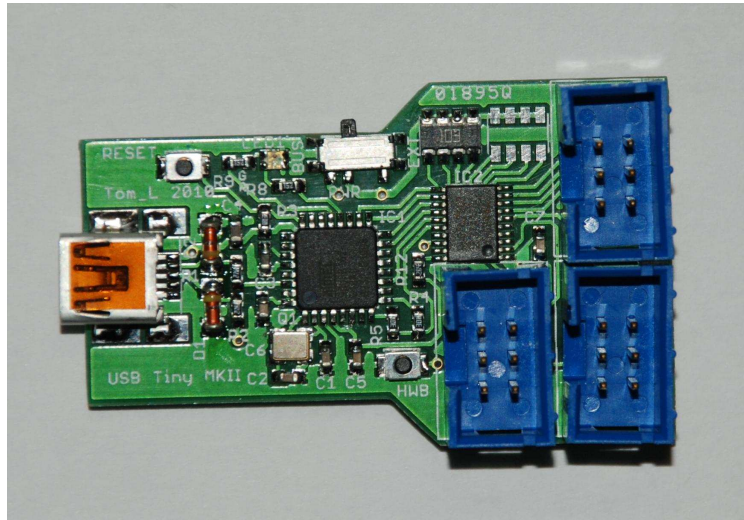




Contents

- 1) Overview
- 2) Switch Positions
- 3) Connector Pinouts
- 4) Using USBTinyMkII
- 5) Software Emulation mode
- 6) Preparing the Firmware
- 7) Firmware Update
- 8) Disclaimer Statement

Overview

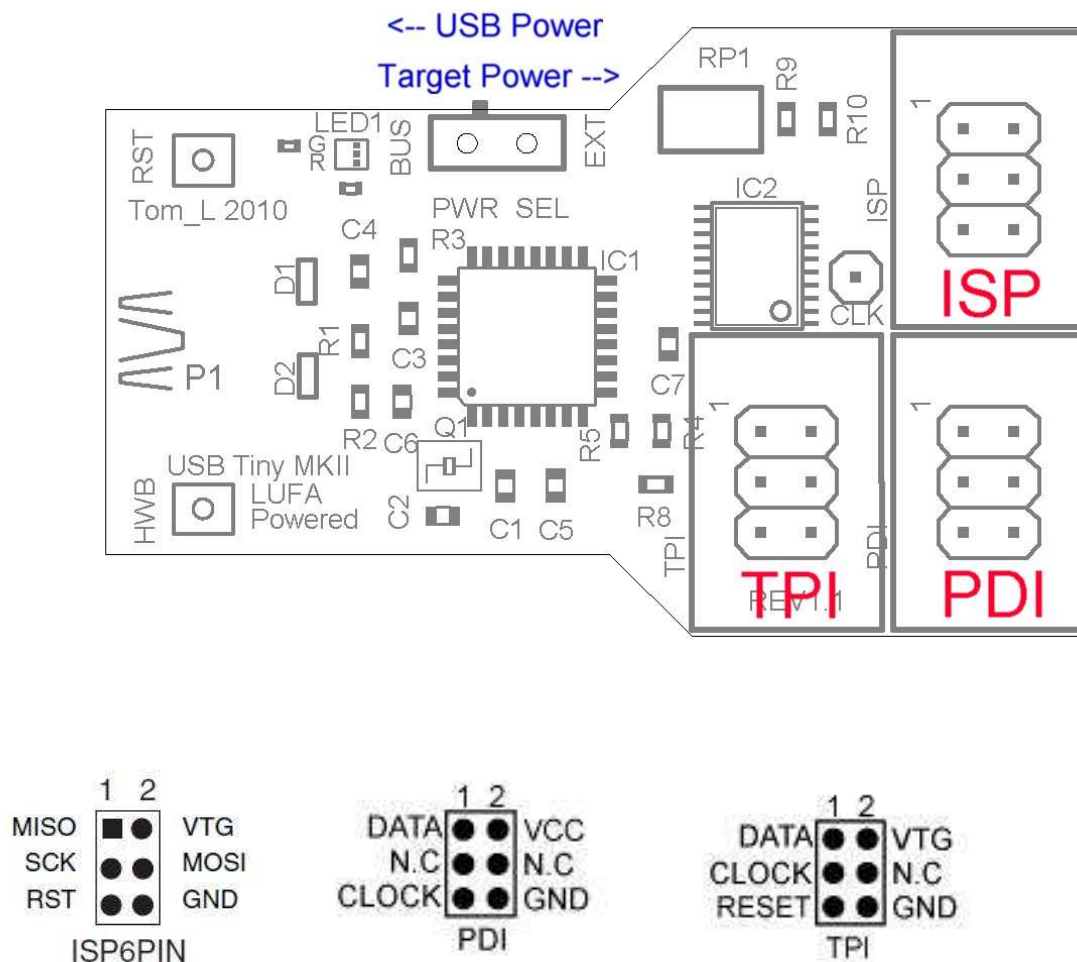
The USBTinyMkII is an avr programmer that uses the LUFA firmware to emulate an Atmel AVRISP-MkII. It supports the programming of the entire 8-bit AVR range via ISP as well as the PDI (for the newer XMEGA AVR) and TPI (for the 6-pin TINY AVR). Features include upgradeable firmware which can be configured for hardware or software emulation and switch selectable USB or target supplied power.

Switch positions

NOTE: *** **ONLY USE TARGET POWER IN PDI MODE** ***

The target chip can be powered from the USB VBUS (+5v) or from the target board itself. When the power select switch is in the position toward the USB plug it will be USB powered and when switched toward the target plugs it is powered by the target board as indicated below. The IC2 uses the power from the target device to determine the programming voltage. Target power has been tested from 1.8 to 5v. In USB mode, the target must be 5v tolerant. In PDI mode, xmega chips are not 5v tolerant!

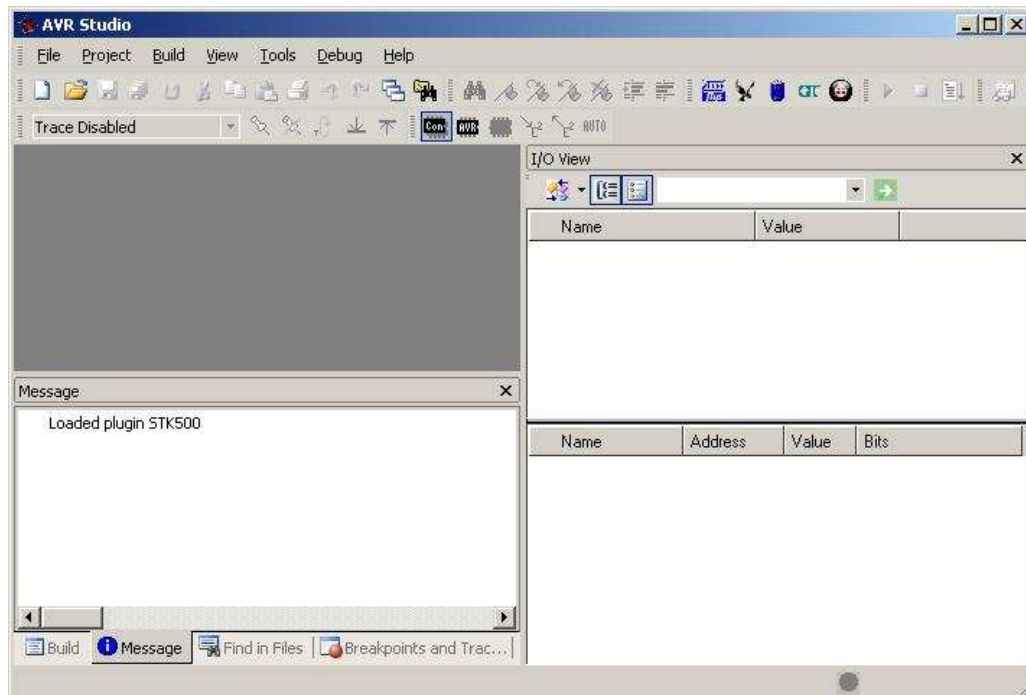
Connector Pinouts



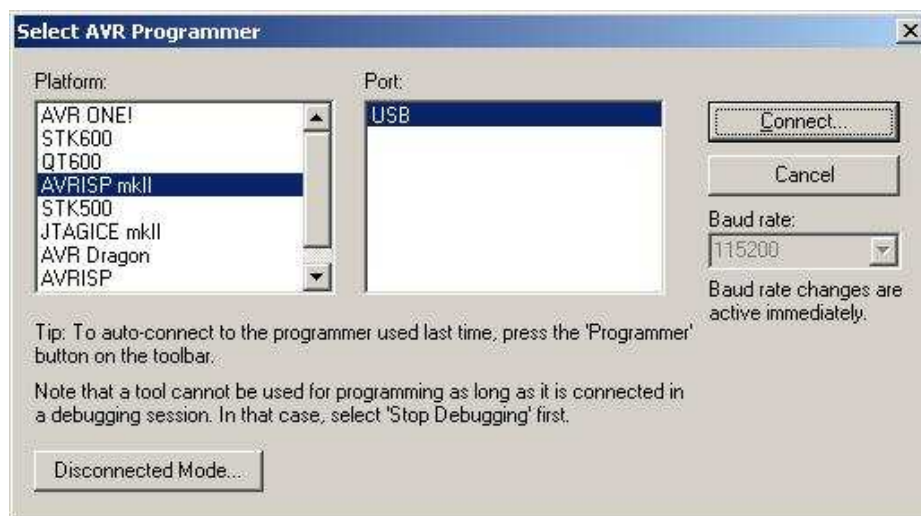
The ISP header is used for standard 8bit avr devices, the PDI header for the newer XMEGA AVRs and the TPI header for the 6-pin TINY AVRs.

Using USBTinyMkII

The USBTinyMkII is programmed to emulate a standard avrisp mkii. To connect to the programmer, plug it into your pc USB port and start up Studio 4.

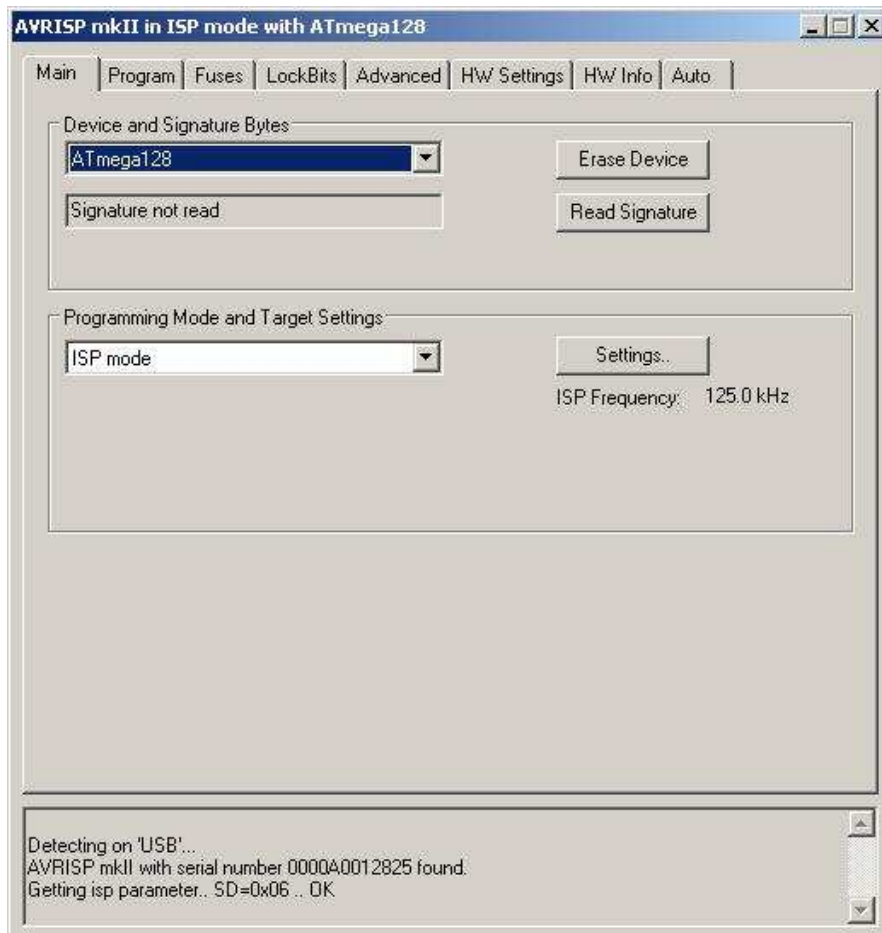


Click on the connect icon to open up the programmer selection dialog and select AVRISP MkII from the menu and press the connect button.



Once connected the device selection menu will pop up. Connect the programmer cable to the appropriate header to flash the target device depending on target device selected. Be sure your target device is 5v tolerant if you select the USB powered mode.

CAUTION: *** **ONLY USE TARGET POWER IN PDI MODE** ***



Software Emulation Mode

USBTinyMkII is configured for hardware mode when shipped, however [LUFA](#) can be configured for software emulation mode where the USBTinyMkII will use a single-header pinout identical to the official AVRISP-MKII. In software emulation mode the ISP PDI and TPI signals come from the single ISP header and the PDI and TPI headers are unused. Changing a line in the project makefile found in the installed AVRISP-MKII LUFA directory will alter how the programmer operates. By commenting line "**CDEFS += -DXPROG_VIA_HARDWARE_USART**" will set the option for software emulation mode. Uncommenting it will cause it to run in full speed hardware mode and is the

default configuration when shipped. Hardware mode brings the signals out the separate headers and runs faster than software emulation mode.

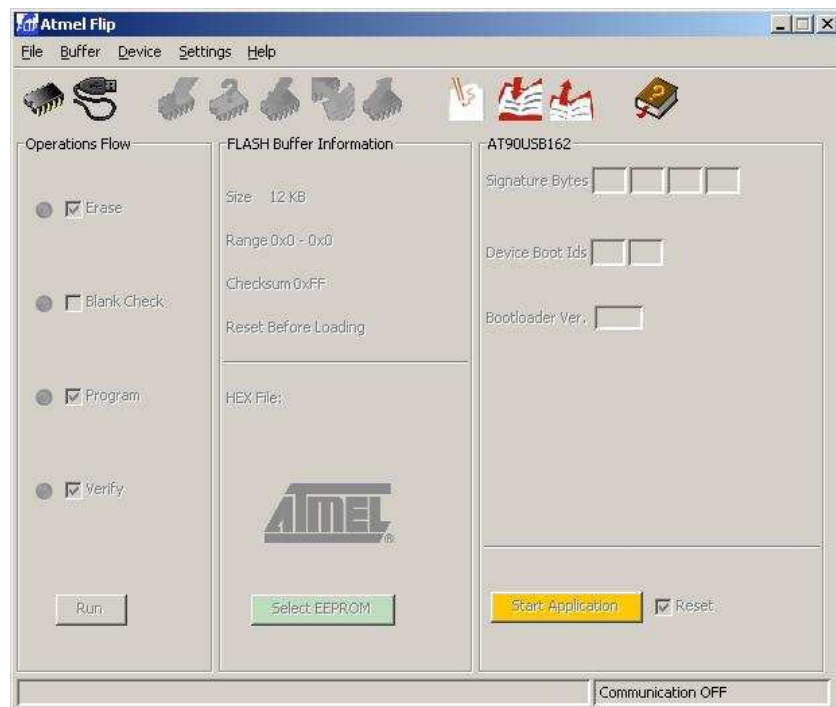
Preparing the Firmware

The firmware for the USBTinyMkII is entirely open source and can be downloaded from the [LUFA project](#). Board configuration files are available from the USBTinyMkII site. First copy the LUFA project to your hdd. Next uncompress the configuration files into the "Projects" folder. This will copy the modified makefile into the AVRISP-MKII directory and add a "Board" subdirectory to it, which has board specific files in it. Configure the makefile line for hardware or software emulation if you wish. Once configured run "Make Clean" and "Make All" from the AVRISP-MKII directory and follow the Firmware Update directions below. By default after compiling, the update file will be in the LUFA\Projects\AVRISP-MKII directory.

Firmware Update

To update the firmware you will need a copy of [FLIP](#) from Atmel (Flexible In-System Programmer) or [DFU-Programmer](#) (Device Firmware Update programmer). Following is a step by step to get the new firmware uploaded to your target programmer using FLIP.

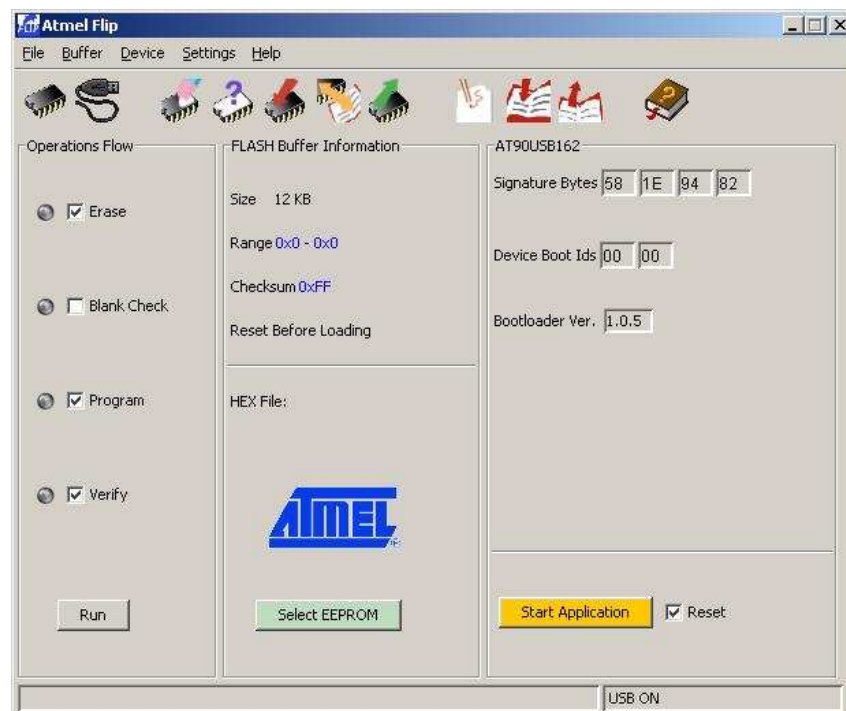
To put the USBTinyMkii into DFU programming mode, push and hold the 'HWB' button. While the 'HWB' button is depressed press the 'RESET' button then release both.



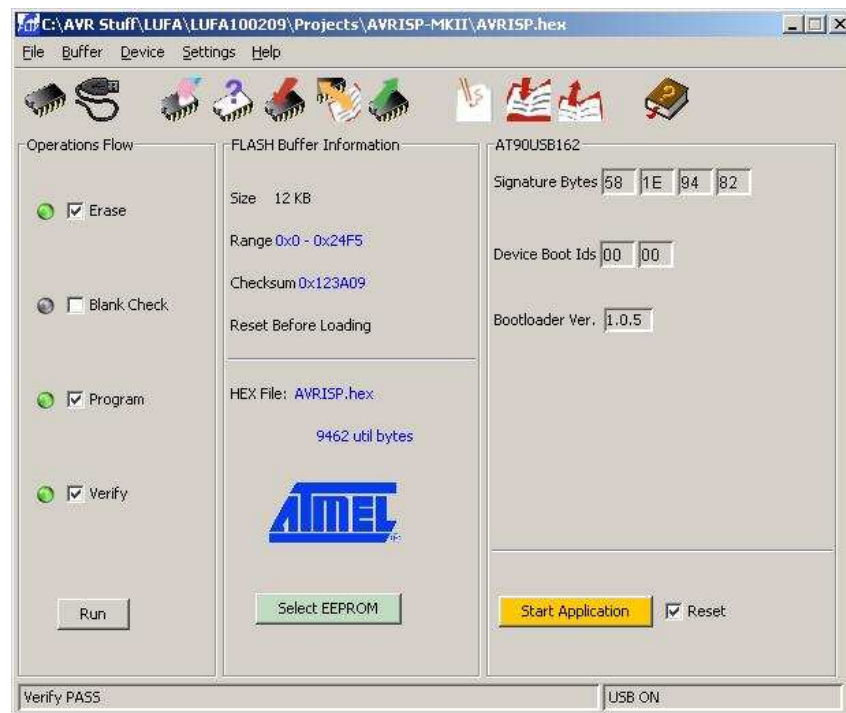
Click on the 'chip' icon or 'Device → Select' menu option and select the device from the dropdown menu (at90usb162).



Click on the 'USB cable' icon and select 'USB' from the menu and connect to the device.



Load the new hex file from the 'LOAD hex file' icon or from the 'File→ LOAD HEX file' menu. Click on the 'RUN' button in the lower left corner.



DONE! Your programmer should be up to date now.

Disclaimer Statement

No warranty is offered for the USBTiny MkII or its use or misuse. I can not be held liable for any damages caused to your USBTiny MkII programmer, target device or computer system. Having said that, I will attempt to ensure the highest level of quality and resolve any issue promptly.