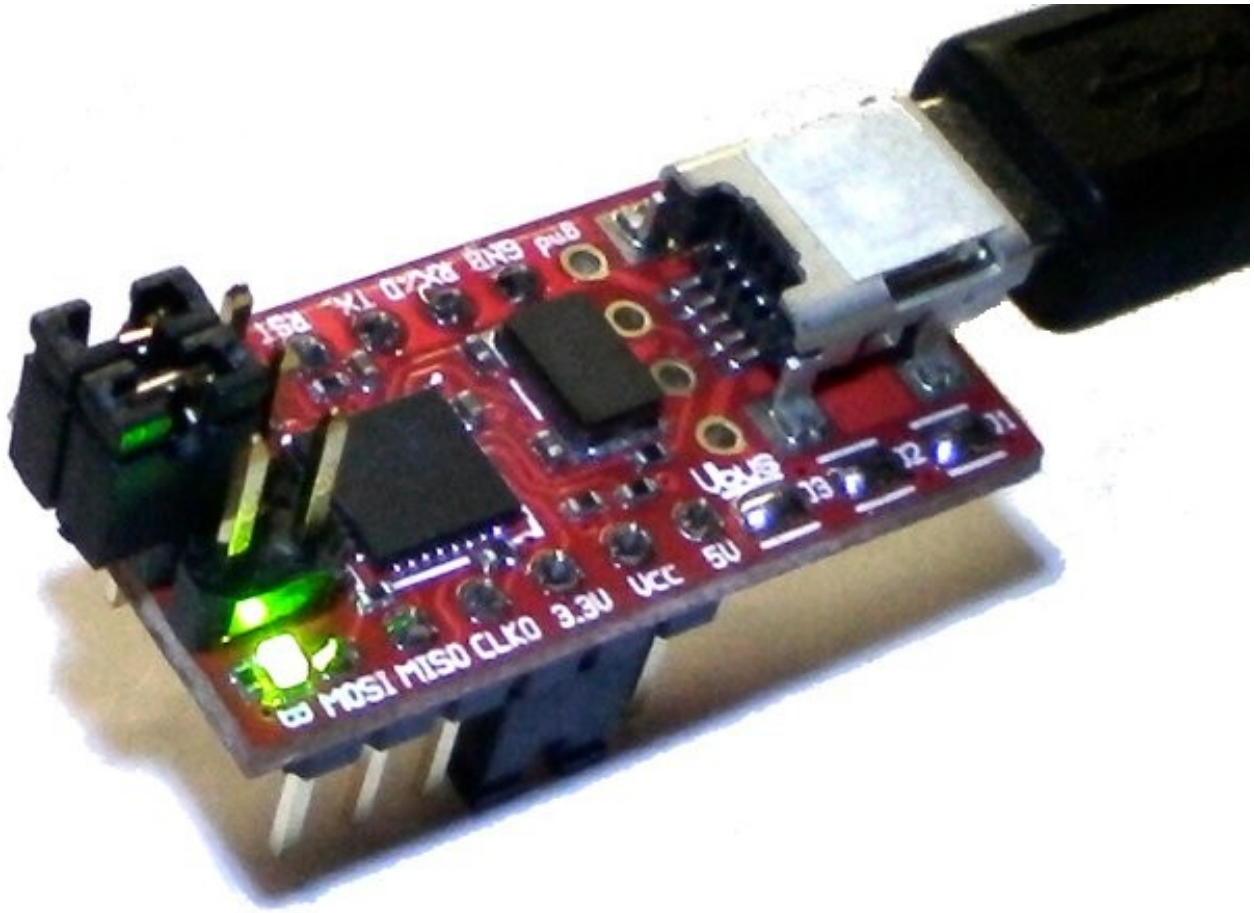
MT-UX Manual



MattairTech

Table of Contents

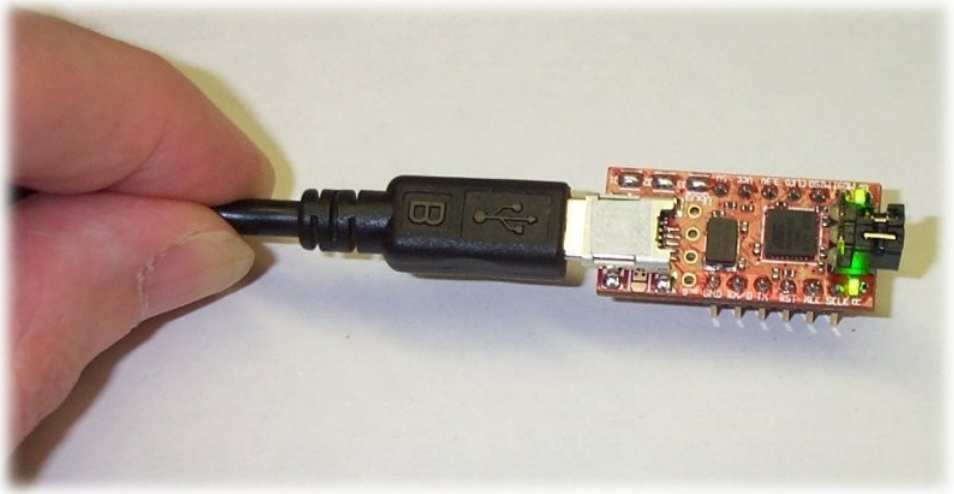
Overview

Introduction

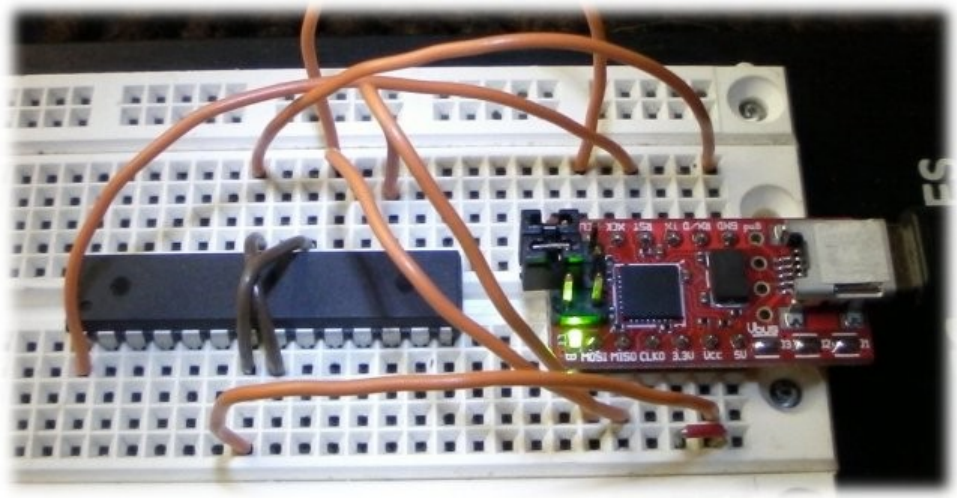
The MT-UX is a compact AVR programmer and USB to serial bridge that can be easily integrated into your project. It can be mounted on a breadboard, perfboard, stripboard (Veroboard), or a custom PCB. If using an enclosure, there are headers for external panel-mount buttons, as well as provisions for a panel-mount USB connector. The AVRISP mkII compatible programmer supports the ISP (Mega, Tiny), PDI (XMEGA), and TPI (6-pin Tiny) protocols. AVR Studio 4.x, 5.x, Atmel Studio 6.x, and AVRDUDE can be used to program your AVR over USB. The serial bridge connects your project at up to 2Mbps to a computer using the CDC class USB device mode (virtual COM port), for communications with a terminal emulator or any program that can use a serial port. The bridge supports both asynchronous or synchronous modes and 5 to 9 bit data widths. The MT-UX can supply your project with either 5V from USB Vbus or 3.3V from the onboard 500mA 3.3V regulator. Additional power supply options are available using three solder jumpers. An 8MHz clock output is provided and low-power operation is supported with two sleep modes.

Features

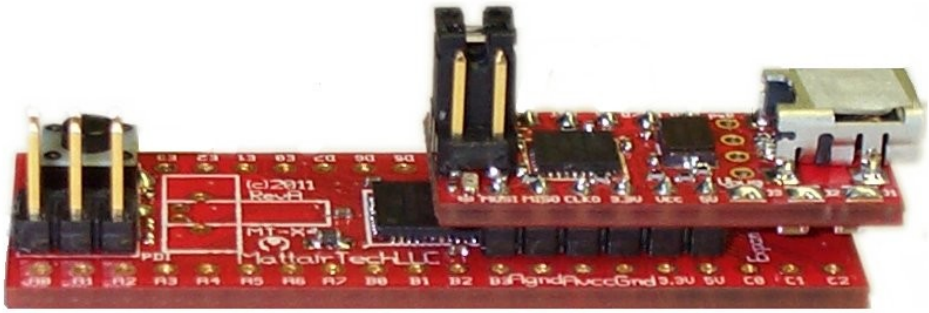
- Easily add USB communications and programming to your project
- Onboard AVRISP mkII compatible programmer
  - supports the ISP (Mega, Tiny), PDI (XMEGA), and TPI (6-pin Tiny) protocols
  - Program flash, EEPROM, fuses, lock bits, and more
  - Works with AVR Studio 4 and 5, Atmel Studio 6, AVRDUDE, Codevision, and BASCOM
- USB to Serial Bridge
  - Up to 2MHz baud rate (1MHz async)
  - Synchronous or asynchronous operation
  - 5 to 9 bit data width
  - Optional USB ready signal
- Powered via mini USB connector or external header
- Onboard 5V from USB Vbus
- Onboard 3.3V, 500mA LDO regulator
- 8MHz clock output (can be disabled during sleep)
- Sleep modes for low-power consumption
- 4 boot modes selectable via 2 jumpers (or external panel-mount buttons)
  - serial bridge (default), AVR programmer, configuration, or DFU bootloader
- Target board reset button can be used to toggle between AVR programmer and serial bridge
- Upgradeable firmware
- Can be mounted on breadboard, perfboard, stripboard, or custom PCB
- Very compact size, measures 3.05cm x 1.4cm (standard 0.1” pin spacing)
- Compatible with Windows XP/Vista/7/8 and Linux (limited Mac support)
- Uses LUFA USB library and AVRISPmkII by Dean Camera (**http://www.lufa-lib.org/**)



The MT-UX is very compact



The MT-UX mounted on a breadboard



The MT-UX mounted on a custom PCB

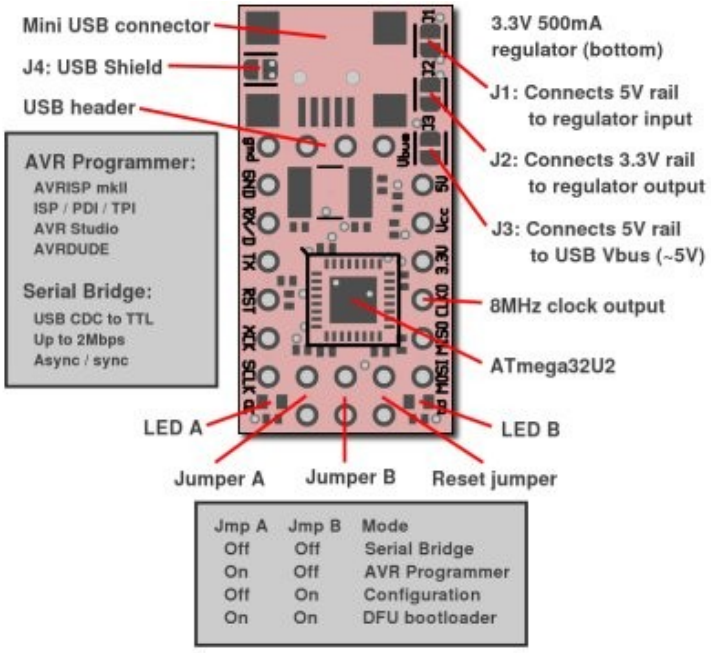
QuickStart

Installation

Before using the MT-UX, you must install at least the AVRISP mkII driver and the MattairTech CDC (virtual COM port) driver. A third driver is available for firmware updates (see Firmware Updates section). If using Atmel Studio 7, the AVRISPmkII driver must now be downloaded separately (see below). Extract the archive to any directory, then, with Vcc connected to 5V or 3.3V (use a jumper if not mounted to a board), plug in the MT-UX with only the A jumper installed. Windows will prompt for drivers, so direct the installer to the new directory. Prior versions of Atmel Studio bundled the AVRISP mkII driver. In these cases, point the installer to "Program Files/Atmel/AVR Jungo USB" and choose the 32 or 64 bit directory. Once installed, unplug the MT-UX, remove all jumpers, then plug it back in. Point the installer to the directory where you downloaded the CDC driver. Remember to rename the file from .txt to .inf if the installer does not see it.

Software	Version	Driver	URL
AVRISPmkII Driver	latest	AVRISPmkII driver	https://www.mattairtech.com/software/MattairTech_AVRISPmkII_Driver_Signed.zip
MT-UX Driver	latest	CDC driver	https://www.mattairtech.com/software/MattairTech_CDC_Driver_Signed.zip
AVR Studio / Atmel Studio	4.19, 5.x, 6.x, 7.x	Old AVRISPmkII	http://www.atmel.com/tools/atmelstudio.aspx OR http://www.atmel.com/tools/studioarchive.aspx (for AVR Studio)

Operation



With the default power jumper configuration (J1-J3 soldered), both 5V and 3.3V power pins are active. The operating voltage is selected by connecting one of these two sources to the Vcc pin, which can be done on a breadboard or perfboard. This voltage can also be output to the target. For ISP programming, MISO, MOSI, SCLK, RST and RST must be connected. For PDI and TPI, RX/D connects to PDI/TPI data, RST connects to the target PDI clock, and XCK to the TPI clock. Additionally, RST must be connected to TPI reset in TPI mode. An optional 8MHz clock can be output to the target. The serial bridge is a USB to TTL serial converter that can be used to connect the target board to a computer over USB, where it will show up as a virtual COM port. It can run at up to 2Mbps, and has support for asynchronous and synchronous modes, and 5-9 data bits. Connect RX to the target TX and TX to the target RX. Ground must also be connected. Be sure that the baud rate matches that of the target.

Windows Installation

Before plugging in the MT-UX for the first time, the latest software and drivers must be downloaded. The MT-UX is supported under Windows XP, Vista (32 and 64 bit), and Windows 7 (32 and 64 bit). There is limited support for Windows 2000. The MT-UX appears as three different devices to the PC depending on which mode is selected by the jumpers. These devices are the AVRISP mkII compatible programmer, the DFU bootloader for firmware updates, and the USB CDC device (virtual COM port) which is used for all other modes. Therefore, three drivers are required. The DFU driver is included with software available on the Atmel website. The CDC driver is included with Windows, but requires an .inf file available on the MattairTech website. The following table lists the minimum versions of the required software. If the software provides a driver, is is listed as well. See the Firmware Updates section for installation of the DFU bootloader driver.

Required Downloads

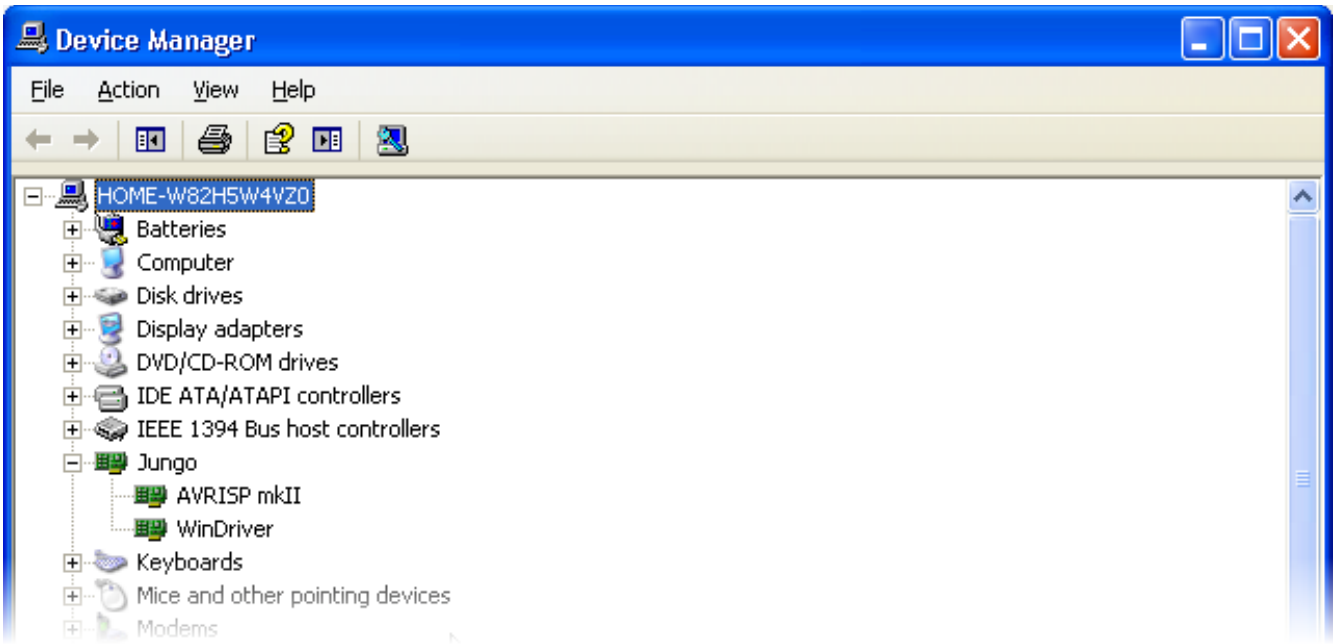
Software	Version	Driver	URL
AVRISPmkII Driver	latest	AVRISPmkII driver	https://www.mattairtech.com/software/MattairTech_AVRISPmkII_Driver_Signed.zip
MT-UX Driver	latest	CDC driver	https://www.mattairtech.com/software/MattairTech_CDC_Driver_Signed.zip
AVR Studio / Atmel Studio	4.19, 5.x, 6.x, 7.x	AVRISPmkII	http://www.atmel.com/tools/atmelstudio.aspx OR http://www.atmel.com/tools/studioarchive.aspx (for AVR Studio)

Atmel Studio (AVR Studio) / AVRISPmkII driver

Atmel Studio is a free IDE provided by Atmel that runs on Windows operating systems. It includes an assembler, debugger, simulator, and an AVR chip programming utility. As of April 2016, there are four main versions supported, AVR Studio 4.x and 5.x, and Atmel Studio 6.x and 7.x. The 4.x series is mature and stable, and can run on older hardware, however, it requires the use of the WinAVR gcc toolchain, which is out of date. It also lacks proper support for newer devices, like the XMEGA microcontrollers, but is still a good option for older devices. AVR Studio 4.x is also smaller and less demanding on PC resources. If you choose to use the 4.x series, download version 4.19. You will also need to download and install WinAVR 20100110 prior to installation.

If installing Atmel Studio 7, the AVRISPmkII driver must now be downloaded separately (see above). Extract the archive to any directory, then, with Vcc connected to 5V or 3.3V (use a jumper if not mounted to a board), plug in the MT-UX with only the A jumper installed. This will run the AVRISPmkII compatible AVR programmer. LED A should be lit and LED B should be pulsing on and off. Windows will prompt for drivers, so direct the installer to the new directory. Prior versions of Atmel Studio bundled the AVRISP mkII driver. In these cases, point the installer to “Program

Files/Atmel/AVR Jungo USB" and select the appropriate directory (usb32 or usb64). Do not use the driver in the AVR Tools/usb directory.



WinAVR / AVRDUDE / BASCOM

WinAVR contains the GNU GCC compiler for C and C++, compiler tools, and libraries (including AVR Libc). It also includes AVRDUDE for Windows, which is a command line tool for transferring firmware to AVR microcontrollers. A graphical tool is included with AVR Studio. Download WinAVR from http://sourceforge.net/projects/winavr/files/WinAVR/20100110/ and install it first. To use AVRDUDE, you will need to download and install an update to libusb-win32 available at http://sourceforge.net/projects/libusb-win32/files/libusb-win32-releases/. Choose the latest version, download, and extract. You will also need to change the MT-UX AVRISP mkII Programmer host configuration to AVRDUDE (configuration mode). Then, with the board in programming mode, run the install-filter-win.exe program included with libusb, which will allow you to install the filter driver. Note that WinAVR is outdated. It is not recommended for newer devices like the XMEGA series. AVRDUDE can also be installed separately.

BASCOM is supported with the programmer in AVRDUDE mode. Thus, the required setup shown above applies (installation of WinAVR is optional). Details of this process are covered on this BASCOM AVRISP MKII support page: http://avrhelp.mcselec.com/libusb.htm. They use AVR Studio 4, but you may install Atmel Studio 5 or higher instead. Follow scenario 1.

MT-UX Driver / Serial Configuration

Next, the MT-UX CDC driver can be installed, which is used by the serial bridge and configuration mode. This driver allows the board to appear as a COM port. The driver itself is included with Windows, but an .inf file is needed to configure it. Download the .inf file from https://www.mattairtech.com/software/MattairTech_CDC_Driver_Signed.zip. Note that Windows Vista 64-bit, Windows 7 64-bit and Windows 8 require the signed driver. Now, plug in the MT-UX with no jumpers installed. This will run configuration mode. Only LED A will be lit. Windows will then prompt you for the MT-UX CDC driver. Point the installer to the directory where you downloaded the driver and install. Note that you may need to rename the driver in order for it to show up in the installer. Windows may add the .txt extension to the file after downloading. Rename it so that it ends with .inf. Ignore any warnings given by the installer (ie: unsigned driver). Once the driver is loaded, the device will appear as the MT-UX CDC device using a COM port in the device manager. There is no need to configure serial port parameters. The buad rate, for example, is ignored. The MT-UX will always communicate with the computer at full speed (up to 2Mbps). If you experience any buffering problems, for example, a delayed response to user input, then change both buffer sizes to 1.

Terminal Emulator

Finally, the terminal emulator can be configured. Windows XP includes HyperTerminal, which has been tested with the MT-UX and will be documented here. There are several other terminal emulators available freely on the Internet. If you wish to use any of them, it should be no trouble to adapt the instructions presented here.

Next, start HyperTerminal. Create a new connection. You will refer to this connection again, so give it an appropriate name (after it is configured, you can copy it to your desktop). Select the MT-UX COM port (ie: COM4) and continue. It is not necessary to configure the baud rate or any other serial parameters. Now, click on the connect icon.

After connecting, you may see garbage on the terminal screen. If this is the case, click on the configuration icon and change the emulation to ANSI (or ANSIW). The configuration mode requires an ANSI terminal to allow drawing of the menu system. Normally, when first entering a mode that uses the CDC driver, a message that reads “Press any Key” is printed periodically. If you do not see this message, just press any key to continue.

It is important to always click the disconnect icon before switching to the AVR Programmer. Then click the connect icon a couple seconds after returning. This is required because changing to the AVRISPmkII driver unloads the CDC driver, then loads the AVRISPmkII driver. In order for the terminal to use the same COM port as before, it must be disconnected when returning to the CDC driver so that it does not assign a new COM port.

Linux Installation

Linux is supported as well. You must download and build the toolchain from the latest script available at AVR Freaks on the AVR GCC Forum (Script for building AVR GCC sticky at http://www.avrfreaks.net/index.php?name=PNphpBB2&file=viewtopic&t=42631). All firmware written for the MT-UX is developed under Linux using this toolchain.

Drivers

TODO (drivers should already be installed)

GCC Toolchain

TODO (see opening paragraph)

AVRDUDE

TODO (ie: avrdude -p x128a1 -c avrisp2 -P usb -U flash:w:"myfirmware.hex")

dfu-programmer

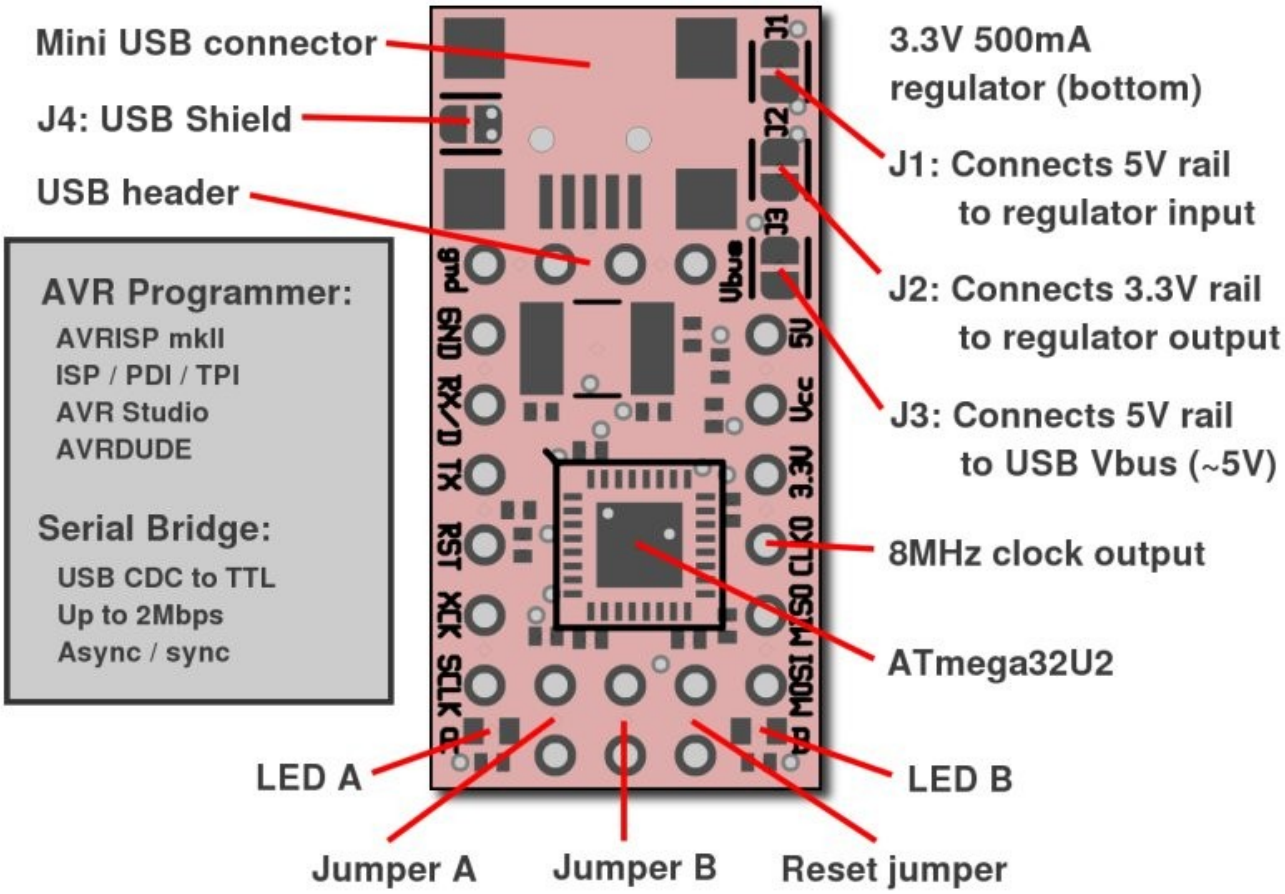
TODO (must use version 0.5.2 (currently available via SVN only) or higher)

Terminal Emulator

TODO (can use minicom, config port (ie: /dev/tty/ACM0), save config, run with minicom -o)

MT-UX Hardware

Layout / Pins



Jmp A	Jmp B	Mode
Off	Off	Serial Bridge
On	Off	AVR Programmer
Off	On	Configuration
On	On	DFU bootloader

Pin Descriptions

Pin	Description
5V	By default, this pin outputs 5V from USB Vbus. This pin is connected to the input of the onboard 3.3V regulator (located on the back of the PCB). 5V can be supplied externally on this pin if Jumper 3 on the USB module is desoldered, which will disconnect USB Vbus from the 5V pin.
Vcc	This pin is the power supply input for the MT-UX. It connects to the ATmega32U2 Vcc and AVcc pins. Power can be supplied to this pin by connecting 5V from USB Vbus, 3.3V from the onboard regulator, or an external source.
3.3V	By default, this pin outputs 3.3V from the onboard regulator (located on the back of the PCB). It can be connected to Vcc when running the MT-UX at 3.3V. Note that this pin is always connected to the UCAP pin of the ATmega32U2. This is because 3.3V is required for the USB data pins (D+ and D-) regardless of the operating voltage of the board (Vcc). If supplying 3.3V externally, disconnect the output of the regulator by desoldering J2.
CLKO	This pin outputs an 8MHz clock signal. The clock can be configured to either stop or continue running while in sleep mode. If powerdown is selected for sleep mode, the clock will stop during sleep.
MISO	This pin is the MISO input during ISP programming, otherwise it is tri-stated. A 20K-50K pullup is activated during ISP programming and during sleep mode.
MOSI	This pin is the MOSI output during ISP programming, otherwise it is tri-stated. A 20K-50K pullup is activated during sleep mode.
SCLK	This pin is the SCLK output during ISP programming, otherwise it is tri-stated. A 20K-50K pullup is activated during sleep mode.
XCK	This pin is the TPI clock output. It is also used as the XCK output by the USB-UART bridge when in synchronous mode. Additionally,if configured, this line can signal when USB is ready by pulling the line low. When USB is not ready or disconnected, the line will tristate. The target chip must enable the pullup on this line if using the USB ready signal.
RST	This pin is the reset output. It is used with all programming modes. During PDI programming, the line is connected to XCK by the bilateral switch located on the back of the PCB. This is due to the fact that the reset and PDI clock share the same pin on XMEGA devices. If a reset button is present on the target board, pressing reset will toggle the MT-UX between the AVR programmer and serial bridge, thus eliminating the need to change jumpers and cycle power when changing modes. This pin is open-drain, and a 20K-50K pullup is active on this line at all time when not driving low.
TX	This pin is the TX output of the serial bridge. During PDI and TPI programming, it is connected to RX by the bilateral switch located on the back of the PCB. It is tri-stated when unused. Thus, the target board should provide a pullup if serial is used. A 20K-50K pullup is activated during sleep.
RX/D	This pin is the RX input of the serial bridge. A 20K-50K pullup is activated when in serial bridge mode. It is also used during PDI and TPI programming as the data pin, when it is connected to TX by the bilateral switch located on the back of the PCB (in this case it is bi-directional).
GND	This pin is the MT-UX ground. It must always be connected in all modes of operation.
Vbus	This pin is the USB 5V from the optional USB header.
D-	This pin is the USB D- signal from the optional USB header.
D+	This pin is the USB D+ signal from the optional USB header.
GND	This pin is the ground from the optional USB header.

Jumpers

There are three jumpers on the MT-UX. Two are used to select one of four modes of operation, and one can be used to reset the MT-UX. Since the MT-UX is designed to be incorporated into a project requiring USB communications and/or programming, jumpers are used, rather than buttons. This saves board space and allows the connection of external buttons (for example, panel mount buttons). Jumper A is next to LED A , Jumper B is in the center, and RESET is next to LED B. The mode is selected when powering the board. Additionally, the mode can be toggled between the AVRISP mkII programmer and serial bridge by pressing the RESET button on the target board. The reset jumper is only connected to the MT-UX reset input. There is a 10K pullup on this reset input. A jumper cap can be used to reset the MT-UX, but it is mainly intended for an external button. The following table lists the jumper functionality.

Jumper Functionality

Jumper A	Jumper B	Mode
Installed	Not Installed	AVRISP mkII Programmer
Not Installed	Installed	Configuration
Not Installed	Not Installed	Serial Bridge
Installed	Installed	DFU Bootloader

LEDs

There are two green LEDs that are used to indicate the mode of operation, communication activity, and programmer status. The following table lists LED functionality in each mode. When an LED is used to display communication activity, the default state of the LED is shown on the left. For example, during serial bridge RX activity, the LED blinks off for a short time then returns to the default on state.

LED Functionality

Mode	LED A	LED B
AVR Programmer	On / Programmer Activity	PWM pulsing
Configuration	On	Off
Serial Bridge	On / RX Activity	On / TX Activity
DFU Bootloader	Off	On

Clock Output

The CLKO pin outputs an 8MHz clock signal. The clock can be configured to either stop or continue running while in sleep mode. If powerdown is selected for sleep mode, the clock will stop during sleep. This setting is useful for battery powered applications when the lowest power consumption is needed. If standby is selected, the clock will continue running during sleep.

Sleep Mode

The MT-UX will enter sleep mode when the USB host suspends the MT-UX, or when USB is disconnected. If using USB power-only cable (no USB signals), then the MT-UX will enter sleep. The mode of sleep, powerdown or standby, can be set in the configuration. Powerdown will reduce power consumption to a minimum, by stopping the crystal oscillator and the CLKO output. In both sleep modes, the signal pins (MISO, MOSI, SCLK, XCK, TX, RX/D, and RST) will have 20K-50Kohm pullups enabled. Additionally, both LEDs will be turned off.

USB Ready Signal

The USB ready signal, if enabled, is useful when the target needs to know when the USB cable is disconnected or the USB bus suspended. The signal is open-drain active-low from the XCK pin, which may also be used for synchronous serial operation. The target must enable the pullup on this line before reading it. If it reads low, USB is enumerated and ready. When in sleep mode, or when USB is connected but not yet enumerated, it will read high. If synchronous operation in used, the XCK clock signal will override this.

Target Board Reset

When the MT-UX is reset via the reset jumper, the target board will also be reset via the RST pin by bringing the line low for approximately 1 millisecond. The line is otherwise pulled high via a 20K-50Kohm pullup resistor. The RST line is also monitored by the MT-UX for an external reset, for example, by a reset button on the target board. If this line is pulled low externally and then released, the MT-UX will toggle between the AVR programmer and the serial bridge. This is useful for debugging, when it would be inconvenient to change jumpers and cycle power to change modes.

USB Shield

Jumper J4 on the USB module can be soldered to connect the USB shield to ground. The USB specification calls for the USB shield to be connected to ground on the host side only. However, it may be desired to ground this on the device side. An 0603 SMT component may be soldered on the solder jumper pads as well. Soldering this jumper will also help dissipate heat from the onboard 3.3V regulator through the USB connector.

USB Header

An optional 4-pin header can be installed to allow connection of an external panel-mount USB connector. The pinout from left to right: GND, D+, D-, Vbus.

Power Configuration

WARNING
Care must be taken when configuring the solder jumpers. It is possible to cause permanent damage to the device or the target board by improperly setting the jumpers, or by supplying an incorrect external voltage (if used). Do not change any solder jumpers while the unit is powered.

The MT-X4-UX can be powered in a variety of ways by utilizing 3 solder jumpers. By default, the board is configured to be powered via USB. At all times, 3.0V-3.6V must be supplied on the 3.3V pin. By default, this is provided by the onboard 3.3V regulator on the back of the PCB. Additionally, the Vcc pin must be supplied with a voltage in the range listed in the table below. Vcc is a power input connected to the ATmega32U2 Vcc and AVcc pins. It sets the operating voltage of the MT-UX and thus the voltage used for programming and the serial bridge. It must match the target board voltage. Connection of Vcc can be made on a breadboard, perfboard, stripboard (Veroboard), or a custom PCB. Alternatively, a jumper cap can be used if connecting the MT-UX to the target board via jumper wires. See schematic and pinout table for details. The following table lists some of the configurations.

Power Configuration	Jumper J1	Jumper J2	Jumper J3	Regulator
USB bus powered (default)	**Soldered**	**Soldered**	**Soldered**	Used
Externally powered – 4.0 to 5.5V	**Soldered**	**Soldered**	**Not Soldered**	Used
Externally powered – 3.0 to 3.6V	**Not Soldered**	**Not Soldered**	**Doesn't Matter**	Not Used

USB Bus Powered

This is the default configuration. Vbus is connected to the 5V rail (J3). The regulator input is connected to the 5V rail (J1) and the regulator output is connected to the 3.3V rail (J2). The MT-UX operating voltage comes from the Vcc input pin. You may connect either 5V or 3.3V to the Vcc pin.

Externally Powered – 4.0V to 5.5V

A voltage from 4.0V to 5.5V can be supplied externally to the 5V pin. In this case, desolder J3 to disconnect USB Vbus from the 5V rail. This external voltage can optionally be supplied to the onboard 3.3V regulator input through J1, as well as to the Vcc power input.

Externally Powered – 3.0V to 3.6V

A voltage from 3.0V to 3.6V can be supplied externally to the 3.3V pin. In this case, desolder both J1 and J2 to disconnect the onboard 3.3V regulator from the 5V and 3.3V rails. This external voltage can optionally be supplied to the Vcc power input as well.

AVRISP mkII Compatible Programmer

The MT-UX AVR Programmer is based on the AVRISP mkII compatible programmer written by Dean Camera (http://www.fourwalledcubicle.com/). It supports programming of all Atmel AVR microcontrollers with an ISP, PDI, or TPI programming interface. These include the megaAVR series (ISP), the tinyAVR series (ISP, TPI), the XMEGA series (PDI), the USB AVRs (ISP), and the listed CAN and PWM AVRs (see Appendix B for device listing). AVR Studio 4.19 and 5.x, Atmel Studio 6.x and 7.x, and AVRDUDE are supported. See hardware section for details on the pinouts.

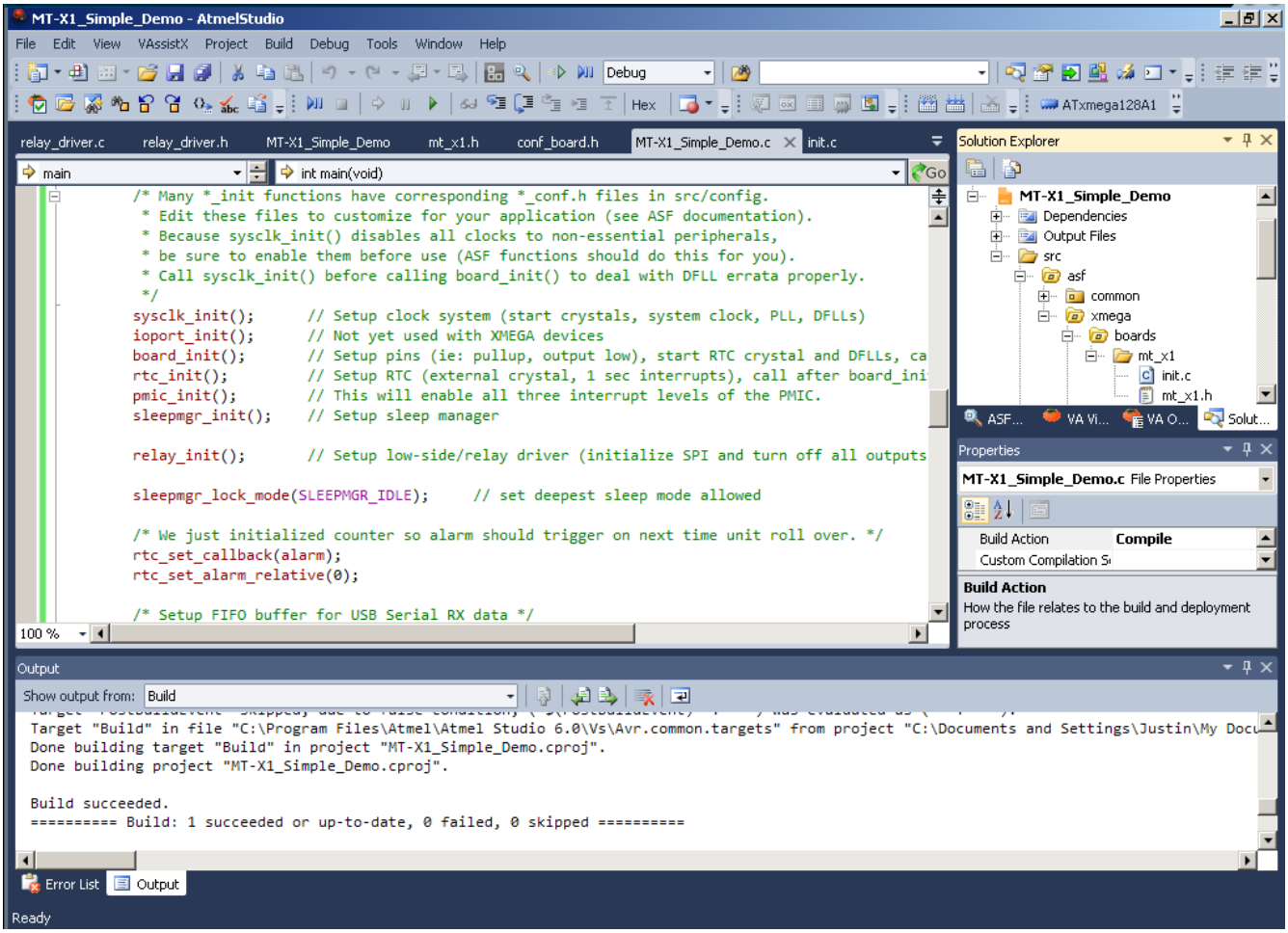
Programming speeds of up to 8MHz are supported in ISP mode. However, current AVRs require a programming speed less than ¼ of the target clock speed. For 20MHz AVRs, this is 4MHz. For 16MHz, 2MHz is the limit. It is not recommended to operate at exactly ¼ of the target frequency, especially when programming fuses, as this can cause them to become incorrectly set and possibly render the AVR useless (unless parallel programming is available). Note that many AVRs come from the factory with the clock source set to the internal 8MHz oscillator and with the CKDIV8 fuse programmed, resulting in a clock speed of 1MHz. In these cases, the ISP programming speed should be set to 125KHz or less until CKDIV8 is unprogrammed and power cycled.

For all modes, the 8MHz clock output can be connected to the target clock input and used as a recovery clock. This is useful, for example, to allow resetting of fuses that were mis-configured to use an external clock when intending to use a crystal or internal oscillator.

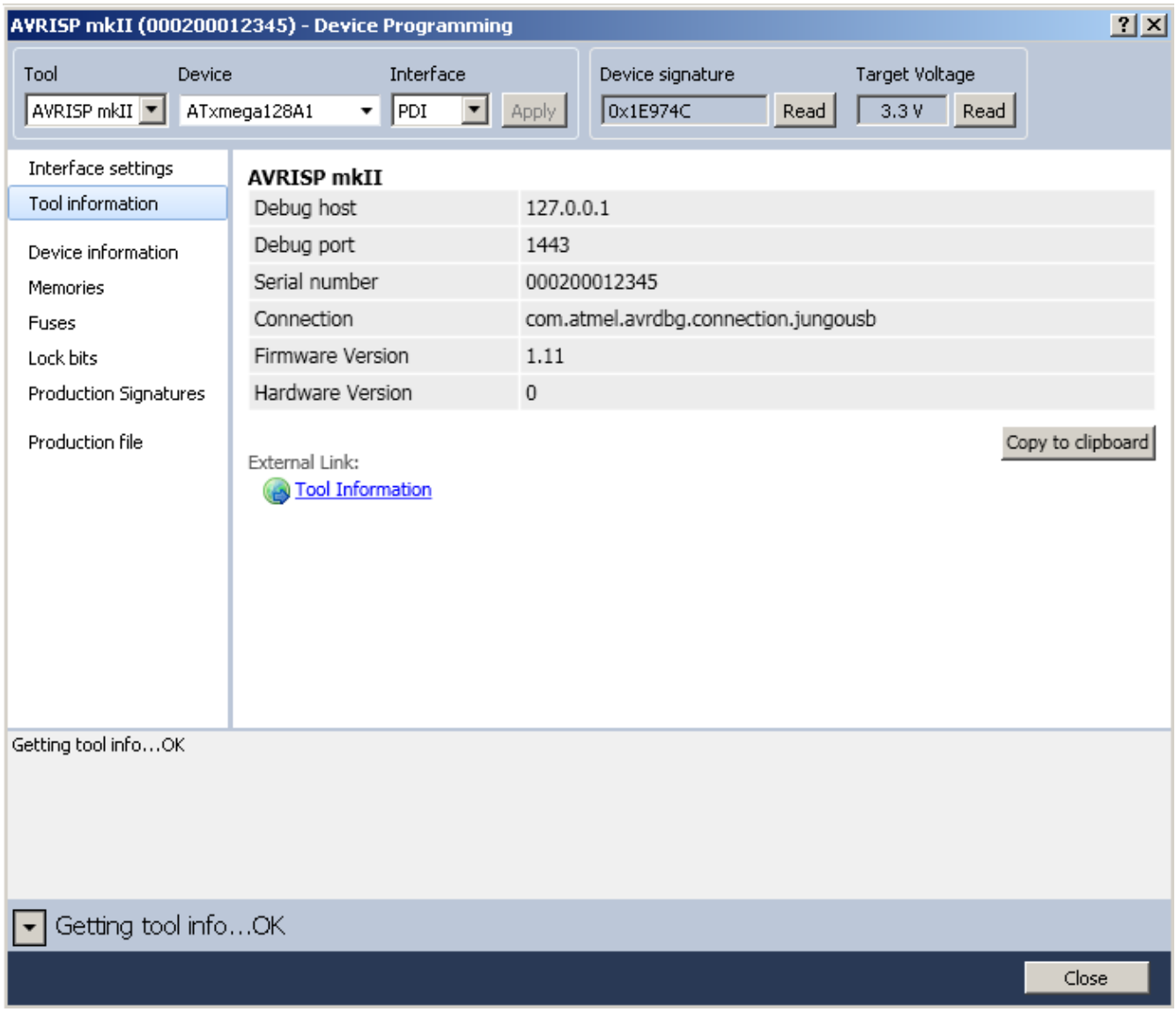
The MT-UX supports target devices operating at 3.0V to 5.5V. The voltage of the programming outputs from the MT-UX is determined by the voltage present on the Vcc power input pin. This voltage is usually provided by the MT-UX, which can supply either 5.0V or 3.3V. Alternatively, the target board can supply this voltage. In either case, Vcc and ground must be connected to the target board. Consult the Hardware section for details. All outputs from the MT-UX have 300ohm series resistors that limit current and control overshoot and ringing.

Using Atmel Studio (AVR Studio)

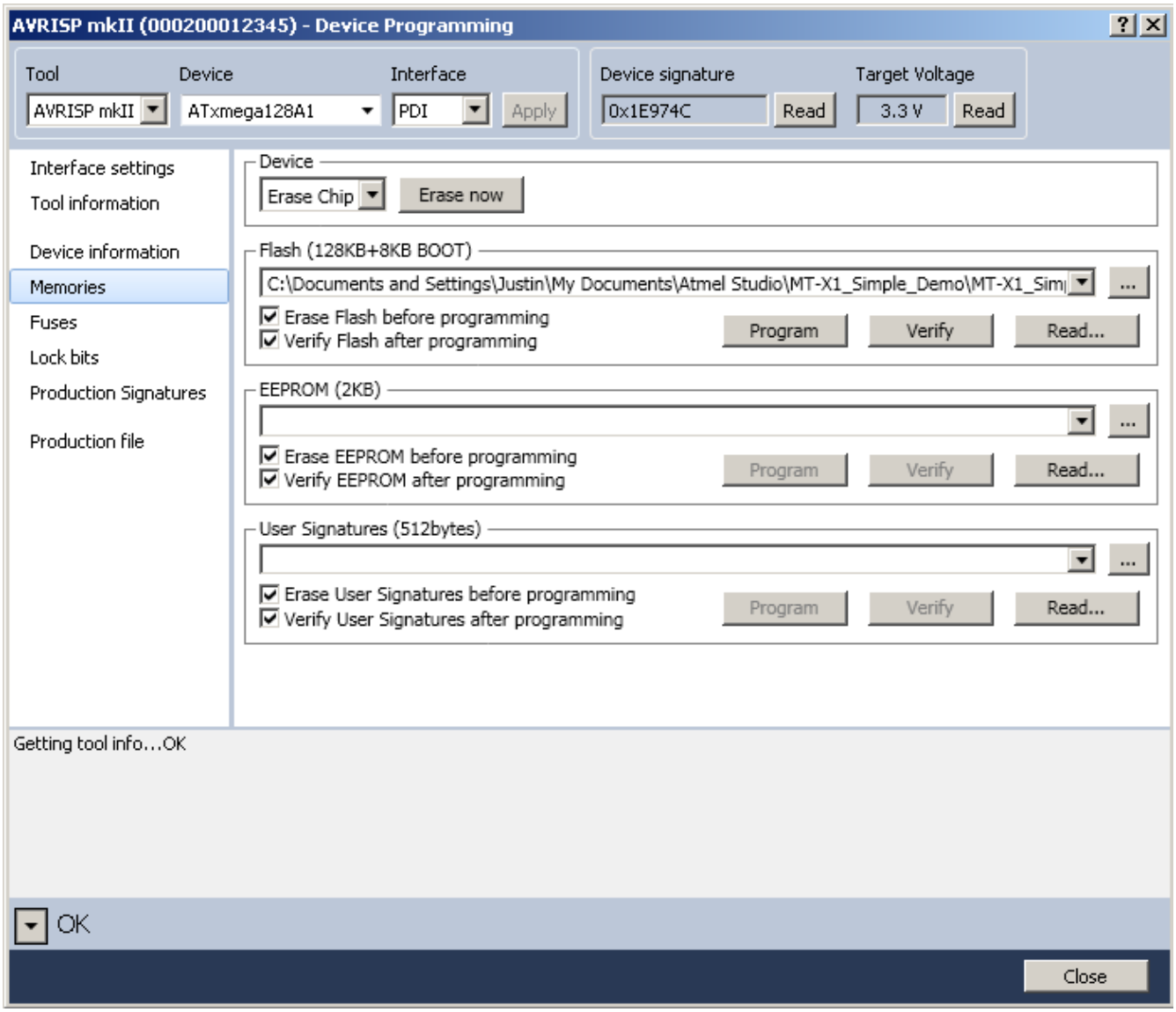
Start Atmel Studio and open or create a new project. The following screenshots from Atmel Studio 6 show the MT-X1S_Simple_Demo template for the MattairTech MT-X1S ATxmega128A1 development board.



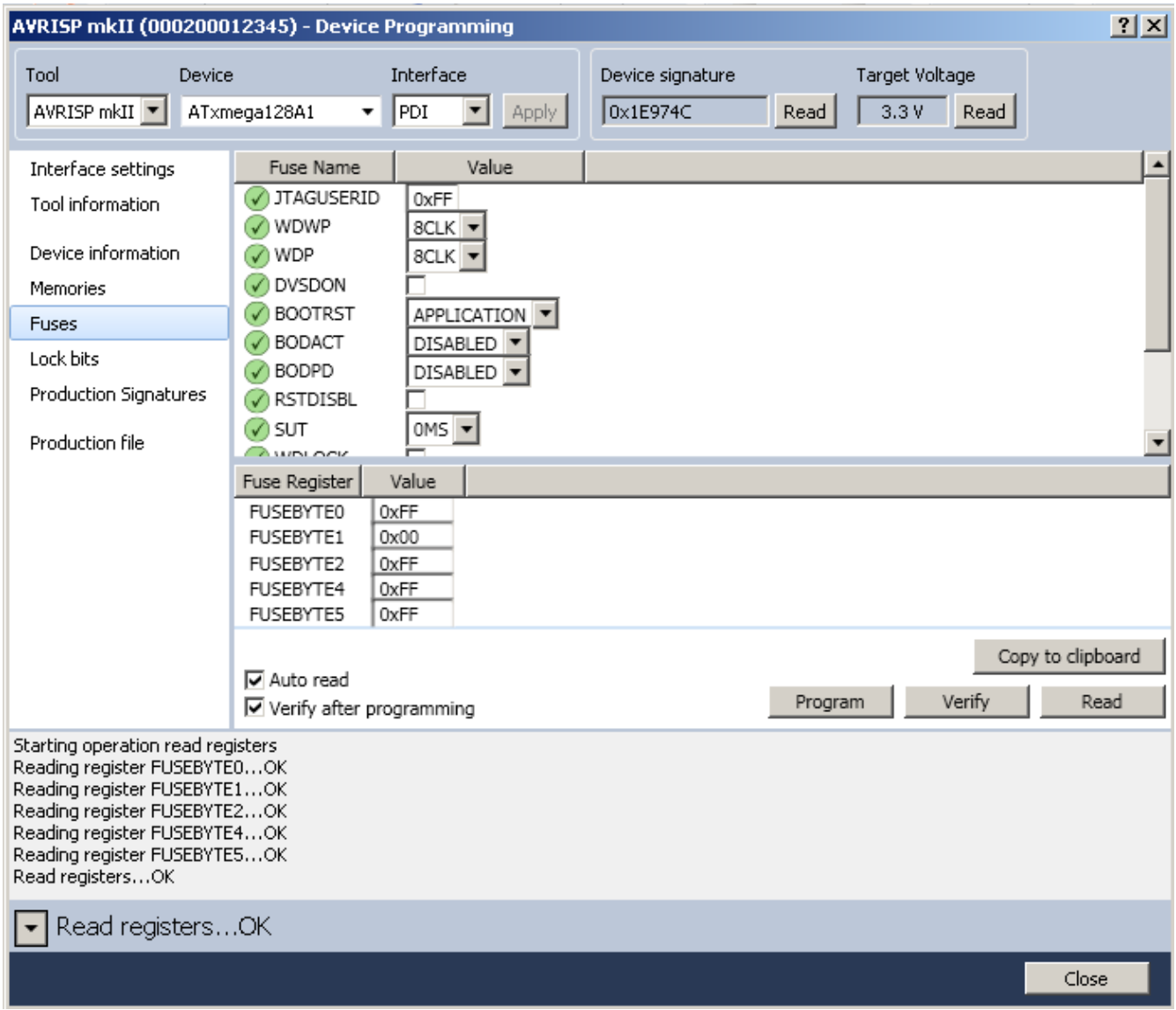
Next, click on the Device Programming button. In the Device Programming window, select the AVRISP mkII as the tool. If no tool appears, be sure that the MT-UX is plugged in and in programming mode (LED B will be pulsing). Select the appropriate device and either ISP, PDI, or TPI as the interface and click Apply. You should now be connected to the AVRISP mkII compatible programmer with serial number 000200012345. Now click Read next to Device signature. It should match the device if all is well. It is recommended to always perform this step first to verify the connection. The target voltage will always read 3.3V, regardless of the actual voltage.



Next, select the Memories page. In the Flash section, a hex file can programmed into the targets flash memory. Load your hex file, then click Program. The hex file is located in the Debug folder. You will need to erase the target first if you do not have “Erase Flash before programming” checked. You should also verify the flash as well.



Next, click on the Fuses tab. It is best to leave the fuse settings alone until you understand what they do. In particular, if using ISP, do not program RSTDISBL or unprogram SPIEN, as this will lock you out of the target chip. Do not set the BOD (Brown-out detection) voltage to a level above the target chip voltage, as this will cause the target to be held perpetually in reset. You must also be careful with the clock settings as well. If you select the wrong clock source, then your target chip will not operate if the configured clock source is not present. However, the MT-UX provides a clock output which can be used to recover from this situation (see above).



Now you may wish to look at the other pages. Note that any firmware upgrade feature should not be used. The MT-UX programmer is not an actual AVRISP mkII, it just emulates one, so you should not attempt to update the MT-UX firmware using Atmel Studio. Any firmware updates will be posted to the website and loaded using FLIP or dfu-programmer.

Using AVRDUDE

TODO (ie: avrdude -p x128a1 -c avrisp2 -P usb -U flash:w:"myfirmware.hex")

Serial Bridge

The serial bridge can connect the target board to a host application (ie: terminal emulator) over USB. On the host side, the MT-UX will appear as a virtual COM port. The MT-UX simply relays bytes between the host and target. Speeds of up to 2Mbps are supported.

Configuration

Before using the serial bridge, it must be configured to be compatible with the target. This configuration is stored in EEPROM. There is no need to duplicate the settings on the host side, as communication between the host and MT-UX will always be the maximum supported USB speed, and the other parameters are ignored by the host. Only the connection between the MT-UX and target use these settings. Note that when configuring the speed to be manual, it is possible to set the speed higher than 2MHz, but the maximum speed supported by the USB link is 2MHz. The serial bridge is configured in configuration mode (jumper A off, jumper B on).

Serial Bridge Configuration Options

Configuration Option	Possible Values
Speed	2M, 1M, 500K, 250K, 125K, 76.8K, 57.6K, 38.4K, 19.2K, 9600, 2400, manual
Baud Rate Register	0x0000 - 0x0FFF (if manual selected as speed)
Clock 2X	1X, 2X
Clock Mode	async, sync
Data Bits	5, 6, 7, 8, 9

Serial Connections

Connect the MT-UX TX pin to the target RX. Connect the MT-UX RX line to the target TX. Gnd and Vcc must also be connected to the target board. When in synchronous mode, the MT-UX is the master, so the XCK pin is enabled as an output. The target board must enable its clock pin as an input and be configured as a slave. When using 9-bit data frames, two bytes are sent or received for every frame. The first byte simply contains the 9th bit, thus the first byte will always be 0 or 1. The second byte contains the rest of the 8 bits.

Baud Rate Register Value (Manual Speed)

Async 1X	Async 2X	Synchronous
$UBRR = \frac{f_{osc}}{16 * BAUD} - 1$	$UBRR = \frac{f_{osc}}{8 * BAUD} - 1$	$UBRR = \frac{f_{osc}}{2 * BAUD} - 1$
$BAUD = \frac{f_{osc}}{16 * (UBRR + 1)}$	$BAUD = \frac{f_{osc}}{8 * (UBRR + 1)}$	$BAUD = \frac{f_{osc}}{2 * (UBRR + 1)}$

where $f_{osc} = 16000000$

Configuration

The MT-UX programmer, serial bridge, and other features can be configured by entering configuration mode. This configuration is stored in non-volatile EEPROM memory. Configuration mode requires an ANSI terminal emulator. Configuration options are highlighted by using the up and down arrow keys, and selected using the enter key. Some dialogs are for entering numbers in hexadecimal. Here, the left arrow key, right arrow key, and backspace can be used. The following lists the structure of the menu system:

- Config Serial
  - Auto Config (Set baud rate using host (like FTDI))
    - Disabled, Enabled
  - Serial Speed (Serial bridge speed selection)
    - List of selectable speeds: 2400, 9600, 19.2K, 38.4K, 57.6K, 76.8K, 125K, 250K, 500K, 1M, 2M
    - Manual (when selected, configure using Manual Settings below)
      - Baud Rate Register (enter value in hex)
      - Clock 2X (async mode only)
  - Clock Mode
    - Asynchronous or synchronous
  - Polarity
    - Sample Falling or Sample Rising
  - Data Bits
    - 5, 6, 7, 8, or 9
  - Stop Bits
    - 1 or 2
  - Parity
    - None, Even, or Odd
- Sleep Mode (Which sleep mode is used when USB is disconnected or suspended)
  - Power Down or Standby
- Ready/RTS Signal (Use USB state or RTS line to control XCK pin state, open-drain active low)
  - Disabled, USB Ready, RTS Normal, RTS Inverted
- AVRISPmkII (select which software will be interfacing with the MT-UX programmer)
  - AVR Studio or AVRDUDE
- LED Brightness (Set LED brightness using PWM (LEDs are especially bright at 5V))

    - Disabled, Min, Low, Med, High, or Max
- Arduino Auto Reset (Use serial bridge with bootloader for use with Arduino IDE (FTDI or Leonardo))
    - Disabled, Standard, or Leonardo
- Credits (displays list of firmware authors)

The AVRISP mkII programmer has two configuration options. The first is the selection of the host application, which can be either AVR Studio or AVRDUDE. This is required because these two modes use a slightly different USB endpoint configuration. If you are using Linux, then this setting will not matter, as they both work with AVRDUDE for Linux (AVR Studio is not available for Linux).

The USB AVR automatically enters sleep mode when the USB cable is disconnected or the USB bus is suspended. Sleep mode is by default set to Power Down, which provides for the lowest current consumption. Otherwise, the 8MHz CLKO output will be disabled during sleep.

The USB ready signal is useful when the target needs to know when the USB cable is disconnected or the USB bus suspended. The signal is open-drain active-low from the XCK pin, which may also be used for synchronous serial operation. The target must enable the pullup on this line before reading it. If it reads low, USB is enumerated and ready. Otherwise, it will read high. If synchronous operation in used, the XCK clock signal will override this.

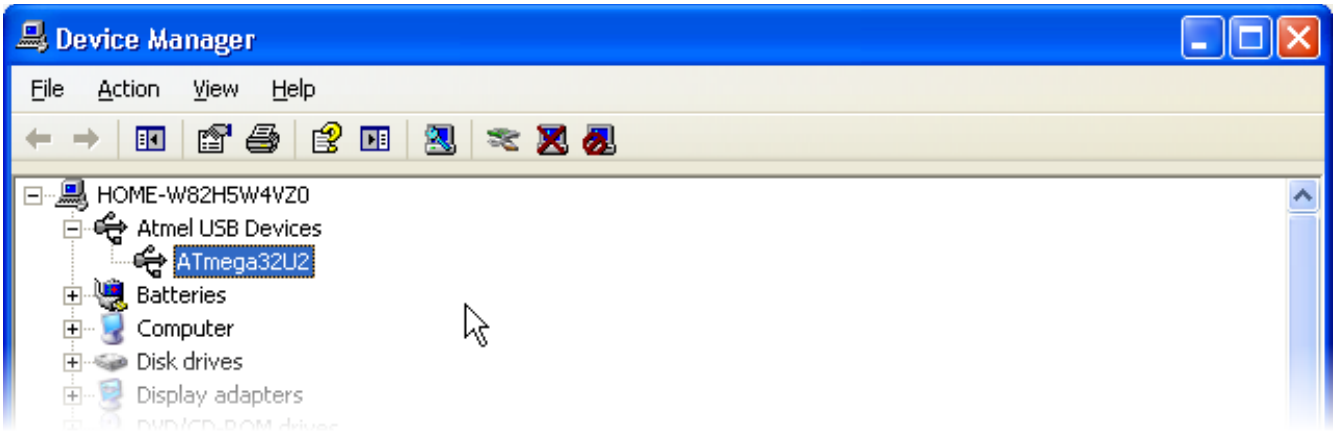
Firmware Updates

The MT-UX firmware will be updated periodically to add new features and fix bugs. These updates will be available on the MattairTech website. The updates may include just a hex file (for programming flash), or both a hex file and eep file (for programming both flash and EEPROM). FLIP is a graphical utility for Windows used to load firmware updates onto the MT-UX. FLIP includes the DFU bootloader driver. Download FLIP 3.4.2 or higher from http://www.atmel.com/tools/FLIP.aspx and install.

Downloads required for Firmware Updates

Software	Version	Driver	URL
MT-UX Firmware	latest	N/A	http://www.mattairtech.com/software/MT_UX/MT_UX.hex
FLIP	3.4.2 +	DFU driver	http://www.atmel.com/tools/FLIP.aspx
Signed DFU Driver*	latest	DFU driver	http://www.avrfreaks.net/index.php?module=Freaks%20Academy&func=viewItem&item_type=project&item_id=2196

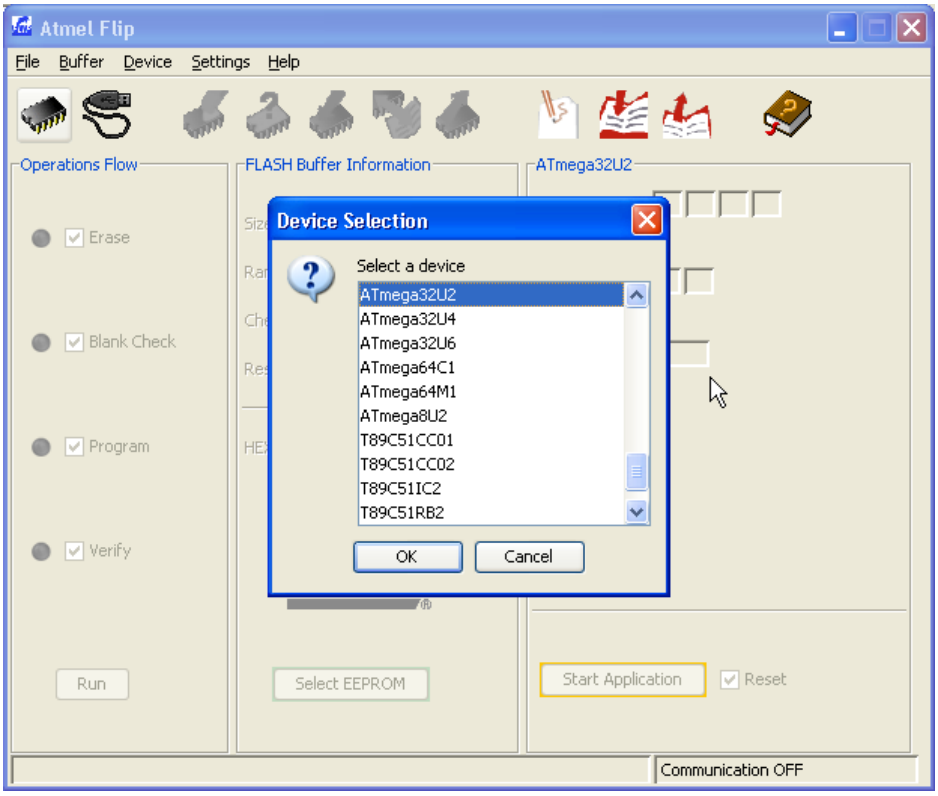
** Newer versions of FLIP include a signed driver. In this case, do not use the driver from avrfreaks.*



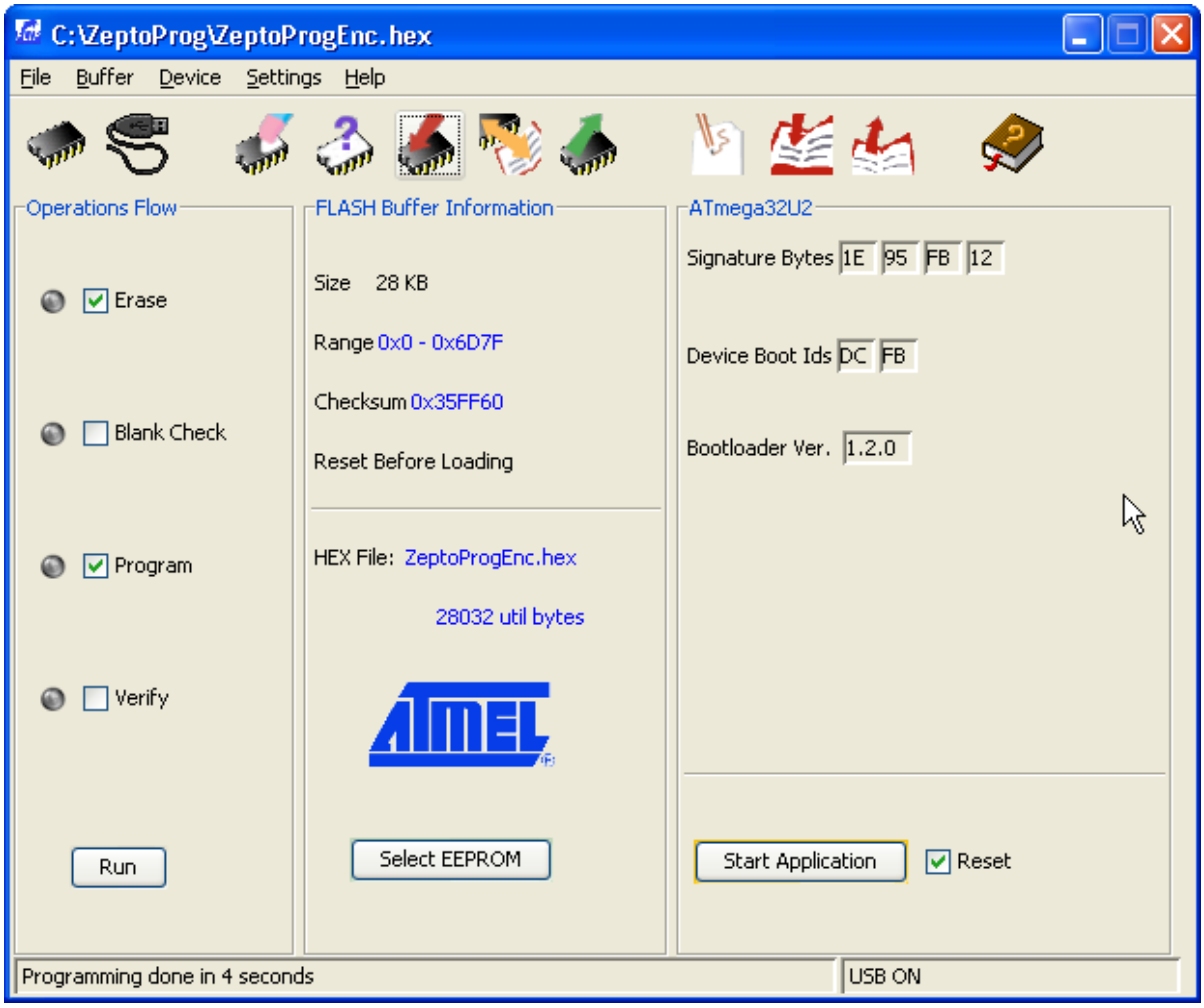
Once FLIP is installed, the DFU bootloader driver can be loaded. Press and hold both buttons while plugging in the MT-UX to run the DFU bootloader. LED A should be on and LED B should be off. Windows will then prompt you for the ATmega32U2 driver. By default, this is located in the Program Files/Atmel/Flip 3.4.2/usb directory. Point the installer to this location (do not use Windows update, then click 'Install from a list or specific location'). If Windows does not find the driver, try the “don't search/have disk → show incompatible hardware” method. If Windows does not prompt you for drivers, open device manager and look for Unknown Devices → ZeptoProg II Bootloader. Right-click this and select Update driver, hen continue as above. If using an older version of FLIP, you may need to download the signed driver from avrfreaks (see table above). Once the driver is loaded, device manager will show an ATmega32U2 device under Atmel USB Devices.

FLIP

Plug in the MT-UX with both A and B jumpers installed to run the DFU bootloader. LED A should be off and LED B should be on. Now launch the FLIP utility. When it has loaded, click on the chip icon and select the Atmega32U2.



Next, click on the USB icon, select USB, then connect. The screen should now show information about the ATmega32U2. Click on the File menu, and open the appropriate hex file. More information will appear about the program. Be sure that erase is checked. The MT-UX firmware cannot be loaded unless the flash is erased first. Uncheck Blank Check, as it is not supported. Program must be checked. Verify must be unchecked. Reading of the flash is not allowed, so verification is not possible. Verification is less useful when programming over USB anyway, as USB provides error detection and correction. Now click on the Run button in the lower-left of the screen, and the firmware will be quickly loaded onto the MT-UX. If you encounter problems, then you will need to unplug the MT-UX, disconnect FLIP, and start over making certain that the above settings are observed.



You may also need to program the EEPROM. If so, click on Select EEPROM at the bottom. Then, click on the File menu and open the appropriate eep file. You will have to change the file filter to allow you to see the eep file. Note that eep files are just hex files but with the eep extension instead of hex. More information will appear about the file when selected. Both Program and Verify should be checked. Click run to program the EEPROM.

dfu-programmer (Linux)

A chip erase must be performed first. The flash cannot be read.

dfu-programmer atmega32u2 erase

dfu-programmer atmega32u2 flash-eeprom MT_UX-120401.eep (if applicable)

dfu-programmer atmega32u2 flash --suppress-validation MT_UX-120401.hex

Troubleshooting / FAQ

1. **The board is plugged into USB, but both LEDs remain off.**
   Be sure that Vcc is connected to a power source, such as 5V or 3.3V. If the board is not mounted, then a jumper can be placed across Vcc and either 5V or 3.3V. Otherwise, the connection is provided by the breadboard (or perfboard, etc.).

2. **Atmel Studio (AVR Studio) reports 3.3V when the board in fact is operating at 5V.**
   Ignore this. The MT-UX cannot measure the voltage of the target (but it can detect the presence of a voltage on Vtgt as low as 2V). Voltage reporting is informational only, it does not affect programming.

3. **Updating the firmware with recent versions of FLIP (ie: 3.4.7) may fail.**
   If so, try using the batchisp command line tool included with FLIP.:C:\Program Files\Atmel\Flip 3.4.7\bin>batchisp -device ATMEGA32U2 -hardware USB -operation erase F loadbuffer "C:\MT_UX.hex" program start reset 0

4. **Communications are garbled or unreliable.**
   Be sure that ground is connected between the two boards.

5. If Atmel Studio wants you to upgrade the firmware, first check that you are not in AVRDUDE programming mode, which can trigger this warning. If you are in AVR/Atmel Studio mode and still get the message, please check the website for firmware updates or email support@mattairtech.com if you are already using the latest version.

6. If BASCOM or AVRDUDE does not work, be sure to put the programmer into AVRDUDE mode.

7. Prior to releasing the signed driver on January 28, 2015, Windows 8 users needed to disable driver signing to use Tool mode (terminal emulator or Java app.).

8. AVRDUDE 6.x does not yet support the ZeptoProg II. A working patched version can be found at http://www.mattairtech.com/software/avrdude_6.0.1_patched_windows.zip. Thanks to Larry Viesse. For support on Linux 64-bit, download http://www.mattairtech.com/software/avrdude_6.0.1_patched_Linux_64.zip.
   For support on other Linux (especially with xhci (USB 3.0)), replace the usb_libusb.c file from 6.0.1 with http://www.mattairtech.com/software/usb_libusb.c.

9. If you are having problems communicating with the programmer using Atmel Studio 6.x, download the Zadig USB driver manager at http://zadig.akeo.ie/.
   Under options, List All Devices. The AVRISP mkII should show up in the list. Replace the current driver with libusb-win32 (v1.2.6.0), which comes embedded with Zadig.

Alternatively, please use the procedure at https://www.olimex.com/forum/index.php?topic=4188.0

10. Prior to November 16, 2014, the 16MHz crystal (18pF) was -20C - 70C and the capacitors were 22pF. The new crystal (12pF) is -40C – 85C.

11. If you are having problems communicating with the programmer using Atmel Studio 7.x, please ensure that you are using the new AVRISPmkII driver, which now must be downloaded separately (see installation). Prior versions of Atmel Studio included this driver.

Support Information

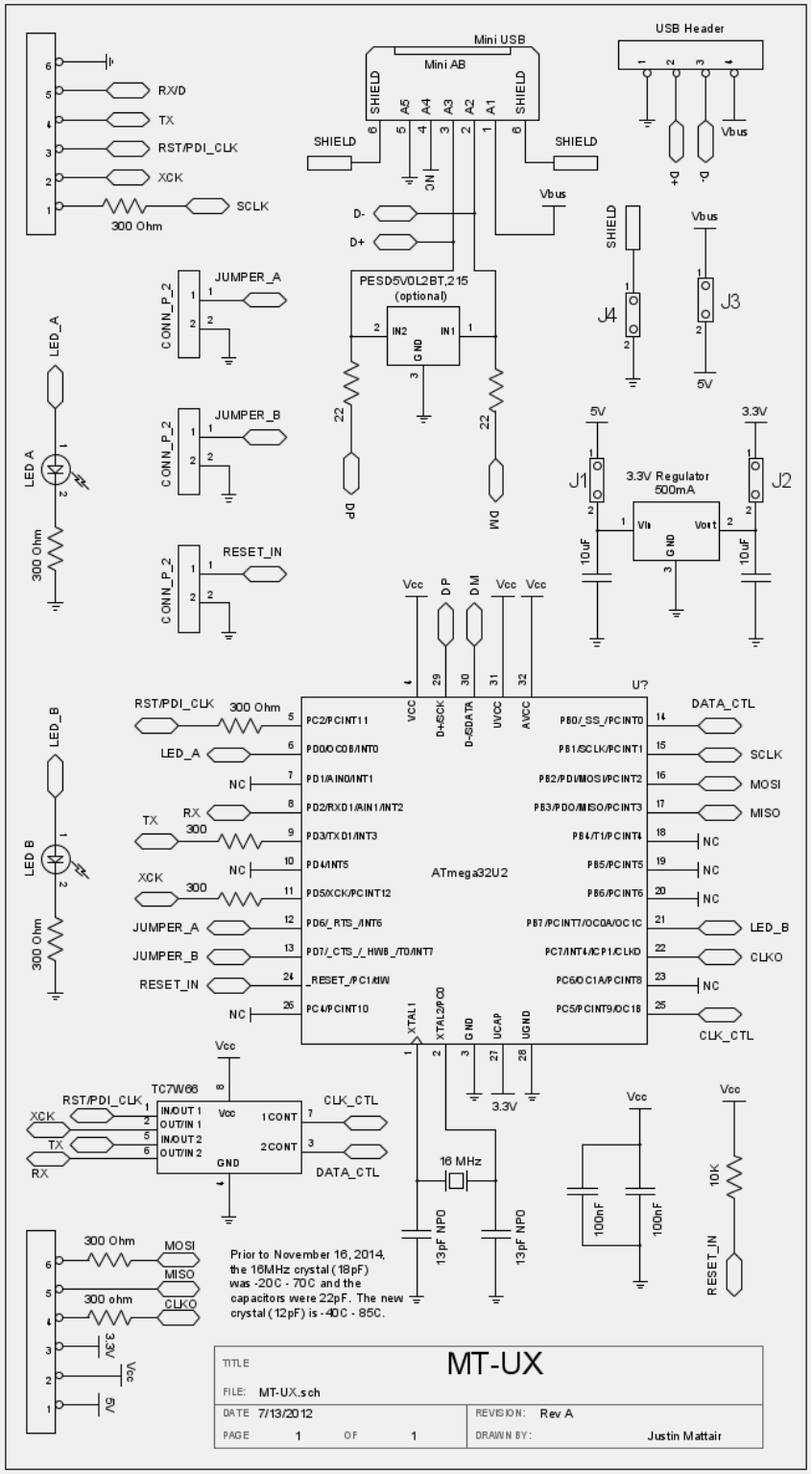
Please check the MattairTech website (http://www.MattairTech.com/) for firmware and software updates. Email me if you have any feature requests, suggestions, or if you have found a bug. If you need support, please contact me (email is best). You can also find support information at the MattairTech website. A support forum is planned. Support for AVRs in general can be found at AVRfreaks (http://www.avrfreaks.net/). There, I monitor the forums section as the user physicist.

Justin Mattair
MattairTech LLC
PO Box 1079
Heppner, OR 97836 USA
541-626-1531
justin@mattair.net
http://www.mattairtech.com/

Acknowledgments

Thanks to Dean Camera (http://www.fourwalledcubicle.com/) for his excellent LUFA library, AVRISPmkII clone, and DFU bootloader, all of which are used in the MT-UX firmware. Thanks to the members of AVRfreaks (http://www.avrfreaks.net/) for their support. Finally, thanks to Atmel for creating a great product, the AVR microcontroller.

Schematic


Mini USB
Mini AB
USB Header
SHIELD
RXD
TX
RST/PDI_CLK
XCK
SCLK
300 Ohm
D-
D+
Vbus
JUMPER_A
JUMPER_B
RESET_IN
CONN_P_2
PESD5V0L2BT,215
(optional)
LED_A
LED_B
LED A
LED B
J1
J2
J3
J4
5V
3.3V
3.3V Regulator
500mA
10uF
22
DP
DM
U?
ATmega32U2
Vcc
D-SCK
D-SDATA
UVCC
AVCC
DATA_CTL
SCLK
MOSI
MISO
CLKO
CLK_CTL
NC
TC7W66
16 MHz
13pF NPO
100nF
10K
Prior to November 16, 2014, the 16MHz crystal (18pF) was -20C - 70C and the capacitors were 22pF. The new crystal (12pF) is -40C - 85C.
TITLE
MT-UX
FILE: MT-UX.sch
DATE 7/13/2012
REVISION: Rev A
PAGE 1 OF 1
DRAWN BY: Justin Mattair

Legal Information

Copyright Notices

Software Disclaimer

Hardware Disclaimer

Trademarks

Licenses

LUFA USB Library

Appendix A: Precautions

WARNING

Care must be taken when configuring the solder jumpers. It is possible to cause permanent damage to the device or the target board by improperly setting the jumpers, or by supplying an incorrect external voltage (if used). Do not change any solder jumpers while the unit is powered. Do not power an XMEGA device with greater than 3.6V. Do not supply the 3.3V rail (if externally supplied) with a voltage outside the range of 3.0V to 3.6V.

CAUTION

The MT-UX contains static sensitive components. Use the usual ESD procedures when handling. For example, touch a grounded metal object prior to handling.

Appendix B: AVR Programmer Supported Devices

The MT-UX supports all Atmel AVR microcontrollers with an ISP, PDI, or TPI programming interface. These include the megaAVR series (ISP), the tinyAVR series (ISP, TPI), the XMEGA series (PDI), the USB AVRs (ISP), and the listed CAN and PWM AVRs. The following lists most of the supported models. Add to these the different voltage and speed grade variants. **New chips are usually automatically supported with updates to the host software (ie: Atmel Studio or AVRDUDE).**

MT-UX AVR Programmer megaAVR® Series Device Support

ATmega128, ATmega1280, ATmega1281, ATmega1284, ATmega1284P, ATmega128A, ATmega16, ATmega162, ATmega164A, ATmega164P, ATmega164PA, ATmega165, ATmega165A, ATmega165P, ATmega168, ATmega168A, ATmega168P, ATmega168PA, ATmega169, ATmega169A, ATmega169P, ATmega169PA, ATmega16A, ATmega16HVB, ATmega2560, ATmega2560, ATmega2561, ATmega32, ATmega324A, ATmega324P, ATmega324PA, ATmega325, ATmega3250, ATmega3250A, ATmega3250P, ATmega325A, ATmega325P, ATmega328, ATmega328P, ATmega329, ATmega3290, ATmega3290A, ATmega3290P, ATmega329A, ATmega329P, ATmega329PA, ATmega32A, ATmega32C1, ATmega32HVB, ATmega32M1, ATmega48, ATmega48A, ATmega48P, ATmega48PA, ATmega64, ATmega640, ATmega644, ATmega644A, ATmega644P, ATmega644PA, ATmega645, ATmega6450, ATmega6450A, ATmega6450P, ATmega645A, ATmega645P, ATmega649, ATmega6490, ATmega6490A, ATmega6490P, ATmega649A, ATmega649P, ATmega64A, ATmega64HVE, ATmega8, ATmega8515, ATmega8535, ATmega88, ATmega88A, ATmega88P, ATmega88PA, ATmega8A, ATmega8HVD

MT-UX AVR Programmer tinyAVR® Series Device Support

ATtiny10, ATtiny12, ATtiny13, ATtiny13A, ATtiny15, ATtiny167, ATtiny20, ATtiny2313, ATtiny2313A, ATtiny24, ATtiny24A, ATtiny25, ATtiny26, ATtiny261, ATtiny261A, ATtiny4, ATtiny40, ATtiny4313, ATtiny43U, ATtiny44, ATtiny44A, ATtiny45, ATtiny461, ATtiny461A, ATtiny48, ATtiny5, ATtiny84, ATtiny85, ATtiny861, ATtiny861A, ATtiny88, ATtiny9

MT-UX AVR Programmer XMEGA™ Series Device Support

ATxmega16A4U, ATxmega32A4U, ATxmega64A3U, ATxmega128A3U, ATxmega192A3U, ATxmega256A3U, ATxmega256A3BU, ATxmega64B3, ATxmega128B3, ATxmega64B1, ATxmega128B1, ATxmega16A4, ATxmega32A4, ATxmega64A4U, ATxmega128A4U, ATxmega64A3, ATxmega128A3, ATxmega192A3, ATxmega256A3, ATxmega256A3B, ATxmega64A1, ATxmega128A1, ATxmega16D4, ATxmega32D4, ATxmega64D4, ATxmega128D4, ATxmega64D3, ATxmega128D3, ATxmega192D3, ATxmega256D3, ATxmega128A1U, ATxmega128A4, ATxmega192A1, ATxmega256A1, ATxmega384A1, ATxmega64A4, ATxmega8e5, ATxmega16e5, ATxmega32e5, XMEGA B series, XMEGA C series

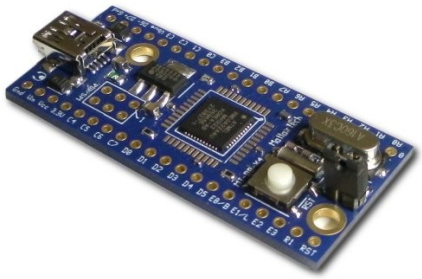
MT-UX AVR Programmer USB AVR Device Support

AT90USB1286, AT90USB1287, AT90USB162, AT90USB646, AT90USB647, AT90USB82, ATmega16U2, ATmega16U4, ATmega32U2, ATmega32U4, ATmega8U2

MT-UX AVR Programmer CAN AVR and PWM AVR Device Support

AT90CAN128, AT90CAN32, AT90CAN64, AT90PWM2, AT90PWM216, AT90PWM2B, AT90PWM3, AT90PWM316, AT90PWM3B

Appendix C: Other MattairTech Products

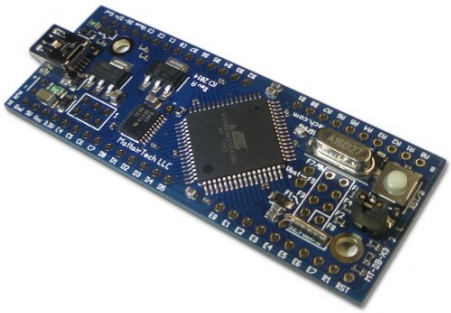


MT-DB-X4 USB AVR XMEGA board

- ATxmega128A4U USB XMEGA AVR
- 128KB FLASH, 8KB SRAM, 2KB EEPROM
- 3.3V LDO regulator (low quiescent current)
- 16MHz and 32.768KHz crystals
- LED, boot jumper, PDI header
- Reset button, mounting holes
- USB DFU bootloader preinstalled

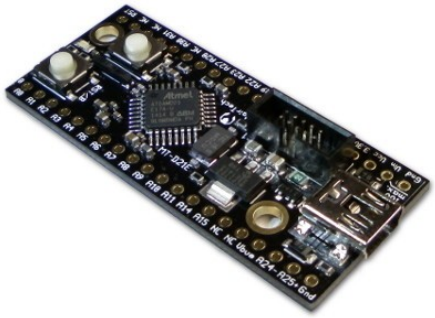
MT-DB-U6 USB AVR development board

- AT90USB646 / AT90USB1286 USB AVR
- 64KB/128KB FLASH, 4KB/8KB SRAM
- 5V, 500mA LDO regulator (3V-30V input)
- Automatic power source selection IC
- 16MHz and 32.768KHz crystals
- Arduino compatible
- CDC or DFU bootloader



MT-DB-X3 USB AVR XMEGA board

- XMEGA A3U, A3BU, C3, and D3 (64-pin)
- 32KB - 384KB FLASH, 4KB – 32KB SRAM
- 3.3V 250mA regulator (2uA quiescent current)
- Optional 5V 500mA regulator (23uA quiescent current)
- Optional auto-direction sensing level shifter
- 16MHz and 32.768KHz crystals, optional coin cell holder
- LED, boot jumper, PDI header, button, TWI pullups
- USB DFU bootloader preinstalled (except D variant)



MT-D21E USB ARM Cortex M0+ board

- ATSAMD21E17A or ATSAMD21E18A (32-pin)
- 128KB/256KB FLASH, 16KB/32KB SRAM
- Onboard 3.3V, 250mA LDO regulator (2uA quiescent)
- 16MHz and 32.768KHz crystals
- USB connector (power by USB or external up to 15V)
- Blue LED, 10-pin Cortex header, 2 buttons, I2C pullups
- USB Mass Storage Bootloader (no programmer required)