
EQTools Version 2.1.0 - Release - Build 888

Software Title: EQTools Version 2

Version: 2.1.0 build 888

Release date: 6th March 2008

Supported Programmers

EQTools Version 2 supports the following Equinox programmers:

- Epsilon5
- FS2003
- PPM3 MK2

***** The FS2000A firmware does not currently support some of the functionality implemented in this release of EQTools. Please use EQTools build 837 for the FS2000A programmer. *****

Please note:

The PRO101 and PPM3 MK1 programmers are NOT supported by **EQTools Version 2**. Please use **EQTools Version 1** for these programmers.

Software versions

This release of EQTools contains the following software utility versions:

- EQTools V2.1.0 build 888
- Upload Wizard V2.1.0 build 885
- ConsoleEDS V2.1.0 build 888

New Programmer Firmware V3.07 required for this EQTools release

Firmware rev 3.07

This version of EQTools requires that you install firmware version 3.07. You can find this firmware upgrade project in the **program files\equinox\firmware** directory once EQTools has been installed.

The new features implemented in this firmware are as follows:

- Faster programming of Atmel ATmega AVR microcontrollers using SPI – EEPROM Page Based EEPROM Programming Mode
- Faster Power Supply voltage Settling time (PPM3-MK2 only)
- Leave Power Supply on at end of Standalone Programming Project (PPM3-MK2 only)
- Faster JTAG programming of Atmel ATmega AVR microcontrollers
- JTAG-in-a-chain programming support for Atmel ATmega AVR microcontrollers

- JTAG ID can now be read and validated (works for both Atmel and non Atmel JTAG devices)
- JTAG chain validation now possible
- Skip 'Standalone Project Image Check' in manual Keypad mode
- Skip 'Standalone Project Image Check' in 'ASCII Text Communications' mode. (new command 'FPROGRAM')
- Locked Keypad Projects – new functionality
- RESET_PPM command now does not wait after the command has been issued.

Firmware upgrade instructions:

You can upgrade your programmer firmware by uploading the relevant 'Firmware Upgrade Project Collection File' (*.ppc) using EQTools / Upload Wizard. For detailed Firmware Upgrade instructions, please see Application Note AN112.

New features / functionality in this release

1. EQTools

i. JTAG-in-a-chain functionality now implemented

This EQTools version now supports programming of any Atmel ATmega AVR device via JTAG when the device is part of a JTAG chain. It is possible to declare the position of the AVR Device in the JTAG chain by specifying the number of 'Bits before' and 'Bits after' the target device in the chain. Please see Application Note – AN101 for further details.

ii. Supports reading / validation of the JTAG ID + Silicon Revision check

It is now possible to read the 4-byte 'JTAG ID' from any JTAG device. This allows the programmer to check not only the Target Device Part Code, but also the 'Silicon Revision'. This can be useful for tracking that the correct part (correct revision) has been fitted to each Target Board. The software supports reading the JTAG ID of any JTAG device, not just Atmel AVR devices.

iii. JTAG Chain Validation

It is possible to validate all the devices in a 'JTAG Chain' to ensure all devices are correctly fitted to the PCB. A series of 'Chained Projects' is used to interrogate each JTAG Device in the chain for its JTAG ID and once all ID's have been verified, this proves that the JTAG Chain is complete and all devices are responding. This function will work with an JTAG device.

iv. Faster ATmega AVR JTAG programming algorithms

The speed of the ATmega AVR JTAG programming algorithms has been optimised. An overall decrease in the programming time should now be seen with all AVR JTAG devices.

v. Faster programming of Atmel AVR microcontrollers using SPI - Page Based EEPROM Programming Mode

A new algorithm has been implemented in 'SPI Programming Mode' which supports faster programming of the EEPROM area of many Atmel AT90S, ATtiny, AT90USB, AT90CAN and AT90USB microcontrollers. It is now possible to program the EEPROM in pages of eg. 4 or 8 bytes instead of programming just a single byte at a time. This can reduce the overall EEPROM Programming time considerably as otherwise many AVR devices take 4 – 8ms per EEPROM byte to program. The time saving can be in the order of ¼ or 1/8 of the original time to program the EEPROM when the Page Size was set to 1 byte.

Please note:

- For existing projects, the EEPROM Page Size will remain at 1 byte. This can be changed to the higher Page Size on the 'Target Device' tab in the project. This maintains compatibility with projects compiled using an older version of EQTools and also with firmware 3.04 and below.
- For newly created projects, the EEPROM Page Size will default to the highest possible value for the selected device. Eg. 4 or 8 bytes. This project is then only compatible with firmware 3.05 and above.

vi. Faster RDY/BUSY FLAG Polling algorithm (SPI Programming Mode)

A new algorithm has been implemented in 'SPI Programming Mode' which supports faster programming of both the FLASH and EEPROM areas by polling the RDY/BUSY FLAG. This algorithm should produce marginal programming time improvements as the programmer no longer needs to read back a byte each time it needs to poll the device.

Please note:

- For existing projects, the 'Polling Method' will remain at 'Conventional Byte Polling algorithm'. This can be changed to the 'Ready/Busy' polling algorithm on the 'Target Device' tab in the project. This maintains compatibility with projects compiled using an older version of EQTools and also with firmware 3.04 and below.
- For newly created projects, the 'Polling Method' will default to the 'Ready/Busy' polling algorithm. This project is then only compatible with firmware 3.05 and above.

vii. Power up – Voltage Stabilise Time reduced (PPM3-MK2 only)

A new power supply algorithm has been implemented for the PPM3-MK2 programmer. When the programmer is powering the Target System, the 'Target Voltage' specified in the Programming Project is now generated at the start of the first project to be executed. The programmer then retains this 'Target Voltage' internally inside the programmer so that the next time the power supply is switched on to the Target System, the 'Target Voltage' is already very close to the specified value. In other words, if the 'Target Voltage' is +5.0V, then the programmer will maintain +5.0V internally so the

next time the power is switched on to the Target System, the voltage generation time will be much faster.

viii. Leave Power Supply ON at end of Standalone Programming Project (PPM3-MK2 only)

A new option has been added to the 'Power Supply Control' of the PPM3-MK2 programmer to allow the Power Supply to be left switched ON at the end of a Standalone Programming Project. This can be used to permanently power the programmer Line Drivers at a specified voltage even when the programmer is not actually programming.

The power supply can be switched on permanently by running the first project and then will remain on until switched off by another project.

ix. Target Vcc Voltage now defaults to lowest voltage supported by selected device

The 'Target Power Supply' GUI has been changed so that when a new device is selected, the lowest voltage supported by selected device is automatically selected. This is to stop the incorrect voltage eg. +5V being applied to a 3.3V Target System by mistake.

x. Skip Standalone Project - Image Check

When a project is uploaded to the programmer and then executed, the programmer then automatically scans the entire Programming Project in FLASH Memory and performs a CRC check. This is a safety precaution in case the data in the programmer FLASH Memory has been corrupted.

It is now possible to instruct the programmer not to perform the 'Project Image Check' on Standalone Programming Projects. The check can be disabled for both 'Manual Keypad' and 'ASCII Text Communications' mode. This can save a few seconds per programming cycle as the Image Check is quite time consuming for larger FLASH Size devices.

To disable the '**Standalone Project - Image Check**':

- Open your existing programming project
- On the **<Programmer and Project Type>** tab, check the box '**Skip Image Check**'
- Re-compile and upload the project to the programmer.

xi. Locked Keypad – Standalone Projects

The ability to lock the keypad of an FS2003 or PPM3-MK2 programmer once a Programming Project has initially been selected was implemented back in firmware 3.02.

This functionality has now been modified slightly as follows:

- If a project is locked then once the PASS / FAIL message has been displayed, the programmer will now always go back to the screen 'Press YES to program Target System'.
- At the end of a project, only the **<YES>** key will now cause the project to be executed again.

xii. AT89S8253 – Hardware SPI Mode now implemented

If firmware 3.07 is installed in the programmer, it is now possible to use the 'Hardware SPI' mode to program the AT89S8253 device. This results in faster and more reliable programming of this device.

2. ASCII Text Mode Communications

i. New 'FPROGRAM' command

A new command 'FPROGRAM' has been implemented in ASCII Mode. This command is exactly the same as the existing 'PROGRAM' command except that the 'Image Check' is skipped when the project is executed. This can lead to much faster programming for projects with large FLASH / EEPROM files.

3. ConsoleEDS

i. Reading the JTAG ID

It is now possible to check the 'JTAG ID' of any JTAG device (Atmel or any other manufacturer) using the '/READSIG' command. The ability to read the 'JTAG ID' is set up from the Programming Project by checking the 'Read JTAG ID' check box.

New Device support

The following devices have been added to EQTools in this release:

Atmel - ATtiny AVR Family (SPI):

- ATtiny13-20 (SPI)
- ATtiny2313-10 (SPI)

Atmel – AT90USB AVR Family (SPI):

- AT90USB82 (SPI)
- AT90USB162 (SPI)

Atmel ATmega AVR Family (SPI):

- ATmega169V-1 (SPI)
- ATmega169-16 (SPI)
- ATmega3290 (SPI)
- ATmega3290V (SPI)
- ATmega644V-20 (SPI)
- ATmega644-20 (SPI)

Atmel ATmegaxxP 'PICO Power' AVR Family (SPI):

The 'PICO Power' AVR devices have now been listed as a separate Atmega family to make them easier to distinguish from the standard devices.

- ATmega48P-20
- ATmega48PV-20MMU
- ATmega48PV-10
- ATmega88PV-10
- ATmega88P-20
- ATmega168PV-10
- ATmega168P-20
- ATmega329P-20
- ATmega329PV-10
- ATmega3290P-20
- ATmega3290PV-10

Atmel ATmega AVR Family (JTAG):

- ATmega169V-8 (JTAG)
- Atmega645-8 (JTAG)
- Atmega645-16 (JTAG)
- Atmega6450-8V (JTAG)
- Atmega6450-16 (JTAG)
- Atmega325V-8 (JTAG)
- Atmega325-16 (JTAG)
- Atmega3250V-8 (JTAG)
- Atmega3250-16 (JTAG)

Atmel ATmegaxxxP 'PICO Power' AVR Family (JTAG):

- ATmega329P-20 (JTAG)
- ATmega329PV-10 (JTAG)
- ATmega3290PV-10 (JTAG)
- ATmega3290P-20 (JTAG)

Problem / Bug corrections

EQTools

i. EDS Blank Check of EEPROM – Byte / Page problem

Problem description:

In EDS mode, the Blank Check operation was checking the incorrect address range for AVR devices as it was expecting the address range specified in PAGES not BYTES.

Fix:

This problem has now been fixed.

ii. Zensys – Page Write time defaulted to incorrect value

Problem description:

The 'Page Write time' for Zensys devices was defaulting to 31ms in the EQTools GUI. This was just outside the 'Safe working range' for most Zensys chips so the FLASH programming could either fail completely or be unreliable.

Fix:

The 'Page Write time' for Zensys devices now defaults to 20ms which should operate properly for all devices. This also gives the fastest possible programming speed.

iii. AT89S8253 'Hardware SPI' algorithm not implemented

Problem description:

The AT89S8253 'Hardware SPI' algorithm was not actually implemented in firmware 3.04 – 3.06 so selecting 'Hardware SPI' in EQTools would always cause the programming to fail.

Fix:

The AT89S8253 'Hardware SPI' algorithm is now implemented in firmware 3.07. This has been fully tested and appears to work reliably now. The 'Hardware SPI' method should now be used in preference to 'Software SPI' as it provides a defined 50:50 mark space ration clock signal. This gives much more reliable serial programming of the AT89S8253 device.

ConsoleEDS

i. Problem with writing of Hex files with non 0x0000 start address

Problem description:

If the /FLASHWRITE or /EEPROMWRITE commands were used to program an Intel Hex file which had a Start Address which was not 0x00000, then ConsoleEDS would report 'PASS' when in fact the data was not written properly.

Fix:

This problem has been fixed in ConsoleEDS build 882 and above. It is now possible to program an Intel Hex file where the start address of the data in the file is a non zero address.

ii. Problem with writing data to a Hex files with non 0x0000 start address

Problem description:

If the /FLASHREAD or /EEPROMREAD commands were used to read data from a Target IC to an Intel Hex file and the Start Address specified was not 0x00000, then ConsoleEDS would pad the Hex File with 0xFF from address 0x00000 up to the specified Start Address.

Fix:

This problem has been fixed in ConsoleEDS build 888 and above. It is now possible to read data from a Target IC to a Hex file with a non 0x00000 Start Address.