RN0008
Release notes

ST9 toolchain
release 6.2.0

Introduction

About these release notes...

This is revision 4.0 of the release notes for the ST9+ toolchain release 6.2.0.
The ST9+ toolchain includes the following elements:

- The ST9+ software compiler toolchain.
- ST9+ Visual Develop (STVD9), which is the ST9+ debugger and integrated development environment.
- Emulator monitor and sets of programmable logic.

These release notes are updated periodically in order to keep you abreast of any problems or limitations found in this release. Check the ST microcontroller support web site at *www.st.com* to ensure that this is the latest version of these release notes.

Online documentation

In order to access STVD9's online help (*.chm format), you must have installed Microsoft® Internet Explorer 4.0 or higher.

Customer support

For more information or help, please contact the nearest sales office. You will find a complete and up-to-date listing of ST offices and distributors at *www.st.com.*

Software updates

You can get software updates from the ST internet web site ST microcontroller support web site at *www.st.com*. For information on monitor and hardware revisions, call your nearest sales office.

www.BDTIC.com/ST

Contents

www.BDTIC.com/ST

www.BDTIC.com/ST

1 Read me first

1.1 Host PC system requirements

PC and compatibles running with Windows® 95, 98, NT® 4.0 (with Service Pack 3 or later), 2000 and XP operating systems. In order to access STVD9's online help (*.chm format), you must have installed Internet Explorer 4.0 or higher.

The host PC should have the following minimum system requirements:

- Intel® Pentium-class processor.
- 64 Megabytes of RAM.
- 50 Megabytes of free disk space.
- Super VGA monitor.

Note: Installing the ST9+ Toolchain on a host PC running Windows® NT, 2000 or XP, will trigger the installation of the STM Parallel Driver (parstm.sys) if it has not yet been installed. In this case, administrator rights are required.

1.2 Supported emulation hardware

All ST9+ HDS2V2 emulators are supported.

www.BDTIC.com/ST

2 What's new in ST9 Toolchain version 6.2.0

2.1 Changes in version 6.2.0 (Dec 2004)

What's new, compared to the previous release?

- **Project Settings** window for managing toolchain-specific build options.
- New MCU selection interface in the **Project Settings** window.
- Corrections and new features in the compiler toolchain.
- Corrections in the debugger.
- Corrections in the user documentation.

2.1.1 ST9 software debugger

Summary of user interface improvements

- Provides a **Project Settings** window that allows the user to customize the building of the application using the build options that are specific to ST9 Assembler-Linker and C toolchain.
- The MCU selection window is now integrated in the **Project Settings** interface. Users select from a full list of supported MCUs and the selection of the target MCU is done prior to building or debugging the application.

Other evolutions in the STVD9 debugger

- Perf Analysis now works even when your memory block is set to internal memory.
- The user can now rebuild an application while it is running and when he restarts a debug session, he will be sure to debug the most recent build of the application.
- STVD9 now provides easy access to variable values with the mouse.
- In a `Makefile` workspace, the `makefile build` arguments are now correctly saved. The number of lines in the **Watch** window is limited to 256, otherwise the time to display the array would be too long.
- Increasing the number of lines in the **Watch** window is now possible: In a debug session, do the following: **Tools>Options>Edit/debug**. In the **Watch Arrays**: `Expand ___ items field`, enter the number of lines to add in the **Watch** window.
- Evolutions in system include files for ST92F150/F250:
- Adding some lines in *sys/st92f150/rccu.[h/spp]*.
- Adding some lines in *sys/st92f150/sci_a.[h/spp]*.
- *ihex* and *s19* file generation is now available in the **Project Settings Linker** tab (Run converter check-box). This feature is no longer available in the **Tools** menu.
- **Dump Header** feature is now available in the **Project Settings Linker** tab (Dump Header check-box). This feature is no longer available in the **Tools** menu.
- The new way to change the paths of system libs and system includes is to select the **Tools>Directories** tab. The user can no longer individually set the paths of the system libs or the system includes from **Tools>Options>Toolset** menu.

Note: In case the user wants to move the system libraries to another location, he should leave the following sub-executables at the default location: as.exe, cc1.exe, cpp.exe, ld.exe.

- Memory mapping window new working:

www.BDTIC.com/ST

- In case there is a loaded workspace:
  The user can only change the mapping of the current reference, or reset to the standard reference mapping (he can't select a new reference any more).
  => **Reset to Standard Mapping** button replaces **Select Reference** button.
- in case there is no loaded workspace:
  The user can either change the mapping of the current reference, or select a newreference.
  => **Select Reference** button replaces **Reset to Standard Mapping** button.

Corrections in the STVD9 debugger

- Using *idprs* to pop the stack for example, didn't take into account the current value of *dprx*. No longer an issue, as *gdb9* now uses the *dprx* value instead of *idprs* to pop the stack.
- Corrects crashes of **Peripheral register** window contextual menu.
- Sources of the workspace tree view can no longer be removed while the build operation is in progress.
- Advanced breakpoints *sequential.bem* and *snapshot.bem* examples have been fixed.
- Now, default button applies to the active tab only (e.g. Compiler tab). When clicking on a **Default** button from a **Project Settings** tab wherever it stays, all the other tabs are no longer reset.
- STVD9 editor `Replace all` command is now working.
- Corrects spurious communication error after number of steps under Windows XP (STVD9 was losing synchronization with the emulator).
- Corrects issue when migrating an old workspace (v6.1.8) to STVD9v6.2 (source files were not kept).
- Corrects issue with reverse inheritance in Compiler/Assembler settings.
- Adding a file to a project from a network path is now properly handled.
- After building an application, I and S files are now generated in the same directory as O, U and MAP files.
- MCU selection is can no longer be modified in debug mode.
- There is no **Saving prompt** message after closing an unmodified workspace.

Corrections in gcc9 compiler toolchain

- In some cases, C statements (e.g. flag = CICR) were optimized in spite of manipulating a volatile variable (e.g. CICR). This optimizer problem has been fixed.
- In some cases, gcc9 with no optimization generated two extra ld instructions (see example below). Since register R254 is written to twice, if an Underflow interrupt occurs when the second instruction (ld r0,R254) is being executed, the interrupt is not registered in R254. This optimizer issue has been fixed.

```
ld r0,R254
ld R254,r0
```

- Internal compiler error no longer occurs when a function call with a parameter passed through the stack is converted at the same time through zero extension to a bigger type (from short to long in our case).
- The compiler no longer performs an `spm` instruction before loading the Switch table index, whereas it is stored in the stack (in some cases - lack of available registers- ,the compiler stores the switch table index into the stack).

www.BDTIC.com/ST

- In the case of some complex operations (e.g. `(((io_cnxn->Trigger)--) == 0)`), gcc9 was generating erroneous assembly code (incrementing was done before instead of after the next operator). This optimizer issue has been fixed.
- There is now a warning when a segment is overwritten.

2.1.2 ST9 user documentation

Enhancements and corrections:

- as9, ld9 & Binutils reference manual
- gcc9, Libraries & Startup files reference manual: On page 91, new example of asm instruction: `asm ("addw %0,%2" : "=r" (foo) : "0" (foo), "g" (bar));`
- User manual
  - The section 9.2.11 bitfields allocation has been added in user manual to explain that ST9 compiler allocates bit fields from MSB to LSB.
  - Updates to Chapter 6 describing how to set up a new workspace, change project build options and build the tutorial application.
  - Addition of Appendix B describing how to create a new workspace from an existing makefile.
- V4 to V6 migration notes: no modifications.
- Online help
- Corrections in *st92f150-emu2\v1.05\readme_flash.txt* , and *st92f120-emu2\v1.05\readme_flash.txt*

2.2 Important information about this version

Caution: Do not change the format of the system include files, located in *..\st9-elf\2.7.2\include\sys*. These files are necessary when creating new STVD9 application windows that display all of the registers with their symbolic names.

2.3 Targets supported by this version

All ST9+ MCUs supported by ST9 HDS2V2 emulators are completely supported by this ST9+ Toolchain version.

www.BDTIC.com/ST

3 Known problems/limitations of this release

3.1 Known problems/limitations in this release of STVD9

- In the **Advanced Breakpoints** window, if you try to reset event 1 of level 1 several times by clicking on the **Reset level1** button, and then return to level 1 - event 1, graphical problems with the **Advanced Breakpoints** window may occur.
- The **Peripheral Registers** window can only display peripheral register information for the following MCU targets: ST92F150, ST92F120, ST90158, ST92E141, ST92163, ST92195.
- If the path and filename combination of your project is greater than 64 characters, you cannot build your application under Windows® 95 or 98. You must reduce the length of the filename, or reduce the number of folders in your path.
- In **Watch** window, there is a display issue when you try to expand some table type structure fields. E.g. in:
  `struct Stest { unsigned char tab1[10]; } Stest;`
  If you expand the tab1 field, the displayed value is wrong.
- Declaring 2 distinct program memory sections inside a single source file can lead to unpredictable behavior (e.g. in the **Advanced Breakpoint** window, you will not be able to set the range of an Opcode Fetch event).
- In the **Mapping** window, the “commercial” devices’ Flash memory (e.g. ST92F150JDV1) can only be defined as system (internal is not allowed).
- The environment variable $(WorkspacePath) does not work, i.e. does not override the default paths used by STVD9. This variable can be applied in **Tools>Customize** and the **Prelink**, **Post Build** and **Custom Build** tabs of the **Project Settings** window.
- **Break on register** is always in "single shot" mode: When you set a break on register, and run your program, it stops correctly on the first break. But when you open again the **Break on register** window, the break is no longer active.
- When operating under Windows 98, commands in the Post Build step cannot save .i and .s files in the output directory if these files have already been created in this directory.
- Contextual help (F1) for the **Project>Dependencies** window does not work.
- After uninstalling STVD9, HKEY_LOCAL_MACHINE/SOFTWARE/STMicroelectronics/ST Debug is not deleted from the registry database.
- After opening a workspace for the first time, an asterisk (indicating an unsaved change) appears next to the workspace name in the banner of the **Workspace** window. The asterisk is removed when the user saves the workspace.

3.2 Workarounds to common STVD9 problems

The following sections provide workarounds to common problems in STVD9.

How do I load two (or more) independent applications in emulator memory?

It is possible to download 2 (or more) independent applications in the emulator memory while having debug information for one application only. With V6.x.y toolchain you have to:

www.BDTIC.com/ST

1. Generate an Intel files (*.hex) of the files you want to download without debug information. Two ways to do this:
   a) In your makefile add the following line:
   `bjcopy9 -O ihex filename.u filename.hex`
   b) Use **Run Convertor** check box in the LInker tab of STVD9's **Project Settings** window.
2. Open the workspace for the application that you want to debug with the debug information.
3. Open the **Memory** window (**View>Memory**).
4. Right click in the **Memory** window and select **File>Restore** layout.
5. Select the Hex file that you want to download without debug information.
6. Do a `Debug>Chip-reset` command.

What does the message box "Failed to send command to gdb." mean?

It means the underlying debugging software has crashed. You must close and restart STVD9.

What do I do if STVD9 indicates that the debugger has been working for an abnormally long time?

This usually means that the underlying debugging software has crashed for some reason. You will have to close and restart STVD9.

Can I use spaces in source filenames?

You may edit such files, but you won't be able to build these files with the toolchain.

What do I do if I experience problems with the visual environment, such as misplaced windows, or missing or misplaced graphical objects with some windows?

The solution is to delete the two files <myworkspace>.wed and <myworkspace>.wdb.

How do I load an application into the flash memory?

Before loading an application into the flash memory, you must be sure that either the current flash sector has already been erased (i.e. each bit has been set to 1), or the application starts at the beginning of one flash sector (that will erase the whole flash sector).

To use flash in STVD9 you first have to declare the flash area as INTERNAL memory in the **Memory Mapping** window (**Emulator>Memory Mapping**). Then you have to stop and restart the debug session, so your program will be loaded in flash.

Note: 1 When loading your application, the Flash sector is erased when the first address of the sector is written to. So for each sector of your program must start at the first sector address.

2 The only way for the hardware to write a bit in flash memory is to reset its value from1 to 0. (e.g. if you try to write 0x55 in place of 0x50, STVD9 will only write 0x50 and then display the message: "gdi-emu in hds: Required memory cannot be written").

www.BDTIC.com/ST

How do I write some data into the Flash from the “Memory” window?

Before writing data in Flash from the **Memory** window, you must be sure that the current Flash sector has been erased, otherwise you get the message: gdi-error in hds: Required memory cannot be written.

Note: The flash sector is erased when the first address of the sector is written to.

How do I troubleshoot the load into a not erased Flash memory?

You may experience problems when attempting to load an application not from the beginning of a Flash sector. In this case you may not be able to restart a debug session. As a workaround, you can restore a SYTEM map by editing the "*filename.wsp*", where you will find the following lines:

```
set_map ORIGIN=0x000000 END=0x00001f TYPE=I*RW
set_map ORIGIN=0x000020 END=0x01ffff TYPE=I*RW
```

that you can modify "`I*RW`" by "`S*RW`" as follows:

```
set_map ORIGIN=0x000000 END=0x00001f TYPE=S*RW
set_map ORIGIN=0x000020 END=0x01ffff TYPE=S*RW
```

3.3 Known problems in this release of the ST9+ software compiler

- While the ST9+ assembler can now handle binary integers like 010101010b correctly, it can no longer handle binary integers like 0b01010100.
- In the map file, (obtained with `-map` option of `ld9`), the given `REGFILE` size is not aware of registers allocated manually by the user. Thus the percentage only takes into account registers allocated automatically (i.e. the size you have reserved in the register file from the linker script file).

3.4 Workarounds to common ST9+ software compiler problems

How do I prevent stack overflow?

Use the “`-mcheck-stack`” option, which provides protection against stack overflow.

www.BDTIC.com/ST

4 Release information for previous releases

4.1 ST9+ Toolchain V6.1.8 (March 2003)

What's new, compared to the previous release?

- Correction and new feature in compiler toolchain.
- Correction in debugger.
- Correction in monitor.
- Correction in user documentation.

4.1.1 ST9 compiler toolchain

Enhancements and corrections

Corrections in gcc9 compiler:

When incrementing or decrementing the field of a far structure pointer, the compiler will no longer generate some erroneous assembly code.

Enhancements in objcopy9 :

The `IEEE-CallDS` object file format is now generated by `objcopy9` (see as9, ld9 & Binutils Reference Manual).

4.1.2 ST9 software debugger

Mapping enhancements:

- ST92F150 devices names and mapping have changed according to the last datasheet of Dec 2002 Rev 1.3.
  - target_emulator=emulator_ST92F150
  - target1=ST92F150JDV1
  - target2=ST92F150JCR1_V1
  - target3=ST92F124R9
  - target4=ST92F124V1
  - target5=ST92F250CV2
- Homogeneous zones: Testflash and Flash-otp sectors are now homogeneous. Homogenous zones control All areas that are part of an homogeneous zone still need to have the same memory location (I/S/U), but their access type (R/W) is no longer meaningful. Hence Flash sectors 0/1/2/3 can be defined as R or R/W, testflash as R (only) and flash-otp as R or R/W.
- `emulator_ST92F150` Flash location can now be either `internal` or `system`.

Note: For other devices (i.e ST92F150JDV1), flash can only be defined as system.

4.1.3 ST9+ emulator monitor and sets of programmable logic

New ST92F150-EMU2 V1.05 Downloader (for V2.28 monitor version) is available. It includes the following enhancements:

www.BDTIC.com/ST

TRACE

- Testflash initialisation routines (i.e. routines that are run after a reset command) are no longer displayed in **Trace** window. Consequently, after a CALLS instruction (0x3F) has been done from monitor (seg21) to testflash (seg23), all the following accesses don't appear any more in **Trace** window:
  - program in segment 23,
  - data read/write in segment 22,
  - data read in seg00-add6/7 if data=0xff (external watchdog),
  - data read in seg00-add8/9 (external watchdog).

*Note: 1 User watchdog program still appears in **Trace** window.*

*2 User ISP program located in RAM starting at segment 20 address 0018 still appears in **Trace** window.*

- After a run command, accesses in segment 00 addresses 0,1 (reset vector) are no longer displayed in **Trace** window.
- The "noise" occurring between a reset command and the first monitor instruction (i.e. instruction in seg21) doesn't appear any more in **Trace** window.
- When an application is reading some data in internal memory, data trace is done on emulation memory instead of internal memory because ST92F150 doesn't output data. Hence when internal flash is selected, the monitor now duplicates internal memory contents to emulation memory to be able to trace data:
  - it copies internal testflash contents into emulation memory,
  - it loads user program into both memories (internal and emulation/system),
  - and at runtime, it writes data into both memories.

TESTFLASH

Subroutines located in testflash can now be called even if Flash memory is defined as System.

4.1.4 ST9 user documentation

Enhancements and corrections

- as9, ld9 & Binutils Reference Manual: In section 3.7.1.32", the definition of @nobits has been enhanced.
- gcc9, Libraries & Startup Files Reference Manual: On P.80, the sentence: "the condition that bit DPRREM=0 is a strong requirement in the way far pointers are handled by the compiler.", has been replaced by: "the condition that bit DPRREM=1 ...".
- User Manual: In Section 6.6.6, added explanation about the use of .section FLAGS argument
- V4 to V6 Migration Notes: No modifications.
- Online help: No modifications.

www.BDTIC.com/ST

4.2 ST9+ Toolchain V6.1.7 (December 06, 2002)

What's new, compared to the previous release?

- Correction in ST9+ software compiler toolchain.
- Correction and new features in ST9+ debugger.
- Correction and new features in user manuals.

4.2.1 ST9 software compiler toolchain

Corrections:

- gcc9 compiler:
  When building an application under a Windows 2000 platform, you will no longer experience problems with cpp, when it can't find the TEMP directory: temporary directory no longer depends on the platform since cpp now uses its own environment variable GCCTMP, initialized automatically at the installation.
- _overflow_stack function no longer includes the stack size check, when compiling with the stack overflow check -mcheck-stack option.

Enhancements:

The objdump9 --variable option has been added. It retrieves both information of addresses and size for all kinds of variables (see as9, ld9 & Binutils Reference Manual).

4.2.2 ST9 software debugger

Corrections:

- There is no longer a troubleshooting about stepping into the following Assembly instructions: calls (R),(rr), calls (r),(rr), jps (R),(rr), jps (r),(rr).
- A step instruction followed by a change of PC value no longer duplicates last trace line, in **Trace** window.
- In the **Memory** window, the suffix ".hex" is now automatically added to the filename prefix.
- In the **Memory** window, **File>Save layout** contextual menu now saves correctly the last address of your range.
- In the **Memory** window, **File>Save Layout** contextual menu doesn’t allow you any more to save non existent memory: it now displays an error message, and no longer creates an empty file.
- In the **Memory** window, you can’t write any more inside some forbidden memory (e.g. in some non erased flash) without a warning.
- In the **Memory** window, after each memory write, the whole window is now refreshed and you can watch exactly the contents of hardware memory (the memory cache in gdb9 has been disactivated).
- There is no longer a troubleshooting about loading an application in flash memory, once the map has been set to INTERNAL from the **Mapping** window (STVD9 now sets the INTERNAL mapping before loading the application in memory)
- In the **Trace** window, the **Save trace contents contextual** menu has been replaced by a **Copy to File** menu item, which leads to a more easy-to-use dialog box.

www.BDTIC.com/ST

- In the **Trace** window, it is now possible to display only DMA access lines from trace buffer.
- In the **Memory Mapping** window, mapping of ST92163,has been enhanced (possibility for SEG23 to be external)

Enhancements:

- In the **Advanced Breakpoints** window, three new memory events are now available in the combo-box of Memory Event: DMA memory, IT memory and stack memory. e.g. you can now choose a DMA dress range as an event).
- **Break on Register** window has been modified in a previous release: when setting a register value, the user can either select an hexa value, or a binary value.

4.2.3 ST9 emulator monitor and sets of programmable logic

- New ST92F120-EMU2 V1.05 Downloader (for V2.23 monitor version) is available. An issue concerning the writing of data into FLASH or EEPROM memory has been resolved (the first time the user was trying to write inside some not erased flash memory, the monitor was correctly displaying an error message, however even after erasing the whole flash sector, monitor was wrongly displaying this message). Monitor now resets the flash/eeprom write error bit. As a result a write Error message is now displayed only one time)
- New ST92F150-EMU2 V1.04 Downloader (for V2.27 monitor version) is available (same fix that for ST92F120-EMU2 V1.05).

4.2.4 ST9 user documentation

Enhancements and corrections:

- as9, ld9 & Binutils Reference Manual : In section 3.7.1.26, a note was added saying that you can use the \@ variable to implement a local label in a assembly macro.
- gcc9, Libraries & Startup Files Reference Manual
  - Added new functions (seg, sof, pag, pof) to Library chapter (See sections 3.74, 3.75, 3.76 and 3.77)
  - The section 2.7.1 was extended to explain what to do if you have an error message that doesn't seem to make sense.
  - The section 2.2.13 Options for code generation conventions was extended, to include -fno-short-enums description of option.
  - The section 2.9.2.1 Scalar types was extended about enum type.
  - In sections 2.8.4.4 and 2.8.4.5, examples 1 and 2 have been altered
  - In section 3.5.4.1, the NO_SECINFO description was extended.
- User Manual
  - Added explanation of what to do if you have an error message that doesn't seem to make sense.
  - Added in section 9.4.7 ST9+ tools: "blanks in Window pathnames are not supported".
  - In section 4.1.1.1 DPR-mapped data object, multimapping examples has been added.

www.BDTIC.com/ST

  - In section 9.2.3, Pitfall added: the initialized value of a #pragma REGISTER FILE variable is always stored in .secinfo section.
  - In section 9.3.4, a note was added about volatile bidirectional port.
- V4 to V6 Migration Notes enhancements in: Modify chap 3.1.3 Service symbols.
- Online help
  - In the **Advanced Breakpoints** window subtopic, the new Single events is described.
  - In the **Trace** window - Copy to file contextual menu, pitfall added: in saved file, the timestamp unit is always "clock Cycles".

4.3 ST9+ Toolchain V6.1.6 (May 29, 2002)

What's new, compared to the previous release?

- Installation process modified.
- Banner of GNU components modified (e.g.: objdump9 -V).
- New license files.

4.4 ST9+ Toolchain V6.1.5 (March, 2002)

What's new, compared to the previous release?

- Delivery of ST92F250 include files
- Correction in ST9+ software compiler toolchain.
- Correction in ST9+ debugger.

4.4.1 ST9 software compiler toolchain

New features for ST92F250 chip:

System include files for ST92F250 are now delivered with the V6.1.5 system libraries. Corrections:

- gcc9 include system files of ST92F150/F250: Added in *sys/flash.[h|spp]* the following defines:

```
#define FLASH_REG_PG    ((unsigned char)0x88)  /* Page control
register */
#define PROT_REG_PG ((unsigned char)0x8C) /* Flash Protection
Page Register */
```

- gcc9 compiler:
  - The optimizer was using too much memory (in the stack) in the case there was several time the same operands in a program bloc. That's now solved. The stack doesn't grows up anymore and the code is slightly better.
  - In case of combination between far pointer and bit field, there had an Internal compiler error (program cc1 got fatal signal 22). That is fixed now.

www.BDTIC.com/ST

– There had an error in the optimizer in the condition of several peephole optimizations, leading to a compiler error (program cc1 got fatal signal 22).That is fixed now.

4.4.2 ST9 software debugger

New feature for ST92F150 emulators: Memory mapping of ST92F250 chip is now supported, and can be loaded from an ST92F150 emulator.

Corrections:

- There is no longer a memory leak induced by the use of the visual environment, when you used to change several times you focus inside stvd. Example given: in the previous stvd9 release, clicking in several windows inside your edit or debug graphical environment could run your system out of memory.
- No more problems when writing some data into ST92F150/F250 internal flash memory: writing in flash can be very slow depending on the production lot. Consequently, the debugger memory read/write time-out has been increased.

4.5 ST9+ Toolchain V6.1.4 (October 26, 2001)

What's new, compared to the previous release?

- Correction in ST9+ software compiler toolchain.
- Emulator monitor changes.

4.5.1 ST9 software compiler toolchain

Corrections:

- In previous releases, the libst9plus.a library was forgotten for all models but compact memory model. It is now available for all models.
- libst9plus.a library contains some support routines that give seg, sof, pag or pof on near/far data at runtime. These routines are named st9_sof, st9_seg... for near data, and _f_st9_sof, _f_st9_seg... for far data.
  These routines are now available for all programming models—the page pointer is changed when getting the page value on compact/specmed memory model. The functions are defined in the <sys/st9plus.h> header file.

4.5.2 Emulator monitor changes

New features:

- New ST92F120-EMU2 V1.04 Downloader (for V2.22 monitor version) is now available. With this new monitor release, the problems that concern the writing of data into EEPROM have been resolved, and internal Flash memory is supported (see readme_flash.txt).
- New ST92F150-EMU2 V1.03 Downloader (for V2.26 monitor version) is now available. Internal Flash memory is now supported (see readme_flash.txt).

www.BDTIC.com/ST

4.6 ST9+ Toolchain V6.1.3 (September 14, 2001)

What's new, compared to the previous release?

- Corrections in STVD9.
- Corrections in software compiler toolchain.
- Corrections in emulator monitor.

4.6.1 STVD9 V6.1.3

Corrections:

- Mapping of ST9216x: The default RAM Memory mapping for the ST9216x is now corrected.
- ST92F150 **ST9 peripheral registers** window: the “intrusive read” information is now available on *eftimer.h* and *sci_a.h*.

4.6.2 ST9 software compiler toolchain

New features:

- A new library libst9plus.a is available (for compact model only). It contains the following set of useful functions
  - `st9_seg`,
  - `st9_sof`,
  - `st9_pag`,
  - `st9_pof`
- These support routines allow to get seg, sof, pagor pof on near/far data at runtime. These routines combine into functions: `st9_...` for near data, and `_f_st9_...` for far data.
  The functions are defined in the <*sys/st9plus.h*> header file.

Corrections:

- as9: Correction concerning long jump resulting from a condition. After a condition, the compiler generates a `btjxt <branch label>` instruction. This is a pseudo-instruction (see assembler documentation) which is implemented by the assembler either as `btjf` or as a combination of `tm` and `jp<cc>` according to the relative position of the branch label. In the case of a close label, the instruction “`btjf <branch label>`” can be used, but in the case of faraway label, the instructions “`tm r1,#0x08`” and “`jpnz <branch label>`” must be used. In this last case, the object file generated by the assembler was incorrect because “`jpt <branch label>`” was used, instead of “`jpnz <branch label>`”. This has now been corrected.
- gcc9:
  - For the ST92F150 system including the *eftimer.[h|spp]* file. The problem in the *sys/st92f150/eftimer.[h|spp]* file has been fixed by adding the following line: `#pragma register_file E0_ACR 246`
  - For the ST92F150 system when including files *eftimer.h* and *sci_a.h*. New comments have been added in the *sys/st92f150/eftimer.h* and sys/st92f150/sci_a.h files.

www.BDTIC.com/ST

– For the ST92F120 system including the *eftimer.[h|spp]* file. The problem in the *sys/eftimer.[h|spp]* file has been fixed by adding the following line:
`#pragma register_file E0_ACR 246`

4.6.3 Emulator monitor, and sets of programmable logic

New ST92F150-EMU2 V1.02 Downloader (for V2.25 monitor version) is available. With this new monitor release, the problems concerning the writing of data into EEPROM have been resolved.

4.7 ST9+ Toolchain V6.1.2 (June 27, 2001)

4.7.1 ST9 software compiler toolchain V6.1.2

Corrections:

- gcc9:
  - Correction concerning return from an interrupt routine. The return from an interrupt routine at the end of an else statement is now correctly executed, an `iret` instruction is generated instead of an `rets` instruction.
  - Enhancement to the `-mcheck-stack` option. The `-mcheck-stack` option now tests stack underflow—i.e. there is now a call to an error routine, if the stack pointer is strictly lower than the stack smaller boundary. In V6.1.1, `-mcheckstack` option was designed for testing stack overflow (i.e. testing if the stack pointer is greater than the stack upper boundary). When using gcc9 with a backward stack growth, this testing of stack overflow was not useful for the user.
  - Correction to system include files *can[0|1]*. The unterminated comment problem in *sys/st92f150/can[0|1].[h|spp]* has been resolved.
- libc: Implementation of `seg/sof/pag/pof` C routines. Added support routines to get `seg`, `sof`, `pag` or `pof` on near/far data at runtime. These routines result in the new functions: `st9_...` for near data, and `_f_st9_...` for far data.
  These routines are now available for all programming models—the page pointer is changed when getting the page value on compact/specmed memory model. The functions are defined in the <*sys/st9plus.h*> header file.

4.8 ST9+ Toolchain V6.1.1 (May 21, 2001)

- The ST92F150 HDS2V2 emulator is now supported.
- An emulator monitor downloader is now delivered. It allows you to upgrade your emulator monitor to support different DBEs (emulator probes). For more details, refer to *C:\ST9PlusV6.1.1\downloader\readme.txt*.

4.8.1 STVD9 V6.1.1

New features:

The ST92F150 HDS2V2 emulator is now supported by STVD9, and ST92F150-specific information is included in the ST9 **Peripheral Registers** window, which been validated by the application field.

www.BDTIC.com/ST

Corrections:

- The problem of stopping debug session when the program is running on the emulator has been fixed. Previously, it was impossible to restart debugging, with a message indicating that “the executable could not be found”, due to a lock of the file. This problem has now been corrected.
- The problem with the ordering of source directories in the **Workspace** window has been resolved. Previously, when there were several source files with same name in the **Workspace** window, the source directories were not correctly sorted. This has been corrected.
- The **Recent Workspaces** section of the **File** menu has been made larger in order to better display the project path.

4.8.2 ST9 software compiler toolchain V6.1.1

New features:

ST92F150 *include* files are now delivered, (they have been built and validated by the application field). These new ST92F150 include files are located in:

C:\ST9PlusV6.1.1\lib\gcc-lib\st9-elf\2.7.2\include\sys\st92f150

Corrections:

Instantiating a far pointer with the far address field of a structure is now handled correctly by the compiler (see example below):

```
Definitions:
   typedef struct
     USEROBJDEF far *pLogoStruct;
     } LOGOINFO;
   LOGOINFO far *pLogoInfo;
   USEROBJDEF far *pLogo;
Fixed problem:
   /* there was a problem when doing this kind of indirection */
   pLogo = &pLogoInfo->pLogoStruct[recvb];
```

- Previously, the `-O3` compiler option generating a `jrcnz` instruction did not work. This problem has now been corrected.
- Correction of system library file “*wk_up_it.h*”: `WMm_imwu7l` replaced by `WUm_imwu7l`, and `WMm_imwu7h` replaced by `WUm_imwu7h`.
- Correction of system library file “*imc.h*”, in order to display correctly the page 48 inside **peripheral registers** debugger window.

4.9 ST9+ Toolchain V6.1.0 (March 2001)

- This version of the ST9+ Toolchain can be run on the Windows® 2000 operating system.
- New functions in STVD9.
- New functions in the ST9+ software compiler.

www.BDTIC.com/ST

4.9.1 STVD9 V6.1.0

New features:

- Workspace file. Workspace files (<*workspace*>*.wsp*) no longer include information on the window size and position. It now only contains information related to the project (such as build command lines, and executable filename) or related to the debug session (such as memory mappings, breakpoint settings). The visual environment is now stored in two separated files: <*workspace*>*.wed* (for “workspace edit”) and <*workspace*>*.wdb* (for workspace “debug”).
- **Symbols Browser**. The **Symbols Browser** window is now resizable. The window now includes one more column to separate the type and the symbol name. Click on the list header to sort the column. A Correction has also enabled the display of underscores.
- Editor breakpoints. Breakpoints are saved when you leave a debug session. If the source file is modified, they move along with the source line to which they are attached and they are restored (insofar as possible) when you return to Debug mode. Therefore, now the **Instruction Breakpoints** window is available when editing in Build mode.
- **Break on Registers**. This window now displays the register value and mask with added control possible. The configuration is automatically enabled as soon as it has been modified. A message-box is displayed when validating a disabled configuration.
- **Disassembly** window. When no symbol found, the address is displayed at the top of the window (ensuring the PC is displayed). When a page is displayed with the PC/Frame pointer/breakpoint mark not exactly on the start address of a line, it is displayed with an different color. New option set in **Options>Edit>Debug** allows the automatic opening of the **Disassembly** window upon a stop, when PC is not in any source file.
- **Registers** window. Several copies (up to 4) of this window are possible at once. Fixed several problems concerning the cursor display and the ability to enter/modify information in the different copies of the **Register** window.
- **Peripheral Registers** window. New window that shows the peripheral registers of the target MCU with their complete names. Supported MCUs are ST92F120, ST90158, ST92E141, ST92163.
- **Debug Options**. In the **Debug Options** dialog box (accessible from the main menu by selecting **Tools>Option>Edit>Debug**) the option 'Beep when execution stop' has been added. If this option is selected, a beep will be emitted when the emulator stops for the following reasons:
  - Occurrence of an instruction breakpoint.
  - Occurrence of an advanced breakpoint.
  - Occurrence of a break on registers.
  - If Execution has been aborted because of errors (Write Protect access, Non existent memory access...).
- Advanced Breakpoints. When the BEM is disabled:
  - the tree control is greyed, but it is still possible to navigate through the BEM configuration.

www.BDTIC.com/ST

  - in the synoptic representation the squares figuring the levels are also greyed, and a sentence explicitly says that the configuration is disabled. Any BEM configuration modification will enable the BEM, which results in:

    - the tree control becoming ungrayed.

    - the synoptic the level squares painted according to their status.

    - the BEM configuration status becoming “configuration is enabled (modified)”.

- **Trace** window:
  - Timestamp output format has slightly changed—now each group of 3 digits are separated with a blank character.
  - A new contextual menu option “**Select timestamp unit**” allows you to choose the timestamp unit from 4 different units: seconds (s), millisecond (ms), microseconds (µs) and nanoseconds (ns).
- **Workspace** window. The window to browse for folders now includes an option to store the folder with a relative or an absolute path.
- Miscellaneous:
  - A dialog box is opened to stop the 'copy to file' operation.
  - The default color for the preprocessor block lines is now blue, like the default for MS Visual C++.
  - Added option in STVD9 to log result of **Build** window in a file named “*build.res*”, and locate it in the output directory.

Corrections:

- Toolset output parser fixed the problem to display some messages in bold.
- Advanced Breakpoints:
  - When saving a BEM configuration, a coherency check is done. It was previously possible to save an erroneous BEM configuration even though it was impossible to load it again.
  - Opcode fetch memory event definition has been improved. Previously no warning was issued when entering an address that was inside a source line but did not match with the beginning of an assembly instruction. Now in such a case the file/line edit fields remains empty and confirmation is requested when the event definition is validated.
  - Previously, if you set a valid opcode fetch on an assembly instruction and re-opened the event, the address that was shown was the address of the first source line instruction and not the address of the assembly instruction on which the event was set. This has been corrected.
  - When adding a new memory event, the item tree corresponding to the type of memory item in the tree control (opcode fetch, memory access, opcode fetch with data, memory access with data) was not expanded.
  - When the analyser probe end value becomes greater than the first value, the first value is updated with the last value.
  - The event counter value is now expressed in decimal format both in the tree control and the dialog box edit field.
  - When the event counter is selected, the event negation is expressed the same way both in the tree control display and in the selection combo box, for example: COUNTER (EVENT1)<1273).

www.BDTIC.com/ST

  - When programming an ELSEIF Event3 and Event4 combination the tree control was displayed correctly (i.e. “Event3 and Event4”) but if the **Advanced breakpoints** dialog box was re-opened after validation, the tree display was “and event3 event4”.
  - Advanced breakpoints settings are now saved in workspace (they are no longer lost when closing a project).
- **Trace** window. Previously, when resizing the trace window in some circumstances the window size could be reduced to nothing. This has been corrected.
- **Workspace** window:
  - Display of files inside a folder is no longer altered by changing the current directory (i.e. opening a file from somewhere else on the disk). Now it is always related to the project directory.
  - Window to browse for folder should always be open, even when the path is relative.
- **Call Stack** window. Corrections allow this feature to:
  - Work correctly even if “`__Reset`” procedure is not in “`.init`” section.
  - Work properly if the PC is inside an assembler far function.
  - Display the whole stack, including “`_Reset`” procedure at level 0.
  - Work properly even when the “`__Reset`” procedure is not inside a section named “`.init`”.
- **Break on registers** window: Inside “**Break on registers**” window, setting a register breakpoint with the range from R250 to R255 is now allowed.
- **Watch** window. The **Watch** window now properly displays variables mapped in a register files, and also properly displays “`bob.var1`”, where the “`bob`” type is a structure that includes a field “`var1`” typed array[x][y].
- Concerning memory sections, STVD9 can now highlight the source line inside a “`__Reset`” procedure, even if the PC is inside a “`__Reset`” procedure is not in the “`.init`” section. In the event of an executable file without a section named `.text`, when the application is loaded into STVD9 the error msg: “Can't find .text section in symbol file” will occur. In addition, no symbols will be available for the debugging session.
- Project building: Under Windows® 95 or 98, the “Build succeeded” message was displayed even when the build couldn’t be completed due to a linker error. Now, to avoid a false 'build succeed' message, a check is done on time of application file; the time must differ between before and after build process (and size must not be 0).
- **Editor** window:
  - Inserting a bookmark on a line by selecting the line (clicking in the margin) and pressing F8, now works.
  - Previously, the **Edit>Goto** feature did not work properly, especially when the file containing the function being searched was not yet open. (Sometimes the result of the Goto feature was just to open the file where this function was.) This has now been corrected.
  - A whole project can now be moved under Windows® Explorer to another location, without having the error message: “cannot open the file <file name>”. STVD9 now always saves the relative path name for editor files.
- Runtime features:

www.BDTIC.com/ST

  - The stack pointer and PC are no longer lost when stopping after a “continue”, in an optimized application, compiled in medium mode, with stack in another segment than code.
  - If the program is stopped in a function returning a structure, the return value displayed by step-out command is no longer erroneous.
  - A communication problem between the PC and the emulator (for example, if you switch the off/on button on the emulator), is no longer followed by a 6-second time-out before the message “Error: Emulator answer reception time-out” is displayed.
- Miscellaneous:
  - In the output console when the verbose mode was set, the focus was lost when editing an opcode fetch event line in the **Advanced Breakpoints** window or in the **Performance Analysis** window.
  - The “chip variant” information and the “memory mapping” configuration are now stored in the workspace file.
  - In the “Cmd line” field of a MS-DOS session, the size of the command for installing the toolchain has been reduced. As a result the user can modify the cmd line without it being truncated.
  - The *_istmp1.dir* directory is now removed after installation.
  - It is now possible to launch a "*.bat*" file from the **Build** menu of STVD9; the path of gcc9 is now kept inside a `$PATH` environment variable.
  - "*gdi9-hds2.dll*", and "*st9-hds2v2.dll*", are now striped.
  - In the "compact" and "medium" examples, the `makefile` can now have a name different than the workspace’s name.
  - With the compact and medium example makefiles, building the application generates the file <*application.hex*>, in Intel-Hex format, in addition to the file <*application.u*>. which is built.

4.9.2 ST9 software compiler toolchain V6.1.0

Corrections:

- *as9* does not permit nested procedures anymore. Warnings are emitted if nested procedures are defined, and only one '.proc' and '.endproc' sequence is kept (the outermost one).
- When allocating a long variable in the register file, the memory transfer between this variable and the memory is now handled correctly.
- The C-compiler was previously only generating 1 line out of 2 when using optimization level 2 and loading an array inside a loop. This problem has been fixed in the optimizer.
- In the `ASM` statement, a minor correction in the compiler was performed to permit it to work with the constraint 'T'.
- In the `ASM` statement, constraint 't' is not supported anymore.
- Previously, the assembler couldn't handle binary integers like 010101010b correctly. (For example, .equiv m_SDA, 0000010b). The default binary format accepted by gas compiler is 0[bB][01]*. But, for compatibility with tr9, as9 can now support binary integers like [01]*[bB],
- The *cpp* (C preprocessor) now correctly handles assembler comments beginning with a ';'. The option '`-lang-asm`' is used on the cpp command line to inform cpp that the input source file is an assembler file and the comment using ';' must be handled.

www.BDTIC.com/ST

4.9.3 System include files

- Correction in “*imc.h*”.
- Soft modifications (new comments, change in declaration order, ...) in all system includes files, needed by new “**ST9 peripheral registers**” window.

4.10 ST9+ Toolchain V6.0.0 (April 2000)

This is the first release of the ST9+ Toolchain including STVD9, a 32-bit debugging software and integrated development environment. The “V6” toolchain replaces the 16-bit toolchain whose last release was V4.3.3 (delivered with the WGDB9 debugger).

Note: There is a rupture in compatibility between the old ST9+ Toolchain V4.33 and the new ST9+ Toolchain Version 6.0.0 (and later).

www.BDTIC.com/ST

5 Revision history

Table 1. Document revision history

Date	Revision	Changes
07-Jan-2005	1	First release.
07-Feb-2005	2	Updated Section 3.1 with: Environment variable $(WorkspacePath) limitation Post build save limitation for Windows 98 users Uninstall limitation Contextual help limitation
04-Aug-2006	3	Template update Minor text corrections
16-Mar-2009	4	Revalidated. Document revision number on cover now corresponds with revision history.

www.BDTIC.com/ST

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER’S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2009 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com

www.BDTIC.com/ST