



USBasp AVR Programmer

User Guide

Version 2.0

January 2017

TF1105PC1606PR02J

Table of Contents

1. Overview	2
1.1. Introduction and Features	2
1.2. References	2
1.2.1. Referenced Web Pages	2
1.2.2. Acronyms and Abbreviations	3
1.3. Supported Microcontrollers	4
1.4. Warnings	5
2. Layout	5
2.1. USB Type B	6
2.2. JP1 Target Connector	6
2.3. JP2 Slow SCK	6
2.4. JP3 Self Program	6
2.5. LEDs	6
3. Using the Programmer	6
3.1. Connecting the Programmer to your computer	6
3.1.1. Windows 7 (32 and 64 Bit)	7
3.1.2. Windows Vista (32 Bit)	10
3.1.3. Windows XP (32 bit)	13
3.1.4. Other Operating Systems	16
3.2. Downloading Firmware to your Microcontroller	16
3.2.1. Required Items	16
3.2.2. Assumptions	17
3.2.3. Procedure	17
3.2.4. More Information	18
4. Comment and Questions	20

1. Overview

1.1. Introduction and Features

USBasp is a USB in-circuit Programmer for Atmel AVR controllers. It simply consists of an ATmega8 and a couple of passive components. The Programmer uses a Firmware-only USB driver, no special USB controller is needed. Some of the key features include:

- a) Works under multiple platforms, Linux, Mac OS X and Windows.
- b) Allows you to read or write the microcontroller EEPROM, firmware, fuse bits and lock bits.
- c) Programming speed is up to 5kBytes/sec.
- d) Hardware and software controlled SCK (Serial Clock).
- e) Slow SCK (Serial Clock) option to support targets with low clock speed (< 1.5MHz).
- f) Indicator LED for normal operation and busy activity.
- g) Male pin connector interface for target connecting flexibility.
- h) Self programming option for updating firmware.

1.2. References

1.2.1. Referenced Web Pages

The web pages referenced in this User Guide are listed in Table 1.

Name	Address
USBasp Programmer for Atmel AVR Controllers	http://www.fischl.de/usbasp/
AVRdude	http://savannah.nongnu.org/projects/avrdude/
AVRdude Documentation	http://www.nongnu.org/avrdude/usermanual/avrdude.html
WinAVR	http://winavr.sourceforge.net/

Table 1. Referenced Documents

1.2.2. Acronyms and Abbreviations

The acronyms and abbreviations utilized in this User Guide are listed in Table 2.

Acronym and Abbreviation	Description
AVR	According to Atmel, AVR stands for nothing, it's just a name. Others say it stands for Advanced Virtual RISC. However, the inventors of the AVR series chips are named Alf Egil Bogen and Vegard Wollan, so you be the judge.
IDC	Insulation Displacement Connector
ISP	In System Programmer
LED	Light Emitting Diode
RISC	Reduced Instruction Set Computing
USB	Universal Serial Bus
SCK	Serial Clock
MISO	Master In Slave Out
MOSI	Master Out Slave In
RST	Reset
VCC	Positive Supply Voltage
GND	Ground
SIL/DIP	Single in Line/Dual In-line Package
SPI	Serial Peripheral Interface
IDE	Integrated Development Environment
MCU	Microcontroller Unit
Hz/KHz/MHz	Hertz/Kilo Hertz/Mega Hertz
XTAL	Crystal Oscillator
PCB	Printed Circuit Board
I/O	Input and Output
HVPP	High-Voltage Parallel Programming

Table 2. Acronyms and Abbreviations

1.3. Supported Microcontrollers

Lists of the Microcontrollers that are supported by the USBasp Programmer.

Mega Series				
ATmega8	ATmega8A	ATmega48	ATmega48A	ATmega48P
ATmega48PA	ATmega88	ATmega88A	ATmega88P	ATmega88PA
ATmega168	ATmega168A	ATmega168P	ATmega168PA	ATmega328
ATmega328P	ATmega103	ATmega128	ATmega128P	ATmega1280
ATmega1281	ATmega16	ATmega16A	ATmega161	ATmega162
ATmega163	ATmega164	ATmega164A	ATmega164P	ATmega164PA
ATmega169	ATmega169A	ATmega169P	ATmega169PA	ATmega2560
ATmega2561	ATmega32	ATmega32A	ATmega324	ATmega324A
ATmega324P	ATmega324PA	ATmega329	ATmega329A	ATmega329P
ATmega329PA	ATmega3290	ATmega3290A	ATmega3290P	ATmega64
ATmega64A	ATmega640	ATmega644	ATmega644A	ATmega644P
ATmega644PA	ATmega649	ATmega649A	ATmega649P	ATmega6490
ATmega6490A	ATmega6490P	ATmega8515	ATmega8535	
Tiny Series				
ATtiny12	ATtiny13	ATtiny13A	ATtiny15	ATtiny25
ATtiny26	ATtiny45	ATtiny85	ATtiny2313	ATtiny2313A
Classic Series				
AT90S1200	AT90S2313	AT90S2333	AT90S2343	AT90S4414
AT90S4433	AT90S4434	AT90S8515	AT90S8535	
Can Series				
AT90CAN128				
PWN Series				
AT90PWM2	AT90PWM3			

Table 3. Supported Microcontrollers

1.4. Warnings



Some of the components discussed in this document are very sensitive to electrical static discharges. The reader should take precautions to ensure that components are protected against these discharges.



Whilst the voltages typically seen in microcontroller circuits are low, the reader should be aware of the risk of working with electrical circuits and take necessary precautions.



Supply target with 5V (from USB voltage). Be careful with this option, the device isn't protected against short circuit. Please avoid short circuit to prevent damage this device and also your computer. Please be careful.

2. Layout

The layout of the USBasp Programmer is shown in Figure 1.

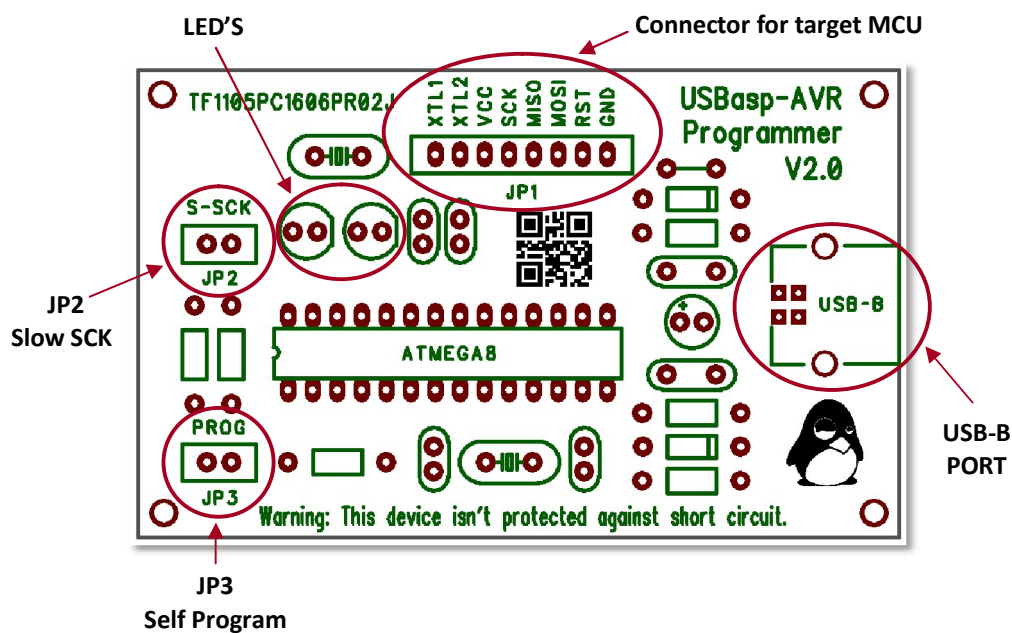


Figure 1. Device Layout

2.1. USB Type B

The USB end of the Programmer connects directly into your computers USB port.

2.2. JP1 Target Connector

The 8 pin male connector provides an interface to connect the target microcontroller.

2.3. JP2 Slow SCK

When this jumper is selected or closed, the slow clock mode is enabled. If the target clock is lower than 1.5 MHz, you need to set this jumper. Then SCK is scaled down from 375 kHz to about 8 kHz.

2.4. JP3 Self Program

This jumper is used to update the firmware of the USBasp Programmer. In order to update the firmware you will need 2 Programmers. One to be programmed and the other to do the programming. Set JP2 open and JP3 close mode for updating firmware.

2.5. LEDs

The USBasp Programmer has 2 LEDs near the target (JP1) connector. These have the following functions:

- a) LED RED (RW) Read/Write and Programmer communicating with target microcontroller.
- b) LED GREEN (PW) Power Indicator.

3. Using the Programmer

3.1. Connecting the Programmer to your computer

Connecting the Programmer to your computer comprises of 2 steps:

- a) Physically connecting the Programmer to the USB port, and
- b) Windows requires a driver for USBasp. Installing drivers in order for it to work. Whilst the USBasp Programmer will work on a wide variety of operating systems, this procedure will focus on Windows 7, Vista 32 bit and Windows XP. Please follow these instructions.

3.1.1. Windows 7 (32 and 64 Bit)

3.1.1.1. Required items

Items required to run this procedure are: a) USBasp Programmer. b) Computer with USB port and Windows 7 installed. c) Drivers and Applications DVD.

3.1.1.2. Assumptions

This procedure assumes that:

a) The logged in user has sufficient privileges to install device drivers.

3.1.1.3. Procedure

To install the USBasp Programmer:

a) Insert the Programmer into an available USB port

b) A message will appear in the system tray, as shown below. Click on “Click here for status” to see what is going on.

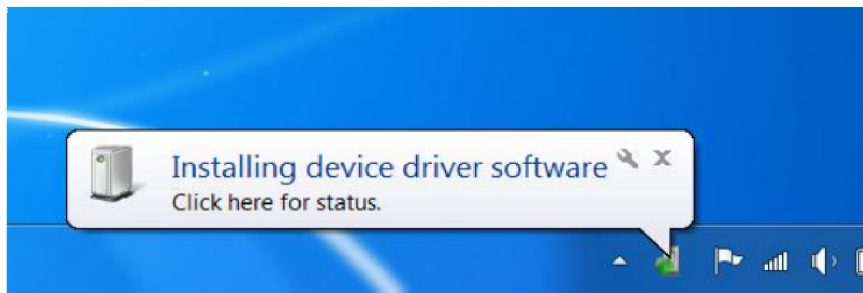


Figure 5. Driver Installation on Windows 7 – System Tray Message

c) Windows will attempt to install a driver from Windows Update.

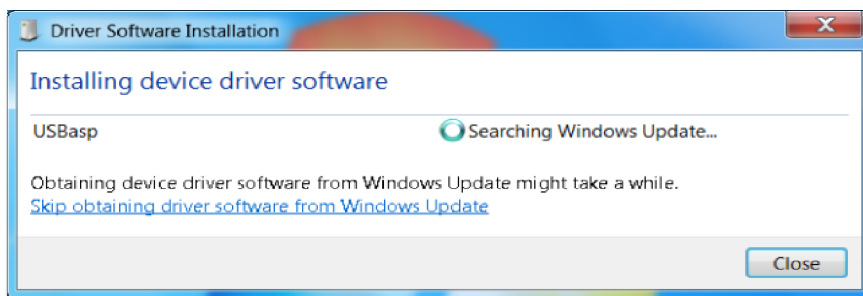


Figure 6. Driver Installation on Windows 7 – Installing Device Driver Software.

d) And will fail.

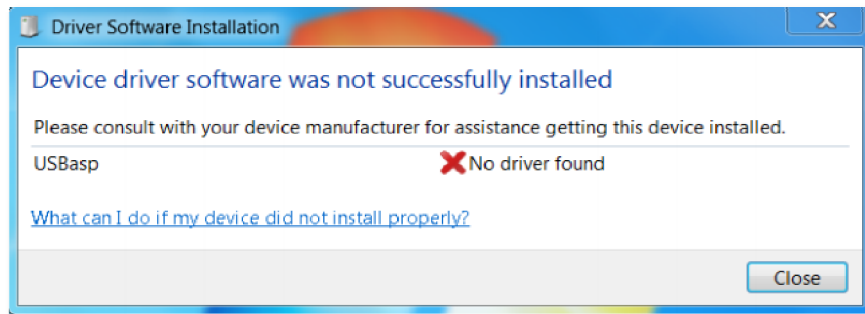


Figure 7. Driver Installation on Windows 7 – No Driver found.

e) At this point, go into device manager and find the entry for the USBasp Programmer. It should be displayed with a yellow alert icon next to it.

f) Right click on the device and select “Update Driver Software”

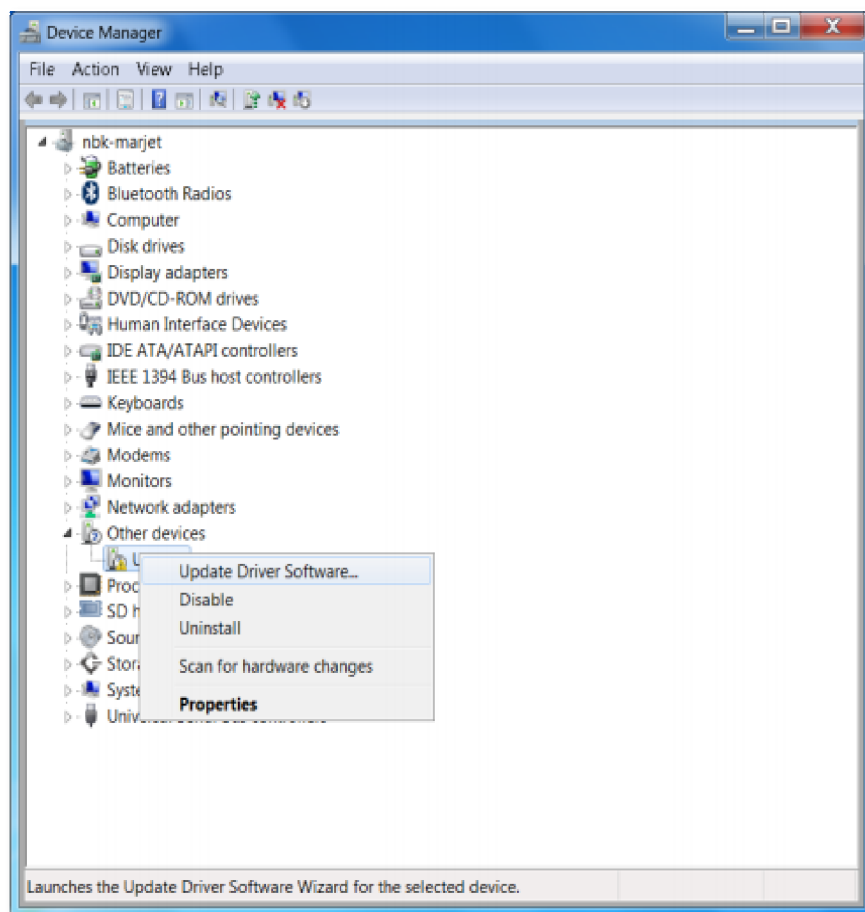


Figure 8. Driver Installation on Windows 7 – Device Manager

g) When prompted “How do you want to search for driver software”, select “Browse my computer for driver software”

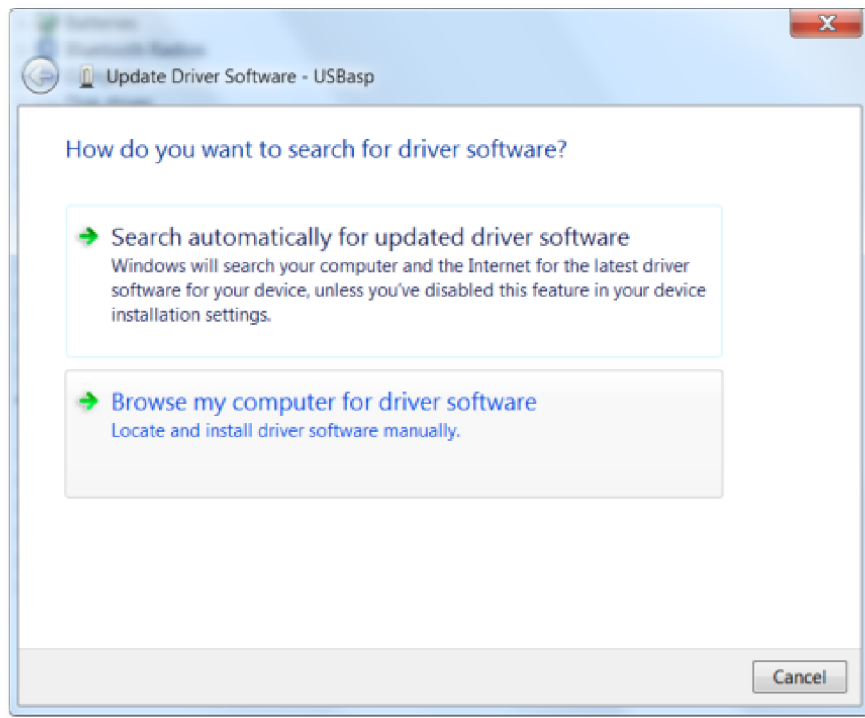


Figure 9. Driver Installation on Windows 7 – Update Driver Software

h) Select the folder where the driver files then click “Next”

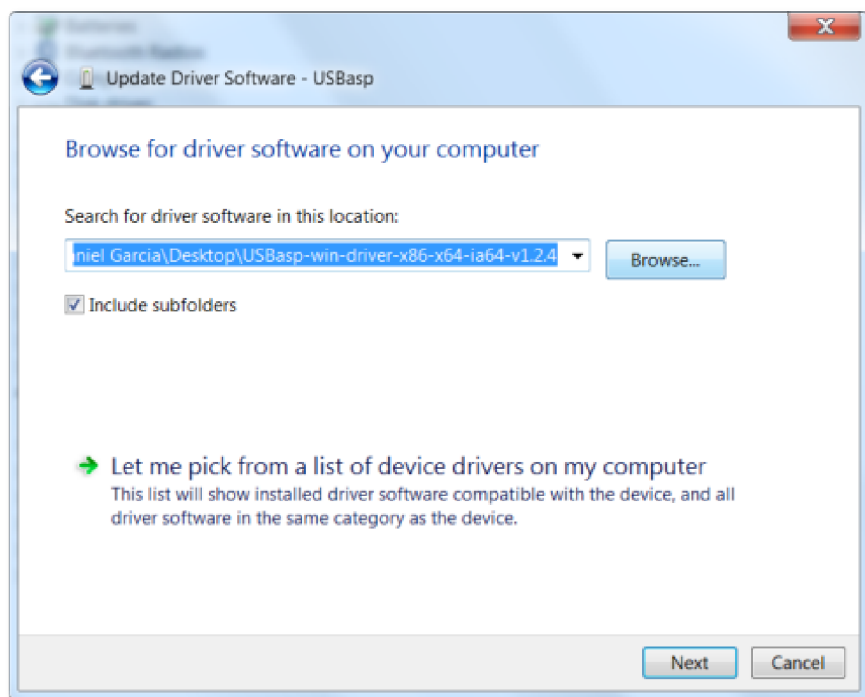


Figure 10. Driver Installation on Windows 7 – Browse for Driver Software

i) When the installation is complete, a confirmation screen will be displayed. Click Close to close it.

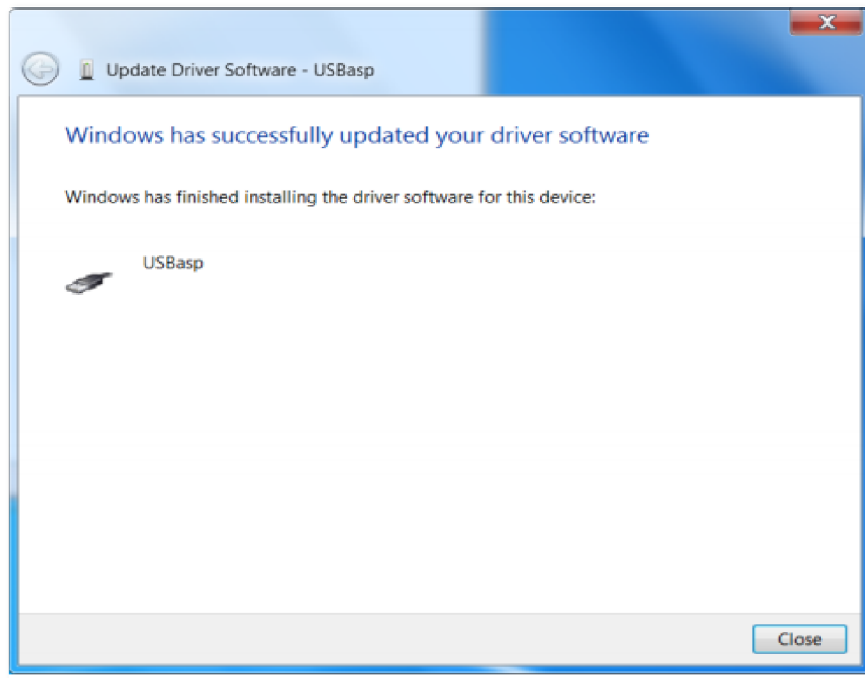


Figure 11. Driver Installation on Windows 7 – Confirmation

j) Your Programmer is now ready for use.

3.1.2. Windows Vista (32 Bit)

3.1.2.1. Required items

Items required to run this procedure are:

- a) USBasp Programmer.
- b) Computer with USB port and Windows Vista 32 Bit installed.

3.1.2.2. Assumptions

This procedure assumes that:

- a) The logged in user has sufficient privileges to install device drivers.

3.1.2.3. Procedure

To install the USBasp Programmer:

- a) Insert the Programmer into an available USB port
- b) When the “Found New Hardware” dialog opens, select “Locate and install driver software (recommended)”



Figure 12. Driver Installation on Vista 32 bit – Found new hardware

- c) Wait for a while Windows Vista attempts to locate a driver
- d) When the “Found New Hardware – USBasp” dialog box is displayed, select “I don’t have the disc. Show me other options”

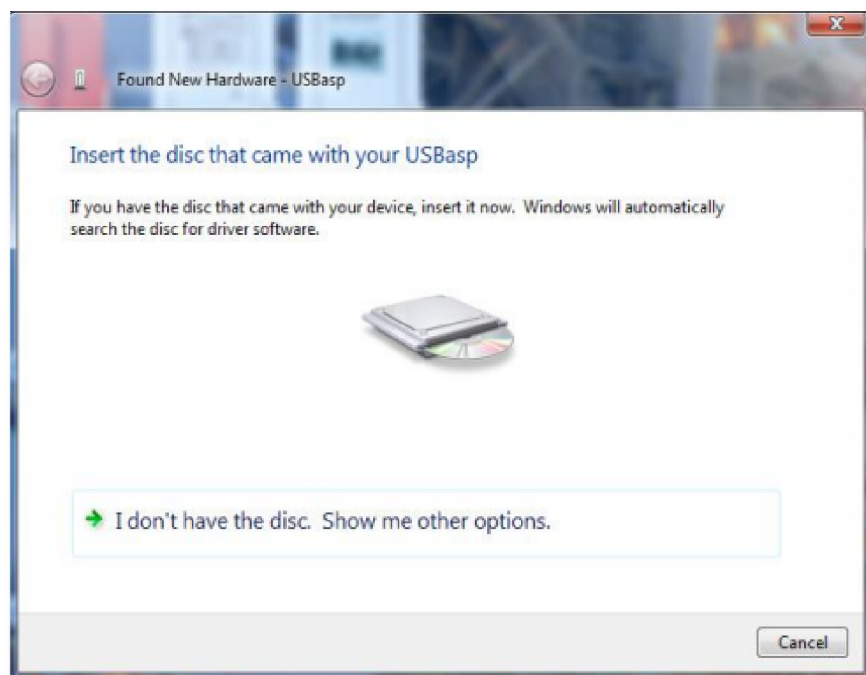


Figure 13. Driver Installation on Vista 32 bit - Found New Hardware – USBasp

e) On the next screen select “Browse my computer for driver software (advanced)”

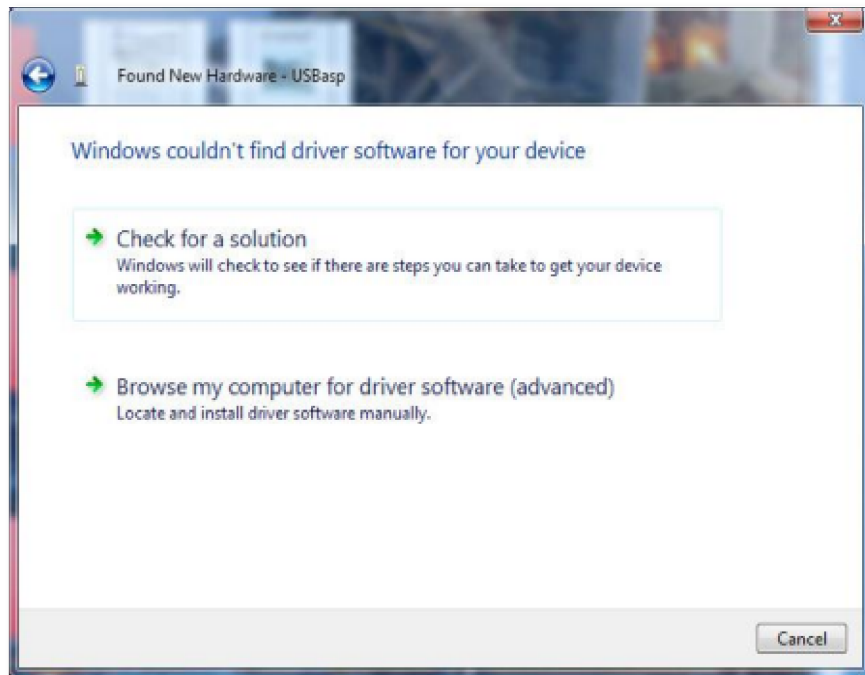


Figure 14. Driver Installation on Vista 32 bit – Windows couldn't find driver software for your device

f) Click Browse and select the folder which contain the USBasp drivers, then click Next

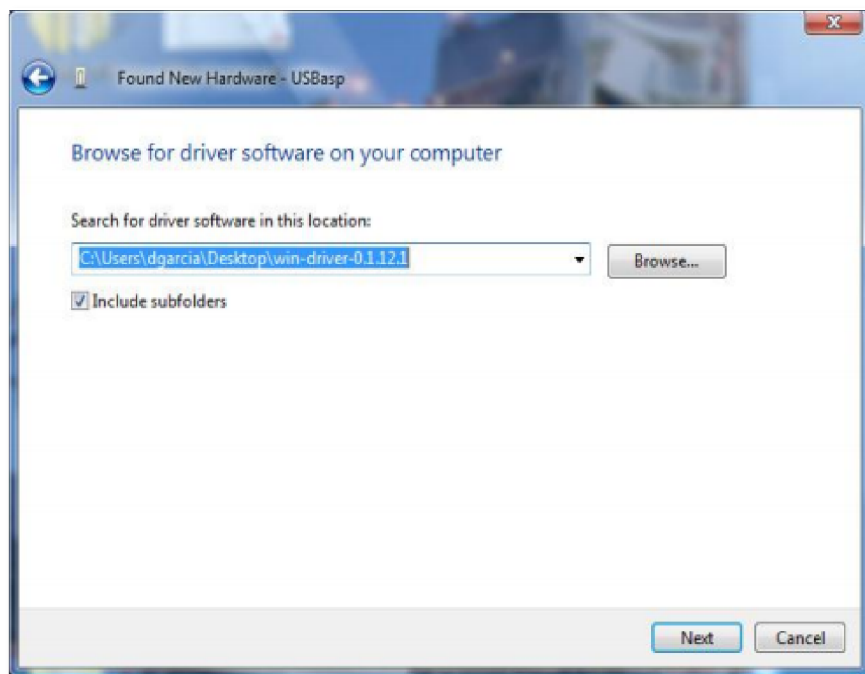


Figure 15. Driver Installation on Vista 32 bit – Browse for driver software

g) When the installation is complete, a confirmation screen will be displayed. Click Close to close it.

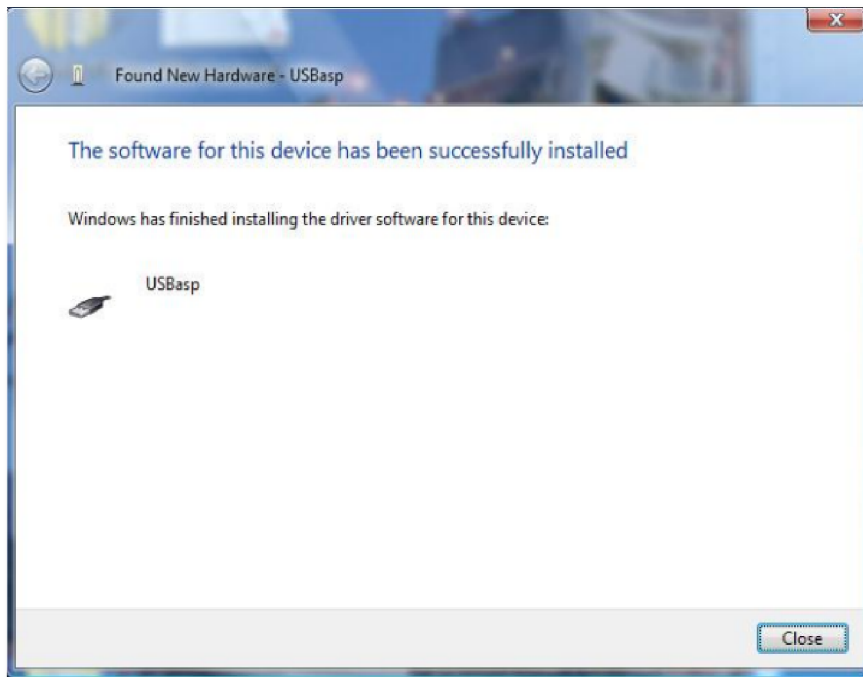


Figure 16. Installation on Vista 32 bit – Confirmation

h) Your Programmer is now ready for use.

3.1.3. Windows XP (32 bit)

3.1.3.1. Required items

Items required to run this procedure are:

- a) USBasp Programmer
- b) Computer with USB port and Windows XP 32 bit installed

3.1.3.2. Assumptions

This procedure assumes that:

- a) The logged in user has sufficient permissions to install device drivers

3.1.3.3. Procedure

To install the USBasp Programmer:

- a) Insert the Programmer into an available USB port
- b) When the “New Hardware Wizard” dialog box is displayed, select “No, not this time” then click Next



Figure 17. Installation on Windows XP – New Hardware Wizard

c) On the next page select “Install from a list of specific location (Advanced)” then click Next



Figure 18. Installation on Windows XP – Insert CD or install from specific location

d) On the Search and Installation options page

(d.1) Ensure that “Include this location in the search” is checked,

(d.2) Click Browse and select the folder where the USBasp drivers, then (d.3) Click Next



Figure 19. Installation on Windows XP – Specify Location

e) Wait for the driver to install

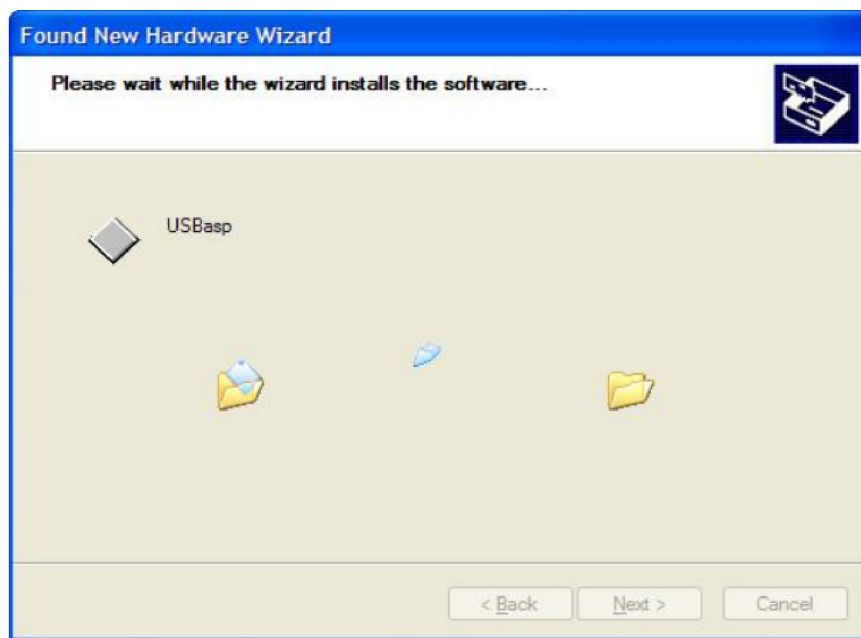


Figure 20. Installation on Windows XP – Driver installation

f) When the installation is complete, a confirmation screen will be displayed. Click Finish to close it.

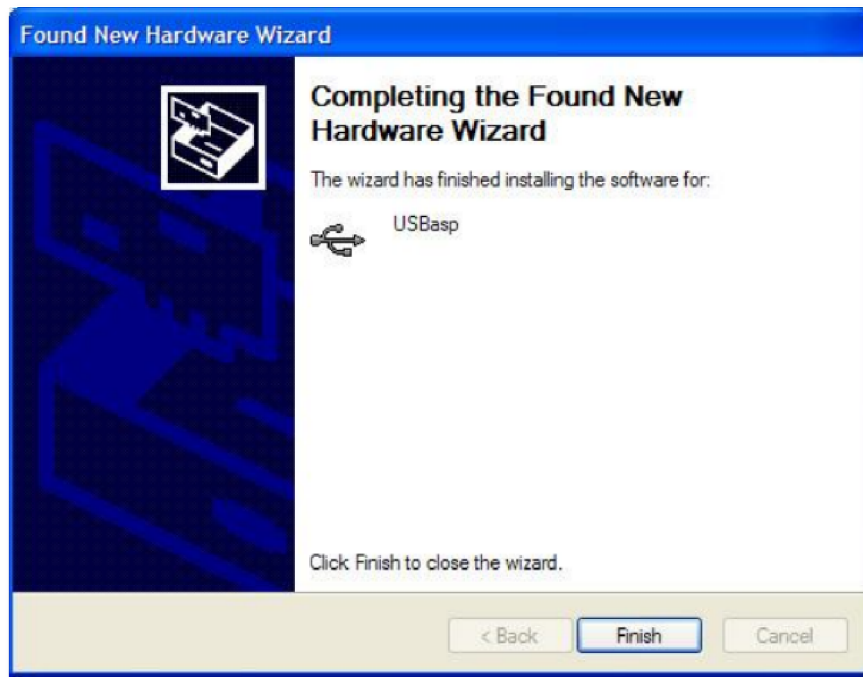


Figure 21. Installation on Windows XP – Installation Confirmation

g) Your Programmer is now ready for use.

3.1.4. Other Operating Systems

USBasp Programmer works under other operating systems such as Linux and Mac OS X. On Linux and Mac OS X no kernel driver is needed.

The USBasp Programmer uses the V-USB library and driver source code can be downloaded from this page <http://www.obdev.at/products/vusb> and <http://libusb-win32.sourceforge.net/>

Mac users may also want to look at <http://www.obdev.at/products/crosspack>

3.2. Downloading Firmware to your Microcontroller

3.2.1. Required Items

Item required for this procedure include:

- a) USBasp Programmer,
- b) Computer with USB port and AVRdude software installed,
- c) Precompiled firmware to be loaded,
- d) 8 pin male to female connecting wire, and
- e) AVR Microcontroller with interface wired to it (e.g. an AVR microcontroller on a Project board)

3.2.2. Assumptions

This procedure assumes that:

- a) AVRdude is in the path, and
- b) USBasp drivers have already been installed.

3.2.3. Procedure

To download the firmware to your microcontroller:

- a) Insert the Programmer into an available USB port.
- b) Connect the Programmer directly to the microcontroller via JP1 using connecting wire (same as the following table).

Pin Connection Settings		
Programmer JP1 Pin		Target Microcontroller Pin
XTL1	↔	XTAL1
XTL2	↔	XTAL2
VCC	↔	VCC
SCK	↔	SCK
MISO	↔	MISO
MOSI	↔	MOSI
RST	↔	RESET
GND	↔	GND

Table 4. Pin Connection Settings

- c) Open a command prompt
- d) Enter the following command where
 - (1) <DEVICE> is the microcontroller type you are programming (e.g. ATMEGA8 or m8).
 - (2) <FILE> is the filename of the precompiled binary or hex file.

The full list of device codes is listed at:

http://www.nongnu.org/avrdude/usermanual/avrdude_4.html

```
avrdude -p <DEVICE> -c usbasp -U flash:w:<FILE>.hex
```

```

Administrator: Command Prompt
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. All rights reserved.

C:\Users\dgarcia>cd C:\Users\dgarcia\Documents\Personal\Electronics-old\AVR Projects\My Projects\AVR Demo
C:\Users\dgarcia\Documents\Personal\Electronics-old\AVR Projects\My Projects\AVR Demo>avrdude -p atmega8 -P usb -c usbas
-U flash:w:AVRDemo.hex
avrdude: AVR device initialized and ready to accept instructions
Reading | ##### | 100% 0.01s
avrdude: Device signature = 0x1e9307
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 "AVRDemo.hex"
avrdude: input file AVRDemo.hex auto detected as Intel Hex
avrdude: writing flash (1706 bytes):
Writing | ##### | 100% 0.42s
avrdude: 1706 bytes of flash written
avrdude: verifying flash memory against AVRDemo.hex:
avrdude: load data flash data from input file AVRDemo.hex:
avrdude: input file AVRDemo.hex auto detected as Intel Hex
avrdude: input file AVRDemo.hex contains 1706 bytes
avrdude: reading on-chip flash data:
Reading | ##### | 100% 0.50s
avrdude: verifying ...
avrdude: 1706 bytes of flash verified
avrdude: safenode: Fuses OK
avrdude done. Thank you.

C:\Users\dgarcia\Documents\Personal\Electronics-old\AVR Projects\My Projects\AVR Demo>

```

Figure 22. AVRdude writing a flash image to the microcontroller

(3) You can use AVRpal.net or eXtreme Burner Windows based GUI software from DVD.

3.2.4. More information

Please refer to the AVRdude documentation at

<http://www.nongnu.org/avrdude/usermanual/avrdude.html> for more information.

4. Comment and Questions

If you have any questions or comments regarding this documentation or any of our products,
please contact us via

smjamil.bd@gmail.com

Thanks for using USBasp AVR Programmer

******* Warning Notice *******

Use this device at your own risk. Manufacturer and retailer are not responsible for any damages of user equipment. Because this is an open electronic device, wrong pin connection, short circuit, and any other misuse may damage this device also your personal computer or laptop. So please use this device carefully.