



AVR-ISP500-ISO

Users Manual



All boards produced by Olimex are ROHS compliant

Rev.B, January 2009

Copyright(c) 2009, OLIMEX Ltd, All rights reserved

INTRODUCTION:

AVR-ISP500-ISO is professional serial in-system programmer for AVR microcontrollers, featuring optoisolation between PC and target AVR board. It implements the STK500v2 protocol as defined by Atmel which makes it compatible with a range of tools, including [AvrStudio](#) and [avrdude](#).

It is distinguished from other programmers by its support for autonomous operation. AVR-ISP500-ISO contains a 2Mb FLASH memory where it can store entire programming sessions, including FUSE, LOCK, EEPROM and FLASH write and verification, as well as signature checking. Just hold the button, perform all the necessary steps with AvrStudio, and disconnect the programmer from the PC. Next time button is pressed the programmer will invoke all previously recorded steps without the need for connection to the PC. We believe this feature makes AVR-ISP500-ISO the perfect tool for firmware upgrades in remote places.

FEATURES:

- Fully STK500v2 compatible;
- Works with AvrStudio, WinAVR, Avrdude and every other software compatible with STK500v2;
- USB port for connection to PC;
- Opto-isolation 1000VDC between PC and target AVR board allows programming of targets under high voltages;
- Stand alone operation without the need for PC connection, allowing firmware mass programming on many devices with single button press;
- One bi-color LED for current operation status;
- One bi-color LED for last autonomous operation status;
- Supports both standard ICSP10 and ICSP6 connectors;
- Button for initiating autonomous operations and command logging;
- Powered either by USB or external AC/DC power supply;
- External clock output on ICSP10 pin 3 for rescuing AVR's with enabled external clock fuse;
- Supports target voltages ranging from 1.8V to 5.5V.
- ISP clock frequencies ranging from 5kHz to 1.843MHz.

ELECTROSTATIC WARNING:

The AVR-ISP500-ISO board is shipped in protective anti-static packaging. The board must not be subject to high electrostatic potentials. General practice for working with static sensitive devices should be applied when working with this board.

REQUIREMENTS:

Cables: 1.8 meter USB A-B cable. Ten- or six-wire ribbon cable for connection to target AVR chip.

Power: External 9-12V DC or 6-9V AC power supply (needed only for autonomous operation when USB is not plugged).

Software: Software with support for STK500v2 protocol:

- AvrStudio, available from Atmel.
- avrdude, included in the WinAVR distribution.

SUPPORTED MICROCONTROLLERS:

The following AVR microcontrollers are supported for programming:

- Classic 8-bit AVR.
- megaAVR
- tinyAVR
- USB AVR

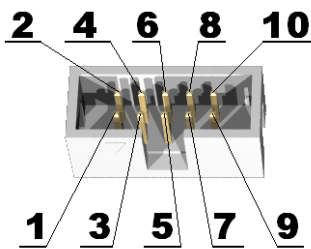
The following AVR microcontrollers are **not supported**:

- XMEGA
- AVR32

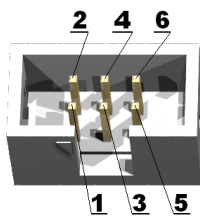
The following programming methods are **not supported**:

- JTAG
- debugWire
- Parallel High Voltage Programming
- Serial High Voltage Programming
- PDI

CONNECTOR SCHEMATICS:



ICSP10		
pin	Abbrev.	description
1	MOSI	Serial Output
2	V_TAR	Target VCC
3	CLKO	Clock output
4	GND	Ground
5	TRST	Target RESET
6	GND	Ground
7	SCK	Serial Clock
8	GND	Ground
9	MISO	Serial Input
10	GND	Ground



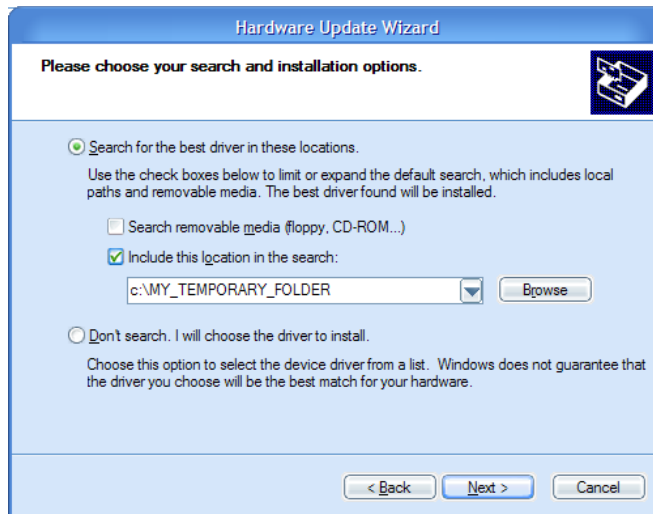
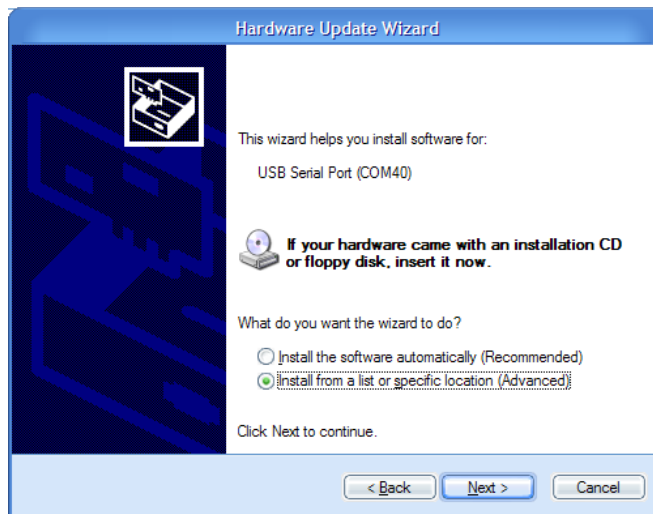
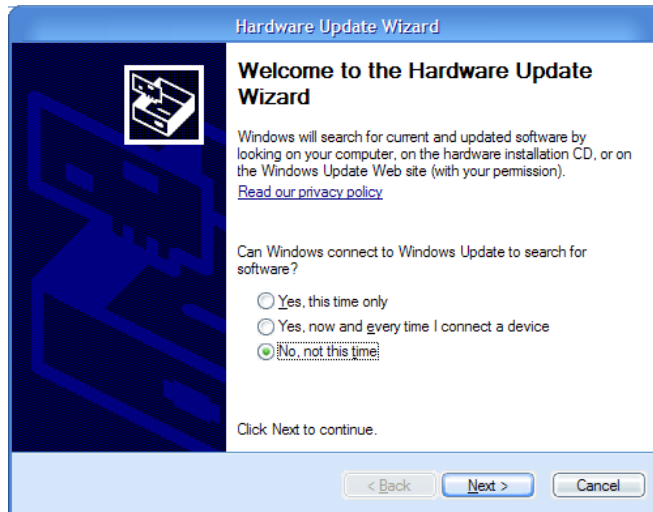
ICSP6		
pin	Abbrev.	description
1	MISO	Serial Input
2	V_TAR	Target VCC
3	SCK	Serial Clock
4	MOSI	Serial Output
5	TRST	Target RESET
6	GND	Ground

PC DRIVER INSTALLATION:

Download drivers for your Operating System from our website. Windows installation steps are the following:

1. Download and unzip the file "AVR-STK500-ISO-drivers.zip" in a temporary directory.
2. Plug the programmer in the USB port.
3. Point the Device Wizard to the temporary directory.
4. Click finish.

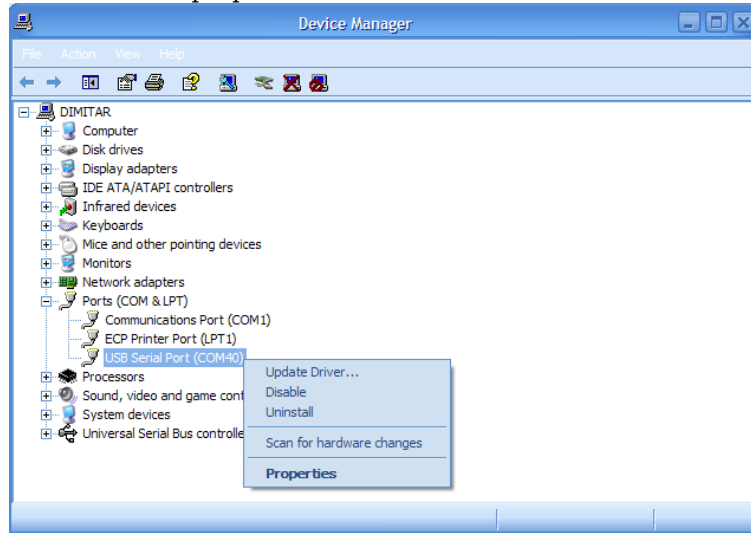
Screenshots of the steps are shown below:



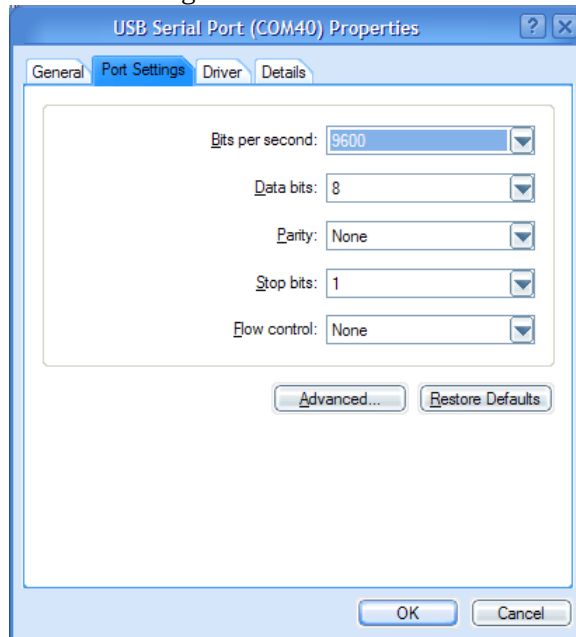
WARNING: The COM port number assigned by Windows to AVR-ISP500-ISO must be COM4 or below. Otherwise AvrStudio might not be able to detect the programmer. Here are the steps to change it:

1. Go to Device Manager.

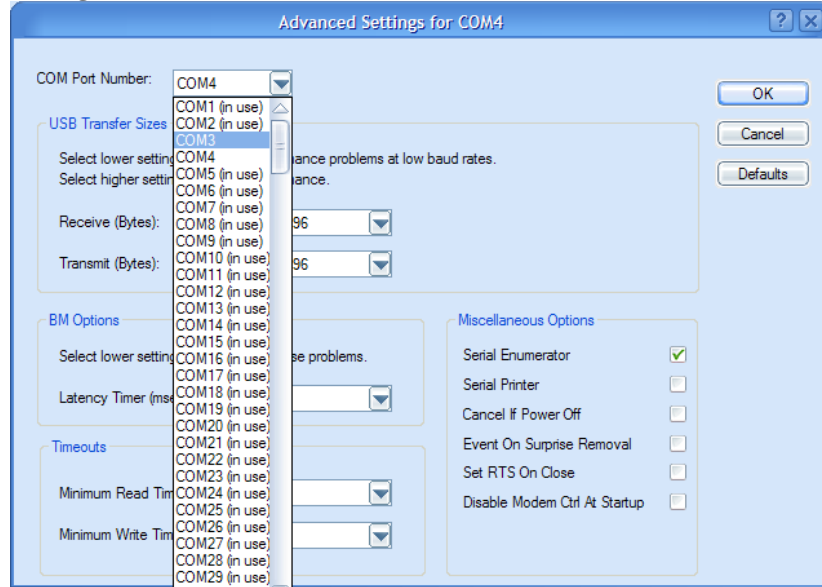
2. Unfold “Ports (COM&LPT)” and right-click on “USB Serial Port (COMxx)” where COMxx can be anything between COM1 and COM255. Select properties.



3. Go to the “Port Settings” tab and click the “Advanced” button.



4. Change the “COM Port Number” to COM3 or COM4.

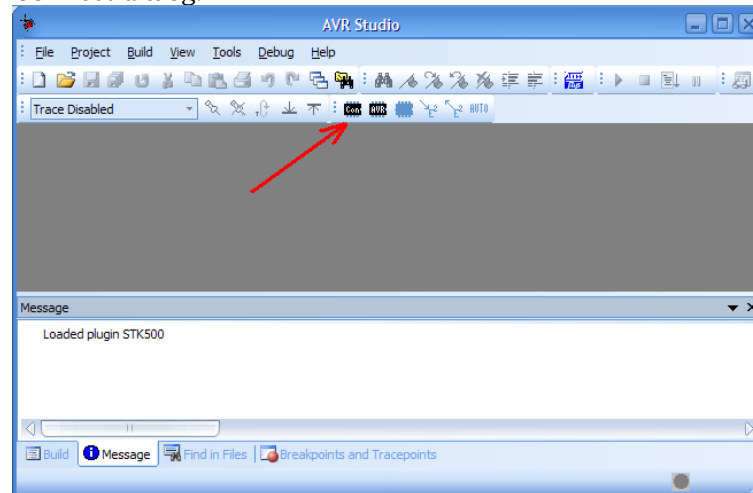


5. Click OK.
6. If a warning message pops up and complains about COM port being used by another device, click “Yes”.
7. Click OK to close the device properties.

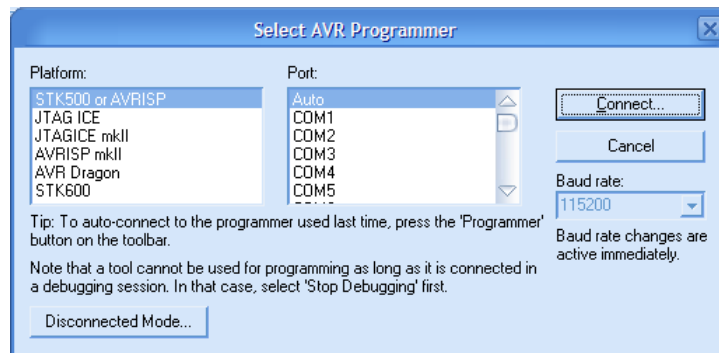
USING AVR-ISP500-ISO WITH AVRSTUDIO:

WARNING: The COM port number assigned by Windows to AVR-ISP500-ISO must be COM4 or below. Otherwise AvrStudio might not be able to detect the programmer. See the section for PC Drivers Installation for more information.

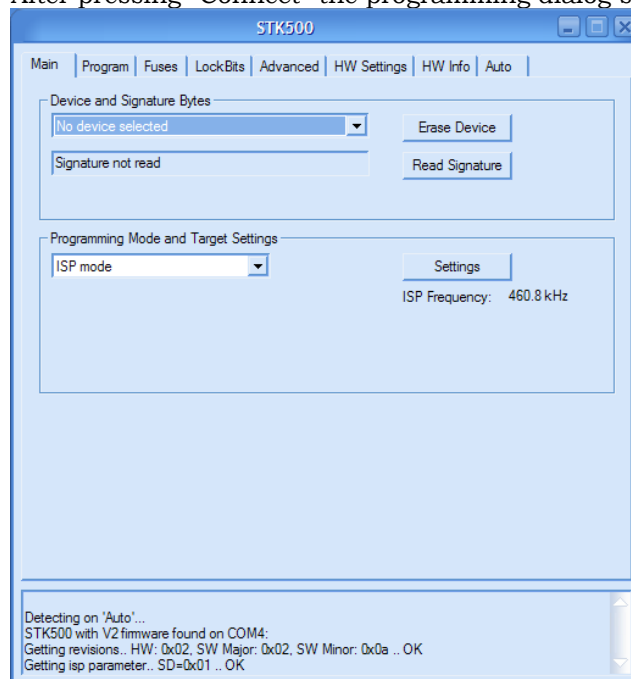
Usage under AvrStudio is straightforward. First open the Programmer Connect dialog:



Then select “STK500 or AVRISP option” with automatic port detection:



After pressing “Connect” the programming dialog should appear:



Target AVR now can be erased, flashed with a provided HEX file, FUSES and LOCK bits can be written and/or verified. For more information please consult the AvrStudio documentation.

CAVEAT: Although the programmer will happily accept setting VTARGET and ARef in the “HW SETTINGS” tab, these actions will have no effect. The programmer hardware can only read target VCC and show it simultaneously in the VTARGET and ARef sliders.

AUTONOMOUS OPERATION:

When the button is held pressed for 5 seconds programmer enters command logging state. In this state it will execute and log all commands received from the PC software in its internal FLASH memory. When the button is pressed again, programmer will enter its normal state where commands will be executed but not logged.

Logged commands can be executed again without the presence of a connected PC by simply pressing quickly the programmer button while in normal state. The bi-color LED will glow yellow while operation is in

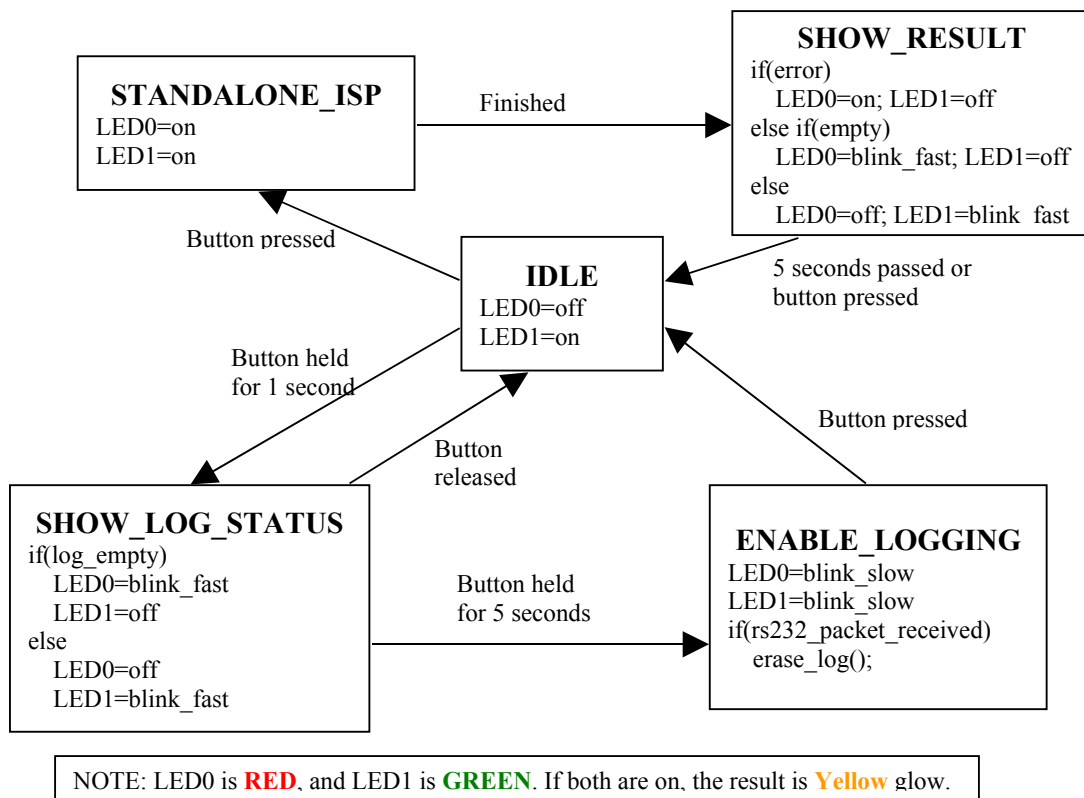
progress. After operation finishes successfully the GREEN LED will blink for a while, otherwise on error the RED LED will blink for a while. After that the programmer will enter normal state where only the GREEN LED is on constantly.

The right LED of AVR-ISP500-ISO is a bi-color LED which shows the last autonomous operation status for indefinite time. Again, RED is for error, GREEN is for success, off if no status is available due to ongoing operation. This LED is also off after power up to signal that no autonomous operation has taken place so far. This LED is helpful for long autonomous programming sessions where user is not required to look for the end of a programming session.

If programmer button is pressed and held for more than 1 second but less than 5 seconds, it will show the current log status. A blinking GREEN indicates a valid log, and blinking RED indicates invalid/erased log.

The formal definition of the programmer states is given below.

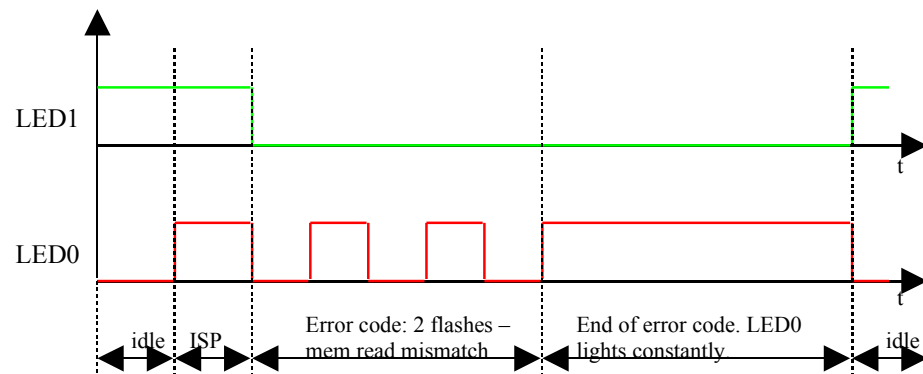
AVR-ISP500 STATES



In case of autonomous operation error, the RED LED will blink for a while. The number of flashes indicates the error that caused the failure. Error codes are given in the table below.

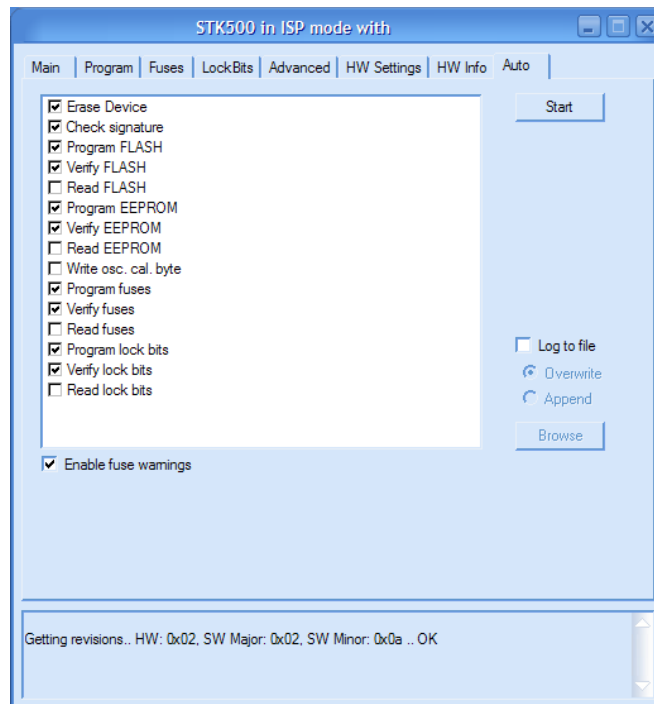
Number of RED flashes	Action that failed
1	Enter programming mode.
1	Signature read mismatch.
2	FLASH read mismatch.
2	EEPROM read mismatch.
2	FUSE read mismatch
2	LOCK read mismatch
3	Corrupted log
3	Empty log
4	Other command/answer error.

Example:



CAVEAT: RC calibration in autonomous mode is **not** supported. The reason is that the programmer has no way to distinguish RC value writes to FLASH from normal data ones.

TIP: Sometimes AvrStudio will issue FUSE/LOCK reads when the corresponding tabs are selected. This, however, will be recorded by the programmer if it is in command logging state. So for best results it is recommended to issue all commands in a predefined order. This is achieved by the “Auto” feature of AvrStudio Programming Dialog. After setting the FLASH and EEPROM files, FUSE and LOCK settings, and selecting the target AVR device, switch to the “Auto” tab of the programming dialog.

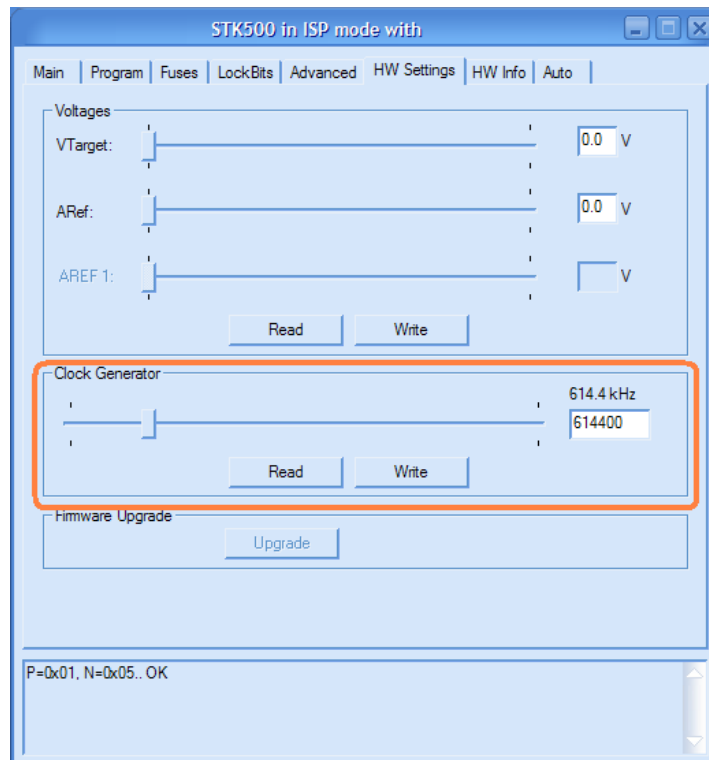


Select the operations that should be performed and hold the programmer button for 5 seconds to activate command logging. Then hit the “Start” button to initiate a programming operation. After “Auto” operation finishes successfully, turn off command logging.

TIP: If target signature must be verified during autonomous operation, READ SIGNATURE command must be issued while in command logging state.

CLOCK OUTPUT:

Pin 3 of the ICSP10 connector is normally left unconnected by other ISP programmers. Not ours. In normal programmer state this pin is tri-stated. During programming operation, however, this pin is made output and a square-wave clock is generated. Clock frequency can be adjusted with the AvrStudio Programmer Dialog:



After programming operation finishes, this pin is again tri-stated in order not to interfere with the target circuit.

This clock output can be really helpful when target AVR is accidentally programmed with External Clock FUSE option. To resurrect it just wire ICSP10 pin 3 to XTAL1 pin of the target AVR chip and initiate a programming session to fix the FUSE values.

CAVEAT: If an error occurs in standalone mode, the “Leave Programming Mode” command will not be executed, so the external clock output will stay active. Due to the AVR-ISP500 hardware implementation, this can cause target VCC to be incorrectly read as Vtarget=5V.

TIP: If clock output is needed when target is not being programmed then the following steps must be followed. Make sure that programmer contains a valid log, no matter its contents. Disconnect the programmer, then press the button, and wait for RED LED to blink for error. After that the clock output will stay activated.

FIRMWARE UPGRADE:

All AVR-ISP500 devices contain a built-in boot loader for easy firmware upgrade. Device enters boot loader mode if the application FLASH section is corrupted. To force the device to enter boot loader mode for manual firmware update do the following:

1. Disconnect the programmer from any power source (external AC/DC, USB).
2. Press and hold the button.
3. Power up the device by connecting it to USB.
4. Device now must be in boot loader mode, indicated by the LED light sequence:

- a. RED on, GREEN off.
- b. RED off, GREEN on.
- c. RED off, GREEN off.

Button can now be released. The device will stay in boot loader mode until it is power cycled.

All boot loaders implement the standard protocol XMODEM with CRC16 for firmware update. User is free to use his favorite terminal client (HyperTerminal, minicom, etc) to upload the firmware images taken from our website. The serial port settings for the AVR-ISP500-ISO boot loader are: 9600 baud/sec, 8 data bits, no parity, 1 stop bit. The AVR-ISP500-TINY/MASS/NANO boot loaders implement a USB virtual serial port that is baudrate agnostic.

As an alternative we provide a simple Windows GUI application for users who don't want to or cannot use terminal software.

After the firmware image is uploaded the target will blink the GREEN LED if update was successful, otherwise it will blink RED LED if firmware image was invalid or update was unsuccessful. Device stays in this state until it is power cycled.

TROUBLESHOOTING GUIDE:

Problem: AVR Studio cannot find my programmer.

Probable causes and solutions:

- Look whether the programmer is listed in Device Manager under the "Ports (COM & LPT)" section. If it's not there then check your USB cables and hubs. Reinstall the driver.
- Check that the green LED is constantly on. If not then the programmer might be in firmware upgrade mode. Go to the firmware upgrade section for more details.
- Your serial port number might be too high. Check the manual section "Installing drivers" for more information on assigning serial port numbers. Also go to AVR Studio menu "Tools->Options" and set the "Number of COM ports to try" field to at least 20.
- Remove all applications that might be using or scanning your computer's serial ports, including any serial port monitors. They might mess up the STK500v2 communication between AVR Studio and the programmer.

Problem: Programming fails. Target AVR chip cannot enter programming mode.

Probable causes and solutions:

- ISP frequency might be too high. Set the ISP frequency to well below $\frac{1}{4}$ of target MCU clock. Please note that the AVR MCU clock depends on its fuses configuration.
- Target power might not be stable enough or supply voltage might be too low. Check your target board circuit Ensure that the used AVR chip specifications match your supply voltage.
- Target power supply VCC might not be connected to pin 2 of ICSP6/10. Check your target board circuit.
- The target AVR MCU might have fuses programmed for disabling SPI ISP or selecting another programming method. In such case you'll need a High Voltage Parallel programmer to unlock the chip.
- The reset line cannot be pulled low by the programmer. Check your schematic and ensure that only a weak pull-up (and possibly a small capacitor) are connected to RESET. Otherwise, if the pull-up

is too small or there is another output driving RESET, the programmer won't be able to pull it down.

- Another circuit is driving the ISP lines (MISO, MOSI, SCK or RESET) resulting in a contention with the programmer. The solution is to remove all such circuits (LEDs, RS232 drivers, resistors below 2k, etc) from the four ISP lines during programming. AVR-ISP500 is more susceptible to this kind of fault than other programmers on the market because it has 560 ohm resistors in series with all its outputs. This protects both the target MCU and the target board circuit.

ELECTRICAL CHARACTERISTICS:

Symbol	Description	Condition	Min	Typ	Max	Units
V _{TARG}	Target Supply Voltage.		1.8		5.5	V
F _{ISP}	ISP Frequency on SCK SPI pin.		5		1843	kHz
V _{CC}	Programmer Power Supply Voltage.			5		V
I _{CC}	Programmer Power Supply Current.	Idle		92		mA
		Autonomous operation, V _{TARG} =1.8V, F _{ISP} =1.843MHz, powered from USB		110	160	mA
		Programming with AvrStudio, logging enabled, V _{TARG} =1.8V, F _{ISP} =1.843MHz, powered from USB		110		mA
	Time for a full programming and verification of ATmega128 (128k FLASH + 4k EEPROM + FUSES + LOCK + Signature check)	Autonomous operation, V _{TARG} =5V, F _{ISP} =1.843MHz		51		s
		Programming with AvrStudio, logging enabled, V _{TARG} =5V, F _{ISP} =1.843MHz		105		s
		Programming with AvrStudio, logging disabled, V _{TARG} =5V, F _{ISP} =1.843MHz		94		s
	Time for a FLASH programming and verification of ATmega128 (128k FLASH)	Autonomous operation, V _{TARG} =5V, F _{ISP} =1.843MHz		15		s
		Programming with AvrStudio, logging enabled, V _{TARG} =5V, F _{ISP} =1.843MHz		67		s
		Programming with AvrStudio, logging disabled, V _{TARG} =5V, F _{ISP} =2MHz		58		s

ORDER CODE:

AVR-ISP500-ISO - assembled and tested (no kit, no soldering required)

How to order?

You can order to us directly or by any of our distributors.

Check our web www.olimex.com/dev for more info.

Revision history:

REV.A - create April 2008

REV.B - modify January 2009

Added a troubleshooting guide

Disclaimer:

© 2008-2009 Olimex Ltd. All rights reserved. Olimex®, logo and combinations thereof, are registered trademarks of Olimex Ltd. Other terms and product names may be trademarks of others.

The information in this document is provided in connection with Olimex products. No license, express or implied or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Olimex products.

Neither the whole nor any part of the information contained in or the product described in this document may be adapted or reproduced in any material from except with the prior written permission of the copyright holder.

The product described in this document is subject to continuous development and improvements. All particulars of the product and its use contained in this document are given by OLIMEX in good faith. However all warranties implied or expressed including but not limited to implied warranties of merchantability or fitness for purpose are excluded.

This document is intended only to assist the reader in the use of the product. OLIMEX Ltd. shall not be liable for any loss or damage arising from the use of any information in this document or any error or omission in such information or any incorrect use of the product.