AVR-ISP-MK2 programmer
USER'S MANUAL

Document revision M, February 2016
Designed by OLIMEX Ltd, 2012

lead-free

All boards produced by Olimex LTD are ROHS compliant

DISCLAIMER

© 2016 Olimex Ltd. Olimex®, logo and combinations thereof, are registered trademarks of Olimex Ltd. Other product names may be trademarks of others and the rights belong to their respective owners.

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.

It is possible that the pictures in this manual differ from the latest revision of the board.

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.

This evaluation board/kit is intended for use for engineering development, demonstration, or evaluation purposes only and is not considered by OLIMEX to be a finished end-product fit for general consumer use. Persons handling the product must have electronics training and observe good engineering practice standards. As such, the goods being provided are not intended to be complete in terms of required design-, marketing-, and/or manufacturing-related protective considerations, including product safety and environmental measures typically found in end products that incorporate such semiconductor components or circuit boards.

Olimex currently deals with a variety of customers for products, and therefore our arrangement with the user is not exclusive. Olimex assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein.

THERE IS NO WARRANTY FOR THE DESIGN MATERIALS AND THE COMPONENTS USED TO CREATE AVR-ISP-MK2. THEY ARE CONSIDERED SUITABLE ONLY FOR AVR-ISP-MK2.

The product is based on Dean Camera's LUFA USB stack. More info at:
http://www.fourwalledcubicle.com/

The LUFA library is currently released under the MIT license, included below.

Copyright (C) Dean Camera, 2016 dean [at] fourwalledcubicle [dot] com

www.lufa-lib.org

Permission to use, copy, modify, and distribute this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the name of the author not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

The author disclaims all warranties with regard to this software, including all implied warranties of merchantability and fitness. In no event shall the author be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortuous action, arising out of or in connection with the use or performance of this software.

Table of Contents

CHAPTER 1 OVERVIEW

1. Introduction to the chapter

Thank you for choosing the AVR-ISP-MK2 programmer from Olimex! This document provides a user's guide for the Olimex AVR-ISP-MK2 programmer. As an overview, this chapter gives the scope of this document and lists the programmer's features. The document's organization is then detailed.

1.1 Features

Some of the unit's features are listed below:

- Tested and working with: AVR Studio 4; AVR Studio 5; Atmel Studio 6; Atmel Studio 7
- Tested and working with AVRDUDE and Arduino IDE
- Connects to PC via USB type B connector at 2.0 Full speed
- Does not need external power supply as it takes the power supply from USB
- Uses Atmel's 2×5 pin ICSP and 2×3 pin PDI and TPI connector layout
- Works with 5V and 3.3V targets (selectable) and can supply target with power
- Programs both flash and EEPROM
- Supports fuses and lock bit programming
- Upgradeable for future device support
- Supports target voltages from 1.8V to 5.5V
- 2 ribbon female-female cables – 10pin and 6pin each 20 cm (8") long
- Adjustable ISP programming speed (50Hz to 8MHz SCK frequency)
- Dimensions of the box: 45×30mm ~ (1.7×1.2)"

AVR-ISP-MK2 can program tinyAVR and megaAVR devices using the ISP Interface, tinyAVR devices using the TPI interface, and AVR XMEGA devices using the PDI Interface.

1.2 Target market and purpose of the board

AVR-ISP-MK2 is a ready-to-use programmer that behaves like Atmel AVRISP mkII. The board is suitable for programming microcontrollers that, according to their respective datasheets, have ISP, PDI, or TPI interfaces available for programming.

The firmware of Atmel AVRISP OLIMEX AVR-ISP-MK2 is based on the LUFA (Lightweight USB Framework for AVRs) software project started and maintained by Dean Camera. More info about the initial software and hardware idea can be found here:
http://www.fourwalledcubicle.com/AVRISP.php.

1.3 Document organization

Each section in this document covers a separate topic, organized as follow:
- Chapter 1 is an overview of the board usage and features
- Chapter 2 contains the general board layout
- Chapter 3 provides a guide for quickly setting up the board
- Chapter 4 mentions the main software tools used with AVR-ISP-MK2
- Chapter 5 is an explanation of the interfaces, the LEDs and the jumpers position
- Chapter 6 contains the revision history, useful links and support information

CHAPTER 2 PROGRAMMER DESCRIPTION

2. Introduction to the chapter

Here you would get acquainted with the main parts of the board. Note the names used on the board differ from the names used to describe them. For the actual names check the AVR-ISP-MK2 itself.

2.1 Layout (top view)
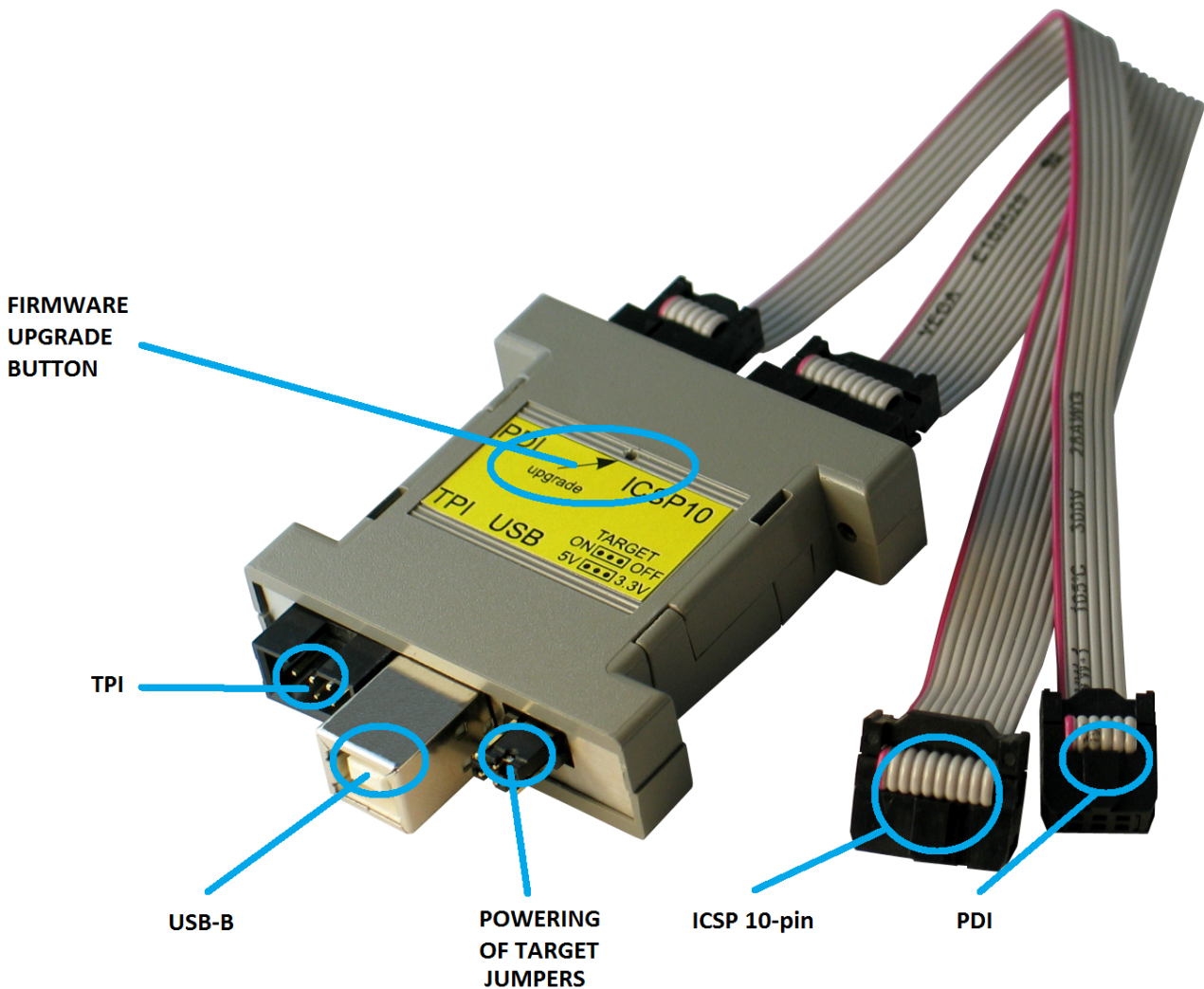
The programmer comes with two cables (10-pin one and 6-pin one) with female-female plastic connectors. The 10-pin cable fits only the ICSP10 connector. The 6-pin cable fits both the PDI and the TPI connectors – it can be used for either of these interfaces.

Note that there is no 6-pin ISP connector provided; there is only 10-pin ISP connector available.

CHAPTER 2 SETTING UP THE AVR-ISP-MK2 BOARD

2. Introduction to the chapter

This section helps you set up the AVR-ISP-MK2 development board for the first time. Please consider first the electrostatic warning to avoid damaging the board, then discover the hardware and software required to operate the board.

The procedure to power up the board is given, and a description of the default board behavior is detailed.

2.1 Electrostatic warning

The AVR-ISP-MK2 comes in a plastic cover but make sure boards and devices interfacing with the programmer are properly grounded.

2.2 Requirements

In order to set up the AVR-ISP-MK2 optimally, you might need to update your set of hardware and/or software tools. The major components needed are listed below.

Required hardware tools:

- USB type A to USB type B cable
- Atmel board or chip that can be programmed via ISP, TPI, or PDI interface. The list of supported devices is similar to the official AVRISP mkII list that can be found in the official Atmel web-site: http://www.atmel.com/tools/AVRISPMKII.aspx?tab=devices
- Personal computer with USB port

Recommended operating systems:

- Windows 7 or Windows 8 or Windows 10
- any frequently updated Linux distribution
- any Mac OS/OSX

Recommended software tools:

- Atmel Studio 6 or Atmel Studio 7
- AVRDUDE 6.0.1 or newer
- Arduino IDE

Depending on the connectors available on your target board you might need an adapter called AVR-ICSP since AVR-ISP-MK2 doesn't provide 6-pin ISP connector. This adapter is purchased separately. Alternatively, you can use jumper wires to establish the proper connections (you may find more details on the hardware connections at the end of chapter 4).

2.3 Quick setup

The purpose of this sub-chapter is to provide instructions on how to quickly set up AVR-ISP-MK2 under Windows.

The quick setup had been tested under Windows 10 x64 and under Windows 7 x64. The quick setup had been tested with Atmel Studio 7, AVRDUDE v6.2 compiled for Windows, and Arduino IDE 1.6.7 (which seems to have AVRDUDE v6.0.1).

This quick setup is not suitable for AVR Studio 4.19 (since the drivers for AVR Studio and the drivers for Atmel Studio are incompatible). The installation instructions for AVR-ISP-MK2 and AVR Studio 4.19 can be found further down in this document.

The steps below describe the typical way to use AVR-ISP-MK2 under Windows.

Step 1: Download and extract the drivers for AVR-ISP-MK2. These can be downloaded from the following location:

https://www.olimex.com/Products/AVR/Programmers/AVR-ISP-MK2/resources/DRIVER-MK2-AS-6-7-W10.zip

Step 2: Plug AVR-ISP-MK2 to your computer via a suitable USB cable.

Step 3: Open "Windows Device Manager" and find the entry for AVR-ISP-MK2. Right-click over it and select "Update Driver Software". A dialog would pop-up → select the bottom option "Browse my computer for driver software". In the next window click "Browse" and point to the folder where you extracted the drivers (Step 1).

Step 4: Open "Windows Device Manager" again. The programmer should be listed as "AVRISP mkII" under the "Atmel USB Devices" group. Confirm that the entry for AVR-ISP-MK2 looks like the picture below:
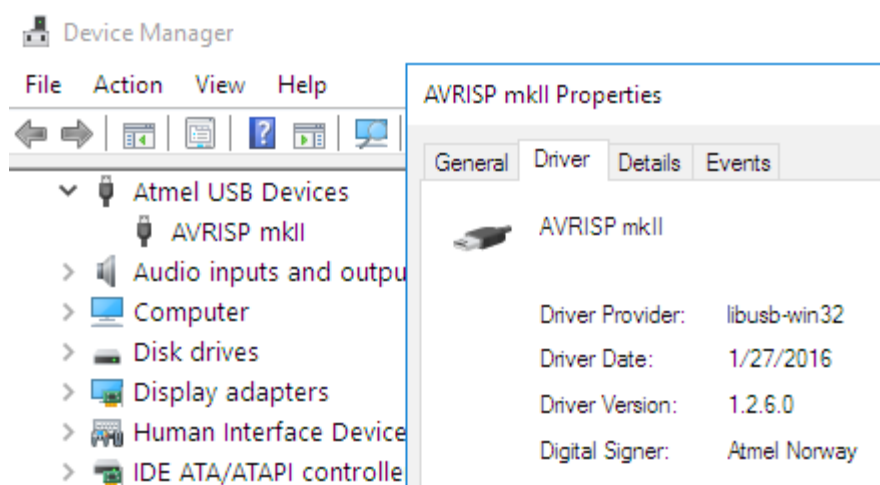

Further down in this sub-chapter you can find some optional steps that would help you use the newly installed AVR-ISP-MK2 with several software tools for Windows – Atmel Studio 7; AVRDUDE 6.2; Arduino IDE 1.6.7.

(Optional) Step 5: Visit Atmel's web-site and download the latest Atmel Studio. You would need to make a free registration in case you don't have one already. By the time of writing the web page to download Atmel Studio is this one:

http://www.atmel.com/Microsite/atmel-studio/

Install Atmel Studio. Inside the studio Olimex AVR-ISP-MK2 is recognized as "AVRISP mkII". Connect your target to the programmer, make sure that sufficient power supply is provided, and test the connection from "Tools" → "Device Programming". Select the proper target from the drop down menu and finally click "Apply". A successful connection is shown below:
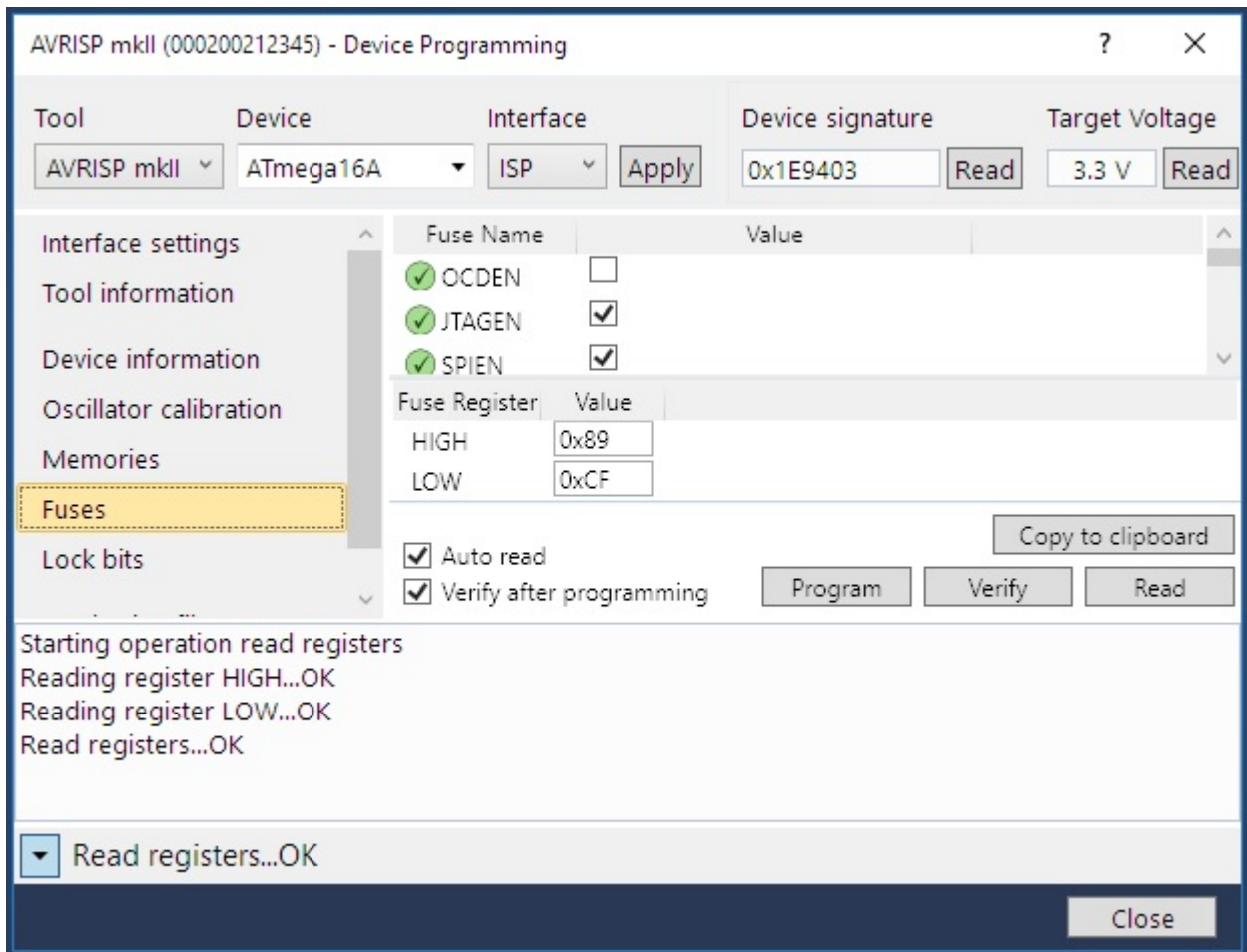
At this point you have AVR-ISP-MK2 and Atmel Studio 7 successfully working together. If you also wish to use AVRDUDE continue with the optional steps below.

(Optional) Step 6. Either compile from sources or download a prebuilt release of AVRDUDE. You can find archives with stable releases at the link below:
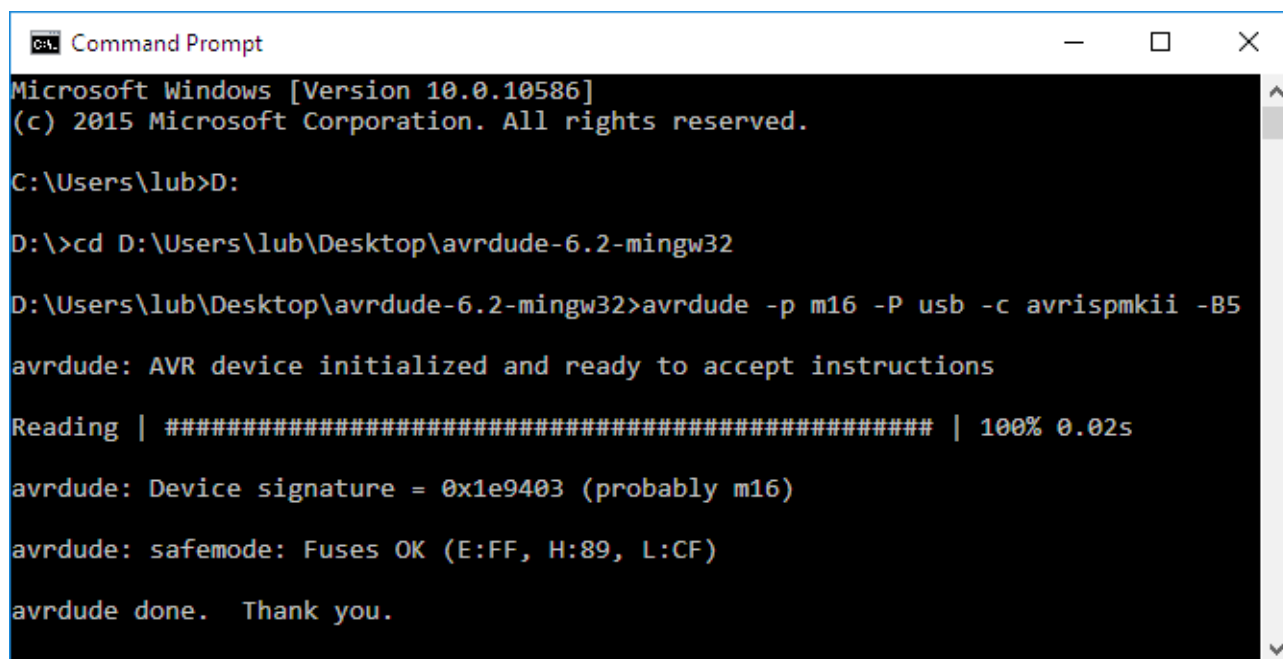
http://download.savannah.gnu.org/releases/avrdude/

For the Windows test listed below we used avrdude-6.2-mingw32.zip that was uploaded on 20th of November 2015. We also used our board MOD-IO that has an ATmega16a.

Once you have acquired a working version of AVRDUDE - open a command prompt (open search or run "CMD") and navigate to the folder that contains avrdude.exe. Make sure your target is powered and execute a sample command. For example, to check the microcontroller ID of ATmega16a execute:

avrdude -p m16 -P usb -c avrispmkii -B5

Refer to the picture below:

At this point you have AVR-ISP-MK2 properly working with AVRDUDE v6.2 and also with Atmel Studio 7. If you wish to also test the Arduino IDE connection continue with the next optional step.
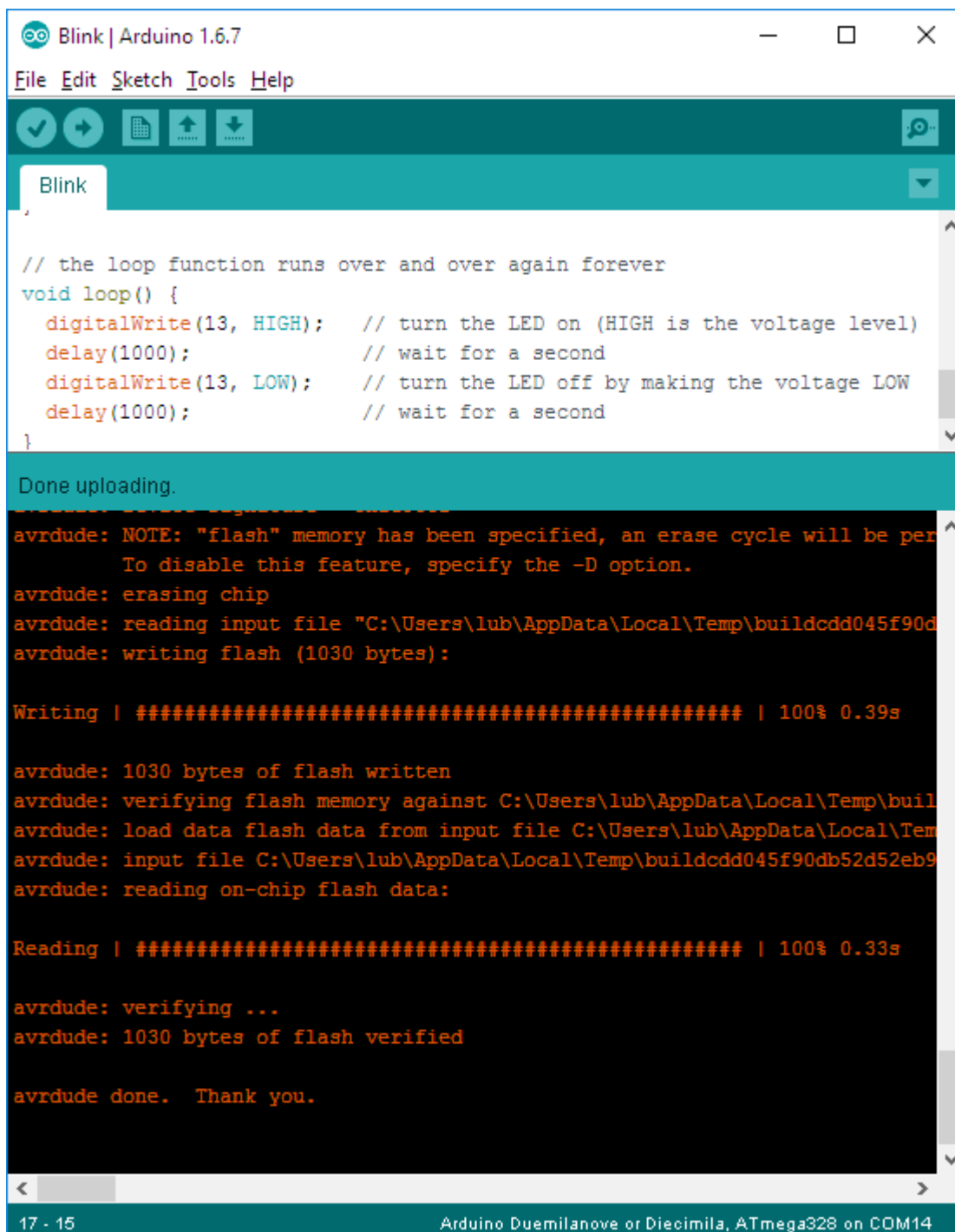
(Optional) Step 7. Download the install the latest version of Arduino IDE from the page below:

https://www.arduino.cc/en/Main/Software

For the Windows test listed below we used Arduino version 1.6.7 and our board OLIMEXINO-328 which comes with ATmega328p.

Start Arduino. Load the blinking LED example from File → Examples → Basics → Blink. Set the proper board and processor from Tools → Board and Tools → Processor. Remember to select AVRISP mkII from Tools → Programmer.

Finally, select Sketch → Upload Using Programmer. If you have verbose enabled you should receive messages similar to the one shown on the next page:

CHAPTER 4 RECOMMENDED SOFTWARE TOOLS

4. Introduction to the chapter

Here you will find short info on the most popular choices for software that can be used with the AVR programmer.

4.1 AVR Studio and Atmel Studio in Windows

The most popular choice amongst AVR fans that are using Windows OS. The IDE has everything you need for software development with AVR. You can download it from the official Atmel web-site.

AVR-ISP-MK2 comes with firmware compatible with AVR Studio, Atmel Studio and AVRDUDE under Windows. If you have doubt that you have the latest firmware version (or if you have accidentally corrupted the firmware or changed it with other software) you might want to refer to one of the following sub-chapters that explain how to upload the latest firmware: "4.2.1 Changing firmware for Windows users"; "4.2.2 Changing firmware for Linux users"; or "5.2 Upgrade firmware button"

Important – the Windows drivers for AVR Studio and Atmel Studio are incompatible. AVR Studio uses either Jungo or Atmel USB drivers and Atmel Studio requires LibUSB drivers. Atmel does not recommended using AVR Studio and Atmel Studio simultaneously on the same machine – in order to do it each time you would need to manually switch the AVR-ISP-MK2 drivers.

Something more – to complicate things further – the Windows drivers for AVR Studio and the Windows drivers for AVRDUDE are incompatible (but the drivers for Atmel Studio and AVRDUDE are the same so if you are using the latest releases you are already set).

The default way of installing the drivers for Atmel Studio, AVRDUDE and Atmel Studio under Windows is available in the previous chapter "2.3 Quick setup".

If you want to use AVR Studio 4 or 5, please refer to the next sub-chapter ("4.1.1 Driver installation for AVR Studio in Windows").

In order to use the product with AVRDUDE or Arduino IDE (which uses AVRDUDE) under Linux you would need to change the firmware – follow the procedure in chapter "4.2 Arduino, AVRDUDE, Linux" and "4.2.2 Changing firmware for Linux (AVRDUDE) users".

4.1.1 Driver installation for AVR Studio in Windows

Using AVR Studio 4 and 5 with AVR-ISP-MK2 is possible, but not recommended. Consider that the last release of AVR Studio 4 (v4.19) was released in September 2011. For comparison, Windows 8 was released October 26, 2012.

If you wish to use AVR Studio 4 or 5 make sure to use the installer provided in the archive with the firmware (usually named "driver-atmel-bundle-x.x.xxx.exe).

If you wish to use AVR Studio 4 and newer Atmel Studio together refer to the official Atmel workaround here:

 http://atmel.force.com/support/articles/en_US/FAQ/Downgrading-tools-to-use-older-Jungo-driver

4.2 Arduino, AVRDUDE, Linux

AVR-ISP-MK2 comes with firmware that works fine with the LibUSB driver.It is possible to use the AVR-ISP-MK2 with the latest versions of AVRDUDE and Atmel Studio without changing the firmware or the drivers.

It is impossible to use the AVR-ISP-MK2 simultaneously with AVRDUDE and AVR Studio. It would depend on you which one you prefer. You need to change the firmware of AVR-ISP-MK2 (using Atmel Flip or dfu-programmer software) AND the Windows drivers every time you switch between Arduino and AVR Studio (or vice versa).

How to proceed changing the firmware?

4.2.1 Changing firmware for Windows users

All files needed except for Atmel Flip (which is available at Atmel's web-site for free) should be downloaded from the official AVR-ISP-MK2 page at our web-site.

1. Connect the device to the USB and with a needle or sharp object press the button on it's back side (the one that the arrow points as "Upgrade"; the LEDs should turn off) which will put the processor in bootloader state – ready for firmware upgrade.

2. Download and install the Atmel Flip software. Check in device manager if the device is recognized as AT90USB162 if it isn't search for proper drivers in Atmel Flip installation folder.
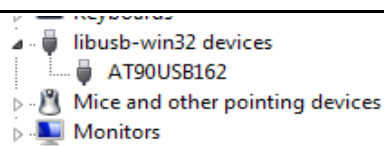
3. After that run the Atmel Flip and select AT90USB162 in Device -> Select
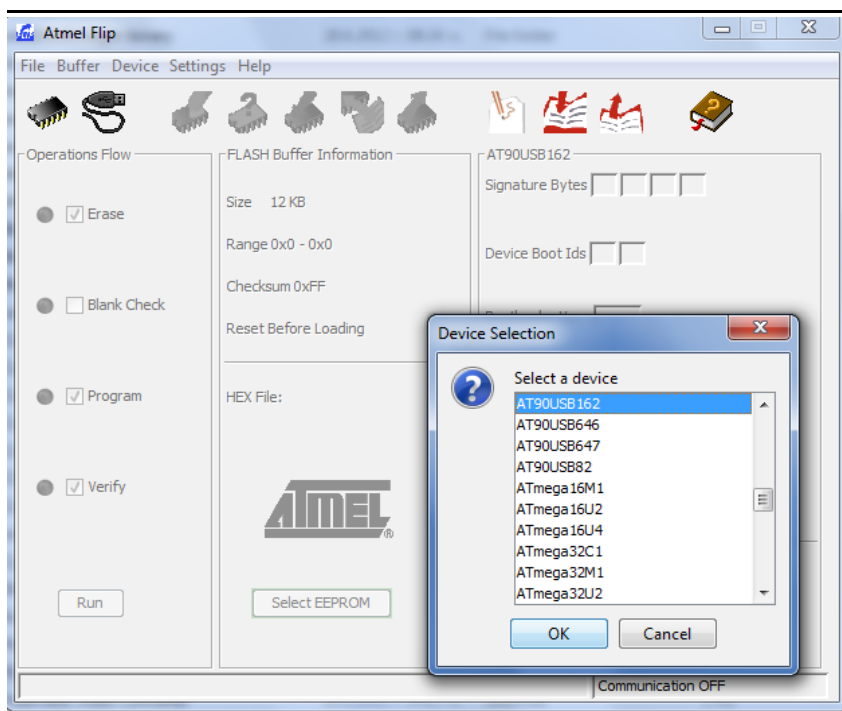
4. Select Settings -> Communication -> USB -> Open
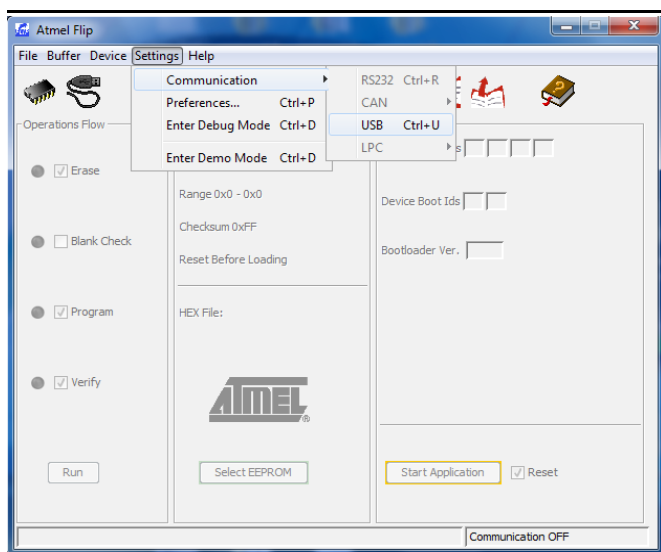
5. From File -> Load hex files choose either AVR456-studio-AVRISP-MKII.hex or libUSB-AVRISP-MKII.hex – the first hex is the firmware for AVR/Atmel Studio/Atmel USB/Jungo, the second one is for AVRDUDE/Arduino/libusb.

6. Click Run

7. Unplug and plug again AVR-ISP-MK2 from the USB.

8. Uninstall any previous drivers (selecting delete option) from device manager and scan for hardware changes or plug/unplug again. **Note that for the Atmel USB/Jungo drivers you might need to go to the control panel and uninstall the drivers from "Programs and Features" - the**

drivers are called either "Jungo" or "Atmel USB".
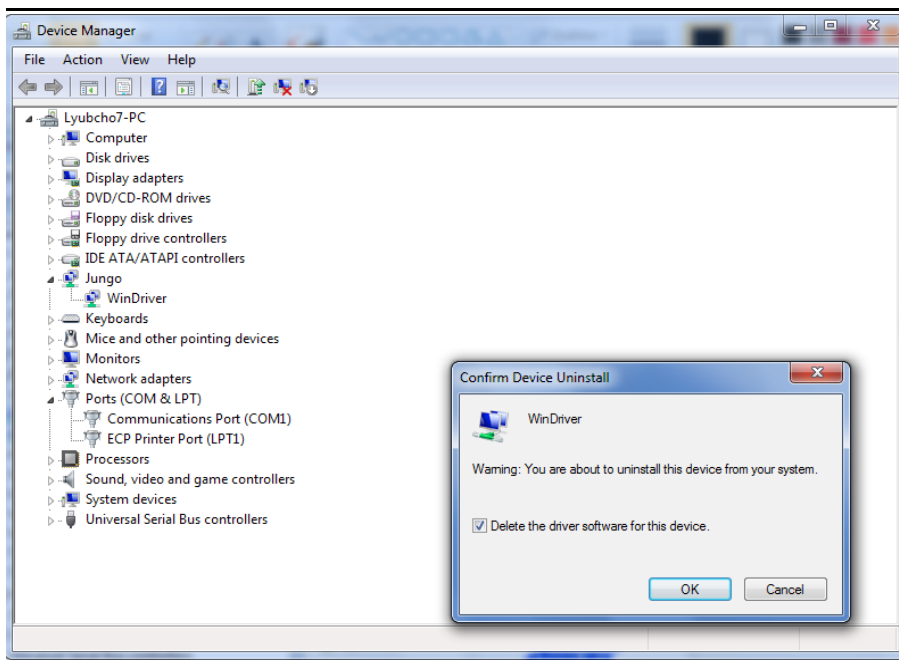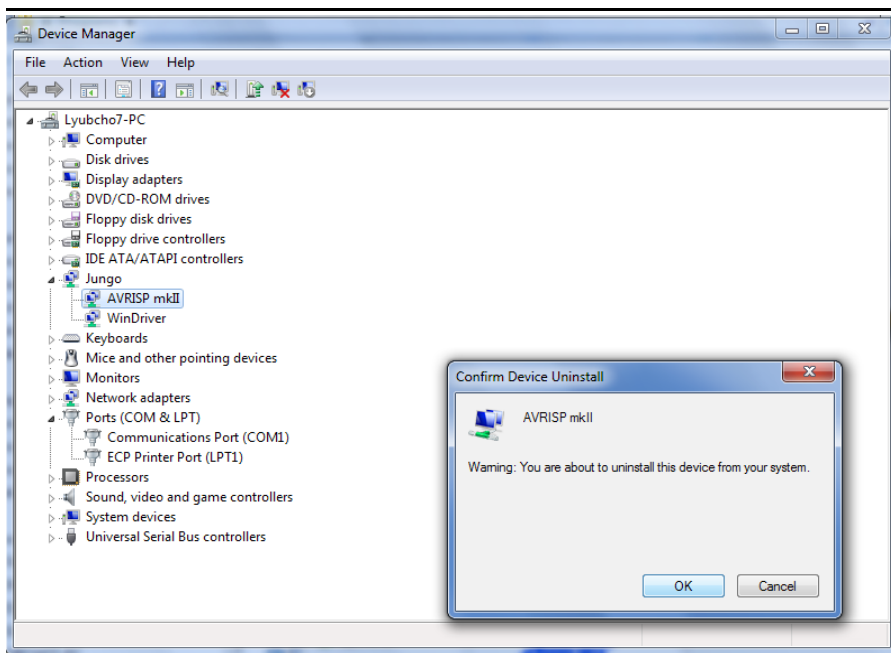
9. Install the new drivers corresponding to your needs (usually by right-clicking over the programmer entry and selecting "Update Driver Software...")

The picture below shows how to install the libusb-win32 drivers over the Jungo ones – you have to point to the libusb folder and agree to install despite the warning.
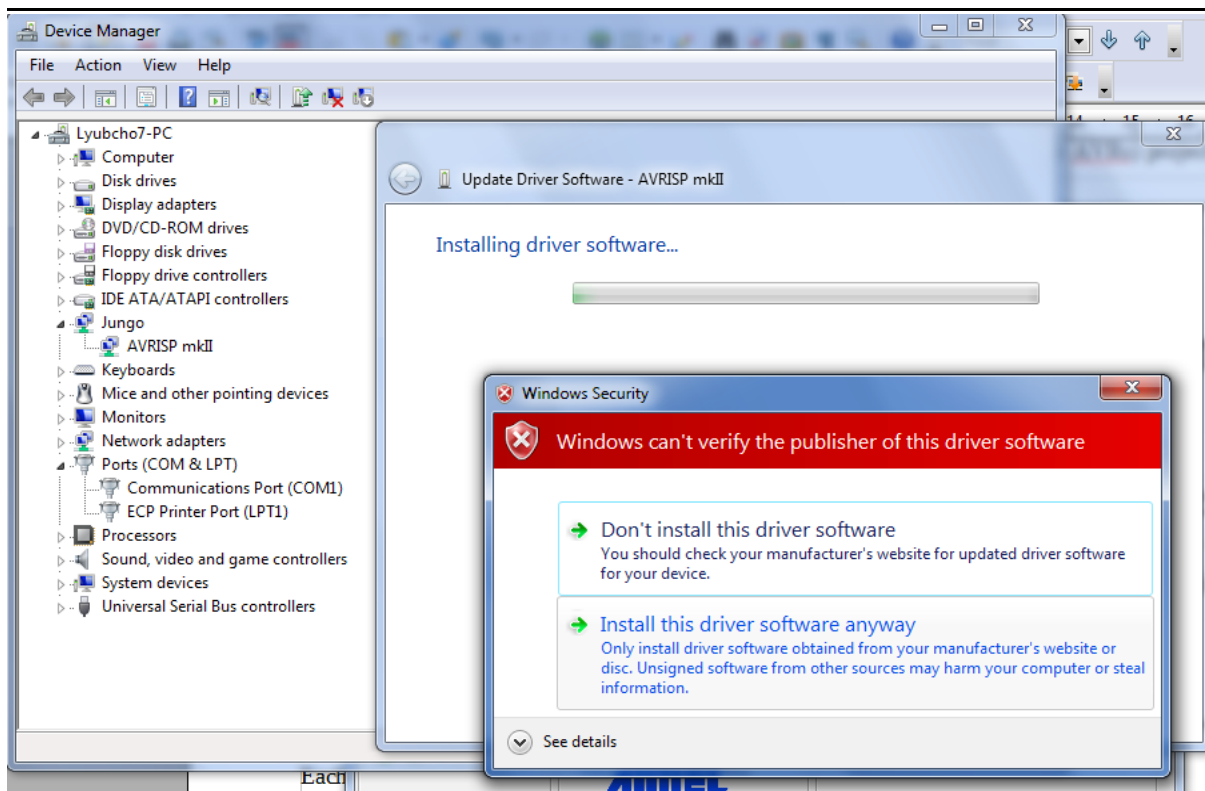
Important: it is possible that the drivers get blocked under Windows 7, Windows 8, or Windows 10. If this is the case when you visit the "Windows Device Manager" there would be yellow triangle notifying for the lack of driver signature. You would need to disable the "Driver Signature Enforcement", which is enabled by default.

4.2.2 Changing firmware for Linux (AVRDUDE) users

1. Download and install "dfu-programmer" package (apt-get...).

2. Connect the programmer AVR-ISP-MK2 to the computer and press the upgrade button with a sharp object (you can use a piece of thick wire or a needle).

3. Erase the microcontroller inside AVR-ISP-MK2 with the command:

```
dfu-programmer at90usb162 erase
```

4. Flash the firmware with:

```
dfu-programmer at90usb162 flash --debug 6 libUSB-AVRISP-MKII.hex
```

5. Create new file /etc/udev/avrisp.rules

```
SUBSYSTEM!="usb_device", ACTION!="add", GOTO="avrisp_end"

# Atmel Corp. JTAG ICE mkII
ATTR{idVendor}=="03eb", SYSFS{idProduct}=="2103", MODE="660", GROUP="dialout"
# Atmel Corp. AVRISP mkII
ATTR{idVendor}=="03eb", SYSFS{idProduct}=="2104", MODE="660", GROUP="dialout"
# Atmel Corp. Dragon
ATTR{idVendor}=="03eb", SYSFS{idProduct}=="2107", MODE="660", GROUP="dialout"

LABEL="avrisp_end"
```

In some kernels SYSFS module is disabled by default. There are two flags responsible and you can enable SYSFS by manipulating one or both of the flags CONFIG_SYSFS and CONFIG_SYSFS_DEPRECATED_V2. In most cases it might be better and faster to swap SYSFS{idProduct} with ATTR{idProduct}.

6. Create a virtual link to the file and give it a rule priority

```
cd /etc/udev/rules.d
sudo ln ../avrisp.rules 60-avrisp.rules
```

7. Check you're in the dialout group

```
groups
```

8. Restart udev

```
sudo restart udev
```

That's it! You can test the connection following the tips on the next page.

To test the connection try:

```
avrdude -p m16 -c avrispmkII -P usb -b 115200,
```

assuming that you have ATmega16 microcontroller (check the AVRDUDE documentation for list of microcontrollers). The output should be like:

```
C:\Users\       \Desktop\avrdude-5.11-Patch7610-win32\avrdude-5.11-Patch7610-win
32>avrdude -p m16 -c avrispmkII -P usb -b 115200

avrdude: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.03s

avrdude: Device signature = 0x1e9403

avrdude: safemode: Fuses OK

avrdude done.  Thank you.
```

A sample programming line for AVRDUDE would be:

```
avrdude -p m328p -P usb -c avrispmkii -e -U flash:w:test.hex,
```

assuming you have ATmega328P and your binary code is named "test.hex". The output should be like:

```
C:\Users\       \Desktop\avrdude-5.11-Patch7610-win32\avrdude-5.11-Patch7610-win
32>avrdude -p m16 -c avrispmkII -P usb -b 115200  -U flash:w:test_leds.hex

avrdude: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.03s

avrdude: Device signature = 0x1e9403
avrdude: NOTE: FLASH memory has been specified, an erase cycle will be performed
        To disable this feature, specify the -D option.
avrdude: erasing chip
avrdude: reading input file "test_leds.hex"
avrdude: input file test_leds.hex auto detected as Intel Hex
avrdude: writing flash (260 bytes):

Writing | ################################################## | 100% 0.14s

avrdude: 260 bytes of flash written
avrdude: verifying flash memory against test_leds.hex:
avrdude: load data flash data from input file test_leds.hex:
avrdude: input file test_leds.hex auto detected as Intel Hex
avrdude: input file test_leds.hex contains 260 bytes
avrdude: reading on-chip flash data:

Reading | ################################################## | 100% 0.09s

avrdude: verifying ...
avrdude: 260 bytes of flash verified

avrdude: safemode: Fuses OK

avrdude done.  Thank you.
```

Note that latest AVRDUDE versions (starting with AVRDUDE 6.x.x) feature a bug that affects AVR-ISP-MK2. Either use AVRDUDE versions prior to 6.x.x or apply this patch to the AVRDUDE

sources and compile.

The same problems affects the newest releases of Arduino IDE – they use AVRDUDE versions 6.x.x or newer. The last suitable Arduino IDE version that uses pre-6.x.x AVRDUDE version is Arduino IDE 1.5.7.

4.2.3 Using Arduino IDE with AVR-ISP-MK2

1. Download suitable Arduino IDE version and run it
2. Plug the AVR-ISP-MK2 to the computer and follow chapter 4.2.1 for firmware and driver install
3. Power your target Arduino-compatible board via the AVR-ISP-MK2 (setting the jumpers properly) or externally (make sure AVR-ISP-MK2 doesn't provide additional voltage – TERGET jumper is set to OFF)
4. Make sure you use the correct interface. If you need adapter cable (10-pin ICSP to 6-pin ICSP) – please check the schematic at the next chapter – 4.2.4
5. Set the correct values for board and programmer in the Arduino IDE – Tools->Board and Tools->Programmer.
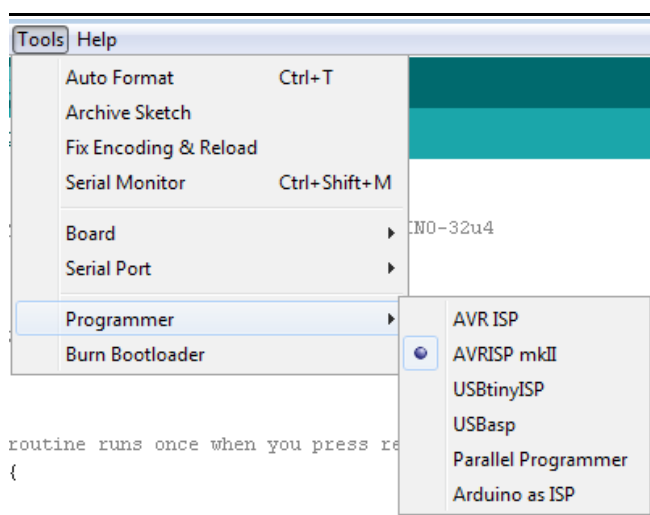
6. Open a simple blinking LED example and adjust the pin values according to the schematic of your device (some boards doesn't require a pin number change).
7. Remember to click "Upload Using Programmer"! By default the right arrow in the interface would attempt to upload via the bootloader!

In case you encounter erros consider the AVRDUDE bug mentioned at the end of the previous sub-chapter. Try using Arduino IDE version 1.5.7 or earlier.
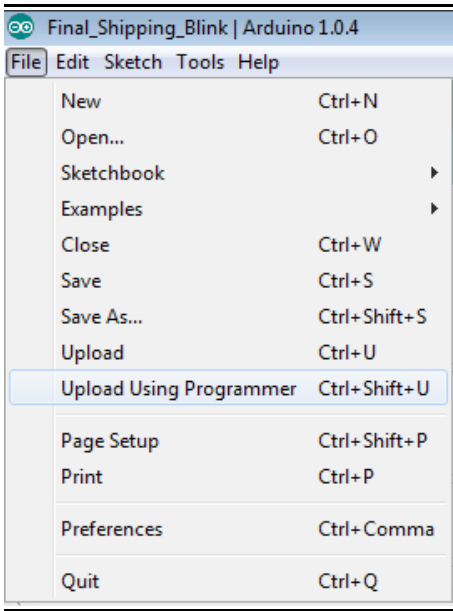
Special mention for Olimex boards – sometimes the board refuses to reset before programming when using Arduino. Try holding the reset button and release it simultaneously while clicking "Upload Using Programmer".

4.2.4 10-pin ICSP to 6-pin ICSP cable

ICSP is the most common one but some boards have only 6-pin ICSP. In such a case you should either get the low-cost adapter AVR-ICSP we manufacture or make a 10-pin to 6-pin adapter as show below (or just use male-female jumper cables):
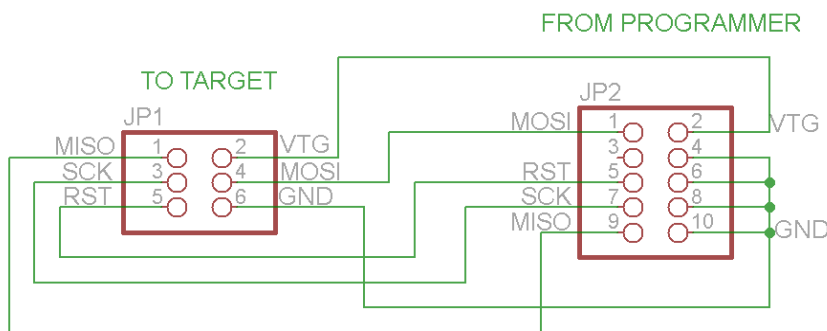
CHAPTER 5 INTERFACES AND HARDWARE

5. Introduction to the chapter

In this chapter are presented the interfaces found on the programmed that can be found on the board. Jumpers functions are described.

5.1 Programming interfaces

AVR-ISP-MK2 has three programming interfaces aiming at different processors. More information about each of them is located in the following sub-chapters.

5.1.1 10-pin ICSP

The ICSP (in-circuit serial programming) interface is suitable for programming tinyAVR and megaAVR chips.

The ICSP connector is not over-voltage/over-current protected. The two simple protection methods implemented are (1) diodes for electrostatic discharge and (2) series of resistors for basic protection.

Depending on the target you might need https://www.olimex.com/Products/AVR/Programmers/AVR-ICSP/ since AVR-ISP-MK2 doesn't provide 6-pin ICSP connector. Note it is bought separately.

5.1.2 6-pin PDI

The 6-pin PDI is used for programming AVR XMEGA.

5.1.3 6-pin TPI

The interface is typically used for programming tinyAVR.

5.2 Upgrade firmware button

The upgrade firmware button is used to change the firmware of the board through a bootloader. This is considered a normal upgrade procedure and requires no additional hardware. If however, the bootloader program is not working properly or it got corrupted then you would need to program the board with a working bootloader – please refer to chapter "5.3 Bootloader restore".

The upgrade firmware button might be found in a small hole between the PDI interface and the ICSP10 interface. The button is used to enter bootloader mode. The memory can then be programmed with new firmware. If you wish to update your firmware check if there is newer version in the web and follow the algorithm:

1) Connect the programmer to the USB and with a sharp object (needle or pin) press the upgrade

pin – it is in a small hole at the back of the board (this will start the bootloader and will turn off the LED, also probably will show a new unrecognized device in device manager for which we will install driver in step 3)

2) Download and install the latest version of "Atmel Flip" software (it can be downloaded from the Atmel's web-site)

3) Open its install folder and update the software of the unrecognized device (usually under "Other devices" tab) with the drivers from folder named "usb"; the device should now be recognized as AT90USB162 under "libusb-win32" tab

4) Start "Atmel FLIP" and click "Select a target device" -> choose AT90USB162

5) Click "Select a Communication Medium" and then USB medium

6) From "File -> load HEX file" choose this HEX ([CLICK FOR DOWNLOAD](#)) and click "RUN" in the "Operations Flow" section

7) Disconnect the AVR-ISP-MK2 from the USB and connect it again

Please refer to *4.2 Arduino, AVRDUDE, Linux* for more detailed description.

5.3 Bootloader restore

Simply said, the bootloader is a small software program which allows the user to change the firmware of AVR-ISP-MK2 via USB and "Atmel Flip". It is stored on the main microcontroller inside AVR-ISP-MK2. In rare cases it is possible to corrupt this software bootloader and, thus, make it impossible to change the firmware inside using only USB and "Atmel Flip". In such cases you would need to re-upload it with the help of a third-party programmer.

Even if the board never had any issue it is possible that the bootloader might work with some operating systems, and refuse to work with others. This is because the different USB policies and timings – such changes require a change in the bootloader itself. That is why we are updating the provided bootloader regularly with the latest official version. Please note:

Changing the firmware with the provided elf file should be performed only in extreme scenarios when all other options had been long exhausted. The user should proceed with extreme caution because the upgrade procedure might seriously damage the hardware of the board.

Performing the bootloader update requires a third-party programming tool, soldering skills and voids the warranty of the unit!

The steps to follow are listed below:

1. Open the case of the programmer carefully. Please note:

Opening the case of the debugger voids the warranty!

2. Change the position of the SMT jumper ER/PGM to the opposite position. This would require

cutting between the pads of the original position and soldering the pads of the new position. After the change the jumper should be in PGM position. This changes the direction of programming on the 10-pin ICSP connector – instead of programming – it would get programmed.

3. Connect a compatible programmer (able to program AT90USB162, it can also be another AVR-USP-MK2) to AVR-ISP-MK2's 10-pin ICSP connector on one side and to your computer on the other.

4. Download and extract the archive with the drivers (CLICK FOR DOWNLOAD).

5. Launch your favorite programming tool (AVR Studio, Atmel Studio, AVRDUDE, etc.) and import the elf – it is located in folder DEFAULT-ELF in the archive you downloaded in point 4.

6. Program the AT90USB162.

7. At this point you should test if the bootloader works as expected. It is now ready-to-use. Refer to chapters 4.2 and 5.2 of this document for detailed institutions on how to use the bootloader to connect to "Atmel Flip". Note that to be able to use the ICSP interface for the programming of targets you would need to perform the next step 8.

8. Switch back the position of the jumper to original position (after the switch the jumper should be in position ER).

9. Test and carefully close the case.

5.4 Jumpers description

Please note that the two jumpers on the board are PTH (plated-through hole) type and should be easy to mount/dismount.

5.4.1 TARGET jumper

TARGET jumper controls the powering of the target board. If it is in position ON (check the diagram on the back of the plastic cover) it will provide either 3.3V or 5V to the target board (depending on the position of the POWER jumper)

The default position is OFF.

5.4.2 POWER jumper

If you have set the TARGET jumper to position ON POWER jumper controls whether 3.3V or 5V are provided to the target board.

The maximum amperage available for both 3.3V and 5V modes is 300mA.

The default position is 5V.

5.5 LEDs explained

There are two 3mm LEDs – ORANGE and a double colored GREEN/RED.

Green – when GREEN LED is on – USB connection is active
Green – GREEN LED on and RED blink – USB connection active and programming activity
(erasing, flashing, programming) – unless TARGET is on

Orange – when ORANGE LED is off – TARGET jumper is in position OFF
Orange – ORANGE LED on – TARGET jumper is in position on, BLINK means programming
activity (erasing, flashing, programming)

After a firmware change the LEDs might behave in a different way:

Orange – turns on during programmer initialization and after that turns off; flashes when target
operation is performed – read//write
Red – turns on after successful USB initialization

5.6 USB connector

The AVR-ISP-MK2 has a USB type B connector. It is used to establish a connection between a
personal computer and AVR-ISP-MK2.

The typical hardware setup is as follows:

PERSONAL COMPUTER ↔ USB CABLE ↔ AVR-ISP-MK2 ↔ [ADAPTER] ↔ TARGET

If you encounter unstable irregular behavior of the programmer or if the powering from the
programmer to your target is unreliable, please follow the recommendations below:

- As with most USB devices please ensure that your cables are good enough. Low-cost low-
quality cables might have very high inner resistance, thus lowering the voltage that the unit
receives.
- Avoid using USB hubs and splitters.
- Avoid using USB 3.0 ports.
- Test with another USB port.
- Test with another computer.

CHAPTER 6 REVISION HISTORY AND SUPPORT

6. Introduction to the chapter

In this chapter you will find the current and the previous version of the document you are reading. Also the web-page for your device is listed. Be sure to check it after a purchase for the latest available updates and examples.

6.1 Document revision table

Revision	Changes	Modified Page #
A, 11.07.12	Initial Creation	All
B, 23.07.12	Included LUFA license text	3
C, 30.08.12	Added info how to update the firmware for AVRDUDE/Arduino IDE	9, 10, 11
D, 21.01.13	Fixed a number of broken links and made a couple of points clearer	13, 14, 17
E, 05.02.13	Added info on maximum power available; added info on the LEDs and expanded the ICSP information	13, 15
F, 10.05.13	Made a couple of references to make firmware changes clearer; Fixed several links	5, 7, 9, 13, 14
G, 16.05.13	Added a FAQ section	17
H, 21.08.13	Expanded chapter 4 with a lot more specifics	12, 13, 14, 15
I, 11.11.13	Added new FAQ	21
J, 01.09.14	Added few specifics about the programmer installation under Linux. Added information on bootloader restore Added information about OLIMEX AVR-ICSP	12, 13 16, 17, 18 21
K, 08.07.15	Improved dfu-programmer information Added information about AVRDUDE 6.x.x bug that affects LUFA-based tools Added USB connector information Additional FAQ	13 14 20 23
L, 08.10.15	General document improvements	All
M, 17.02.16	Improved the information about the drivers	All

6.2 Useful web links and purchase codes

The web page you can visit for more info on your device is
https://www.olimex.com/Products/AVR/Programmers/AVR-ISP-MK2/.

For more info on the original ISP-MK2 design please visit:
http://www.atmel.com/tools/AVRISPMKII.aspx.

You can get the latest updates on the software at: http://www.fourwalledcubicle.com/LUFA.php.

ORDER CODES:

AVR-ISP-MK2 – assembled and tested AVR-ISP compatible programmer

AVR-ICSP – 10 ↔ 6 pin ICSP adapter cable, since AVR-ISP-MK2 has only 10-pin ICSP interface

How to purchase Olimex products?

You can order from our web-shop or from any of our distributors. The list of official Olimex
distributors might be found at: https://www.olimex.com/Distributors/.

Please check https://www.olimex.com/ **for more info.**

6.3 FAQ

Q1: The programmer is not recognized in the Studio! What should I do?

A1: Most of the problems that customers encounter with AVR-ISP-MK2 are related to the driver
installation since the different versions of AVR and Atmel Studios require different drivers. Atmel
Studios require LibUSB drivers; AVR Studios require Jungo drivers.

For example, Atmel Studio 7 would not work with Jungo drivers.

**Q2: I can connect to the programmer, but can't connect the programmer to the target. Also
when I try to read the device ID there are errors. What should I do?**

A2: The most common ways to resolve such an issue are:

1. Check if you have used the proper interface according to your target.

2. If using ISP try changing the ISP frequency with lower or higher values – sometimes settings
lower ISP frequency restricts reading/writing – try to keep 4KHz at least.

3. If you are using a custom made board or you connect the microcontroller directly with wires
make sure you follow the AVR ICSP programming standards.

4. Make sure you are using the ISP interface of AVR-ISP-MK2. If your target board has a 6 pin
connector instead of the 10 pin on AVR-ISP-MK2 make appropriate adapter.

5. Ensure the target is powered. It can be powered from AVR-ISP-MK2 by changing the "Target" jumper to position On (to the USB).

Q3: I use Atmel Studio 6. The device is visible and properly installed but my target microcontroller is nowhere to be found in the drop-down menus. How shall I proceed?

A3: Some target microcontrollers lack ready configuration files in the stock versions of Atmel Studio 6. You might need to add such manually follow the pattern of the already existing .xml files located by default in C:\Program Files (x86)\Atmel\Atmel Studio 6.x\tools\AVRISPmkII\xml

Q4: I have your AVR-ISP-MKII programmer and it works OK but in Atmel Studio it shows that the target voltage is 3.3V, when I'm sure the board gets 5.0V. What is wrong?

A4: Olimex AVR-ISP-MK2 cannot detect the target voltage! The information about the voltage in Atmel Studio is just a place holder – it shouldn't be taken as a correct reading!

Q5: Does the firmware of AVR-ISP-MK2 differ from the original Atmel firmware?

A5: The firmware of AVR-ISP-MK2 is based on the LUFA project by Dean Camera.

Q6: I want to use AVR-ISP-MK2 with Arduino IDE. What should I do?

A6: Refer to chapter "2.3 Quick setup" above.

Q7: I have performed a firmware switch. My drivers are properly installed. Still I can't properly program my target with AVRDUDE. I also tested with the latest Arduino IDE. What is the problem? Is my programmer bricked?

A7: You are not using the latest firmware and the latest LibUSB drivers recommended.

If you are using other set of drivers and you don't want to install more driver then try AVRDUDE versions prior to 6.x.x or apply this patch to the AVRDUDE sources and compile.

The same problems appears in newest releases of Arduino IDE – they use AVRDUDE versions 6.0.1 or newer. The last suitable Arduino IDE version that uses pre-6.x.x AVRDUDE version is Arduino IDE 1.5.7.

More about the issue can be found here: http://www.avrfreaks.net/forum/lufa-based-avrisp-mkii-doesnt-work-avrdude-601

Q8: Previously I used AVR Studio 4 successfully. After I installed Atmel Studio 7 I can no longer connect to AVR-ISP-MK2 in AVR Studio 4. What can I do to use AVR Stduio 4 and Atmel Studio 7 at the same time?

A8: The drivers installed with Atmel Studio 7 overwrote the drivers that worked with AVR Studio 4. These drivers are not compatible you would have to downgrade the drivers manually. The official Atmel workaround can be found here:

http://atmel.force.com/support/articles/en_US/FAQ/Downgrading-tools-to-use-older-Jungo-driver

6.4 Product support

For product support, hardware information and error reports mail to: support@olimex.com. Note that we are primarily a hardware company and our software support is limited.

Please consider reading the paragraph below about the warranty of Olimex products.

```
Warranty and returns:


Our boards have lifetime warranty against manufacturing defects and components.

During development work it is not unlikely that you can burn your programmer or
development board. This is normal, we also do development work and we have damaged A
LOT of programmers and boards during our daily job so we know how it works. If our
board/programmer has worked fine then stopped, please check if you didn't apply over
voltage by mistake, or shorted something in your target board where the programmer was
connected etc. Sometimes boards might get damaged by ESD shock voltage or if you spill
coffee on them during your work when they are powered.


Please note that warranty do not cover problems caused by improper use, shorts, over-
voltages, ESD shock etc.


If the board has warranty label it should be not broken. Broken labels void the
warranty, same applies for boards modified by the customer, for instance soldering
additional components or removing components – such boards will be not be a subject of
our warranty.


If you are positive that the problem is due to manufacturing defect or component you
can return the board back to us for inspection.

When we receive the board we will check and if the problem is caused due to our fault
and we will repair/replace the faulty hardware free of charge, otherwise we can quote
price of the repair.


Note that all shipping costs back and forth have to be covered by the customer. Before
you ship anything back you need to ask for RMA. When you ship back please attach to it
your shipping address, phone, e-mail, RMA# and brief description of the problem. All
boards should be sent back in antistatic package and well packed to prevent damages
during the transport.
```