

# Entwicklungsumgebung BiDiBone Programmierung

Als Entwicklungsumgebung für den BiDiBone mit dem Prozessor ATMXMega128D3 bietet sich das Atmel Studio 6 an.

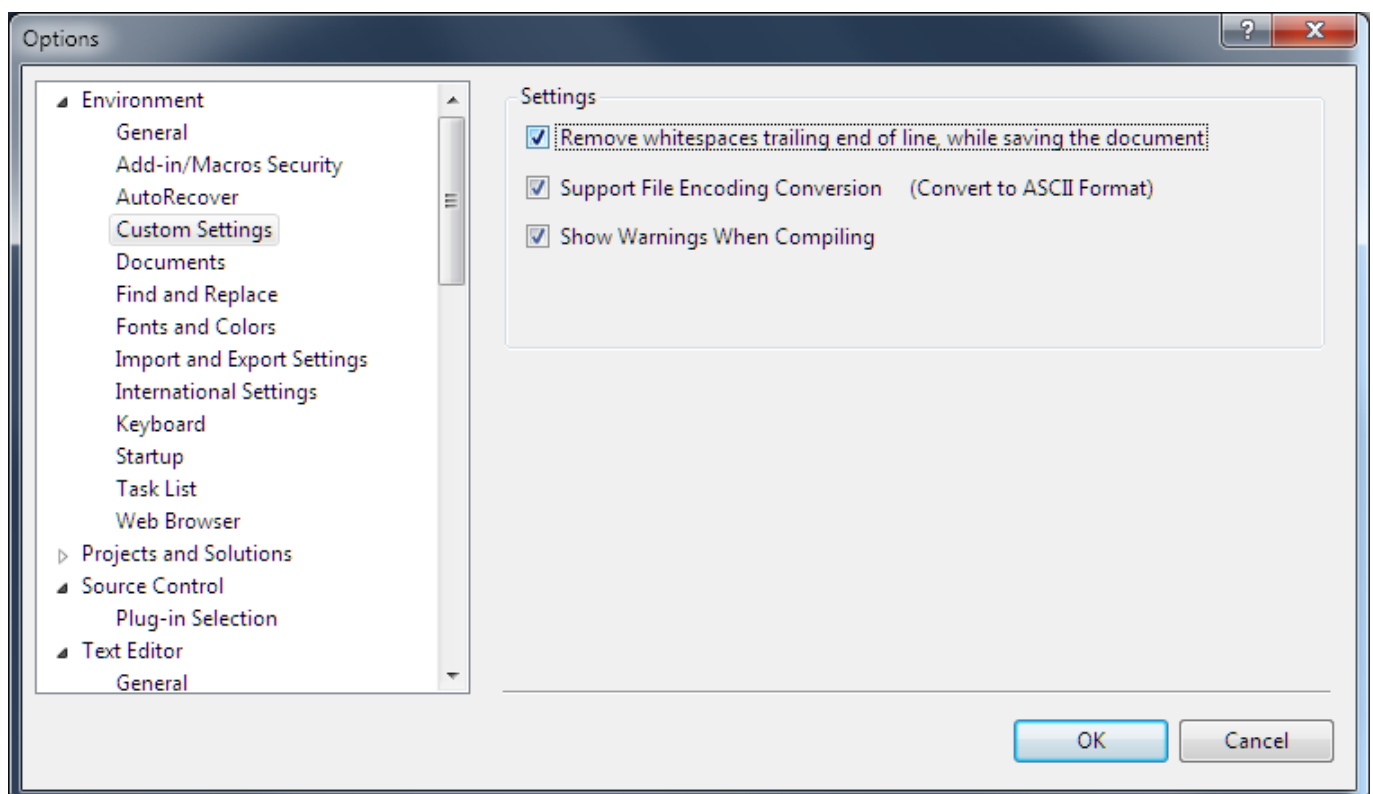
## AtmelStudio 6 einrichten

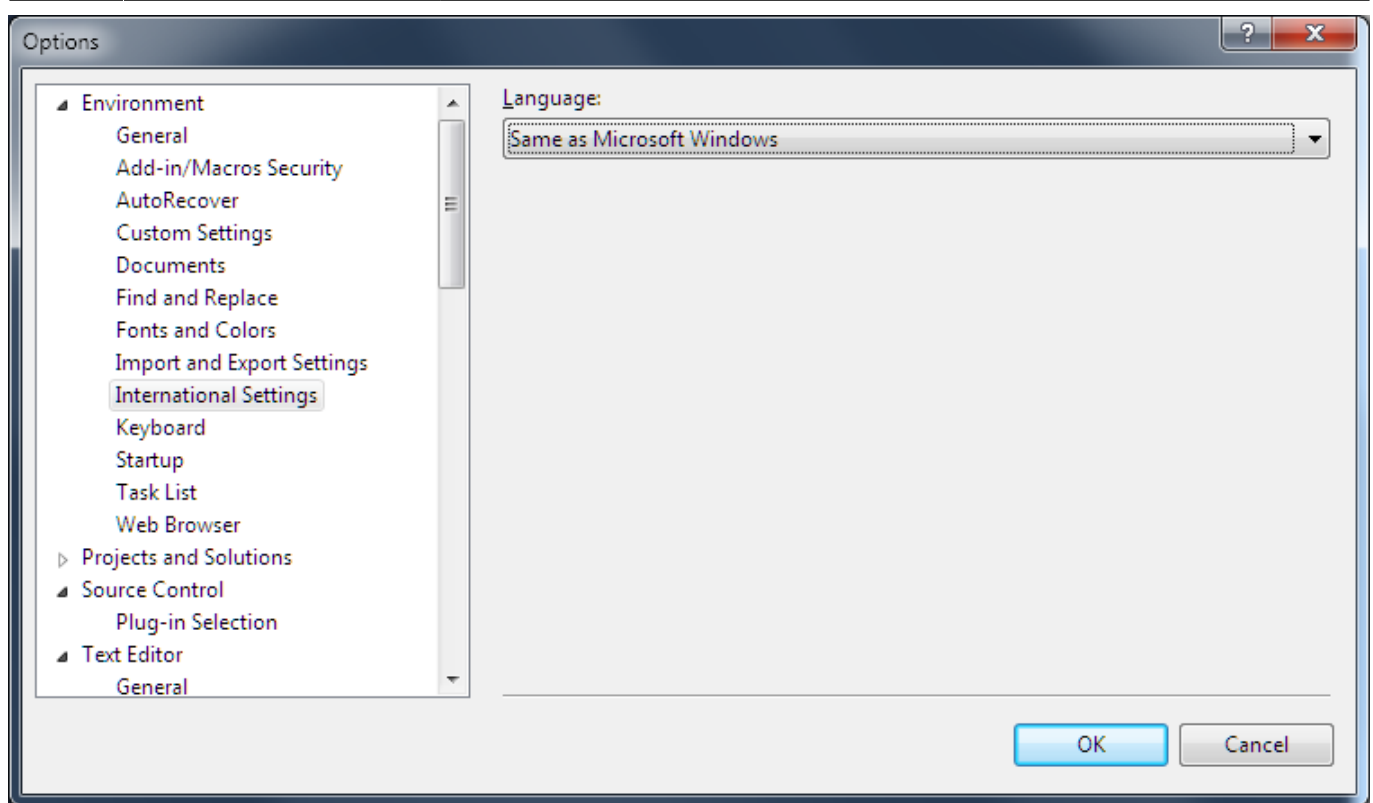
Eine Beschreibung zum Einrichten des AtmelStudios 6 findet sich auf: [Atmel Studio - Mikrocontroller.net](https://mikrocontroller.net)

Hier stehen Hinweise, die das BiDiBone-Projekt betreffen.

## Einstellungen

### Tools | Options

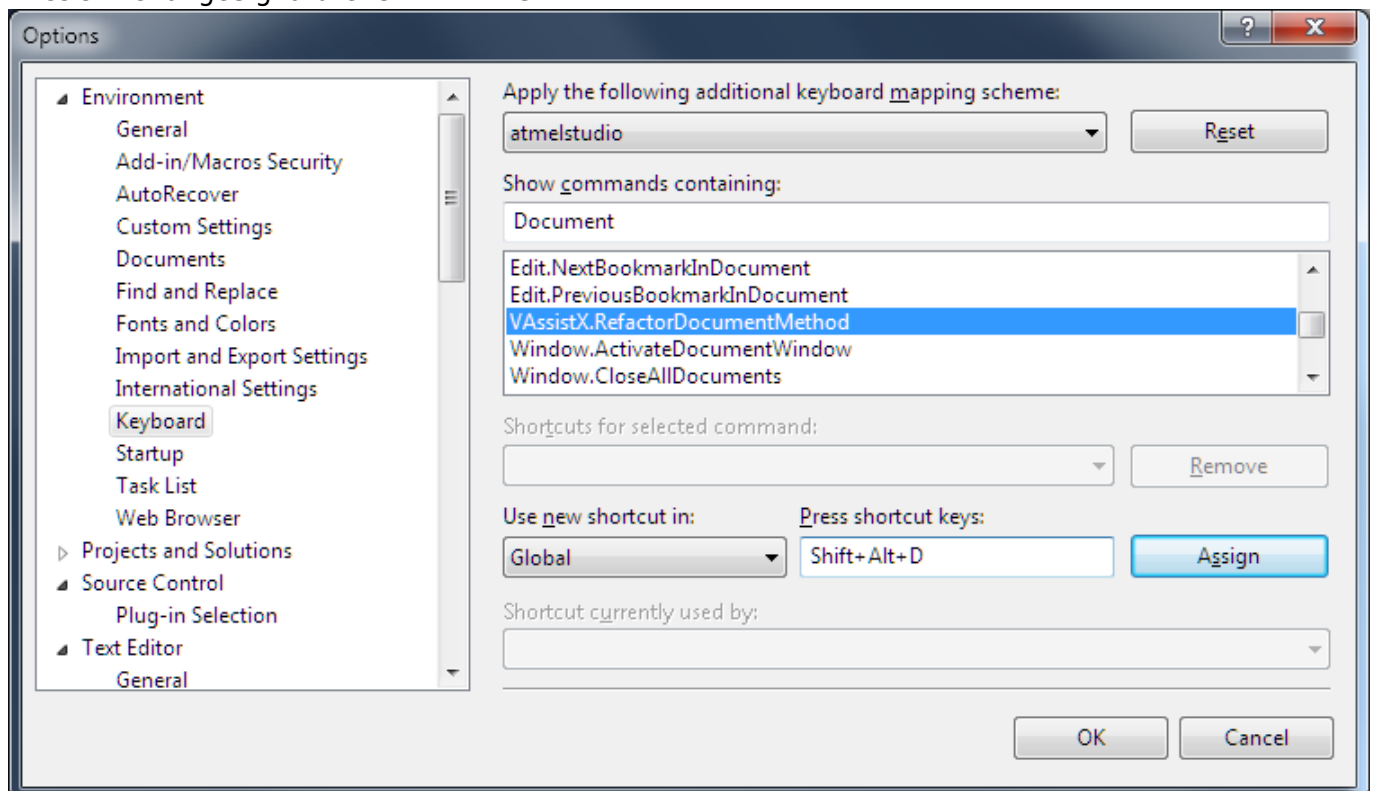




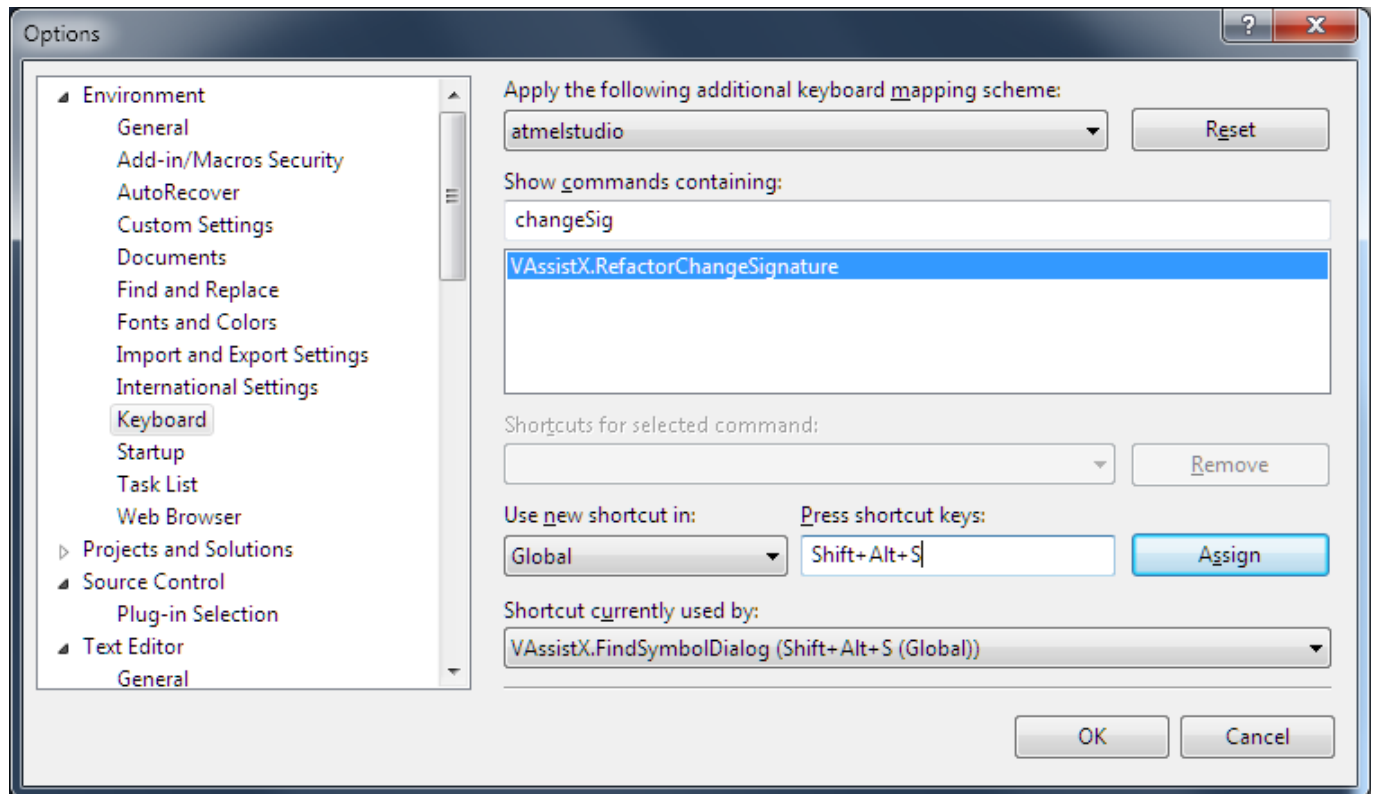
## Tastaturkürzel

(VAssistX.RefactorRename: Shift+Alt+R ⇒ schon definiert)

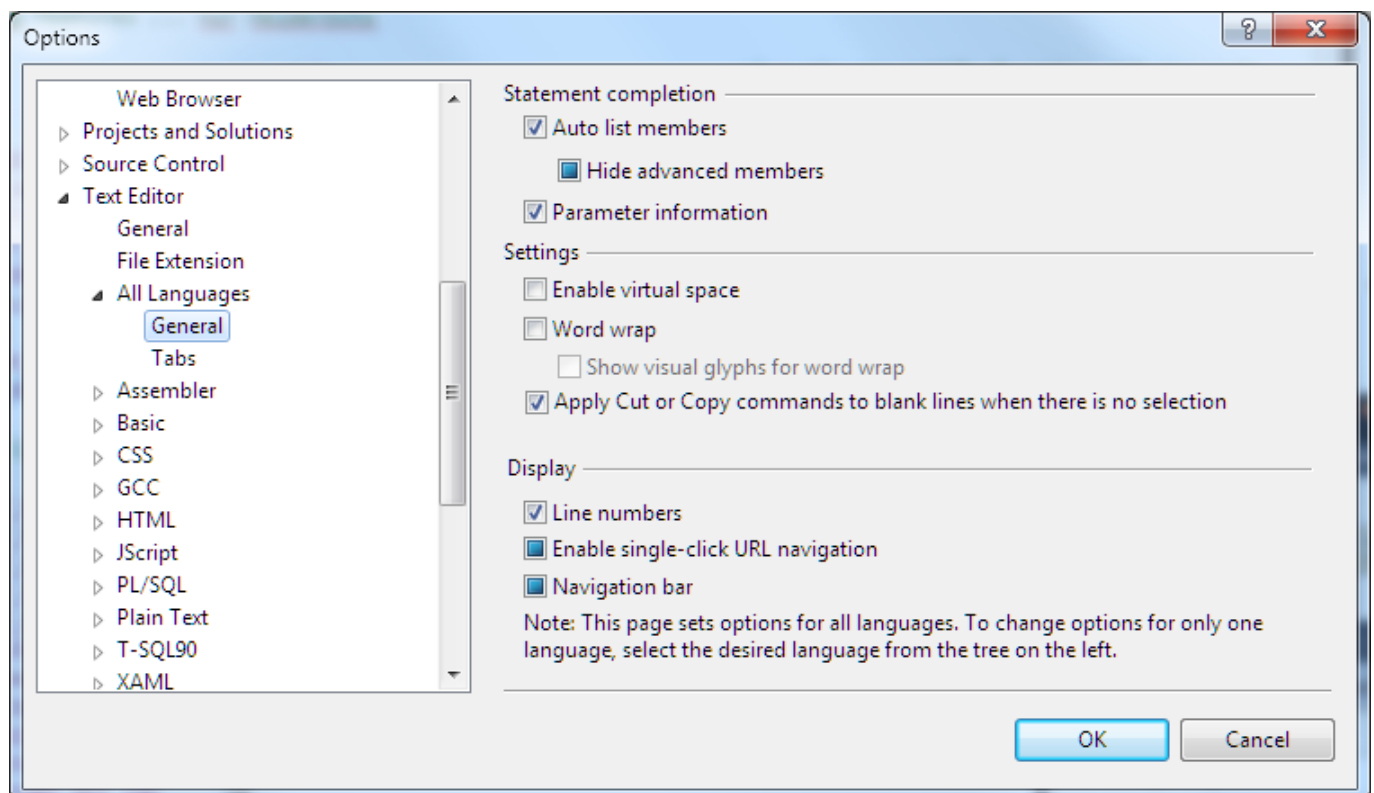
VAssistX.ChangeSignature: Shift+Alt+S



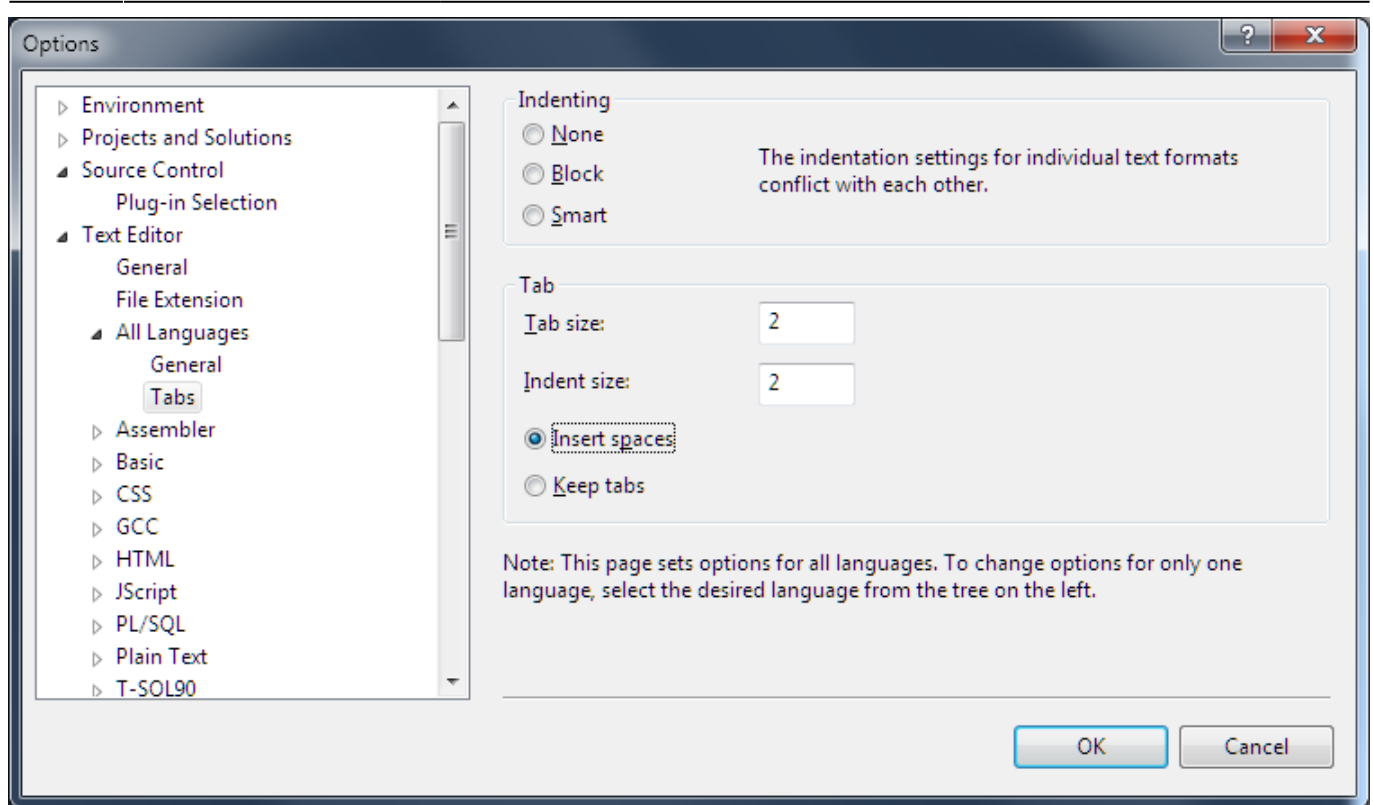
## VAssistX.RefactorDocumentMethod: Shift+Alt+D



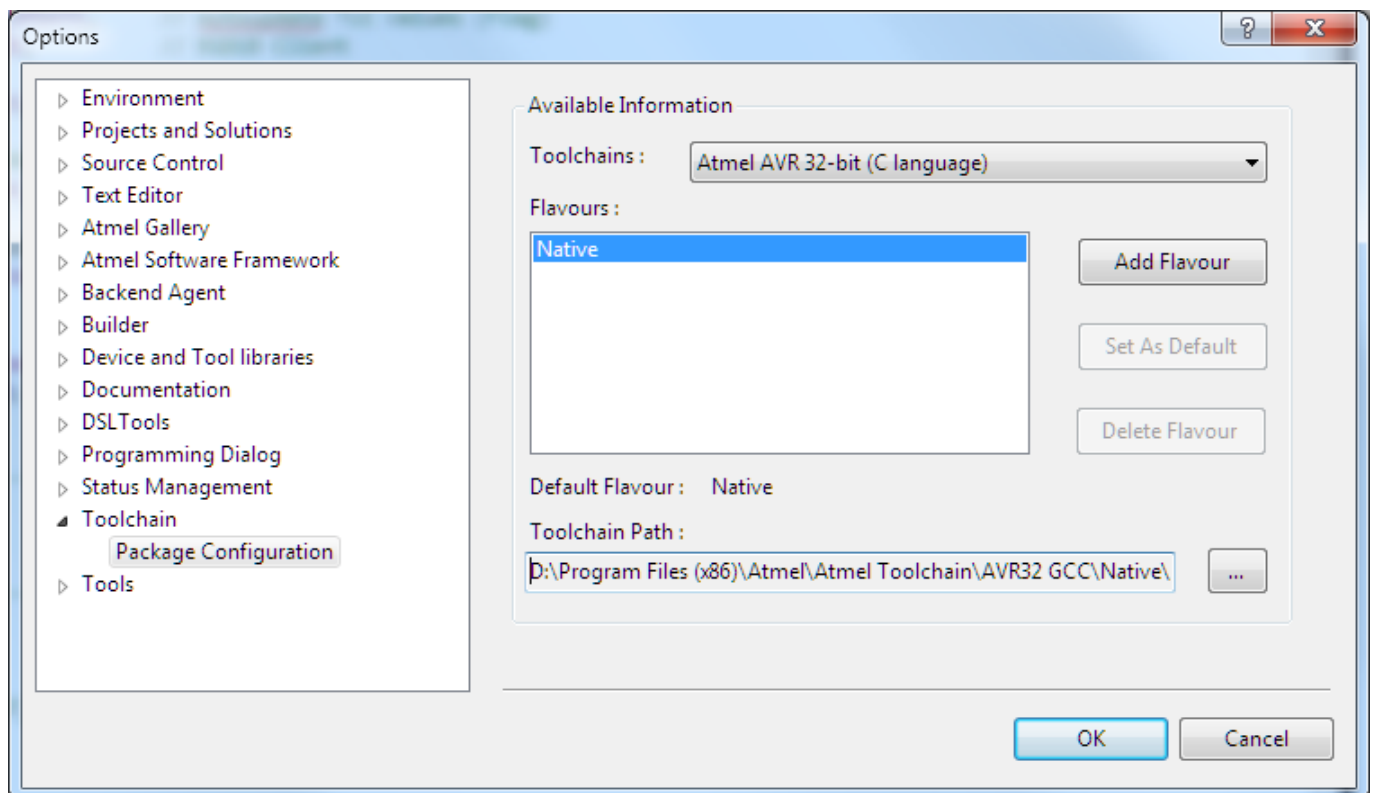
## Zeilennummern



## Tabulatoren und Einzüge



## Pfad zu Compiler und Linker



(z.B. D:\Program Files (x86)\Atmel\Atmel Toolchain\AVR32 GCC\Native\3.4.2.1002\avr32-gnu-toolchain\bin)

## Toolchain

Projekt anwählen, Kontext-Menü: PROJEKT-Properties – Toolchain

Abhängig vom Projekt die Eigenschaften einstellen.

## FUSES

Tools | Device Programming – TOOL auswählen ⇒ Apply – Fuses:

Als beste Einstellung für die Fuses hat sich folgende Kombination bewährt:

- USERID = 0xFF
- WDWP = 8KCLK
- WDP = 8KCLK
- DVSDON = [ ]
- BOOTRST = APPLICATION
- BODPD = CONTINUOUSLY
- RSTDISBL = [ ]
- SUT = 0MS
- WDLOCK = [ ]
- BODACT = CONTINUOUSLY
- EESAVE = [X]
- BODLVL = 2V1
  
- FUSEBYTE0 = 0xFF
- FUSEBYTE1 = 0xAA
- FUSEBYTE2 = 0xFE
- FUSEBYTE4 = 0xFF
- FUSEBYTE5 = 0xE5

## Modulinformationen

VAssistX | Tools | VA Outline

Zeigt die „Innereien“ eines geladenen Moduls an, wie #defines und Funktionen.

## Extensions

Via Tools | Extension Manager ...

Beachte: Atmel-Konto notwendig.

## Versionskontrolle

Mit der Wahl auf GiT bietet sich das Extension: **Git Source Control Provider** an.

## Installation und Anmeldung bei GiTLab sind in

und  
beschrieben.

Installation GiT  
GiT Anmeldung

## Dokumentation

Zur Dokumentation oder zum Lesen dokumentierter Quellen bietet sich Doxygen an.

- laden, neu starten
- Konfiguration: Tools | Options | Documentation | Doxygen Settings:
  - Doxygen Executable Path  $\Rightarrow$  z.B. D:\Dokumente\Michael\Anwendungsdaten\Doxygen
- Dokument erzeugen: Project | Doxygen ...
  - Config File Editor - Edit  $\Rightarrow$  z.B. OUTPUT\_LANGUAGE = German
  - ...
  - Save Settings
  - Generate  $\rightarrow$  als HTML- und TeX-Dokument nutzbar

## Tipps

## Spickeln

Atmel stellt in seinem Atmel Software Framework (ASF) Module für Teillösungen zur Verfügung, die in die eigene Entwicklung einbezogen werden können. Diese Module sind in der Regel sehr gut dokumentiert und beschreiben die verwendeten Techniken. Lädt man interessante Module z.B. in einem separaten Projekt und lässt die Dokumentation mit Doxygen erstellen, erhält man eine prima API-Beschreibung.

## Zugriff auf Systemdaten gestatten

Bestimmte Security-Software blockiert standardmäßig den Zugriff auf Systemdateien etc. Dieser Zugriff muss für manche Applikationen aus dem Atmel Studio 6 explizit gewährt werden:

- arm-none-eabi-gcc.exe
- cc1.exe
- as.exe
- collect2.exe
- id.exe
- arm-none-eabi-objcopy.exe
- arm-none-eabi-objdump.exe
- arm-none-eabi-size.exe

Einige Funktionen werden erst im Laufe der Zeit bzw. beim Debugging aufgerufen.

## Unit Tests

Atmel bietet mit den ASF ein Testframework (z.B. Unit test framework (driver)) an.

Für Unit Test ist ein separates Projekt in der Solution anzulegen.

Die Tests werden über eine Test-Suite aus der main-Funktion dieses Projektes gestartet. Dieser Suite werden die gewünschten Testfälle bekannt gegeben.

Die Ausgabe der Testergebnisse erfolgt über Wrapper. Denkbar wäre z.B. dbg\_info, das die pc\_send-Funktionen nutzen könnte.

(Note: The test suite framework feeds its output to printf. It is left up to the test application to set up the stream.)

## Item Templates

Zur einheitlichen Gestaltung der Software-Quellen stehen jeweils zwei „Item Templates“ zur Verfügung.

- BiDiB Include Datei
- BiDiB Programm Datei
- BiDiB Include Datei im Doxygen-Stil
- BiDiB Programm Datei im Doxygen-Stil

Die Templates im Doxygen-Stil ermöglichen die Dokumentation mit dem Doxygen-Integrator ([Atmels Doxygen Integrator](#)).

```
//=====
//
// OpenDCC - BiDiB - OneControl // Projektname
//                               // Erstellungsjahr
// Copyright (c) 2013 OpenDCC
//
// This source file is subject of the GNU general public license 2,
// that is available at the world-wide-web at
// http://www.gnu.org/licenses/gpl.txt
//
//-----
//
// file:      power_output.h // Dateiname
// author:    Michael // User
// created:   03.07.2013 16:27:44 // Erstellungsdatum
// contact:   develop@opendcc.de
// website:   http://www.opendcc.de
// history:   DATE USER CHANGED // Bei Änderung ausfüllen
//
//-----
//
// purpose:   TODO Write a short description
//
//=====

#ifndef POWER_OUTPUT_H_
#define POWER_OUTPUT_H_

#endif /* POWER_OUTPUT_H_ */
```

Beispiel: Include Datei (Header)

Die Templates müssen ins Template-Verzeichnis des Atmel Studios kopiert werden (z.B. C:\Users\My Documents\Atmel Studio\Templates\ItemTemplates) und stehen dann im SolutionExplorer durch Add\Item sofort zur Verfügung (siehe auch [How to: Locate and Organize Project and Item Templates](#)).

From:  
<https://forum.opendcc.de/wiki/> - **BiDiB Wiki**

Permanent link:  
<https://forum.opendcc.de/wiki/doku.php?id=bidiboneentwicklungsumgebung&rev=1385143608>

Last update: **2016/07/05 10:47**

