

## AVR-ISP-MK2 programmer USER'S MANUAL

Document revision R, August 2024  
Designed by OLIMEX Ltd, 2012



All boards produced by Olimex LTD are ROHS compliant

## DISCLAIMER

---

© 2024 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](http://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

<b><u>DISCLAIMER.....</u></b>	<b><u>2</u></b>
<b><u>CHAPTER 1 OVERVIEW.....</u></b>	<b><u>5</u></b>
<b><u>1. Introduction to the chapter.....</u></b>	<b><u>5</u></b>
<b><u>1.1 Features.....</u></b>	<b><u>5</u></b>
<b><u>1.2 Target market and purpose of the board.....</u></b>	<b><u>5</u></b>
<b><u>1.3 Document organization.....</u></b>	<b><u>6</u></b>
<b><u>CHAPTER 2 PROGRAMMER DESCRIPTION.....</u></b>	<b><u>7</u></b>
<b><u>2. Introduction to the chapter.....</u></b>	<b><u>7</u></b>
<b><u>2.1 Layout (top view).....</u></b>	<b><u>7</u></b>
<b><u>CHAPTER 3 SETTING UP THE AVR-ISP-MK2 BOARD.....</u></b>	<b><u>8</u></b>
<b><u>3. Introduction to the chapter.....</u></b>	<b><u>8</u></b>
<b><u>3.1 Electrostatic warning.....</u></b>	<b><u>8</u></b>
<b><u>3.2 Requirements.....</u></b>	<b><u>8</u></b>
<b><u>3.3 Quick setup.....</u></b>	<b><u>9</u></b>
<b><u>CHAPTER 4 RECOMMENDED SOFTWARE TOOLS.....</u></b>	<b><u>13</u></b>
<b><u>4. Introduction to the chapter.....</u></b>	<b><u>13</u></b>
<b><u>4.1 Microchip Studio, Atmel Studio, AVR Studio in Windows.....</u></b>	<b><u>13</u></b>
<b><u>4.1.1 Driver installation for AVR Studio in Windows.....</u></b>	<b><u>14</u></b>
<b><u>4.1.2 Manually pointing to new drivers (Windows only).....</u></b>	<b><u>15</u></b>
<b><u>4.1.3 Manually switching between all installed drivers under Windows.....</u></b>	<b><u>15</u></b>
<b><u>4.2 Arduino, AVRDUDE, Linux.....</u></b>	<b><u>16</u></b>
<b><u>4.2.1 Changing firmware for Windows users.....</u></b>	<b><u>16</u></b>
<b><u>4.2.2 Changing firmware for Linux (AVRDUDE) users.....</u></b>	<b><u>19</u></b>
<b><u>4.2.3 Using Arduino IDE with AVR-ISP-MK2.....</u></b>	<b><u>21</u></b>
<b><u>4.2.4 10-pin ICSP to 6-pin ICSP cable.....</u></b>	<b><u>22</u></b>
<b><u>CHAPTER 5 HOW THE LATEST FIRMWARE WAS MADE.....</u></b>	<b><u>23</u></b>
<b><u>CHAPTER 6 INTERFACES AND HARDWARE.....</u></b>	<b><u>25</u></b>
<b><u>6. Introduction to the chapter.....</u></b>	<b><u>25</u></b>
<b><u>6.1 Programming interfaces.....</u></b>	<b><u>25</u></b>
<b><u>6.1.1 10-pin ICSP.....</u></b>	<b><u>25</u></b>
<b><u>6.1.2 6-pin PDI.....</u></b>	<b><u>25</u></b>
<b><u>6.1.3 6-pin TPI.....</u></b>	<b><u>25</u></b>
<b><u>6.2 Upgrade firmware button.....</u></b>	<b><u>25</u></b>
<b><u>6.3 Bootloader restore.....</u></b>	<b><u>26</u></b>
<b><u>6.4 Jumpers description.....</u></b>	<b><u>27</u></b>
<b><u>6.4.1 TARGET jumper.....</u></b>	<b><u>27</u></b>
<b><u>6.4.2 POWER jumper.....</u></b>	<b><u>27</u></b>
<b><u>6.5 LEDs explained.....</u></b>	<b><u>28</u></b>
<b><u>6.6 USB connector.....</u></b>	<b><u>28</u></b>
<b><u>CHAPTER 7 REVISION HISTORY AND SUPPORT.....</u></b>	<b><u>29</u></b>
<b><u>7. Introduction to the chapter.....</u></b>	<b><u>29</u></b>
<b><u>7.1 Document revision table.....</u></b>	<b><u>29</u></b>
<b><u>7.2 Useful web links and purchase codes.....</u></b>	<b><u>30</u></b>
<b><u>7.3 FAQ.....</u></b>	<b><u>31</u></b>
<b><u>7.4 Product support.....</u></b>	<b><u>33</u></b>

---

## 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:

- ◆ Open-source hardware, open-source software product, all design files are available to the customer
- ◆ Suitable for programming AVR, tinyAVR, AVR XMEGA microcontrollers
- ◆ Tested and working with: Microchip Studio 7; Atmel Studio 7; Atmel Studio 6; AVR Studio 5; AVR Studio 4;
- ◆ Tested and working with AVRDUDE 7.3 and Arduino IDE;
- ◆ Test and working under Windows and Linux, reported to be working with Mac OS;
- ◆ Connects to PC via USB type B connector at USB 2.0 full speed;
- ◆ Does not need external power supply as it takes the power supply from the USB port;
- ◆ Uses Atmel's 2×5 pin ICSP, and 2×3 pin PDI, and TPI connector layouts;
- ◆ Works with 5V and 3.3V targets and can supply target with power (jumper selectable);
- ◆ Programs both flash memory and EEPROM;
- ◆ Supports fuses and lock bit programming;
- ◆ Upgrade-able 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 OLIMEX AVR-ISP-MK2 is based on the AVRISPMKII project that is part of 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 contains an explanation on how we made the firmware
- Chapter 6 is an explanation of the interfaces, the LEDs and the jumpers position
- Chapter 7 contains the revision history, useful links and support information

---

## CHAPTER 2 PROGRAMMER DESCRIPTION

---

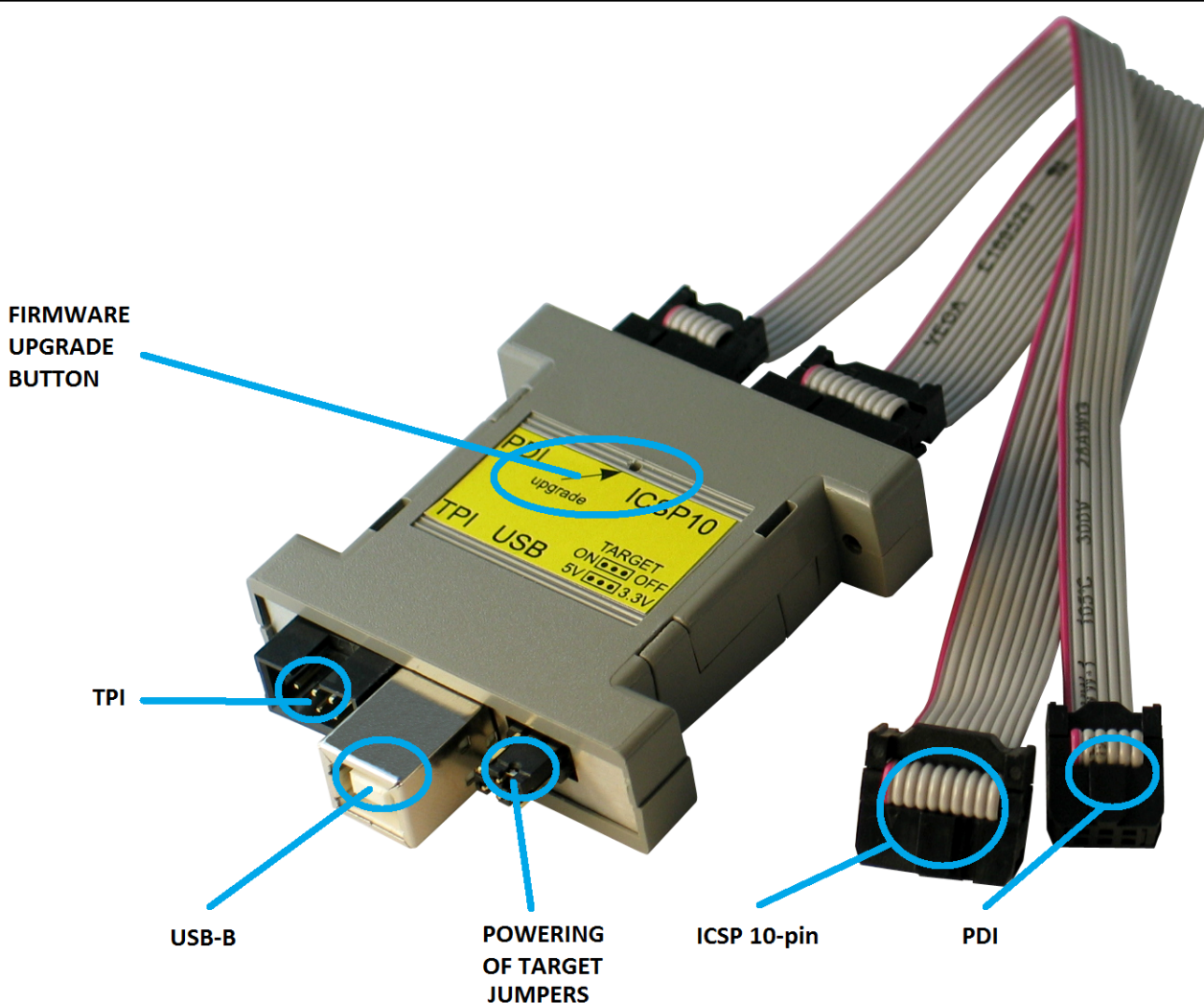
### 2. Introduction to the chapter

---

In this chapter 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.

#### 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 3 SETTING UP THE AVR-ISP-MK2 BOARD

---

### 3. Introduction to the chapter

---

This section helps you set up the AVR-ISP-MK2 programmer 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.

#### 3.1 Electrostatic warning

---

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

#### 3.2 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:**

- A computer with USB port;
- USB cable with at least one USB type B connector and the other end fitting your computer's USB port;
- AVR 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. 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).

##### **Recommended operating systems:**

- any Windows
- any frequently updated Linux distribution
- any Mac OS/OSX

##### **Recommended software tools:**

- Microchip Studio 7
- AVRDUDE 7.0 or newer
- Arduino IDE



### 3.3 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 Microchip Studio 7, AVRDUDE v7.3 compiled for Windows, and Arduino IDE 1.8.19 (which seems to use AVRDUDE v6.3.0).

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.

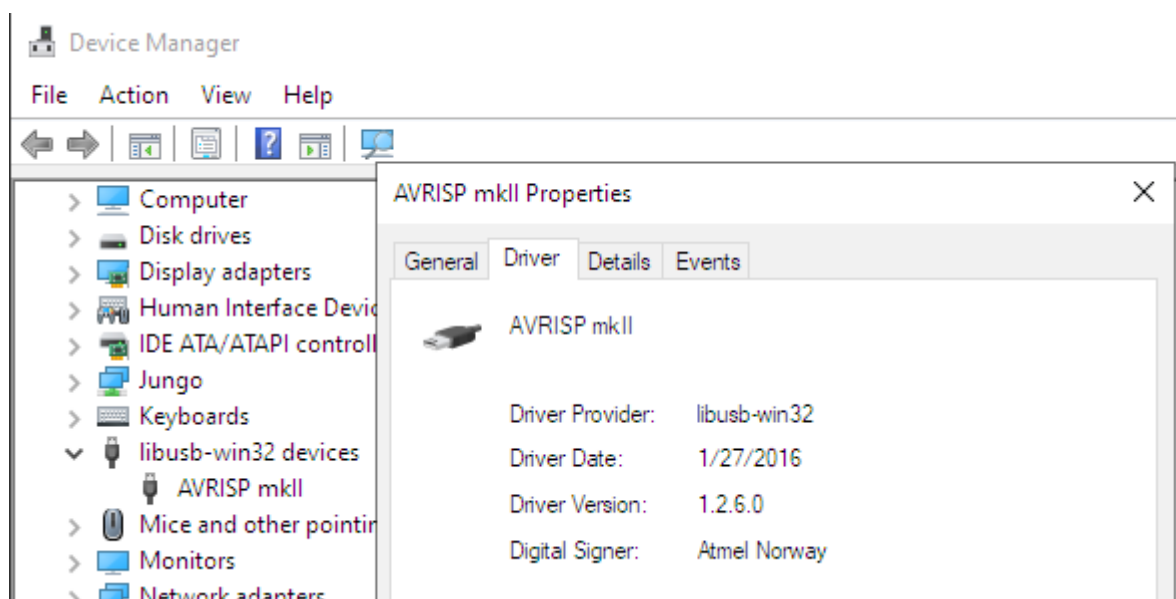
**Step 1:** Download and extract the drivers for AVR-ISP-MK2. These can be downloaded from the following location:

<https://www.olimex.com/Products/AVR/Programmers/AVR-ISP-MK2/resources/DRIVER-MK2.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 “libusb-win32 devices” group. Confirm that the entry for AVR-ISP-MK2 looks like the picture below:



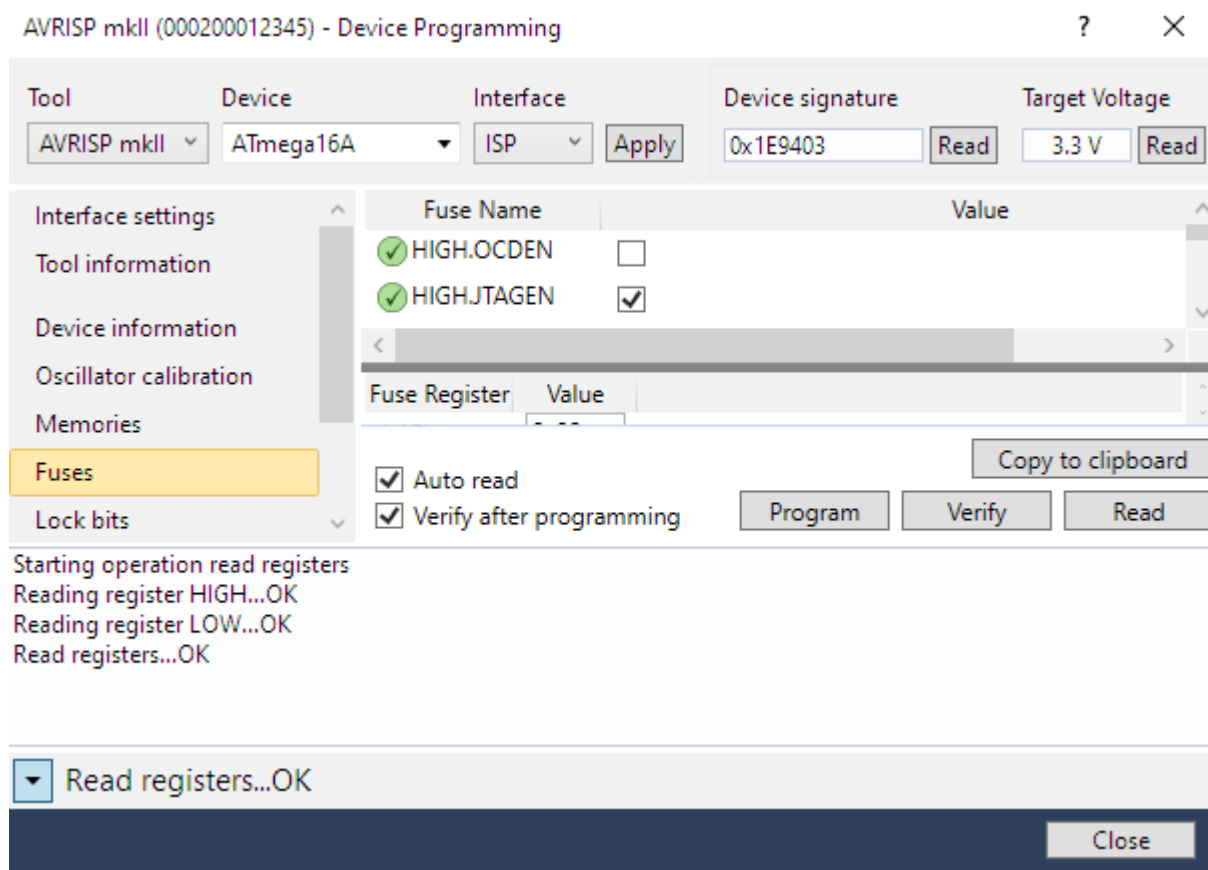
If there are problems with the drivers, notice that sometimes Windows automatically assigns improper drivers to the unit. You would need to manually switch to the proper drivers. Refer to chapter “4.1.2 Manually pointing to new drivers”.

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 – Microchip Studio 7; AVRDUDE 7.3; Arduino IDE.

**(Optional) Step 5:** Visit Microchip's web-site and download the latest Microchip Studio for AVR and SAM device. At the time of writing this document, the web page to download Micorchip Studio is this one:

<https://www.microchip.com/en-us/tools-resources/develop/microchip-studio#Downloads>

Install Microchip 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 Microchip 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:

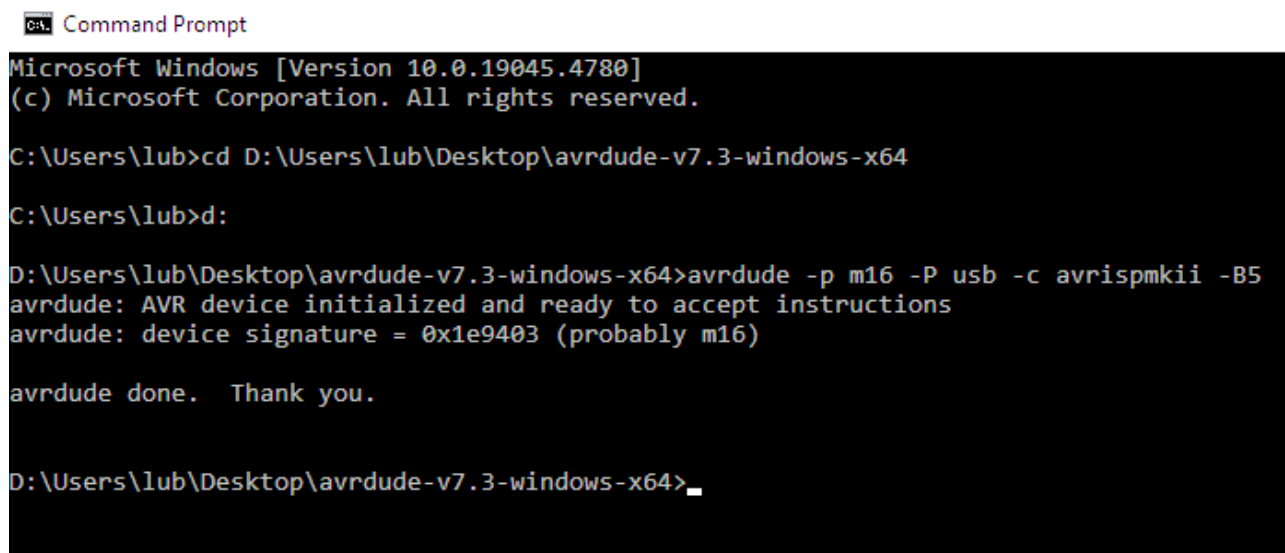
<https://github.com/avrdudes/avrdude/releases>

For the Windows test listed below we used avrdude-v7.3-windows-x64.zip that was uploaded on 7<sup>th</sup> of February 2024. We also used our board MOD-IO that has an ATmega16A chip.

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:



```
Command Prompt
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lub>cd D:\Users\lub\Desktop\avrdude-v7.3-windows-x64

C:\Users\lub>d:

D:\Users\lub\Desktop\avrdude-v7.3-windows-x64>avrdude -p m16 -P usb -c avrispmkii -B5
avrdude: AVR device initialized and ready to accept instructions
avrdude: device signature = 0x1e9403 (probably m16)

avrdude done. Thank you.

D:\Users\lub\Desktop\avrdude-v7.3-windows-x64>
```

I can also program my target with the following:

```
avrdude -u -c avrispmkII -p m16a -B 115200 -U flash:w:MOD-IO_prebuilt.elf
```

At this point you have AVR-ISP-MK2 properly working with AVRDUDE v7.3 and also with Microchip Studio 7. If you wish to also test the Arduino IDE connection continue with the next optional step.

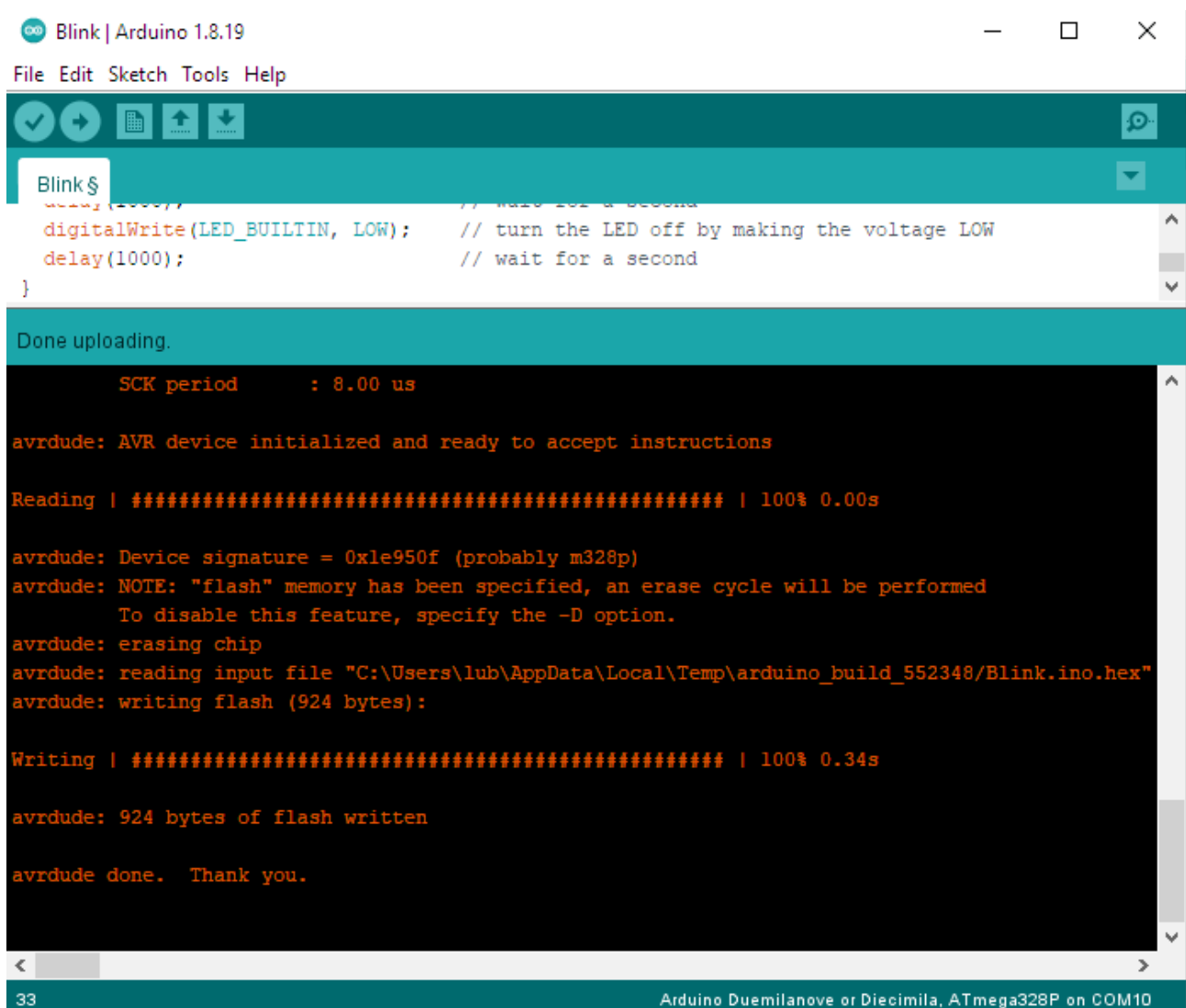
**(Optional) Step 7.** Download the install the latest version of Arduino IDE from the page below:

<https://www.arduino.cc/en/software>

For the Windows test listed below we used Arduino version 1.8.19 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 below:



The screenshot shows the Arduino IDE window titled "Blink | Arduino 1.8.19". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar contains icons for running, saving, and uploading. The sketch editor shows the following code:

```
Blink $
// turns on an LED connected to digital pin 13
// using the delay() function
// make LED a constant
const int LED_BUILTIN = 13; // the LED built in to the board
digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
delay(1000); // wait for a second
}
```

The serial monitor displays the following output:

```
Done uploading.
SCK period      : 8.00 us
avrdude: AVR device initialized and ready to accept instructions
Reading | ##### | 100% 0.00s
avrdude: Device signature = 0x1e950f (probably m328p)
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 "C:\Users\lub\AppData\Local\Temp\arduino_build_552348/Blink.ino.hex"
avrdude: writing flash (924 bytes):
Writing | ##### | 100% 0.34s
avrdude: 924 bytes of flash written
avrdude done.  Thank you.
```

The status bar at the bottom shows "33" and "Arduino Duemilanove or Diecimila, ATmega328P on COM10".

---

## 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 Microchip Studio, Atmel Studio, AVR Studio in Windows

---

The most popular choice among 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 Microchip website.

AVR-ISP-MK2 comes with firmware compatible with Microchip Studio, AVRDUDE, and Arduino. 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 “6.2 Upgrade firmware button”

**Important – the drivers for the old AVR Studio and Microchip/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 – Jungo Windows drivers for AVR Studio and the LibUSB Windows drivers for AVRDUDE are incompatible (but the drivers for Atmel Studio and AVRDUDE are the same if you are using newer Windows so if you are using the latest releases you are already set).

The default way of installing the drivers for Microchip Studio, AVRDUDE, and Arduino under Windows is available in the previous chapter “3.3 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.

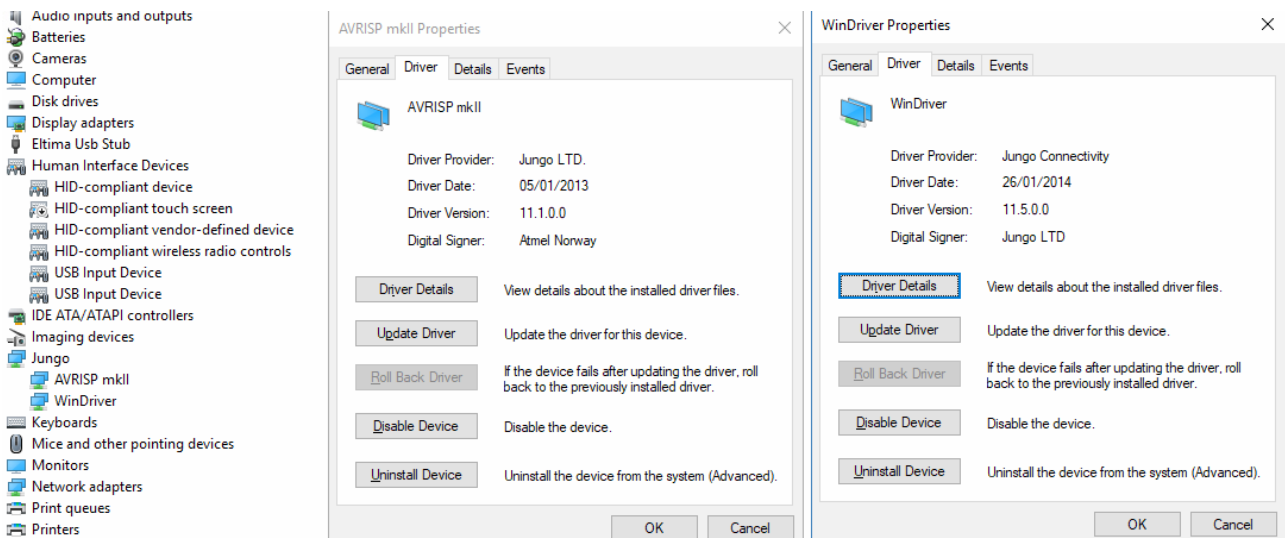
If you wish to use AVR Studio 4 or 5 make sure to use the installer provided in the archive with the firmware (usually named “driver-atmel-bundle-x.x.xxx.exe). Only these drivers are compatible but you might need extra steps in certain cases or under certain newer versions of Windows.

The author of this document managed to get AVR Studio 4 working under Windows 10 only on a clean Windows 10 installation (that never had any newer Studio installed, e.g. Atmel Studio 6 or Atmel Studio 7). After AVR Studio 4 had been installed, it was safe to install newer studio and using one or the other required manual driver switching. If you installed newer studio first, and then AVR Studio 4, you would probably need to re-install Windows (or find some other workaround, like registry magic) since driver switching was insufficient to get it working.

If you are using Windows 10 x64, and you want to use AVR Studio 4 or 5 (notice that this is not Atmel Studio 7 or newer) consider the following driver instillation advice from a customer (thank you, James):

On a new, clean (!) Windows 10 x64 machine:

1. Install AVR Studio.
2. Install WinAVR-20100110 (popular compiler, required to compile many project).
3. Overwrite msys-1.0.dll with 64-bit version (else compiler doesn't work).
4. Connect AVR-ISP-MK2 to the computer. Windows would install some driver automatically and show a message that device installed was installed successfully. This is a lie, it wasn't.
4. Disable Windows driver signature enforcement.
5. Manually point to the appropriate driver that AVR Studio installed (refer to “4.1.3 Manually switching between all installed drivers under Windows”). You might also use the driver-atmel-bundle-7.0.888.exe from AVR-ISP-MK2-Firmware-WindowsDrivers. The two entries should look similar to the picture below (it is ok if the WinDriver is older version):



### 4.1.2 Manually pointing to new drivers (Windows only)

Sometimes Windows automatically assigns improper drivers to the unit. You would need to manually switch to the proper drivers.

1. Make sure that you have downloaded and extracted the drivers provided for AVR-ISP-MK2 available here:

<https://www.olimex.com/Products/AVR/Programmers/AVR-ISP-MK2/resources/DRIVER-MK2.zip>

2. Connect AVR-ISP-MK2 to the personal computer.
3. You should now have some entry for it in “Windows Device Manager”. If you can't identify the entry unplug and re-plug the AVR-ISP-MK2 until you are sure. It might be good idea to uninstall any drivers related to the tool, but in most cases if you do everything correctly this shouldn't matter.
4. Right-click over the entry for the cable.
5. Select “Update Driver Software”.
6. Select “Browse my computer for driver software”.
7. Select the bottom option “Let me pick from a list of drivers on my computer”.
8. Click "Have Disk" and point to the place where you extracted the drivers from the archive and select "AVRISP\_mkII.inf" then click "Open".
9. Select "Next", and if a warning pops up select “Install Anyway” and after installation close the window.
10. Inspect the new entry in “Windows Device Manager” after the driver installation and ensure that there are no errors or warnings.
11. Test.

### 4.1.3 Manually switching between all installed drivers under Windows

While the unit can be associated only with one specific set of drivers at specific time; there can be multiple different drivers installed on the system. How to manually switch between different sets of drivers available for AVR-ISP-MK2:

1. First make sure that you have disabled the automatic Windows driver update. It can take control of driver installation before you can react. In Windows 7 search for "Device Installation Settings". Else when you install the working driver it might get overwritten by another one that Windows thinks would work better.
2. Also make sure that “Driver Signature Enforcement” is disabled before trying to change the drivers (since some drivers might be unsigned). There are a lot of instructions online on how to do it (it requires a restart and entering the Windows boot menu with F8 usually in older Windows releases and entering advanced startup options in newer Windows releases).
3. Connect AVR-ISP-MK2 to the personal computer.
4. Right-click over the AVR-ISP-MK2 entry (in your case probably “libusb-win32” driver) in “Windows Device Manager”.
5. Select “Update Driver Software...”.
6. Select “Browse my computer for driver software”.
7. Select the bottom option “Let me pick from a list of drivers on my computer”.
8. Pick one of the driver entries in the list under “Model”, if there are more then one start with the first (top-most) and finish the installation (steps 9 and 10), then test with your software and if it still doesn't work start all over from point 1. but this time select the second entry in the “Model” list,

until you exhaust all entries under “Model” one by one!

9. Select next, and if a warning pops up select “Install Anyway” and after installation close the window.

10. Inspect the new entry in “Windows Device Manager” after the driver installation and ensure that there are no errors or warnings.

11. Test.

## 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 Microchip Studio without changing the firmware or the drivers.

It is impossible to use the AVR-ISP-MK2 simultaneously with 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?

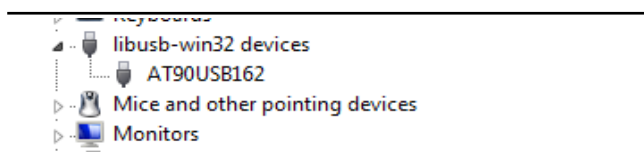
### 4.2.1 Changing firmware for Windows users

All files needed except for Flip (which is available at Microchip's web-site for free) should be downloaded from the official AVR-ISP-MK2 page at our web-site. They can be found in this archive:

<https://www.olimex.com/Products/AVR/Programmers/AVR-ISP-MK2/resources/DRIVER-MK2.zip>

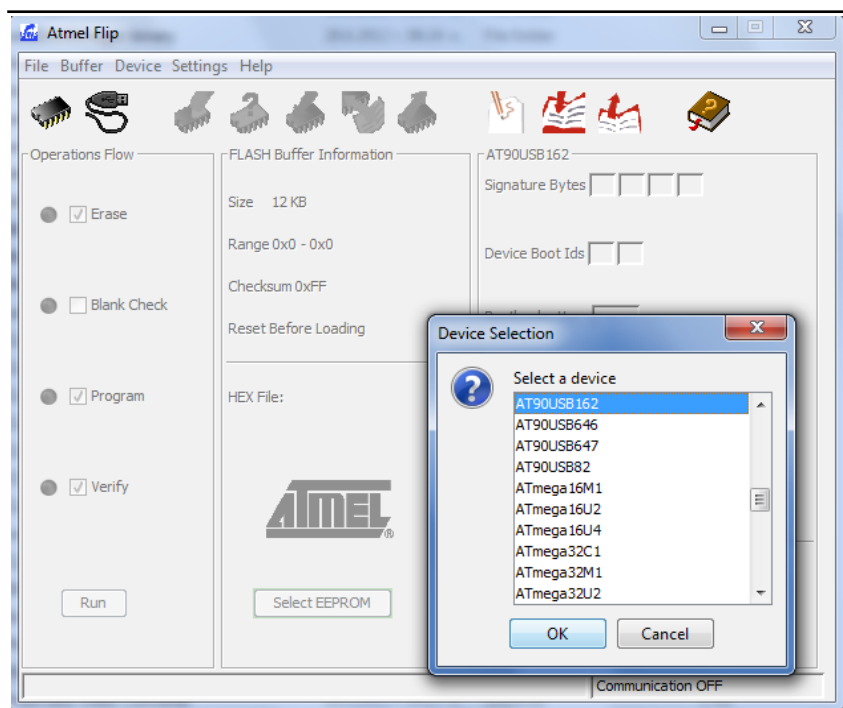
1. Connect the device to the USB and with a needle or sharp object press the button on it's back side (the one that the arrow points as "Upgrade"; the LEDs should turn off) which will put the processor in bootloader state – ready for firmware upgrade.

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.

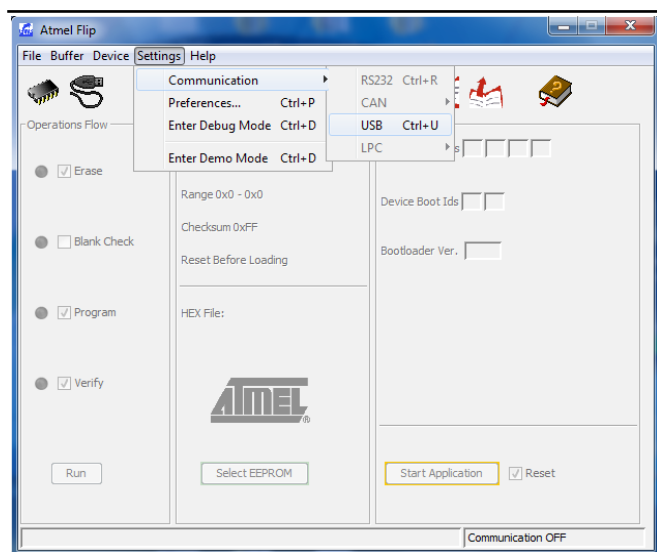


3. After that run the Flip and select AT90USB162 in Device -> Select





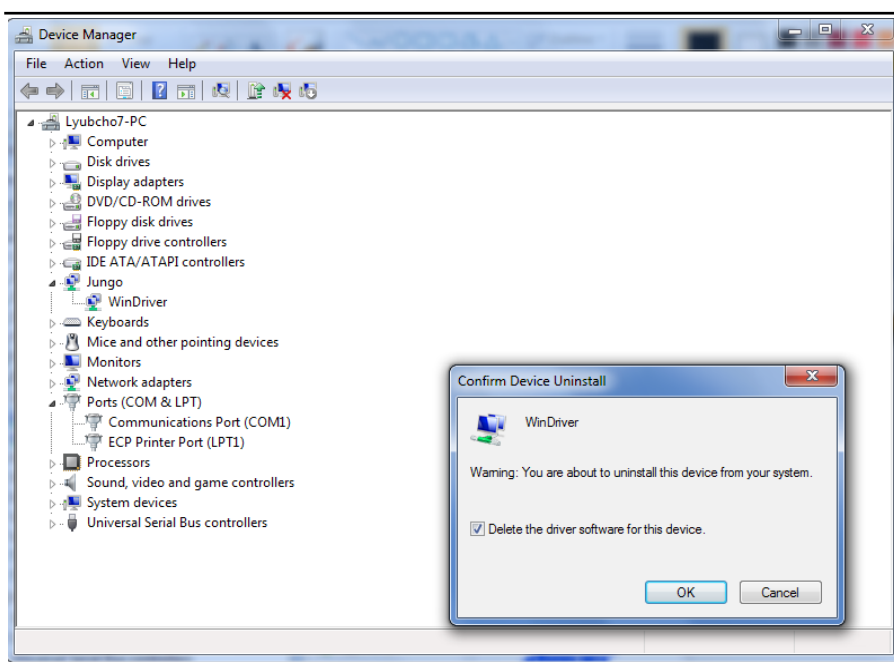
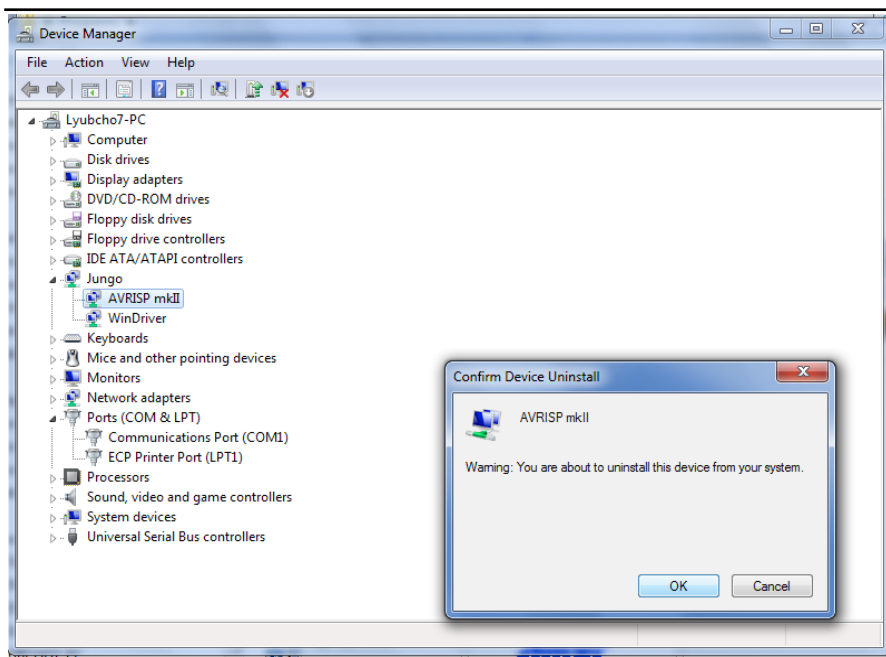
#### 4. Select Settings -> Communication -> USB -> Open



5. From File -> Load hex files choose OLIMEX-AVR-ISP-MK2.hex – the latest and greatest firmware based on latest LUFA, compatible with LibUSB driver, if you want to switch to older versions of the firmware browse the “OLDER-VERSIONS” folder.

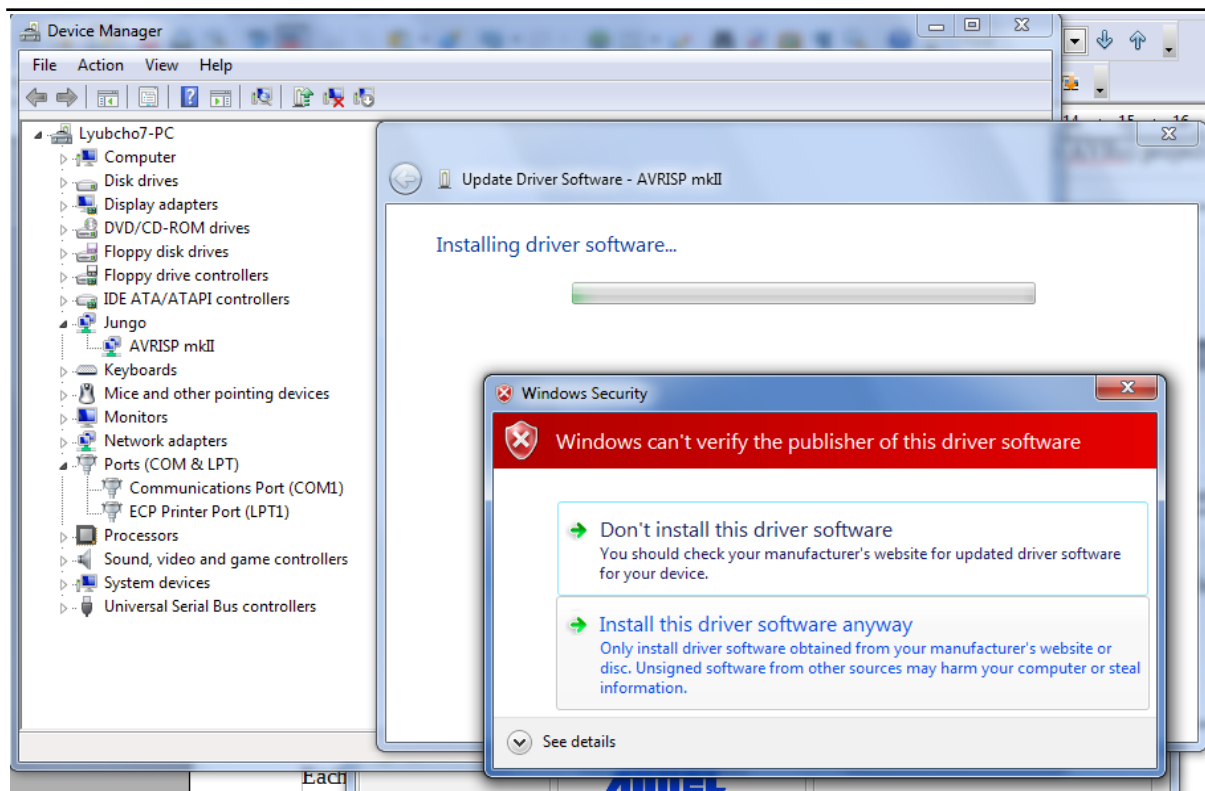
6. Click “Run” and after it completes, unplug and plug again AVR-ISP-MK2 from the USB.

7. 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 next picture 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, Windows 10, or Windows 11. 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 now test the connection:

```
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-win32>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 avrispkmII -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.

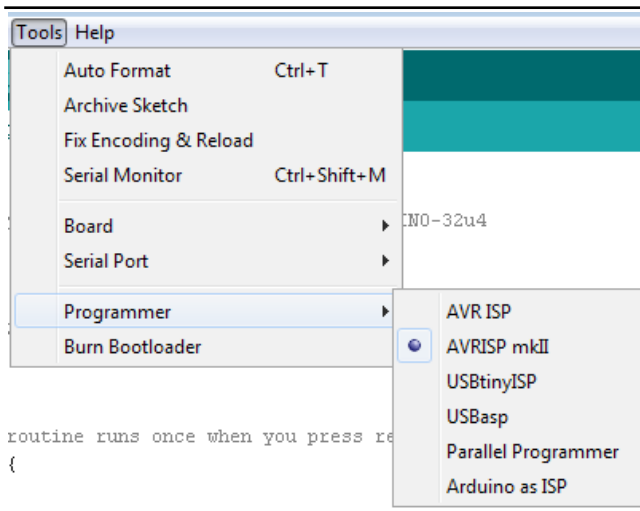
```

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 – TARGET 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.

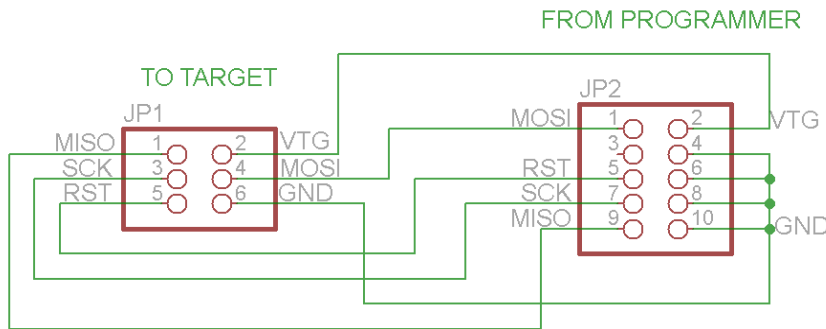


- 6. Open a simple blinking LED example and adjust the pin values according to the schematic of your device (some boards doesn't require a pin number change).
- 7. Remember to click “Upload Using Programmer”! By default the right arrow in the interface would attempt to upload via the bootloader!

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

10-pin ICSP is the most common connector but some boards have only 6-pin ICSP. In such a case you should either get the affordable adapter [AVR-ICSP](#) that we manufacture or make a 10-pin to 6-pin adapter as show below (or just use male-female jumper cables):



---

## CHAPTER 5 HOW THE LATEST FIRMWARE WAS MADE

---

We download an unmodified archive with LUFA-210130 as basis. What we do with LUFA-210130 – change configuration in two files to fit our hardware. Then we recompile with WinAVR-20100110 and AVR Studio 4 (but if you are experienced you can use other compiler and environment, as long as you do the minor configuration changes that we do). We program the resulting hex via FLIP to an AVR-ISP-MK2 target (the same hex distribute to customers to update their units) and then read the target to create an ELF that is used to program all boards. The changes we make to LUFA-210130 are:

- in file `\lufa-master\lufa-master\Projects\AVRISP-MKII\makefile` and change the first few lines to:

```
MCU = at90usb162
ARCH = AVR8
BOARD = OLIMEXISPMK2
F_CPU = 16000000
OPTIMIZATION = s
TARGET = AVRISP-MKII
```

Save the changes.

-in file `\lufa-master\lufa-master\Projects\AVRISP-MKII\Lib\V2ProtocolParams.h` and change:

```
#define FIRMWARE_VERSION_MINOR 0x29
```

Change the number to a bigger one. If it is lower than the current one used in Studio it might not work.

Save the changes.

Either use your favorite compiler and environment to compile the project (make sure to use the make file that we edited above). Here we do the following:

1. Open AVR Studio 4 and create a new blank project. Select AVR GCC from the left box. Remove the tick from the blank .c file. **IMPORTANT - NAME THE PROJECT "AVRISP-MKII" - or compilation errors would occur (elf not found)!!!**
2. Open the project properties/configuration and in the general tab point to the makefile we edited earlier (`\lufa-master\Projects\AVRISP-MKII\makefile`). Save. Do not change anything else (it would not be taken into account anyway). (don't forget to point WinAVR to the `avr-gcc.exe` and to the `make.exe`! - these are in the place where you installed WinAVR-20100110 – one in subfolder “bin” the other in sub-folder “utils”).
3. Clean and rebuild the project. If there are issues notice that WinAVR-20100110 might require a patch to work under Windows 8.1, Windows 10, and Windows 11. Search online for a file to replace `msys-1.0.dll` in `\WinAVR-20100110\utils\bin`.

4. If build is successful, get the hex for Atmel upload from the main folder (\lufa-master\lufa-master\Projects\AVRISP-MKII\)) and upload it to the AVR-ISP-MK2 using Flip (remember to put board in bootloader mode via the button) – or under Linux you can use dfu-programmer.



## CHAPTER 6 INTERFACES AND HARDWARE

---

### 6. 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.

#### 6.1 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.

##### 6.1.1 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 [AVR-ICSP](#) adapter since AVR-ISP-MK2 doesn't provide 6-pin ICSP connector. Note that it is bought separately.

##### 6.1.2 6-pin PDI

---

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

##### 6.1.3 6-pin TPI

---

The 6-pin TPI interface is typically used for programming tinyAVR.

### 6.2 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 "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 "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.

## 6.3 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 "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 “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.

## **6.4 Jumpers description**

---

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

### **6.4.1 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.

### **6.4.2 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.

## 6.5 LEDs explained

---

There are two 3mm LEDs – an ORANGE LED and a double colored GREEN/RED LED. Using latest firmware the behavior is the following:

Red – turns on and stays on after successful USB initialization;

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

During firmware update (after pressing the button) – both LEDs remain turned off.

**Using different firmware might alter the behavior of the LEDs. The LEDs behavior shouldn't be used as sign for hardware problem.**

## 6.6 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 7 REVISION HISTORY AND SUPPORT

### 7. 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.

### 7.1 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
O and P, 23.02.18	Updated several links, added manual driver switch details, added information about AVR Studio 4 under Windows 10 x64	14, 15
R, 23.08.24	Major clean up due to LUFA-210130 update	All

## 7.2 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:

<https://www.microchip.com/en-us/development-tool/atavrisp2>

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.**

## 7.3 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, Atmel, Micorchip Studios require different drivers. Atmel/Microchip 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 Micorchip 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 Microchip 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: Yes. 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: Previously I used AVR Studio 4 successfully but when I installed Microchip Studio 7 I can no longer connect to AVR-ISP-MK2 in AVR Studio 4. What can I do to use AVR Studio 4 and Microchip Studio 7 at the same time?**

A7: The drivers installed with Microchip 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 Microchip workaround can be found here:

<https://microchip.my.site.com/s/article/Downgrading-tools-to-use-older-Jungo-driver>

**Q8: This programmer is unrecognized in Microchip/Atmel Studio 7.x. What should I do in order to get it recognized?**

A8: - If you get error with message "MPLABCOMM\_INVALID\_HANDLE" - you can fix it by navigating in the menus of Atmel Studio. Go to Tools -> Options -> Tools -> Tool Settings -> Check Firmware -> False. (shown here: <https://i.imgur.com/XWjed2c.jpg>)

- Update the firmware. Make sure that the suitable drivers for Microchip Studio are installed.

To update the firmware please, follow the steps listed above in this document.

**Q9: OSCCAL doesn't seem to be working. What should I do?**

A9: This feature was added in the 2024 firmware update. Make sure to update the firmware of your advice.



## 7.4 Product support

---

For product support, hardware information and error reports mail to: [support@olimex.com](mailto: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 anti-static package and well packed to prevent damages during the transport.