



SOFTWARE DEVELOPMENT TOOLS FOR ST6 MCU FAMILY

COMPLETE SOFTWARE DEVELOPMENT SUPPORT

The AST6 Macro-assembler, LST6 linker and WGDB Windows GNU Debugger support the whole range of ST6 microcontrollers including all the different variations.

MAIN FEATURES

- Macro-Assembler (AST6)
 - Support for Nested Macros
 - Conditional Assembly
 - Support for Paged Data Space
 - Support for Data ROM window
 - Generates Intel .hex format executable files, listing files or object files.
- Linker (LST6)
 - Links up to 32 modules
 - Extensive symbol manipulation
 - 33 sections (including interrupt vectors)
 - Error diagnostics
 - Generates Mapping Information, Cross reference Tables, Symbol Tables
 - ROM masking option
 - Generates Intel .hex format executable files
 - Optional generation of symbol information for source level debugger and emulator
- Windows GNU Debugger (WGDB6)
 - Source Level Debugging
 - Supports GNU SDB commands
 - Runs under Windows 3.x, Windows 95 and Windows NT
 - Works with ST6 Simulator, ST6 Starter kits or ST6 HDS Emulators
 - Software breakpoints definable on source or disassembled code.
 - Memory breakpoints triggered on access to a pre-defined area of memory and optionally when a hardware-related condition is met.
 - Line-by-line source code and machine code execution. Function calls can optionally be considered as a single instruction, depending

on the tracing level.

- Keep a trace of memory access during program execution.
- View the stack contents.
- View and modify the simulated/emulated ST6 memory and register contents.
- View and modify data symbol values in real time, while the program is running.
- Automatically execution of GDB6 command batch files, with or without the WGDB6 graphical interface, at start-up.

MACRO-ASSEMBLER (AST6)

The ST6 Macro assembler accepts a source file written in ST6 assembly language and transforms it into an executable file in relocatable object code format. When the whole program is in one file only, the assembler also generates a hexadecimal file (INTEL hex format) ready to be programmed into an EPROM device.

The assembler recognizes the use of section, symbols, macros and conditional assembly directives. In addition, the ST6 Assembler is able to produce detailed assembly listing and symbol cross reference file.

LINKER (LST6)

The ST6 Linker is responsible for combining a number of object files into a single program, associating an absolute address to each section of code, and resolving any external references.

The ST6 Linker produces a hexadecimal file in INTEL format to be downloaded into an EPROM device and an object code file to be used with the simulator. The linker also produces a map file which gives information about the sections, pages, modules and labels. Finally, listing files are produced which update the assembler listings with real addresses of symbols and statements.

This software program allows the user to develop modular programs, which may then be combined and addressed as defined by the user. The flexibility of the ST6 Linker is greatly increased by the use of sections allowing the user to group pieces of software from different modules. The location and the size of each section is user selectable.

WINDOWS GNU DEBUGGER (WGDB6)

WGDB6 is a debugger for programs developed using either the ST6 Macro-Assembler (AST6) or the ST6 C Compiler. It uses the GNU Debugger (GDB) command set, with additional ST6-specific commands, allowing execution of any of these commands by choosing the appropriate item from the standard Windows interface.

CONFIGURATION

WGDB6 can be used either with the ST6 Simulator, the ST6 Starter Kit or the ST6 HDS Emulator.

- ST6 Simulator is a program that simulates the execution of ST6 programs. It includes Wave Form Editor, that enables the simulation of ST6 pin input and output.
- ST6 Starter Kits include the simulator and a board that simulates the ST6 pin input and output.
- ST6 HDS Emulators are hardware devices that enable real-time execution of ST6 applications. The emulator can be connected to your own board, offering the same debugging capacities as the simulator.

GENERAL DESCRIPTION

WGDB6 enables the user to execute ST6 programs, and view the contents of the ST6 data and program memory as the program progresses. You can examine source code, as you would with a C-language source program, and assembler code. Program execution history can be viewed at source or instruction level using the Trace and Stack displays. WGDB6 lets you read and write all ST6 registers and memory locations.

DATA DISPLAY

WGDB6 can display data in either 'normal', 'hot' or 'real-time'. Normal display data is displayed as it was when you chose to view it, it is not automatically updated. Normal display data is displayed on a white background. Hot display data is updated every time the execution of the program you are debugging is suspended, and is displayed on a yellow background. Real-time display data is updated as the program is running, and is displayed on a red background.

Figure 1. Using WGDB6 with ST Simulator

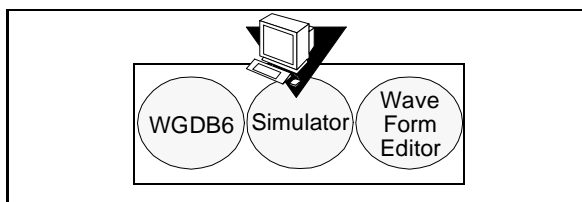


Figure 2. Using WGDB6 with the ST6 Starter Kit

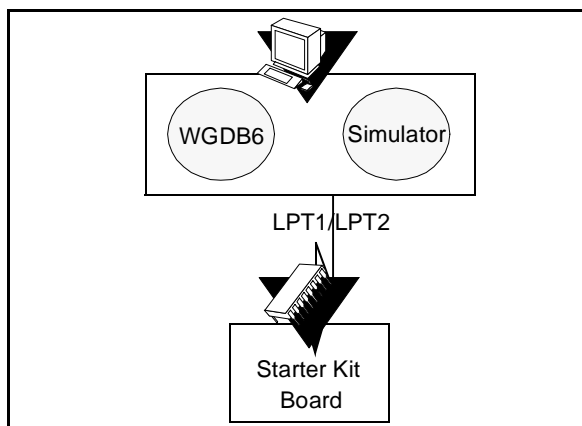
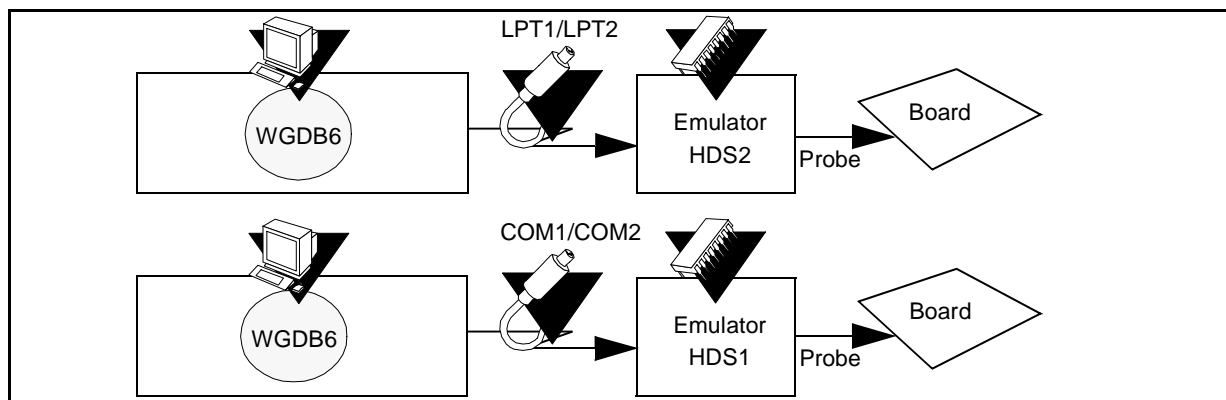


Figure 3. Using WGDB6 with the ST6 HDS Emulator



WORKSPACES

WGDB6 enables you to save and load workspaces. Workspaces are snapshots of windows and option choices that are taken when you close a program. Each program you debug using WGDB6 has its own default workspace definition. When

you load a program, the workspace that you were using when you last closed it is restored, thus you can continue working from where you left off. You can also save workspace definitions at any time, so that you can restore them at a later date.

Figure 4. Development Flow Chart.

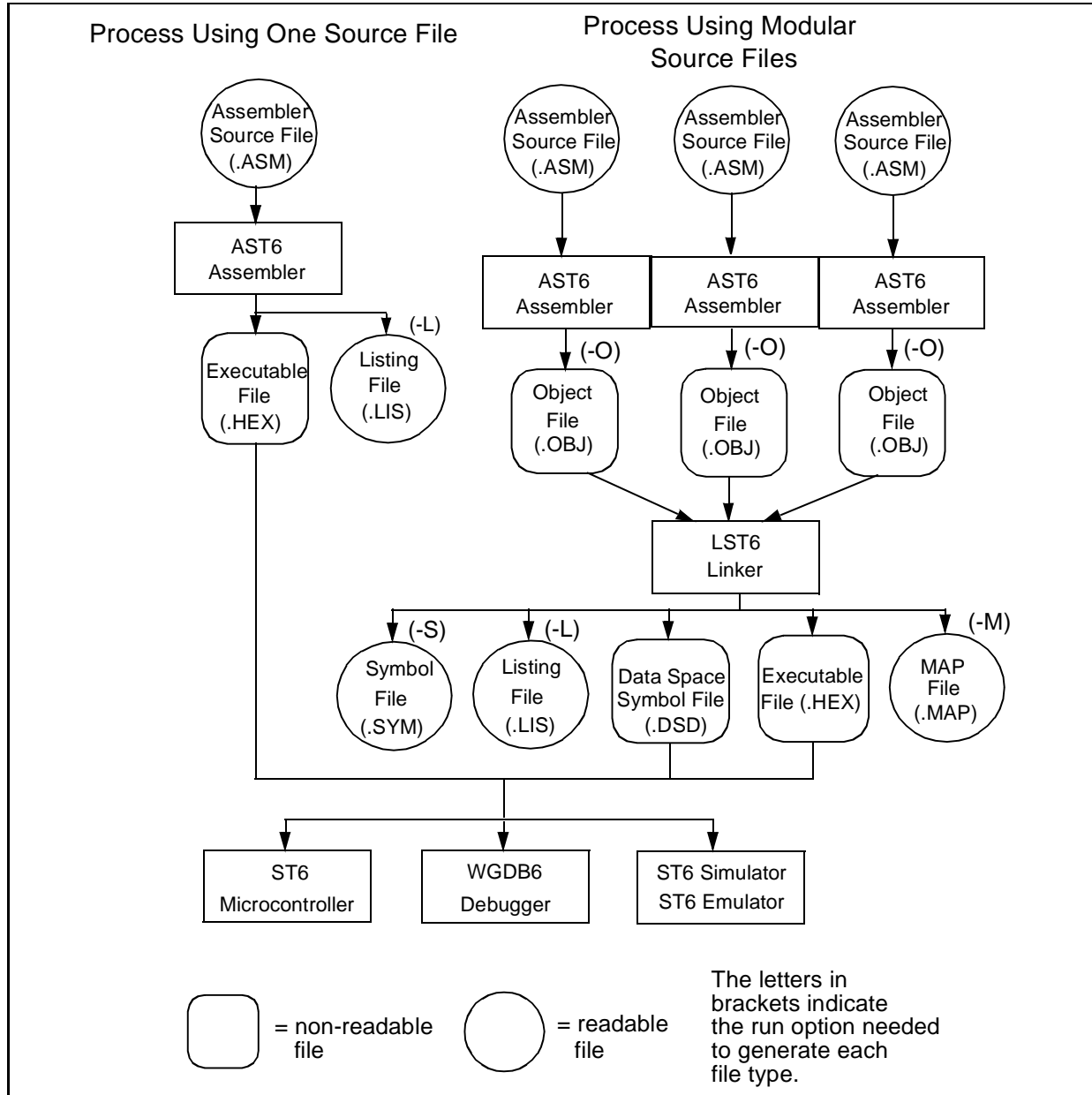


Table 1. AST6 Directives

| | |
|-------------|---|
| .ASCII | Stores in program space a string as a sequence of ASCII codes |
| .ASCIZ | Same as.ASCII followed by a null character |
| .BLOCK | Reserves a block of contiguous memory location |
| .BYTE | Stores successive bytes of data in program space |
| .DEF | Defines the characteristics of a data space location |
| .DISPLAY | Displays a string during assembly process |
| .DP_ON | Segments the data space |
| .EJECT | Starts a new listing page |
| .ELSE | Beginning of the alternative part in conditional assembly block |
| .END | End of source file |
| .ENDC | End of conditional assembly block |
| .ENDM | End of a macro definition |
| .EQU | Assigns the value of an expression to a label |
| .ERROR | User defined assembly error |
| .EXTERN | Defines a symbol as external |
| .IFC | Beginning of conditional assembly block |
| .INPUT | Includes an additional source file in the present one |
| .GLOBAL | Defines a symbol as global |
| .LABEL.W | Initializes Data ROM Window Register |
| .LABEL.D | Gains access to a label in a Data ROM Window |
| .LINESIZE | Set listing line length |
| .LIST | Enables the listing of specified fields of the source file |
| .MACRO | Beginning of a macro definition |
| .MEXIT | End of a macro expansion |
| .NOTRANSMIT | Inhibits symbol transmission to the linker |
| .ORG | Set current location counter |
| .PAGE_D | Specifies the page number in data space |
| .PL | Set listing page length |
| .PP_ON | Segments the program space in 2K pages |
| .ROMSIZE | Defines the available ROM size |
| .SECTION | Provides a logical partitioning of program space |
| .SET | Same as.EQU, but can be redefined in the source file |
| .TITLE | Assigns title to the document |
| .TRANSMIT | Transmits symbol definitions to the linker |
| .VERS | Defines the target ST6 device |
| .WARNING | User defined assembly warning |
| .WINDOW | Defines a continuous relocatable block of program code |
| .W_ON | Enables the use of the.WINDOW directive |
| .WORD | Stores successive words of data in program space |

ORDERING INFORMATION

| Sales Type | Description |
|------------|--|
| ST6-SW/PC | ST6 software development tools (includes AST6, LST6 and WGDB6) |

Note: The ST6 software package is included in all ST6xxx-EMU real time development tools.

Notes

Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without the express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics

©1998 STMicroelectronics - All Rights Reserved.

Purchase of I²C Components by STMicroelectronics conveys a license under the Philips I²C Patent. Rights to use these components in an I²C system is granted provided that the system conforms to the I²C Standard Specification as defined by Philips.

STMicroelectronics Group of Companies

Australia - Brazil - Canada - China - France - Germany - Italy - Japan - Korea - Malaysia - Malta - Mexico - Morocco - The Netherlands - Singapore - Spain - Sweden - Switzerland - Taiwan - Thailand - United Kingdom - U.S.A.

<http://www.st.com>