Application note

Using DoCD debugger with Keil uVision2 tool ver 1.03

All trademarks mentioned in this document
are trademarks of their respective owners.

http://www.DigitalCoreDesign.com
http://www.dcd.pl

1. Overview

The DoCD debugger fully supports Keil uVision2 AGDI interface to hardware debuggers. It allows Keil software users to work within uVision2 environments without switching between Keil compiler and external DoCD software. Everything - compilation, simulation, debugging - is under control of a single application. This document describes how to install and use DoCD debugger with programs written using Keil C/ASM tools.

2. Supplied files

Delivered package contains a complete set of files, needed to run DoCD debugger inside uVision2 interface. The following directories can be found inside package:

.**C51\BIN**\	: *Contains DoCD debugger DLL Target driver file*
.**C51\INC\DCD**\	: *Contains DCD's 8051/80390 include files*
.**C51\EXAMPLES\DoCD**\	: *Contains example C/ASM 8051 application*
.**DATASHTS\DCD**\	: *Contains DCD's 8051/80390 related PDF files*
.**UV2**\	: *Contains DCD's 8051/80390 uVision2 database*

This package is delivered to all existing DCD's customers. You may also order the DoCD FPGA evaluation kit by emailing us directly info@dcd.pl, and receive complete debug system to start working with our 8051/80390 IP Cores.

3. Installation

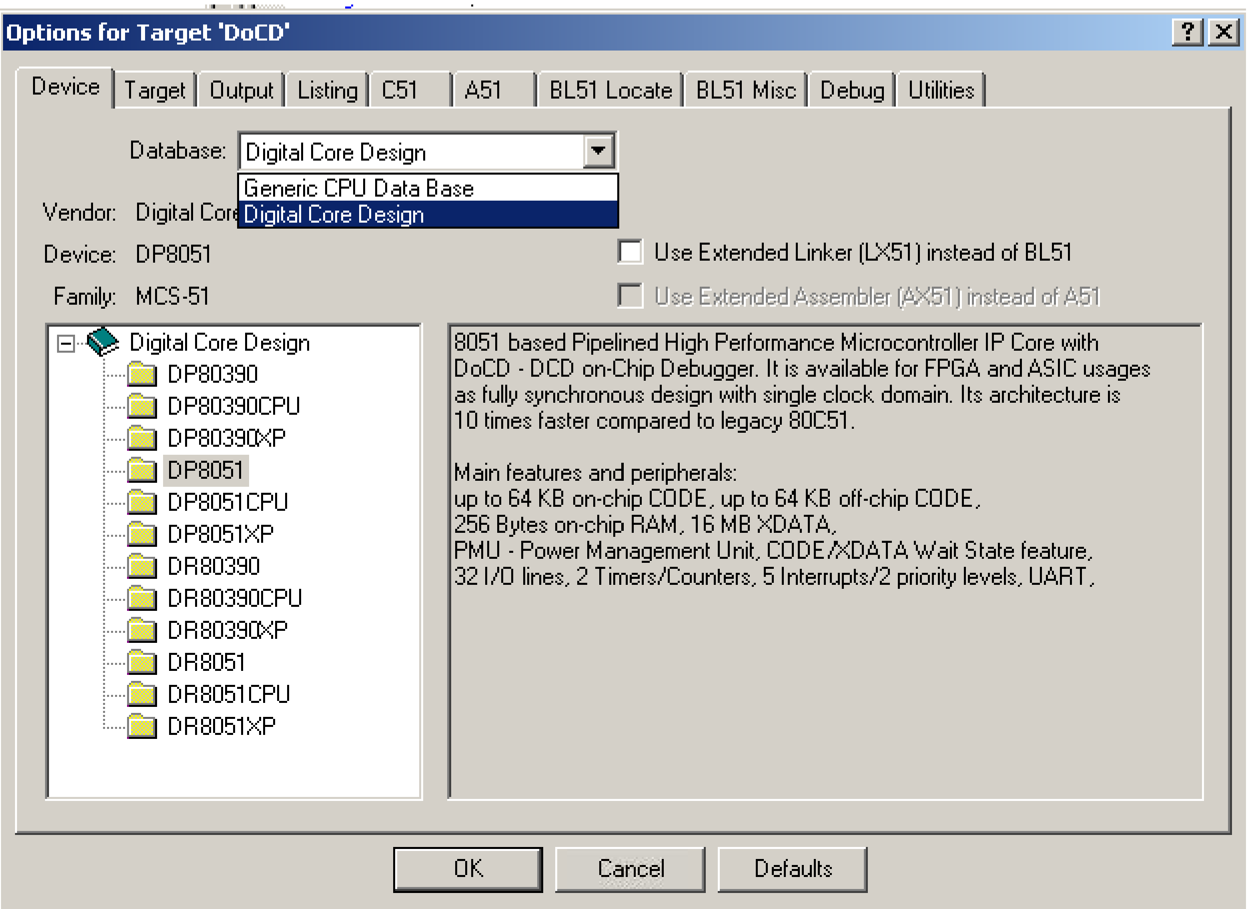
Manual installation of DoCD target driver is no longer required, because package is delivered as executable program, and all steps are performed automatically by setup program. The description provided below is left for informational purposes.
In order to use DoCD target driver follow the steps:

- Install µVision2 and the C51 Compiler on your machine. You may try to use evaluation version of 8051 Keil software at http://www.keil.com/demo/eval/c51.htm
- Unzip the file DoCDTarget.zip (if delivered) inside the *<Keil install path>* folder. The new files should appear inside *<Keil install path>\C51\BIN*, *<Keil install path>\UV2* and *<Keil install path>\C51\EXAMPLES\DoCD* directories
- Add the following lines to the TOOLS.INI file located inside *<Keil install path>* folder:
  - *section [C51]:*
    TDRV0=BIN\DCD_DoCD.DLL ("DoCD 8051 Target Driver")
    Note: if TDRV0 is already in use, then use the next free digit, for example TDRV1
  - *section [UV2]:*
    CDB0=UV2\DCD.CDB("Digital Core Design")
    Note: if CDB0 is already in use, then use the next free digit, for example CDB1

4. PREPARING TO WORK WITH UVISION2 INTERFACE

In order to run your application inside hardware using DoCD target driver follow the steps:

- Run µVision2. Select 'Project – Open Project', the Select Project dialog comes up. Select the 'DoCD.uv2' project. It can be found normally in the folder - *<Keil install path>\C51\EXAMPLES\DoCD*.
- Select 'Options for Target – Devices'. From the 'Database' list-box, select "Digital Core Design" which is our devices database. Select required microcontroller type you actually have. If everything is right, then the dialog should look like this:



Options for Target 'DoCD'

Database: Digital Core Design

Generic CPU Data Base
Digital Core Design

Vendor: Digital Core Design

Device: DP8051

Family: MCS-51

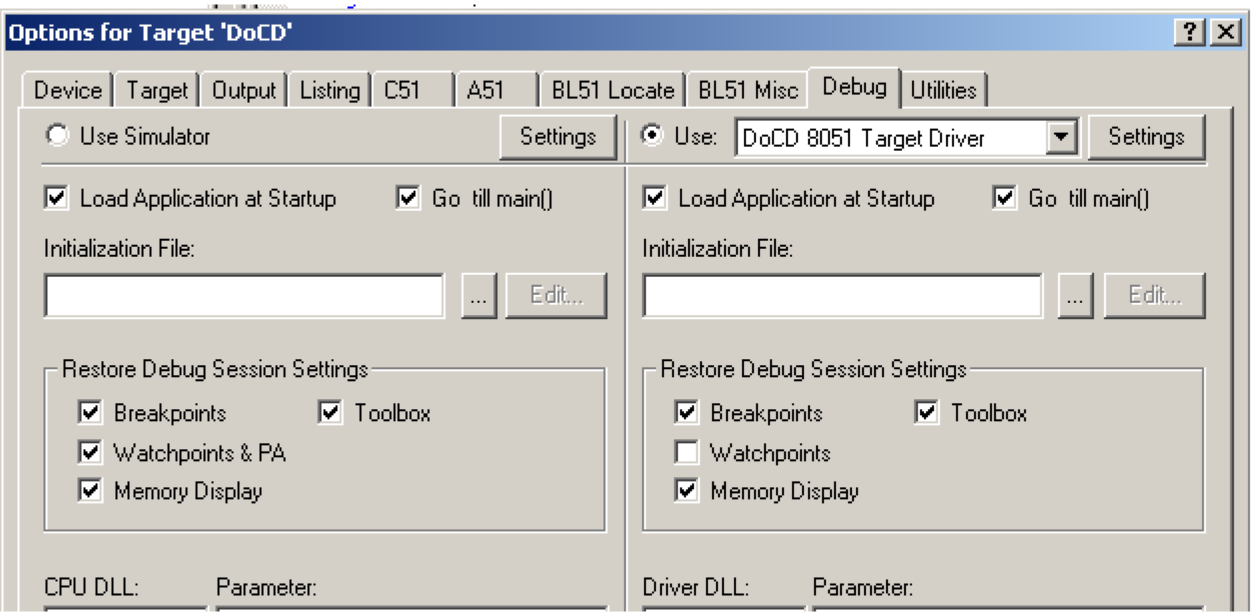
☐ Use Extended Linker (LX51) instead of BL51

☐ Use Extended Assembler (AX51) instead of A51

8051 based Pipelined High Performance Microcontroller IP Core with DoCD - DCD on-Chip Debugger. It is available for FPGA and ASIC usages as fully synchronous design with single clock domain. Its architecture is 10 times faster compared to legacy 80C51.

Main features and peripherals:
up to 64 KB on-chip CODE, up to 64 KB off-chip CODE,
256 Bytes on-chip RAM, 16 MB XDATA,
PMU - Power Management Unit, CODE/XDATA Wait State feature,
32 I/O lines, 2 Timers/Counters, 5 Interrupts/2 priority levels, UART,

- Select 'Rebuild all target files' to build the project.
- Select 'Options for Target – Debug'. From the combo-box, select "DoCD 8051 Target Driver" which is our DoCD driver. Make sure that the 'Use:' radio button is checked. If everything is right, then the dialog should look like this:


Options for Target 'DoCD'

○ Use Simulator

☑ Load Application at Startup ☑ Go till main()

Initialization File:

Restore Debug Session Settings

☑ Breakpoints ☑ Toolbox

☑ Watchpoints & PA

☑ Memory Display

CPU DLL: Parameter:

● Use: DoCD 8051 Target Driver

☑ Load Application at Startup ☑ Go till main()

Initialization File:

Restore Debug Session Settings

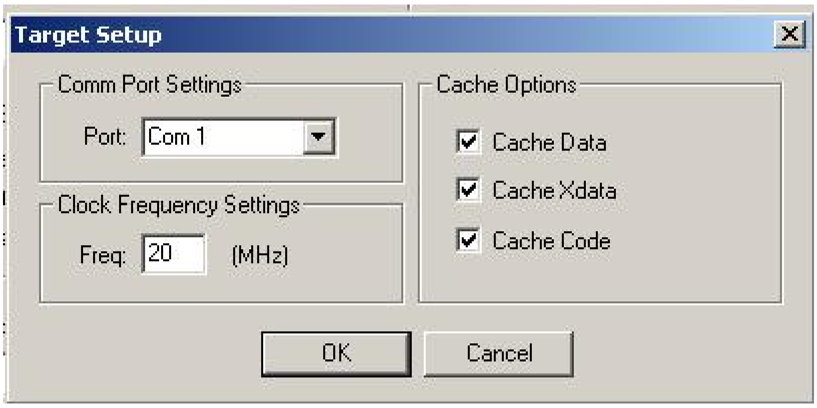
☑ Breakpoints ☑ Toolbox

□ Watchpoints

☑ Memory Display

Driver DLL: Parameter:

- Select 'Settings' button. The 'Target Setup' dialog comes up.


Target Setup

Comm Port Settings

Port: Com 1

Clock Frequency Settings

Freq: 20 (MHz)

Cache Options

☑ Cache Data

☑ Cache Xdata

☑ Cache Code

Choose a COM port used by HAD board, enable at least CODE 'Caching Options'. Enter the actual clock frequency applied on-board to global CLK pin of 8051 IP Core. Allowed values are 1 MHz to 250 MHz. It means that DoCD will be working properly with CLK clock frequencies 2 times lower and 2 times higher than entered value.

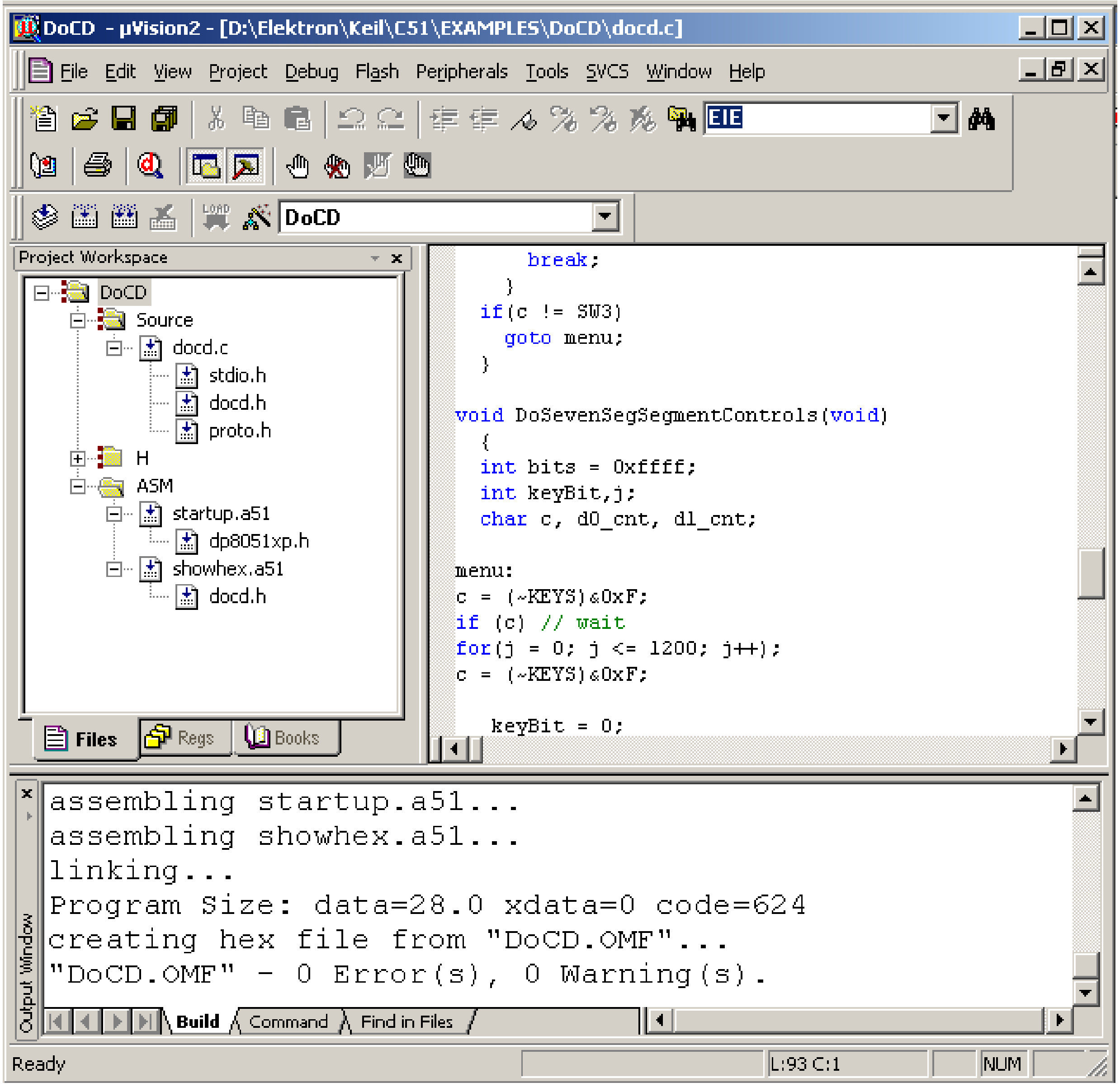
- Close the both dialogs using 'OK' button. To discard all changes use 'Cancel' button.

5. Debugging an example project

This chapter describes how to use DoCD debugger with 8051 C/ASM application basing on example DoCD project. It is assumed that the installation and configuration steps had been performed. ***You should also have DCD's 8051 core with Debug IP running inside some hardware prototype board equipped with FPGA or ASIC chip.***

Please follow the steps below:

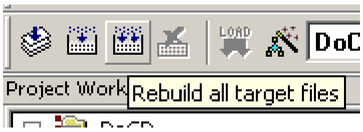
- Run µVision2. Select 'Project – Open Project', the Select Project dialog comes up. Select the 'DoCD.uv2' project. It can be found normally in the folder - *<Keil install path>\C51\EXAMPLES\DoCD*. The dialog should look like this:



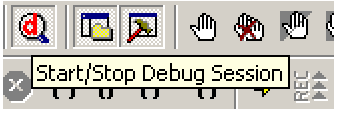
```
      break;
    }
  if(c != SW3)
    goto menu;
  }

void DoSevenSegSegmentControls(void)
  {
  int bits = 0xffff;
  int keyBit,j;
  char c, d0_cnt, d1_cnt;

menu:
c = (~KEYS)&0xF;
if (c) // wait
for(j = 0; j <= 1200; j++);
c = (~KEYS)&0xF;

   keyBit = 0;
```

```
assembling startup.a51...
assembling showhex.a51...
linking...
Program Size: data=28.0 xdata=0 code=624
creating hex file from "DoCD.OMF"...
"DoCD.OMF" - 0 Error(s), 0 Warning(s).
```

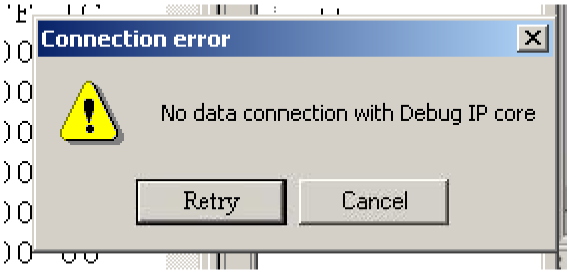
- Select 'Rebuild all target files' to build the project. The build window shown above should report no warnings and errors.



- Start debugging by selecting 'Start/Stop Debug session'



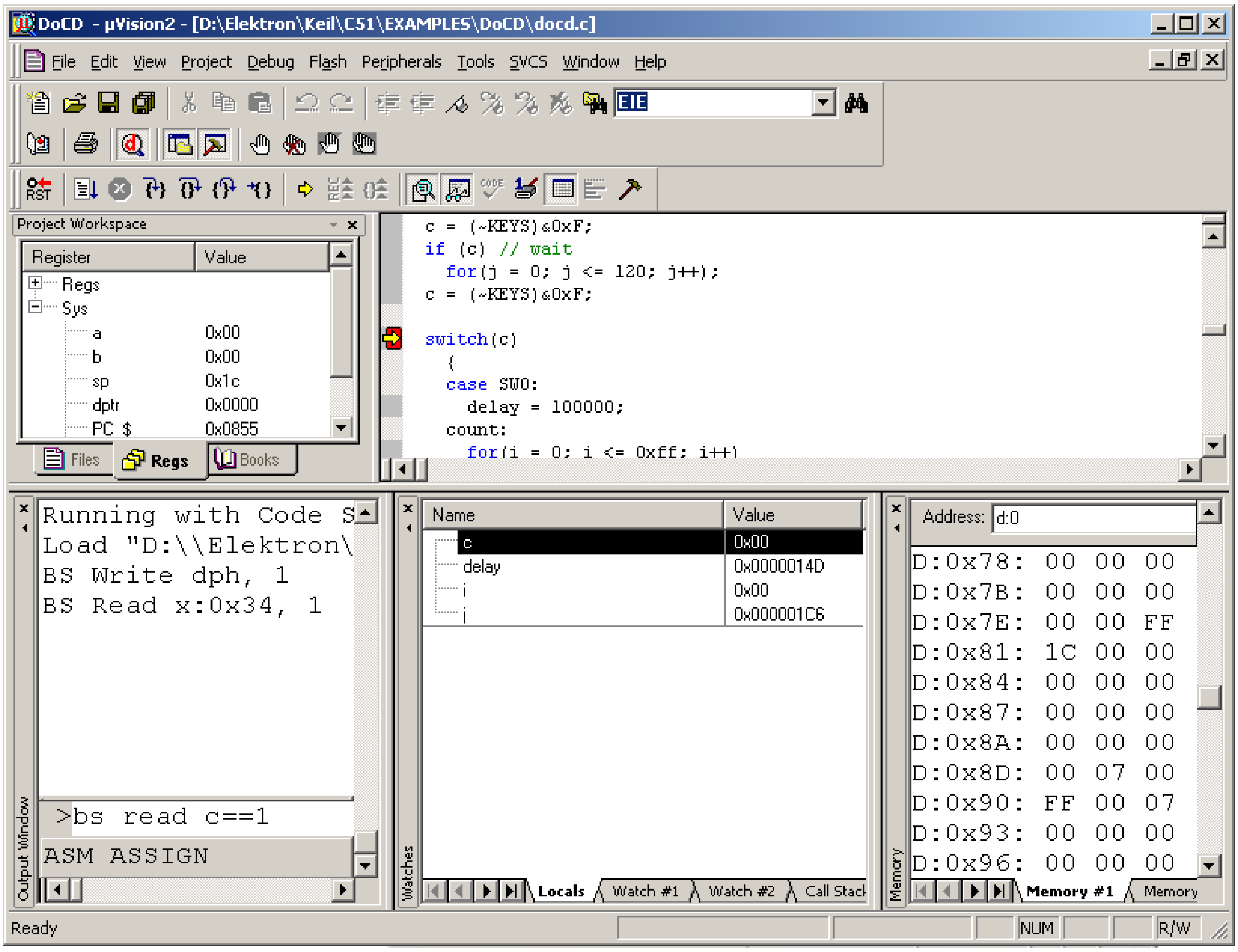
- A warning dialog box may possibly be displayed after 5 seconds, if some communication problems with hardware board would be encountered.



No data connection with Debug IP core

In this case the cables should be checked for proper connection, and 'Retry' button pressed. 'Cancel' button closes debugging session.

- When hardware is properly configured, and program successfully uploaded/downloaded, then the debugger window should look similarly to picture below:



Register	Value
Regs	
Sys	
a	0x00
b	0x00
sp	0x1c
dptr	0x0000
PC $	0x0855

```
c = (~KEYS)&0xF;
if (c) // wait
  for(j = 0; j <= 120; j++);
c = (~KEYS)&0xF;

switch(c)
  {
  case SW0:
    delay = 100000;
  count:
    for(i = 0; i <= 0xff; i++)
```

```
Running with Code S
Load "D:\\Elektron\
BS Write dph, 1
BS Read x:0x34, 1

>bs read c==1
ASM ASSIGN
```

Name	Value
c	0x00
delay	0x0000014D
i	0x00
j	0x000001C6

Address: d:0

```
D:0x78:  00  00  00
D:0x7B:  00  00  00
D:0x7E:  00  00  FF
D:0x81:  1C  00  00
D:0x84:  00  00  00
D:0x87:  00  00  00
D:0x8A:  00  00  00
D:0x8D:  00  07  00
D:0x90:  FF  00  07
D:0x93:  00  00  00
D:0x96:  00  00  00
```

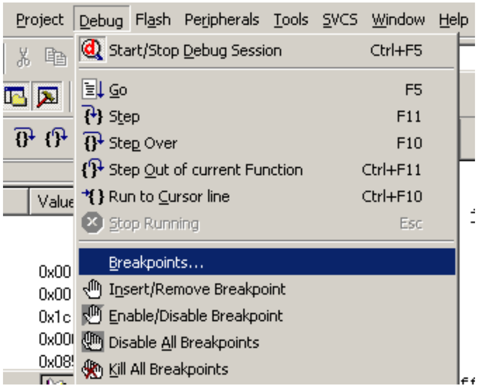
- At this point debugging process is similar as in uVision2 simulator. The program can be run, halted, run step by step, breakpoints can be set/cleared. Variables can be watched, memory areas read/written/modified.

6. NOTES

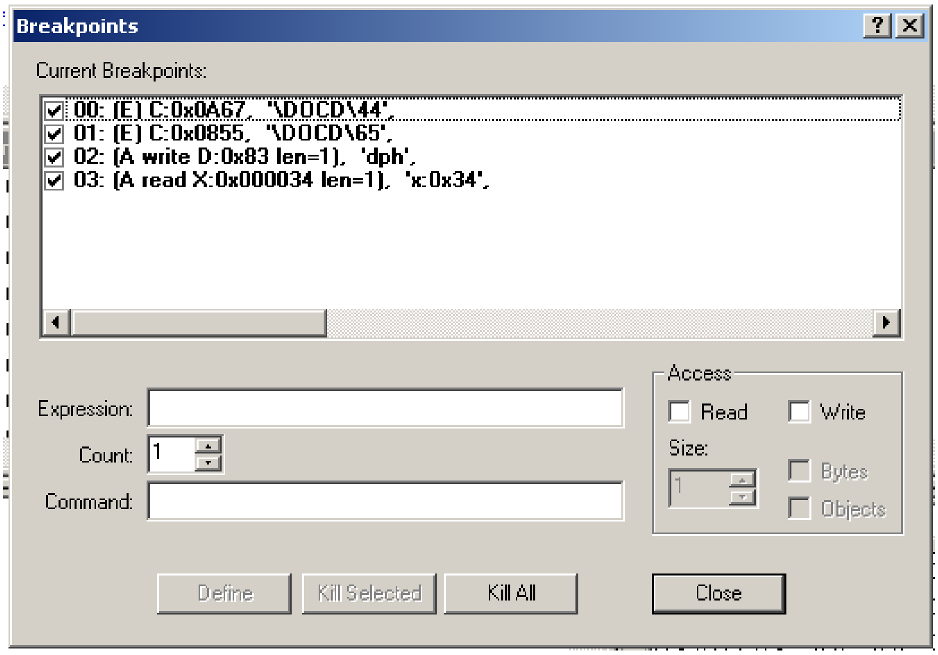
The following limitations are applied to the uVision2 debugger:

1. There are 4 independent complex hardware breakpoints available. There can be set using command line option or graphical boxes of uViison2 tool while debugging session is active:
   - at CODE space (PC value). Example command line:
     BS main - breakpoint at *main* address
     The only one breakpoint is active, regardless of number of set breakpoints (a red square displayed at the left side of C/ASM source editor).
   - at DATA space (READ, WRITE only). Example command line:
     BS READ delay - break when delay variable is being read
   - at SFR space (READ, WRITE only). Example command line:
     BS WRITE DPH - break when DPH SFR register is being written
   - at XDATA space (READ, WRITE only). Example command line:
     BS WRITE x:0x34 - break when 0x34 address is being written

   Please refer to uVision2 Debug Commands help file. Conditional breakpoints are not supported. To manage breakpoints select 'Debug | Breakpoints...' menu:


Project Debug Flash Peripherals Tools SVCS Window Help
Start/Stop Debug Session Ctrl+F5
Go F5
Step F11
Step Over F10
Step Out of current Function Ctrl+F11
Run to Cursor line Ctrl+F10
Stop Running Esc
Breakpoints...
Insert/Remove Breakpoint
Enable/Disable Breakpoint
Disable All Breakpoints
Kill All Breakpoints
Value
0x00
0x00
0x1c
0x00
0x08

   You should see the Breakpoints window. The selected breakpoint can be enabled, disabled, defined, or deleted.



☑ 00: (E) C:0x0A67, '\DOCD\44',
☑ 01: (E) C:0x0855, '\DOCD\65',
☑ 02: (A write D:0x83 len=1), 'dph',
☑ 03: (A read X:0x000034 len=1), 'x:0x34',

2. Code is automatically uploaded by uVision2 while debugging session is being started, and is assumed that CODE memory is writtable as SRAM. Please check DCD's 8051 IP Cores specification for more details about Program Memory writes.
3. In case of read only CODE memory (ROM, FLASH), user is responsible for proper image loading into physical chip holding CODE. The CODE inside hardware memory must be identical with CODE used in uVision2 software. In other case debugging won't work correctly.

7. REFERENCES

The DoCD uVision2 debugger package is delivered by DCD free of charge upon customer request. The customer should have DCD's 8051 IP Core equipped with DoCD on chip debugger, and HAD communication board. DoCD debug system is available at DCD web-site http://www.dcd.pl.

The Keil 8051 uVision2 tool suite is also required. It can be obtained from Keil GmbH at http://www.keil.com.

8. CONTACTS

For any questions or comments please contact DCD.

Headquarters:
Wroclawska 94

41-902 Bytom, POLAND

e-mail: info@dcd.pl

tel. : +48 32 282 82 66

fax : +48 32 282 74 37

Distributors:

Please check http://www.dcd.pl/apartn.php