U1 DSP56002 EVM Board Installation

Umfeld

Das DSP56002 EVM (Evaluation Module) ist ein Hilfsmittel zur Einarbeitung und Test einfacher Signalprozessoranwendungen mit dem DSP56002 von Motorola. Der Lieferumfang des Board umfasst:

- Board mit DSP56002 Signalprozessor, 96kB Speicher, Audio Codec, Interface Logik und verschiedenen Schnittstellen und Steckernetzteil.
- DSP56002 Assembler von Motorola
- (DOS) Fensterorientierte Debug / Download Software von Time Domain Technologies

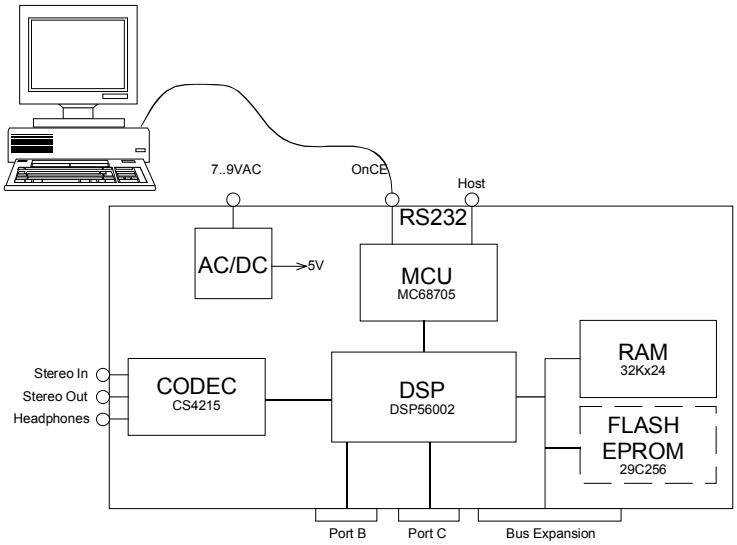


Bild 1-1

Blockschaltbild DSP56002 EVM mit Schnittstellen.

Die Programmerstellung erfolgt auf dem PC unter Windows 9x oder NT/2000 mit Assembler oder C. Der fertige Maschinencode wird über die serielle Debug-Schnittstelle (OnCE) vom PC in das EVM geladen. In ähnlicher Weise wie bei Visual Studio kann das Programm direkt gestartet oder mit Debugfunktionen ausgetestet werden. Ein C-Compiler ist als GNU-C frei verfügbar, umfasst aber nicht den Komfort der kommerziell angebotenen Entwicklungsumgebungen. Ebenso unterstützt die Time Domain Technologies Software nur Debugging auf Stufe Assembler/ Maschinencode.

Unsere Programmierprojekte werden deshalb mit der integrierten Entwicklungsumgebung EDE C5600X der Firma TASKING durchgeführt. Sie beinhaltet einen C/C++-Compiler, Assembler, Linker, Locator und einen recht leistungsfähigen Debugger mit Simulator.

Installation

Folgende SW- Produkte werden erstmalig auf dem PC installiert:

- Tasking C5600X mit Crossview Debugger (ca. 25MB) in geeignetem Verzeichnis.
- MIDI-Files mit Musik zum Testen der Filtereffekte, beliebig auf dem PC.

Diese Softwareprodukte sind ausschliesslich für Ausbildungszwecke bestimmt und dürfen nicht anders genutzt werden.

Bei der Installation ist das DSP56002 EVM und die zu verwendende Kommunikations-schnittstelle mit 115kB oder 230kB auszuwählen.

- Motorola DSP Development Package (ca. 100MB)
- DSP56002 EVM:
  - 1 Board
  - 1 Steckernetzteil
  - 1 Verbindungskabel 9-pol DSub
  - 1 Verbindungskabel Klinkenstecker 3.5mm
  - 1 Kopfhörer

Inbetriebnahme

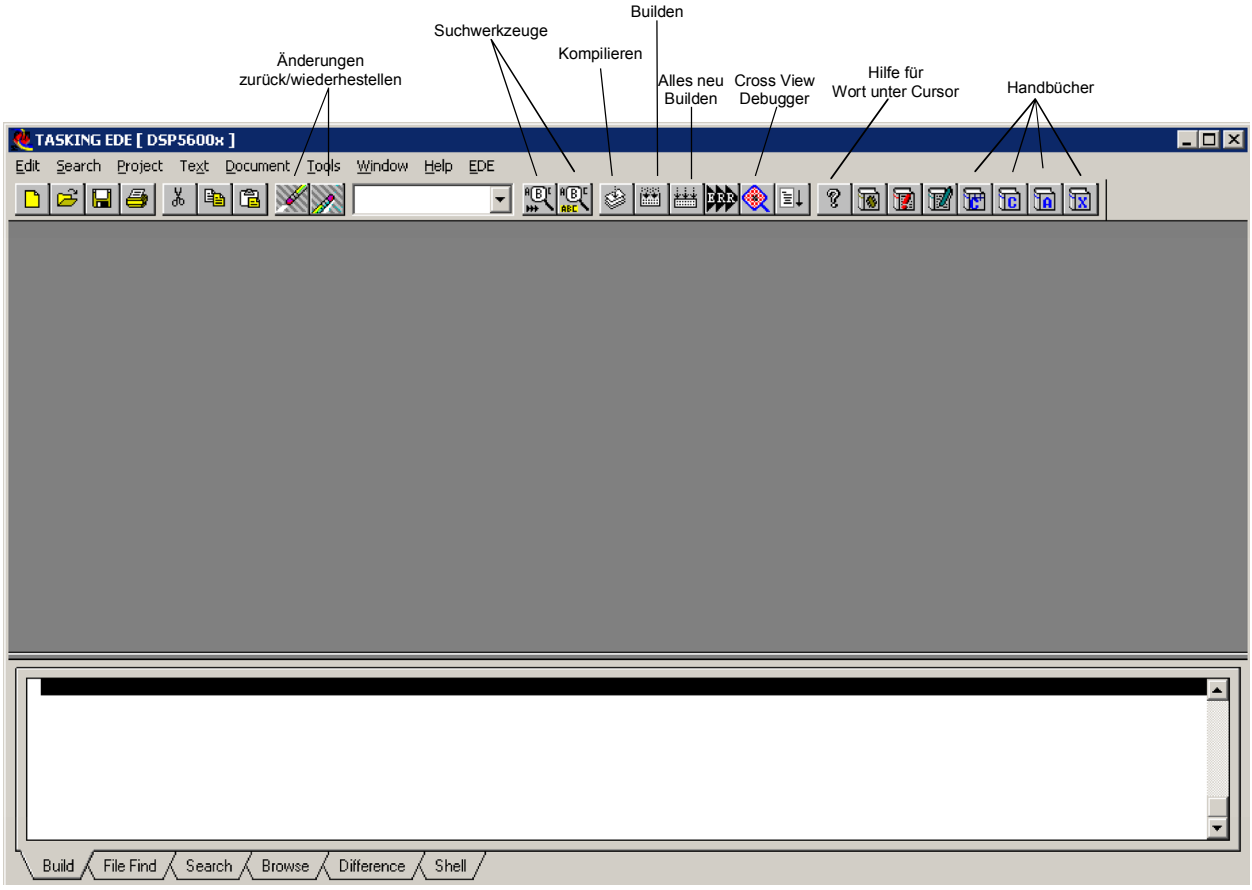
1. OnCE-Schnittstelle (P4) mit 9-poligem Verbindungskabel an der seriellen Schnittstelle am PC anschliessen (COM1: oder COM2:).
2. Steckernetzteil mit 7..9VAC an P1 anschliessen.
3. Signalquelle, z.B. von der Soundkarte des PC, und Kopfhörer anschliessen.
4. PC starten.
5. Tasking EDE starten.
6. Projekt AudioDirect aufsetzen gemäss späterer Wegleitung.
7. Nach erfolgreichem Builden den CrossView Debugger aus der IDE starten (Download beginnt automatisch)
8. Soundquelle einschalten
9. Mit „Go“ das Programm starten.

DSP Programmerstellung

Wichtig: File- und Directoryname dürfen unter der Benutzung mit Tasking-Produkten keine Leerschläge beinhalten!

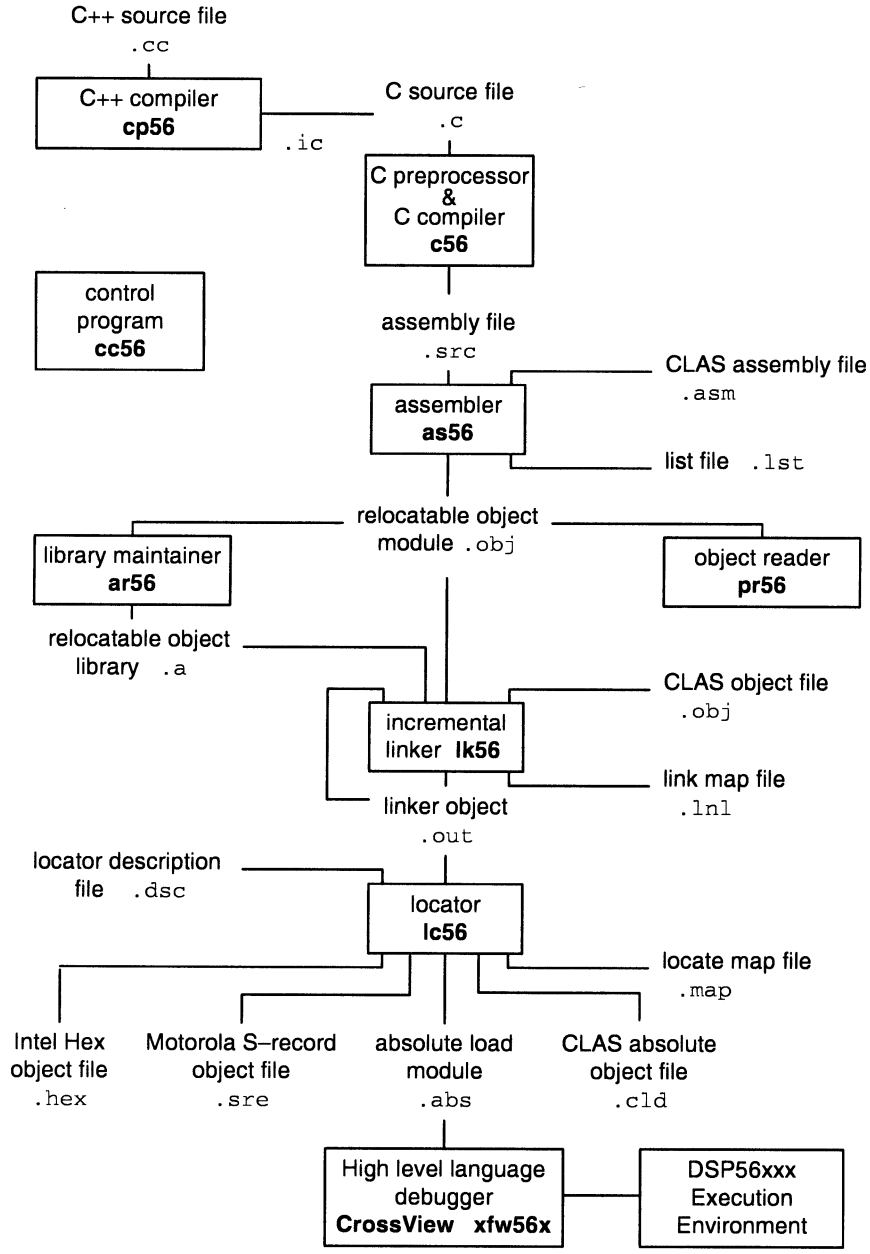
EDE Start

Nach dem Start der Tasking-EDE erscheint ein neue leere Oberfläche:



Wurde die EDE mit einem offenen Projekt verlassen, erscheint automatisch das letzte offene Projekt. Alle Knöpfe haben ToolTip-Texte und weitere Hilfestellung kann aus den umfangreichen Handbüchern gelesen werden.

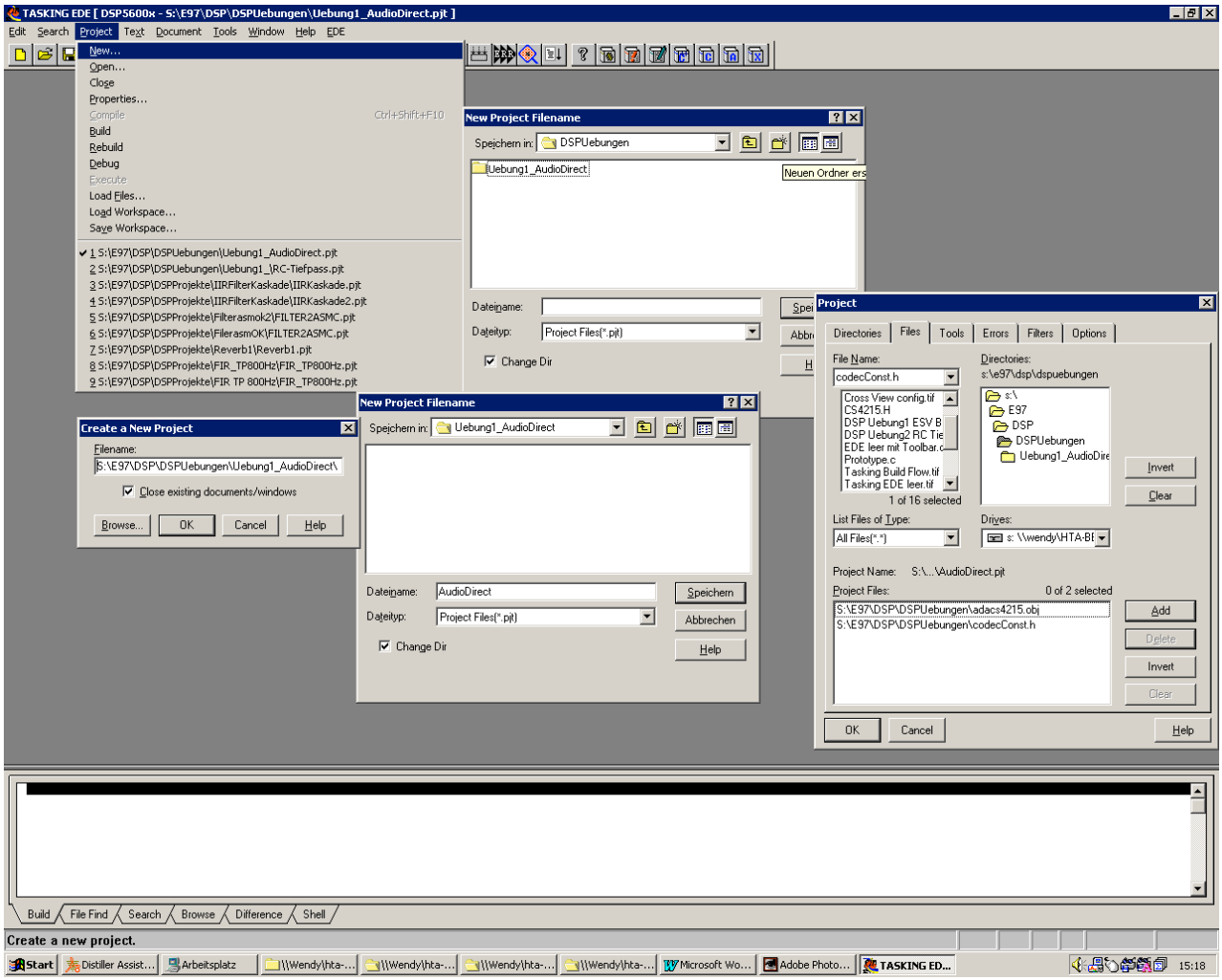
Projekterstellungsablauf, Dateitypen



Zum Download in das DSP56002 EVM kann nur das .abs Format benutzt werden.

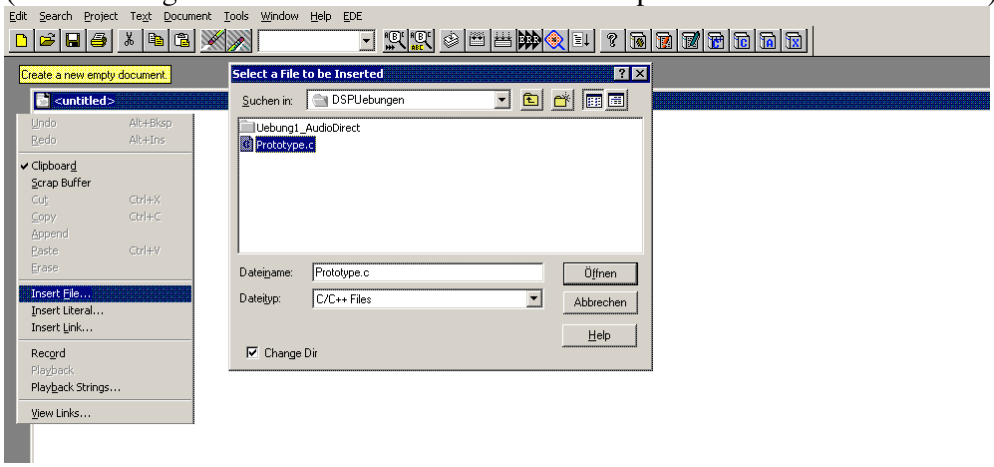
Neues Projekt definieren

Eine C56-Anwendung bedarf immer eines gesamten Projektes. Es umfasst die Quell-, Objekt- und Absolutcodes, Spezialbibliotheken sowie alle Einstellungen zum Projekt, wie Targetplattform, Schnittstellen, Speichermodelle, Editorkonfiguration, u.v.m. Diese Einstellungen werden in der Projektdefinitionsdatei `.prj` festgehalten.

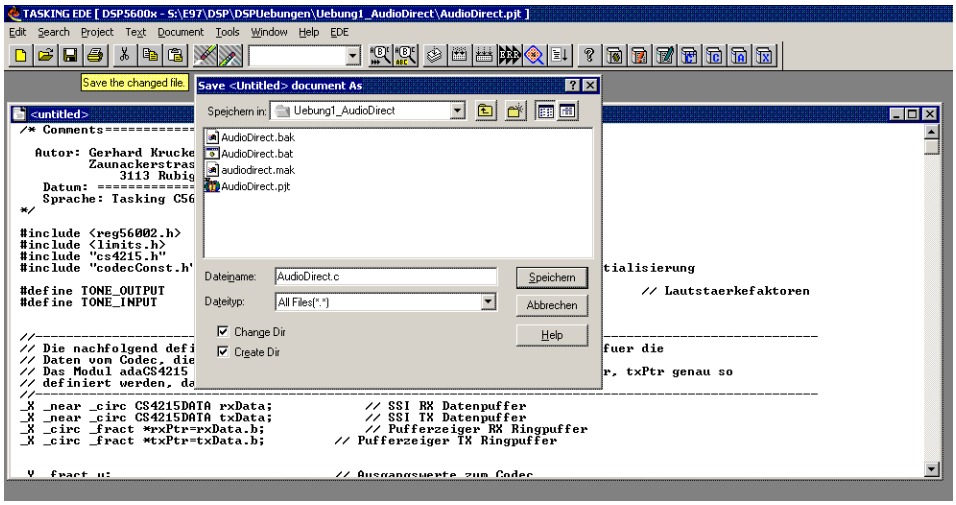


Dem Project ist das Objektfile adacs4215.obj für die Codec-Routinen zuzufügen. Nachher wird der Dialog mit OK abgeschlossen.

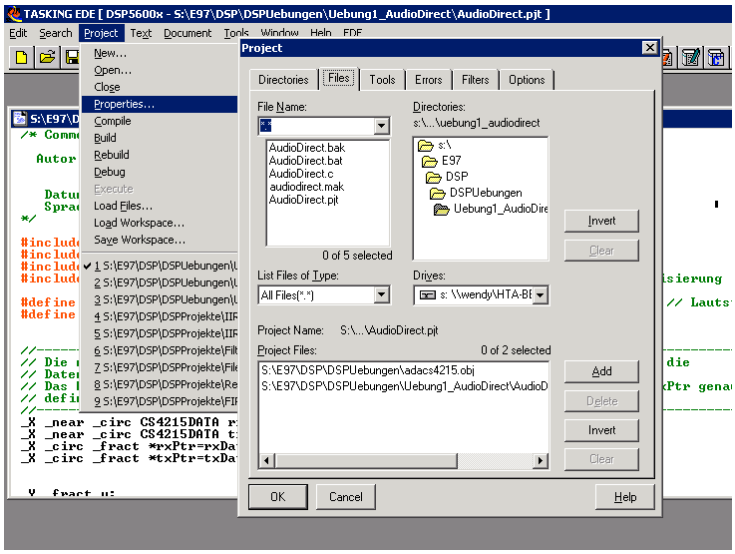
Als nächster Schritt wird ein leeres File erstellt und das Prototypenfile hinein geladen und direkt mit neuem Namen in das Projektverzeichnis abgespeichert:
(Zum Erzeugen des leeren Dokumentes den Knopf in der Toolbar benützen.)



(Zum Speichern des neuen Dokumentes den Knopf in der Toolbar benutzen. Bei einem <untitled>-Dokument wird man zur Eingabe des Namens aufgefordert. Es ist mit der Extension *.c zu speichern. Nach Speicherung erscheint das File syntaxsensitiv eingefärbt.)

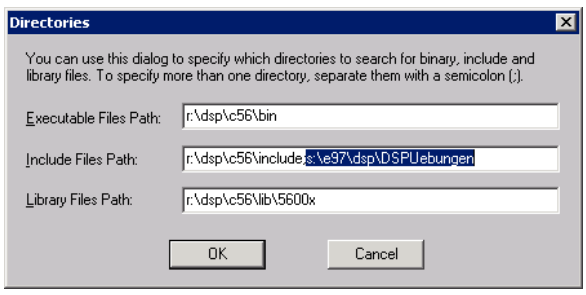


Das neue File ist nun dem Projekt zuzufügen:

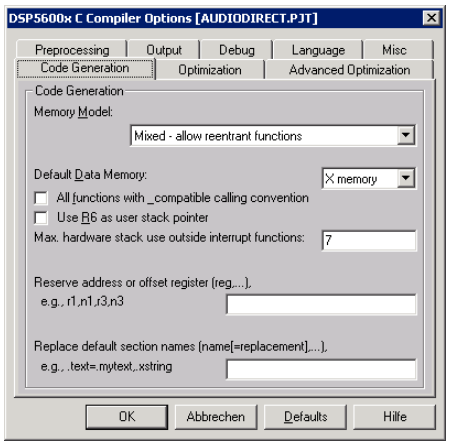


Ferner sind ein Reihe von Konfigurationen durchzuführen:
- Include Directory mit dem Pfad für die benutzerdefinierten #include-Files:

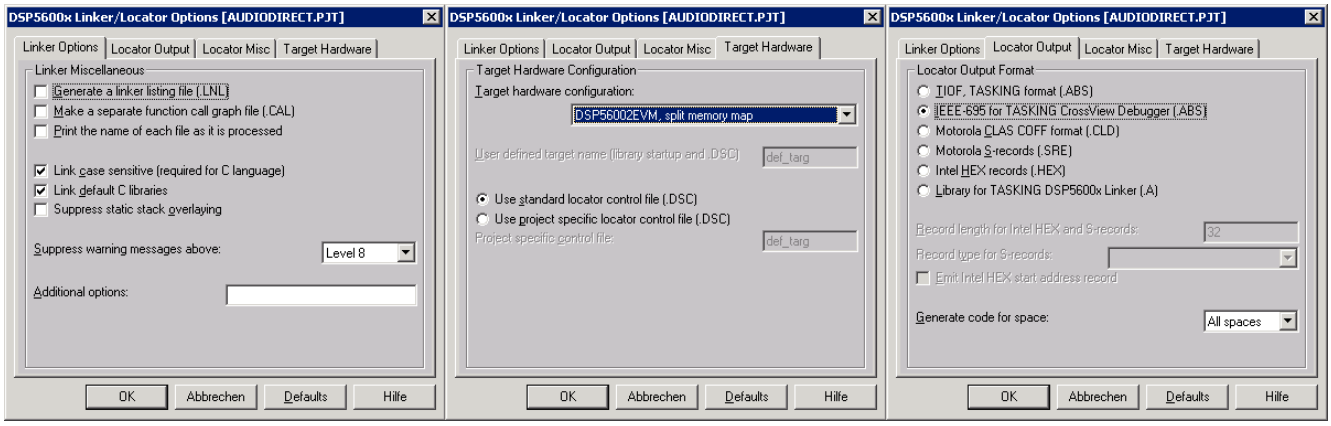
-Directory mit den benutzerdefinierten Files zu setzen: (Menüpunkt EDE/Directories)



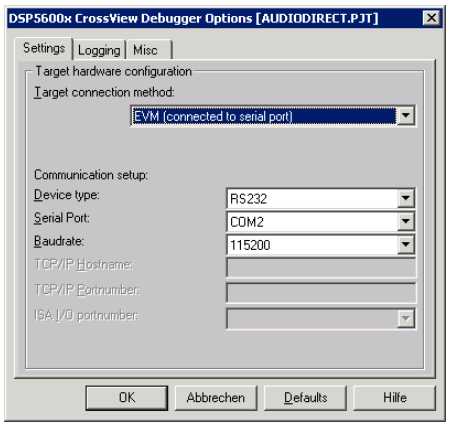
- Kontrolle dass C-Copmpileroptionen korrekt eingestellt sind: (Menüpunkt EDE/C-Compiler)



- Linkeroptionen kontrollieren: (Menüpunkt EDE/Linker Options)

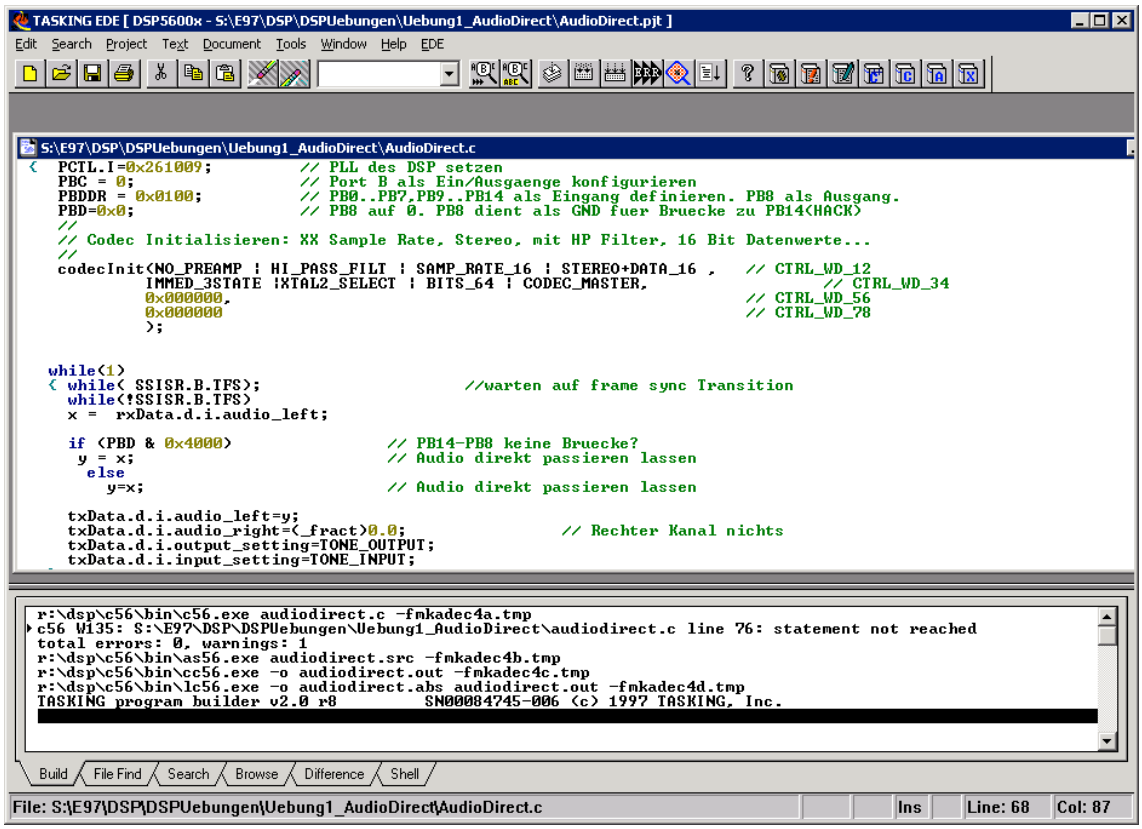


- Crossview Debugger Downloadoptionen: (Menüpunkt EDE/CrossView Options)
Schnittstelle passend wählen und Baudrate auf 115kB setzen, damit Download rasch erfolgt.

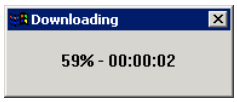


Projekt erstellen (builden)
Nach Setzen aller Optionen kann das Programm codiert und erstellt werden. Das Prototypenfile implementiert das Passieren der Audiodaten, daher sind für diese Übung keine Ergänzungen im Code notwendig.

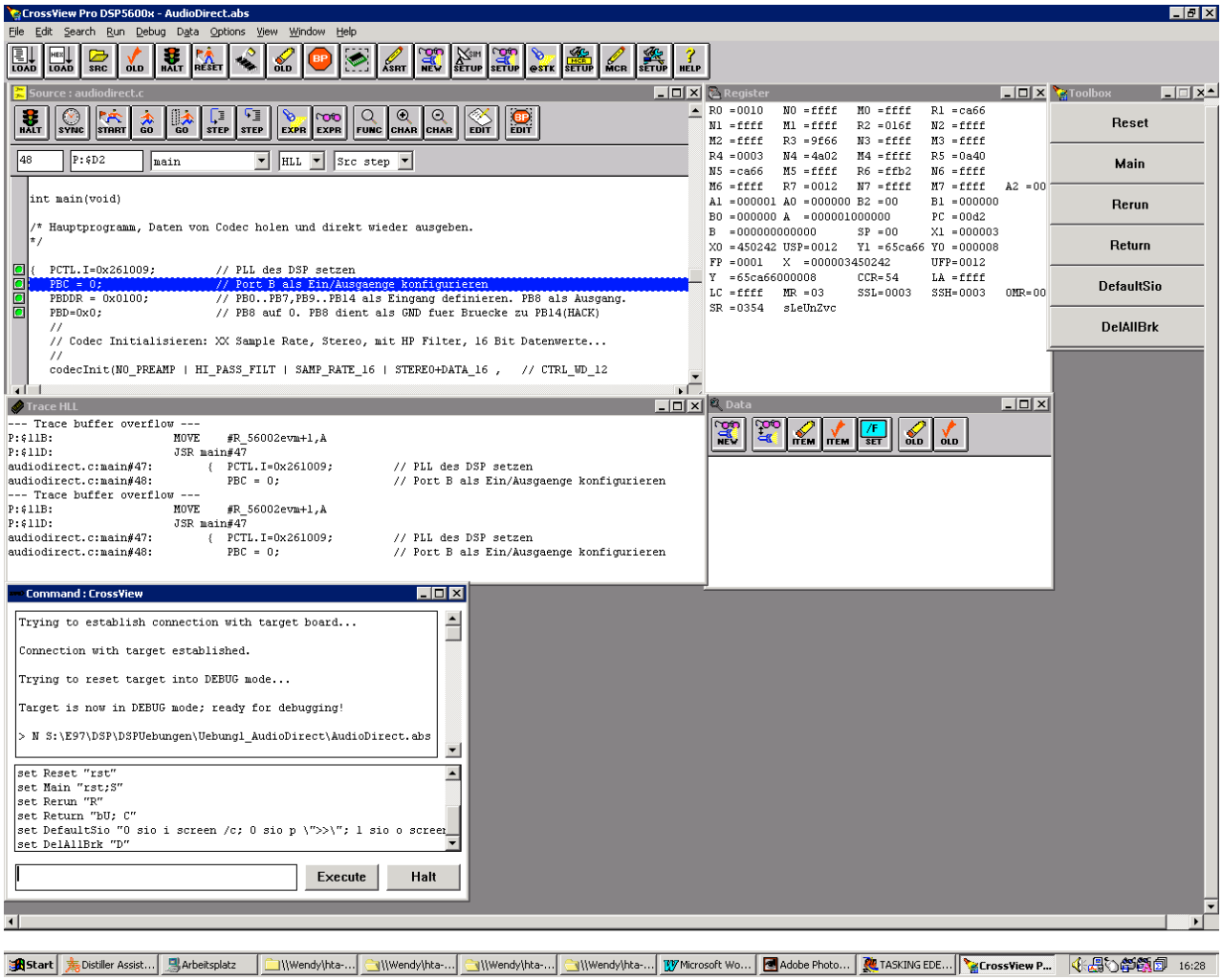
Zum Erstellen (Kompilieren/Linken/Locaten) den Build-Knopf drücken und den Erfolg beurteilen:



Crossview Debugger
Wurden keine Fehler festgestellt, ist das `.abs`-File für den Download erzeugt worden und der Debugger kann gestartet werden. Dazu den CrossView-Knopf in der EDE-Toolbar drücken. Das Debugger Fenster öffnet sich und der Download beginnt.



Anschliessend werden alle Debuggerfenster geöffnet und das Programm kann mit „Go“ gestartet werden. „Halt“ stoppt das laufende Programm.


CrossView Pro DSP5600x - AudioDirect.abs
File Edit Search Run Debug Data Options View Window Help
Source : audiodirect.c
int main(void)
/* Hauptprogramm, Daten von Codec holen und direkt wieder ausgeben.
*/
{ PCTL.I=0x261009; // PLL des DSP setzen
PBC = 0; // Port B als Ein/Ausgaenge konfigurieren
PBDDR = 0x0100; // PB0..PB7,PB9..PB14 als Eingang definieren. PB8 als Ausgang.
PBD=0x0; // PB8 auf 0. PB8 dient als GND fuer Bruecke zu PB14(HACK)
//
// Codec Initialisieren: XX Sample Rate, Stereo, mit HP Filter, 16 Bit Datenwerte...
//
codecInit(NO_PREAMP | HI_PASS_FILT | SAMP_RATE_16 | STEREO+DATA_16 , // CTRL_WD_12
Register
R0 =0010 N0 =ffff M0 =ffff R1 =ca66
N1 =ffff M1 =ffff R2 =016f N2 =ffff
M2 =ffff R3 =9f66 N3 =ffff M3 =ffff
R4 =0003 N4 =4a02 M4 =ffff R5 =0a40
N5 =ca66 M5 =ffff R6 =ffb2 N6 =ffff
M6 =ffff R7 =0012 N7 =ffff M7 =ffff A2 =00
A1 =000001 A0 =000000 B2 =00 B1 =000000
B0 =000000 A =000001000000 PC =00d2
B =000000000000 SP =00 X1 =000003
X0 =450242 USP=0012 Y1 =65ca66 Y0 =000008
FP =0001 X =000003450242 UFP=0012
Y =65ca66000008 CCR=54 LA =ffff
LC =ffff MR =03 SSL=0003 SSH=0003 OMR=00
SR =0354 sLeUnZvc
Toolbox
Reset
Main
Rerun
Return
DefaultSio
DelAllBrk
Trace HLL
--- Trace buffer overflow ---
P:$11B: MOVE #R_56002evm+1,A
P:$11D: JSR main#47
audiodirect.c:main#47: { PCTL.I=0x261009; // PLL des DSP setzen
audiodirect.c:main#48: PBC = 0; // Port B als Ein/Ausgaenge konfigurieren
--- Trace buffer overflow ---
P:$11B: MOVE #R_56002evm+1,A
P:$11D: JSR main#47
audiodirect.c:main#47: { PCTL.I=0x261009; // PLL des DSP setzen
audiodirect.c:main#48: PBC = 0; // Port B als Ein/Ausgaenge konfigurieren
Data
Command : CrossView
Trying to establish connection with target board...
Connection with target established.
Trying to reset target into DEBUG mode...
Target is now in DEBUG mode; ready for debugging!
> N S:\E97\DSP\DSPUebungen\Uebung1_AudioDirect\AudioDirect.abs
set Reset "rst"
set Main "rst;S"
set Rerun "R"
set Return "bU; C"
set DefaultSio "0 sio i screen /c; 0 sio p \">>\"; 1 sio o screen
set DelAllBrk "D"
Execute
Halt
Start
Distiller Assist...
Arbeitsplatz
Microsoft Wo...
Adobe Photo...
TASKING EDE...
CrossView P...
16:28