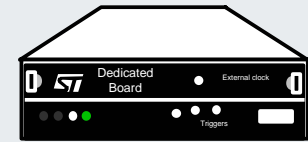
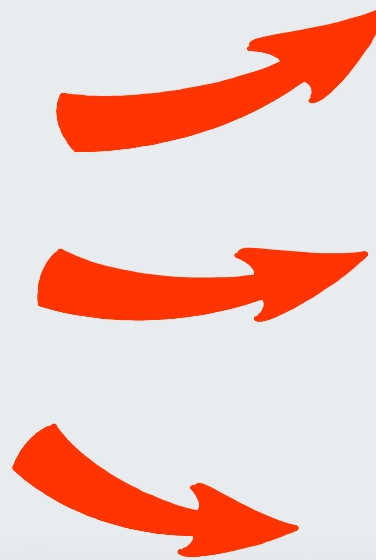
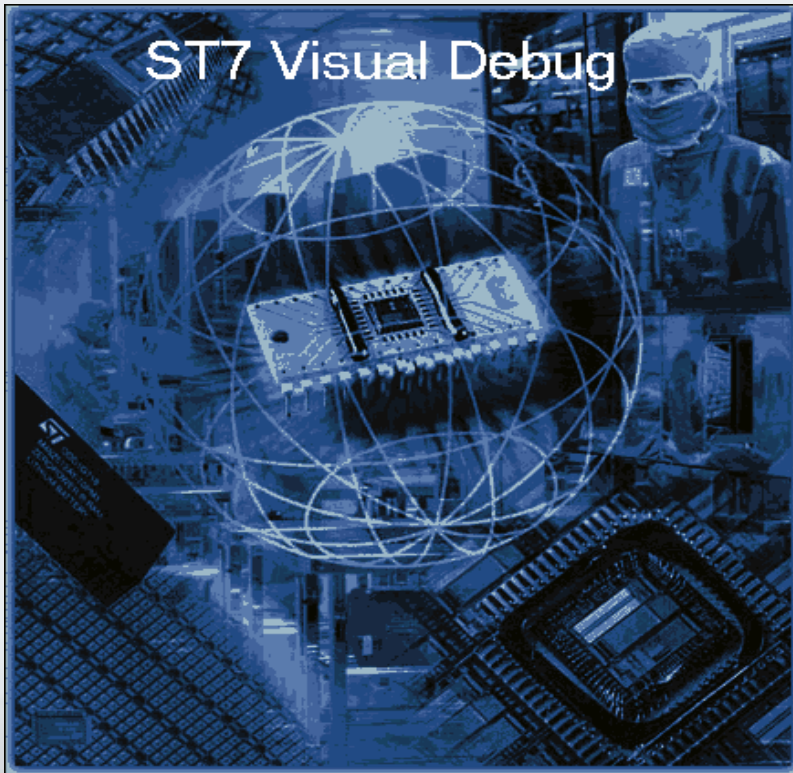


ST7 MICROCONTROLLER TRAINING

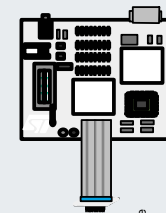
1. INTRODUCTION
2. CORE
3. ADDRESSING MODES
4. ASSEMBLY TOOLCHAIN
5. **STVD7 DEBUGGER**
6. HARDWARE TOOLS
7. PERIPHERALS
8. ST-REALIZER II
9. C TOOLCHAINS



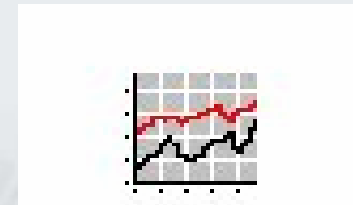
STVD7: ST WINDOWS DEBUGGER FOR ST7 MCUs



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

Project Settings [?] [X]

Workspace filename:

Software Toolchain:

Executable

Filename: ...

Build

Maker:

Maker filename: ...

Makefile or batch file: ...

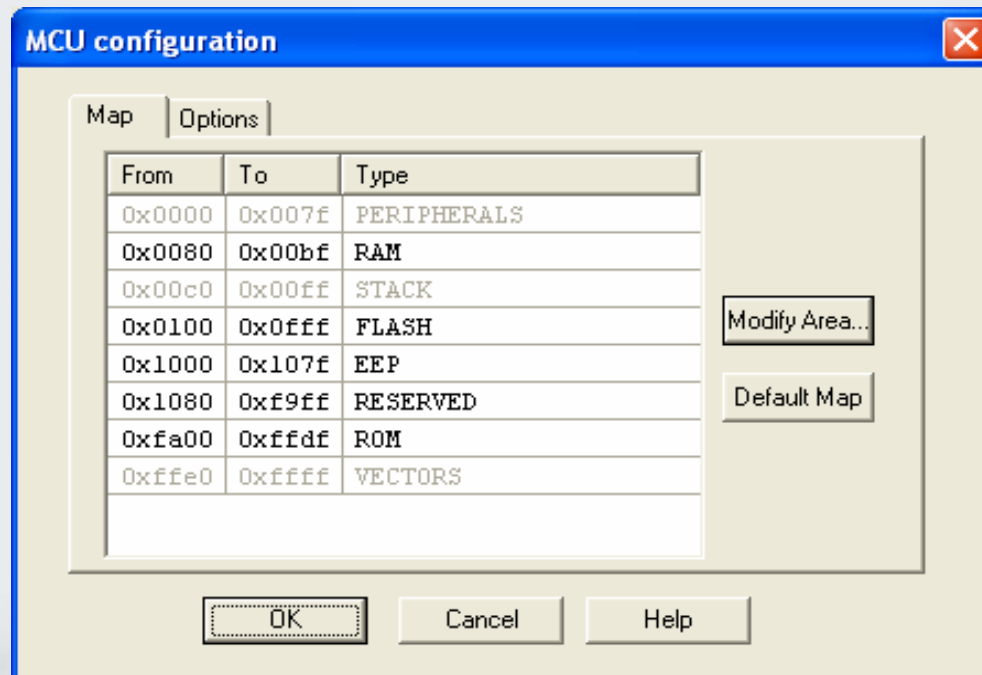
"Build" command line:

"Rebuild All" command line:

Start "Build" / "Rebuild All" in: ...

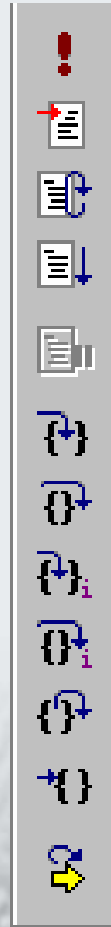


ST7 Visual Debug Micro Configuration



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

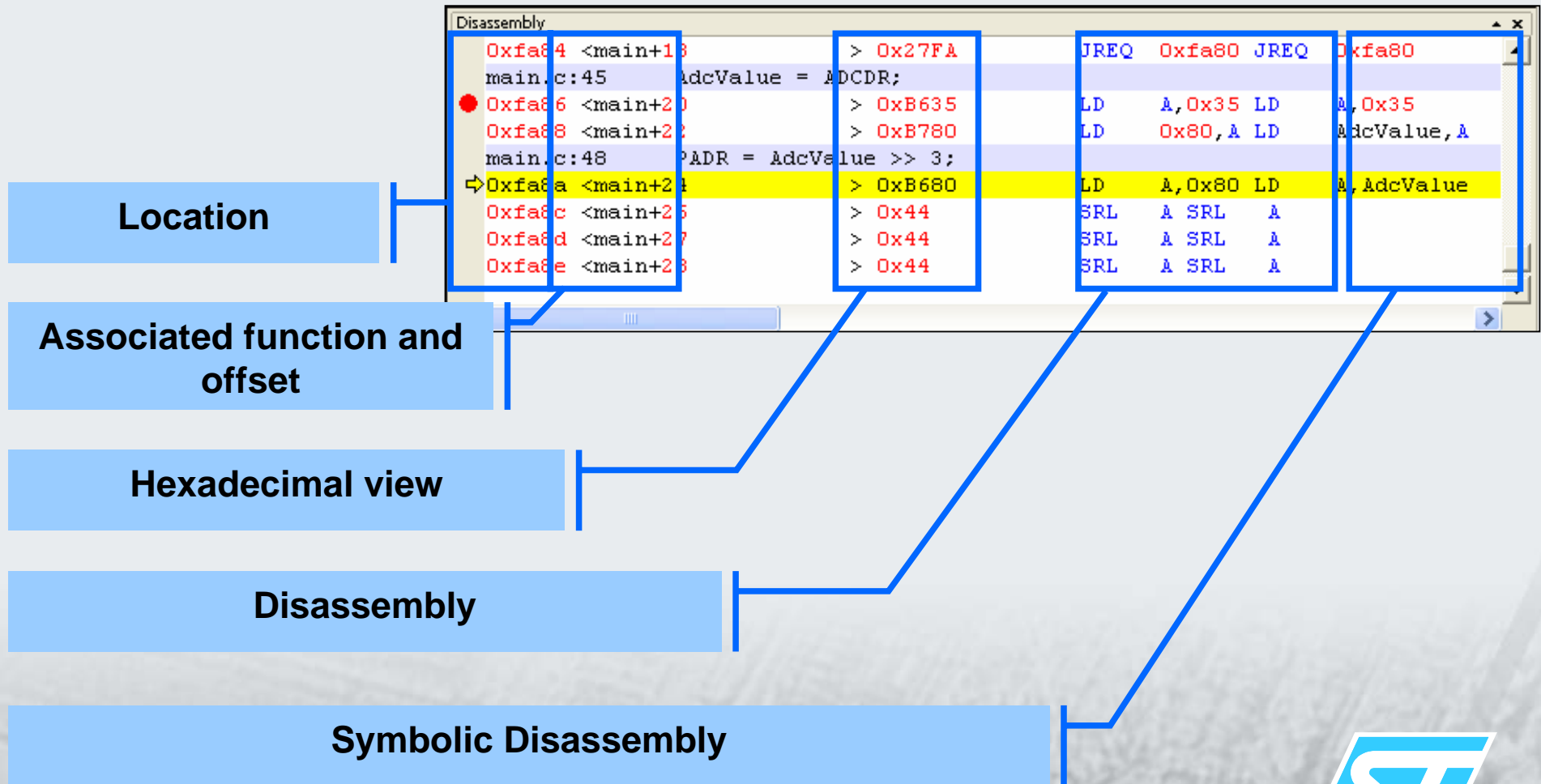


ST7 Visual Debug Debugging Features

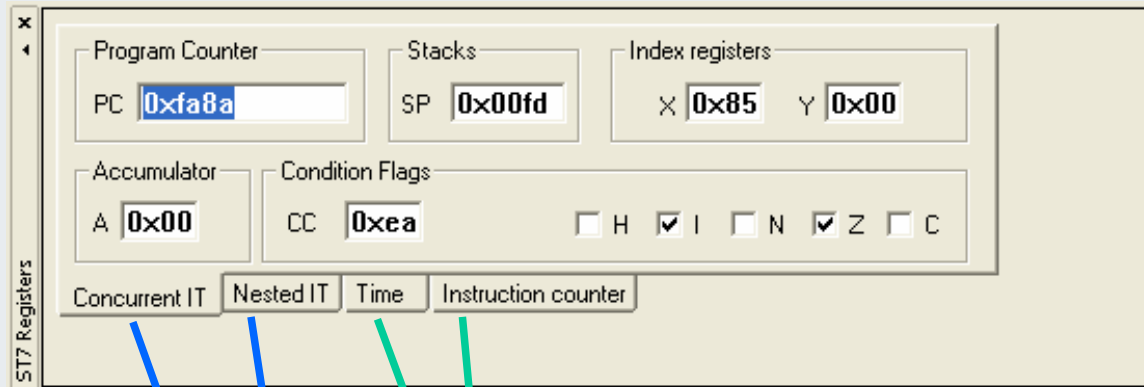
- Common Debug Features
 - **Disassembly**
 - **Memory Dump**
 - **Instruction Breakpoints**
 - **Data Breakpoints**
 - **Watch**
 - **Call Stack**
 - **Local Variables**
 - **ST7 Registers Window**
 - **Symbol browser**
 - **Peripheral registers window**



Disassembly window



ST7 registers



Simulator-specific pseudo registers

Two ways to interpret CC registers



Memory

Starting address of the dumped memory block

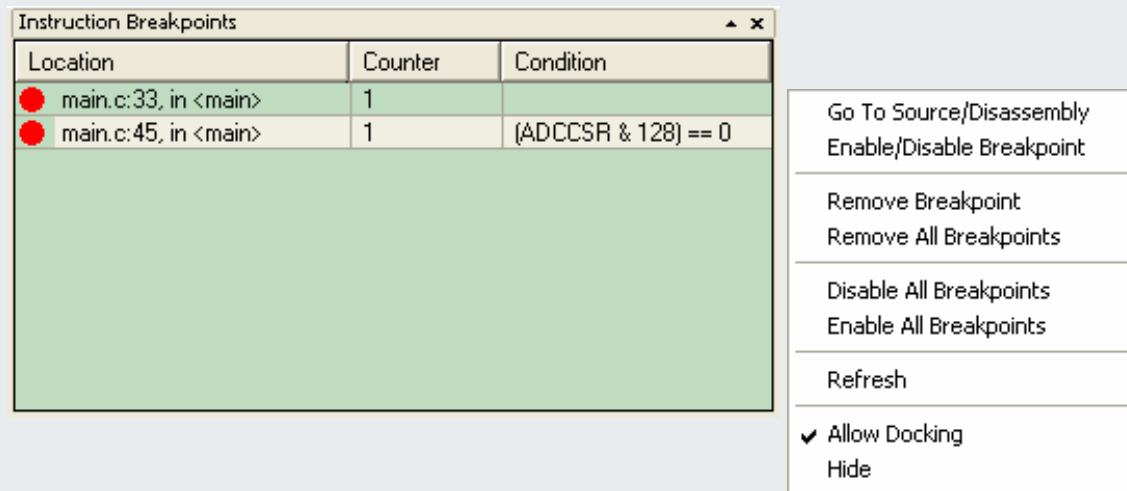
| Memory | | |
|--------|--|-----------------|
| 0x80 | | |
| 000080 | 00 00 00 00 00 00 03 00 00 00 01 00 00 A0 00 00 | |
| 000090 | 00 00 03 00 00 00 00 FF FF FF FF 00 00 00 00 00 | |
| 0000A0 | 30 39 05 00 00 00 10 0F 00 01 41 C6 4E 6D 00 00 00 | 09.....AENra... |
| 0000B0 | 81 00 00 10 0F 00 03 00 00 FF FF FF FF FF FF FF | |
| 0000C0 | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF | |
| 0000D0 | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF | |
| 0000E0 | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF | |
| 0000F0 | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF | |
| 000100 | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF | |
| 000110 | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF | |
| 000120 | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF | |

Binary Dump

ASCII dump



Instruction breakpoints



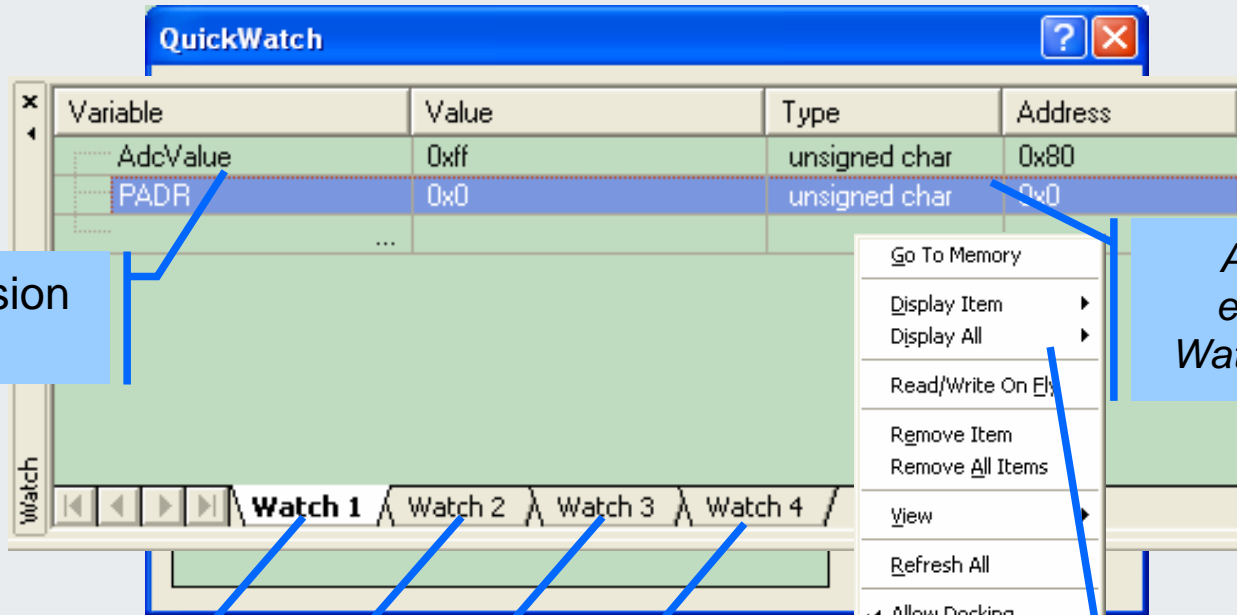
The screenshot shows a window titled "Instruction Breakpoints" with a table containing two rows of breakpoints. The first row is highlighted in green. A context menu is open over the table, listing various actions such as "Go To Source/Disassembly", "Remove Breakpoint", and "Allow Docking".

| Location | Counter | Condition |
|------------------------|---------|---------------------|
| ● main.c:33, in <main> | 1 | |
| ● main.c:45, in <main> | 1 | (ADCCSR & 128) == 0 |

- Go To Source/Disassembly
- Enable/Disable Breakpoint
- Remove Breakpoint
- Remove All Breakpoints
- Disable All Breakpoints
- Enable All Breakpoints
- Refresh
- Allow Docking
- Hide

- 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 |
| 0xfb2f | <main+2 | > | 0xADC6 |
| 0xfb31 | <main+4 | > | 0x81 |
| 0xfb32 | <_rand | > | 0xAE9E |
| 0xfb34 | <_rand+2 | > | 0xCDFBC0 |
| | | > | 0xAEFD |
| | | > | 0xBFA7 |
| | | > | 0xAE4E |
| | | > | 0xCDFC71 |
| | | > | 0xAEFD |
| | | > | 0xBFA7 |
| | | > | 0xAE52 |
| | | > | 0xCDFC21 |
| | | > | 0xAE9E |
| | | > | 0xCDFBD0 |

timer.c *

```
146
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

| Variable | Value | Type | Address |
|----------|--------|---------------|---------|
| count | 196608 | unsigned long | 0xb5 |

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

The screenshot shows the 'Peripheral registers' window in the ST7 Visual Debug tool. The window is titled 'Peripheral registers' and has a 'Value' column. The registers are organized into a tree view under 'ST7FLITE09'. The 'Lite Timer' section is expanded, showing registers [0x000b] LTCSR and [0x000c] LTICR. The 'EEPROM' section is also expanded, showing registers [0x0030] EECRSR, E2PGM, and E2LAT. The 'Serial Peripheral Interface (SPI)' section is expanded, showing registers 0 - Read mode and 1 - Write mode. A context menu is open over the 'Intrusive read' error message, which is circled in blue. The menu items are: Refresh, Display Item, Display All, Display Address, Forced Read, Allow Docking, and Hide. A blue arrow points from the 'Intrusive read' error message to the 'Forced Read' menu item.

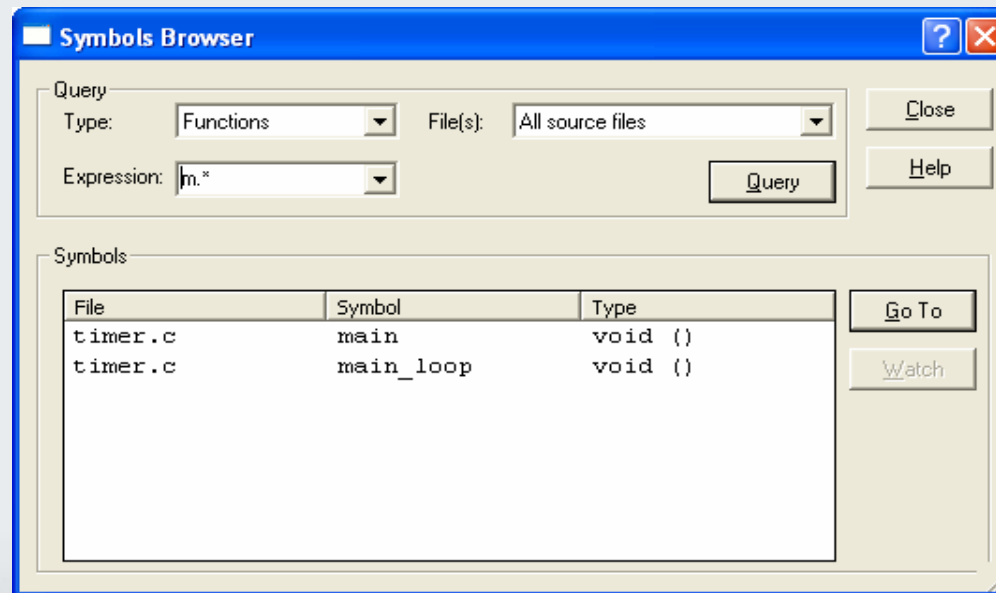
| Register Name | Value |
|---|--------------------------|
| [0x0000] PADR - Data Register | 0x00 |
| [0x0001] PADDR - Data Direction Register | 0x00 |
| [0x0002] PAOR - Option Register | 0x40 |
| [0x000b] LTCSR - Lite Timer Control/Sta... | Intrusive read |
| [0x000c] LTICR - Lite Timer Input Captur... | Intrusive read |
| [0x0030] EECRSR - Data EEPROM Contr... | 0x00 |
| E2PGM - Programming Control and s... | 0 - Finished/not started |
| E2LAT - Latch Access Transfer | 0 - Read mode |
| 0 - Read mode | 0 - Read mode |
| 1 - Write mode | 1 - Write mode |

Bit-level description and control



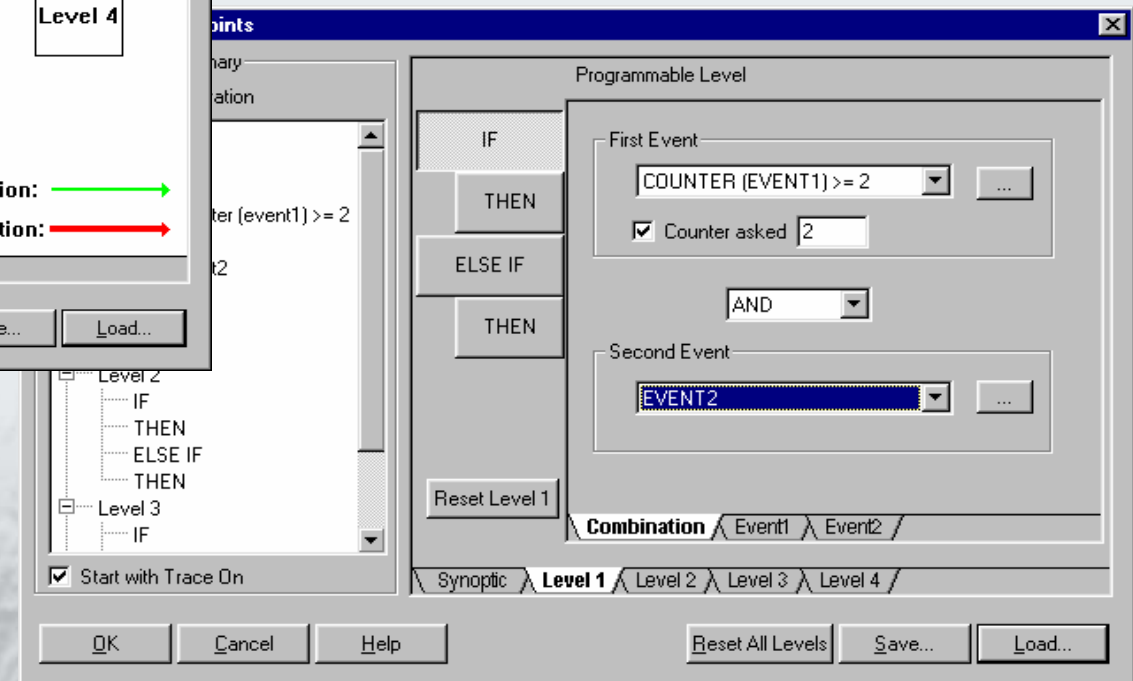
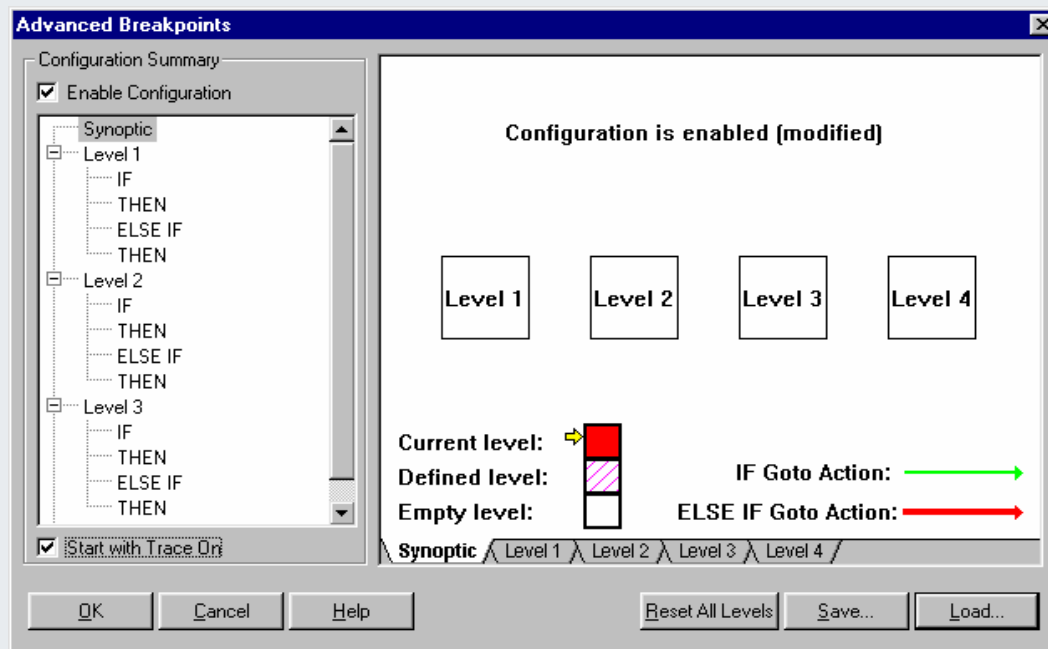
Symbol browser

- Based on debug information



ST7 Visual Debug Emulator-Specific functions

- BEM (Bus Event Machine)



ST7 Visual Debug Emulator-Specific functions

- Performance Analysis (on EMU3):

The screenshot displays two overlapping windows from the ST7 Visual Debug software. The background window is the 'Performance Analysis' dialog box, which is currently on the 'Settings' tab. It features input fields for 'Number of passes' (set to 100), 'Starting address' (Bookmark, File: sample1.c, Address: 0x2AD), and 'Ending address' (Bookmark, File: sample1.c, Address: 0x2E8). A checkbox for 'Disable breakpoints' is checked, and an 'Apply' button is visible. The foreground window is also titled 'Performance Analysis' but is on the 'Statistics' tab. It displays the following data:

Initial settings

- Starting Address : 0x2AD
- Ending Address : 0x2E8
- Passes specified : 100

Statistics for one pass through portion of code

| | | |
|--------------------|--------|----|
| Average time | 1.0850 | ms |
| Minimum time | 1.0850 | ms |
| Maximum time | 1.0850 | ms |
| Standard deviation | 0.0000 | ms |

Statistics for elapsed test time

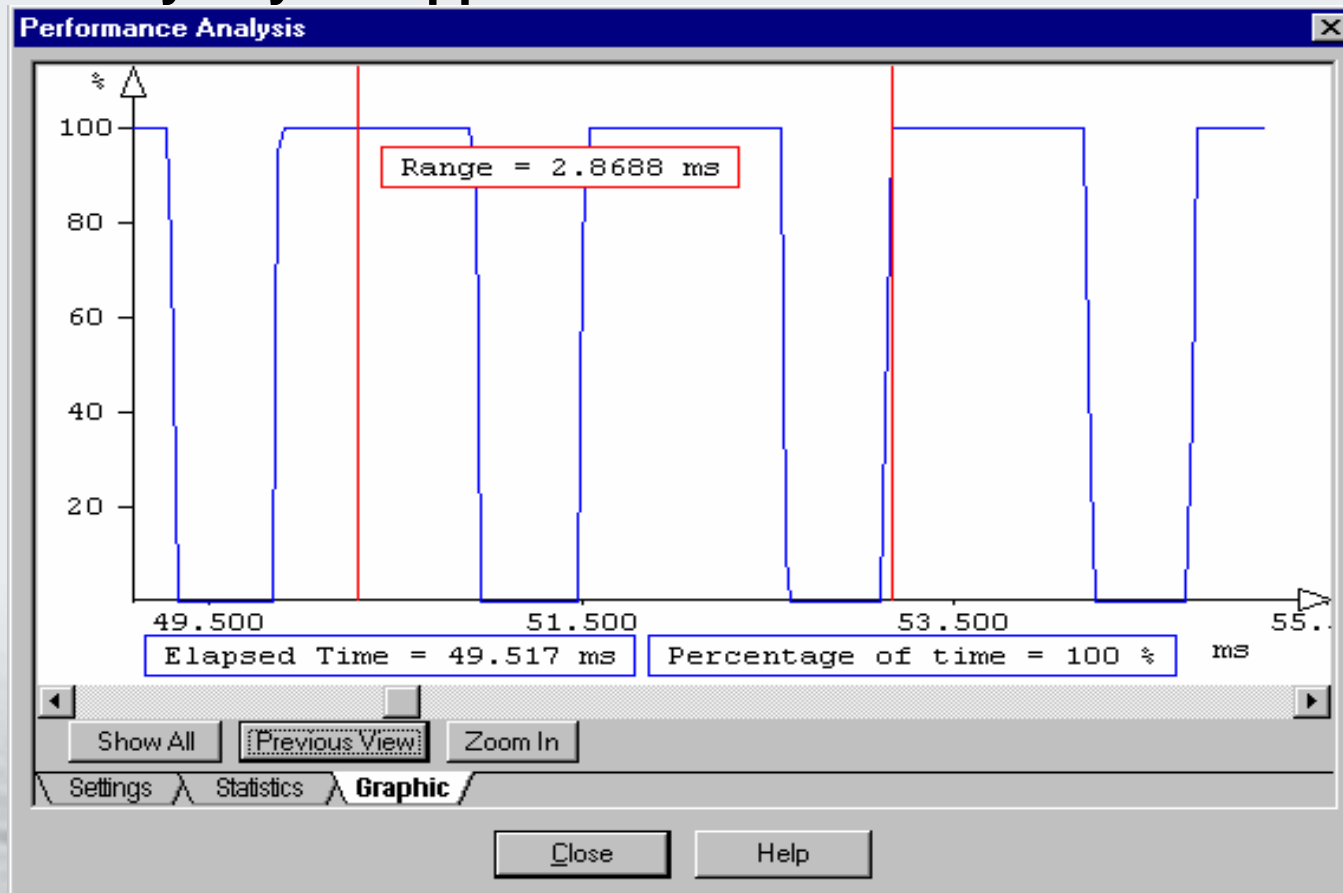
| | |
|---------------------------|-------------|
| Number of executed passes | 100 |
| Total time | 195.1675 ms |
| % time in code | 55.5922 % |

At the bottom of the foreground window, there are 'Close' and 'Help' buttons. The background window has 'Settings', 'Statistics', and 'Graphic' tabs, with 'Settings' currently selected.

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

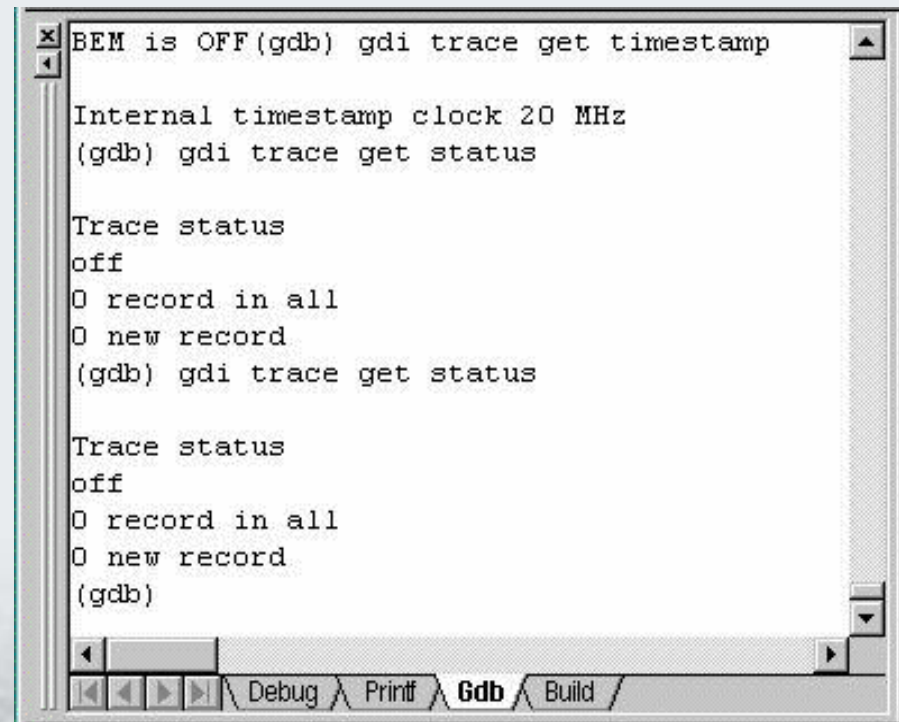
- **Trace:**
 - Choice of columns displayed free
 - Timestamp only present in EMU3
 - 1k Trace in EMU2B, 256k Trace in EMU3

| 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



The screenshot shows a GDB console window with the following text:

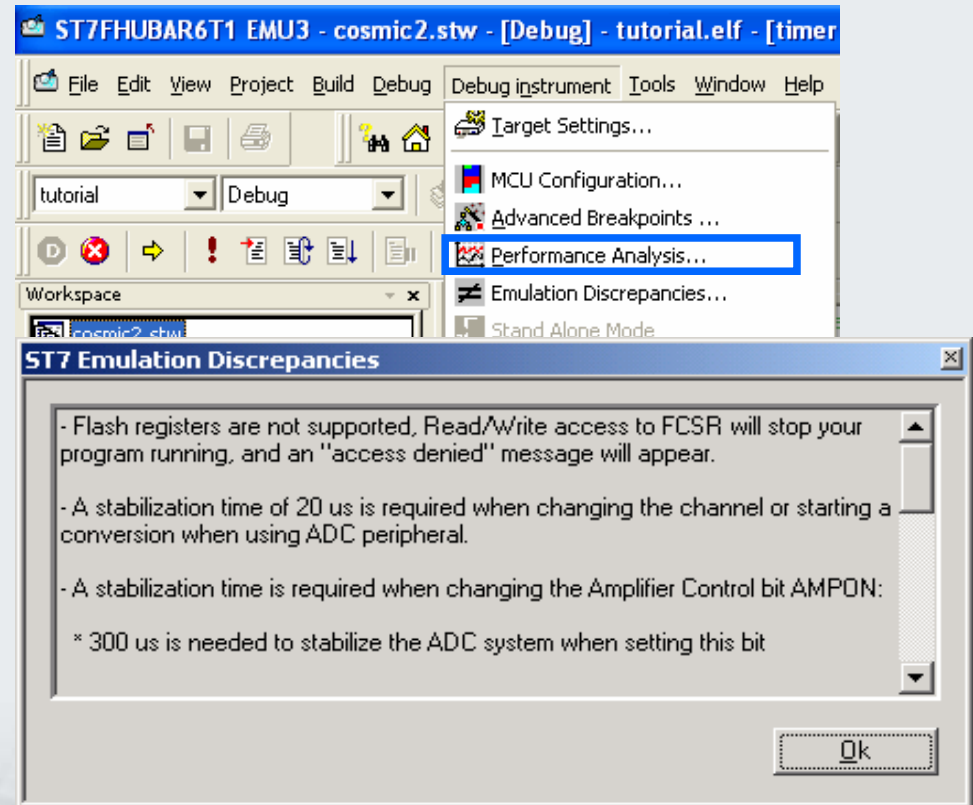
```
BEM is OFF(gdb) gdi trace get timestamp
Internal timestamp clock 20 MHz
(gdb) gdi trace get status
Trace status
off
0 record in all
0 new record
(gdb) gdi trace get status
Trace status
off
0 record in all
0 new record
(gdb)
```

The window has a title bar with a close button (X) and a scroll bar on the right. At the bottom, there is a toolbar with buttons for 'Debug', 'Print', 'Gdb', and 'Build'.



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

The image illustrates the ST7 emulator update process through several screenshots:

- Warning Dialog:** A dialog box titled "ST7 Visual Debug" with a yellow warning icon. The text reads: "Your emulator needs to be updated, would you like to proceed now?". It has "Yes" and "No" buttons.
- Update Progress Window:** A window titled "ST7 Emulator update" showing the progress of the update. The steps are:
 1. Checking ST MICRO CONNECT firmware (164.129.123.249) **PASSED**
ST MICRO CONNECT firmware needs to be updated
version detected: 1.25
last version: 1.50
 2. Updating ST MICRO CONNECT firmware... **PASSED**
 3. Connecting ST MICRO CONNECT (164.129.123.249) **PASSED**
 4. Configuring ST MICRO CONNECT **PASSED**Below the steps, it says: "MDTH1 EMU3 target detected with firmware versions:
* deb_pld_ver : 126
* teb_pld_ver : 1"
At the bottom, it shows "Elapsed time 23 seconds".
- Error Dialog:** A dialog box titled "ST7 Visual Debug" with a red error icon. The text reads: "** Connection error (ip://164.129.123.249): gdi-error in hds: [30507] the st micro connect needs a firmware update and cannot be used as is". It has an "OK" button.
- Main Interface:** The "ST7 Visual Debug" main interface showing various windows like "Instruction Breakpoints", "Workspace", "Local Variables", "Watch", "Disassembly", and "Output".

Blue arrows indicate the flow from the warning dialog to the update progress window, and from the update progress window to the main interface. A yellow arrow points from the warning dialog to the error dialog.



STVD7

The screenshot displays the ST72254G2 Simulator interface for the file 'exo4.s19'. The main window shows assembly code with the following instructions:

```
call    init    ;call
call    init_adc ;call
;call    init_timer
; *****
; Main program
; *****
wait btjf ADCCSR, #COCO, wait
ld       A, ADCDR
; clr    TBOC1HR
; ld     TROC1LR, A
```

The disassembly window on the right shows the corresponding machine code instructions:

```
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
```

The bottom section of the simulator provides system status and register values:

- Peripheral registers:** ST72254G2, Port C, Port B, 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)
- Variable Watch:** ADCCSR 0x0

The status bar at the bottom indicates the current instruction is at Ln 79, Col 1, with a status of 'Stop Ready'.

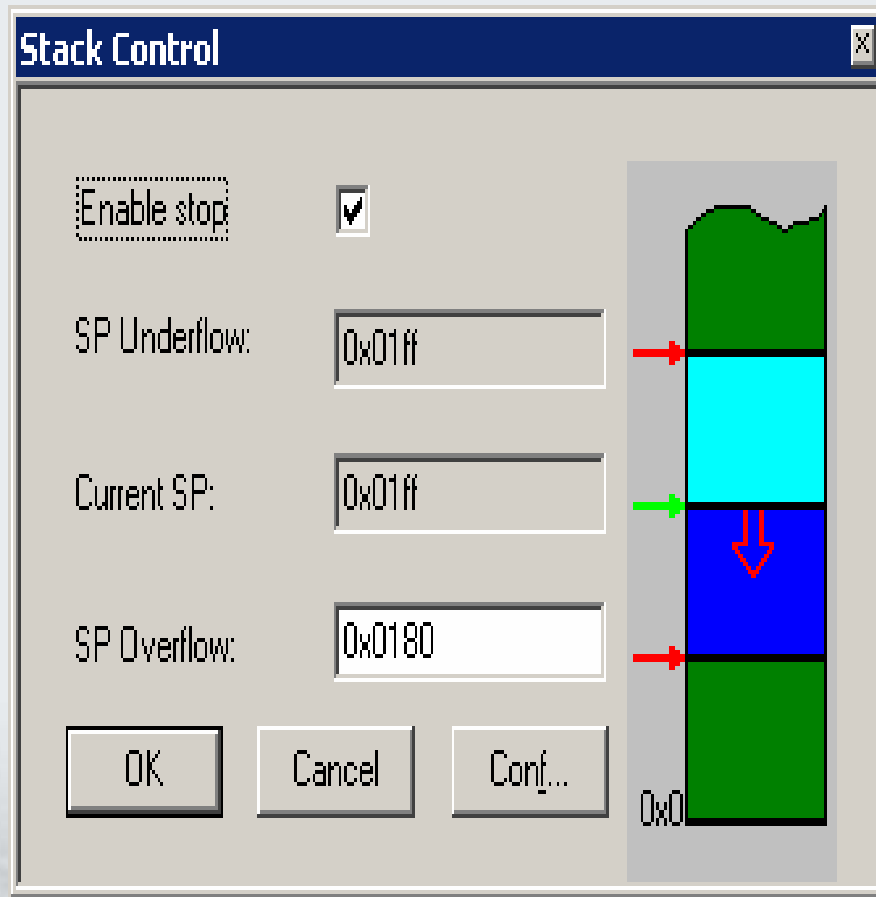


SIMULATOR OVERVIEW

- 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

The screenshot displays the ST7FLITE29 Simulator interface. The main window shows the assembly code for 'art.asm' with a breakpoint set at the start of the '.loop' section. The memory window shows a memory dump of 00 00 00 00 FF FF FF FF. The system clock and user clock are both set to 0x00000073. The registers window shows the MCCR register at address 0x38 with value 0x1. The output console shows the application stopped at breakpoint 2.

```
segment 'rom'
.main
call init_var
; call init_port
call init_watchdog
call init_timer ; Timer initialization
rim
ld A, #01
ld MCCR, A
.loop ; never ending loop
ld X, CNTRH
ld A, CNTRL
ld Y, PADDR
cp X, counter
jreq counter_msb_unchanged
ld counter, X
counter_msb_unchanged
cp A, (counter+1)
jreq counter_lsb_unchanged
ld (counter+1), A
counter_lsb_unchanged
cp Y, port_A
jreq port_A_unchanged
```

| Variable | Value | Type | Address |
|----------|-------|---------------|---------|
| ATCSR | 0x13 | unsigned char | 0xd |
| MCCR | 0x1 | unsigned char | 0x38 |
| PADDR | 0x0 | unsigned char | 0x1 |
| PADR | 0xf | unsigned char | 0x0 |
| SICSR | 0x0 | unsigned char | 0x3a |
| WDGCR | 0xc0 | unsigned char | 0x2e |

Output console:
-> Continue...
** Application stopped: Breakpoint 1, ma...
-> Continue...
** Application stopped: Breakpoint 2, loc...



Input pin stimulator

- Stimulate each pin with digital or analog value

| Name | Current Value | new Value |
|-------|---------------|-----------|
| RESET | 1 | |
| PA0 | 0 | |
| PA1 | 0 | |
| PA2 | 0 | VDD |
| PA3 | 0 | GND |
| PA4 | 0 | |
| PA5 | 0 | |
| PA6 | 0 | |

I/O Stimulation

PA1 Current Value: 0

Signals

Digital

Analog

Periodic

Value

VDD

GND

Delay time

OK Cancel



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

The screenshot shows the ST7FLITE09 Simulator interface. The title bar reads "ST7FLITE09 Simulator - ADC.stw* - [Debug] - adc_ast7.s19 - [ADC_Stimuli.in]". The menu bar includes File, Edit, View, Project, Build, Debug, Debug instrument, Tools, Window, and Help. The toolbar contains various icons for file operations and simulation control. The workspace on the left shows a project tree with folders for Source Files, Include Files, Stimuli, and External Dependencies, and files for adc_cosmic and adc_metrowerks. The main editor window displays the content of the ADC_Stimuli.in file, which consists of a list of stimuli commands for PIN PB1.

```
PIN PB1 -a 0.0
PIN PB1 -a 5.5 100
PIN PB1 -a 0 200
PIN PB1 -a 2.8 300
PIN PB1 -a 3 500
PIN PB1 -a 3.8 700
PIN PB1 -a 3.9 900
PIN PB1 -a 4 1000
PIN PB1 -a 4.3 1200
PIN PB1 -a 4.7 1400
PIN PB1 -a 5 2000
PIN PB1 -a 2.4 2020
PIN PB1 -a 2.8 3050
PIN PB1 -a 3 3500
PIN PB1 -a 3.8 3700
PIN PB1 -a 3.9 3900
PIN PB1 -a 0.2 4000
```

At the bottom of the window, the status bar shows "For Help, press F1", "Ln 11, Col 20", and status indicators for MODIFIED, READ, CAP, NUM, SCRL, OVR, Stop, and Ready.



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!

The screenshot displays the ST72254G2 Simulator interface. The main window shows a plotter with two signals: 'pn::PB1' (red) and 'pn::PB0' (green). The plotter has a menu bar (File, Edit, View, Project, Tools, Help) and a toolbar. The plot area shows a grid with two signals. The x-axis represents time, with markers at 135014.4, 135065.6, 135116.8, 135168.0, 135219.2, 135270.4, 135321.6, 135372.8, and 1354. The y-axis represents signal levels, with markers at 0.000 and 23714.715. The plotter also shows a table of signals:

| Name | Value |
|---------|-------|
| pn::PB1 | 1 |
| pn::PB0 | 1 |

The 'Plot Select' dialog box is open, showing a list of selectable items: Registers, Variables, and I/O Pins. The 'Selected Item' list is empty. The dialog also has a 'Manage unknown data type as:' section with radio buttons for 'Unsigned' (selected) and 'Signed'. The dialog has 'Add Item ->', '< - Del Item', 'OK', 'Cancel', and 'Help' buttons.

At the bottom of the simulator, there is a status bar with the following information: Active Marker Time: 202107.250, Time: 135104.928, Max Sim: 395856.702, Time Unit: us.