AVR-ISP-MK2 programmer
USER'S MANUAL

Revision I, October 2013
Designed by OLIMEX Ltd, 2012

All boards produced by Olimex LTD are ROHS compliant

DISCLAIMER

© 2013 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 2013 Dean Camera (dean [at] fourwalledcubicle [dot] com)

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 disclaim 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

- Tested and working with AVR Studio 4, AVR Studio 5, ATMEL Studio 6
- Connects to PC via USB 2.0 Full speed
- Does not need external power supply as it takes the power supply from USB
- Uses Atmel's 2x5 pin ICSP and 2x3 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
- Adjustable ISP programming speed (50Hz to 8MHz SCK frequency)
- Dimensions: 45x30 mm (1.7x1.2") + 20 cm (8") ribbon cable

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 low-cost clone of AVRISP-MKII. The board is suitable for programming microcontrollers that according to their respective datasheets allow ICSP, PDI or TPI programming modes.

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

1.3 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 provides a guide for quickly setting up the board
- Chapter 3 contains the general board diagram and layout
- 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 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.

2.3 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 needed components are listed below.

Hardware tools:

- USB-A to USB-B cable
- Atmel board or chip that can be programmed by AVR-ISP-MK2. The list of supported devices can be found in the official Atmel web-site: http://www.atmel.com/tools/AVRISPMKII.aspx?tab=devices
- Personal computer with USB port

Software tools:

- AVR-STUDIO 4, 5 or ATMEL-STUDIO 6 with AVR-GCC compiler
- AVR-DUDE

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.

CHAPTER 3 AVR-ISP-MK2 BOARD DESCRIPTION

3. Introduction to the chapter

Here you 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.

3.1 Layout (top view)
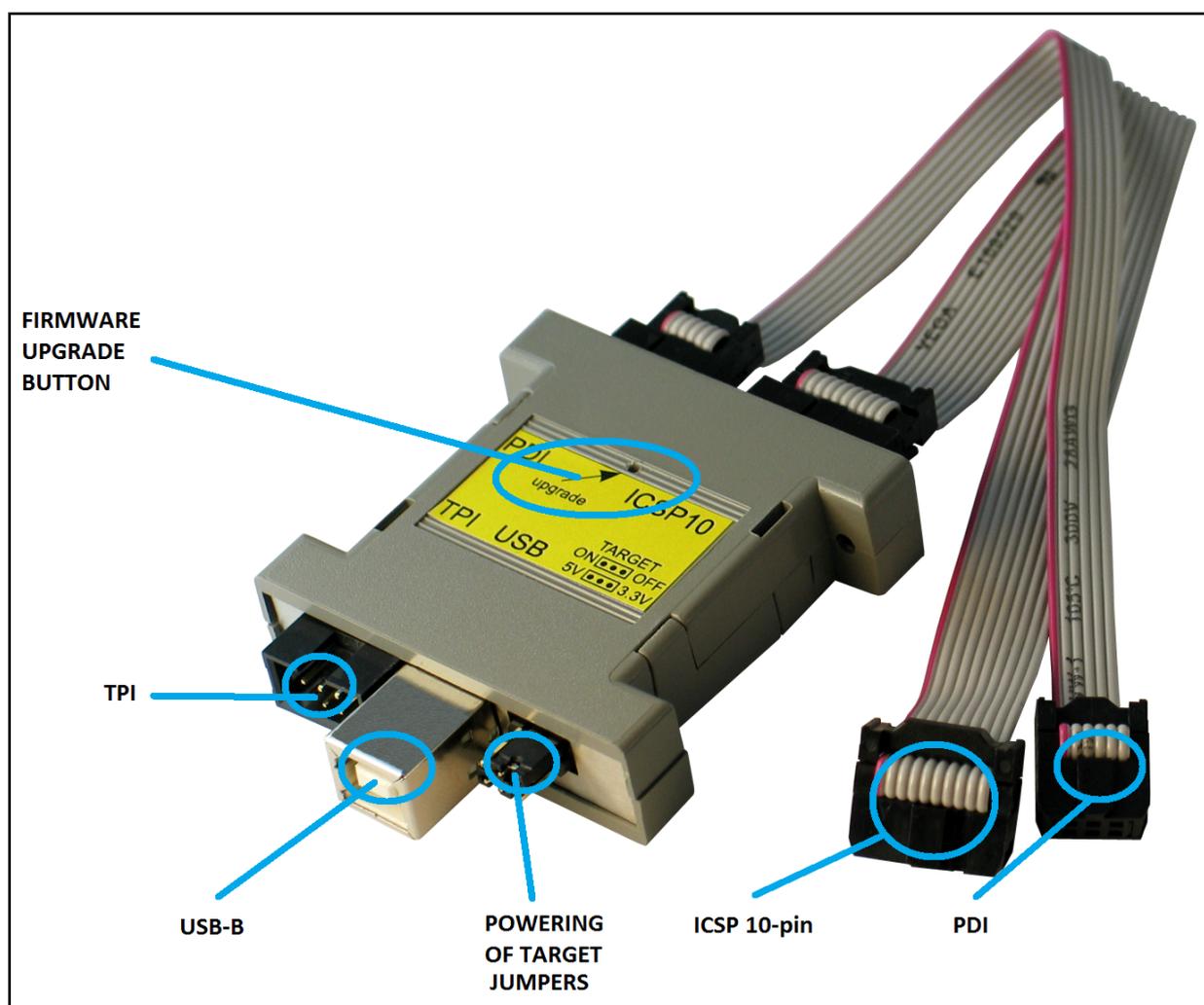
CHAPTER 4 RECOMMENDED SOFTWARE TOOLS

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 in Windows

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

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 sub-chapter 5.2, where firmware changes for AVR-STUDIO are explained.

AVR-ISP-MK2 comes with firmware for AVR-STUDIO in order to use it with AVRDude or Arduino IDE (which uses AVRDude) follow the procedure in next sub-chapter (4.2).

4.2 Arduino, AVRDude, Linux

AVR-ISP-MK2 comes with firmware that supports AVR-Studio. In order to use AVR-ISP-MK2 with AVRDude (GCC, LibUSB, Arduino) you need to perform firmware change.

There is a way to use AVR-ISP-MK2 with Arduino/AVRDude using the LibUSB drivers and changing the firmware BUT it is not possible at the moment to use the AVR-ISP-MK2 both at Arduino/AVRDude and AVR Studio. It would depend on you which one you prefer. You need to change the firmware of AVR-ISP-MK2 (using the Atmel Flip program) 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 our web site and the AVR-ISP-MK2 page.

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.
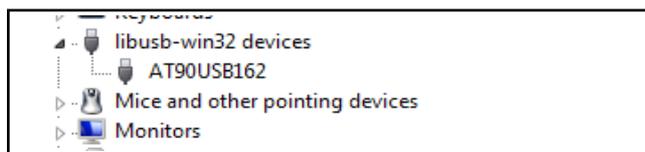
3. After that run the Atmel Flip and select AT90USB162 in Device -> Select
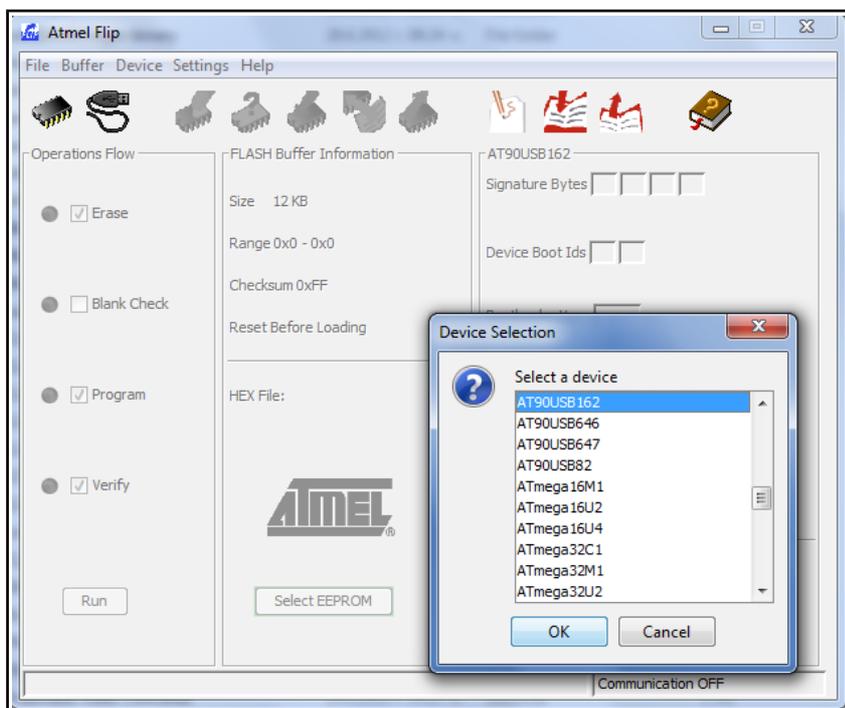
4. Slect Settings -> Communication -> USB -> Open
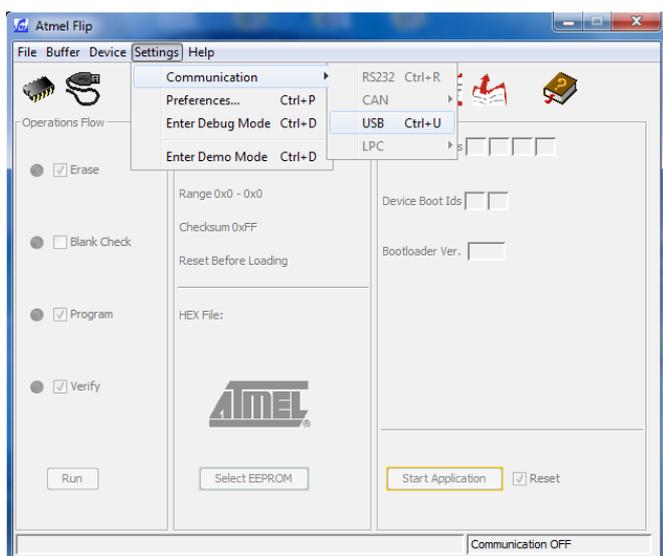
5. From File -> Load hex files choose either AVR456-studio-AVRISP-MKII.hex or libUSB-AVRISP-MKII.hex - the first hex is the firmware for AVR/Atmel Studio, the second one is for AVRdude/Linux/Arduino.

6. Click Run

7. Unplug and plug again AVR-ISP-MK2 from the USB. If you have uploaded the AVRdude/Linux/Arduino firmware the LEDs would change their behavior and remain lit.
8. Uninstall any previous drivers (selecting delete option) from device manager and scan for hardware changes or plug/unplug again. **Note that for the 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".**
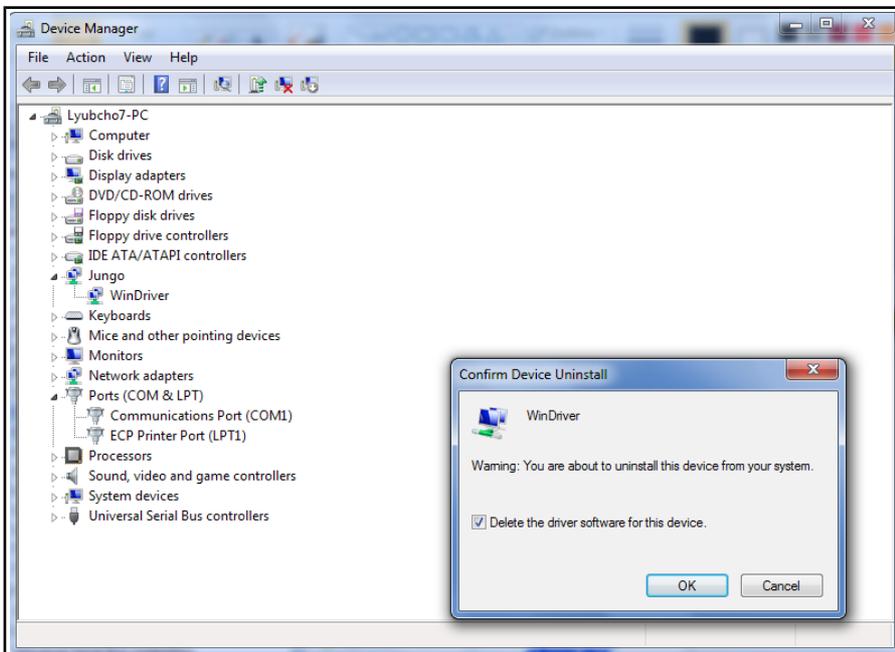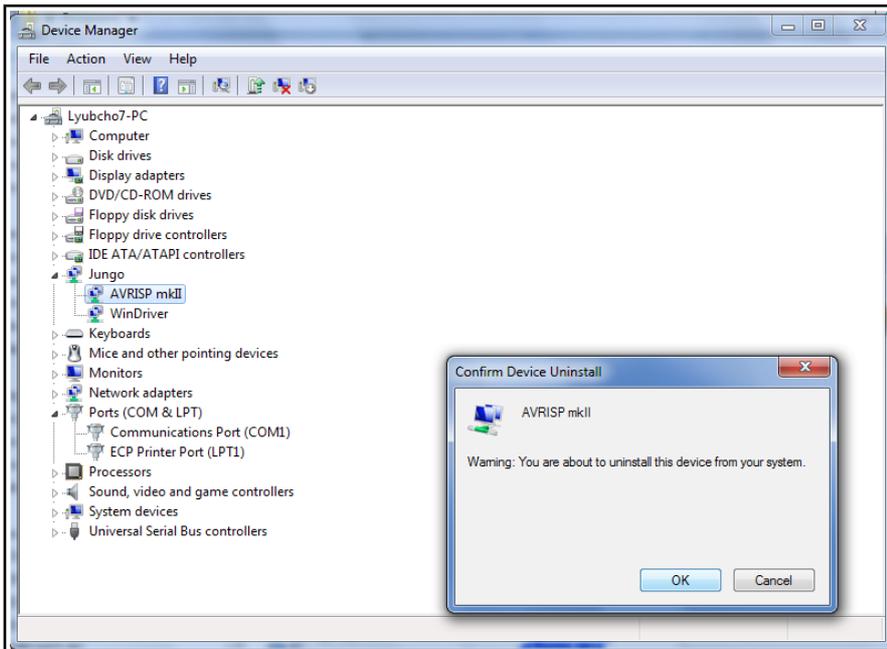
9. Install the new drivers corresponding for your needs (the picture below shows how to install the LibUSB drivers over the Jungo ones – you have to browse to the LibUSB folder and install anyway)
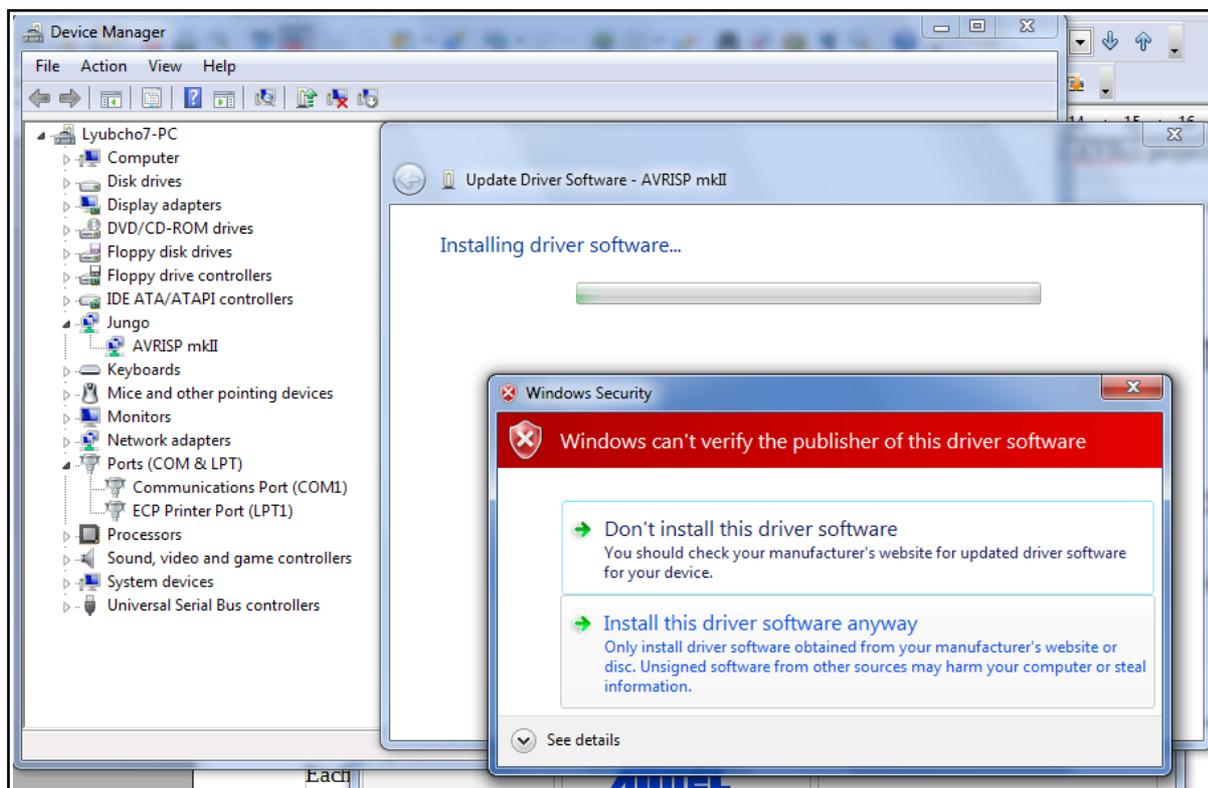
Important: it is possible that the drivers get blocked under Windows 7 x64 or Windows 8. If this is the case when you visit the "device manager" there would be yellow triangle notifying for the error in driver signatures. You would need to "disable driver signing" for your operating system.

4.2.2 Changing firmware for Linux (AVRdude) users:

1. Download and install "dfu-programmer" package (apt-get...).
2. Flash the firmware with:

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

3. 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"
```

4. 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
```

5. Check you're in the dialout group

```
groups
```

6. Restart udev

```
sudo restart udev
```

That's it!

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.
```

4.2.3 Using Arduino IDE with AVR-ISP-MK2

1. Download latest Arduino IDE 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.
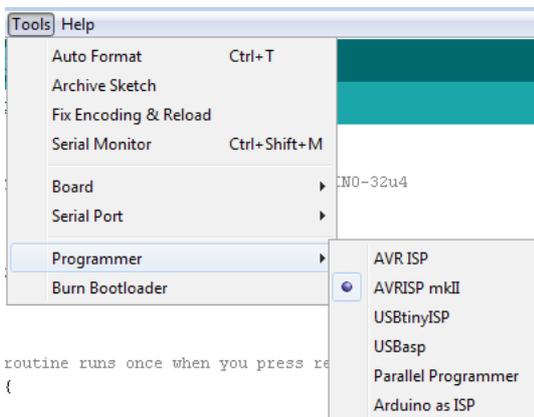
6. Open a simple blinking LED example and adjust the pin values according to the schematic of your device (some boards doesn't require a pin number change).
7. Remember to click "Upload Using Programmer"! By default the right arrow in the interface would attempt to upload via the bootloader!
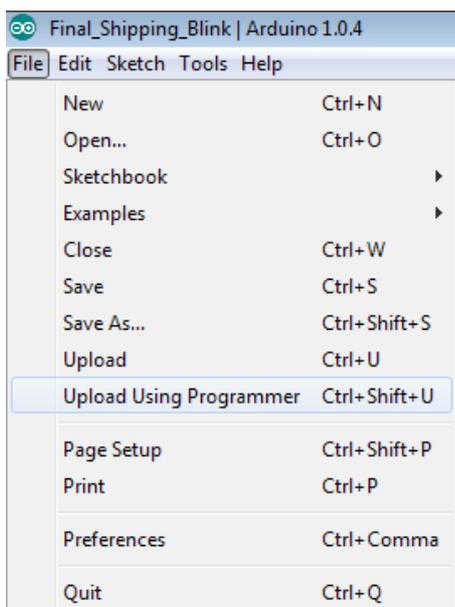
Special mention for Olimex boards – sometimes the board refuses to reset before programming when using Arduino. Try holding the reset button and release it simultaneously while clicking "Upload Using Programmer".

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 make a 10-pin to 6-pin adapter as show below (or just use male-female jumper cables):
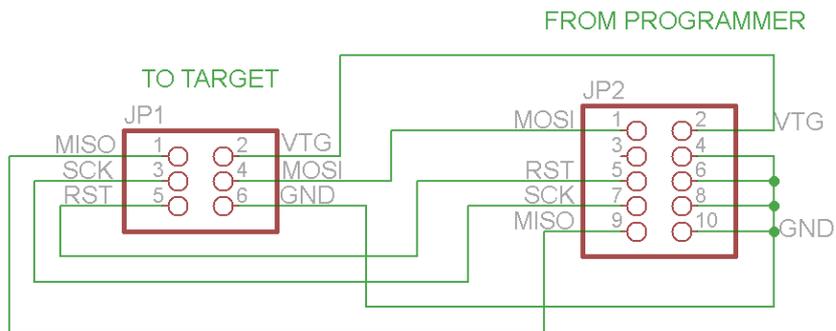
CHAPTER 5 INTERFACES AND HARDWARE

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

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 used for programming tinyAVR.

5.2 Upgrade firmware button

There is a small hole between the PDI interface and the ICSP10 interface which nests a button below. 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 Jumpers description

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

5.3.1 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.

5.3.2 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.

5.4 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 firmware change the LEDs might act in 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

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

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

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

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 AVR studio! What should I do?

A1: The most common ways to resolve such an issue are:
1. Make sure you have installed the proper driver. Sometimes the Jungo driver behaves strange. Try to uninstall it completely from Control panel –> Add/Remove programs (Programs and Features in Windows 7) – the driver is usually named "AVR Jungo USB". Then install it again.

2. When AVR-ISP-MK2 is connected to the USB, in Windows Device Manager you should see a tab named "Jungo". When you click to expand this tab you should see "AVR-ISP mkII" and "WinDriver" devices and there SHOULD NOT be any warnings with exclamation marks!!! If there is exclamation mark this probably means the driver is not enabled because the Third Party drivers enforcement rules in Windows 7 and 8 (please seek more information for how to disable driver enforcements). If the device is recognized as other tool you might need to change the firmware with the Atmel compatible one (which is the default one).

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 ICP 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 6 is just a place holder - it shouldn't be taken as a correct reading!

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.
```