ST200-TOOLSET

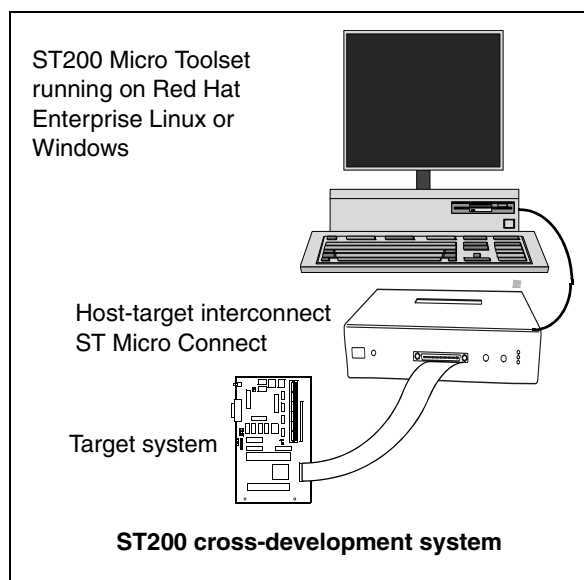## ST200 Micro Toolset for developing ST200 applications

Features

- GNU code development tools
  - ANSI C and C++ compiler for ST200
  - C (**newlib**) run-time library and C++ (**libstdc++**) including STL
  - GNU archiver and other utilities
  - Board specifications defining memory area for linker
  - Boot from Flash ROM support
  - Extensive set of trace and profiling tools
- Cross development with GDB
  - The GNU debugger supports the ST200 simulator and development boards
  - Includes Text User Interface (TUI) and the Insight GUI on all host platforms
  - **st200xrun** target loader
- OS21 real-time kernel, provides:
  - High performance, low memory footprint
  - Tasks, semaphores, message queues
  - Memory management and virtual memory API
  - Mutexes and events
  - Interrupt handling, cache control and timers
  - Sources supplied, royalty free
- Instruction set simulator
  - Programs can be run without hardware
  - Low-level execution tracing
- Interface to the target through ST Micro Connect and ST TargetPack technology (supplied separately)
- STWorkbench IDE (supplied separately)
  - Built on the Eclipse IDE
  - State-of-the-art IDE for use with a range of ST toolsets
  - Powerful editing tools and ST200 specific plug-ins customize the IDE for building and debugging ST200 target applications
  - Advanced trace and profiling tools

ST200 Micro Toolset running on Red Hat Enterprise Linux or Windows

Host-target interconnect ST Micro Connect

Target system

ST200 cross-development system

Description

The ST200 Micro Toolset supports application development for ST200-based systems.

Applications can be efficiently implemented in ANSI C or ANSI C++ and mapped to the target hardware.

In conjunction with the on-chip emulation support of the ST200, the ST200 Micro Toolset supports board level debugging through the JTAG interface. The tools can be used through the STWorkbench Integrated Development Environment (IDE).

The toolset supports host-target interconnection through the ST Micro Connect[a], using either Ethernet or USB connections.

---

a. The original ST Micro Connect product was named the "ST Micro Connect". This product is now known as the "ST Micro Connect 1" and the term "ST Micro Connect" refers to the family of ST Micro Connect devices.

www.st.com

Contents

1 Introduction

The ST200 Micro Toolset is a cross-development system for developing and debugging C and C++ embedded applications on ST200 board devices. All ST200-based devices include the Debug Support Unit (DSU), available through the JTAG port of the device that provides on-chip emulation capabilities such as:

● code and data breakpoints

● memory peeking and poking

The ST200 Micro Toolset can be enhanced by the STWorkbench IDE (supplied separately). STWorkbench is built on the Eclipse IDE, providing a state-of-the-art IDE for a wide range of toolsets provided by STMicroelectronics. ST200-specific plug-ins customize STWorkbench to support the full development cycle of ST200 target applications, including:

● program editing

● project workspace management

● program building

● program debugging

● advanced trace and profiling tools

See the *STWorkbench data brief* (8063672) for further information.

1.1 Toolset features

The ST200 Micro Toolset provides an integrated set of tools to support the development of embedded applications.

● Code development tools (assembler, linker and compiler)
Program development is supported by the GCC compatible C/C++ optimizing compilers, assembler, linker and archiver (librarian) tools. Assembler code can be inserted in-line in C/C++ programs. The compiler and linker support mapping of application code and data objects to target memory.

● A C/C++ run-time system
The **newlib** C library provides ANSI C/C++ run-time functions including support for C I/O using the facilities of the host system. The C++ run-time system is provided by the GNU GCC **libstdc++** library which includes support for the STL and **iostream** ISO C++ standard libraries.

● Cross development with GDB
The GNU debugger (GDB) supports both the ST200 simulator and the hardware development boards. GDB also includes a text user interface and the Insight GUI as a graphical user interface on all supported host platforms. The **st200xrun** tool is also available to provide a command-line driven interface to simplify downloading and running applications on the ST200-based targets.

● OS21 real-time kernel
The OS21 real-time kernel supports the design of embedded systems. OS21 facilitates the decomposition of a design into a collection of communicating tasks and interrupt handlers.

● Trace and statistical data analysis tools
The toolset supports tracing of OS21 kernel activity and OS21 API function calls. The user may control OS21 trace either by using GDB commands or function calls embedded in the application. Trace and other statistical information can be viewed graphically in STWorkbench.

● Profiler support
Performance data can be obtained when running an application on an ST200 simulator and used to generate statistical and trace information. Performance data can also be acquired from an application running on a target board connected to an ST Micro Connect. The data can be analyzed using STWorkbench.

● ST200 simulator
This provides an accurate software simulation of the ST200 core and has three modes:

  – Reference (or cycle accurate) mode, which simulates the core in detail, including pipelines and memory subsystem; this mode runs the slowest but is the most accurate

  – ISS (or instruction set accurate) mode, which simulates the core accurately, modelling the caches but ignoring details such as pipelines

  – Fast (functional) mode, which uses a minimal set of components to run the code; this mode runs the fastest but has the lowest accuracy and does not model details such as the caches, memory subsystem or external interrupts

● Creation of Flash ROM bootstrap code and the building of application ROM files is supported by the tools. The bootstrap supports run-from-ROM and run-from-RAM operation.

● Host-target interface support
The ST200 Micro Toolset supports host-target connection using the
ST Micro Connect 2[b], which can be ordered separately. The ST Micro Connect 2 supports:

  – Ethernet and USB connection of a target system to a PC (running Windows or Linux)

  – Connection to a target development board's JTAG connector enabling the host software to start up the target board, download programs and debug them in the target

  – Use of ST TargetPacks to configure a target and bring up target hardware (ST TargetPacks for the current evaluation boards containing ST200 parts are supplied with the ST Micro Connection Package, or are available separately from STMicroelectronics)

For more information, see the *ST Micro Connect 2 data brief* (8161370).

---

b. The previous product in this family of devices, the ST Micro Connect 1 is still supported but can no longer be ordered.

1.2 OS21 real-time operating system features

OS21 is a real-time operating system, available for a range of cores including the ST200. It provides an embedded application with the features listed below.

● OS21 is multitasking. The OS21 kernel performs real-time scheduling of all the tasks in the system, based upon the relative priority of each task. This means that an embedded system can be designed as a collection of communicating tasks and interrupt handlers.

● Memory blocks can be allocated and freed from memory partitions for dynamic memory management. OS21 supports three pre-defined types of memory partition (**heap**, **fixed**, and **simple)** in order to use limited memory resources efficiently. User-defined memory partitions are also permitted. The standard C library functions (such as `malloc()`) are available.

● Inter-task communication is undertaken by means of message queues. Tasks can send and receive messages through a simple API.

● Semaphores, mutexes and event flags provide mechanisms for mutual exclusion and access control to a shared resource.

● OS21 provides several clock functions to read the current time; to pause the execution of a task until a specified time; and to time-out an input communication. OS21 uses a 64-bit signed integer to record time as a given number of clock ticks. Time calculations can be carried out using simple arithmetic.

● The interrupt API provides a mechanism for external events (such as peripherals) to communicate directly with the CPU. Whenever an interrupt is asserted, the CPU stops executing the current task and executes the interrupt handler for that interrupt. When the interrupt handler has completed, the CPU resumes execution of the interrupted task.

● The exception API processes unexpected events that occur during the execution of an instruction by passing control to an exception handler. If the exception is fatal, the exception handler outputs an informative message before terminating the process. For non-fatal exceptions, the exception handler may perform remedial actions before passing control back to the task that was running when the exception occurred. It is also possible to add user-defined exception handlers.

● The callback API enables user-supplied hook routines to be called whenever a given OS21 event occurs.

● OS21 can access devices and memory over a bus using virtual addresses that map to physical addresses in the device or memory.

● OS21 provides a comprehensive set of functions for handling the CPUs instruction and data caches. If a level 2 cache is available, OS21 can (optionally) drive it. OS21 maintains the state of the level 2 cache from within the cache management API.

● OS21 provides a support framework for power management. Several different power levels are defined, together with a mechanism for switching between levels. There is a mechanism for operations such as RAM power management and wake-up interrupt validation.

● Board Support Packages (BSPs) are available for all supported platforms, both as pre-built binaries and as source files. The BSPs declare board-specific data (such as a description of the interrupt system for the board) and configuration options, and thereby allow OS21 to be customized for any valid board, chip and CPU combination.

OS21 is distributed in the form of source code, ready to be compiled for the target SoC.

1.2.1 OSPlus

OSPlus (available separately) is an enhancement package for OS21 that provides a number of additional APIs for enabling device drivers (such as modems, printers, and removable storage, for example) and I/O infrastructure (including FAT and Ext2 file systems).

OSPlus extensions are available to provide such additional features as a TCP/IP stack and a USB stack. The extension packages require the OSPlus base package to be installed as a prerequisite.

See the *OSPlus data brief* (7813502) for more information.

2 Toolset overview

This chapter describes the contents of the ST200 Micro Toolset.

2.1 Software tools

The ST200 software tools include:

From the GNU binutils package

st200as	GNU assembler
st200ld	GNU linker
st200addr2line	Convert addresses into file names and line numbers
st200ar	Create, modify, and extract from archives
st200elfedit	Edit the header of the ELF format files
st200c++filt	Demangle encoded C++ symbols
st200gprof	GNU profiler
st200nm	List symbols from object files
st200objcopy	Copy and translate object files
st200objdump	Display information from object files
st200ranlib	Generate index to archive contents
st200readelf	Display the contents of ELF format files
st200size	List file section sizes and total size
st200strings	List printable strings from files
st200strip	Discard symbols
binopt	Binary optimizer

From the GNU make package

mingw32-make	GNU make (only on MS Windows)

From the GCC GNU package

gcov	GNU test coverage tool

From the GDB/Insight GNU package

st200gdb	GNU target debugger
st200gdbtui	Text User Interface for the debugger
st200insight	Graphical User Interface for the debugger

Others

st200c++	GCC compatible optimizing C++ compiler
st200cc	GCC compatible optimizing C compiler
st200xrun	ST200 target loader
st200rebase	Change the memory layout for an ELF executable file
st200rltool	Relocatable library tool (implemented as a Perl script)
st200version	Display of the ST200 Micro Toolset version
st200symbolise	Augment the simulator STISS trace information
os21decodetrace	Decodes OS21 Trace output files
os21prof	OS21 profiler (implemented as a Perl script)
os21usertrace	User trace tool for OS21 Trace (implemented as a Perl script)
os21usertracegen	Tool to generate definition files for **os21usertrace**.

2.2 Software libraries

The toolset supplies libraries for each of the possible target configurations supported by **st200cc**. There is one version for each permutation of the ST200 specific compiler options that affect code generation and for the Application Binary Interface (ABI), such as floating-point and endianness. Therefore, for whatever permutation of target configurations a user program is compiled, the compiler driver automatically selects a library with the same permutation (except for optimizations).

Compiler run-time libraries

The newlib package supplies an ISO/ANSI C run-time library (**libc** and **libm**) and header files. The run-time libraries also provide support for low-level I/O and additional math functions. The Data Transfer Format library (**libdtf**) implements the low-level I/O. A run-time library (**libgprof**) is also provided to support profiling with **st200gprof**.

An ISO/ANSI C++ run-time library (**libstdc++**) and header files supporting I/O streams and the standard templates library (the STL).

Compiler support libraries

Compiler intrinsic libraries (**libgcc** and variants) and a run-time library **libgcov** to support code coverage with **st200gcov** are also provided.

Others

Other libraries include:

- the OS21 real-time kernel library and header files, and OS21 board support libraries for the various supported platforms
- the relocatable loader library and header files
- the zlib compression library and header files

2.3   Documentation

The documentation set for the ST200 Micro Toolset includes:

● *ST200 Micro Toolset User Manual* (8063762)

● *ST200 Micro Toolset Compiler Manual* (7508723)

● *ST200 Run-time Architecture Manual* (7521848)

● *ST200 ELF Specification* (7932400)

● *ST231 Core and Instruction Set Architecture Manual* (7645929)

● *OS21 User Manual* (7358306)

● *OS21 for ST200 User Manual* (7410372)

● Application notes applicable to the release

2.4   Source and example files

The installation delivers the following sets of files:

● the source for the OS21 real-time kernel library

● various example applications, including those using OS21 and showing the construction of Flash ROM systems

2.5   Distribution media

The release is distributed through the STMicroelectronics ftp site (ftp.st.com). STMicroelectronics supports its products worldwide through Sales Offices and authorized distributors.

Also available for download from the ftp site are:

● STWorkbench for ST200 Micro Toolset

● ST Micro Connection package

● the combined source package containing the open-source components of the ST200 Micro Toolset

● OSPlus base package

2.6   Operating requirements

The ST200 Micro Toolset has the following operating requirements.

● The PC package can run on an Intel-compatible PC running Windows XP or Windows 7.

● The Linux package can run on an Intel-compatible PC running Linux version Red Hat Enterprise Workstation 4 or 5.

2.7 System Interface

The toolset operates in conjunction with the host-target interfaces listed in *Table 1*.

Table 1. Host-target interfaces

Description	Order number
ST Micro Connect 1	None[1]
ST Micro Connect 2 with STMC I/O convertor Type A to support a target debug connector type: TTL IDC 20-way JTAG.	STMC2-40/200

1. The ST Micro Connect 1 is a legacy product that can no longer be ordered.

For more information about the ST Micro Connect 2, see the *ST Micro Connect 2 data brief* (8161370).

Acknowledgements

The ST200 series cores are based on technology jointly developed by Hewlett-Packard Laboratories and STMicroelectronics.

Microsoft$^®$ and Windows$^®$ are registered trademarks of Microsoft Corporation in the United States and/or other countries.

Linux$^®$ is a registered trademark of Linus Torvalds.

Red Hat$^®$ and Enterprise Linux$^®$ are registered trademarks of Red Hat, Inc.

Revision history

Table 2. Document revision history

Date	Revision	Changes
10-May-2012	5	Add information relating to OS21 that was previously in a separate data brief. Remove references to ST240 core which is no longer supported in toolset releases later than ST200 Micro Toolset R7.1.
26-Jan-2011	D	Updated to reflect the ST200 Micro Toolset R7.1. Main changes are: – supported hosts – addition of **os21usertrace** and **os21usertracegen** – order numbers for the STMC2
5-Jan-2009	C	Changes to formatting to align the document with other data briefs.
30-Jul-2008	B	Minor correction.
08-Jul-2008	A	Initial release.

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 TWO AUTHORIZED ST REPRESENTATIVES, 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.

STMicroelectronics group of companies

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

www.st.com