FS2009 USB

Portable ISP Programmer for Field-Service and Production applications

User Guide

EQUINOX TECHNOLOGIES
EQUINOX TECHNOLOGIES UK LIMITED
The Embedded Solutions Company

Contents

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.

Atmel$^{\text{TM}}$ and AVR$^{\text{TM}}$ are trademarks of the Atmel Corporation

Microsoft, MS-DOS, Windows$^{\text{TM}}$, Windows 95$^{\text{TM}}$, Windows 98$^{\text{TM}}$, Windows XP$^{\text{TM}}$ and Windows NT4$^{\text{TM}}$ 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 **'FS2003 Programmer'**, **'FS2009 Programmer'** and **'FS2009USB Programmer'** are CE Approved Products. They are designed for use in an ESD controlled environment i.e. in 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 programmer 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.

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

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

i. User Manual

ii. 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.

iii. Internet Web Site
The support page for all Equinox ISP Programmers can be found at:
http://www.equinox-tech.com/products/downloadsearch.asp

iv. E-mail
Please e-mail any technical support questions about this product to:
support@equinox-tech.com

v. 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

i. Overview

This manual provides an overview of the contents of the FS2009USBUSB / FS2003 Programming Systems 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

Software:	
EQTools	**EQTools Script Builder – Manual** This software is used to create and upload ***'Standalone Programming Projects'*** to the programmer. The following sources of documentation are available for this software: • Installation and Getting Started Guide (pdf manual) • Help file
	ASCII Text Communications Protocol – Application Note This protocol can be used to control the programmer from an external controller via RS-232. The following sources of documentation are available for this protocol: • Application Note – AN110.
	Upload Wizard - Standalone Project Upload Utility This software utility is used to upload ***'Standalone Programming Projects'*** to any Equinox programmer. These projects can then be used in Standalone Mode, i.e. without a PC. • Please follow the on-screen instructions within the Upload Wizard utility itself. • Application Note - AN117
	SPI In-System Programming (ISP) – Application Note This application note describes how to develop and implement ISP support for the Atmel AVR microcontroller family. This is a standard feature with the FS2009USB programmer. The following sources of documentation are available for this software: • Application Note – AN101 • Device Support List

	JTAG In-System (ISP) Upgrade – Application Note This license upgrade enables the FS2009USB Programmer to support high-speed In-System Programming (ISP) of the Atmel ATmega microcontroller family using the JTAG algorithm. The following sources of documentation are available for this software: • Application Note – AN105 • Device Support List
	Atmel AT91SAM7 In-System (ISP) Upgrade – Application Note This license upgrade enables the FS2009USB Programmer to support high-speed In-System Programming (ISP) of the AT91SAM7 FLASH microcontroller family using the JTAG algorithm. The following sources of documentation are available for this software: • Application Note – AN122 • Device Support List
	I2C Serial EEPROM In-System (ISP) Upgrade – Application Note This license upgrade enables the FS2009USB Programmer to support In-System Programming (ISP) of I2C Serial EEPROM devices from many manufacturers The following sources of documentation are available for this software: • Application Note – AN118 • Device Support List
	Error Message Descriptions This document lists all the possible error messages which can be generated by the EQTools / ISP-PRO applications.

ii. Documentation and software for the FS2009USB programmer

In line with our policy of continuous improvement, the software and associated documentation for this product are updated on a regular basis. You can download the latest software, firmware, User Manuals and application notes for the FS2009USB, FS2009 & FS2003 programmers from the following pages on the Equinox website:

1. FS2009USB programmer
See http://www.equinox-tech.com/products/details.asp?ID=1561&displ=tl

2. FS2009USB programmer
See http://www.equinox-tech.com/products/details.asp?ID=1303&displ=tl

3. FS2003 programmer
See http://www.equinox-tech.com/products/details.asp?ID=370&displ=tl

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

iii. Device algorithm - Application notes

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

Application Note	Device Family	Programming Interface
AN100	Atmel - AT89Sxxxx FLASH microcontrollers	SPI
AN101	Atmel - AVR FLASH microcontrollers via the SPI Interface	SPI
AN105	Atmel - AVR FLASH microcontrollers via the JTAG Interface	JTAG
AN118	Generic I2C 24xxx Serial EEPROM memories	I2C
AN122	Atmel - AT91SAM7 ARM7 FLASH microcontrollers	JTAG
AN128	NXP – LPCxxx ARM7 FLASH microcontrollers	JTAG
AN130	Zensys – ZWxxx – Z-WAVE Series devices	SPI

These application notes can be found in PDF format on the CD-ROM which was supplied with the programmer. You can also find the very latest versions on the ***"FS2009USB, FS2009 & FS2003 Download Page***" on the Equinox website.

iv. Programmer related - Application notes

The table below lists the Application Notes available for the FS2009USB, FS2009 & FS2003 programmer range which describe the USB driver installation, the different control methods available, firmware update procedure and Oscillator Calibration procedure.

Application Note	Description
AN109	Remote Application Control of Equinox ISP Programmers using ISP-PRO Utility
AN110	ASCII Text Control (ATC) Protocol for Remote Control of Equinox Programmers
AN111	ConsoleEDS Protocol for Remote Control of Equinox Programmers
AN112	Firmware Update instructions for Equinox ISP Programmers
AN114	Accurate on-chip Oscillator Calibration for Atmel AVR microcontrollers
AN121	Equinox EQTools – Release Notes

These application notes can be found in PDF format on the CD-ROM which was supplied with the programmer. You can also find the very latest versions on ***“FS2009USB, FS2009 & FS2003 Download Page”*** on the Equinox website.

1.0 Programmer Overview / Specifications

1.1 Programmers covered in this manual

This manual covers the following Equinox programmers:

- FS2009USB
- FS2009

1.2 Comparison of the FS2009USB and FS2009 programmers

The FS2009USB programmer is exactly the same hardware as the FS2009 programmer except that the ***RS232 COM port*** on the FS2009 has been replaced by a ***USB port*** on the FS2009USB. This allows the FS2009 programmer to support high-speed USB communications with a host PC. In most circumstances, it is also possible to power the FS2009USB programmer internal electronics from the PC USB port. It is NOT possible to power an attached target board when powering the programmer from the PC USB port.

The FS2009USB programmer will load and execute exactly the same ***'Standalone Programming Projects'*** as the FS2009 programmer. It is also backwards compatible with the FS2003 programmer.

1.3 Comparison of the FS2009 & FS2003 programmers

The FS2009 & FS2003 programmers are based on exactly the same hardware design. The only difference is that the FS2009USB features a more up-to-date microcontroller which has more on-chip firmware space and more RAM. This allows the FS2009 to hold more algorithms in memory and thus support more devices compared to the FS2003 programmer. It also makes many of the algorithms significantly faster as the extra RAM allows the algorithm to be optimised.

The advantages of the FS2009 programmer are as follows:

Programming speed:

- Faster AVR algorithms in PC controlled modes (EDS, ConsoleEDS and ISP-PRO)
- Faster AT91SAM7 algorithms in PC controlled modes (EDS, ConsoleEDS and ISP-PRO)

Additional device support (only available on FS2009USB):

- Atmel AT91SAM7 – ARM7 FLASH microcontrollers
- NXP LPC21xx - ARM7 FLASH microcontrollers
- I2C Serial EEPROMs

Availability:

- The FS2009USB programmer is available from stock.
- The FS2009 programmer has been replaced by the FS2009 programmer.
- The FS2003 is only available by special order and is subject to an MOQ.

1.4 FS2009USB, FS2009 & FS2003 Project Compatibility

As the FS2003 and FS2009 / FS2009USB programmers are based on the same hardware design, any ***'Standalone Programming Projects'*** developed for use on the FS2003 programmer can also be used on the FS2009 / FS2009USB programmer without requiring any modifications.

- Projects designed for the FS2003 programmer can be uploaded to the FS2009 / FS2009USB programmer without any modifications.
- Projects designed for the FS2009 / FS2009USB programmer can be uploaded to the FS2003 programmer as long as they are not for any of the device families which are only supported on the FS2009 / FS2009USB programmer.
- The latest version of EQTools / ConsoleEDS or ISP-PRO is required which supports the newer FS2009 / FS2009USB programmer.

1.5 FS2009USB - Main Features

Wide ranging Device Support capability

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

High-speed Programming

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

Supports most ISP Protocols

- SPI, JTAG, I2C (2-wire). ATtiny HV SCI

High-speed JTAG port

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

High-speed SPI port

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

Fully ESD and Over-voltage protected I/O

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

Supports programming at Target Voltages down to 3.0V

- An optimised driver circuit delivers fast clean programming waveforms from 3.0 to 5.0V.

Excellent Host Control connectivity

- 1 x USB Port (mini-USB connector)

Standalone Operation

- Programmer can operate in ***'Standalone Mode'*** i.e. without PC Control.
- Programmer can be controlled via 4-button Keypad / Display.

Supports up to 64 independent 'Standalone Programming Projects'

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

Firmware upgradeable

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

Compact physical size ideal for held-held production or field use

- The programmer is designed to be portable so it can be used for production or field use.

1.6 FS2003 - Programmer versions

The FS2003 programmer is only available as a single version – the FS2003(UN).

Programmer version	Description
FS2003(UN)	• Supports programming of Atmel AVR devices **via the SPI interface only.** • Supports programming of Atmel ATtiny devices via the ‘**High-voltage Serial Programming Mode**’. • Supports Atmel AT89S, AT89C51Rxx, AT89Sxxxx microcontrollers. • Supports NXP P89X51Rx2 devices • Supports Zensys 100 / 200 / 300 series devices.

Please note:

- The FS2003 is now a discontinued product. Please use the FS2009USB programmer as the replacement programmer.
- Equinox will continue to add new devices to the FS2003 programmer where possible.

1.7 FS2009USB - Programmer versions

The FS2009USB programmer is now available in three different versions, each offering different device support. Please refer to the table below for an overview of each version.

Programmer version	Description
FS2009USB(UN)	• Supports programming of Atmel AVR devices **via the SPI interface only.** • Supports programming of Atmel ATtiny devices via the ‘**High-voltage Serial Programming Mode**’. • Supports Atmel AT89S, AT89C51Rxx, AT89Sxxxx microcontrollers. • Supports NXP P89X51Rx2 devices • Supports Zensys 100 / 200 / 300 series devices.
FS2009USB(AVR-JTAG)	• Dedicated programmer supporting programming of the Atmel AVR family **via the JTAG interface only**.
FS2009USB(ARM)	• Dedicated programmer supporting many ARM microcontroller devices. • Atmel ARM7 microcontroller family including AT91 SAM7S SAM7SE / SAM7A / SAM7L / SAM7X / SAM7XC ARM7 families • NXP LPC21xx ARM7 families • ST STM32F100Rx Cortex-M3 ARM family • Includes a special 20-way IDC connector cable for interfacing to AT91SAM7 Target Boards.

Please note:
It is possible to upgrade any version of the FS2009USB programmer so it supports other devices as well.

1.8 FS2009USB – Device support for each Programmer version

The table below details which device families are supported by each version of the programmer.

Device Family	Programming Interface	FS2009USB (UN)	FS2009USB (AVR-JTAG)	FS2009USB (ARM)
Atmel AVR (SPI): - AT90S - AT90USB - AT90CAN - ATmega - ATmegaRF - ATtiny LV	LV SPI	YES YES YES YES YES YES	- - - - - -	- - - - - -
Atmel AVR (HV): - ATtiny HV	HV Serial (+12V)	YES	-	-
Atmel AVR (JTAG): - AT90USB - AT90CAN - ATmega - ATmegaRF	AVR JTAG	- - - - -	YES YES YES YES	- - - - -
Atmel AT91SAM7 - AT91SAM7A - AT91SAM7L - AT91SAM7S - AT91SAM7SE - AT91SAM7X - AT91SAM7XC	ARM JTAG	- - - - - -	- - - - - -	YES YES YES YES YES YES
Atmel 8051 - T89C51Rx2 - AT89C51xxx	UART Boot Loader	YES YES	- -	- -
Atmel 89S 8051 - AT89S82xx - AT89Sx051	LV SPI	YES YES	- -	- -
NXP 8051 - P89X51Rx2	UART Boot Loader	YES		-
NXP LPC ARM7 - LPC210x - LPC213x - LPC214x	ARM JTAG	UPGRADE	UPGRADE	YES YES YES
ST - STM32F100Rx	ARM JTAG	UPGRADE	UPGRADE	YES
Zensys - ZW100 / 200 / 300 series	LV SPI	YES		-
All manufacturers 24xxx Serial EEPROM Memories	I2C	UPGRADE	UPGRADE	UPGRADE

1.9 Programming Interface to Target Device

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

#	Programming Interface	Device Family	Description / Comment
1	**LV SPI**	AT89S AT90S AT90USB AT90CAN ATmega ATmegaRF ATtiny Zensys	**Atmel Low Voltage Serial Programming Mode** • Atmel 3-wire SPI interface (SCK, MOSI, MISO) • Target Device RESET • Works at target voltage (no Vpp required)
2	**HV SPI (+12V Vpp)**	ATtiny	**Atmel High Voltage Serial Programming Mode** • Atmel 4-wire SPI Interface (SCK, MOSI, MISO, MOSI2) • +12V Vpp is applied to the RESET pin to enter High Voltage Serial Programming Mode • Requires different connections to Target Device compared with LV SPI. • Requires all pins of target device to be accessible by the programmer.
3	**UART Boot Loader**	T89C51Rx2 P89X51Rx2	**Atmel / Philips UART Boot Loader Interface** • Uses RXD, TXD, PSEN, and Target Device RESET pin • TTL Levels from programmer to Target System
4	**JTAG (Atmel AVR)**	ATmega ATmegaRF AT90USB AT90CAN	**Atmel ATmega AVR – JTAG ISP Interface** • Uses same connector as Atmel JTAG ICE • A chargeable license upgrade is required to enable the ATmega JTAG ISP libraries.
5	**JTAG (Atmel AT91SAM7)**	AT91SAM7A AT91SAM7L AT91SAM7S AT91SAM7SE AT91SAM7X AT91SAM7XC	**Atmel AT91SAM7 ARM – JTAG ISP Interface** • Programming of Atmel AT91SAM7 devices requires a chargeable license upgrade (FS2009USB-UPG15) • This upgrade includes a special 20-way ISP Connector Cable compatible with the 20-way IDC connector found on ARM Target Systems.
6	**JTAG (NXP ARM7)**	LPC21xx	**NXP LPX21xx ARM7 – JTAG ISP Interface** • Programming of NXP LPC21xx ARM7 microcontrollers requires a chargeable license upgrade (FS2009USB-UPG27) • This upgrade includes a special 20-way ISP Connector Cable compatible with the 20-way IDC connector found on ARM Target Systems.

Please note:
The FS2009USB is available as various different versions which support different devices as standard. Not all device libraries may be enabled as standard with the programmer you have purchased.

1.10 System Contents

The FS2009USB programmer comes complete with an external mains power supply, PC Driver Software and cables. Please see the full contents list detailed below.

Hardware

- FS2009USB ISP Programmer
- External Power Supply Adaptor (9V DC regulated @ 1.7A)

Cables

- USB connection cable (mini-USB to USB-A)
- 6-way IDC ISP cable
- 10-way IDC ISP cable
- DC Power Cable (2.5mm jack plug to bare wire ends)

Miscellaneous

- 2 x Jumper Links

Software

(Supplied on an Equinox CD-ROM)

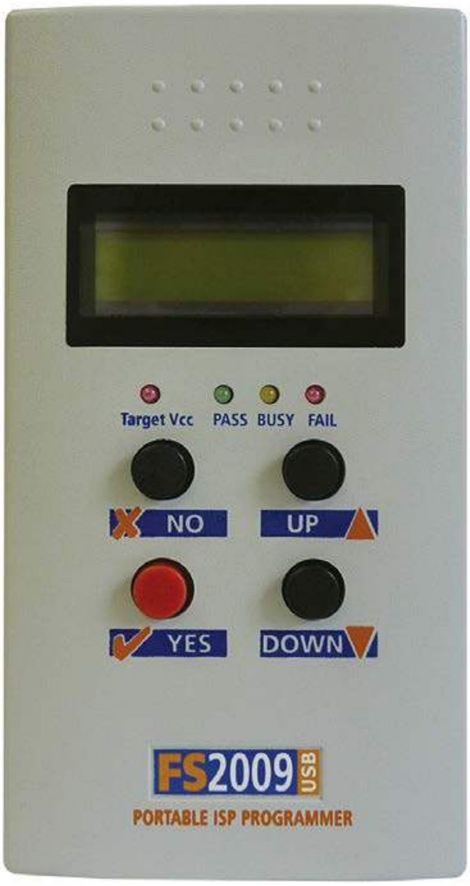
- **EQTools**
  (Project Management Utility for Equinox Production ISP Programmers)

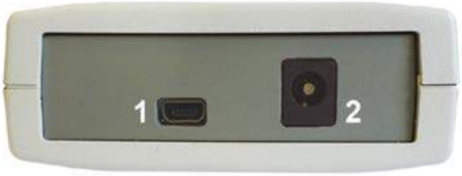
Documentation

- User Guide (printed)

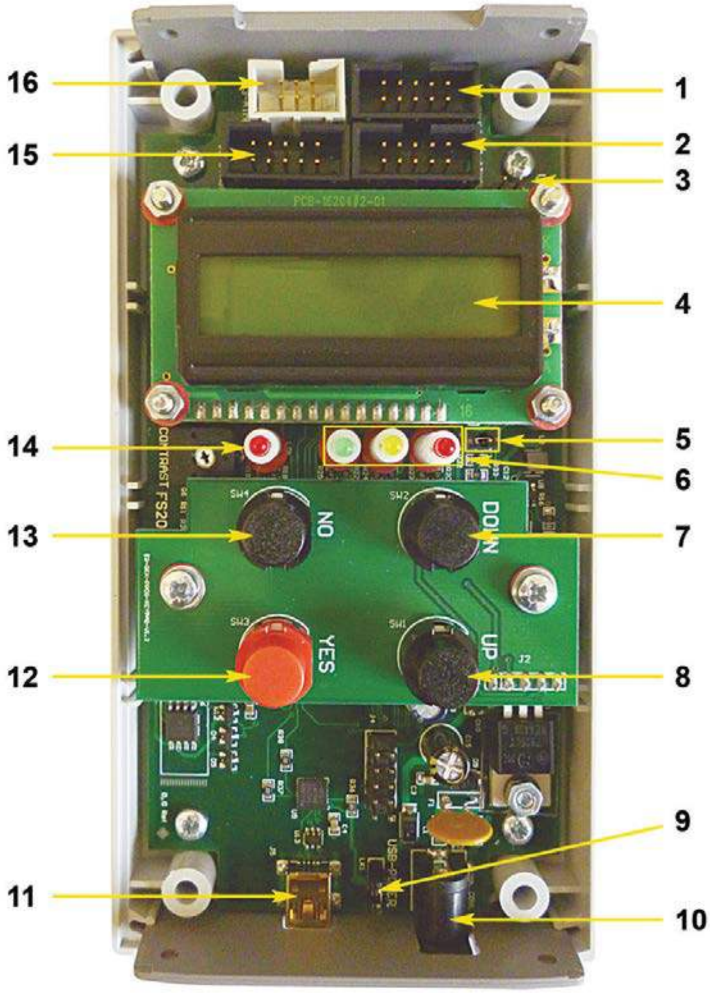
Please note:
The FS2009USB(ARM) version of the FS2009USB programmer comes with a special cable assembly which allows the programmer to connect to any ARM Target Board which is fitted with a 20-way IDC connector. See Appendix 1 for details of this cable.

1.11 Hardware Overview (external layout)

<table>
<tr><td rowspan="2"> </td><td colspan="2"><b>Top Panel</b></td></tr>
<tr><td colspan="2"><b>ISP Cable Slot</b>
The ISP cable plugs into relevant <b>'ISP Header'</b> internally and then exits the case through this slot.</td></tr>
<tr><td rowspan="4"> </td><td colspan="2"><b>Front Panel</b></td></tr>
<tr><td> </td><td><b>&lt;Target Vcc&gt; LED</b>
- Indicates when 'Target Vcc' is ON
- Does not indicate 'Programmer Power' unless Jumper J9 is fitted.

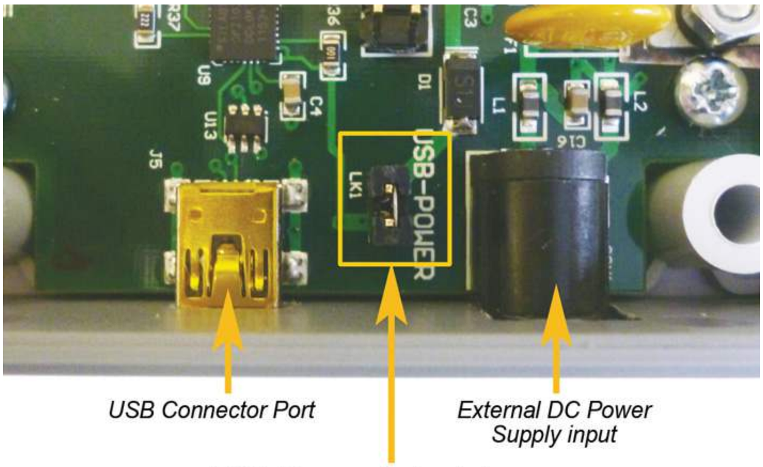
<b>Status LED's:</b>
<b>&lt;PASS&gt; LED</b>
<b>&lt;BUSY&gt; LED</b>
<b>&lt;FAIL&gt; LED</b></td></tr>
<tr><td> </td><td><b>&lt;NO&gt; Button</b>
- Used to cancel a programming operation.</td></tr>
<tr><td> </td><td><b>&lt;YES&gt; Button</b>
- Used to start an 'autoprogram' operation.</td></tr>
<tr><td rowspan="2"> </td><td colspan="2"><b>Bottom Panel</b></td></tr>
<tr><td colspan="2">1. mini-USB Connector Port
2. External DC Power Supply Input (CON1)</td></tr>
</table>

1.12 Hardware Overview (internal layout)



Hardware

1. Equinox 10-way ISP (SPI / UART) Header (J6)
2. Atmel 10-way ISP (SPI) Header (J7)
3. Jumper (J9) – Vcc Link
4. Liquid Crystal Display (LCD)
5. Enable LCD Backlight Jumper (J10)
6. Status LED’s
7. <Down> Key (SW2)
8. <Up> Key (SW1)
9. USB Power select Jumper (LK1)
10. External DC Power Supply Input (CON1)
11. mini-USB Connector port (J5)
12. <YES> Key (SW3)
13. <No> Key (SW4)
14. Target Vcc Present – LED
15. Atmel 10-way JTAG Header (J8)
16. Atmel 6-way ISP (SPI) Header (J3)



USB Power Select Jumper

USB Power Selection Jumper

If the **‘USB-POWER’** link is fitted, this allows the programmer to be powered from the PC USB port.

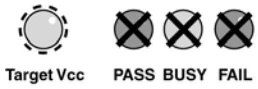
Important notes

- It is not possible to power the Target System via the PC USB port.
- Some PC USB ports may not have sufficient power to supply the programmer.

1.13 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 sect
1	**Target Device Support**	See **Device Support List.** For the latest version please see website.	
2	**Target Device Programming Interfaces**	The FS2009USB programmer supports the following programming interfaces: • Atmel Low Voltage SPI • Atmel High Voltage SPI (+12V Vpp) • Atmel 8051 Boot loader ISP • NXP 8051 - Boot loader ISP • Zensys 3-wire SPI Interface • Atmel AVR JTAG (chargeable update) The FS2009USB programmer supports the following additional programming interfaces: • Atmel AT91SAM7 JTAG (requires additional cable) • NXP LPC21xx LPC ARM JTAG • ST STM32 – ARM JTAG • I2C for programming Serial EEPROMs (chargeable upgrade)	1.8
3	**Operating modes**	The programmer supports the following operating modes: As standard: • ***EDS Development*** Mode (PC controlled) • ***Project Upload Mode*** using Upload Wizard (PC controlled) • ***Standalone Mode*** - Keypad + LCD operation • ***Standalone Mode*** – Run Target	3.4
4	**On-board FLASH Memory Store**	32 Mbits (4 MBytes) FLASH Memory	
5	**Project storage in Memory Store**	64 x Independent Programming Projects	
6	**Keypad entry**	4 x Push Button - <YES>, <NO>, <UP>, <Down>	
7	**Status LED's** 	3 x Status LED's located on the programmer: <PASS>, <BUSY>, <FAIL> These LED's indicate the current status of the programmer.	
8	**Programmer Display**	2 x 16 character Back-lit LCD (backlight configurable)	

9	**Target Vcc Voltage Detection LED** 	• Target Vcc LED will illuminate when Target Vcc voltage is detected. • Threshold detection voltage: approx 3.1V	
10	**PC Control Software**	The programmer can be controlled using: As standard: • EQTools • Upload Wizard (integrated with EQTools) • ConsoleEDS EVAL – Command Line control	3.1
11	**PC requirements**	• PC running Windows 95 / 98 / 2000 / ME / NT4 / XP / Vista / Windows 7 • Spare PC USB Port	
12	**Power Supply Options**	There are 4 possible modes for powering the Programmer / Target System: • Programmer is powered from the Target System (3.0 – 5.0V) • Programmer powers the Target System (+5.0V only) • Programmer and Target System are independently powered (Programmer at 9.0 -12V, Target System at 3.0 – 5.0V) • Programmer is powered from PC USB port (programmer only, not the target system)	2.3
13	**Voltage range**	• Device Programming voltage range: 3.0 – 5.0V • Programmer Internal Vcc Voltage: 5.0V • Possible Target System voltage range: 3.0 – 5.0V	
14	**Vpp Voltage Generator**	• On Board +12.0V Vpp Voltage Generator (Applies +12.0V to RESET pin for Atmel ATtiny HV programming)	
15	**DC Power Connector (jack socket)**	DC Power Connector • 2.5 mm Jack Socket – Centre Positive • 9.0 – 12.0V DC regulated @ 500mA (worst case) • Mates with Power Supply Adaptor supplied with programmer • DC Power Lead also supplied with programmer if a customer power supply is to be used.	
16	**Target Voltage measurement**	• The programmer can measure the Target Voltage in either EDS or Standalone Mode. • Measurement Voltage range: 3.0 – 5.0 Tolerance: +/-3%	
17	**Max. current through programmer**	This refers to the maximum current which the programmer can supply from the internal Vcc rail Max current = 300 mA.	
18	**Target SPI Frequency**	The programmer supports three SPI speeds: • SLOW SPI: 0 – 490.2 kHz (user selectable)	

		• MEDIUM SPI: 0 – 490.2 kHz (user selectable) • FAST SPI: Disabled (SPI speeds are estimations only due to uneven mark/space ratio and non-continuous waveforms)	
19	**Target JTAG Frequency**	The programmer supports a user-defined JTAG frequency: • SLOW: 0 – 833 kHz	
20	**Target UART Speed**	• 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 setting is completely independent from the BAUD rate settings for the PC communicating with the programmer.	
21	**ISP Headers**	The programmer support all the popular ISP Header as standard: • Atmel 10-way (SPI) • Atmel 6-way (SPI) • Atmel 10-way (JTAG) • Equinox 10-way (SPI+UART) It is also possible to program ARM microcontrollers using a separate plug-in cable assembly.	4.0
22	**Frequency generator output (SCK2)**	The FS2009USB is capable of outputting a continuous square wave at various frequencies on the programmer SCK2 pin. The following pre-set frequencies can be selected via the EQTools Programming Project: 1.8432 MHz, 921.6 kHz, 614.4 kHz, 460.9 kHz, 386.6 kHz, 307.6 kHz, 230.4 kHz,167.6 kHz, 102.4 kHz, 51.2 kHz This clock frequency could be used for the following purposes: • To toggle the STROBE pin on a Watchdog Timer IC to stop the IC from timing out during the In-System Programming operation. • To provide a clock to an Atmel ATmega microcontroller to resurrect a device which has an incorrectly configured Internal Oscillator. Please refer to the 'EQTools – Getting Started Guide' for full details of setting up the SCK2 oscillator.	
23	**Target Run**	The programmer is capable of automatically running firmware which has been programmed into a Target Device and waiting until the MISO pin is asserted by the Target Device.	
24	**Pre-Programming State Machine**	The programmer has 5-state 'Pre-programming State Machine' which allows the I/O lines of the programmer to be sequenced in order to place the Target Device into Serial Programming Mode.	
25	**PC Connection**	**USB Port** • Mini-USB connector • Connects to spare PC USB port	1.15

26	**Temperature range**	0 to 50 deg. C	
27	**Dimensions**	189H x 80L x 31D mm	
28	**Shipped Weight**	0.75 kg	

1.14 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.

CON1 – pin-out

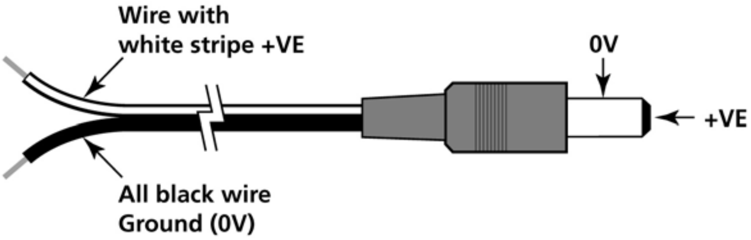
Description	Voltage
Centre pin	+9.0V to +12.0V from external power supply
Outer	0V (Ground connection)

1.15 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 9.0 and 12.0V!



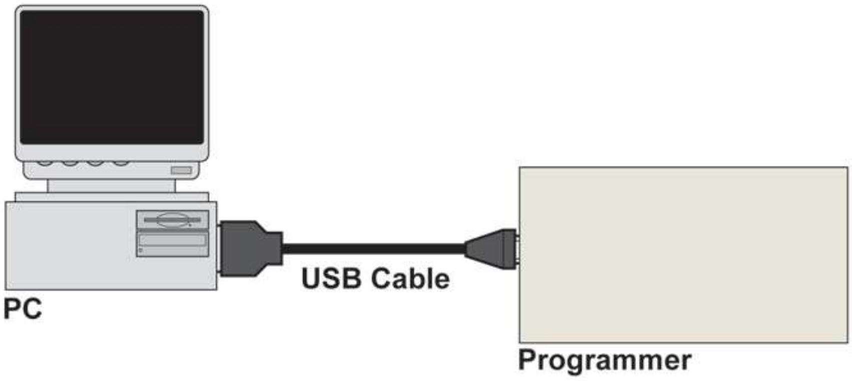
1.16 J5 – USB Communications Port

1.16.1 Connecting the programmer to the PC USB port

The programmer communicates with a PC via the USB Communications Port (J5). A suitable USB cable is supplied with the programmer.

The programmer is supplied with the following cable:

1. **USB Cable**



2.0 Hardware Installation

2.1 Overview

This section details how to set up the programmer hardware including power supply, earthing requirements, PC Serial Port Connection and ISP Header Selection.

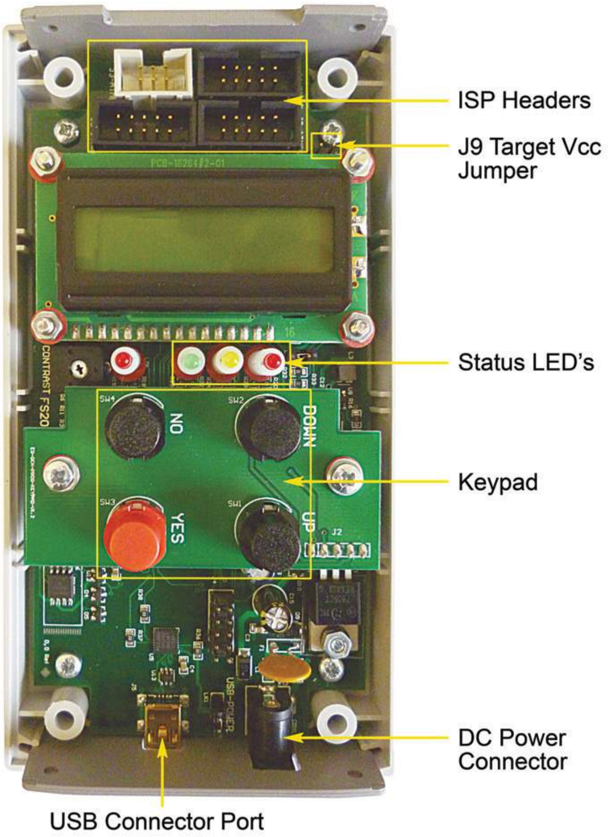
The ***FS2009USB*** programmer is available as three different versions, each with different device support and set up requirements. Please refer to the table below for instructions on how to set up each version.

Programmer version	Instructions
FS2009USB(UN)	• Supports programming of Atmel AVR devices **via the SPI interface only.** • Supports programming of Atmel ATtiny devices via the **'High-voltage Serial Programming Mode'.** • Supports Atmel AT89S, AT89C51Rxx, AT89Sxxxx microcontrollers. • Supports NXP P89X51Rx2 devices • Supports Zensys 100 / 200 / 300 series devices. • Please refer to section 2.2 onwards
FS2009USB(AVR-JTAG)	• Dedicated programmer supporting programming of the Atmel AVR family **via the JTAG interface only**. • Please refer to section 2.2 onwards
FS2009USB(ARM)	• Dedicated programmer supporting many ARM microcontroller devices. • Atmel ARM7 microcontroller family including AT91 SAM7S SAM7SE / SAM7A / SAM7L / SAM7X / SAM7XC ARM7 families • NXP LPC21xx ARM7 families • ST STM32F100Rx Cortex-M3 ARM family • Includes a special 20-way IDC connector cable for interfacing to ARM Target Boards.

2.2 ESD Precautions

- 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.
- The programmer I/O lines are ESD protected, but it always good practice to take proper ESD precautions. This can help to avoid ESD damage to either the programmer or Target System.

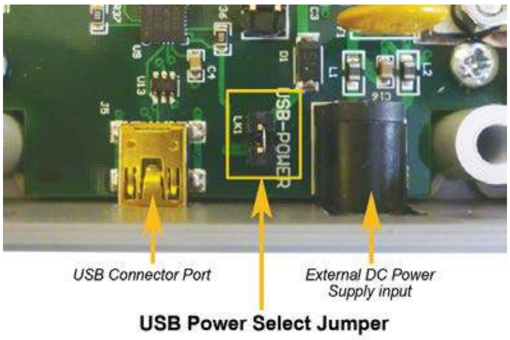
2.3 Removing the programmer cover

a	Remove programmer from all packaging and place display-side down on a flat surface	
b	Remove all four screws from the back of the programmer (if fitted)	The screws will be supplied in a separate small plastic bag if this is a new programmer.
c	Remove programmer cover to reveal internal electronics	

2.4 Powering the Programmer / Target System

2.4.1 Select the required method of powering the Programmer / Target System

The programmer supports the following power supply methods:

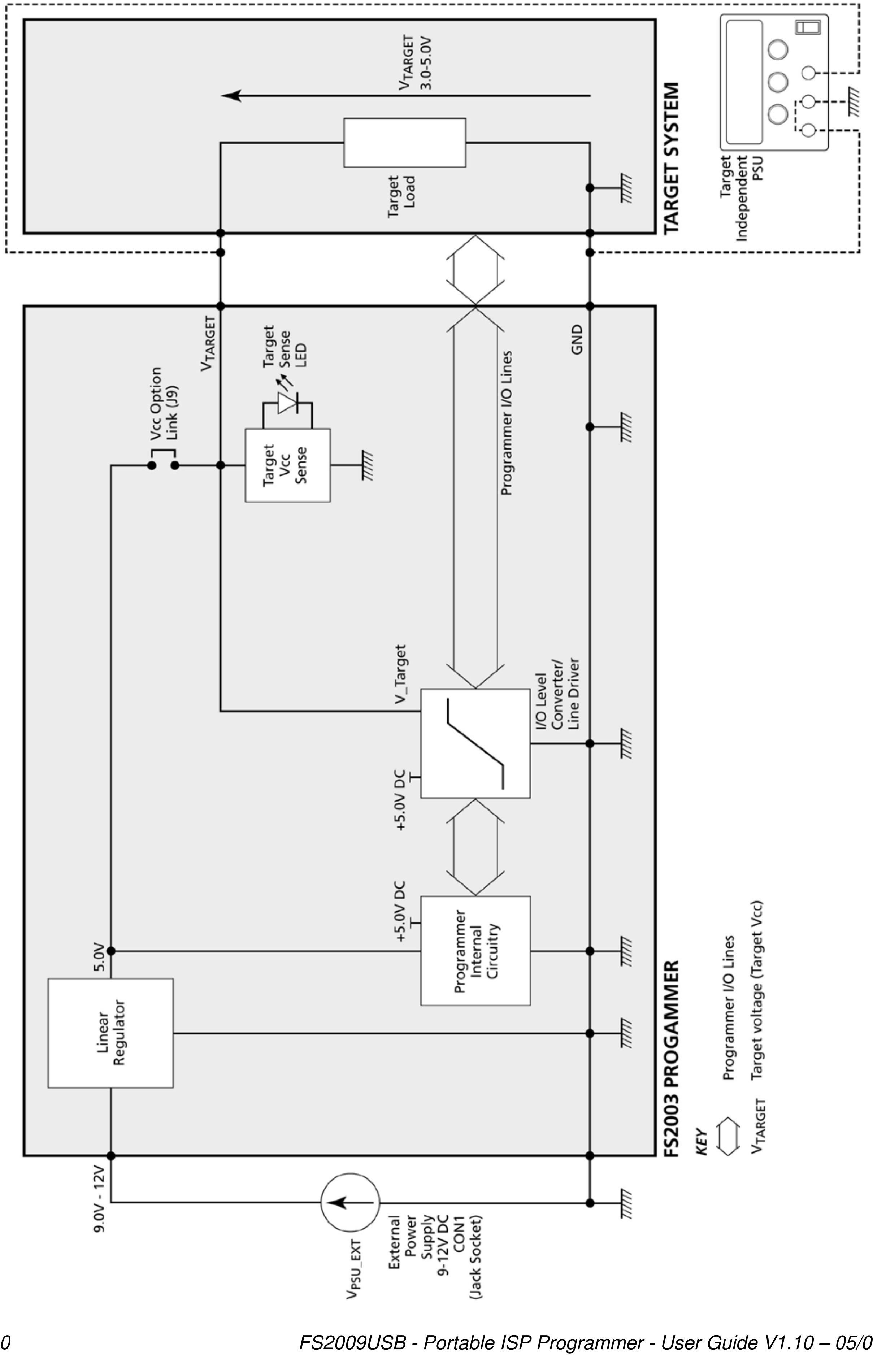
Sect	**Power Supply method**	**Jumper J9**	**Target Vcc (V)**	**CON1 Jack Socket (V)**
2.4.2	Powering the Target System at +5.0V from the programmer (external mains power supply adaptor)	FITTED	5.0V from programmer	9.0V
2.4.3	Powering the Target System at +5.0V from the programmer (external bench power supply)	FITTED	5.0V from programmer	9.0V
2.4.4	Powering the programmer from the Target System at 3.0 – 5.0V	FITTED	3.0 – 5.0	**Do NOT connect**
2.4.5	Programmer and Target System are Independently powered	**NOT FITTED**	3.0 – 5.0	9.0V
2.4.6	Powering the programmer from the USB port  USB Connector Port External DC Power Supply input USB Power Select Jumper	**NOT FITTED**	3.0 – 5.0	**Do NOT connect**

Refer to fig.2.3.1 for a diagram showing the power supply methods of the FS2009USB programmer. Please choose the relevant powering method from the table above and then refer to the relevant section 3b, 3c, 3d, 3e below.

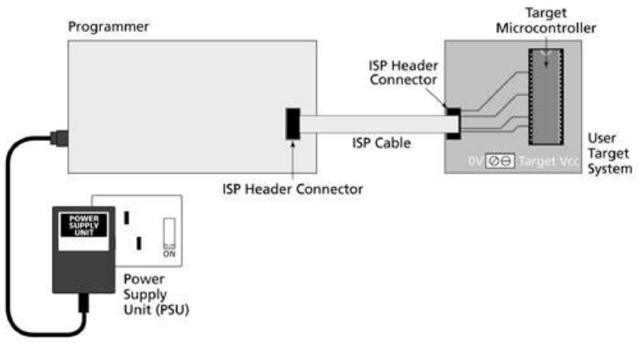
Key:

- **Target Vcc** – This is the voltage of the Target System
- **CON1 Jack Socket** – This is the DC Jack socket on the bottom of the programmer.
- **Programmer Vcc –** This is the +5.0V supply which powers all the internal electronics of the programmer.
- **Jumper J9** – This jumper connects **the 'Programmer Vcc (+5V)'** supply to the **'Target Vcc'** supply.

Fig. 2.4.1 FS2009USB – Power Supply Overview Schematic

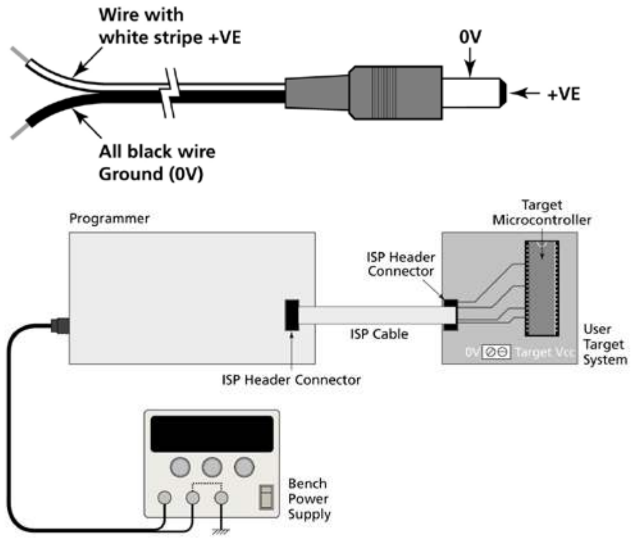


2.4.2 Powering the Target System at +5.0V from the programmer (external mains power supply adaptor)



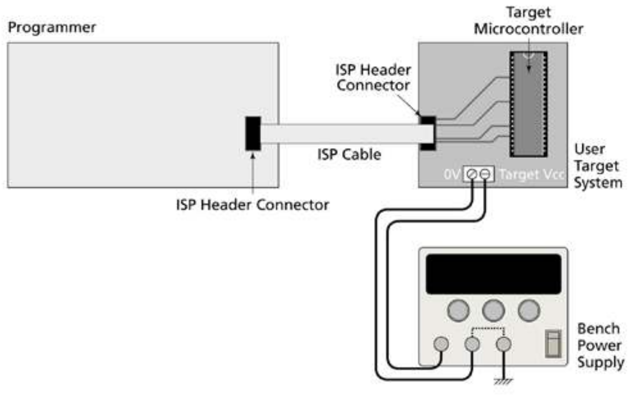
- Plug power supply jack into mating jack socket **CON 1** on the rear of the programmer.
- External power supply must be 9.0 to 12.0V DC regulated (9.0V DC regulated power supply is supplied with the programmer).
- Fit the **Jumper J9** which connects the programmer **+5.0V Vcc** supply to the **Target Vcc** on the ISP Header connectors.
- Plug the mains adaptor into the mains supply but do not switch the power on.

2.4.3. Powering the Target System at +5.0V from the programmer (external bench power supply)



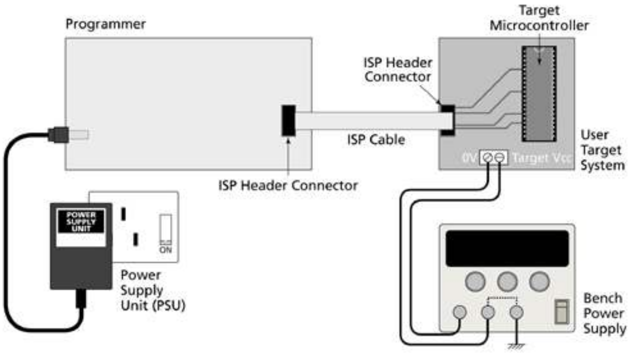
- The programmer is supplied with a DC power cable with a jack plug on one end and bare wires at the other end.
- Plug the jack plug into the mating jack socket CON 1 on the programmer.
- Connect the bare wire ends to a suitable external power supply e.g. bench power supply.
- The wire with the **WHITE STRIPE** on it is **+VE** and the all black wire is **–VE (GROUND).**
- The external power supply must be 9.0 to 12.0V DC regulated
- Fit the **Jumper J9** which connects the programmer **+5.0V Vcc** supply to the **Target Vcc** on the ISP Header connectors.
- If you have a 'Current Limit' setting on your Power Supply, set it to the minimum value to start with and then increase it until the programmer powers up without tripping the Current Limit.

2.4.4 Powering the programmer from the Target System at 3.0 – 5.0V



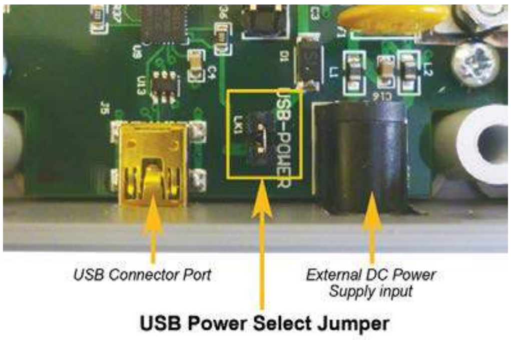
- The Target System can supply power to the programmer as long as the Target Voltage is 3.0 – 5.0V.
- The Target System must be able to supply the worst case programming current to the programmer (approx 60mA).
- Fit the **Jumper J9** which connects the **Target Vcc** (3.0 – 5.0V) on the ISP Header connectors to the Programmer Vcc rail.
- The programmer will automatically boost the **Target Vcc** supply up to +5.0V if it is less than 5.0V.

2.4.5 Programmer and Target System are independently powered



- The programmer is powered from +9.0 to 12.0V via the DC power input (CON1) on the rear panel of the programmer.
- The Target System is powered from an independent power supply.
- The **Jumper J9 must NOT be fitted**. This disconnects the Target Vcc from the Programmer +5V Vcc.
- The **'Target Vcc'** signal must still be connected to the **'Programmer Vcc'** signal on the ISP Header as this voltage is used to power the **'I/O Level Convertor'.**

2.4.6 Programmer powered from PC USB port



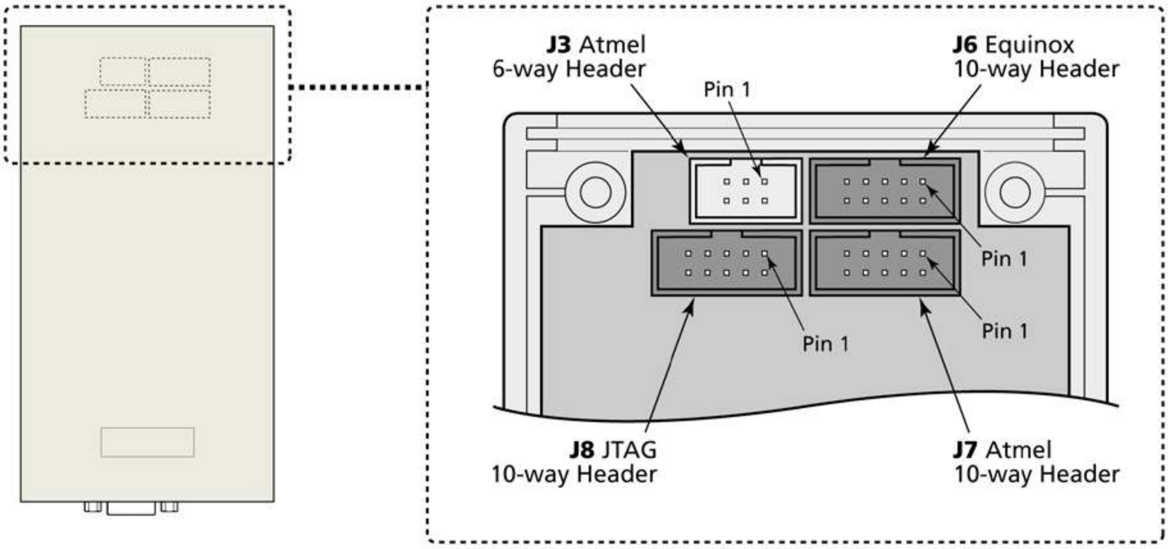
- The programmer is powered from the PC USB port.
- The jumper link ***'USB-POWER' (LK1)*** should be fitted.
- The Target System must be powered from an independent power supply.
- The **Jumper J9 must NOT be fitted**. This disconnects the Target Vcc from the Programmer +5V Vcc.
- The **'Target Vcc'** signal must still be connected to the **'Programmer Vcc'** signal on the ISP Header as this voltage is used to power the **'I/O Level Convertor'.**

2.5 Selecting the correct ISP (In System Programming) Header

a

Select the required ISP Header and then plug the ISP Cable into this Header.

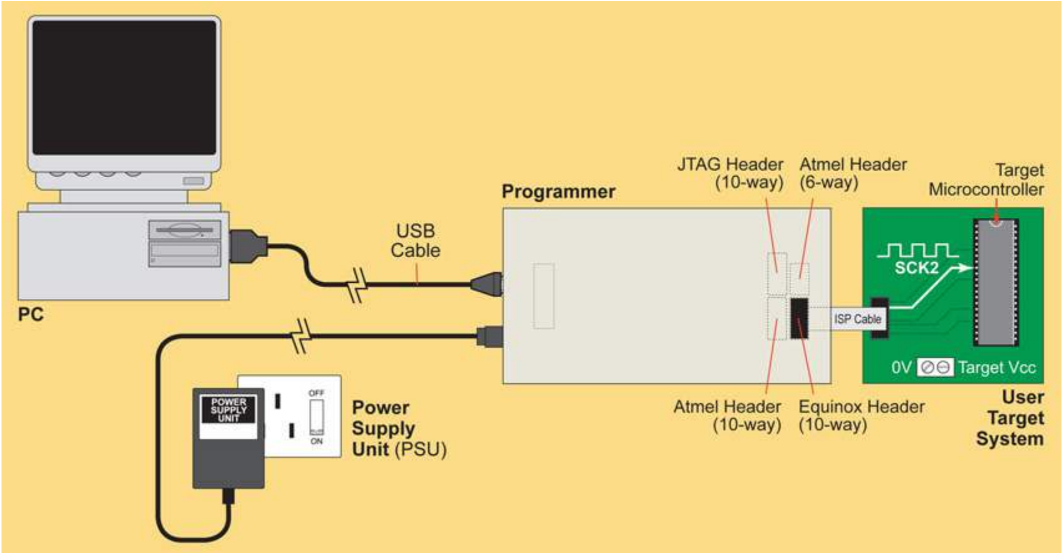
Please refer to section 4 of this manual for full details of all the available headers.



b

Connect the ISP cable to the ISP Header on your Target System

Make sure the programmer and Target System Power are switched OFF !
Make sure that the ISP Cable plug is inserted the correct way around into the ISP Header on the Target System.



Important note:
Please refer to the relevant Application Note for details of how to connect the programmer to the Target Device on the Target System. The Application Notes can be found on the FS2009USB - Download Page on the Equinox website.

2.6 Earthing requirements

2.6.1 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.

2.6.2 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 USB** 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 **PC USB 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)

2.6.3 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!

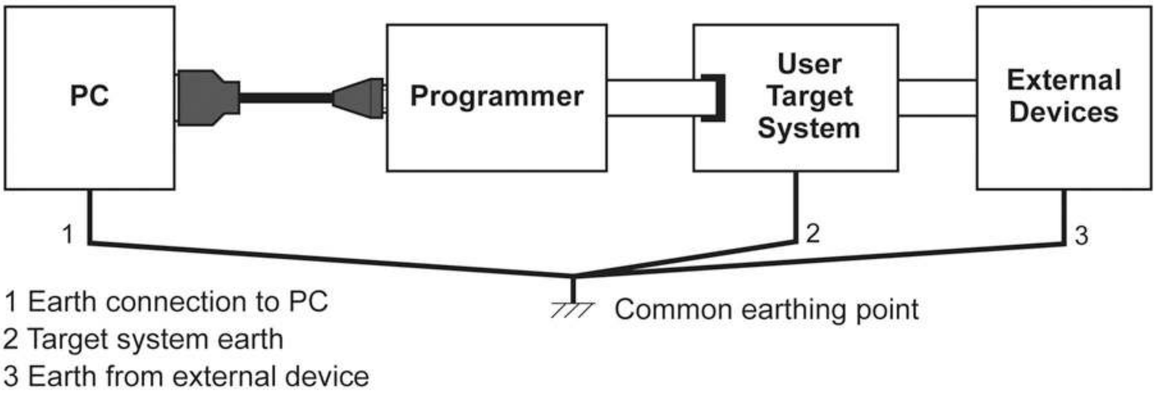
The solution:

- 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.

2.6.4 Earthing recommendations

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

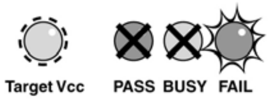
- Ensure that both your target system and PC are connected to a common earth point
- Make sure that all interconnections are made before applying power to PC and target system



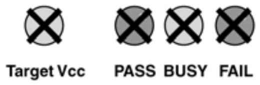
- If you are using a laptop or PC which is not connected to mains earth, it is recommended that you make a hard-wired connection from the USB connector shell and the target system to a common earth point
- Avoid plugging and unplugging ISP connector while either the PC or Target System is powered up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

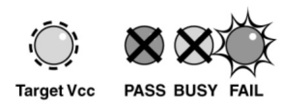
Warning!
Any damage caused to the programmer by Electrostatic Discharge (ESD) through inadequate earthing is not covered under the warranty of the product.

2.8 Testing the programmer hardware

a	**Apply power to the programmer** If you have a 'Current Limit' on your power supply, turn it to approximately 20mA and then gradually increase it until the programmer 'FAIL' LED starts to FLASH.	→ Programmer circuitry powers up to 5.0V. → Target Vcc LED illuminates The LED will illuminate if a target voltage between 3.0 and 5.0V is detected.  → FAIL LED flashes to indicate that the programmer has rebooted. *** The FAIL LED no longer flashes with firmware version 2.46 or greater) → The programmer display now initialises to show the programmer type, firmware version etc. See 2.7(b), (c), (d).
b	**No Programming Projects Loaded** NO / INVALID PROJECT LOADED!	• If the programmer has never been used before or the on-board FLASH Memory Store has been erased, then the programmer will display the message: **'No / Invalid Project Loaded'**. This is normal as the programmer 'FLASH Memory Store' is empty so the programmer cannot be used in Standalone Mode. • It is not possible to check the programmer information until a valid project is uploaded. Please upload a valid project to the programmer using EQTools and then try again.
c	**Check the programmer firmware version** EQUINOX-TECH UK FS2009 Ver: 5.36	• The programmer display now shows the programmer type and the firmware version. e.g. 4.00. • You can check that you have the very latest firmware version by viewing the 'Configit – Firmware Update Utility' on the Equinox website.
d	**Check the programmer serial number, build revision and build date** SN: 1234 1.00 DATE 03-03-2012	After 2 seconds the display will update to show the following information: • Serial Number (SN): eg. 1234 • Hardware Build Version: eg. 1.00 • Build date: eg. 03 /03 / 2012

2.9 Re-assembling the programmer

a	Re-fit lid to programmer	The ISP Cable should fit through the slot in the end panel.
b	Re-fit the four screws in the back of the case	
c	Ensure that the PC, programmer and Target System power is switched OFF.	



Please note:

- The FAIL LED will flash when you initially power up the programmer if you are running firmware 2.46 or below.
- If you are running firmware version 2.46 or greater the FAIL LED will no longer FLASH.
- To clear the FAIL LED condition, use the **<Up>** / **<Down>** keys to select a Programming Project and then press the **<Yes>** key .
- If there are no Programming Projects Loaded into the Programmer – Flash Memory Store, the FAIL LED will continue to FLASH until at least one Project is uploaded.

2.10 USB Driver - Installation Instructions

2.10.1 Driver Installation Instructions

1. Make sure the USB cable is **NOT** connected between the PC and programmer

2. Install the ***'USB Driver'*** – see Application Note AN126 for further instructions.
The ***'USB Driver'*** can be found as follows….

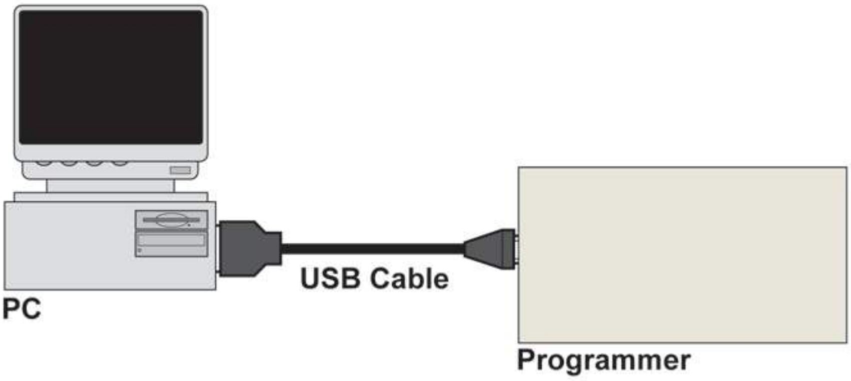
i. On the CD-ROM which came with the programmer
The driver should be in the following sub-folder on the CD:
\USBDrivers\PPM4-MK1_ISPnano_SeriesII

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

To download the USB Driver, please browse to the following web page:
http://www.equinox-tech.com/products/downloads.asp?details=286

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

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



→ The PC should display a message "***Found new hardware…..***"

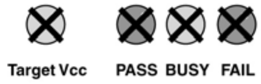
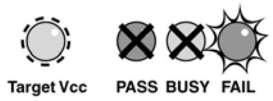
- Follow the on-screen instructions to install the USB Driver.
- Please refer to Application Note AN126 for further instructions.

2.10.2 Selecting the USB – Virtual COM Port (VCP) in EQTools

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

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

2.11 Hardware - Fault Finding Guide

i.	**Target Vcc LED does not illuminate when power is applied to the Programmer / Target System**  	• Switch power off immediately in case of a short-circuit ! • Check that you have chosen the correct ISP Header to connect to your Target System. • Check that the ISP Cable is inserted the correct away around into the ISP Header Socket on the Target System. • If the Programmer is powering the Target System, check that the 'Vcc Jumper (J9) is fitted. • If Programmer and Target System are independently powered, check that the 'Vcc Jumper (J9) is removed. Also check that the Vcc line from the Target System is definitely still connected to the Programmer Vcc pin on one of the ISP Headers.
ii.	**No Programming Projects Loaded** NO / INVALID PROJECT LOADED!	• If the programmer has never been used before or the on-board FLASH Memory Store has been erased, then the programmer will display the message: **'No / Invalid Project Loaded'**. This is normal as the programmer 'FLASH Memory Store' is empty so the programmer cannot be used in Standalone Mode. • Please upload a valid project to the programmer using EQTools and then try again.
iii.	**FAIL LED constantly flashing**  	• If you are running firmware 2.46 or below then the FAIL LED will flash when you initially power up the programmer. • To clear the FAIL LED condition, use the **<Up>** / **<Down>** keys to select a Programming Project and then press the **<Yes>** key . • If there are no ***'Standalone Programming Projects'*** loaded into the Programmer – Flash Memory Store, the FAIL LED will continue to FLASH until at least one Project is uploaded.

3.0 Programmer – Getting Started Guide

3.1 Software Overview

The FS2009 / FS2009USB programmer is supplied with a comprehensive Integrated Development Environment (IDE) called the **'EQTools Software Suite'**. This software supports creation and testing of so-called ***'Standalone Programming Projects'*** which can then be uploaded to the programmer for use in ***'Standalone Mode'***. EQTools contains many separate utilities most of which are enabled as standard for this programmer. Certain options such as compiling and running of ***'Programming Scripts'*** must be purchased from Equinox as a ***'License upgrade'***.

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

Fig. 3.1 Software / Control Options for the FS2009USB Programmer

EQTools Software	**Enabled as standard**
• EQTools - Project Builder	YES
• EQTools - Project Management Utility	YES
• EQTools - Project Upload/Download Utility (Upload Wizard)	YES
• EQTools - Development Suite (EDS)	YES
• EQTools - Script Builder	Upgrade
• EQTools - Script Editor	Upgrade
• EQTools – Incremental Repository (Serial number / MAC address generator)	Upgrade
• EQTools - Development Mode (PC Controlled)	YES
Programmer Control mechanisms	
• Programmer keypad + Display (Standalone operation)	YES

Key
YES – feature available as standard
Upgrade – a license upgrade must be purchased from Equinox to enable this feature.

3.2 Programmer Control Mechanisms

The programmer can be controlled from EQTools, using the programmer keypad / display, 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, FS2009USB, PPM3-MK2, PPM4-MK1

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

3.3 Software Installation

The FS2009USB 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/products/details.asp?ID=1561&displ=tl

EQTools versions are quoted as e.g. ***EQTools V4.1.0 Build 1024***
The filename for this version would be '***EQTools_V4_build1024.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 3 of the guide for detailed EQTools installation instructions.

3.4 Programmer Operating Modes

The FS2009USB programmer can be controlled in various different ways as detailed in the table below:

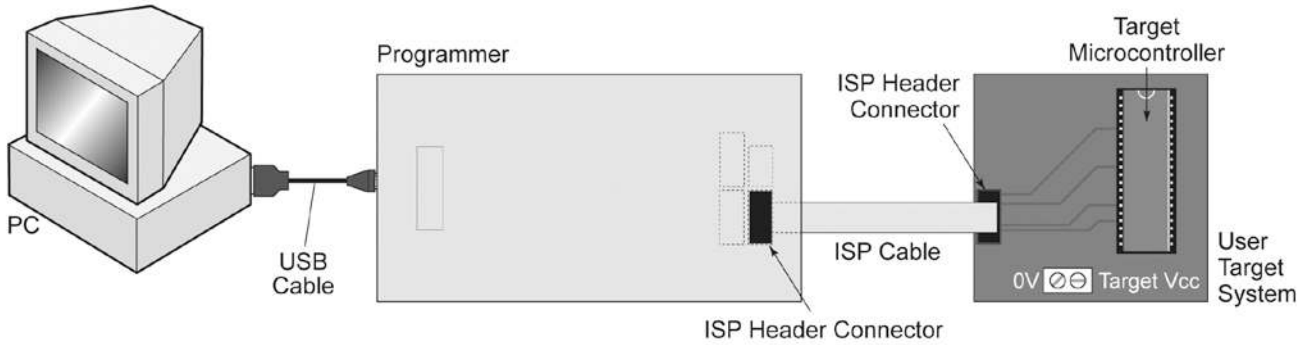
#	Programmer Mode	Connect to PC USB Port	Control Method / Software	Upgrade License Required	Functional description	Refer to section
1	**Development Mode**	Yes	EQTools - EDS	Standard Feature	In this mode, the programmer is controlled by EQTools - Equinox Development Suite (EDS). EDS allows you to read/write the target device FLASH / EEPROM / FUSE / LOCK bits interactively under PC control without uploading a 'Programming Project' to the programmer.	3.5
2	**Standalone Mode**	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.)	3.6
3	**ASCII Control Mode**	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

Key
S – feature available as standard
U – a license upgrade must be purchased from Equinox to enable this feature.

3.5 Development Mode (EDS)

3.5.1 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

3.5.2 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

3.5.3 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

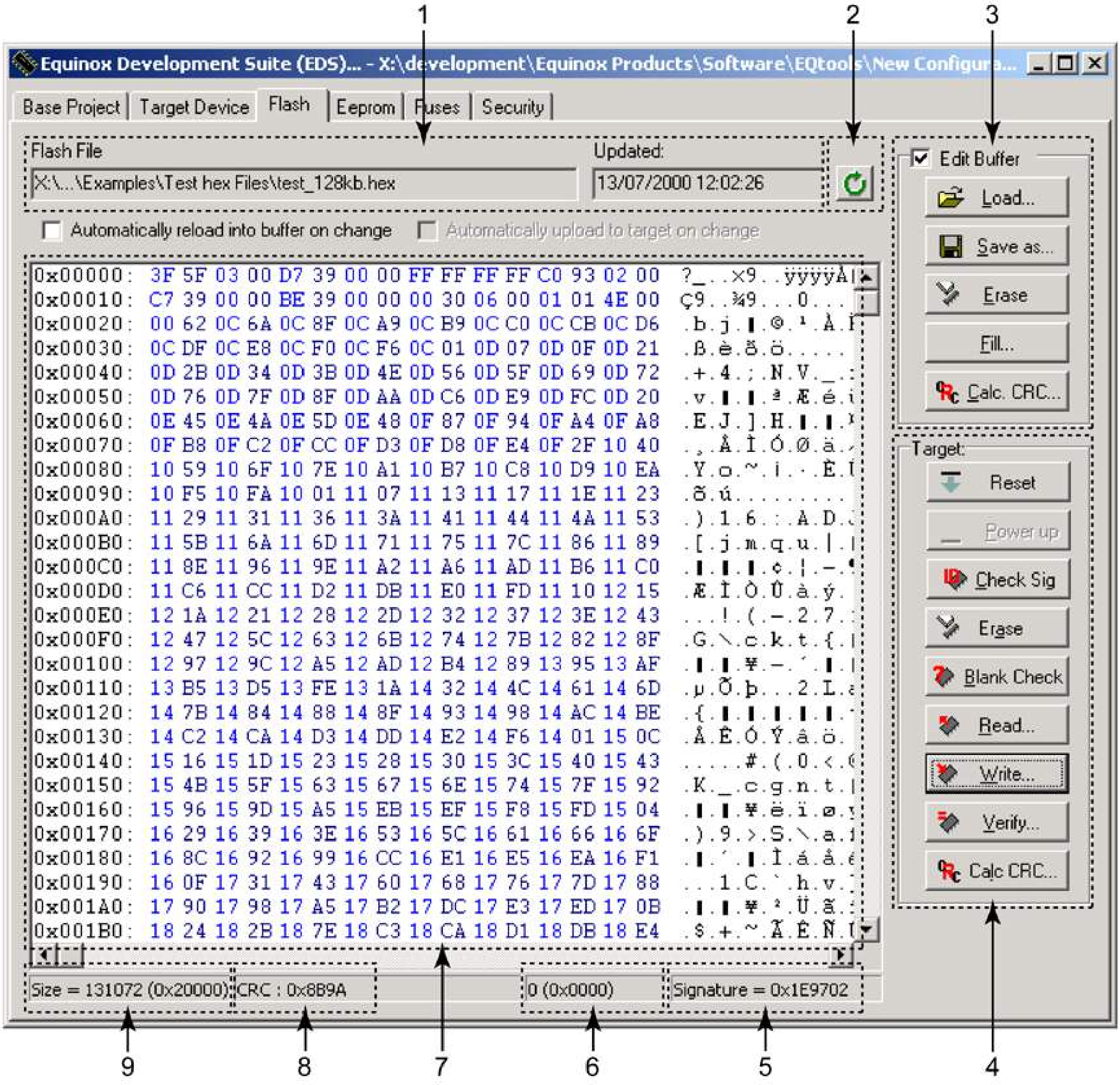
→ Project Collection File launches in **Project Manager** view

- 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 3.5.4

3.5.4 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. 3.5.4.1 EDS – FLASH Tab Overview



Flash File: X:\...\Examples\Test hex Files\test_128kb.hex

Updated: 13/07/2000 12:02:26

☑ Edit Buffer

☐ Automatically reload into buffer on change ☐ Automatically upload to target on change

```
0x00000:  3F 5F 03 00 D7 39 00 00 FF FF FF FF C0 93 02 00  ?_..×9..ÿÿÿÿÀ
0x00010:  C7 39 00 00 BE 39 00 00 00 30 06 00 01 01 4E 00  Ç9..¾9...0....
0x00020:  00 62 0C 6A 0C 8F 0C A9 0C B9 0C C0 0C CB 0C D6  .b.j.▌.©.¹.À.
0x00030:  0C DF 0C E8 0C F0 0C F6 0C 01 0D 07 0D 0F 0D 21  .ß.è.ð.ö.....
0x00040:  0D 2B 0D 34 0D 3B 0D 4E 0D 56 0D 5F 0D 69 0D 72  .+.4.;.N.V._.
0x00050:  0D 76 0D 7F 0D 8F 0D AA 0D C6 0D E9 0D FC 0D 20  .v.▌.▌.ª.Æ.é.
0x00060:  0E 45 0E 4A 0E 5D 0E 48 0F 87 0F 94 0F A4 0F A8  .E.J.].H.▌.▌.
0x00070:  0F B8 0F C2 0F CC 0F D3 0F D8 0F E4 0F 2F 10 40  .¸.Â.Ì.Ó.Ø.ä.
0x00080:  10 59 10 6F 10 7E 10 A1 10 B7 10 C8 10 D9 10 EA  .Y.o.~.¡.·.È.
0x00090:  10 F5 10 FA 10 01 11 07 11 13 11 17 11 1E 11 23  .õ.ú.........
0x000A0:  11 29 11 31 11 36 11 3A 11 41 11 44 11 4A 11 53  .).1.6.:.A.D.
0x000B0:  11 5B 11 6A 11 6D 11 71 11 75 11 7C 11 86 11 89  .[.j.m.q.u.|.
0x000C0:  11 8E 11 96 11 9E 11 A2 11 A6 11 AD 11 B6 11 C0  .▌.▌.▌.¢.¦.-.
0x000D0:  11 C6 11 CC 11 D2 11 DB 11 E0 11 FD 11 10 12 15  .Æ.Ì.Ò.Û.à.ý.
0x000E0:  12 1A 12 21 12 28 12 2D 12 32 12 37 12 3E 12 43  ...!.(.-.2.7.
0x000F0:  12 47 12 5C 12 63 12 6B 12 74 12 7B 12 82 12 8F  .G.\.c.k.t.{.
0x00100:  12 97 12 9C 12 A5 12 AD 12 B4 12 89 13 95 13 AF  .▌.▌.¥.-.´.▌.
0x00110:  13 B5 13 D5 13 FE 13 1A 14 32 14 4C 14 61 14 6D  .µ.Õ.þ...2.L.
0x00120:  14 7B 14 84 14 88 14 8F 14 93 14 98 14 AC 14 BE  .{.▌.▌.▌.▌.▌.
0x00130:  14 C2 14 CA 14 D3 14 DD 14 E2 14 F6 14 01 15 0C  .Â.Ê.Ó.Ý.â.ö.
0x00140:  15 16 15 1D 15 23 15 28 15 30 15 3C 15 40 15 43  .....#.(.0.<.
0x00150:  15 4B 15 5F 15 63 15 67 15 6E 15 74 15 7F 15 92  .K._.c.g.n.t.
0x00160:  15 96 15 9D 15 A5 15 EB 15 EF 15 F8 15 FD 15 04  .▌.▌.¥.ë.ï.ø.
0x00170:  16 29 16 39 16 3E 16 53 16 5C 16 61 16 66 16 6F  .).9.>.S.\.a.
0x00180:  16 8C 16 92 16 99 16 CC 16 E1 16 E5 16 EA 16 F1  .▌.´.▌.Ì.á.å.
0x00190:  16 0F 17 31 17 43 17 60 17 68 17 76 17 7D 17 88  ...1.C.`.h.v.
0x001A0:  17 90 17 98 17 A5 17 B2 17 DC 17 E3 17 ED 17 0B  .▌.▌.¥.².Ü.ã.
0x001B0:  18 24 18 2B 18 7E 18 C3 18 CA 18 D1 18 DB 18 E4  .$.+.~.Ã.Ê.Ñ.
```

Size = 131072 (0x20000) | CRC : 0x8B9A | 0 (0x0000) | Signature = 0x1E9702

Fig.3.5.4.2 FLASH / EEPROM tab – functional description

#		
1	**File name and properties**	This specifies the ‘File Name / path’ and ‘Last updated’ date of the file to be loaded into the Buffer Window. • By default, this will point to the file specified in the associated Programming Project (*.ppm). • To load a different file, click the <Load> button and browse to the required file.
2	**Re-load file to buffer**	Clicking the ⟳ icon re-loads the specified Flash / EEPROM file into the Buffer Window. This will overwrite any information already in the Buffer Window.
3	**Buffer Control Buttons**	This group of buttons control operations on the EDS – Buffer Window. • To use these buttons or to manually edit the Buffer Window, it is necessary to check the ‘Edit Buffer’ check box. • This extra step helps to avoid accidental modification of data in the buffer.
4	**Target Control Buttons**	This group of buttons control operations on the actual Target Device. • A programmer and suitable Target System must be connected when using these buttons.
5	**Signature of Target Device**	This is the signature (Device ID) which is expected for the Target Device.
6	**Buffer Address**	This is the address of the currently selected location in the Buffer Window.
7	**Buffer Window**	The Buffer Window displays a hexidecimal and alphanumeric representation of the data which has either been loaded from file or read back from a Target Device. • The bytes are grouped into rows of 16 bytes with the start address of each row displayed in the left-hand column. • The Hexadecimal representation of the 16 bytes is displayed in the middle column • The ASCII representation of the 16 bytes is displayed in the right-hand column.
8	**CRC**	This is a CRC Checksum of the entire Buffer (i.e. from address 0x00000 to the end address specified in the buffer.)
9	**Size**	This is the physical address range in bytes of the entire Buffer.

3.5.5 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	↻ **Re-load File**	• Re-loads the specified HEX / Binary file into the Buffer Window.
2	**Edit Buffer**	• Tick the box to enable manual editing of the Buffer Window. • This will also allow the <Load>, <Erase> and <Fill> buttons to be selected.
3	**<Load>**	• This button allows you to manually browse to select a file to be load into the Buffer Window. • The path will default to the path set up in the Programming Project (if a file is specified).
4	**<Save as>**	• This button allows you to save the contents of the Buffer Window to a file. It is possible to specify the following properties for the file: • File type (HEX / Binary) • File Start and End Address
5	**<Erase>**	• This button erases the entire contents of the Buffer Window. • All locations are set to the value 0xFF.
6	**<Fill>**	• This button allows you to fill a specified area of the Buffer Window with a specified value.
7	**Calc CRC**	• This button calculates the CRC Checksum of the specified range of the Buffer Window.

3.5.6 Example of using EDS

Here is an example of how to use EDS:

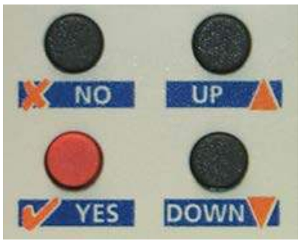
- Select the ***<FLASH>*** tab
- Click the ***<Load>*** button to load a file
- Click the ***<Check Sig>*** button to check the Device Signature of the Target Chip
- Click the ***<Write>*** button to write the data in the Buffer Window to the Target Device
- Click the ***<EEPROM>*** tab to select operations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

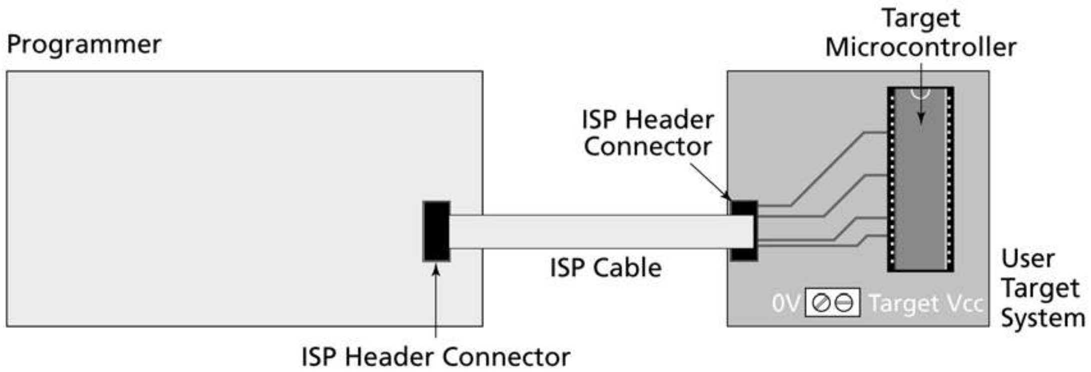
3.6 Standalone Keypad Mode

3.6.1 Overview

In ***'Standalone Mode'***, the FS2009USB is controlled via the push buttons on the front panel of the programmer. The programmer LCD and the LED Status Indicators are used to display the current status of the programmer. In ***'Standalone Mode'*** it is possible to select from 1 of 64 previously uploaded ***'Standalone Programming Projects'*** and then to repeatedly execute this project to program a batch of devices.

LCD Display	Buttons	Status LED's
PRESS YES TO PROGRAM TARGET	 	 PASS BUSY FAIL

This mode 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.



Before using Standalone Mode you need to:

- Connect Programmer to PC USB port
- Upload a valid Project Collection to containing 1 – 64 Programming Projects to the programmer. This will be retained in the Programmer 'Flash Memory Store' even once the power is removed.
- It is a good idea to test these projects using EDS (Equinox Development Suite) before uploading them to the programmer.
- Disconnect the programmer from the PC

Please note:
It is recommended that all Programming Projects are fully tested and debugged using the EDS (Equinox Development Suite) utility **BEFORE** uploading them to the programmer. This can save a lot of time as problems are much easier to debug within the EDS environment.

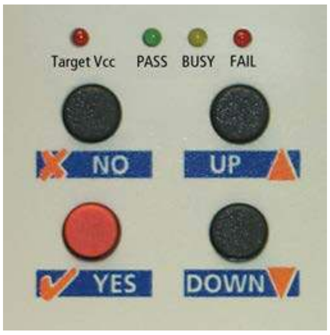
In Standalone Mode you can:

- Select from 1 of up to 64 independent ***'Standalone Programming Projects'*** using the keypad and display
- Plug the Programmer ISP cable into a Target System
- Program the selected ***'Standalone Programming Project'*** into the Target Device
- Unplug the Programmer ISP Cable from the Target System
- Plug the Programmer ISP Cable into the next Target System
- Repeatedly program the same ***'Standalone Programming Project'*** once it is selected

To execute a Programming Project in Standalone Mode:

#	Action	Observation / comment
1	Connect the programmer to a spare PC USB port	
2	Apply power to the Programmer / Target System	→ Programmer Target Vcc LED illuminates
3	Upload your ***'Standalone Programming Project(s)'*** to the programmer	See EQTools Manual – Project Upload Wizard for full instructions.
4	Disconnect power from the Programmer / Target System	
5	Disconnect the serial lead from the programmer	→ Programmer is no longer controlled from PC
6	Follow the instructions to program the Target Device in Standalone Mode	

3.6.2 Push Button Functions

 Target Vcc PASS BUSY FAIL NO UP YES DOWN	The FS2009USB programmer features a four button keypad which is used to control the programmer when it is operating in ‘Standalone Mode’. The function of these buttons is described below.
Button	**Function of button**
	The **<YES>** button has the following functions: i. To execute a programming project which has been pre-loaded into the programmer ‘FLASH Memory Store’. ii. To return to the ‘Standby’ state PASS BUSY FAIL ⊗ ⊗ ⊗
	The **<No>** button has the following functions: i. To cancel a programming action ii. To return to the ‘Standby’ state PASS BUSY FAIL ⊗ ⊗ ⊗
	• The **<Up>** and **<Down>** buttons are used to scroll up and down through the list of available ‘Programming Projects’ stored in the programmer ‘FLASH Memory Store’. • If there are not ‘Programming Projects’ currently in memory, then pressing thee buttons will have no effect.

3.6.3 Standalone Keypad Operation – step-by-step guide

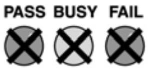
The instructions below detail how to operate the programmer in Standalone Mode using the programmer keypad and LCD. A PC is not required while operating in Standalone Mode.

1. Ensure that you have already uploaded a valid 'Project Collection' to the programmer

- Use the EQTools – Upload Wizard to upload your 'Project Collection'.
- Use the new Standalone Upload Wizard Utility to upload your 'Project Collection' (software upgrade required)

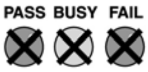
2. Select the 'Programming Project' which you wish to execute:

- Use the <UP> and <Down> keys to move up and down the list of projects currently resident in the programmer 'FLASH Memory Store'.
- If there is one ***'Standalone Programming Project'*** resident in the programmer, then this project will be displayed permanently on the LCD.
- If Project Version Control has been enabled for the project(s), then the project version information will be displayed on the bottom line of the LCD.

LCD Display	Buttons	Status LED Display
Select Project Project1 Select Project Project2 Select Project Project3		

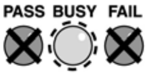
Important note:
If the ***'Lock keypad'*** option has been enabled, then the project can only be selected once after which time the same project will be permanently selected.

3. Press the RED <YES> button to select the Programming Project

LCD Display	Buttons	Status LED Display
Select Project Project1		

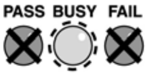
4. Wait for the programmer to CRC check the Programming Project

- The programmer performs a full CRC validation check of the entire Programming Project stored in the programmer 'FLASH Memory Store'.
- If the check is OK, the programmer goes to step (5).
- If the check FAILS, the programmer will display ***'Internal Image is corrupted'***. Please try re-uploading your Project Collection and then repeat the Standalone Mode process from the beginning.

LCD Display	Buttons	Status LED Display
CHECKING IMAGE PAGE n		

5. Press the RED <YES> button again to execute the Programming Project

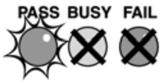
- The Programming Project will now start to execute.
- The <BUSY> LED will illuminate to denote that the programmer is active.
- This may take a few seconds or a few minutes depending on the nature of the Programming Project.
- The Programming Project will either PASS or FAIL

LCD Display	Buttons	Status LED Display
Current programming operation is displayed	To abort the programming sequence at any time, press 	

6. Programming Project – PASS

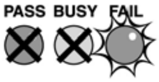
If the Programming Project programs the target device without any errors:

- The <PASS> LED will FLASH
- The LCD will display: 'Operation Complete OK!'
- Press <YES> key to program another device

LCD Display	Buttons	Status LED Display
OPERATION COMPLETE OK	Press To program another device.	

7. Programming Project – FAIL
If the Programming Project fails for some reason:

- The <FAIL> LED will FLASH
- The programmer will automatically remove power from the Target System (if the programmer is controlling power)
- The LCD will display the relevant Error Message.
- The List of possible Error Message can be found in the 'Error Messages' section of this manual.
- Press the <NO> button to clear the Error Message
- Press <YES> key to program another device. The programmer then goes back to step 3.

LCD Display	Buttons	Status LED Display
ERROR MESSAGE	Press to clear the error message.	

3.6.4. Locking the Programmer Keypad

It is possible to lock the programmer keypad so that the ***<Up>*** and ***<Down>*** keys no longer work in Standalone Mode. This will stop a user being able to select any other project stored in the programmer once the initial project has been selected.

To lock the Programmer Keypad:

- Open your Programming Project in Project Builder or EDS
- Select the ***<Programmer and Project Type>*** tab
- Check the ***'Lock Keypad'*** box
- Re-compile the project
- Upload the project to the programmer

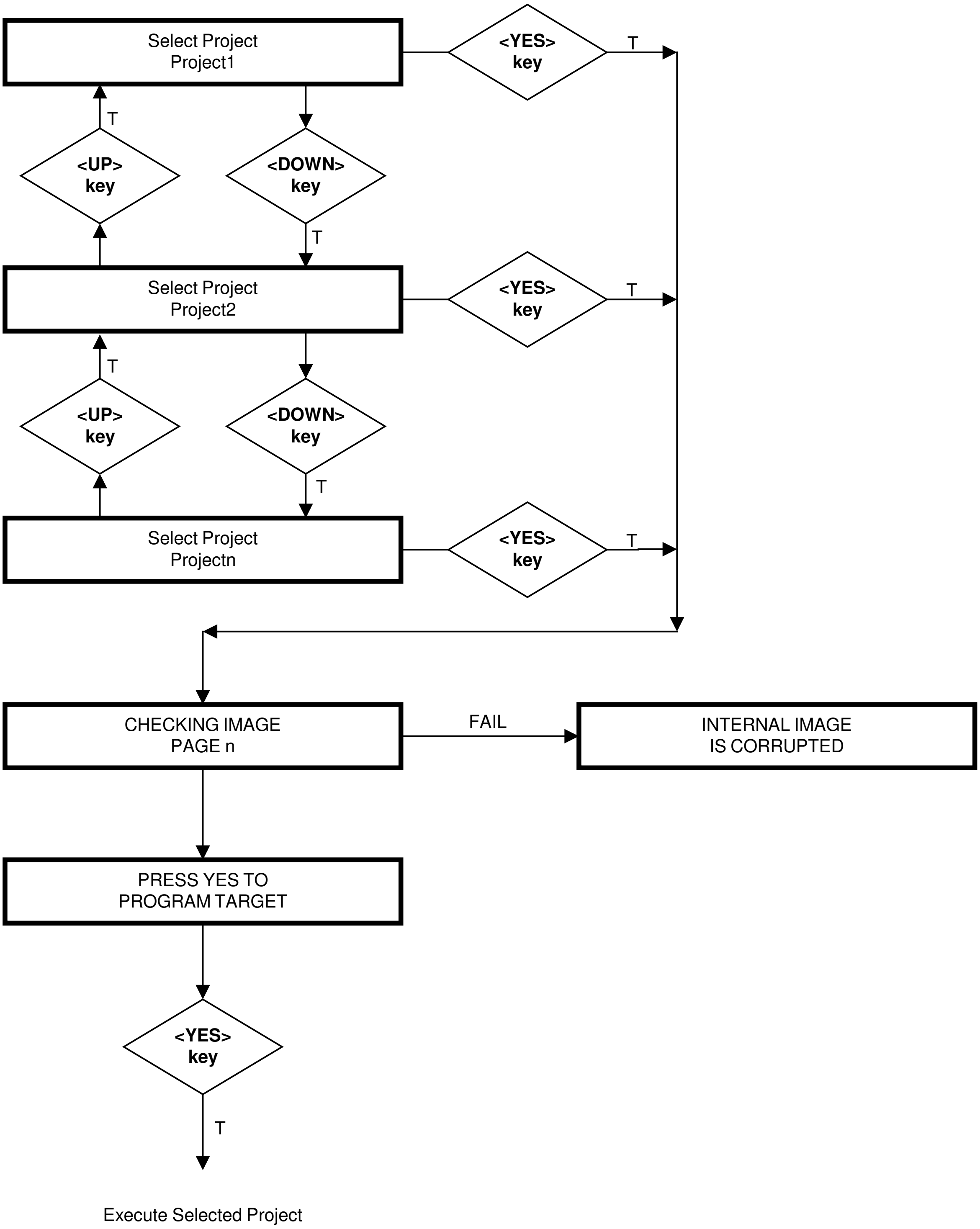
This is how the programmer works when the **'Keypad lock'** function is activated:

- User powers up the programmer
- User can then use the ***<Up>*** and ***<Down>*** keys to select the required project
- User selects the required projects **ONCE** and executes this project
- When this project has finished, the programmer now repeats the same project.
- The only way to be able to re-select the project is to power the programmer off and then on again.

Please note:
The **'Keypad lock'** function feature is only available in firmware 3.01 or above.

3.6.5 Standalone Keypad Operation – Flowchart

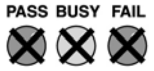
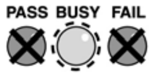
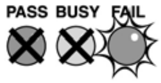
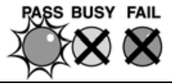
The flowchart below details how a ‘Programming Project’ is selected in ‘Standalone Mode’.



3.6.6 Programmer Status LED's

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

Fig. 2.6.5 Programmer Status LED's – state descriptions

Status LED Display	State	State Description
	**WAITING**	• Programmer is now waiting to start an 'Autoprogram' operation.
	**BUSY**	• Programmer is 'BUSY' performing a programming operation. • If the programmer is controlled from EDS, the BUSY LED will remain on after a programming operation until a 'RESET programmer' command is executed.
	**FAIL**	• Programming operation has FAILED. • See programmer LCD for diagnostics • Programmer will automatically switch off target power if it is controlling power.
	**PASS**	• Programming operation was successful.

Status LED key:

ON	FLASHING	OFF
●	✹	✕

3.7 ASCII Text Communications Mode

3.7.1 Overview

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

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

3.7.2 Limitations of this protocol

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

The limitations of this protocol are as follows:

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

3.7.3 Further Information

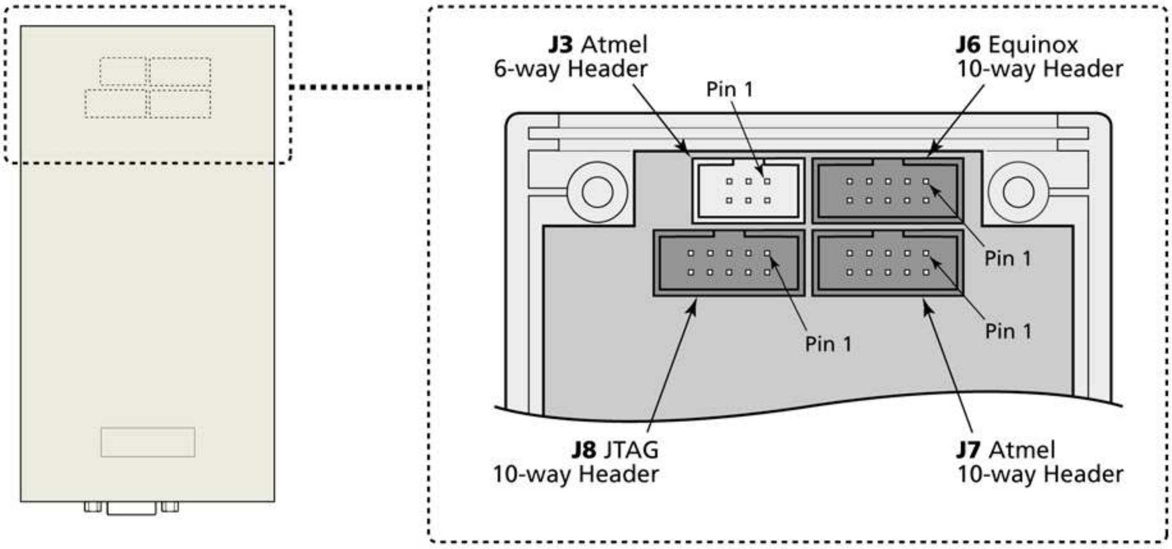
A full description of this protocol can be found in the Application Note ***'AN110 - ASCII Text Control (ATC) Protocol for Remote Control of Equinox Programmers'***.

4.0 ISP Header Selection

4.1 Overview

The FS2009USB 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.

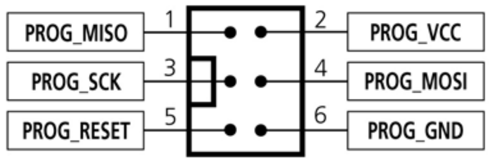
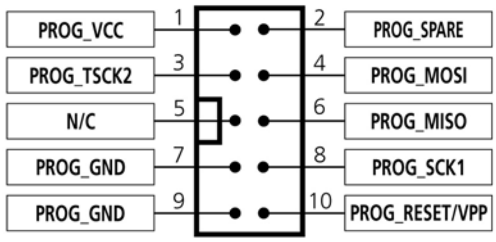
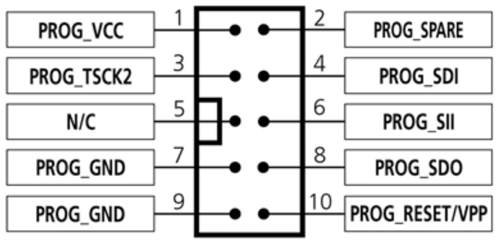
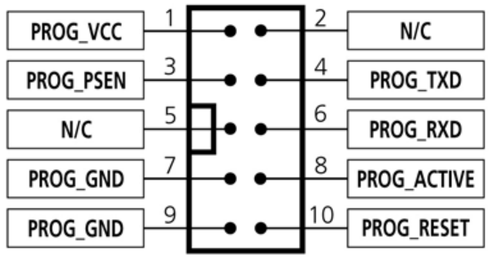
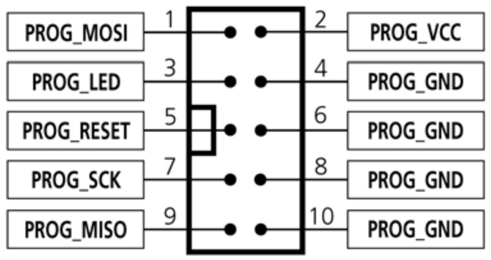
To locate the ISP Headers (J3, J6, J7 & J8), remove the four screws from the back of the programmer and then carefully remove the lid. If you hold the programmer with the RS-232 connector towards your, the four ISP headers will be in the positions shown in the figure below. Pin 1 is designated on each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

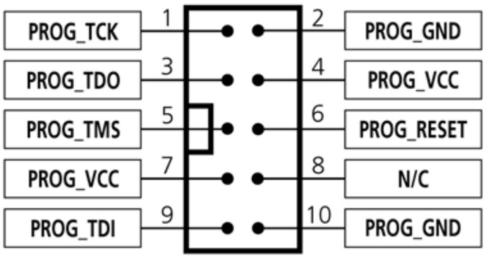


Please see section 4.2 for an overview of each ISP Header.

4.2 ISP Header Selection Chart (by header)

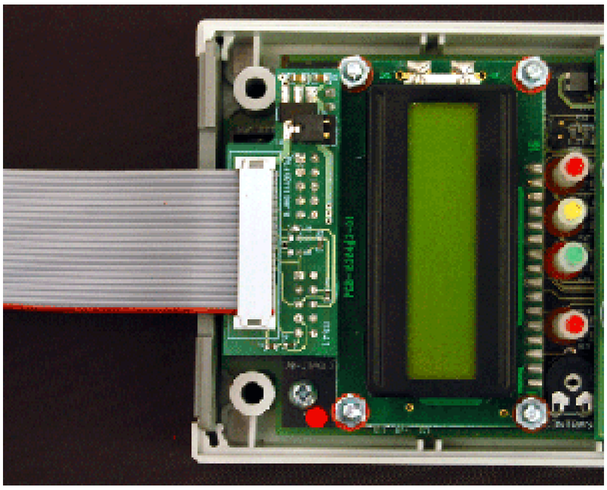
The FOUR ISP Headers featured on the FS2009USB are detailed in the table below.
Please refer to the section indicated in the 'refer to section' column for specific details of each header.

#	ISP Header	Description / Function	Refer to section	ISP Header Pin-out
1	J3	Atmel 6-way ISP Header	4.4	 PROG_MISO 1 2 PROG_VCC PROG_SCK 3 4 PROG_MOSI PROG_RESET 5 6 PROG_GND
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)	**Equinox 10-way Header(a)** **Device support:** Atmel AT90S, ATmega, AT90CAN, AT90USB, ATtiny, AT89S devices	4.5	 PROG_VCC 1 2 PROG_SPARE PROG_TSCK2 3 4 PROG_MOSI N/C 5 6 PROG_MISO PROG_GND 7 8 PROG_SCK1 PROG_GND 9 10 PROG_RESET/VPP
2b	J6(b)	**Equinox 10-way Header(a)** **Device support:** Atmel ATtiny11/12/15 High Voltage (+12V Vpp) Programming Mode	4.6	 PROG_VCC 1 2 PROG_SPARE PROG_TSCK2 3 4 PROG_SDI N/C 5 6 PROG_SII PROG_GND 7 8 PROG_SDO PROG_GND 9 10 PROG_RESET/VPP
2c	J6(c)	**Equinox 10-way Header(b)** **Device support:** Atmel Wireless T89C51Rx2 Philips P89C51Rx2 / 66x	4.7	 PROG_VCC 1 2 N/C PROG_PSEN 3 4 PROG_TXD N/C 5 6 PROG_RXD PROG_GND 7 8 PROG_ACTIVE PROG_GND 9 10 PROG_RESET
3	J7	**Atmel 10-way Header** **Device support:** Atmel AT90S, ATmega, ATtiny, AT89S devices	4.8	 PROG_MOSI 1 2 PROG_VCC PROG_LED 3 4 PROG_GND PROG_RESET 5 6 PROG_GND PROG_SCK 7 8 PROG_GND PROG_MISO 9 10 PROG_GND

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

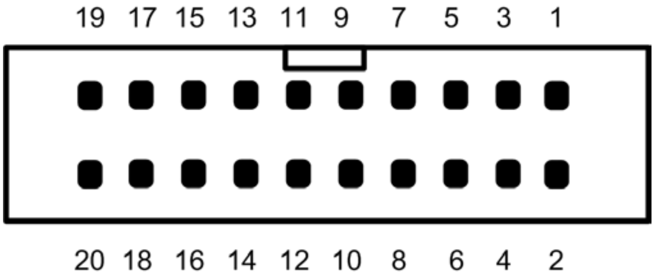
ARM ISP Cable

There is no direct connector for supporting ARM microcontrollers on the FS2003 / FS2009USB programmers. Instead a special ***'ARM ISP Cable'*** is available which plugs into the standard connectors.



Please refer to Appendix 1 for further information about the 'ARM ISP Cable'.

The pin-out of the 20-way connector cable is shown below:

	ARM 20-way cable **Device support:** Atmel AT91SAM7 NXP LPC2xxx		

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	NXP (formerly Philips) P89C51Rx2	P89CRB2/RC2/RD2	UART Boot Loader	J6(c)	4.7	AN106 AN107
9	NXP (formerly 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 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 ***Programmer ISP Header J3*** and plug the other end of the cable into the matching header on the Target System.

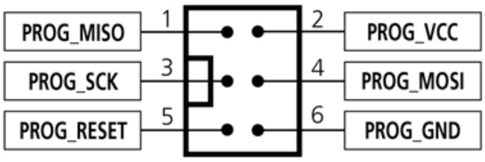


Figure 4.4.1 - Atmel 6-way IDC Header (J3) viewed from above

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)	**Master In Slave Out** 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	**Target Vcc** 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	**SPI Serial Clock Output** This is the SPI clock output signal.
4	PROG_MOSI	O	MOSI (except for ATmega103/128/64 – connect to RXD pin instead)	**Master Out Slave In** 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	**Target RESET control pin** 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	**Ground Connection** 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 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 ***Programmer J6 header*** and plug the other end of the cable into the matching header on the Target System.

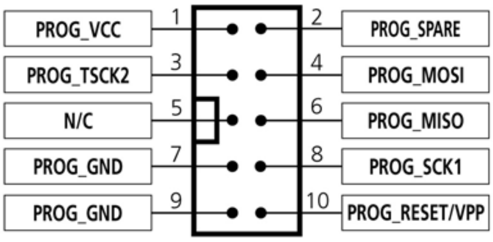


Figure 4.5.1 – Equinox 10-way IDC Header (J8) viewed from above

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	**Target VCC** 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	**Spare Programmer I/O pin** 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!	**SCK2 Clock Output (*Optional*)** 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)	**Master Out Slave In** 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)	**Master In Slave Out** 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	**GROUND**	**Ground Connection** Common ground connection between PROGRAMMER and Target System.
8	PROG_SCK1	O	**SCK**	**SPI Serial Clock Output** This is the SPI clock output signal used to clock data into / out of the Target Device.
9	PROG_GND	P	**GROUND**	**Ground Connection** Common ground connection between PROGRAMMER and Target System.
10	PROG_RESET	O	**RESET**	**Target RESET control pin** This pin controls the Target Device RESET pin. It will be driven HIGH/LOW according to the device type and settings in the <Pre-program State Machine> tab in the Eqtools project. **Warning!** 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 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 ***Programmer 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.

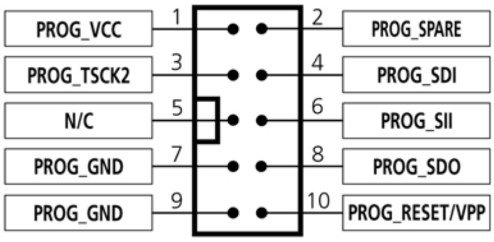


Figure 4.6.1 – Equinox 10-way IDC Header (J8) viewed from above

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	**Target VCC** 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	**Spare Programmer I/O pin** 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	**SCK2 Clock Output** This output pin is used to supply an external clock signal (SCK2) to the target microcontroller.

4	PROG_SDI	O	SDI	**Serial Data Input** 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	**SII (Serial Instruction Input)** 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	**Ground Connection** Common ground connection between PROGRAMMER and Target System.
8	PROG_SDO	I	SDO	**SDO (Serial Data Output)** 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	**Ground Connection** Common ground connection between PROGRAMMER and Target System.
10	PROG_RESET/VPP	O	RESET	**Target RESET control pin** This pin controls the Target Device RESET pin. A +12.0V Vpp voltage will be applied to this pin during programming. **Warning!** 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 FS2009USB 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 ***Programmer 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.

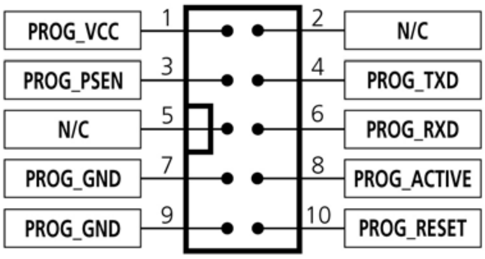


Figure 4.7.1 – Equinox 10-way IDC Header (J8) viewed from above

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	**Target Vcc** 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	**Not connected**
3	PROG_PSEN	O	PSEN	**PSEN pin** This pin is driven LOW by the programmer to enter 'Boot Loader' programming mode.
4	PROG_TXD	O	RXD	**Programmer RXD (Transmit) pin** The programmer transmits serial data on this pin to the Target Device RXD (Receive) pin.
5	N/C	-	N/C	**Not Connected**

6	PROG_RXD	I	TXD	**Programmer RXD (Transmit) pin** The programmer receives serial data on this pin from the Target Device TXD (Transmit) pin.
7	PROG_GND	P	Ground Connection	**Target / Programmer GROUND connection** Common ground connection between PROGRAMMER and Target System.
8	PROG_ACTIVE	O	LED or Interrupt pin	**Programmer ACTIVE LED** 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	**Target / Programmer GROUND connection** Common ground connection between PROGRAMMER and Target System.
10	PROG_RESET	O	RESET	**Target RESET control pin** 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. **Warning!** 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 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 ***Programmer ISP Header J7*** and plug the other end of the cable into the matching header on the Target System.

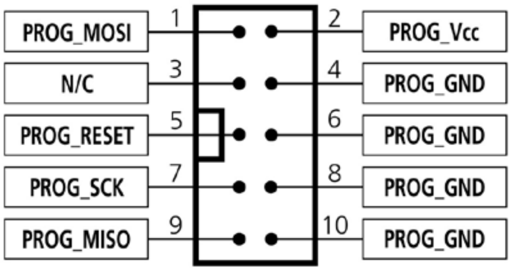


Figure 4.8.1 - Atmel 10-way IDC Header (J7) viewed from above

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)	**Master Out Slave In** 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	**Target Vcc** 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	**Ground Connection** Common ground connection between the programmer and Target System.

5	PROG_RESET	O	RESET	**Target RESET control pin** This pin controls the Target Device RESET pin. It will be driven HIGH/LOW according to the device type and settings in the 'Pre-program State Machine' tab in the Eqtools project.
6	PROG_GND	P	GROUND	**Ground Connection** Common ground connection between the programmer and Target System.
7	PROG_SCK1	O	SCK	**SPI Serial Clock Output** This is the SPI clock output signal.
8	PROG_GND	P	GROUND	**Ground Connection** Common ground connection between the programmer and Target System.
9	PROG_MISO	I	MISO (except for ATmega103/128/64 – connect to TXD pin instead)	**Master In Slave Out** 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	**Ground Connection** 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 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 ***Programmer ISP Header J8*** and plug the other end of the cable into the matching header on the Target System.

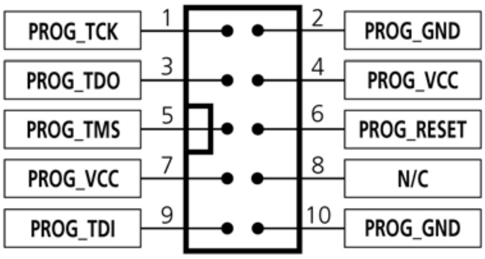


Figure 4.9.1 - Atmel 10-way IDC Header (J8) viewed from above

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

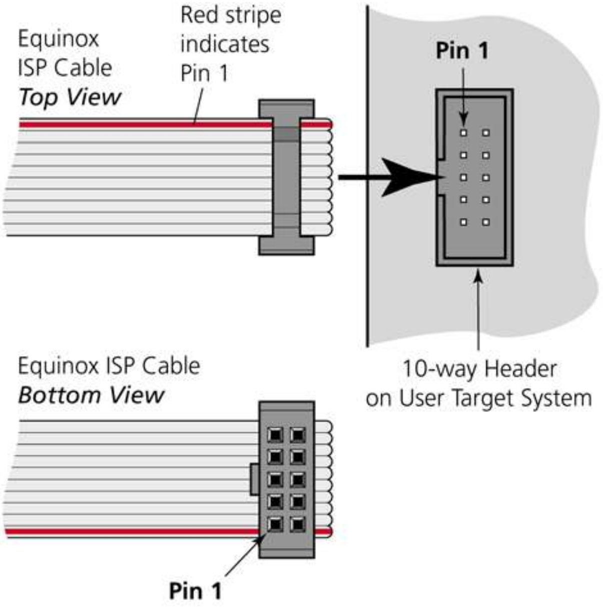
Pin No	Programmer Pin name	Programmer Input / Output	Connect to pin on Target Device	Description
1	PROG_TCK	O	TCK	**JTAG TCK – Test Clock Signal pin** Clock signal from programmer to Target Device JTAG port.
2	PROG_GND	P	GROUND	**Ground Connection** Common ground connection between Programmer and Target System.
3	PROG_TDO	I	TDO	**JTAG TDO – Test Data Output pin** Data signal from Target device JTAG port to programmer.
4	PROG_VCC	P	TARGET_VCC	**Target Vcc Connection** - 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	**JTAG TMS – Test Mode Select pin** Mode Select Signal from programmer to Target Device JTAG port.
6	PROG_RESET	O	RESET	**Microcontroller RESET control signal** 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	**Target Vcc Connection** - See pin 4 - Pins 4 + 7 are physically connected inside the programmer.
8	N/C	O	N/C	**Not Connected**
9	PROG_TDI	O	TDI	**JTAG TDI – Test Data Input pin** Data signal from programmer to Target Device JTAG port.
10	PROG_GND	P	GROUND	**Ground Connection** 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.



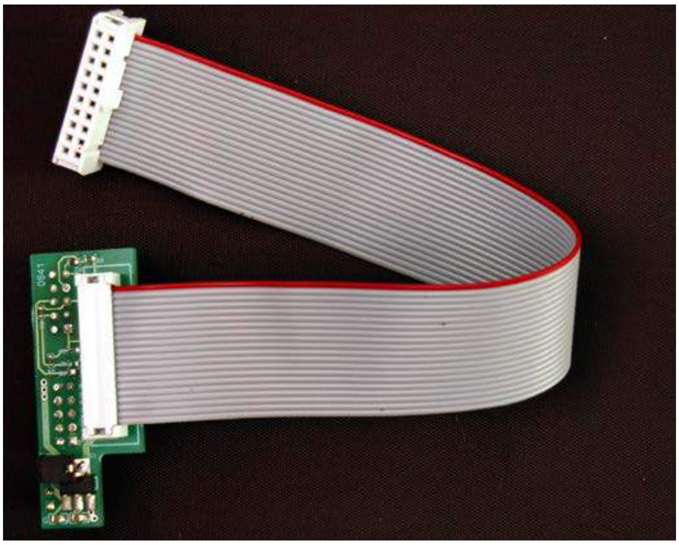
Pin 1 of the ISP cable can be determined by looking for a small arrow on the plastic part of the ISP female plug. If the cable has a RED stripe on one cable, this usually also indicates pin 1.

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.

Appendix 1 – ARM ISP Cable for Epsilon5, FS2003, FS2009 and FS2009USB programmers

1.0 Overview

It is necessary to use a special ***'ARM ISP Cable'*** to connect between an Epsilon5, FS2003, FS2009 or FS2009USB programmer and an ARM ***'Target System'***. This cable converts the programmer pin-out to the standard 20-way ARM JTAG pin-out suitable for plugging into a 20-way IDC socket on any ARM Target System. It also provides a safe way of powering the Target System from the programmer at 3.3V if required.



1.1 Features

- Compatible with Equinox Epsilon5, FS2003 and FS2009USB programmers
- Converts the programmer pin-out to the standard 20-way IDC ARM JTAG connector suitable for plugging into any ARM Target Board
- Supports powering of the Target System with a regulated 3.3V supply from the programmer
- Supports powering of the programmer from a Target System running at 3.0 – 5.V
- Provides 47k ohm pull-up resistors on the JTAG signal lines

1.2 Programmer compatibility

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

- Epsilon5 MK2, Epsilon5 MK3 – Portable ISP Programmer
- FS2003 – Portable ISP Programmer
- FS2009USB – Portable ISP Programmer

1.3 Power Supply - Selection Jumper

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

Link pins	Link pos	Powering scenario	Voltage (V)
1-2	IN	**Target System powers the Programmer** • The Target System voltage is fed directly to the programmer and will be used to power the programmer.	3.0 – 5.0V
2-3	OUT	**Programmer powers the Target System at 3.3V (default)** • The programmer must be powered from an external power supply via the DC Jack Socket in the range of 3.5 to 5.0V. • A 3.3V regulator on the ***'NXP LPC2xxx ISP Cable'*** then feeds 3.3V down to the Target System.	3.3V

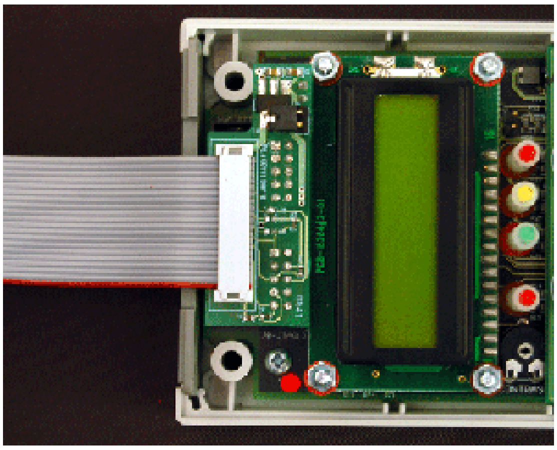
Warning!
If you select link position ***'IN'*** and then input +5V into the DC Jack Socket of the programmer, this will feed +5V directly to the Target System. This +5V could damage circuitry on the Target System if it is only designed to run at 3.3V.

1.4 Cable Installation Instructions

These instructions detail how to fit the ***'ARM ISP Cable'*** to an Epsilon5, FS2003 or FS2009USB programmer.

Instructions:

- The ***'ARM Programming Cable'*** has a small circuit board on one end which plugs into the ISP headers inside the programmer. On the other end it features a 20-way IDC plug.
- Remove the top cover of the programmer.
- Make sure the both the programmer and your Target System are powered OFF.
- Plug the circuit board end of the ***'ARM Programming Cable'*** into the programmer ISP headers – see example picture of cable fitted to FS2003 / FS2009USB programmer below.



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

Important note

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

1.5 Getting Started

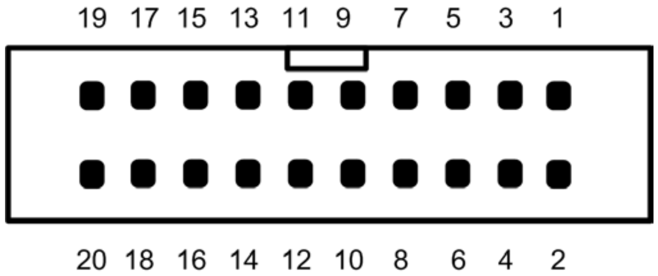
Once you have the installed into the programmer and connected to your Target System, then you are ready to program an ARM device.

Instructions:

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

1.6 ARM JTAG Debug Header

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

	***ARM JTAG Debug Header*** The connector is a 20-pin bump-polarised IDC connector with 0.1" pin spacing. Pin 1 is the top right pin as shown in the diagram opposite.