

ARDUINO UND AVRSTUDIO 6.2

Avr Studio hängt öfters (Microsoft Monster): erst installieren nachdem man das Netzwerk deaktiviert hat. Und das Backend in einem separaten DOS Fenster starten. (siehe unten)

Laufwerke substituieren

- Um für alle Schüler eine einheitliche Umgebung zu schaffen werden zwei Laufwerke substituiert: A: und P: Damit ist die Konfiguration von AVRStudio einheitlich und auch die Skripte verwenden die gleichen Pfade. In der Datei setup62.bat sind die entsprechenden Subst Befehle anzupassen.
- A: zeigt auf die Arduino Installation
- P: zeigt auf das Projektverzeichnis mit den Setup-Batchdateien etc.

Konfiguration importieren

//Tools/Import and Export Settings

Die Konfiguration liegt unter P:/Vorlagen

Pfadnamen

- GCC ist sehr empfindlich auf Sonderzeichen im Pfad! Daher **keine Sonderzeichen in Dateinamen und Verzeichnissen** (auch keine Leerzeichen!)

Templates

Um nicht immer die gleichen Projekteinstellungen machen zu müssen wird ein Template angelegt und als Vorlage für ein neues Projekt verwendet: „./File/Export Template“. In den Einstellungen wird unter „Tools/Options/Projects and Solutions“ die Pfade für die Projekte und Templatevorlagen eingestellt.

User Settings der IDE

Die Einstellungen der IDE lassen sich über **//Tools/Import and Export Settings** abspeichern.

Probleme mit AVRStudio

Kein Treiber für Leonardo in der	die Kennung des Leonardo-Boards ist falsch	Lösung: Ändern der Kennung im .inf File im Ordner arduino/drivers/arduino.inf %leonardo.bootloader.name%=DriverInstall, USB\VID_2341&PID_8036
Backend atbackend debuggen	Das Backend in einem eigenen DOS Fenster starten und beobachten	Im //Tools/Options/Debugger den Punkt „Start local Backend Agent“ abwählen und als Port 4711 angeben. Dann das backend in DOS Fenster starten .../atbackend/atbackend.exe

Probleme mit Arduino Leonardo

- unbedingt kurze Kabel verwenden
- nicht COM4, sondern COM3 eintragen, weil Leonardo in Windows 2 Verbindungen öffnet
- Programmierung unter Virtualbox:
 - 1. Reset am Arduino
 - 2. USB in Vbox neu verbinden
 - 3. Upload Skript starten

Terminal integrieren

Terminal Extension installieren

//Tools/Extension Manager/Terminal Window

oder

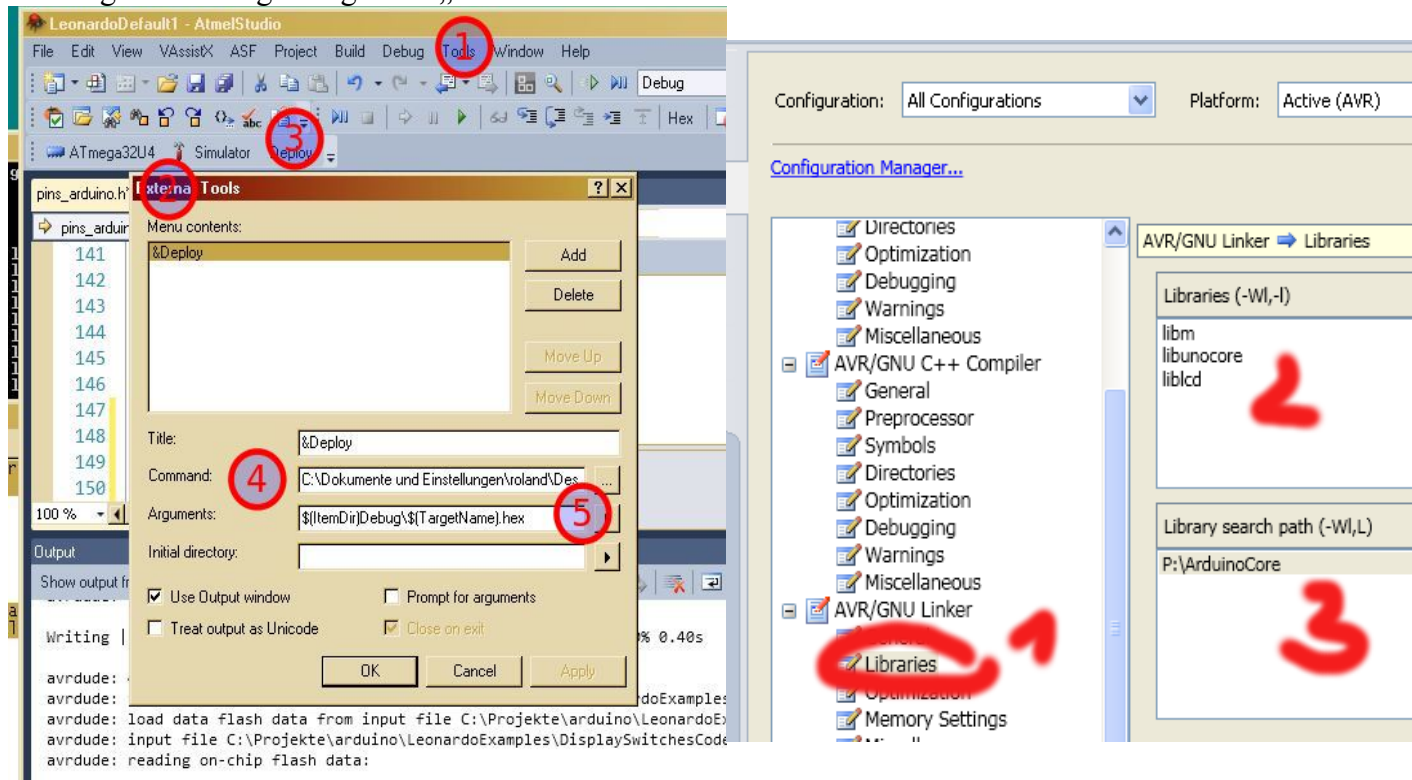
Putty herunterladen, eine Session „sessiona“ einrichten und dann mit //Tools/External Tools einrichten:

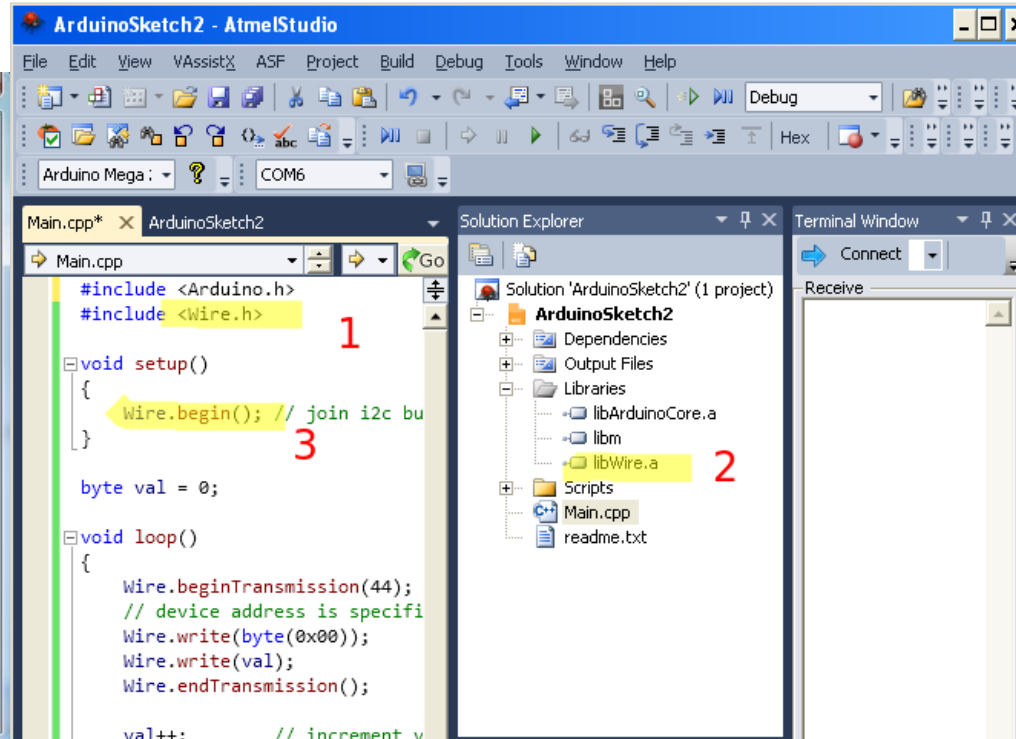
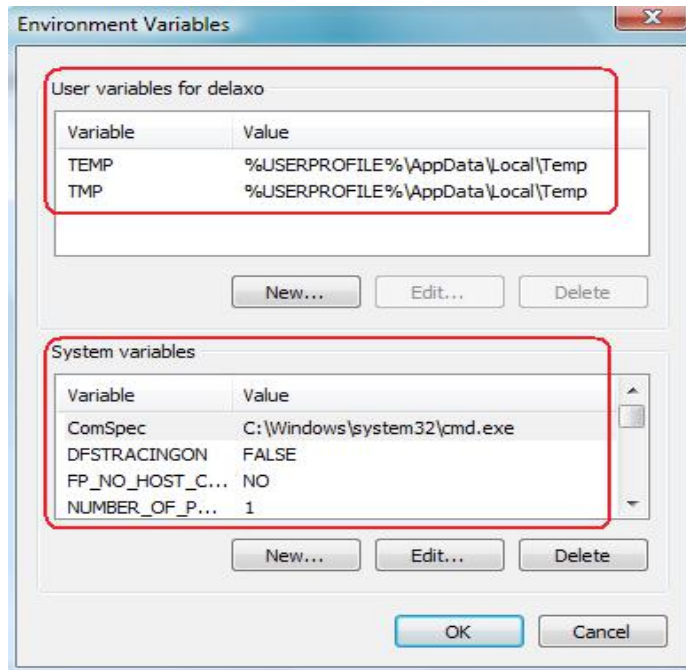
Title: Serial Monitor 9600 Baud

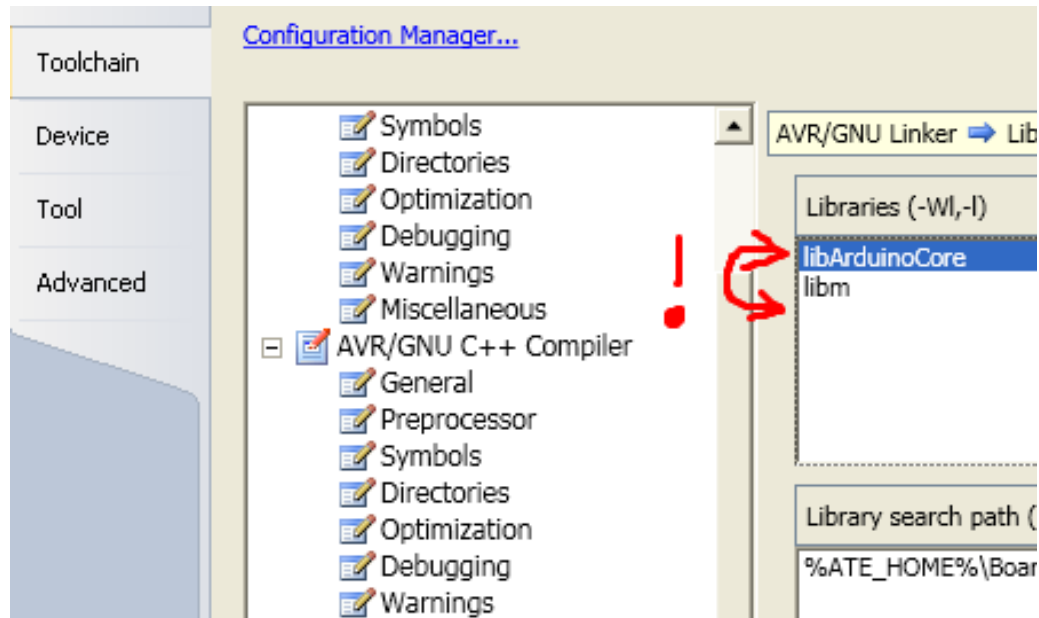
Arguments; -load „sessiona“

Programmierung

die Programmierung erfolgt über „External Tools“







(4) Aufruf von upload.bat

Deploy.bat:

```
@rem die Info über Prozessor und Programmer kriegt man in %ATE_HOME%/Boards.xml
@rem %1 wird von AVRStudio übergeben: $(ItemDir)Debug\$(TargetName).hex
SET PORT=COM5
SET BAUD=57600
rem SET BAUD=115200
SET PROCESSOR=atmega32u4
SET PROTOCOL=avr109
SET FILE=%1
"C:\Programme\Arduino\hardware\tools\avr\bin\avrdude" -C"C:\Programme\Arduino\hardware\tools\avr\etc\avrdude.conf" -v -p
%PROCESSOR% -c%PROTOCOL% -P%PORT% -b%BAUD% -D -Uflash:w:%FILE%.i
```

(5) Argumente an deploy.bat : Pfad zur Datei xxx.hex

\$(ItemDir)Debug\\$(TargetName).hex

Batchdateien

Upload.cmd

```
rem Atmega2560
rem c:\arduino\hardware\tools\avr\bin/avrdude -Cc:\arduino\hardware\tools\avr/etc/avrdude.conf -v -patmega2560 -cwiring
-P%2 -b115200 -D -Uflash:w:%1:i
rem Uno
rem c:\arduino\hardware\tools\avr\bin/avrdude -Cc:\arduino\hardware\tools\avr/etc/avrdude.conf -v -patmega328p -carduino
-Pcom4 -b115200 -D -Uflash:w:%1:i

rem Leonardo

rem Achtung com3, nicht com4, weil Leonardo nach soft-reset den Port4 verliert

rem set file=c:\temp\uno2\uno2\Debug\uno2.hex
set file=%1
C:\Arduino\hardware\tools\avr\bin/avrdude -CC:\Arduino\hardware\tools\avr/etc/avrdude.conf -v -patmega32u4 -cavr109
-Pcom3 -b57600 -D -Uflash:w:"%file%":i
```