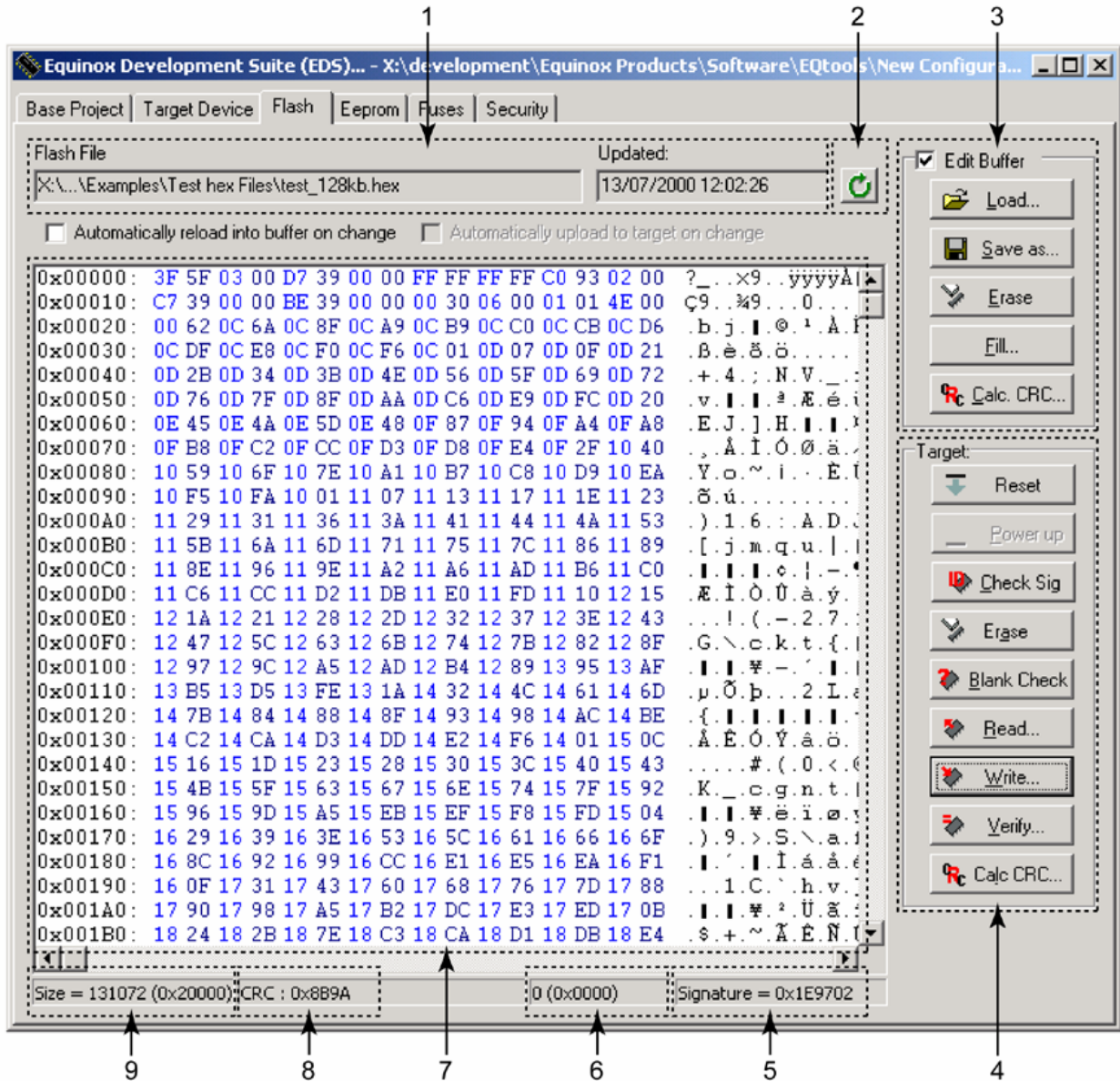






Fig. 2.5.4.2 FLASH / EEPROM tab – functional description

#		
1	<b>File name and properties</b>	<p>This specifies the 'File Name / path' and 'Last updated' date of the file to be loaded into the Buffer Window.</p> <ul style="list-style-type: none"> <li>• By default, this will point to the file specified in the associated Programming Project (*.ppm).</li> <li>• To load a different file, click the &lt;Load&gt; button and browse to the required file.</li> </ul>
2	<b>Re-load file to buffer</b>	<p>Clicking the  icon re-loads the specified Flash / EEPROM file into the Buffer Window. This will overwrite any information already in the Buffer Window.</p>
3	<b>Buffer Control Buttons</b>	<p>This group of buttons control operations on the EDS – Buffer Window.</p> <ul style="list-style-type: none"> <li>• To use these buttons or to manually edit the Buffer Window, it is necessary to check the 'Edit Buffer' check box.</li> <li>• This extra step helps to avoid accidental modification of data in the buffer.</li> </ul>
4	<b>Target Control Buttons</b>	<p>This group of buttons control operations on the actual Target Device.</p> <ul style="list-style-type: none"> <li>• A programmer and suitable Target System must be connected when using these buttons.</li> </ul>
5	<b>Signature of Target Device</b>	<p>This is the signature (Device ID) which is expected for the Target Device.</p>
6	<b>Buffer Address</b>	<p>This is the address of the currently selected location in the Buffer Window.</p>
7	<b>Buffer Window</b>	<p>The Buffer Window displays a hexadecimal and alphanumeric representation of the data which has either been loaded from file or read back from a Target Device.</p> <ul style="list-style-type: none"> <li>• The bytes are grouped into rows of 16 bytes with the start address of each row displayed in the left-hand column.</li> <li>• The Hexadecimal representation of the 16 bytes is displayed in the middle column</li> <li>• The ASCII representation of the 16 bytes is displayed in the right-hand column.</li> </ul>
8	<b>CRC</b>	<p>This is a CRC Checksum of the entire Buffer (ie. from address 0x00000 to the end address specified in the buffer).</p>
9	<b>Size</b>	<p>This is the physical address range in bytes of the entire Buffer.</p>

### 2.5.5 Buffer Window - Control Buttons

The Buffer Window within EDS is used as a virtual data store which allows data to be transferred from file → Buffer Window → Target Device and vice-versa.

The 'Buffer – Control Buttons' support operations on the data in the Buffer Window including loading a file into Buffer or saving the contents of the Buffer to file. These buttons do NOT initiate any actual programming operation of the Target Device.

#	Button	Description / Purpose
1	 <b>Re-load File</b>	<ul style="list-style-type: none"> <li>Re-loads the specified HEX / Binary file into the Buffer Window.</li> </ul>
2	<b>Edit Buffer</b>	<ul style="list-style-type: none"> <li>Tick the box to enable manual editing of the Buffer Window.</li> <li>This will also allow the &lt;Load&gt;, &lt;Erase&gt; and &lt;Fill&gt; buttons to be selected.</li> </ul>
3	<b>&lt;Load&gt;</b>	<ul style="list-style-type: none"> <li>This button allows you to manually browse to select a file to be load into the Buffer Window.</li> <li>The path will default to the path set up in the Programming Project (if a file is specified).</li> </ul>
4	<b>&lt;Save as&gt;</b>	<ul style="list-style-type: none"> <li>This button allows you to save the contents of the Buffer Window to a file.</li> </ul> <p>It is possible to specify the following properties for the file:</p> <ul style="list-style-type: none"> <li>File type (HEX / Binary)</li> <li>File Start and End Address</li> </ul>
5	<b>&lt;Erase&gt;</b>	<ul style="list-style-type: none"> <li>This button erases the entire contents of the Buffer Window.</li> <li>All locations are set to the value 0xFF.</li> </ul>
6	<b>&lt;Fill&gt;</b>	<ul style="list-style-type: none"> <li>This button allows you to fill a specified area of the Buffer Window with a specified value.</li> </ul>
7	<b>Calc CRC</b>	<ul style="list-style-type: none"> <li>This button calculates the CRC Checksum of the specified range of the Buffer Window.</li> </ul>

## 2.5.6 Example of using EDS

Here is an example of how to use EDS:

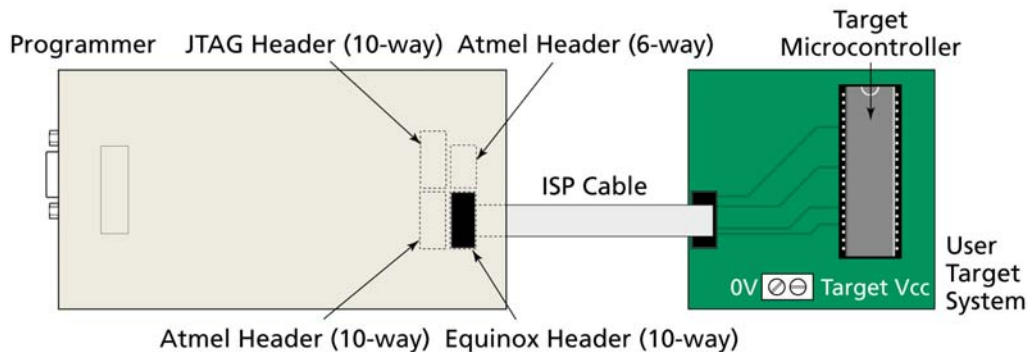
- Select the <FLASH> tab
- Click the <Load> button to load a file
- Click the <Check Sig> button to check the Device Signature of the Target Chip
- Click the <Write> button to write the data in the Buffer Window to the Target Device
- Click the <EEPROM> tab to select operations on the EEPROM area of the Target Chip/
- Click the <Fuses> tab to read / write the Target Chip 'Configuration Fuses'
- Click the <Security> tab to read / write the Target Chip 'Security Options'

For further instructions about how to use the EDS utility, please refer to the '**EQTools – Getting Started Guide**' which can be found on the CD-ROM supplied with the programmer.

## 2.6 Standalone Mode

### 2.6.1 Overview

In 'Standalone Mode', the programmer is not connected to the PC and is controlled via the <YES> and <NO> push buttons on the front panel instead. This is ideal for both field and production ISP requirements as the programming data is held locally in FLASH memory on the programmer so a PC is not required.



The LED Status Indicators are used to display the current status of the programmer. A single 'Programming Project' must be uploaded to the programmer using the EQTools 'Upload Wizard' before the programmer can be operated in Standalone Mode.

### 2.6.2 Steps required before operating the programmer in Standalone Mode

Before operating the programmer in Standalone Mode, you will need to:

- Create a **Development Mode** Project in **EDS**
- Use **EDS** to test that all the project parameters are correct by attempting to program the FLASH, EEPROM, Fuses and Security Options of the Target Chip
- Once you are happy that the **EDS** project is working correctly, you then need to export your **EDS** project to **Project Builder**.
- Compile the Project in **Project Builder**
- Add the Project to a **Project Collection** using **Project Manager**
- Upload the **Project Collection** to the attached programmer using **Upload Wizard**.

### 2.6.3 How to export your Development (EDS) project to Project Builder

To export your Development (EDS) project to Project Builder:

- In EDS Mode, select the <Overview> tab
- Click the <Open/Modify Base Programming Project> button
- EDS window will close and your project will be opened by **Project Builder**
  - On the top icon bar within EQTools, click the <Compile/Build> icon
- the project will be compiled
  - Select <Add Project to new Project Collection>
- your project will be added as an entry in the **Project Manager** window
  - Save the Project Collection with a suitable name eg. *Myprojects.ppc*
  - Click the <Upload> icon on the icon bar to upload the **Project Collection** to the programmer.
- the <Upload Wizard> utility will launch
  - Follow the on-screen instructions to upload the Project Collection to the programmer.

## 2.6.4 Executing a Programming Project in Standalone Mode

#	Action	Observation / comment
1	Connect the programmer to a spare PC COM port	
2	Apply power to the Programmer / Target System  See section 3 for Power Supply scenarios	→ Programmer Target Vcc LED illuminates
3	Upload your 'Programming Project(s)' to the programmer	To upload 'Programming Project(s)' to the programmer: <ul style="list-style-type: none"> <li>• Locate your Project Collection (*.ppc) file.</li> <li>• Click the &lt;Upload&gt; icon → The &lt;Upload Wizard&gt; utility will launch</li> <li>• Follow the on-screen instructions to upload your Project Collection to the programmer.</li> <li>• Your Project(s) are now resident in the memory of the Programmer.</li> <li>• See EQTools Manual – Project Upload Wizard for full instructions.</li> </ul>
4	Disconnect power from the Programmer / Target System	All uploaded Programming Projects will be retained by the programmer in on-board non-volatile FLASH memory.
5	Disconnect the serial lead from the programmer	→ Programmer is no longer controlled from PC
6	Follow relevant instructions in section 3 to program a Target Device.	The instructions to program a device in 'Standalone Mode' differ depending on how the programmer and Target System are powered.

For detailed instructions of operating the programmer in Standalone Mode, please refer to the relevant sections of the manual as detailed in the table below:

#	Power Supply Scenario	Jumper J9	Voltage (V)	Refer to section
1	Power Supply Schematic Overview	-	-	3.2
2	Earthing requirements	-	-	3.3
3	Programmer powers the Target System	FITTED	3.1 – 5.0	3.4
4	Target System powers the Programmer	FITTED	3.1 – 5.0	3.5
5	Programmer and Target System are independently powered	NOT FITTED	3.1 – 5.0	3.6

## 2.7 Script Mode

'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 consult the Equinox website for further details of all the upgrades which are available for this product.

## 2.8 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 '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 Programmer / Target System Power Supply Scenarios + Standalone Mode Programming Instructions

### 3.1 Overview

The EPSILON5 programmer has been designed so that it can be powered both directly from a Target System (3.1 – 5.0V) and also from an external power supply (3.1 – 5.0V). This makes the programmer ideal for both standalone use (not connected to a PC) and development use (connected to a PC). The programmer can be powered from Target System voltages as low as 3.1V allowing it to be used to program 3.3V Target Systems in the field (eg. on vehicles) without requiring the use of an external power supply.

Please refer to the table below to choose the required 'Power Supply Scenario'. The relevant section also gives detailed instructions on how to operate the programmer in 'Standalone' mode for the chosen 'Power Supply Scenario'.

#	Power Supply Scenario	Jumper J9	Voltage (V)	Refer to section
1	Power Supply Schematic Overview	-	-	3.2
2	Earthing requirements	-	-	3.3
3	Programmer powers the Target System	FITTED	3.1 – 5.0	3.4
4	Target System powers the Programmer	FITTED	3.1 – 5.0	3.5
5	Programmer and Target System are independently powered	NOT FITTED	3.1 – 5.0	3.6

#### **Electrostatic Discharge (ESD) Warning!**

There are a number of electrostatic issues which must be taken into account to avoid damage to the programmer and/or the Target System during In-System Programming. Please see section 3.3.



## 3.2 Programmer / Target System Power Supply Schematic

The schematic shown in fig. 3.2.2 details the possible power supply scenarios for powering both the programmer and the Target System.

The key points to note are as follows:

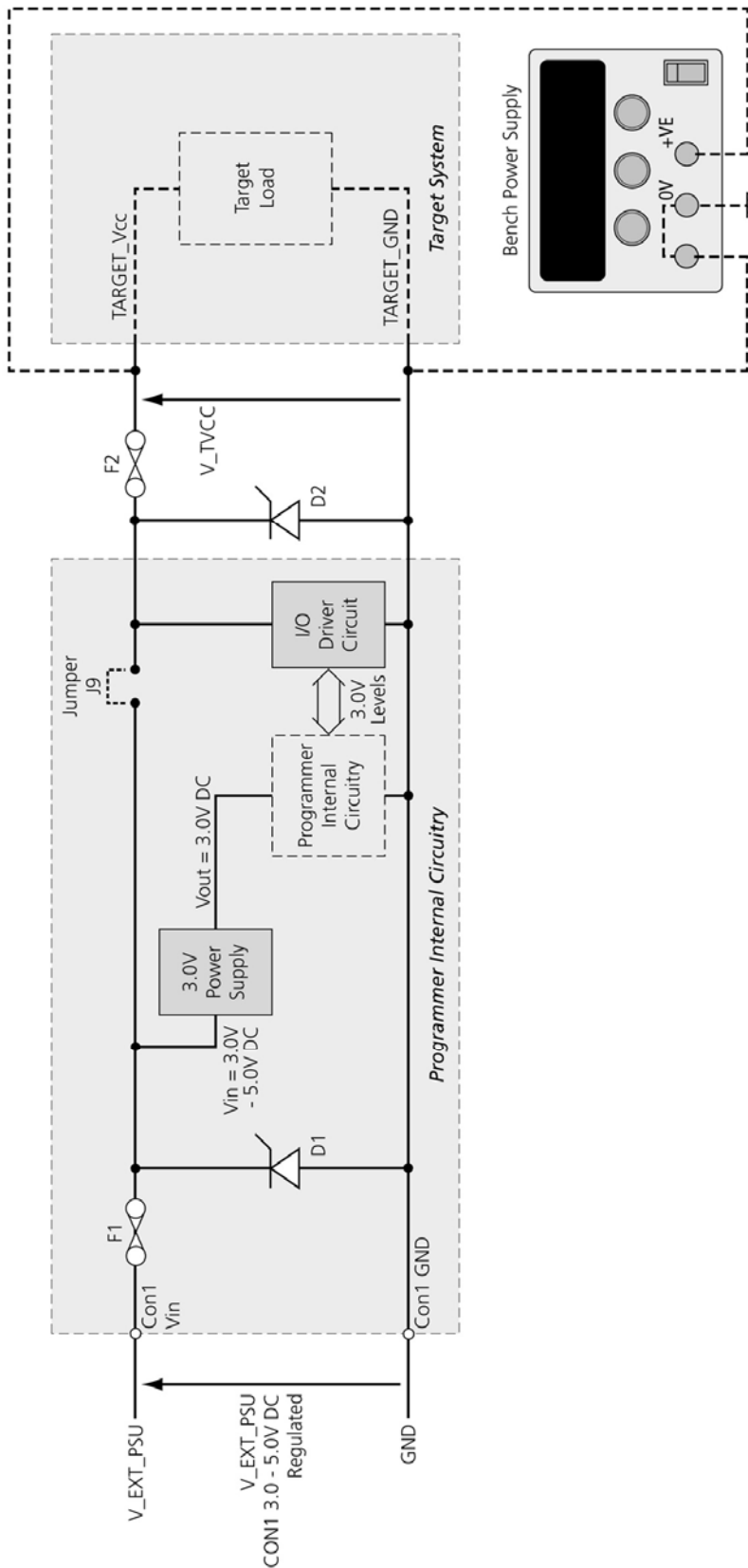
- The programmer can be powered from an external power supply via CON1 (V\_EXT\_PSU) using the DC power cable provided with the programmer.
- The programmer can power the Target System if J9 is fitted and external power is supplied via CON1
- The programmer can be powered from the Target System if Jumper J9 is fitted and the Target System is powered.
- If Jumper J9 is fitted then the Target Vcc is connected to the CON1 Vin terminal.
- The internal circuitry of the programmer is powered at +3.0V via an on-board +3.0V supply.

fig. 3.2.1. Explanation of all Power Supply Circuit Schematic labels

#	Item	Description
1	V_EXT_PSU	This is the voltage applied to the CON1 connector from an external power supply.
2	GND	This is common ground of the programmer which is connected to CON1 GROUND and to the TARGET GROUND via one of the ISP Headers.
3	V_TVCC	This is the voltage measured between the Vcc and Ground of the Target System.
4	I/O Driver Circuit	This is the circuitry which translates the 3.0V I/O signals from the programmer 'Supervisor Microcontroller' to the Target System voltage levels.
5	Target Load	This is the load presented to the programmer across the PROG_VCC and GROUND terminals of any ISP Header.
6	Bench Power Supply	This simply indicates a stable DC power supply with a current limit facility.
7	F1	This is an internal resettable fuse rated at 1.0A.
8	F2	This is an optional fuse which should be placed in the PROG_VCC line if the jumper J9 is fitted and the programmer is powering the Target System.
9	D1	This is a 6V2 5W zener diode which is connected directly across the programmer DC Input (CON1). This diode will limit the supply to <6.2V.
10	D2	This is an optional 6V3 5W zener diode which can be connected across the Target Vcc and GROUND within the Test Fixture. This diode would help to prevent damage to the programmer due to the Target Vcc being > 5V.
11	J9	Jumper (J9) – Vcc Jumper Fitted – CON1 V_EXT_PSU is connected to TARGET_VCC Not Fitted – Programmer PSU is independent of Target Vcc

Fig 3.2.2 – Programmer / Target System Power Supply Overview Schematic

Please refer to fig 3.2.1 for an explanation of all circuit diagram labels etc.



## 3.3 Earthing requirements

### 3.3.1 Overview

When setting up the programmer to In-System Program (ISP) a device on a Target System, extreme care must be taken to ensure that the 0V of the PC, programmer, Target System and any external devices are at the same 'earth potential'. If you are using the programmer with a laptop computer, please read section 3.2.2 carefully.



#### ESD Points to remember

Failure to follow correct ESD procedures when using this product could lead to damage to the programmer and/or the Target System.

- Wear an earthed wrist strap when handling either the programmer and/or any programmable device.
- Ensure that the PC, programmer, Target System and any external devices are all connected to the same GROUND potential.

### 3.3.2 Laptop earthing issues

#### The Problem:

Most laptops use an external double-insulated mains power supply which is not earthed to mains earth. This means that the laptop earth is likely to be floating at some voltage other than 0V. If the programmer is attached to the PC COM port with the PC powered ON and then the ISP lead is plugged into an earthed Target System, then the PC GROUND will discharge through the COM port GROUND, through the programmer into the Target System. This ESD may damage both the programmer and the Target System !

#### The Solution

- Ensure that the laptop is powered OFF when connecting the ISP cable from the programmer to an earthed Target system
- Use the programmer in Standalone Mode if possible (ie. not connected to a PC)

### 3.3.3 Desktop PC earthing issues

#### The Problem:

The chassis on desktop PC must be connected to mains earth to comply with Health and Safety legislation. If the chassis is not earthed for some reason, the PC power supply GROUND will be floating and it is then possible to discharge a high level of energy stored in the power supply mains filter through the programmer into the Target System. This ESD may damage both the programmer and the Target System !

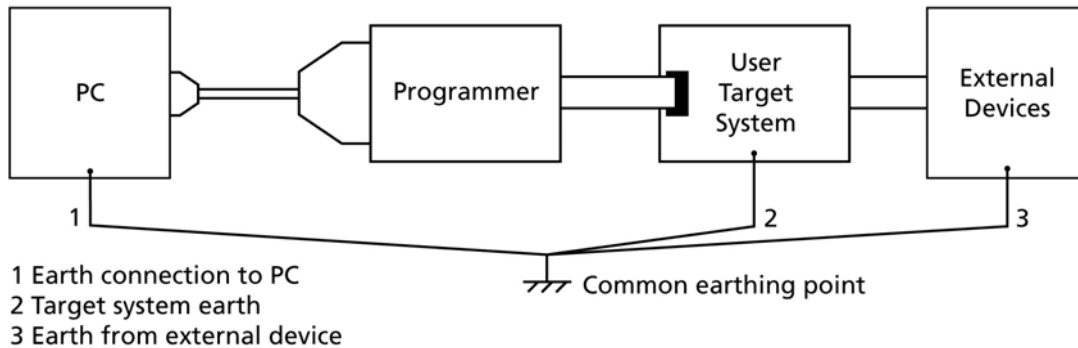
#### The solution:

- Ensure that the PC chassis is properly earthed to mains earth.
- Connect an earthing bond wire from the PC chassis to the GROUND connection of either the programmer or the Target system / Test Fixture.

### 3.3.4 Earthing recommendations

To avoid catastrophic damage to PC, programmer or target system:

- Ensure that both your target system and PC are connected to a common earth point
- Make sure that all interconnections are made before applying power to PC and target system



- If you are using a laptop or PC which is not connected to mains earth, it is recommended that you make a hard-wired connection from the COM port D-connector shell and the target system to a common earth point
- Avoid plugging and unplugging ISP connector while either the PC or Target System is powered up
- Please ensure that any devices connected to the user target system are also properly grounded to the same common earth point

#### Warning!

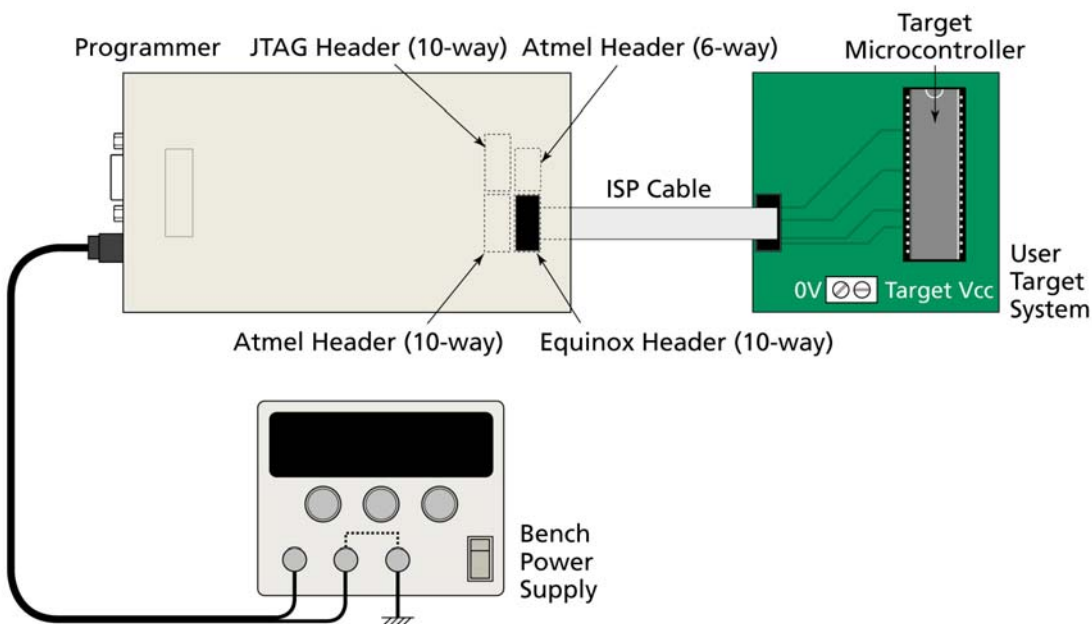
Any damage caused to the programmer by Electrostatic Discharge (ESD) through inadequate earthing is not covered under the warranty of the product.

## 3.4 Programmer Powers the Target System

### 3.4.1 Overview

In this scenario, the programmer is constantly powered from an external power supply (not supplied) via the connector **CON1**. The **Jumper J9** is fitted which connects the voltage applied to **CON1** to the **PROG\_VCC** pin on each ISP header. The external power supply voltage **V\_EXT\_PSU** is therefore routed directly to the **Target System Vcc**. When an ISP Cable is connected from the programmer to a Target System, the Target System will power up to the external power supply voltage **V\_EXT\_PSU**. When the cable is removed, the Target System will power off.

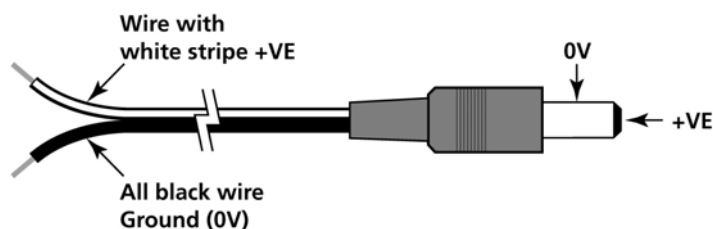
### 3.4.2 Hardware Setup - Programmer powers the Target System



#### DC Power Cable

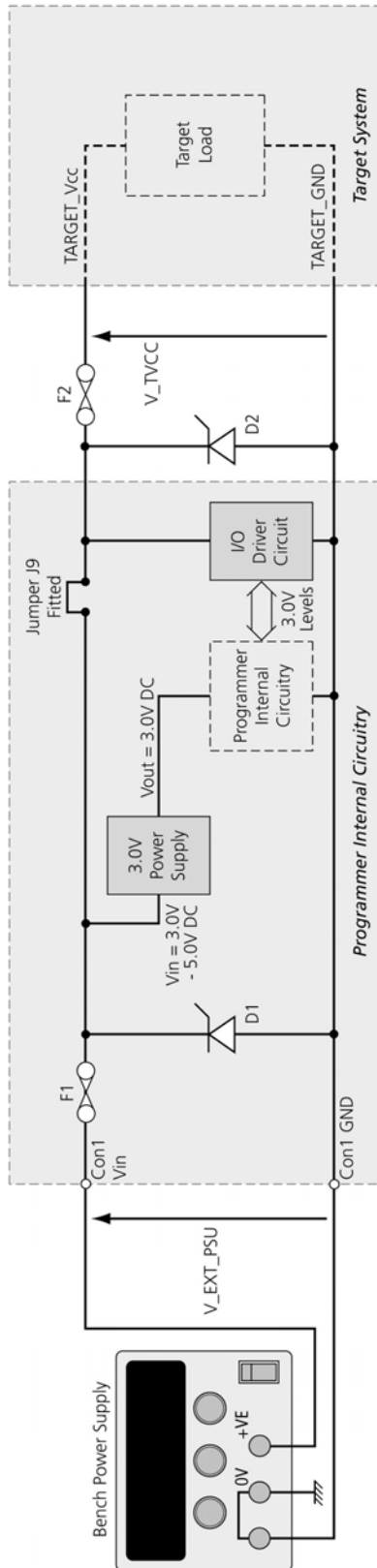
The programmer is supplied with a 'DC Power Cable' which plugs into the CON1 - DC Power Input Connector on the programmer.

- The black lead with the white stripe is the +VE cable.
- The all black lead is the GROUND cable.
- The external voltage applied to CON1 must be between 3.1 and 5.0V !
- The Target System must not draw more than 300 mA from the programmer.



### 3.4.3 Circuit Schematic - Programmer powers the Target System

Please refer to fig 3.2.1 for an explanation of all circuit diagram labels etc.

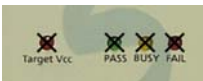

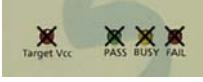
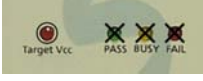

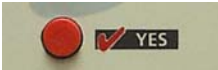


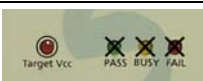
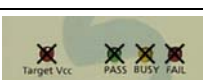



### 3.4.4 Set up instructions - Programmer powers the Target System

The instructions below detail how to set up the programmer / Target System so that the programmer provides power to the Target System.

**Status LED key:**

ON FLASHING OFF  
  

#	Action	Observation	Status of LED's
1	Remove programmer cover and fit Jumper J9.	This connects the <b>Programmer Vcc</b> (from Con1) to the <b>Target Vcc</b> .	
2	Ensure that the programmer is NOT connected to the Target System via the ISP Connector	All LED's are OFF Programmer and Target system are NOT powered.	
3	Connect the ISP cable between correct ISP Header on the programmer and the ISP Header on the Target System.	All LED's are OFF Programmer and Target system are NOT powered	
4	Apply required voltage (3.0V to 5.0V DC) to Con1 Power Connector from external Power Supply	→ Programmer circuitry powers up to 3.0V. → Target System powers up to the voltage applied to CON1. → 'Target Vcc' LED will illuminate as the Target Voltage (V_TVCC) is detected.	
5	Check the <b>Target Vcc voltage (V_TVCC)</b> is the same as the <b>external power supply voltage (V_EXT_PSU)</b> using a volt meter.	V_TVCC = V_EXT_PSU	
6a	Press  button once	→ 6b or 6c	
6b	There is no valid Programming Project in the Programmer.  <b>You need to upload a Programming Project before using the programmer in Standalone Mode.</b>	→ FAIL LED starts flashing.  Pressing any button will have no effect.	
6c	There is a valid Programming Project in the Programmer.	→ programmer is now in 'Autoprogram Mode'.  Press the <YES> button again to execute the Programming Project.	
7	Remove power from programmer	→ All LED's go out	

8	Replace programmer lid and tighten screws		
---	---	--	---


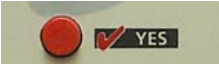


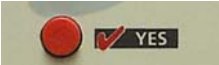









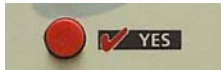
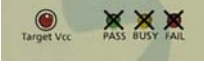
### 3.4.5 Programming Instructions (Standalone Mode) - Programmer powers the Target System

The instructions detailed in the table below demonstrate how to program a Target Device in 'Standalone' mode when the programmer is powering the Target System. It is assumed that a valid 'Programming Project' is already uploaded into the programmer 'FLASH Memory Store'.

#### Status LED Key:

ON FLASHING OFF  
  

#	Action	Observation	Status of LED's
1.	Power up the programmer	→ Programmer circuitry powers up to 3.0V. → 'Target Vcc' LED is ON	
2.	Press  button once	→ Programmer is now in 'Standby Mode' waiting to start an 'Autoprogram' operation.	
3.	Connect the ISP cable between correct ISP Header on the programmer and the ISP Header on the Target System.	→ Target System powers up to the voltage applied to CON1 → Target Microcontroller may start to execute code if has already been pre-programmed.	
4.	Press  button	→ BUSY LED illuminates and stays ON → Programmer starts an 'Autoprogram' operation → Programmer OUTPUT lines will now be driven high/low by the programmer for the duration of the programming operation.	
5.	Wait for Autoprogram operation to finish.  This may take a few seconds or many minutes depending on the programming time of the Target Device.  This operation has 2 possible outcomes: i. <PASS> LED flashing – go to step 7a. ii. <FAIL> LED FLASHING – go to step 6a.  To cancel a programming operation part-way-through press the  button once.	→ BUSY LED remains on throughout the 'Autoprogram' operation	

6a.	Programming operation has FAILED !	→ FAIL LED flashes → Programmer tri-states all outputs connected to the Target System	
6b.	To clear the ERROR CONDITION:  Press  key once	→ FAIL LED goes out → Programmer is in 'Standby' mode again	
6c.	To re-program the same Target System Go to step 4.		
6d.	To program a new Target System after a FAILURE:  i. Disconnect ISP Cable from Target system ii. Go to step 3		
7a.	Programming operation has PASSED.	→ <PASS> LED flashes → Programmer tri-states all outputs to connect to Target system → Target System RESET is released (target code may start to execute)	
7b.	To program a new Target System:  i. Press the  button ii. Go to step 3	→ <PASS> LED goes out → Programmer is now in the 'Waiting State' ready to start an 'Autoprogram' operation.	

## 3.5 Target System powers the Programmer

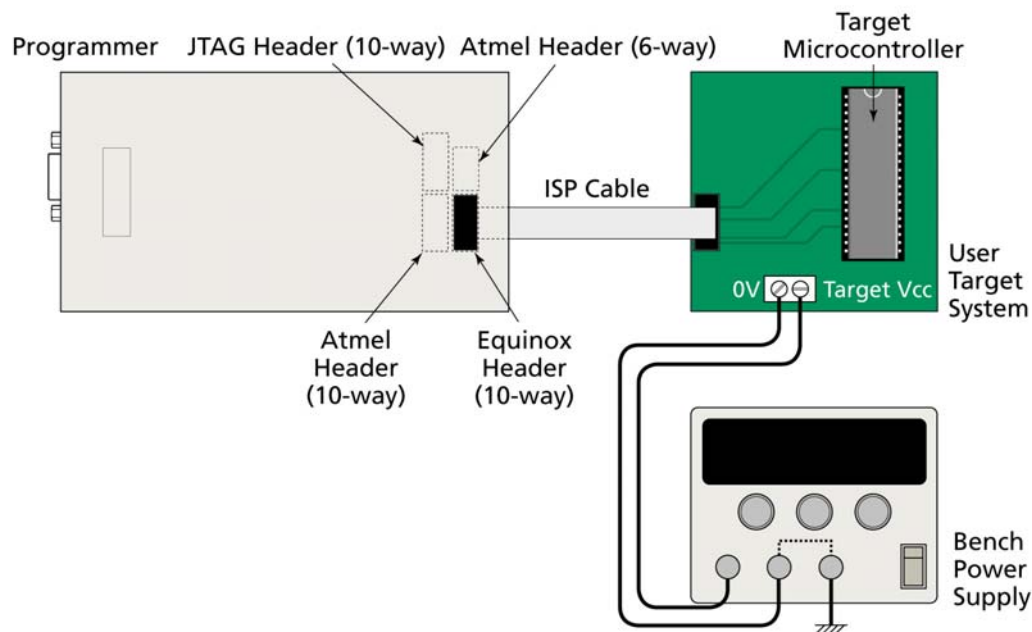
### 3.5.1 Overview

In this scenario, the programmer is powered from the Target System via the ISP cable. The **Jumper J9** is fitted which connects the Target Vcc voltage ( $V_{TVCC}$ ) to **CON1 VIN**. This powers the internal circuitry of the programmer.

#### Explanation of operation:

- When the ISP Cable is connected from the programmer to a Target System and the Target system power supply is switched ON, the programmer will power up.
- When the ISP Cable is disconnected from the Target System or the Target System is powered OFF, the Programmer will power OFF.
- The programmer should only be connected/disconnect from the Target system when the Target power supply is switched OFF

### 3.5.2 Hardware Setup - Target System powers the Programmer

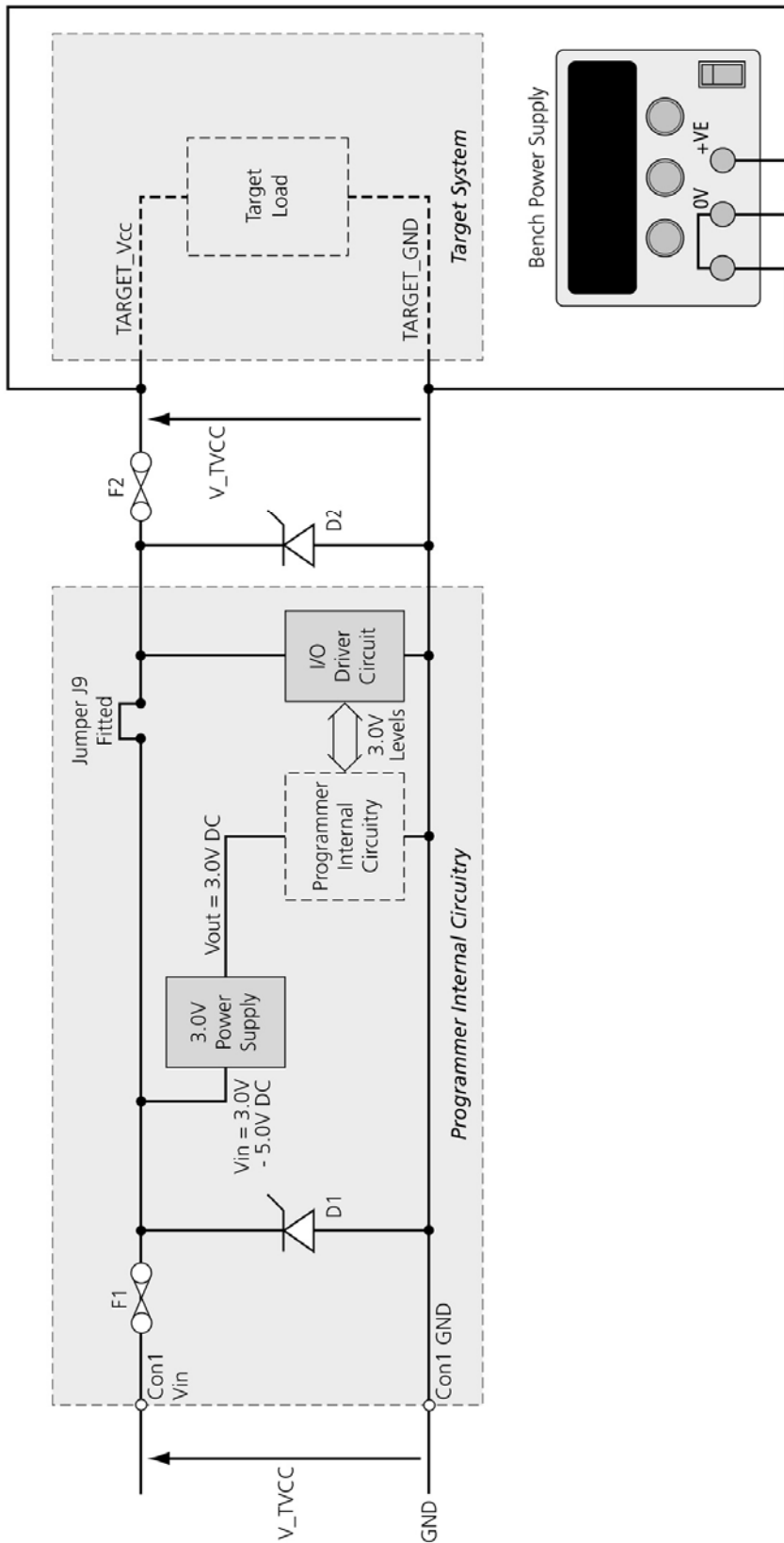


#### Please note

- The external voltage applied to the programmer must be between 3.1 and 5.0V!
- The Target System must be able to supply the worst case current required during a programming operation – see section 1.5.3.
- A 6V2 5W zener diode is internally connected across the programmer supply to protect against over-voltage.

### 3.5.3 Circuit Schematic - Target System powers the Programmer

Please refer to fig 3.2.1 for an explanation of all circuit diagram labels etc.

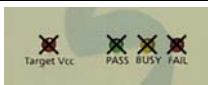



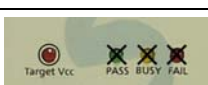








## 3.5.4 Setup instructions - Target System powers the Programmer

The instructions below detail how to set up the programmer / Target System so that the Target System provides power to the programmer.

### Status LED key:

ON FLASHING OFF  
  

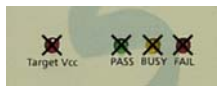
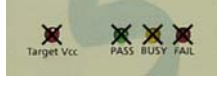

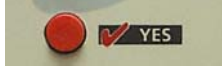

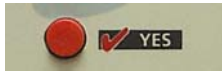



#	Action	Observation	Status of LED's
1	Remove programmer cover and fit Jumper J9.	This connects the <b>Programmer Vcc</b> (from Con1) to the <b>Target Vcc</b> .	
2	Ensure that the programmer is NOT connected to the Target System via the ISP Connector	All LED's are OFF Programmer and Target system are NOT powered.	
3	Connect the ISP cable between correct ISP Header on the programmer and the ISP Header on the Target System.	All LED's are OFF Programmer and Target system are NOT powered	
4	Apply power to Target System from an external supply (V_TVCC must be 3.1 – 5.0V)	→ Target System powers up to V_TVCC → Programmer circuitry powers up to 3.0V	
5	Check the <b>Target Vcc voltage (V_TVCC)</b> is the same as the <b>external power supply voltage</b> applied to the Target System.	V_TVCC = External Target Vcc Voltage	
6a	Press  button once	→ 6b or 6c	
6b	There is no valid Programming Project in the Programmer.  <b>You need to upload a Programming Project before using the programmer in Standalone Mode.</b>	→ FAIL LED starts flashing.  Pressing any button will have no effect.	
6c	There is a valid Programming Project in the Programmer.	→ programmer is now in 'Autoprogram Mode'.  Press the <YES> button again to execute the Programming Project.	
7	Remove power from programmer	→ All LED's go out	
9	Replace programmer lid and tighten screws		

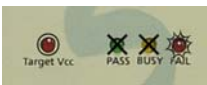


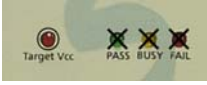

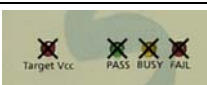
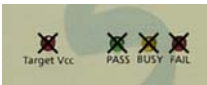
### 3.5.5 Programming Instructions (Standalone Mode) - Target System powers the programmer

The instructions detailed in the table below demonstrate how to program a Target Device in 'Standalone' mode when the Target System is powering the programmer. It is assumed that a valid 'Programming Project' is already uploaded into the programmer 'FLASH Memory Store'.

#### Status LED Key:

ON FLASHING OFF  
  

#	Action	Observation	Status of LED's
1	Ensure Target System is power is OFF	All LED's are OFF	
2	Connect the ISP cable between correct ISP Header on the programmer and the ISP Header on the Target System.	No change as Target power supply is OFF.	
3	Switch ON Target Power Supply	→ Target System powers up → Programmer circuitry powers up to 3.0V (derived from T_VCC). → POWER LED is ON	
4	Press  button once	→ Programmer is now in 'Standby Mode' waiting to start an 'Autoprogram' operation.	
5	Press  button	→ BUSY LED illuminates and stays ON → Programmer starts an 'Autoprogram' operation → Programmer OUTPUT lines will now be driven high/low by the programmer for the duration of the programming operation.	
6	Wait for Autoprogram operation to finish.  This may take a second or many minutes depending on the programming time of the Target Device.  This operation has 2 possible outcomes: iii. <PASS> LED flashing – go to step 8a. iv. <FAIL> LED FLASHING – go to step 7a.  To cancel a programming operation part-way-through press the  button	→ BUSY LED remains on throughout the 'Autoprogram' operation	

	once.		
7a.	Programming operation has FAILED !	<ul style="list-style-type: none"> <li>→ FAIL LED flashes</li> <li>→ Programmer tri-states all outputs connected to the Target System</li> </ul>	
7b.	<p>To clear the ERROR CONDITION:</p>  <p>Press  key once</p>	<ul style="list-style-type: none"> <li>→ FAIL LED goes out</li> <li>→ Programmer is in 'Standby' mode</li> </ul>	
7c.	To re-program the same Target System Go to step 5.		
7d.	To program a new Target System after a FAILURE: Go to step 9.		
8a.	Programming operation has PASSED.  Go to step 9.	<ul style="list-style-type: none"> <li>→ &lt;PASS&gt; LED flashes</li> <li>→ Programmer tri-states all outputs to connect to Target system</li> <li>→ Target System RESET is released (target code may start to execute)</li> </ul>	
9.	Switch Target Power Supply OFF	<ul style="list-style-type: none"> <li>→ Target System powers down</li> <li>→ All LED's go OFF</li> <li>→ Programmer powers OFF</li> </ul>	
10.	Disconnect ISP Cable from Target System	No change	
11.	Repeat procedure from step 1 with new Target System.		

## 3.6 Programmer and Target System are independently powered

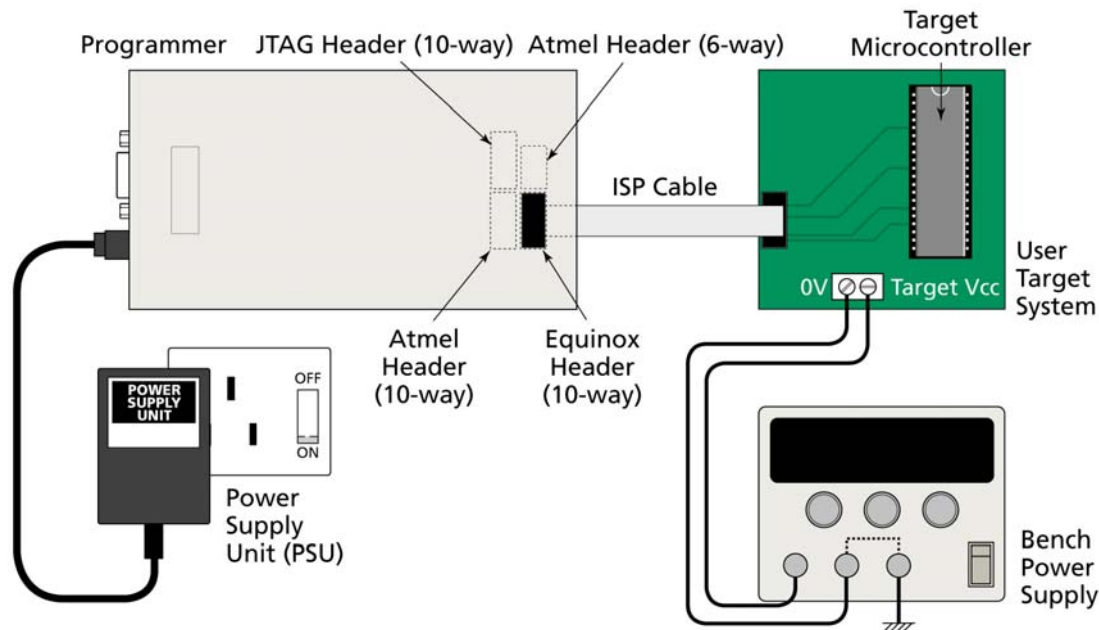
### 3.6.1 Overview

In this scenario, the programmer is powered from a completely independent supply to the Target System. The **Jumper J9** is NOT fitted which isolates the Target Vcc voltage (V\_TVCC) from **CON1** VIN. This means that the programmer is always powered up.

#### Explanation of operation:

- The programmer is always powered ON
- Jumper J9 is removed
- The Target Power Supply should be switched OFF before the ISP cable is plugged into it
- The programmer does not draw any current from the Target System during programming
- When programming is complete, the Target System should be powered OFF and then the ISP cable removed

### 3.6.2 Hardware Setup - Programmer and Target System independently powered



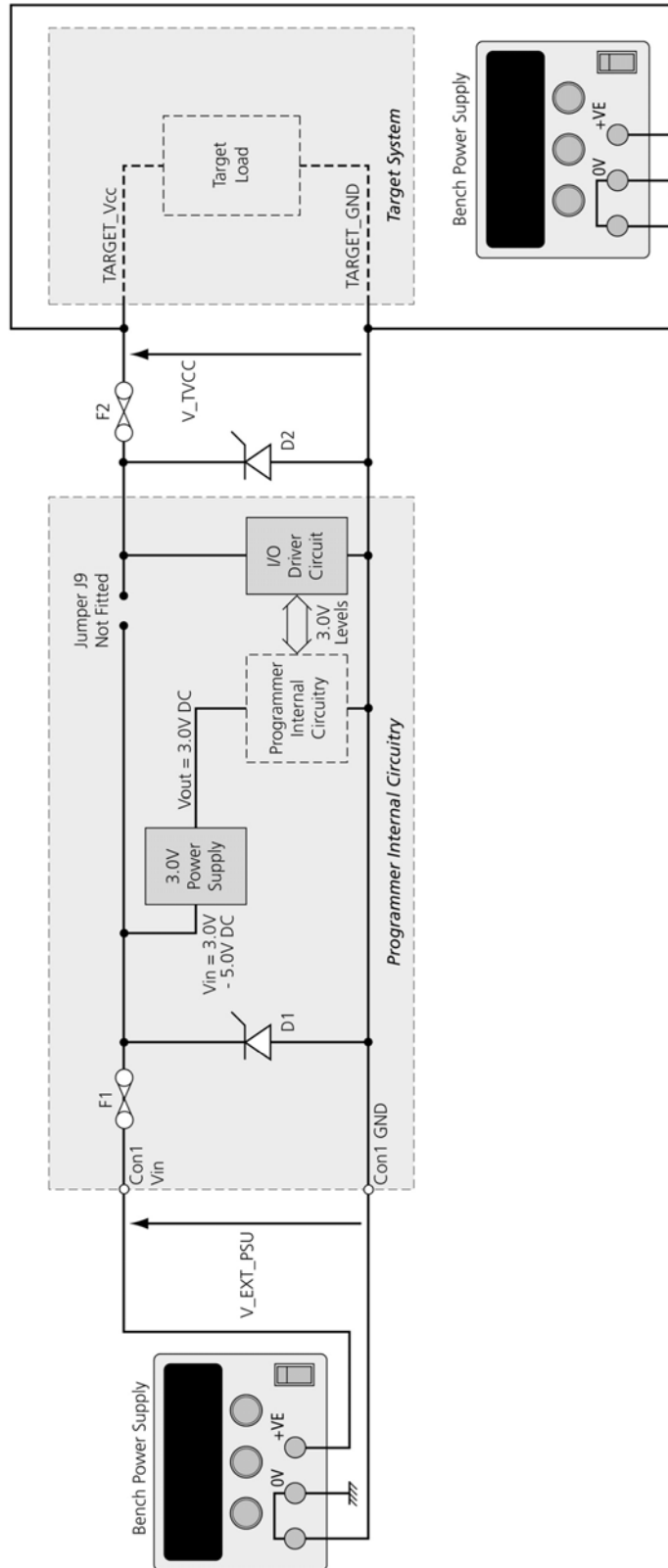
#### Warning!

- The external voltage applied to the programmer via CON1 must be between 3.1 and 5.0V !
- The voltage applied to the programmer via the ISP cable must be between 3.1 and 5.0V as this voltage is used to drive the programmer I/O Stage. A 6V2 5W zener diode is internally connected across the programmer supply to protect against over-voltage.
- J9 must be removed to avoid contention between the two power supplies.



### 3.6.3 Circuit Schematic - Programmer and Target System independently powered

Please refer to fig 3.2.1 for an explanation of all circuit diagram labels etc.






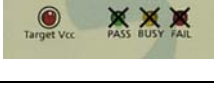

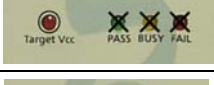
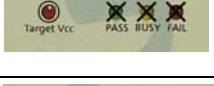





### 3.6.4 Set up instructions - Programmer and Target System independently powered

The instructions below detail how to set up the programmer / Target System so that they are independently powered.

#### Status LED key:

ON FLASHING OFF  
  

#	Action	Observation	Status of LED's
1	Remove programmer cover and remove Jumper J9.	This disconnects the <b>Programmer Vcc</b> (from Con1) to the <b>Target Vcc</b> .	
2	Ensure that the programmer is NOT connected to the Target System via the ISP Connector	All LED's are OFF Programmer and Target system are NOT powered.	
3	Connect the ISP cable between correct ISP Header on the programmer and the ISP Header on the Target System.	All LED's are OFF Programmer and Target system are NOT powered	
4	Apply required voltage (3.0V to 5.0V DC) to Con1 Power Connector from external Power Supply. Note – This voltage is NOT applied to the Target System as J9 is removed.	→ Programmer internal circuitry powers up to 3.0V	
5	Apply power to Target System from an external supply (V_TVCC must be 3.1 – 5.0V)	→ Target System powers up to V_TVCC → Target Vcc LED illuminates	
6	Check the <b>Target Vcc voltage (V_TVCC)</b> is 3.1 – 5.0V using a volt meter	V_TVCC = 3.1 – 5.0V	
7a	Press  button once	→ 7a or 7b	
7b	There is no valid Programming Project in the Programmer. <b>You need to upload a Programming Project before using the programmer in Standalone Mode.</b>	→ FAIL LED starts flashing. Pressing any button will have no effect.	
7c	There is a valid Programming Project in the Programmer.	→ programmer is now in 'Autoprogram Mode'. Press the <YES> button again to execute the Programming Project.	
8	Remove power from programmer	→ All LED's go out	

9	Replace programmer lid and tighten screws		
---	---	--	---

### 3.6.5 Programming Instructions

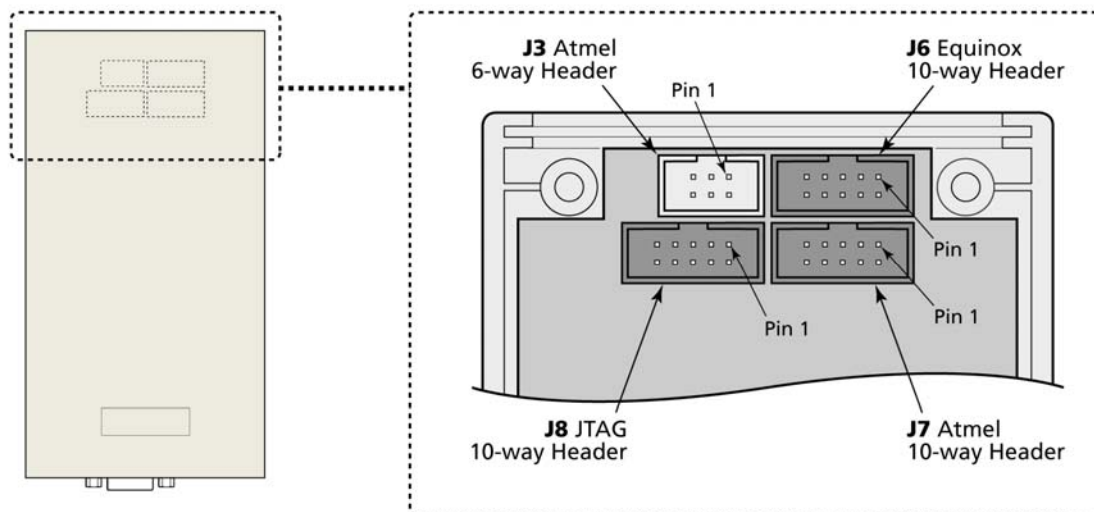
As the programmer is always powered ON, the programming instructions are exactly the same as for when the programmer powers the Target System except that jumper J9 must be removed. Please refer to section 3.4.5.

## 4.0 ISP Header Selection

### 4.1 Overview

The EPSILON5 programmer caters for many different connection methods to a Target System by featuring FOUR possible ISP Header connectors. Each header provides the necessary signals to program the Target Device plus Vcc (power) and GND (Ground). The choice of which header to use depends on the 'device type' and the algorithm being employed to program the device.

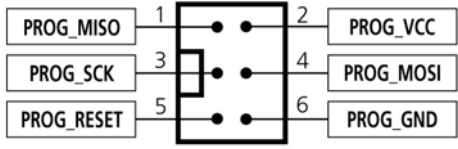
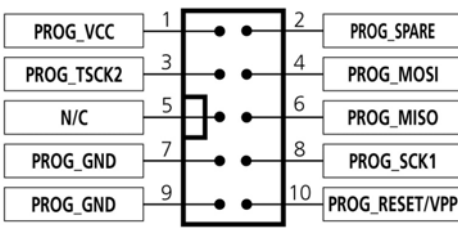
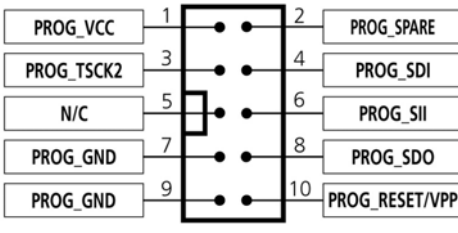
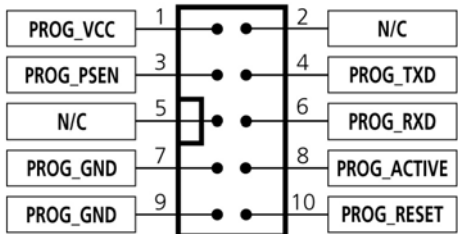
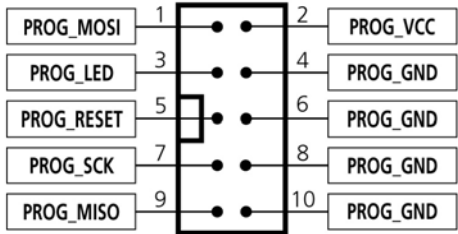
To locate the ISP Headers (J3, J6, J7 & J8), remove the four screws from the back of the programmer and then carefully remove the lid. If you hold the programmer with the RS-232 connector towards your, the four ISP headers will be in the positions shown in the figure below. Pin 1 is designated on each header and should match with the corresponding pin 1 of the cable to be plugged into the header.

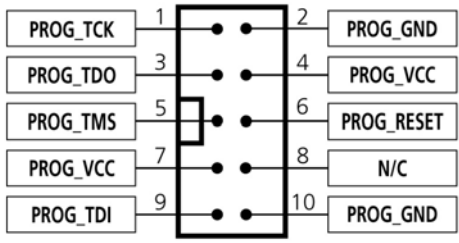


Please see section 4.2 for an overview of each ISP Header.

## 4.2 ISP Header Selection Chart (by header)

The FOUR ISP Headers featured on the EPSILON5 are detailed in the table below. Please refer to the section indicated in the 'refer to section' column for specific details of each header.

#	ISP Header	Description / Function	Refer to section	ISP Header Pin-out
1	J3	Atmel 6-way ISP Header	4.4	
Header J6 can have THREE different pin-outs depending on which Target Device is to be programmed. See (2a), (2b) and (2c).				
2a	J6(a)	<b>Equinox 10-way Header(a)</b>  <b>Device support:</b> Atmel AT90S, ATmega, ATtiny, AT89S devices	4.5	
2b	J6(b)	<b>Equinox 10-way Header(a)</b>  <b>Device support:</b> Atmel ATtiny11/12/15 High Voltage (+12V Vpp) Programming Mode	4.6	
2c	J6(c)	<b>Equinox 10-way Header(b)</b>  <b>Device support:</b> Atmel Wireless T89C51Rx2 Philips P89C51Rx2 / 66x Zensys ZW0102, ZW0201	4.7	
3	J7	<b>Atmel 10-way Header</b>  <b>Device support:</b> Atmel AT90S, ATmega, ATtiny, AT89S devices	4.8	

4	J8	<p><b>Atmel 10-way JTAG Header</b></p> <p><b>Device support:</b> Atmel ATmega32/128 + any new devices with JTAG port</p>	4.9	
---	----	--	-----	--

## 4.3 ISP Header Selection Chart (by Device)

The table below details which header to choose for a specified device or device family. The information contained in this manual does NOT show the actual connections to the Target Microcontroller. Please refer to the separate Application Notes ( see App Note column in table below ) for full details of how to implement In-System Programming for your selected device.

**Please note:**

- The Atmel ATmega323/128/64 devices can be programmed via 'LV SPI' or 'JTAG' by connecting the relevant header to the SPI or JTAG port of the device.
- The header J6 is multi-functional. The pin-out of this header is different depending on which device type is being programmed.

#	Device Family	Devices	Algorithm	Header	Sect.	App Note
1	Atmel AT89S	AT89(L)S51/52/53/8252	LV SPI	J6(a) J7	4.4, 4.5 4.8	AN103
2	Atmel AT90S (AVR)	All AT90S ISP devices	LV SPI	J6(a) J7	4.4, 4.5 4.8	AN100
3	Atmel ATmega (AVR)	All ATmega devices	LV SPI	J6(a) J7	4.4, 4.5 4.8	AN101
4	Atmel ATmega (AVR)	ATmega323/128/64	JTAG	J8	4.9	AN105
5	Atmel ATtiny (Low Voltage)	All ATtiny ISP devices	LV SPI	J6(a) J7	4.4, 4.5 4.8	AN104
6	Atmel ATtiny (High Voltage)	ATtiny11/12/15L	HV SPI	J6(b)	4.6	AN104
7	Atmel W&uC (Temic) T89C51Rx2	T89CRB2/RC2/RD2	UART Boot Loader	J6(c)	4.7	AN106
8	Philips P89C51Rx2	P89CRB2/RC2/RD2	UART Boot Loader	J6(c)	4.7	AN106 AN107
9	Philips P89C66x	P89C662/4/8	UART Boot Loader	J6(c)	4.7	AN106 AN107
10	Zensys Z-Wave	ZW0102, ZW0201	LV SPI	J6(a)	4.5	See Zensy App Note

## 4.4 J3 - Atmel 6-way ISP Header (SPI Interface)

This connection method is suitable for interfacing the EPSILON5 programmer to a Target System which features the following:

- Atmel 6-way IDC ISP Header
- An Atmel device which features the 3-wire SPI + RESET Programming Interface

The 6-way IDC box header is the most compact header i.e. takes up least space on the Target System and so is very popular for designs where PCB space is limited. Unfortunately, this connection method does not have any pins spare for extra functionality such as the SCK2 Oscillator so it should only be used if this functionality is not required.

To implement this connection method, simply plug a 6-way ISP cable (not supplied) into the **EPSILON5 ISP Header J3** and plug the other end of the cable into the matching header on the Target System.

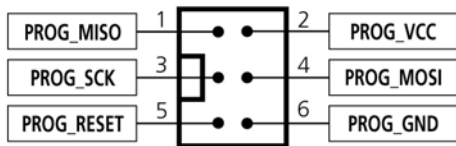


Figure 4.4.1 - Atmel 6-way IDC Header (J3) viewed from above

### Warning!

Connecting to the wrong ISP Header may cause catastrophic damage to the Programmer & Target System

Pin No	Programmer Pin name	Programmer Input / Output	Connect to pin on Target Device	Description
1	PROG_MISO	I	MISO  (except for ATmega103/128/64 – connect to TXD pin instead)	<b>Master In Slave Out</b> This is the SPI data input pin to the programmer. This pin should be connected to the MISO pin on the Target Microcontroller.
2	PROG_VCC	P	TARGET_VCC	<b>Target Vcc</b> This pin should be connected to the Target System Vcc. This voltage could be used to power the programmer depending on the settings of the power switch/jumper on the programmer.
3	PROG_SCK1	O	SCK	<b>SPI Serial Clock Output</b> This is the SPI clock output signal.
4	PROG_MOSI	O	MOSI  (except for ATmega103/128/64 – connect to RXD pin instead)	<b>Master Out Slave In</b> This is the SPI data output pin from the programmer. This pin should be connected to the MOSI pin on the Target Microcontroller.



5	PROG_RESET	O	RESET	<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.
6	PROG_GND	P	GROUND	<b>Ground Connection</b> Common ground connection between PROGRAMMER and Target System.

### Key

O - Output from programmer to Target Device

I - Input to programmer from Target Device

P - Passive eg. GROUND and power rails

N/C - Not connected

## 4.5 J6(a) - Equinox 10-way Header (Generic SPI Interface)

This connection method is suitable for interfacing the EPSILON5 programmer to a Target System which features the following:

- Equinox 10-way IDC ISP Header
- An Atmel device which features the 3-wire SPI + RESET Programming Interface

This is the favoured ISP connector as it allows the following additional functions:

- Application of +12V Vpp for ATtiny High Voltage Serial Programming
- SCK2 Oscillator for clocking devices during programming
- Spare I/O pin which could be used for SPI SLAVE SELECT or to drive an LED on the Target System

To implement this connection, simply plug the 10-way ISP cable into the **EPSILON5 J6 header** and plug the other end of the cable into the matching header on the Target System.

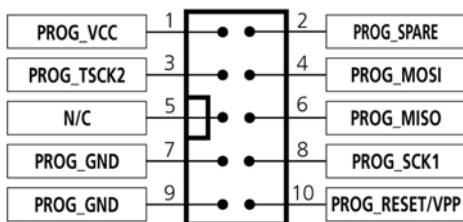


Figure 4.5.1 – Equinox 10-way IDC Header (J6) viewed from above

### Warning!

1. It is possible for +12V to be output on pin 10 of this connector when programming ATtiny11/12/15 devices.
2. Connecting to the wrong ISP Header may cause catastrophic damage to the Programmer & Target System

Pin No	Programmer Pin name	Programmer Input / Output	Connect to pin on Target Device	Description
1	PROG_VCC	P	TARGET_VCC	<b>Target VCC</b> This pin should be connected to the Target System Vcc. This voltage could be used to power the programmer depending on the settings of the power switch/jumper on the programmer.
2	PROG_SPARE	-	No connect	<b>Spare Programmer I/O pin</b> This pin is currently a I/O spare pin which is not used during SPI programming. In order to use this pin, the pin state must be setup in the EQTools <Pre-programming State Machine> tab.

3	PROG_TSCK2	O	XTAL1  Only connect this pin if you are using the SCK2 Clock Output to clock the Target Device during programming !	<b>SCK2 Clock Output (*Optional*)</b> This output signal can be used to supply an external clock signal (SCK2) to the target microcontroller. This function must be enabled in the EQTools <Pre-programming State Machine> tab. If the SCK2 oscillator is not being used, then the pin could be used as a spare output pin.
4	PROG_MOSI	O	MOSI  (except for ATmega103/128/64 – connect to RXD pin instead)	<b>Master Out Slave In</b> This is the SPI data output pin from the programmer. This pin should be connected to the MOSI pin on the Target Microcontroller.
5	N/C	O	N/C	Not connected
6	PROG_MISO	I	MISO  (except for ATmega103/128/64 – connect to TXD pin instead)	<b>Master In Slave Out</b> This is the SPI data input pin to the programmer. This pin should be connected to the MISO pin on the Target Microcontroller.
7	PROG_GND	P	<b>GROUND</b>	<b>Ground Connection</b> Common ground connection between PROGRAMMER and Target System.
8	PROG_SCK1	O	<b>SCK</b>	<b>SPI Serial Clock Output</b> This is the SPI clock output signal used to clock data into / out of the Target Device.
9	PROG_GND	P	<b>GROUND</b>	<b>Ground Connection</b> Common ground connection between PROGRAMMER and Target System.
10	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.  <b>Warning!</b> During 'High Voltage Serial Programming' of the Atmel ATtiny11/12/15 microcontrollers, a +12V Vpp voltage is generated by the programmer on this pin.

## Key

O - Output from programmer to Target Device

I - Input to programmer from Target Device

P - Passive eg. GROUND and power rails

N/C - Not connected

## 4.6 J6(b) - Equinox 10-way Header (ATtiny11/12/15 HV Interface)

This connection method is suitable for interfacing the EPSILON5 programmer to a Target System which features the following:

- Equinox 10-way IDC ISP Header
- An Atmel ATtiny microcontroller eg. ATtiny11/12/15
- Suitable connections to implement 'High Voltage Serial Programming Mode'.

The connector supports programming of the Atmel ATtiny11/12/15 microcontrollers in 'High Voltage Serial Programming Mode'. This mode requires that a +12.0V Vpp voltage is applied to the RESET pin of the target device in order to enter programming mode.

To implement this connection, simply plug the 10-way ISP cable into the **EPSILON5 J6 header** and plug the other end of the cable into the matching header on the Target System. Please ensure that you have made the correct connections from the ISP Header to your Target Device.

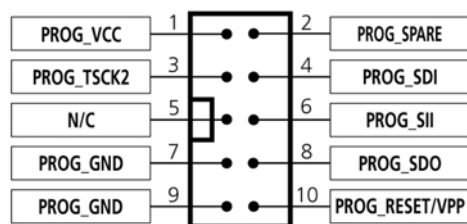


Figure 4.6.1 – Equinox 10-way IDC Header (J6) viewed from above

### Warning!

3. It is possible for +12V to be output on pin 10 of this connector when programming ATtiny11/12/15 devices.
4. Connecting to the wrong ISP Header may cause catastrophic damage to the Programmer & Target System

Pin No	Programmer Pin name	Programmer Input / Output	Connect to pin on Target Device	Description
1	PROG_VCC	P	TARGET_VCC	<b>Target VCC</b> This pin should be connected to the Target System Vcc. This voltage could be used to power the programmer depending on the settings of the power switch/jumper on the programmer.
2	PROG_SPARE	-	No connect	<b>Spare Programmer I/O pin</b> This pin is currently a I/O spare pin which is not used during SPI programming. In order to use this pin, the pin state must be setup in the EQTools <Pre-programming State Machine> tab.
3	PROG_TSCK2	O	XTAL1	<b>SCK2 Clock Output</b> This output pin is used to supply an external clock signal (SCK2) to the target microcontroller.
4	PROG_SDI	O	SDI	<b>Serial Data Input</b>

				This is the serial data output pin from the programmer. This pin should be connected to the Serial Data Input (SDI) pin on the Target Microcontroller.
5	N/C	O	N/C	Not connected
6	PROG_SII	O	SII	<b>SII (Serial Instruction Input)</b> This is the serial data input pin to the programmer from the Target Device. This pin should be connected to the SII pin on the Target Microcontroller.
7	PROG_GND	P	GROUND	<b>Ground Connection</b> Common ground connection between PROGRAMMER and Target System.
8	PROG_SDO	I	SDO	<b>SDO (Serial Data Output)</b> This is the serial data output signal from the Target Device to the programmer. It is an INPUT pin to the programmer.
9	PROG_GND	P	GROUND	<b>Ground Connection</b> Common ground connection between PROGRAMMER and Target System.
10	PROG_RESET/VPP	O	RESET	<b>Target RESET control pin</b> This pin controls the Target Device RESET pin. A +12.0V Vpp voltage will be applied to this pin during programming.  <b>Warning!</b> During 'High Voltage Serial Programming' of the Atmel ATtiny11/12/15 microcontrollers, a +12V Vpp voltage is generated by the programmer on this pin.

### Key

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

## 4.7 J6(c) - Equinox 10-way Header (UART Boot Loader)

This connection method is suitable for interfacing the EPSILON5 programmer to a Target System which features the following:

- Equinox 10-way IDC ISP Header
- An Atmel Wireless T89C51Rx2 / CC01 microcontroller
- A Philips P89C51Rx2 or P89C66x microcontroller
- The relevant connections for in-system programming via the Boot Loader.

To implement this connection, simply plug the 10-way ISP cable into the **EPSILON5 J6 header** and plug the other end of the cable into the matching header on the Target System.

### Please note:

The programmer interfaces to the Target System at TTL voltage levels. If the serial port to your Target System is RS-232, you will need to convert the TTL levels from the programmer to RS-232 using an external convertor.

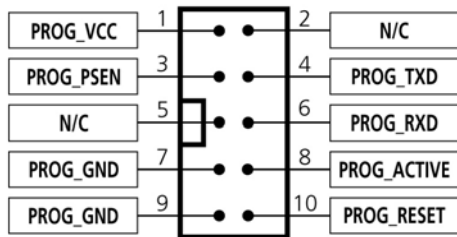


Figure 4.7.1 – Equinox 10-way IDC Header (J8) viewed from above

### Warning!

1. It is possible for +12V to be output on pin 10 of this connector when programming ATtiny11/12/15 devices.
2. Connecting to the wrong ISP Header may cause catastrophic damage to the Programmer & Target System

Pin No	Programmer Pin name	Programmer Input / Output	Connect to pin on Target Device	Description
1	PROG_VCC	P	TARGET_VCC	<b>Target Vcc</b> This pin should be connected to the Target System Vcc. This voltage could be used to power the programmer depending on the settings of the power switch/jumper on the programmer.
2	N/C	-	N/C	<b>Not connected</b>
3	PROG_PSEN	O	PSEN	<b>PSEN pin</b> This pin is driven LOW by the programmer to enter 'Boot Loader' programming mode.
4	PROG_TXD	O	RXD	<b>Programmer RXD (Transmit) pin</b> The programmer transmits serial data on this pin to the Target Device RXD (Receive) pin.

5	N/C	-	N/C	<b>Not Connected</b>
6	PROG_RXD	I	TXD	<b>Programmer RXD (Transmit) pin</b> The programmer receives serial data on this pin from the Target Device TXD (Transmit) pin.
7	PROG_GND	P	Ground Connection	<b>Target / Programmer GROUND connection</b> Common ground connection between PROGRAMMER and Target System.
8	PROG_ACTIVE	O	LED or Interrupt pin	<b>Programmer ACTIVE LED</b> This pin can be driven LOW when a programming action is taking place. The signal can be used to illuminate an LED or trigger an interrupt on the Target Microcontroller.
9	PROG_GND	P	Ground Connection	<b>Target / Programmer GROUND connection</b> Common ground connection between PROGRAMMER and Target System.
10	PROG_RESET	O	RESET	<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.  <b>Warning!</b> During 'High Voltage Serial Programming' of the Atmel ATtiny11/12/15 microcontrollers, a +12V Vpp voltage is generated by the programmer on this pin.

### Key

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



## 4.8 J7 - Atmel 10-way Header (SPI Interface)

This connection method is suitable for interfacing the EPSILON5 programmer to a Target System which features the following:

- Atmel 10-way IDC ISP Header
- An Atmel device which features the 3-wire SPI + RESET Programming Interface

This connector is compatible with the Atmel STK200, STK300 and STK500 Starter Kits. Equinox would recommend not using this connector on new designs, but instead using the 'Equinox 10-way Header' as the Equinox header caters for an External Oscillator, Vpp switching and also features a spare I/O pin for future expandability.

To implement this connection, simply plug the 10-way ISP cable into the **EPSILON5 ISP Header J7** and plug the other end of the cable into the matching header on the Target System.

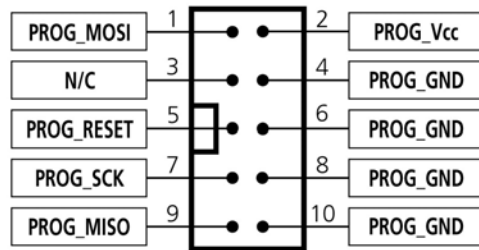


Figure 4.8.1 - Atmel 10-way IDC Header (J7) viewed from above

### Warning!

Connecting to the wrong ISP Header may cause catastrophic damage to the Programmer & Target System

Pin No	Pin name	Programmer Input / Output	Connect to pin on Target Device	Description
1	PROG_MOSI-1	O	MOSI  (except for ATmega103/128/64 – connect to RXD pin instead)	<b>Master Out Slave In</b> This is the SPI data output pin from the programmer. This pin should be connected to the MOSI pin on the Target Microcontroller.
2	PROG_VCC	P	TARGET_VCC	<b>Target Vcc</b> This pin should be connected to the Target System Vcc. This voltage could be used to power the programmer depending on the settings of the power switch/jumper on the programmer.
3	N/C	-	N/C	Not connected
4	PROG_GND	P	GROUND	<b>Ground Connection</b> Common ground connection between the programmer and Target System.

5	PROG_RESET	O	RESET	<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.
6	PROG_GND	P	GROUND	<b>Ground Connection</b> Common ground connection between the programmer and Target System.
7	PROG_SCK1	O	SCK	<b>SPI Serial Clock Output</b> This is the SPI clock output signal.
8	PROG_GND	P	GROUND	<b>Ground Connection</b> Common ground connection between the programmer and Target System.
9	PROG_MISO	I	MISO  (except for ATmega103/128/64 – connect to TXD pin instead)	<b>Master In Slave Out</b> This is the SPI data input pin to the programmer. This pin should be connected to the MISO pin on the Target Microcontroller.
10	PROG_GND	P	GROUND	<b>Ground Connection</b> Common ground connection between PROGRAMMER and Target System.

### Key

O - Output from programmer to Target Device

I - Input to programmer from Target Device

P - Passive eg. GROUND and power rails

N/C - Not connected

## 4.9 J8 - Atmel 10-way JTAG Header (JTAG Interface)

This connection method is suitable for interfacing the EPSILON5 programmer to a Target System which features the following:

- An Atmel device which features a JTAG ISP port e.g. ATmega128 / 323 / 64
- Atmel 10-way IDC JTAG Header
- This is the same header as used on the Atmel JTAG-ICE emulator.

To implement this connection, simply plug the 10-way ISP cable into the **EPSILON5 ISP Header J8** and plug the other end of the cable into the matching header on the Target System.

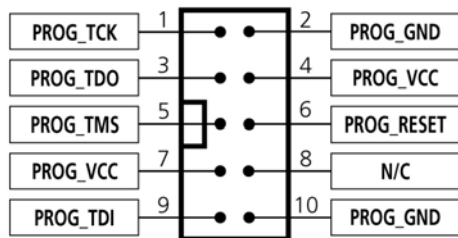


Figure 4.9.1 - Atmel 10-way IDC Header (J8) viewed from above

### Warning!

Connecting to the wrong ISP Header may cause catastrophic damage to the Programmer & Target System

Pin No	Programmer Pin name	Programmer Input / Output	Connect to pin on Target Device	Description
1	PROG_TCK	O	TCK	<b>JTAG TCK – Test Clock Signal pin</b> Clock signal from programmer to Target Device JTAG port.
2	PROG_GND	P	GROUND	<b>Ground Connection</b> Common ground connection between Programmer and Target System.
3	PROG_TDO	I	TDO	<b>JTAG TDO – Test Data Output pin</b> Data signal from Target device JTAG port to programmer.
4	PROG_VCC	P	TARGET_VCC	<b>Target Vcc Connection</b> - Pins 4 + 7 are physically connected inside the programmer. - Connects to Vcc rail of Target System. - Pin referred to as VTref on Atmel JTAG-ICE.
5	PROG_TMS	O	TMS	<b>JTAG TMS – Test Mode Select pin</b> Mode Select Signal from programmer to Target Device JTAG port.
6	PROG_RESET	O	RESET	<b>Microcontroller RESET control signal</b> This pin connects to the main RESET pin of the Target Microcontroller. This pin is not strictly needed for JTAG programming, but it can be used to RESET the Target Device before and after programming.

7	PROG_VCC	P	TARGET_VCC	<b>Target Vcc Connection</b> - See pin 4 - Pins 4 + 7 are physically connected inside the programmer.
8	N/C	O	N/C	<b>Not Connected</b>
9	PROG_TDI	O	TDI	<b>JTAG TDI – Test Data Input pin</b> Data signal from programmer to Target Device JTAG port.
10	PROG_GND	P	GROUND	<b>Ground Connection</b> Common ground connection between PROGRAMMER and Target System.

### Key

O - Output from programmer to Target Device

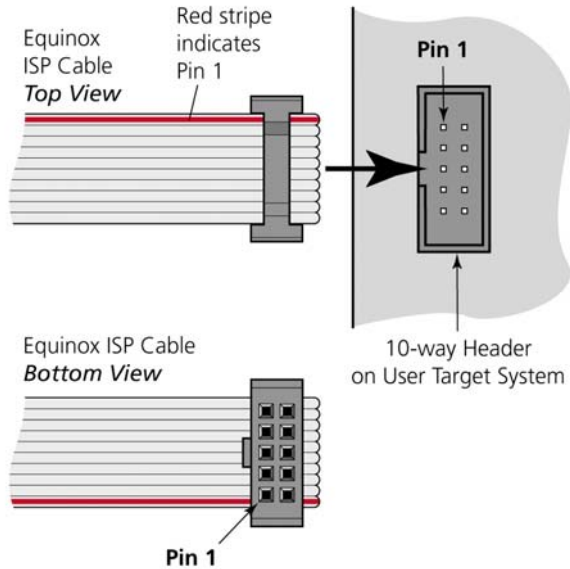
I - Input to programmer from Target Device

P - Passive eg. GROUND and power rails

N/C - Not connected

## 4.10 ISP Cable considerations

The programmer is supplied with a single 10-way ISP Cable as standard. This cable is terminated with a 10-way IDC 0.1" female polarised plug at each end. The cable is wired as a so-called 'straight-through' cable with pin 1-1, 2-2 etc. The polarised ISP Header ensures that the ISP Cable cannot be plugged in the wrong way around by mistake. If you are planning to design such a header onto your Target System, it is strongly recommended that a similar polarised header is used. This will help to prevent accidental damage to both the programmer and the Target System.



Pin 1 of the ISP cable can be determined by looking for a small arrow on the plastic part of the ISP female plug. If the cable has a RED stripe on one cable, this usually also indicates pin 1.

If you are using the programmer in a production environment and constantly plugging / unplugging the ISP cable into/from the Target System, you may find that the cable eventually fails. Spare ISP cables can be ordered from Equinox in this eventuality.