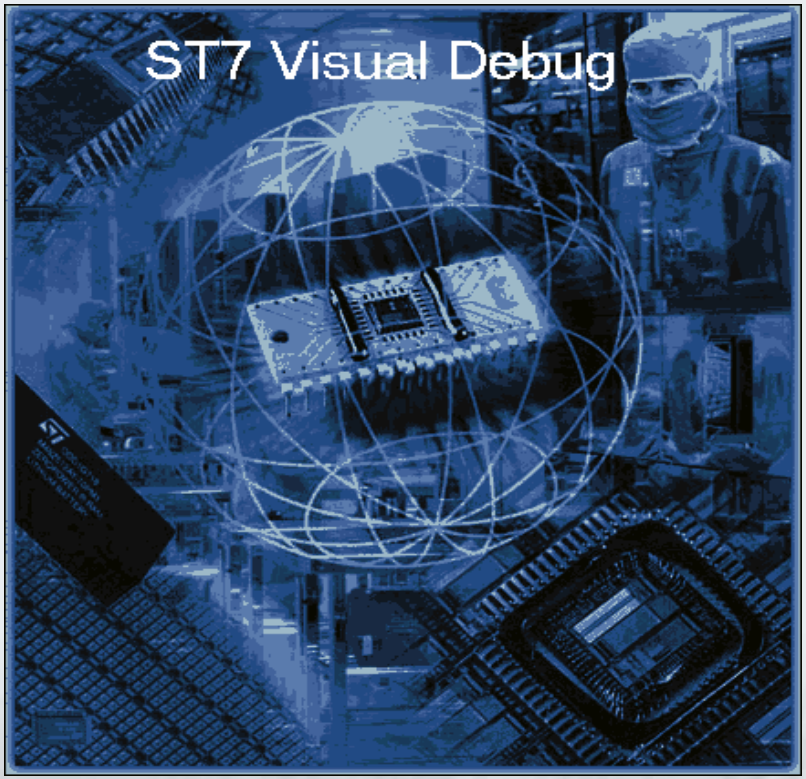
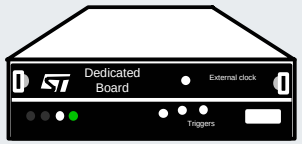
ST7 MICROCONTROLLER TRAINING

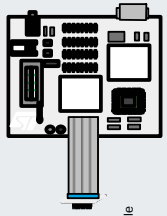
1. INTRODUCTION
2. CORE
3. ADDRESSING MODES
4. ASSEMBLY TOOLCHAIN
5. STVD7 DEBUGGER
6. HARDWARE TOOLS
7. PERIPHERALS
8. ST-REALIZER II
9. C TOOLCHAINS

STVD7: ST WINDOWS DEBUGGER FOR ST7 MCUs

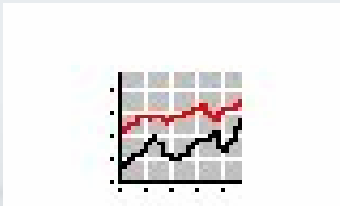




Emulator



Development kit

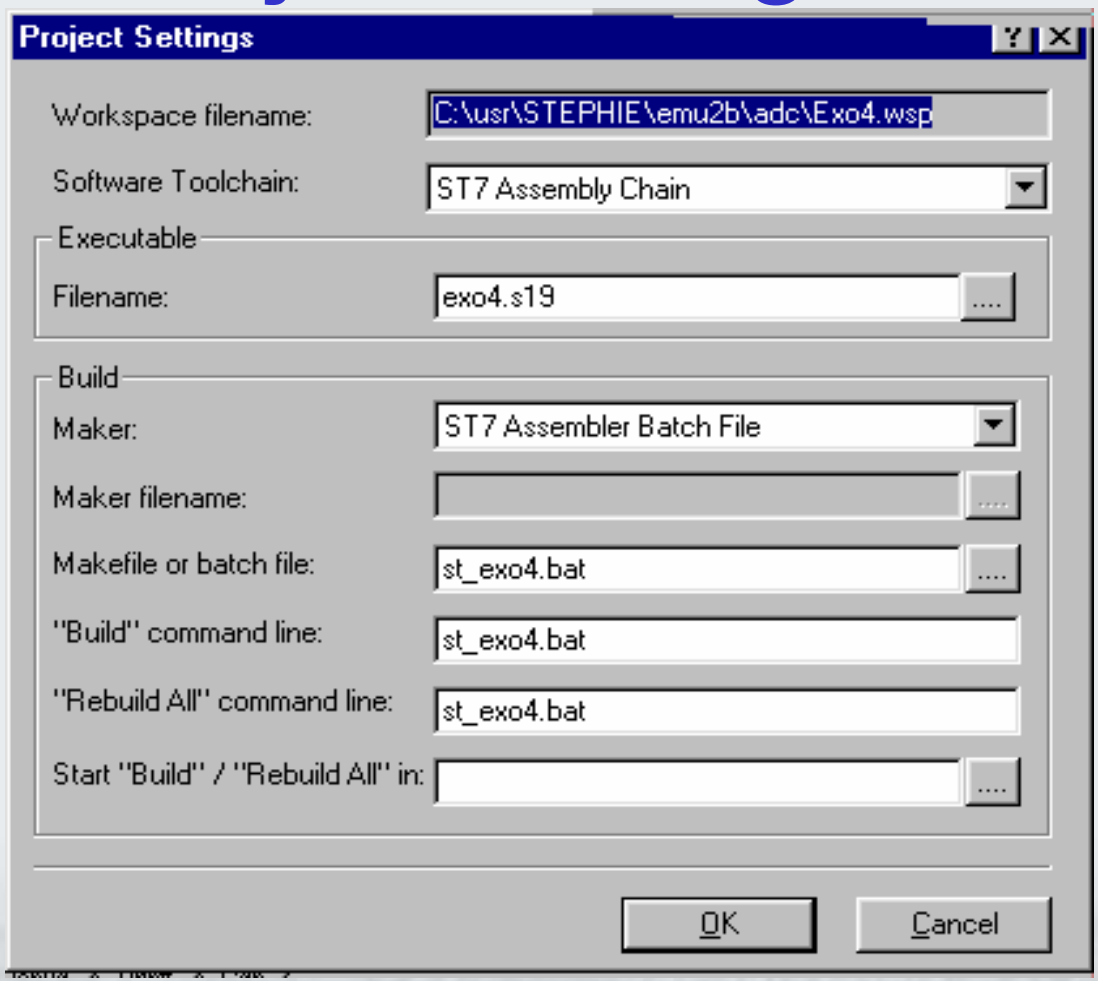


Simulator

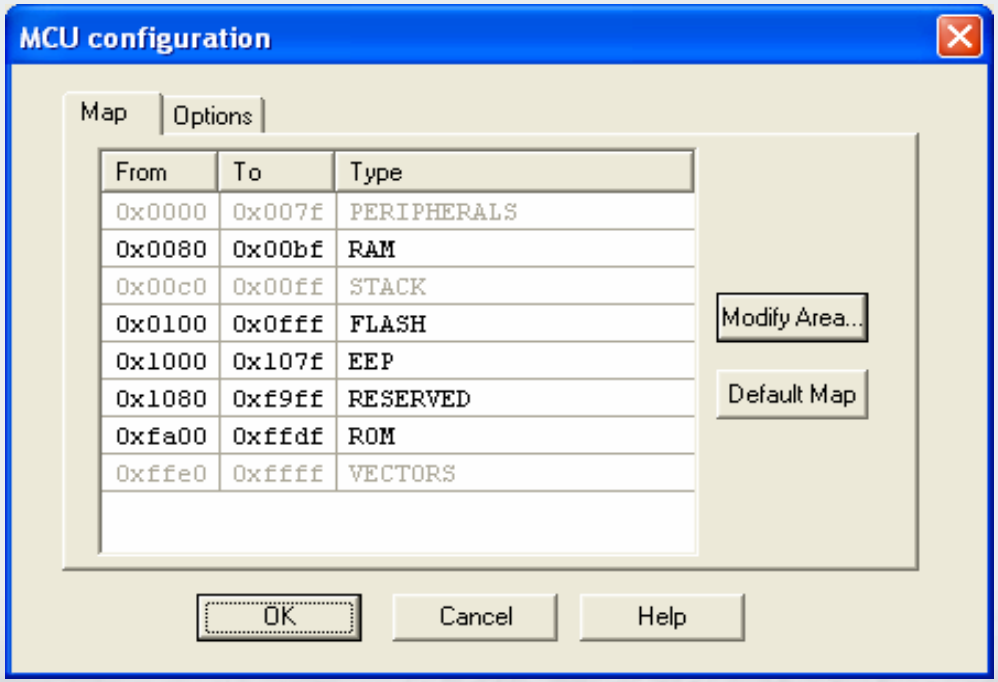
ST7 Visual Debug Overview

- Hardware Development System Graphical Interface (32-bit)
- User Friendly Source Level Debugging
- Compatible with:
  - **ST Macro-Assembler Tool Chain**
  - **Hiware/Metrowerks C Compiler Tool Chain**
  - **COSMIC C Compiler Tool Chain**
- Complete ST7 Family Support
- Real Time Trace
- Breakpoints, Logic Analyser, Hardware Events...
- Version 3.x available: FULL IDE version

ST7 Visual Debug
Project settings

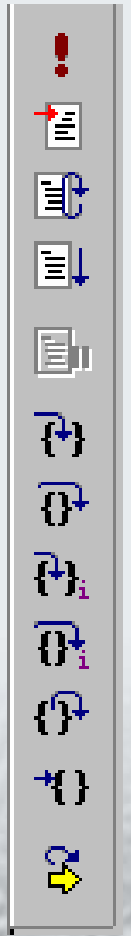


ST7 Visual Debug
Micro Configuration



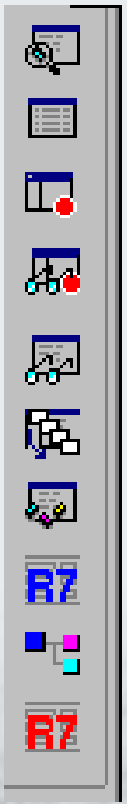
ST7 Visual Debug
Debugging Features

- Running the application
  - **Run**
  - **Chip Reset**
  - **Restart Application**
  - **Continue**
  - **Stop**
  - **Step into**
  - **Step over**
  - **Step int ASM**
  - **Step over Asm**
  - **Step out**
  - **Run to Cursor**
  - **Set PC**

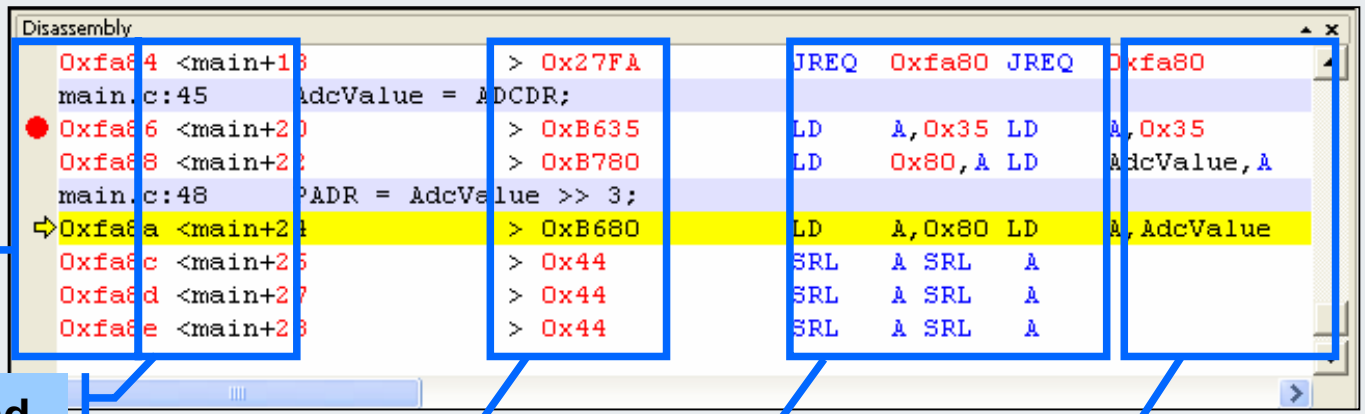


ST7 Visual Debug
Debugging Features

- Common Debug Features
  - **Disassembly**
  - **Memory Dump**
  - **Instruction Breakpoints**
  - **Data Breakpoints**
  - **Watch**
  - **Call Stack**
  - **Local Variables**
  - **ST7 Registers Window**
  - **Symbol browser**
  - **Peripheral registers window**



Disassembly window



Location

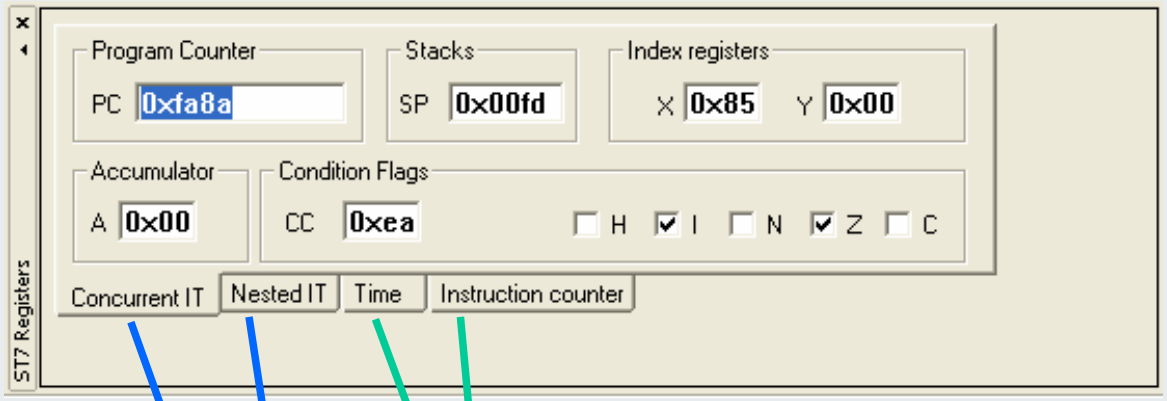
Associated function and offset

Hexadecimal view

Disassembly

Symbolic Disassembly

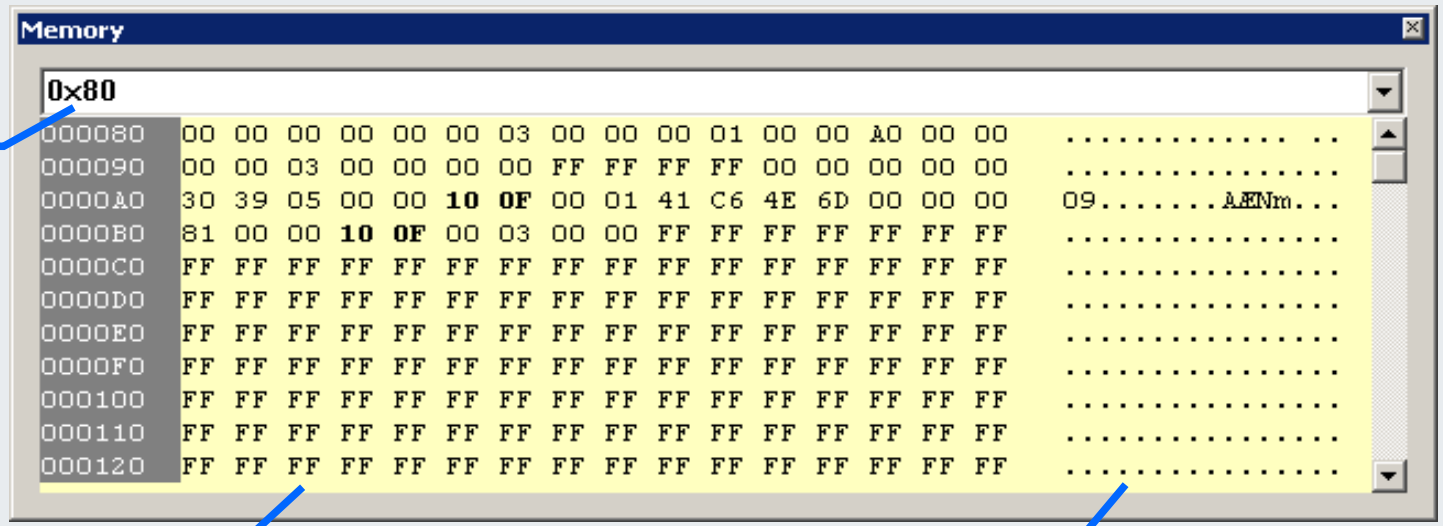
ST7 registers



Simulator-specific pseudo registers

Two ways to interpret CC registers

Memory

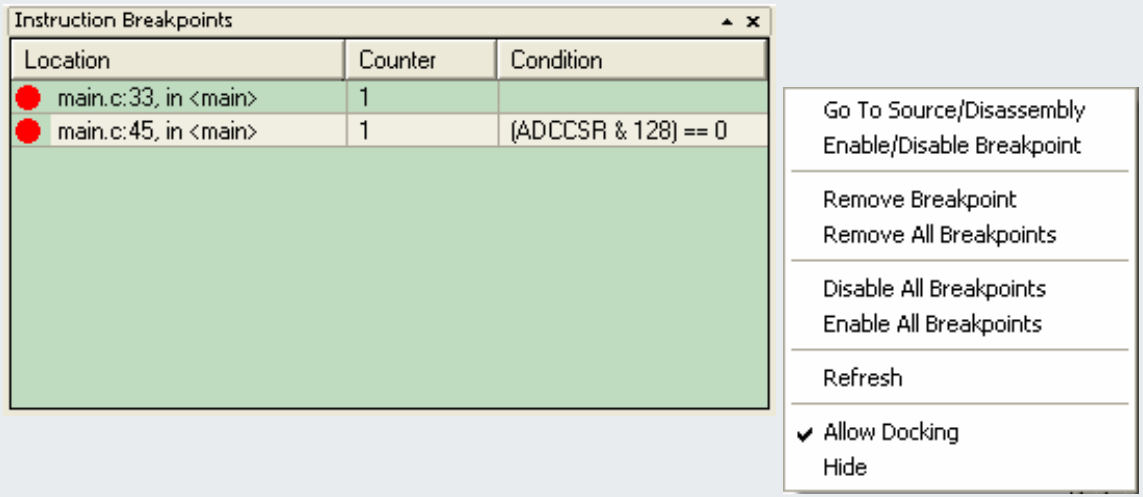


Starting address of the dumped memory block

Binary Dump

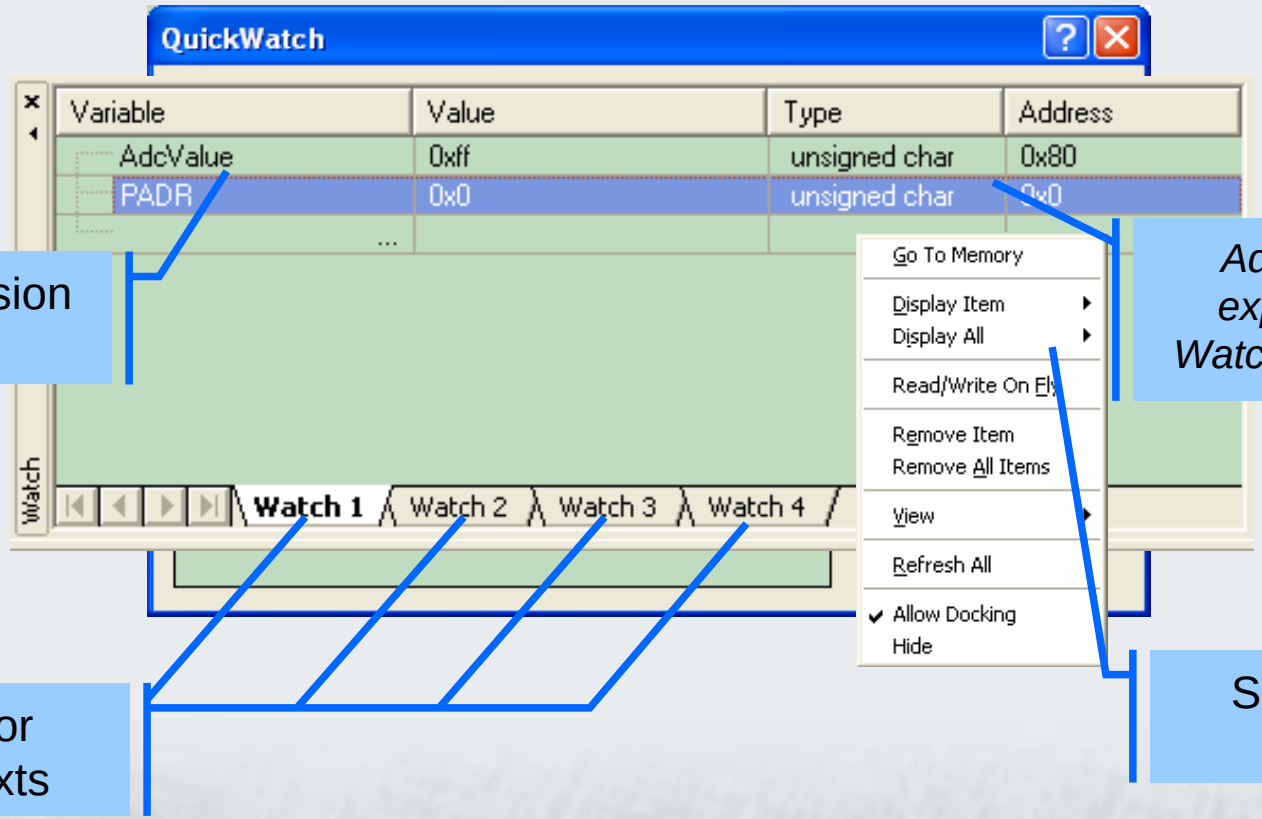
ASCII dump

Instruction breakpoints



- Displays all instruction breakpoints (Active/Inactive)
- Enter counters and conditions

Quick Watch and Watch



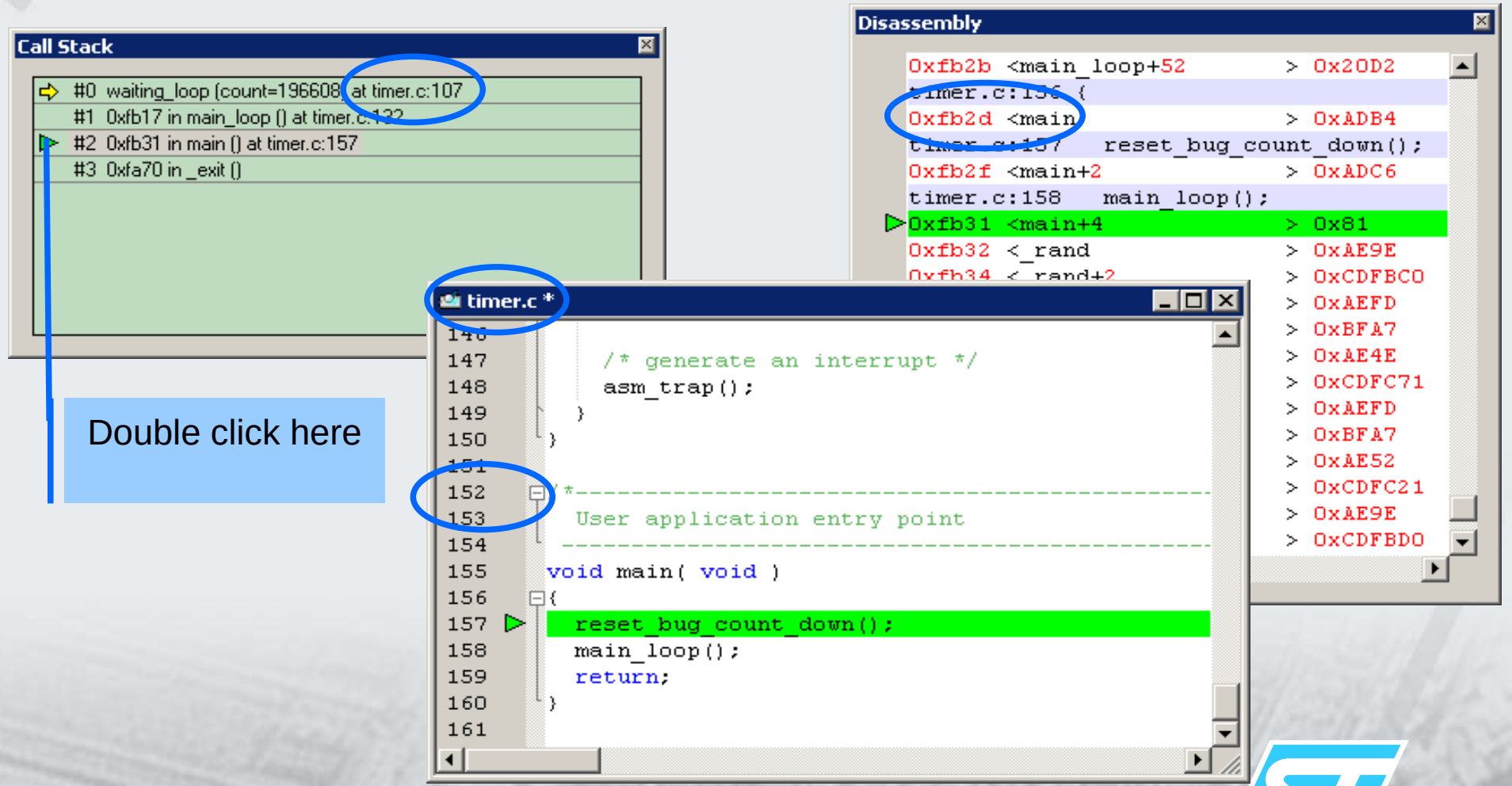
Evaluated expression

Add the selected expression to the Watch window display

Multiple tabs for Different contexts

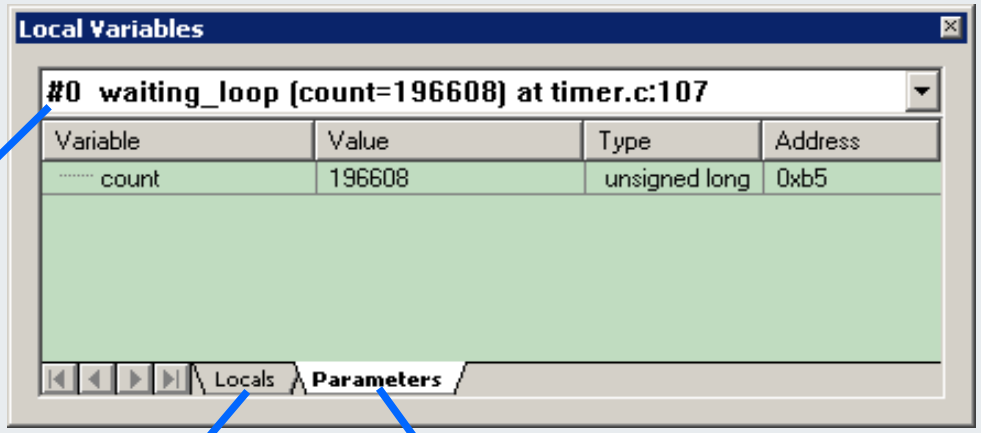
Select display format

Call stack


Call Stack
#0 waiting_loop (count=196608) at timer.c:107
#1 0xfb17 in main_loop () at timer.c:132
#2 0xfb31 in main () at timer.c:157
#3 0xfa70 in _exit ()
Double click here
Disassembly
0xfb2b <main_loop+52 > 0x20D2
0xfb2d <main > 0xADB4
reset_bug_count_down();
0xfb2f <main+2 > 0xADC6
timer.c:158 main_loop();
0xfb31 <main+4 > 0x81
0xfb32 <_rand > 0xAE9E
> 0xCDFBC0
> 0xAEFD
> 0xBFA7
> 0xAE4E
> 0xCDFC71
> 0xAEFD
> 0xBFA7
> 0xAE52
> 0xCDFC21
> 0xAE9E
> 0xCDFBD0
timer.c *
147 /* generate an interrupt */
148 asm_trap();
149 }
150 }
151
152 /*----------
153 User application entry point
154 ----------
155 void main(void)
156 {
157 reset_bug_count_down();
158 main_loop();
159 return;
160 }
161

Local variables

Call Stack

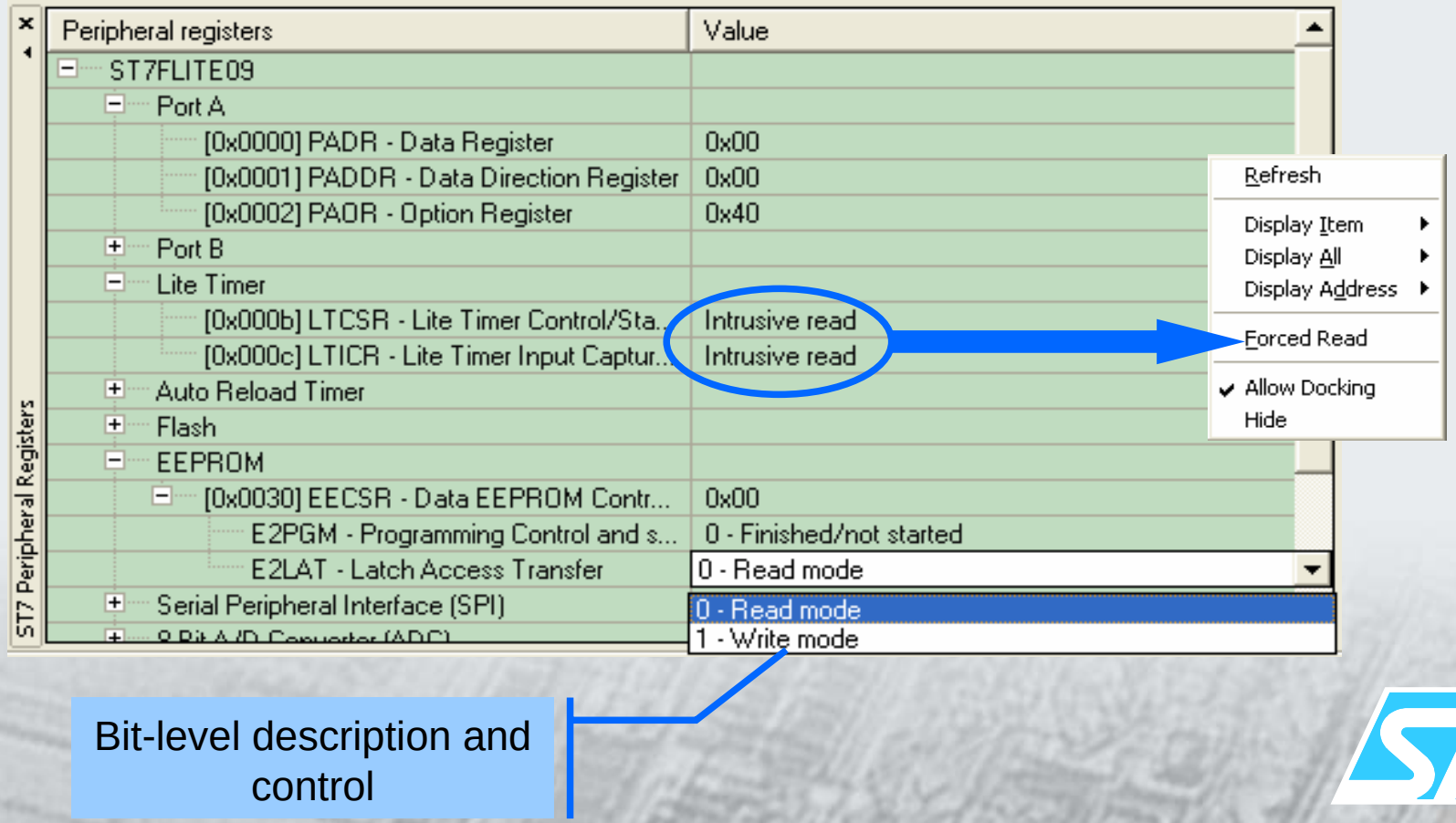


Auto detection of Local Variables and…

… function Parameters

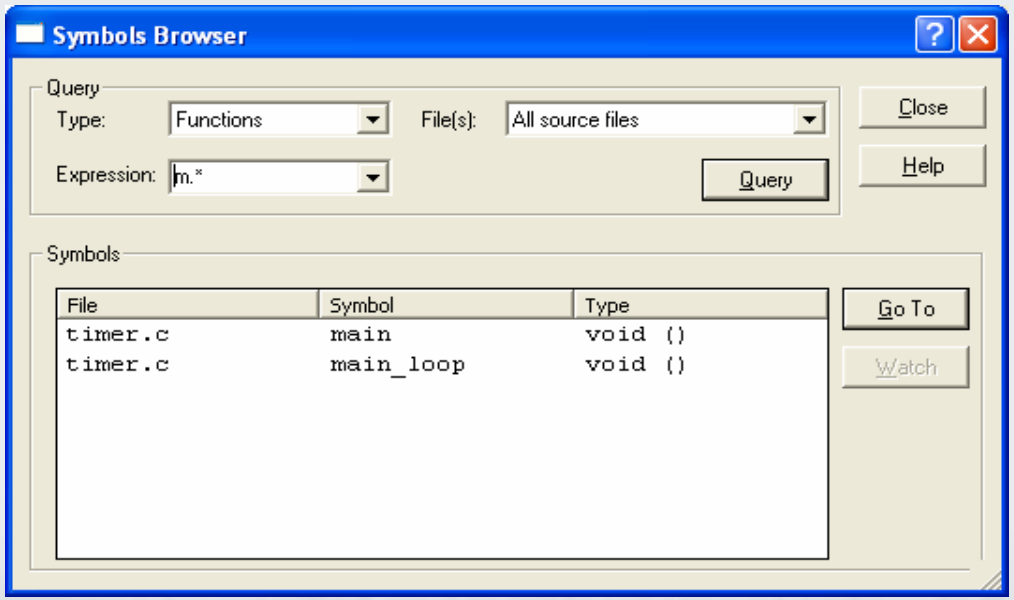
ST7 Visual Debug
Debugging Features

- Peripheral registers window:
  - **To visualise/ change peripheral's values**
  - **Accessing to some status register is not allowed**



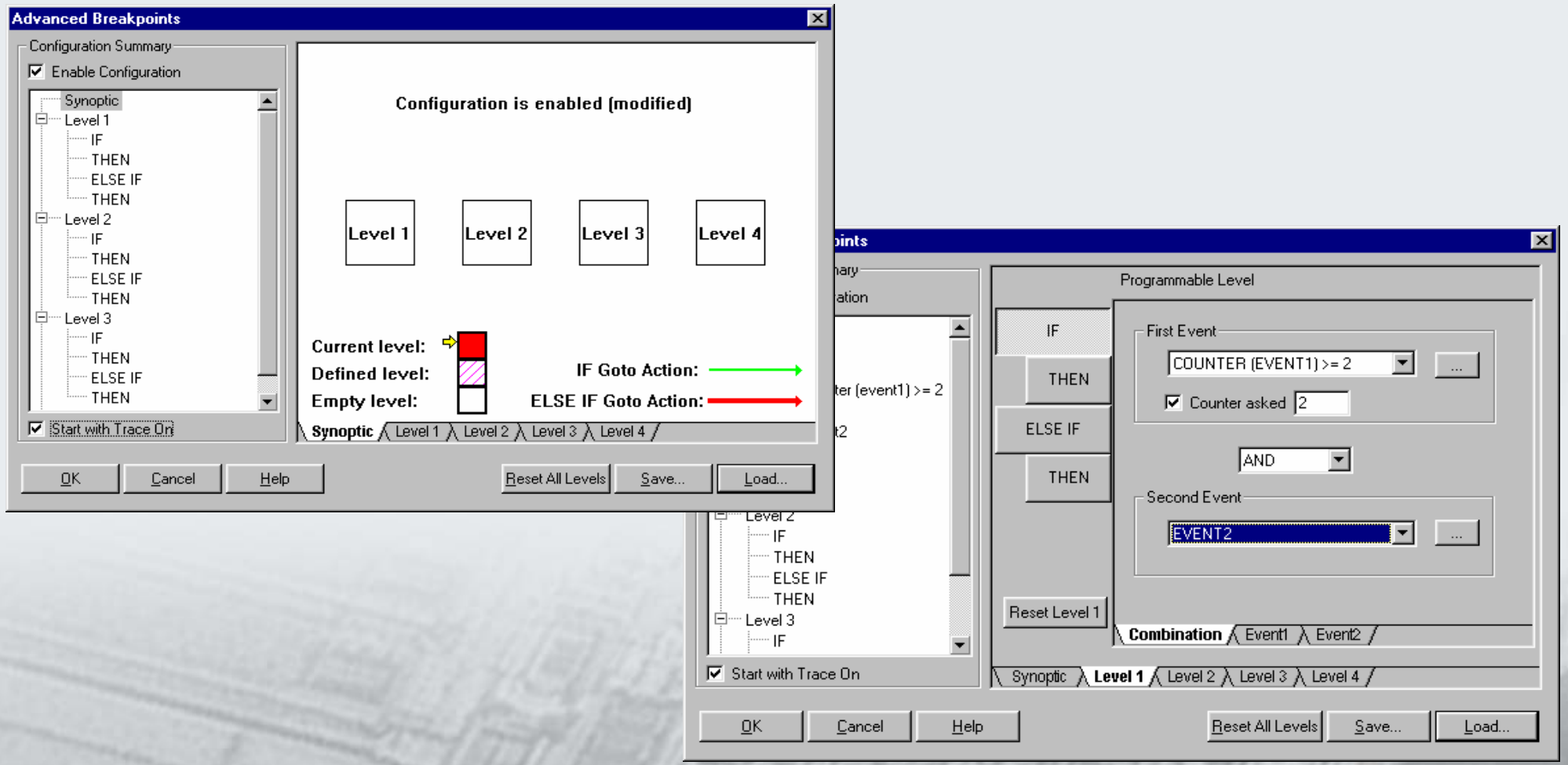
Symbol browser

- Based on debug information



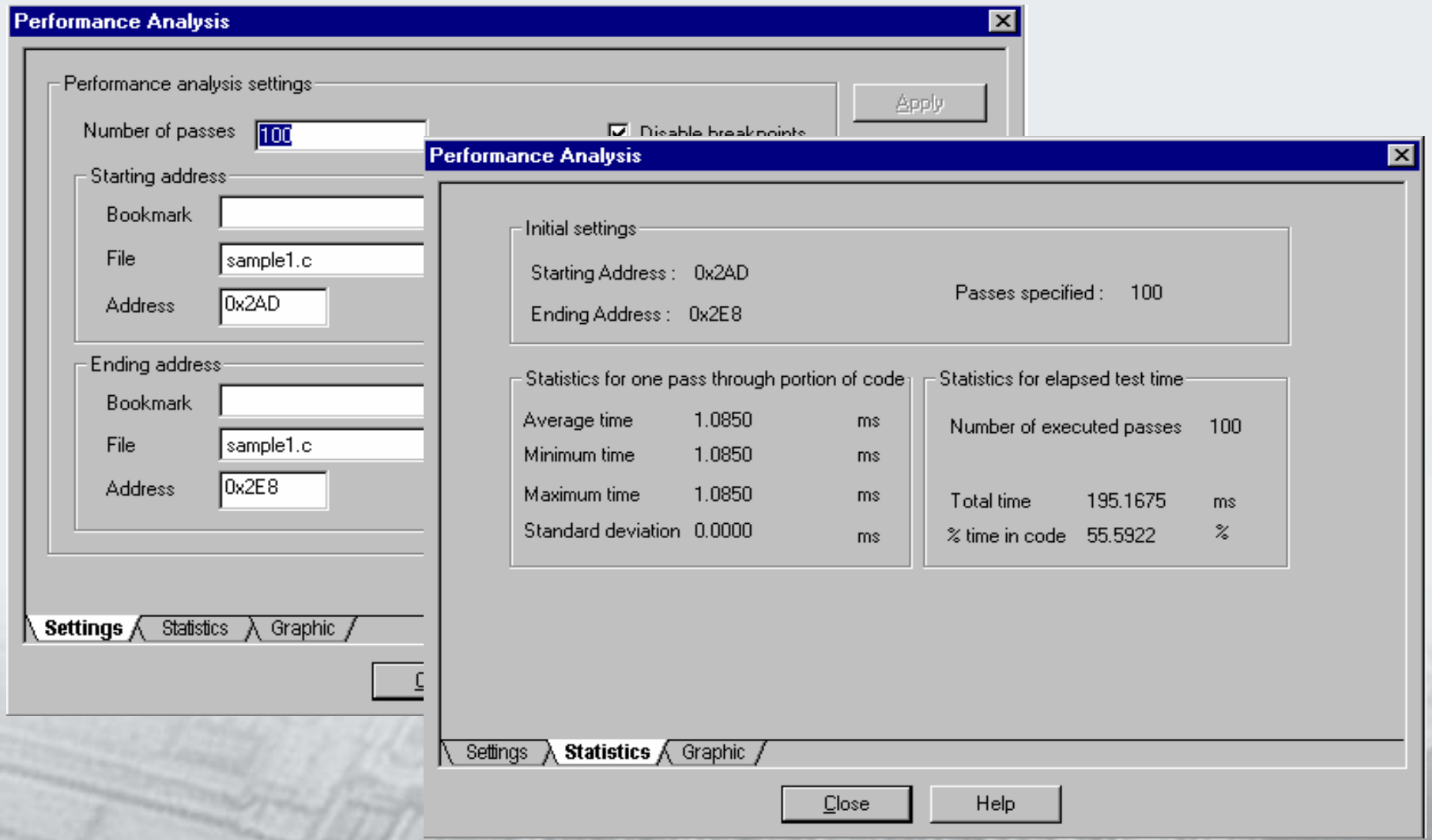
ST7 Visual Debug
Emulator-Specific functions

- **BEM (Bus Event Machine)**



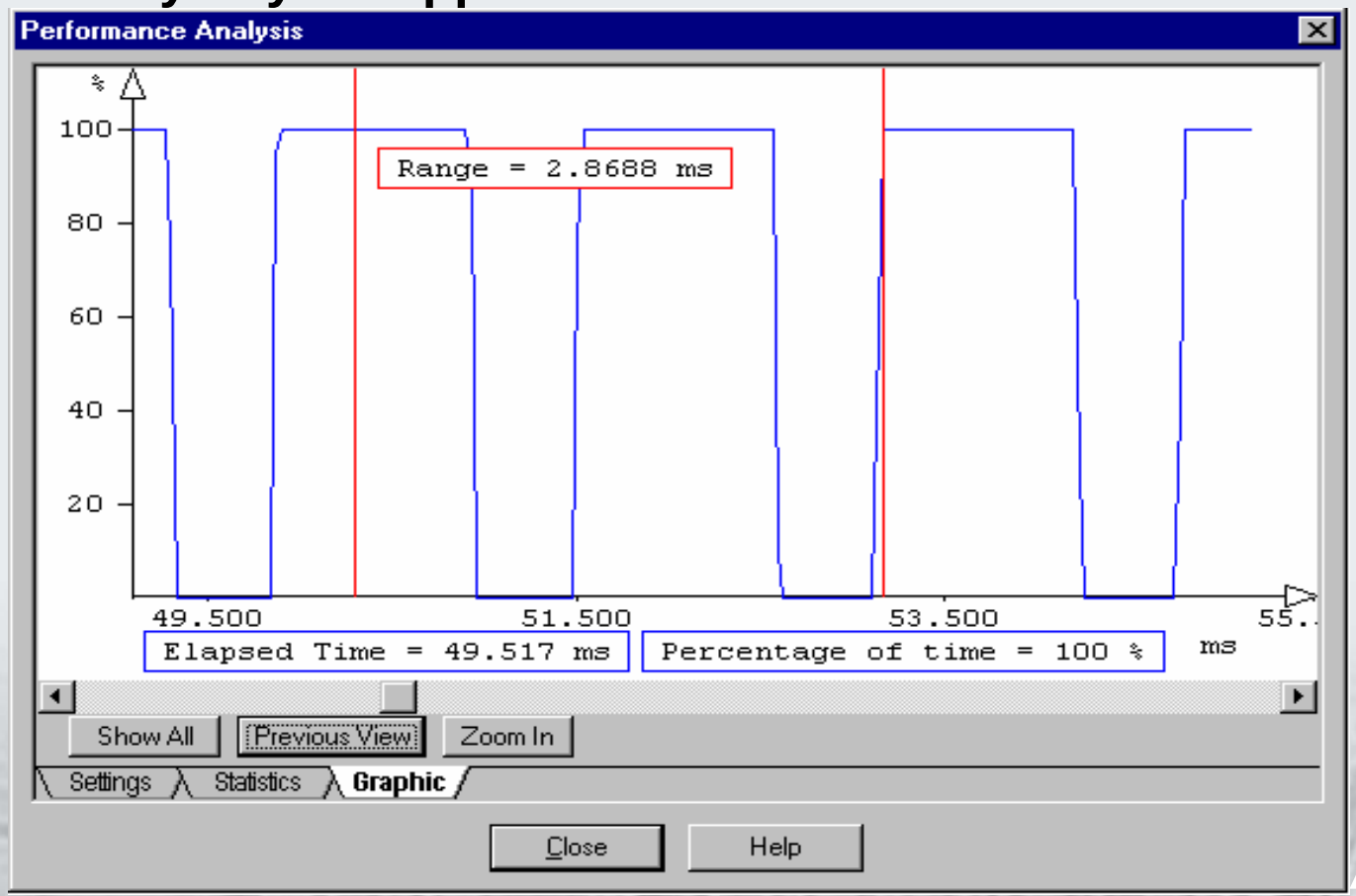
ST7 Visual Debug
Emulator-Specific functions

- **Performance Analysis (on EMU3):**



ST7 Visual Debug
Emulator-Specific functions

- **Performance Analysis:** **allows you to measure the time spent in a given portion of your code in order to evaluate the efficiency of your application**



ST7 Visual Debug
Emulator-Specific functions

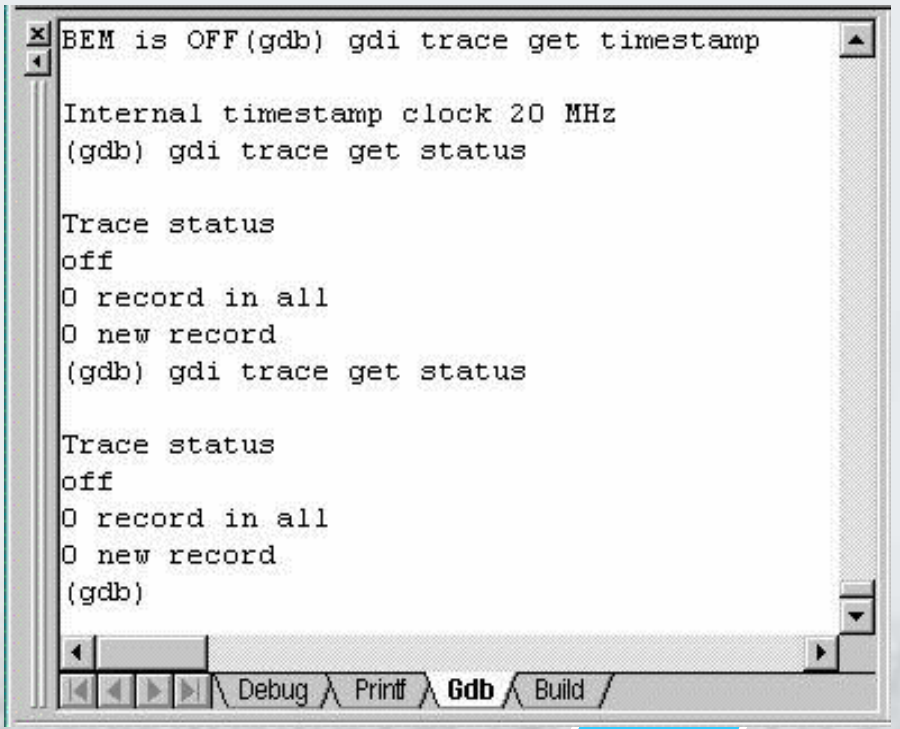
- **Trace:**
  - Choice of columns displayed free
  - Timestamp only present in EMU3
  - 1k Trace in EMU2B, 256k Trace in EMU3

Trace

	Record	Address	Memo...	Data	Event	Hexad...	Disassembly	Symbolic D...	Timestamp
	16	0x000005	In MCU	0xff	Data write				513 950 ns
→		main.asm		215			ld PBOR,A ...		
	17	0x00e20b	EMU	0xb7	Operation code fetch	0x00b706	LD 0x06,A	LD PBOR,A	514 100 ns
	18	0x00e20c	EMU	0x06	Operand fetch				514 200 ns
	19	0x000006	In MCU	0x00	Discarded				514 350 ns
	20	0x000006	In MCU	0xff	Data write				514 450 ns
→		main.asm		217			ld A,#1		
	21	0x00e20d	EMU	0xa6	Operation code fetch	0x00a601	LD A,#0x01	LD A,#0x01	514 600 ns
	22	0x00e20e	EMU	0x01	Operand fetch				514 700 ns

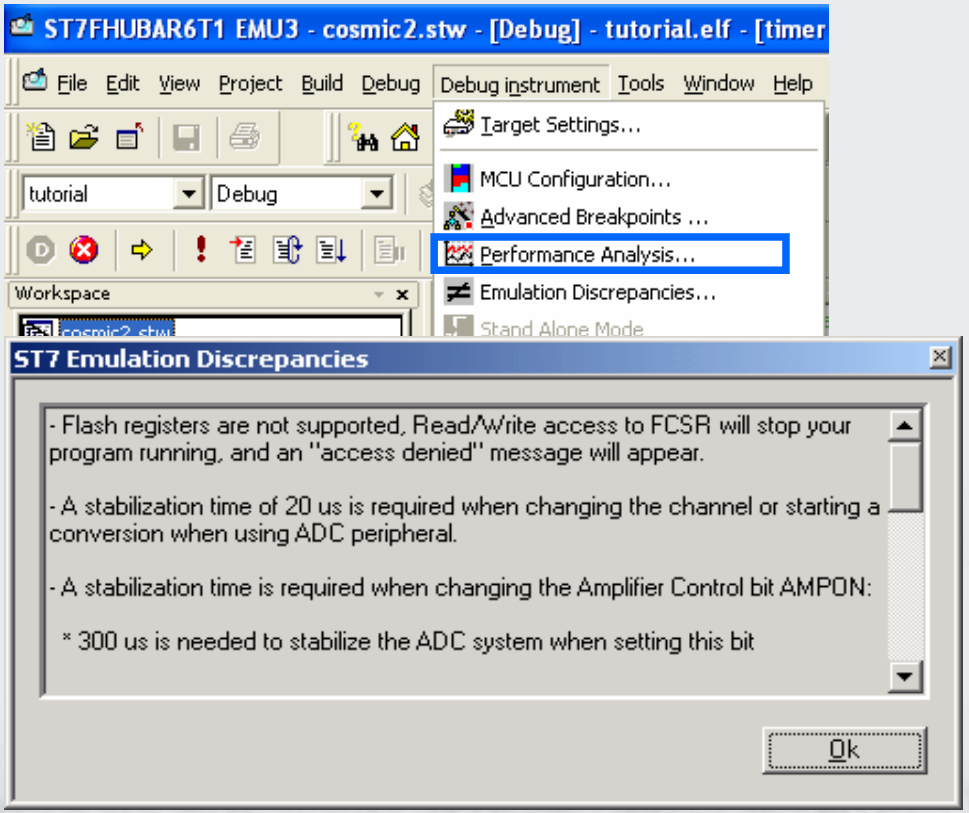
ST7 Visual Debug
Debugging Features

- GDB Console and Information window:
  - **This window displays GDB output**
  - **Permits entry of commands**
  - **Permits to load binary file without debug information**



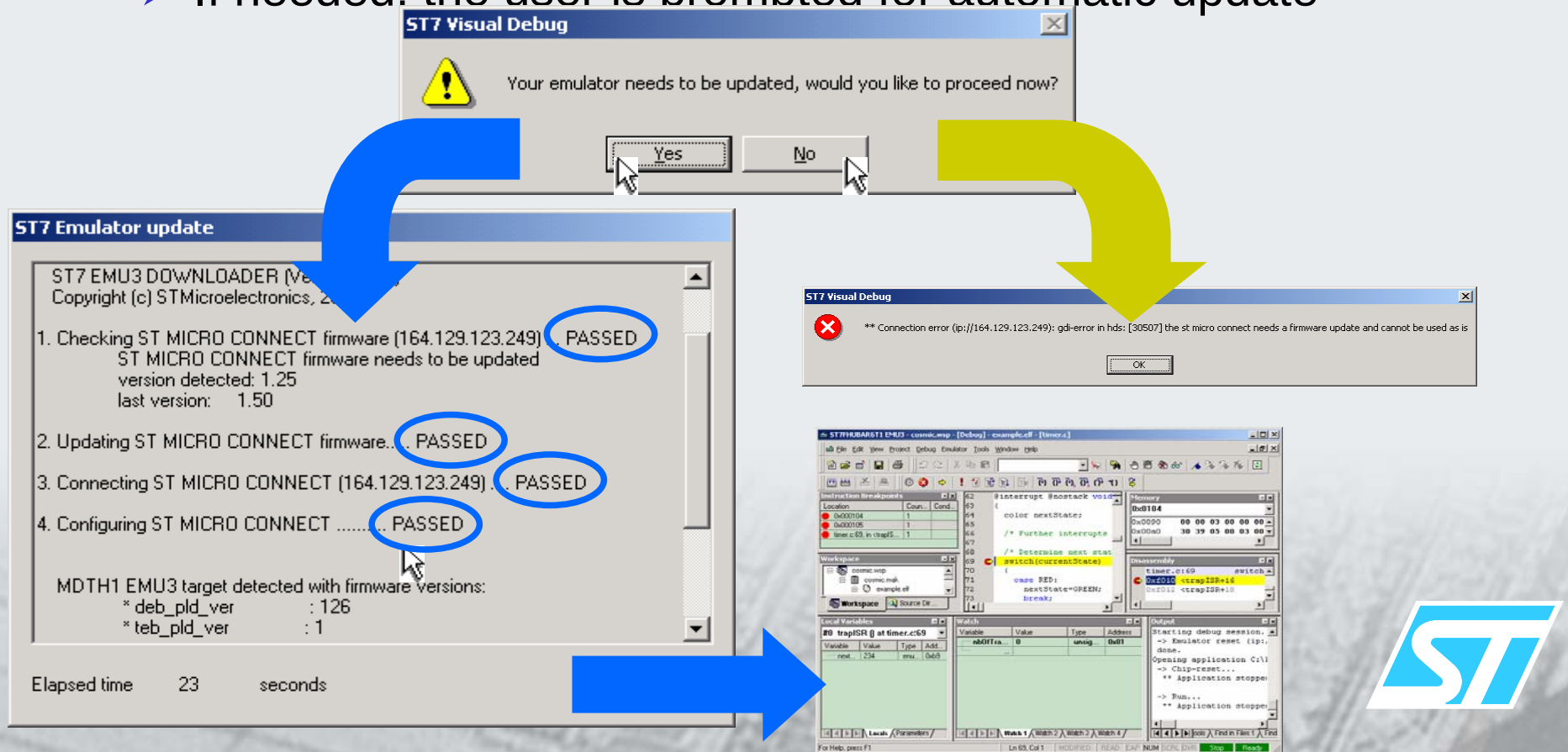
Discrepancies

- Supported for:
  - EMU3, DVP3, ICD
- EMU3, DVP3
  - Describes the differences between emulated chip and your actual ST7
- ICD
  - describes limitations that are chip-specific

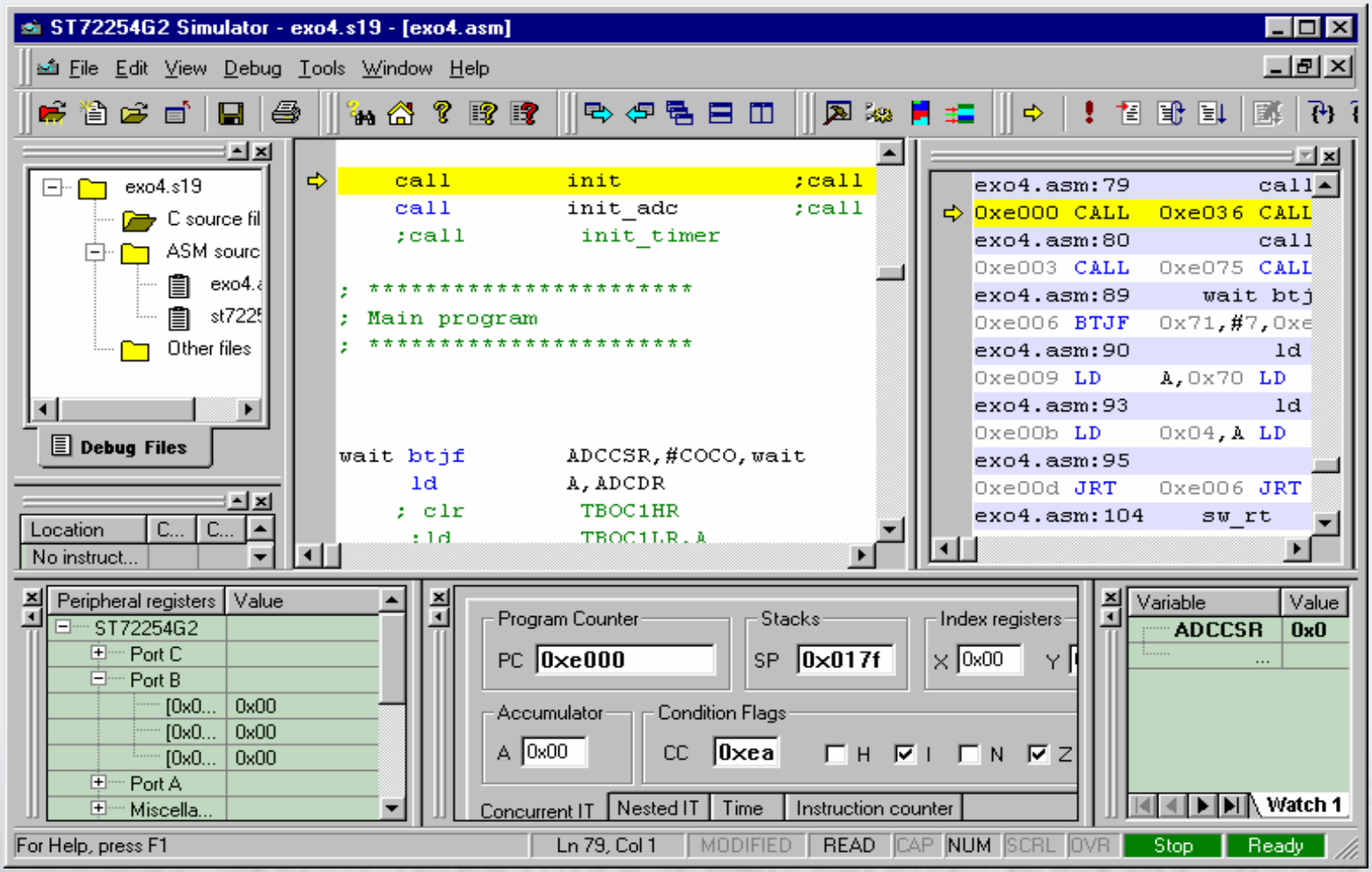


Emulator update (EMU3)

- Emulators contain firmware that may be updated
- When the EMU3 is initialized
  - STVD7 checks the firmware versions
  - If needed, the user is prompted for automatic update



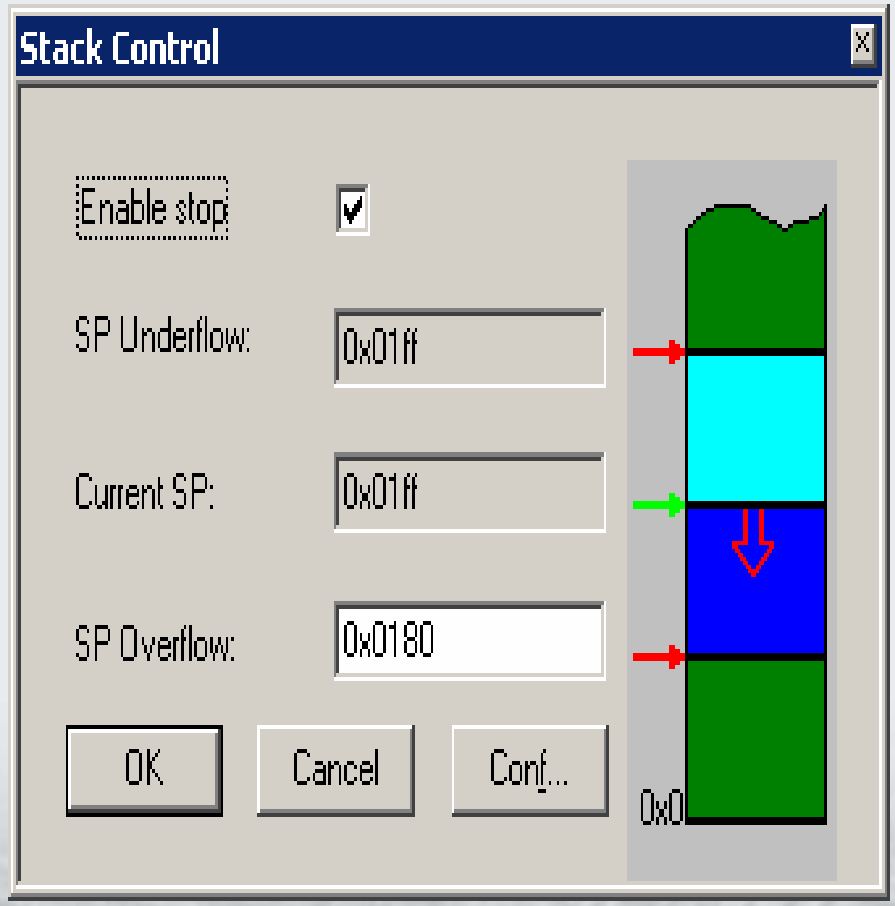
STVD7


ST72254G2 Simulator - exo4.s19 - [exo4.asm]
File Edit View Debug Tools Window Help
exo4.s19
C source fil
ASM sourc
exo4.a
st7225
Other files
Debug Files
call init ;call
call init_adc ;call
;call init_timer
; ***********************
; Main program
; ***********************
wait btjf ADCCSR,#COCO,wait
ld A,ADCDR
; clr TBOC1HR
exo4.asm:79 call
0xe000 CALL 0xe036 CALL
exo4.asm:80 call
0xe003 CALL 0xe075 CALL
exo4.asm:89 wait btj
0xe006 BTJF 0x71,#7,0xe
exo4.asm:90 ld
0xe009 LD A,0x70 LD
exo4.asm:93 ld
0xe00b LD 0x04,A LD
exo4.asm:95
0xe00d JRT 0xe006 JRT
exo4.asm:104 sw_rt
Location C... C...
No instruct...
Peripheral registers Value
ST72254G2
Port C
Port B
[0x0... 0x00
[0x0... 0x00
[0x0... 0x00
Port A
Miscella...
Program Counter
PC 0xe000
Stacks
SP 0x017f
Index registers
X 0x00 Y
Accumulator
A 0x00
Condition Flags
CC 0xea H I N Z
Concurrent IT Nested IT Time Instruction counter
Variable Value
ADCCSR 0x0
Watch 1
For Help, press F1
Ln 79, Col 1
MODIFIED READ CAP NUM SCRL OVR
Stop Ready

SIMULATOR OVERVIEW

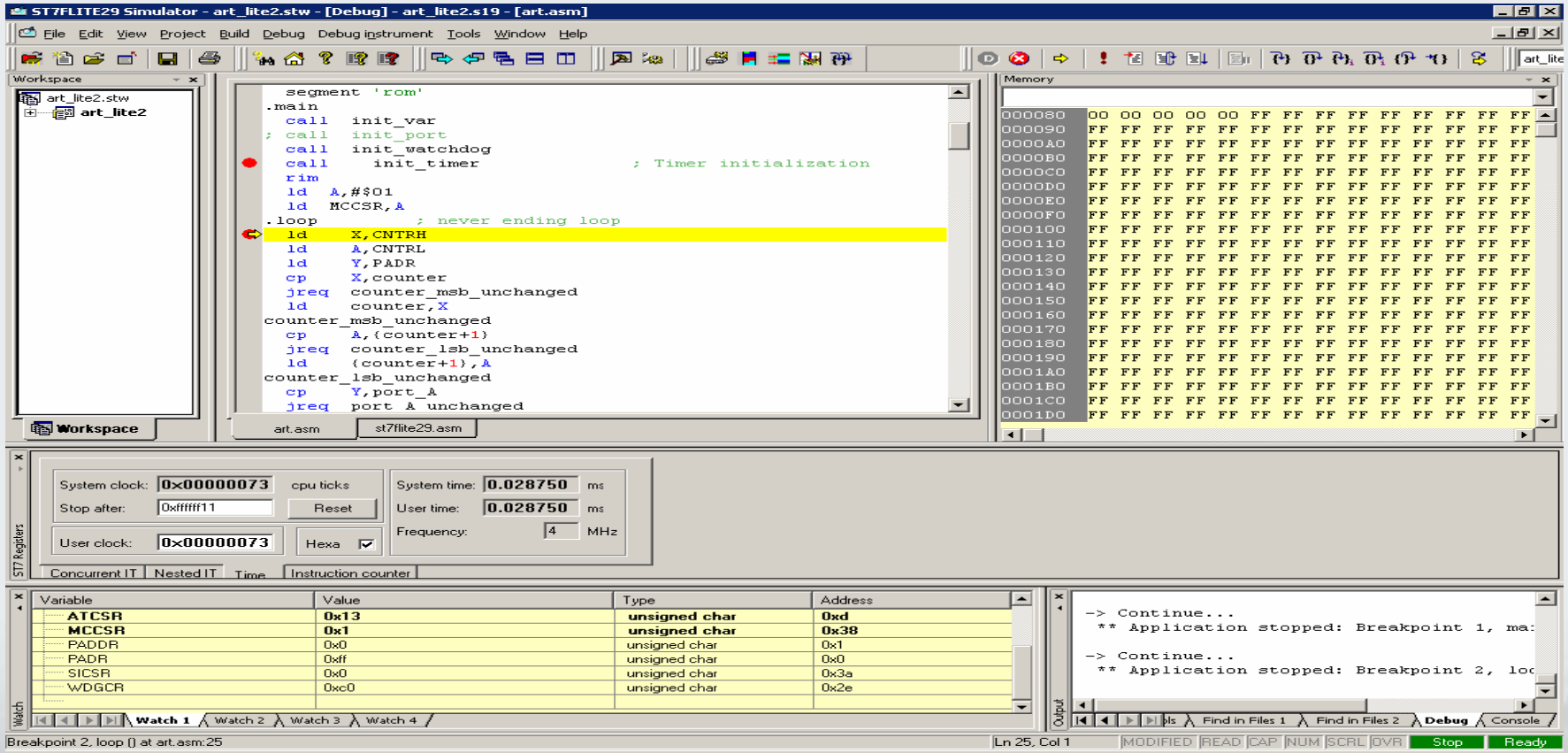
- Scope
  - **Instruction level simulation**
  - **Peripheral simulation: Port, Watchdog, Timers, ADC, SPI, I2C, SCI, E2PROM, WDG, MCC, RTC**
  - **Pin level simulation (plotter)**
  - **ST7 interrupt processing simulation**
  - **To check overall code organization**
  - **To verify simple peripheral configuration**
  - **To check connection of interrupts to interrupt handlers**
- Time management
  - **Display time**
  - **break on time**
- Breakpoints management
  - **Read & write access**
  - **Stack Overflow/Underflow**
  - **Invalid memory access detection**
- No trace and no logic analyser features

Stack control



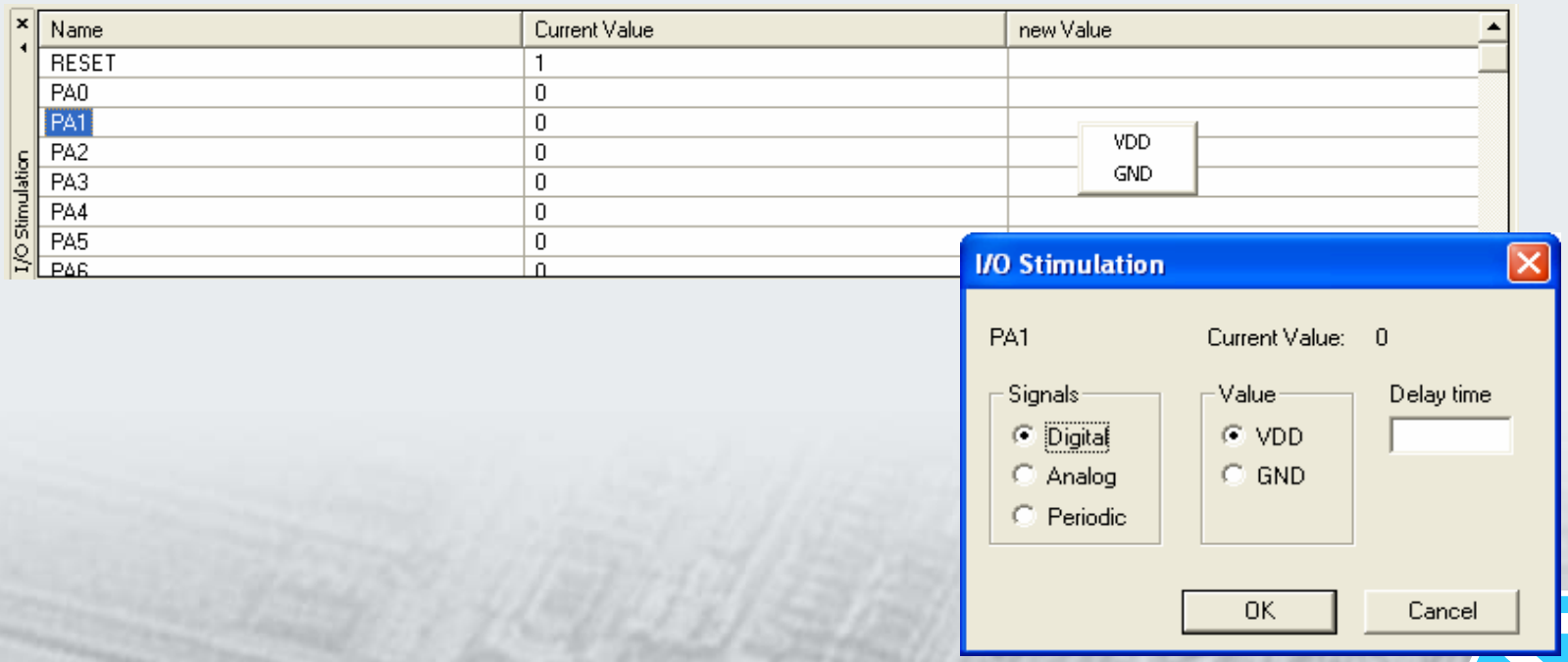
- Stop on stack overflow/underflow
- Configurable stack overflow

Read/write on the fly



Input pin stimulator

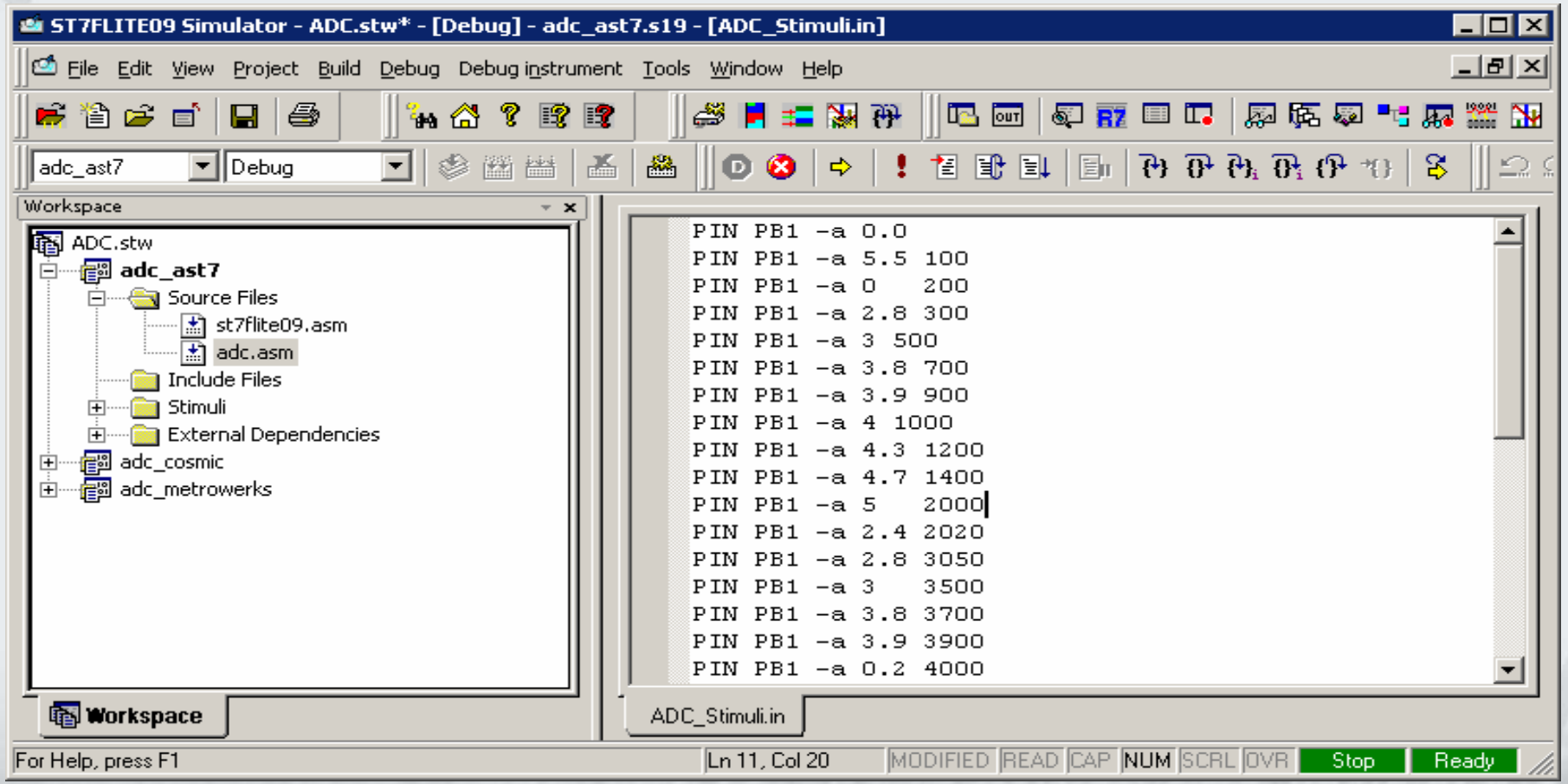
- Stimulate each pin with digital or analog value



Input pin stimulator features

- Trigger an analog or digital signal with or without a time delay
- Trigger a periodic digital signal with or without a time delay
  - Double click in value field
- Trigger a digital signal on the fly
  - Right click in value field

Stimuli file



Stimuli file syntax

- pin <pin name> -i <digital value> [<time>]

Apply a digital value on an input pin

- pin <pin name> -a <analog value> [<time>]

Apply an analog value on an input pin

- pin <pin name> -c <start digital value> [<start time>] <half period>

Apply a square periodic signal on an input pin

Digital value is 0 or 1

Analog value is in the range [0-6.55]

Time is expressed in CPU cycles

Default time is current time

gdi pin command

- gdi pin -output_file <no/yes>

Default is no. If yes, pin input and output stimuli are logged in a file named port.out.

- gdi help pin

Get “gdi pin” syntax.

Stimuli pin commands (pin -i/-a/-c) used outside a stimuli file must be preceded by the gdi keyword.

The plotter

- Can plot the evolutions of global variables, pins, registers -> software oscilloscope!