Albrecht Weinert

AVR ATmega development report

The development tool chain for weAut_01, Arduino and akin

Rev. 1.01, 24.09.2013

Prof. Dr.-Ing. Albrecht Weinert a-weinert.de
weinert – automation weinert-automation.de

Labor für Medien und verteilte Anwendungen (MEVA-Lab) meva-lab.de
Laboratory for Media and versatile Applications

Fachbereich Informatik der Hochschule Bochum
Computer Science department – Bochum University of Applied Sciences

The development tool chain for weAut_01, Arduino and akin

V01.00, 10.09.2013:	first (beta) report
V01.01, 23.09.2013:	tested by one plain slate installation
V01.02, --.--.2013:	-
V01.03, --.--.2014:	-

Version: V1.01

Last modified by A. Weinert at 24.09.2013

Note on numbering:	There is one common numbering for figures, lists, tables etc.
Note on version control, SVN URL:	https://ai2t.de/svn/albrecht/pub/AVRtoolChain.odt
Note on publications: See also	http://a-weinert.de/publication_en.html
This document's URL:	http://a-weinert.de/pub/AVRtoolChain.pdf. That might be newer if this is from elsewhere or on paper.
The tool's download URL (for partners only)	https://ai2t.de/infos/..??.. revealed to eligible persons on demand

Table of content

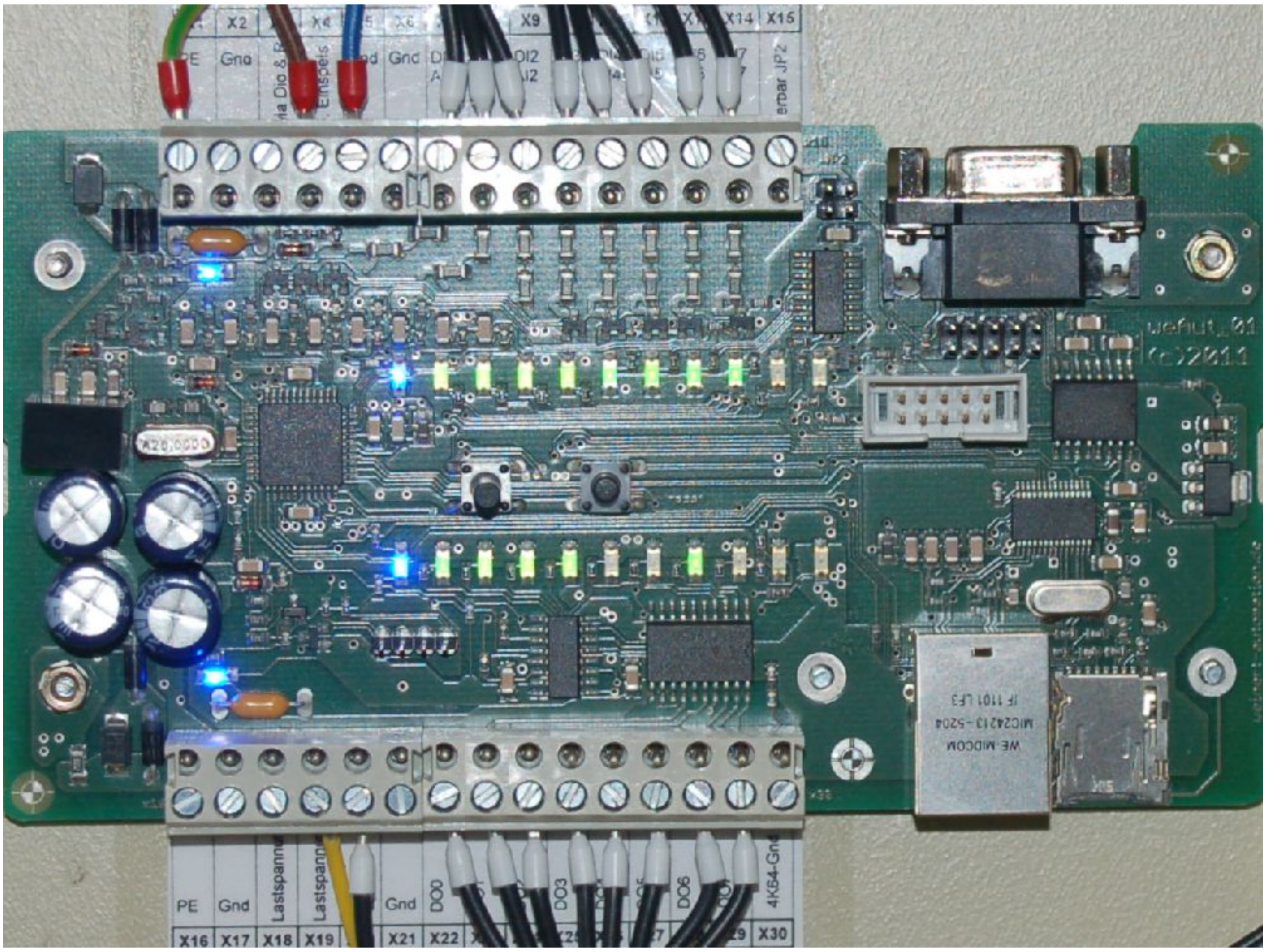

Figure 1: The automation module weAut_01 (weinert – automation 2012)

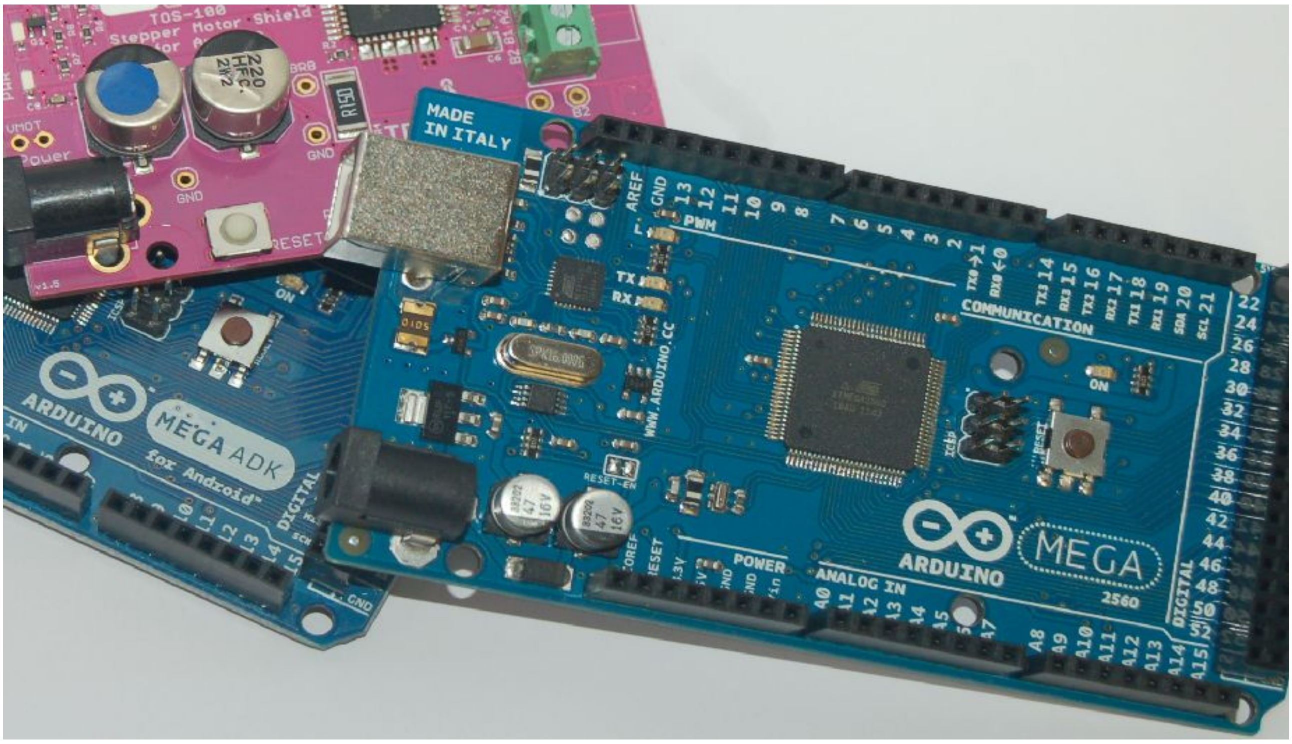

Figure 2: The ArduinoMega2560 (top), the ArduinoMegaADK (covered) and a “shield” (pink)

1. The tools

1.1 Scope and embedded targets

This is about the development and testing of embedded software written in C for AVR Atmel ATmega based systems, like e.g.:

- weAut_01 – automation module with industry standard process I/O (see [We1, We2] and figure 1 on page 3) ,
- ArduinoMega2560 and ArduinoMegaADK – i.e. bare ATmega evaluation boards (see figure 2)
- ArduinoUno – raw ATmega evaluation boards, too

and the like. See also the comparison table in [we4]. The AVR µControllers (µC) used on these boards are the target:

ATmega1284P	(on weAut_01)
ATmega2560, ATmega328P	(on ArduinoMega, ArduinoUno).

1.2 Tool types and tools

The tools needed are to be categorised as

1. infrastructure and system enhancements
2. programming / software development
3. building and deployment
4. target hardware related programming and communication
5. testing measurement

The complete tool chain consists of

- (1) Versioning system
  like **Subversion** (SVN)
- (1) Supporting tools and frameworks
  like **Java8**, **Frame4J** and on Windows
  porting of (Linux) standard tools (done by WinAVR)
- (2) Integrated development environment (IDE)
  like **Eclipse**
- (2) C – compiler suite
  like **GCC** and **WinAVR**
- (3) Automated building tools
  like **make** and to a lesser extend Ant
- (3) Automated documentation generator tools
  like **Doxygen** and to a lesser extend JavaDoc

- (4) Hardware programming tools
  like **avrdude** plus all hardware programmer related
  drivers for USB2serial and USB links and a
  serial bootloader
- (4) Tools for serial (direct or USB2serial) communication
  like **HTerm** and **vsst**.exe
- (5) Tools for hardware debugging and measurement
  like the Salea **logic analyser**
- (5) Tools for Ethernet communication debugging and measurement
  like **wireShark** (runs best on Linux using a “break out switch”)

2. The Installation

2.1 Scope and development workstation targets

All tools listed here – for the development, deployment and testing of embedded C software for ATmega-based systems – are free of charge or open source. Nevertheless the terms of use and Copyright of the authors is to be respected.

Almost all of this software is available and usable for both

- Windows (tested on Windows7 32 and 64 bit and others) and
- Linux (tested on Linux Mint 12, Mint 14 and others)

The exceptions are:

- Wireshark
  is available for Windows. But the sniffing on Ethernet ports requires extra drivers and encroachments to the operation system. There are experiences of that being neither robust nor free of bad side effects.
  On Linux no such extra burden was undergone so far.
- WinAVR's porting of Linux tools and commands (for sake of the incedible „make“ and others) to Windows it's not necessary on Linux systems

All hints given here concentrate on Windows as wide spread development workstation system. Concrete examples and files relate to Windows 7 professional 64 bit.

Some of the tools presented here have a bustling version life. Naming concrete versions and installation files mirrors the time of (never ending) writing. It does not imply that no newer or older versions should/can as well be used. Anyway, any version and file named was successfully installed tested and used.

2.2 Sequence of installation

Many of the tools listed above or suitable alternatives will be installed on a developer's workstation, already. Mostly, that's no problem as the sequence of installation may be arbitrary to a large extend.

On the other hand many tools build on each other and the installers are often clever enough to recognize and register their requirements instead of failing or doing extra installations in an uncontrollable way. Hence there are some recommendations:

- Java: first of all as it is the base for an enormous amount of other software
  before: SVN and Eclipse
- Eclipse:
  after: Java, SVN, WinAVR, GCC, make and if used
  after: ANT, Doxygen

3. Java and Frame4J

A good Java JDK installation is the base of many applications and tools. One may very well use Java 8 even the early access (ea) version even with neither need nor support (Eclipse) for Java8 features. Of course, Java7 can be installed and used instead or additionally.

3.1 Version and installation files

Java8:
version: 1.8.0-ea
Java(TM) SE Runtime Environment (build 1.8.0-ea-b106)
Java HotSpot(TM) 64-Bit Server VM (build 25.0-b48, mixed mode)

Date: Sept. 2013

Files: 112.722.840 jdk-8-ea-bin-b106-windows-x64-05_sep_2013.exe
64.092.613 jdk-8-ea-docs-b106-all-04_sep_2013.zip (oracle)

Installation directory, recommended: C:\util\jdk\ or C:\util\jdk8\ if not the preference

Path setting, recommended: C:\util\jdk\bin

Java7:
version: 1.7.0_40
Java(TM) SE Runtime Environment (build 1.7.0_40-b43)
Java HotSpot(TM) 64-Bit Server VM (build 24.0-b56, mixed mode)

Date: Sept. 2013

Files: 31.337.120 jdk-7u40-windows-x64.exe
60.368.920 jdk-7u40-apidocs.zip (oracle)

Installation directory, recommended: C:\util\jdk7\ or C:\util\jdk\ if it's the preferred JDK

Path setting, recommended: C:\util\jdk\bin

Frame4J:
version: Implementation-Version: 1.07.03
Specification-Version: 1.07

Date: Sept. 2013

Files: 22.029.027 erg.zip (a-weinert.de/frame4j/, ai2t.de/public/frame4j/)

Installation directory, recommended: C:\util\jdk7\

Hint: To extract a .zip or a .jar
go (cd) to the destination directory and say (e.g.)

```
. . . . > cd /D C:\util\jdk7\
C:\util\jdk7> jar.exe  xfv  D:\downloads\jdk-7u40-apidocs.zip
```

The java tool jar as well as others is available as soon as a JDK is installed and on the path.

One may need to run `jar.exe  xf` (respectively the cmd.exe) with administrative privileges on some installation directories.

3.2 The installation

Java:
Run the installer.
Do not forget to change the installation directory.
Kill java.exe and javaw.exe in c:\Windows\System32
Extract the fitting ...docs...zin in jdk's installation directory.

Frame4J on JDK:
Extract erg.zip in jdk's installation directory.

Frame4J on JRE
Copy frame4j.jar from/to ...\jre\lib\ext.

3.3 Configuration and tests

java -version
must work after “the killing” in system32 and display the preferred JDK-version.

java AskAlert
must show the installed Frame4J version

java ShowProps
must display a list of Java and system properties and setting.
Check for sensible values

java ShowPorts

will display all installed serial ports as well as USB2serial ports used so far.
May display a missing or non fitting (32/64bit) .dll conflict.

4. Subversion

It is strongly recommended to use a version control system even in small sized projects. Nevertheless for cross-platform embedded C development this is optional. Hence the installation and installation file details are not given here.

In the end (also on Linux) one will have three types of SVN-clients installed:

- The command line tool for professional and automated work (indispensable)
- A plug-in called Tortoise for the file explorer (very nice comfort)
- A SVN plug-in for the Eclipse IDE (a nice to have comfort function too, but in many cases not very robust/reliable)

4.1 Version and installation files

SVN: 1.6.17-SlikSvn-tag-1.6.17@1130898-X64 (SlikSvn/1.6.17) X64
5.124.096 Slik-Subversion-1.7.2-x64.msi

TortoiseSVN: 1.6.16, Build 21511 - 64 Bit , 2011/06/01 19:00:35
Subversion 1.6.17, apr 1.3.12, apr-utils 1.3.12, neon 0.29.6
OpenSSL 1.0.0d 8 Feb 2011, zlib 1.2.5

Eclipse plugin: Subversion client adapter (Tigris) 1.8.6

or all clients: <= 1.5

4.2 Installation and version tipps

To have no SVN at hand is never a good idea. At least the command line client should be available. To have it run SlikSvn's installer.

As hinted bove a regular SVN user is likely to have three or more SVN-client installations on the same workstation. Of course, they all must work interchangeably on the same local working copies.

Regrettably, the local working copy (special files) format was changed from SVN 1.5.x to 1.6.y. As no older client can handle a working copy initialised or re-formatted by a newer one, one should never mix client versions and, if in doubt, stick to the older one.

5. WinAVR and GCC

5.1 Version and installation files

Version: (WinAVR 20100110) 4.3.3

Date: Jan. 2010 - Sept. 2013

Files: 28.840.282 WinAVR-20100110-install.exe (sourceforge)

Installation directory, recommended: C:\util\WinAVR\

Path setting, recommended: C:\util\WinAVR\bin\;C:\util\WinAVR\utils\bin\

5.2 The installation

Run the installer and set the path.

5.3 Configuration and tests

```
avr-gcc.exe –version
```

must show the installed version.

```
ls -a
```

acts a bit like dir /b

6. Doxygen (optional)

6.1 Version and installation files

Version:		1.8.0, 1.8.1 or 1.8.2	
Date:		Sept. 2013	
Files:	1	3.426.068	doxygen-1.8.0-setup.exe
		3.581.275	doxygen-1.8.0.windows.x64.bin.zip
		3.925.635	doxygen-1.8.2.windows.x64.bin.zip
		doxygen_manual-1.8.1.pdf	

Installation directory, recommended: C:\util\ (doxygen.exe only)

Path setting, recommended: C:\util\;C:\Program Files\MiKTeX 2.9\miktex\bin\x64

6.2 The installation

The easy part – doxygen alone:
Run / unzip the installation file and
move doxygen.exe to an utility directory on the path.

That's it – as log as generating html documentation (like javaDoc) is enough. The toils begin (and MiKTeX comes in) if generation of .pdf documentation is needed too.

7. Further optional tools…

ANT is optional. In the GCC world reasons are to stick to the incredible “make” (comes with WinAVR.

7.1 ANT

Version:	1.8.3	
Date:	Feb. 2012 - Sept. 2013	
Files:	8.093.329	apache-ant-1.8.3-bin.zip

Installation directory, recommended: C:\util\ant\

Path setting, recommended: C:\bat\   (for the starter script/batch)

Installation:

Unpack the .zip file to the target's parent directory and rename the (new) version named directory to just .(C:\util]\).ant\

Make a system environment variables named ANT_HOME and JAVA_HOME:

```
set ANT_HOME=C:\util\ant
set JAVA_HOME=C:\util\jdk
```

Make a batch file like listings 3.

```
@Echo.
@echo ant(Launcher) %*
@if NOT %ANT_HOME%X==X goto :doAnt
@echo No ANT_HOME environment variable
@exit /b 999

:doAnt
@%JAVA_HOME%\bin\java.exe
   -classpath %ANT_HOME%\lib\ant-launcher.jar
        -Dant.home=%ANT_HOME%
              org.apache.tools.ant.launch.Launcher  %*
```

Listing 3:   A sensible path setting in the end (just an example)

Test: ant -version

8. Eclipse

8.1 Version and installation files

Version: Kepler Release   Build id: 20130614-0229

File: 258.900.493 eclipse-jee-kepler-R-win32-x86_64.zip

8.2 The installation

Unpack the .zip to the installation target directory   C:\util\eclipKe.
Alternatively – and almost always better – copy a fitting (proven and configured) installation.

8.3 Final hints

Eclipse, too, suffers from incompatibilities of plug-ins and destroying updates.
It's wise to keep a full copy of a running installation as back-up.
It's also recommendable to have different full installations for different purposes.

9. Hardware programmers and drivers

USB2serial driver for ArduinoUno and ArduinoMega2560

File: 95.890.468 arduino-1.0.4-windows.zip

Installation: Unpack and use only the files under .
..\arduino-1.0.4-windows\arduino-1.0.4\drivers\FTDI USB Drivers\

Test: Use with the serial bootloader [We4], for example, by something like:

```
avrdude -p atmega328p -P com3 -c avr109 -b 38400 -v -t
```

USB-Driver for AVRisp

File: 231.269 avrispmkii_libusb-win32_1.2.1.0.zip

Installation: Unpack and use only the files under .
..\avrispmkii_libusb-win32_1.2.1.0\avrispmkii_libusb-win32_1.2.1.0\

Test: Use with a ISP programmer, by something like:

```
avrdude -p atmega328p -P usb -c avrisp2 -b 38400 -v -t
```

10. Communication and testing tools

10.1 WireShark

Only needed when implementing Etheret protocols for the target hardware.
Best installed and run on a Linux workstation. (Less joy on Windows.)

10.2 HTerm

HTerm is a nice, versatile and comfortable serial Terminal program. It is very well suited to communicate with the UARTs on your ATmega based boards – be it "directly" via RS232 (by MAX level shifter and protection ICs on weAut_01 or via USB2serial link (by ATmega8U2 on some Arduinos).

On HTerm you will enter and optionally edit a line and finally send it with a LF and/or CR.

HTerm will never – in no configuration – send single characters directly on your keystroke. For the (given) cases where just that is necessary do use the much simpler console tool vsst (see bellow).

Version: 0.1.8. beta
File: 1.530.276 hterm.zip

Installation directory & path setting, recommended: C:\util\

Installation: Unpack the .zip; move HTerm.exe to the installation target directory.

Configuration:

Run HTerm and do the settings appropriate for you. Have the HTerm.cfg in the same directory as the .exe.

10.3 VSST

The very simple serial terminal (VSST) is a console program that sends every key input almost immediately (within some 10 ms). The exception is cntl-C which ends the program.

VSST displays every character received also immediately.

The character set is determined by cmd.exe's console encoding that may be checked and changed by the `chcp` (change code page) command.

File: 132.369 vsst.exe (from the author, https://ai2t.de/infos/tools/)

Installation directory, recommended: C:\util\
Path setting, recommended: C:\util\

Installation: Download the file vsst.exe to the installation target directory.

Run: vsst.exe [link [baud]] (link defaults to COM9, baud defaults to 38400.)

10.3 Logic

The Saleae logic analyser is very helpful to explore the timing of digital inputs and outputs on your embedded board (software). It can comfortably translate protocols like CAN, UART, SPI and so on; see figure 4.

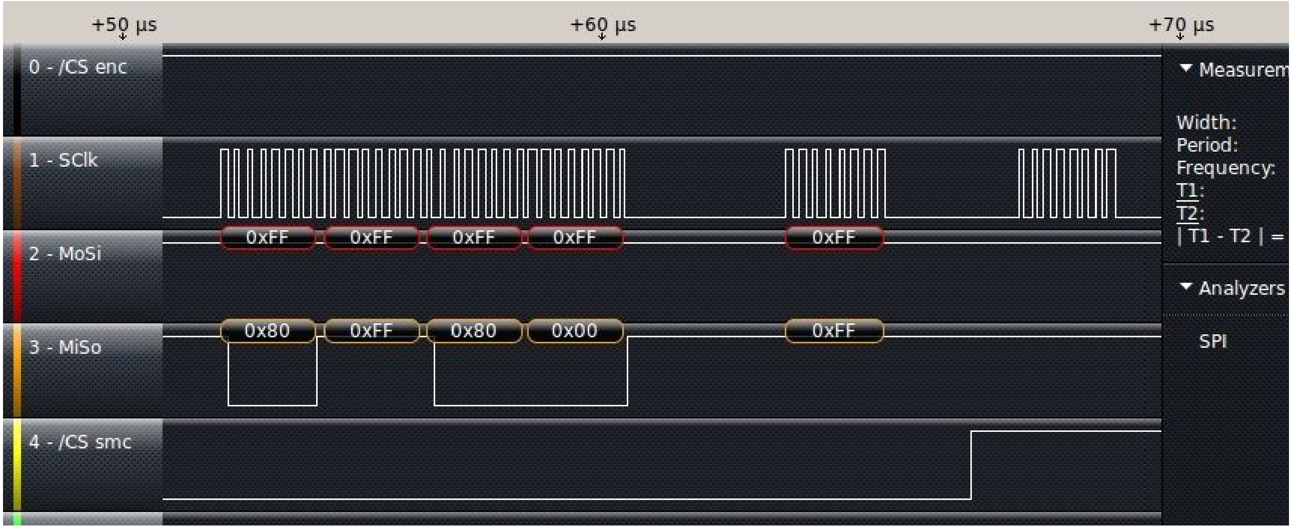

Figure 4: Part of a Saleae Logic screen shot (tracing SPI communication)

The hardware costs about 120€ (8 channels) respectively 200€ (16 channels). If you can get hold of one, you should have the software ready on your development workstation.

Of course, that's optional – but chances are, once you have it you won't miss it any more.

File download: 11.129.104 Logic Setup 1.1.15 (64-bit).exe (saleae.com/downloads)

Installation: Run the Installer
Installation directory, recommended: C:\util\SaleaeLLC

Run: c:\util\SaleaeLLC\Logic.exe (may be put in a batch)

Hint: The Linux version may run only with sudo privileges (may be put in a script).

11. Final Remarks and recommendations

Some of the tools require to be (in a certain sequence) in the systems search path for runnables, and for some it would be dispensable but a requirement of comfort.

Advice: Choose the installation directories judiciously!

That means: a) use common installation directories whenever feasible
(just one .exe plus null or up to three special .dll)
to avoid path pollution and
b) avoid blanks and parenthesis in directory names.
c) be consequent, when using case in filenames.

The importance of b) (and c)) also comes from using free / open source tools that are often developed for Unix first and ported to Windows later. Those tools often crash on file names with blanks or other special characters allowed in Windows (or on case mismatch).

That also means: Avoid the Windows directories

C:\Program Files\ and C:\Program Files (x86)\ disguised as
C:\Programme\ and C:\Programme(x86)\ in the (German) explorer

often proposed by installers, whenever possible.
The only exceptions are cases, when both the 64 and the 32 bit version are be required. One example are both JREs for sake of 32bit and 64bit browsers on a 64 bit Windows.

A sensible path in the light of above recommendations may look like the the example in listings 5.

```
PATH=C:\bat;C:\util;C:\util\jdk\bin;
   C:\util\WinAVR\bin;C:\util\WinAVR\utils\bin;
   C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;
   C:\Windows\System32\WindowsPowerShell\v1.0\;
   C:\Programme\SlikSvn\bin;C:\Program Files\TortoiseSVN\bin;
   C:\Program Files\MiKTeX 2.9\miktex\bin\x64
```

Listing 5: A sensible path setting (just an example)

12. Resume

To provide a professional tool chain for all phases of cross-platform development is no small nor easy tasks. But it pays off to have all links of the tool chain under control.

This report was written as support to this goal.

A Abbreviations

ADC	analogue digital converter
AI	analogue process input (from sensors)
API	application programmer's interface
AVR	Harvard architecture µControllers from Atmel; seems to be no acronym and to have no meaning
C	C programming language
CISC	complex instruction set computer
CLI	command line interface
DI	digital process input (from sensors)
DO	digital process output (to actuators)
HMI	Human machine interface
I/O	Input / Output
ISP	In system programming
JDK	Java development kit
JRE	Java Runtime environment
JTAG	Joint Test Action Group (or serial test access port)
RAM	random access memory (readable and writeable)
RISC	reduced instruction set computer
SPI	serial peripheral interface
SVN	Subversion
USB	universal serial bus
µC	µControllor, micro-controller

L References

[AVR1] Atmel, (doc0856.pdf)
8-bit AVR Instruction set

[AVR2] Atmel, (doc8059.pdf; preliminary)
8-bit AVR Microcontroller with 128K Bytes In-System Programmable Flash ATmega1284P

[AVR3] Atmel, (doc2549.pdf)
8-bit Atmel Microcontroller with 64K/128K/256K Bytes In-System Programmable Flash
ATmega640/V ATmega1280/V Atmega1281/V ATmega2560/V Atmega2561/V

[AVR4] Atmel, (doc1644.pdf)
AVR109: Self-programming

[AVR5] Atmel, (doc2568.pdf)
AVR911: AVR Open Source Programmer

[AVR6] Atmel, (doc8171.pdf)
8-bit Atmel Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash
Atmega48PA Atmega88PA Atmega168PA ATmega328P

[AVR7] Atmel, (doc7799.pdf)
8-bit Atmel Microcontroller with with 8/16/32K Bytes of ISP Flash and USB
ATmega8U2 Atmega16U2 ATmega32U2

[intel1] Intel, (HexFmt.pdf)
Hexadecimal Object File, Format Specification, Revision A, 1/6/88

[tool1] Richard M. Stallman, Roland McGrath, Paul D. Smith
GNU Make, A Program for Directing Recompilation, GNU make Version 3.82, July 2010

[tool2] Brian S. Dean, Jörg Wunsch
AVRDUDE, A program for download/uploading AVR microcontroller flash and eeprom
For AVRDUDE, Version 5.5, 29, October 2007

[tool3] GCC team
avr-libc 1.8.0 January 3 2012

[We1] Rolf Biesenbach, Albrecht Weinert
An economical approach for small sized automation tasks
April 2013, 9th International Symposium on Mechatronics and its Applications (ISMA13)

[We2] Albrecht Weinert
weAut_01 automation controller
user manual (German), Nov. 2011

[We3] Albrecht Weinert, weAutSys software documentation
generated by Doxygen, May 2013 or later, as .html and as .pdf

[We4] Albrecht Weinert, A serial bootloader for weAut_01, ArduinoMega and akin,
AVR ATmega development report, August 2013, as .pdf