SV TEHS SIA

Scenix Development Tools

SX-DEV

User's Guide

SV TEHS SIA

SX-DEV User's Guide

Ruses 14-24 • LV1029 • Riga • Latvia
Phone: +371-9237495 +371-9223895 • Fax: +371-7543384
Email: **info@mail.svtehs.com** • Web: **http://www.svtehs.com**

Chapter 1

Introduction

SX-DEV development system offers friendly an easy to use complete development environment for Scenix microcontrollers.

The SX-DEV development system supports all currently available SX microcontrollers from Scenix Semiconductor, include SX18/20/28AC and SX48/52BD. It uses the SX chip's built-in debugging capabilities to provide a low-cost and efficient real-time emulation solution. It supports real-time debugging for all SX chips at any user-defined speed up to 100 MHz. It allows users to run the target application in real time, step-by-step or in animate mode, debug source codes and program a target SX device when design is complete. It connects to the serial port of any PC and comes with complete integrated development environment.

Debugging and programming with SX-DEV is provided via a 6-pin emulation connector which mates with a 6-pin 0.1-inch square post header in the target system. The connector is wired for OSC1, OSC2, VDD, VSS and Reset pins, another one pin used as key. The oscillator circuit components on the target board should not be disconnected as long as the components are within recommended specifications.

Features

- Real time, full speed execution
- Serial port interface
- Powered from the target device, less than 5 mA
- Voltage range from 3.0V till 5.6 V
- Source level support includes SASM, MPASM, Bytecraft SXC, Hitech PIC C and more
- Built-in in-system programmer
- User-selectable frequency source (quartz, generator, resonator, external source) up to 100 MHz
- Step, Run, Run to Cursor, Animate and Break functions
- Writes to all registers for code testing

- One level software breakpoint
- Unlimited watch variables
- Full-featured debugging driver with all functions support, including timer blocks

Package

SX-DEV package include the following:

- SX-TIPS in-circuit debugger/programmer
- Real-time module
- 50 MHz quartz crystal
- SX-DEV software for Windows 95/98/NT
- SX-DEV User's Guide

Requirements

It is recommended to install SX-DEV Development System on the system with the following minimum requirements:

- IBM compatible 486/Pentium.computer
- One free serial port with 9-pin connector
- 4 MB Ram, 16 MB recommended for the operational system
- 4 MB of free hard disk space
- Microsoft or compatible mouse

About the User's Guide

This User's Guide organized as following:

- **Chapter 1: Introduction** – Summarizes the SX-DEV Development System features and requirements.
- **Chapter 2: SX-DEV Hardware** – Describes the SX-DEV hardware including installation, real-time module, oscillator considerations, connection headers, etc.

- **Chapter 3: Software Installation** – Provides information on installing the Windows 95/98/NT software and SX-TIPS hardware.
- **Chapter 4: Project Constructor** – Summarizes all project parameters and external software products configuration.
- **Chapter 5: Editor** – Describes different built-in editor functions.
- **Chapter 6: Debugger** – Summarizes the debugging functions including breakpoints, watches, snapshots, etc.
- **Chapter 7: Watch Constructor** – Provides detailed description of the powerful watch constructor feature and different watch options.
- **Chapter 8: Programmer** – Describes built-in device programmer.

Software Updates

New versions of the SX-DEV software can be obtained from the manufacturer's web site at:

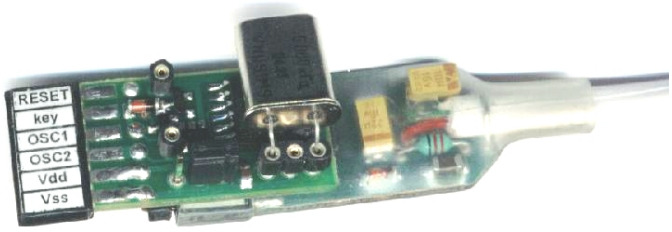
http://www.svtehs.com/sxdev.htm

Chapter 2

The SX-DEV hardware

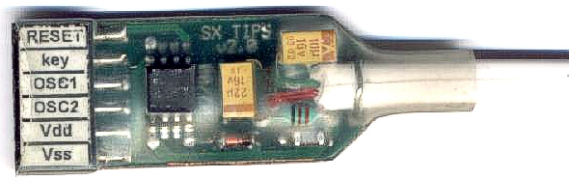
SX-DEV small and easy to use hardware powered from the target board (<5 mA) and able to work at any user-defined speed up to 100 MHz with the realtime module...

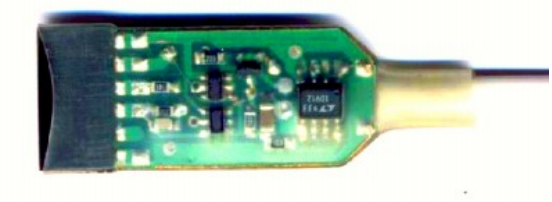
Two main components of SX-DEV hardware described here – SX-TIPS debugger/programmer and realtime module. For real-time debug modules should be connected together, as on the photo. Both modules powered from the target board and consume less than 5 mA (generator current not included). Both modules can work from 3.0 till 5.6 V and allow low-voltage debugging.



SX-TIPS debugger/programmer

SX-TIPS debugger/programmer connected to the free 9-pin PC serial port with standard DE9 connector, attached to the flat 1.5 m cable. It connected to the target board via a 6-pin emulation connector which mates with 6-pin 0.1-inch square post header in the target system. To implement in-circuit programming and debugging features, this connector should be included in target board. It should have 6 pins, attached to the following target microcontroller pins:





1 – Vss, ground

2 – Vdd, power pin, 3.0…5.6V

3 – OSC2, passive components on target board allowed

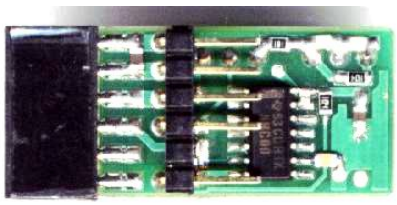
4 – OSC1, passive component on target board allowed

5 – key to prevent wrong connection, no pin on the header

6 – Reset, optional for external reset connection, not used by realtime module

SX-TIPS can also be attached to 4-pin header, compatible with Parallax and TransData in-circuit programmers and emulators. Only pins 1-4 should be used in this case.

Real-time module



Real-time module connected between SX_TIPS debugger/ programmer and target board and powered from the target board. Four different clock sources can be used – DIP 14-pin plastic or metal package oscillator; HC-49/U case or compatible crystal; ZTT/ZTA series or compatible ceramic oscillator or external clock source. Please ensure, that jumper on the top of the real-time module board installed according to the used clock source. Take the board with jumper on top, 6-pin connector on the left (as on the first photo). Than check/change jumper position, as described below:

DIP 14-pin oscillator	jumper on the left two pins
External clock	jumper on the left two pins
HC-49/U crystal	jumper on the right two pins
ZTT/ZTA ceramic oscillator	jumper on the right two pins

DIP 14-pin **oscillator** should be installed with the first pin on the right top corner of the board (as on the photo). Any frequency up to 100 MHz can be used (module can work till 120…150 MHz, however current SX chips not rated to this frequencies). Please take into account oscillator supply current - typical high-frequency oscillators usually draw 30…60 mA, this current will be drawn from the target board.

External **clock** signal should be supplied with short wire on the left bottom receptacle of the DIP14 socket or to the middle pin of the 3-pin jumper header. Do not forget connect also ground wire (for example, to the right bottom receptacle), if ground of the external clock source is not the same, as at the target board ground. CMOS output for the external clock signal recommended, any frequency up to 100 MHz allowed.

HC-49/U or compatible **crystal** should be installed to the two receptacles on the right bottom of the real-time module. Only fundamental crystals can be used. On-board generator optimized for high-frequency crystals – most fundamental microprocessor-grade crystals from 1 MHz till 100 MHz can be used. Generator supply current depended from the crystal frequency and for most crystals will be in range 2…30 mA.

ZTT/ZTA or compatible **ceramic** oscillator can also be used. Both 2-pin and 3-pin (with built-in capacitors) ceramic oscillators works well with on-board generator in range 1…50 MHz. Ceramic oscillator should be installed to the two receptacles on the right bottom of the real-time module. Generator supply current depended from the crystal frequency and for most ceramic oscillators will be in range 2…30 mA.

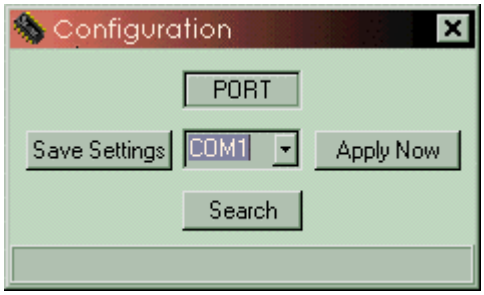
Chapter 3

Software Installation

SX-DEV software installation is very simple.

Software installation procedure is very simple – just copy **sxdevs.zip** archive to the separate directory on your hard drive and unpack it, using Winzip or Zip program. Run **sx_devs.exe** and installation procedure finished. During the first start, two new files will be created – sx_devs.ini and sx_tips.ini. First file store all SX-DEV environment configuration, second file store built-in SX-TIPS programmer configuration.

Configuration



After SX-DEV software installation you need to configure serial port used for SX-TIPS debugger/programmer connection. Ensure, that you have free serial port and no any other software use this port. Connect SX-TIPS debugger/programmer to the target board and apply power to the target board, as the SX-TIPS powered from the target board. Connect SX-TIPS to the free serial port with DE9 connector on the SX-TIPS cable end. Run **sx_devs.exe** software and choose Configure/Tips_Hardware option from the menu. You will get configuration menu. If you did not have other serial ports currently in use on your computer, you can use **Search** option – this option will automatically search SX-TIPS hardware on all available ports, however some conflicts may occur during the search procedure. If you know exactly, what port will be used for SX-TIPS hardware, choose the right port manually and **Save Settings**. When connection with SX-TIPS hardware established, you will get message "SX_TIPS Hardware found on COMx" at the bottom line of the Configuration window.

Chapter 4

Project Constructor

All necessary project parameters, used components and external software described in project constructor window.

New project starts with project constructor window. You can choose list of parameters, described your project. It is recommended to create new directory for each project.

1) Name your project (same name, as directory are OK) and indicate full path to the directory, where project will be stored in ProjectFile box.

2) Choose your source file. You can get some source file with extension ***.asm or *.src** in any directory on your disk and copy it to the project directory here or you can create new file. Filename in SourceFile field should not have a path and file should be placed in project directory.

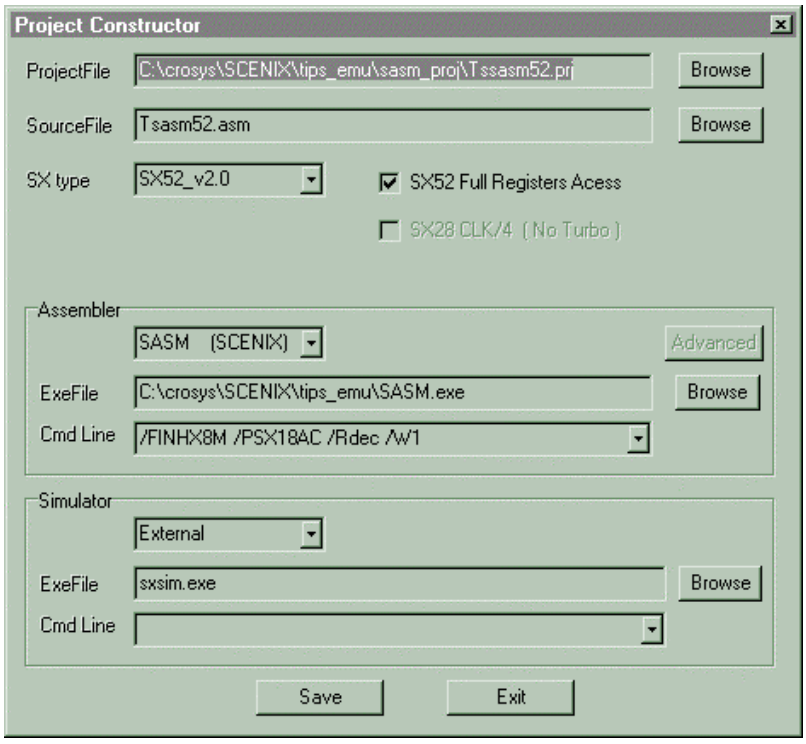
3) Choose the target microcontroller type. To define, what is the correct microcontroller hardware revision, check the marking and date code.

SX28	v2.5	FEE	9815, 9825, 9829, 9830, 9837, 9838, 9841, 9843, 9844, 9848, 9908, 9911, 9913

This and earlier versions of SX18/20/28 did not supported with SX-DEV, all this chips should be replaced to newer silicon revisions.

SX28	v4.1	FDE	9849, 9850, 9910A4, 9912B4
	v5.2	FCE	All newer date codes.
SX52	v1.0	001	
	v2.1	002	

4) Choose the assembler used. SX-DEV supports two external assemblers – SASM from Scenix Semiconductors and MPASM from Microchip Technology. You can download latest version of SASM from **http://www.scenix.com** and latest version of MPASM from **http://www.microchip.com**. Use additional include file sx_defc.inc with MPASM assembler, all additional SX commands defined here as macros. However, MPASM can be used only with SX18/20/28 chips, which are compatible with Microchip PIC16C5X series. You can also add necessary parameters to the assembler command line, see SASM and MPASM manuals for details.



5) External software simulator SXSIM can be used with MPASM for SX18/20/28 software simulation. Indicate the path to the software simulator and necessary command line parameters. See SXSIM manual for details.

6) SX52 Full Registers Access checkbox install full-featured debug driver into SX48/52 target microcontroller. When this checkbox switched on, following SX48/52 resources will be available for debug – BANK_0, Timer Control registers, Compare and Capture registers, WKPND_B, CMP_B, PLP_A…E, TRIS_A…E and other Mode registers. Program memory addresses 0F00…0FFF will be used for debug purposes and user program should not use this addresses.

When this checkbox switched off, only Special Function and Global registers 01…0F and bank 01…bank 0F data memory registers will be available for debug. Program memory addresses 0F7D…0FFF will be used for debug purposes and user program should not use this addresses.

7) SX28 CLK/4 (No Turbo) checkbox allow SX18/20/28 debug in PIC-compatible mode. In this mode each SX command used four OSC1 clocks, i.e. instruction rate is one-fourth of the clock rate.

To finish project creation, press Save. If all necessary fields properly filled, Project Constructor window will be closed, new project file will be created and program source window will be opened.

All project configuration stored in <name>.prj file. It has the following content:

- Source file name, watch file name (<name>.wtc)
- Dimensions and place for all project windows for Edit and Debug modes
- Used fonts
- Target SX microcontroller
- Assembler and Simulator used, paths, command line parameters

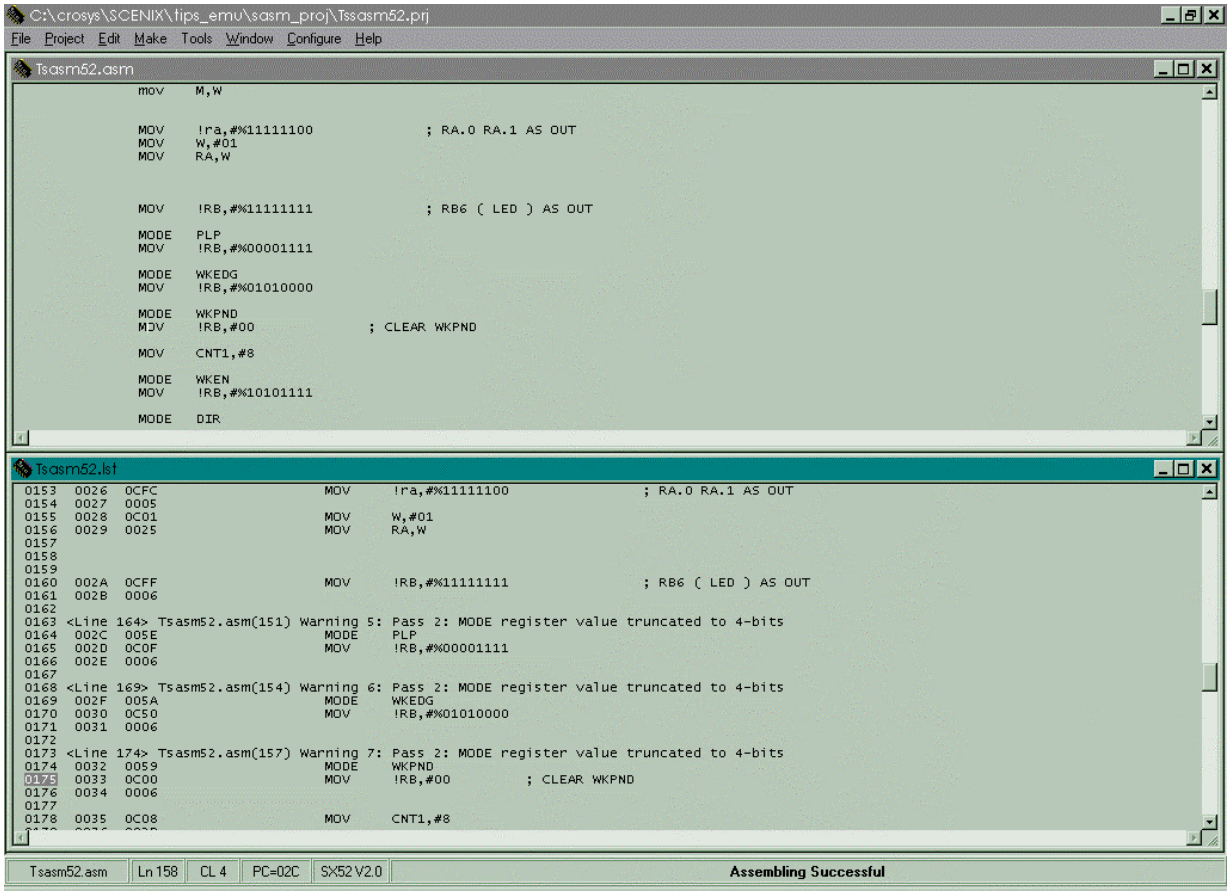
SX-DEV store in configuration file last 10 projects, quick access to this projects available in menu Project/ReOpen.

Chapter 5

Editor

Built-in simple text editor have all necessary functions for easy project writing and changing.

Built-in text editor have COPY (**ctrl+C**), PASTE (**ctrl+V**), CUT (**ctrl+X**), FIND (**ctrl+F**), REPLACE (**ctrl+R**), UNDO (**ctrl+Z**) and REDO (**ctrl+shift+Z**) functions. Undo buffer store last 100 commands, so you can easily delete and restore some text fragments. Font can be changed in Edit/Font menu with Windows font dialog.



Status bar in the bottom border divide into six fields. First field represent source file name, current cursor position line and column displayed in second and third fields, program counter value displayed in fourth field during Debug mode, SX device type set for the current project displayed in fifth field and messages displayed in the last field.

Assembling

To assemble program, choose Make/Assemble option from the drop-down menu or use (**ctrl+F9**) hotkey. Listing window will be created after assembling. To put it in the front, use Windows/Listing option or (**ctrl+L**) hotkey. If no errors detected during assembling, message "Assembling Successful" will be displayed in the status bar, otherwise error code will be displayed and error line in the source window will be colored. If more than one error or warnings detected, appropriate dialog window will be displayed.

In both Edit and Debug modes all cursor movements in source and listing windows are synchronized. When you move cursor in the source window, appropriate listing fragment selected with color and vice versa.

Chapter 6

Debugger

Debug mode allow step-by-step and realtime snapshot visualization for all program flow, register and variables content and many more…

For emulation purposes, there is no need of the special bond-out chip for Scenix microcontrollers. You do not have to worry about the potential variations in electrical characteristics of a bond-out chip and actual chip are used in the target application. Each Scenix microcontroller have built-in hardware support for real-time in-circuit emulation. This built-in extra hardware defines main features of the in-circuit emulation with Scenix microcontrollers: single hardware breakpoint or watchpoint and special debug driver located in main program memory.

Special debug driver should be located in main program memory in Debug mode. Size of this driver depended from the debug resources necessary. To start Debug mode, SX-DEV writes into target SX microcontroller user code and selected debug driver (see Chapter 4 for debug driver selection). To allow debug, user program **should not be located** on the following addresses:

SX28: 1FE…1FF, 78E…7FF

SX52: 1FE…1FF, F00…FFF or F7D…FFF, depended of debug driver selected

Start address and size of the debug drivers can be changed in future releases of SX-DEV software. In any case, you will get warning, if your user code will overlap space, used by debug driver and software will not enter into Debug mode in this case.

Enter Debug mode

To start Debug mode, choose Make/Debug Start option from the drop-down menu or use (**ctrl+H**) hotkey. The following operations will be performed:

- Source code assembled (proceed to the next step, only if assembled without errors and warnings)
- Target SX programmed with user code and debug driver
- Debug connection with target SX established
- All required target SX registers content transferred into SX-DEV software

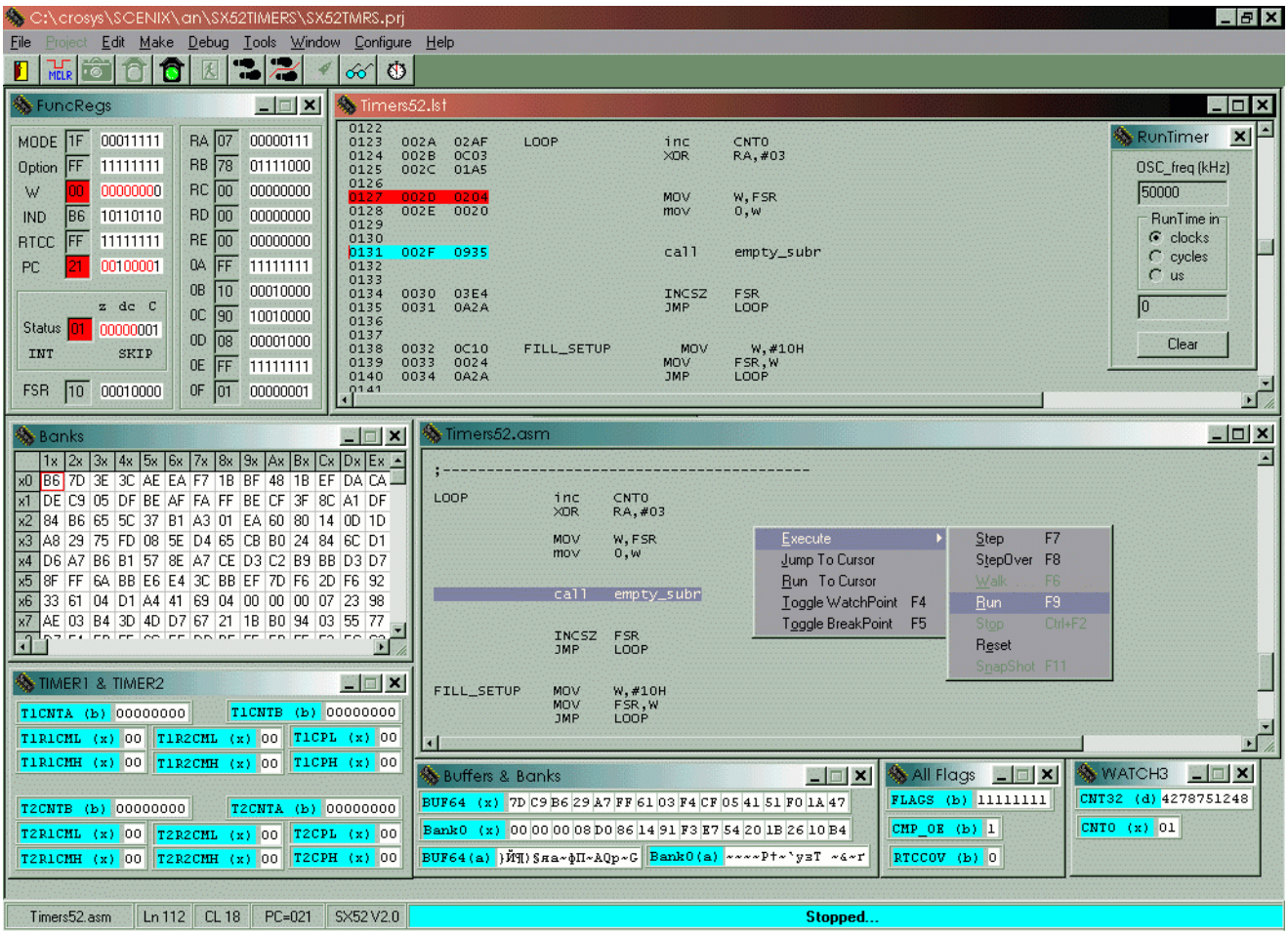
DEBUGGER

- Program counter set to Reset vector
- FuncRegs and FileRegs windows opened

SX-DEV software will be in Debug mode after finishing the operations above.

To restart Debug mode without SX reprogramming, choose Make/Debug Restart option from the drop-down menu or use (**ctrl+0**) hotkey. This option will be useful, if you close and restart SX-DEV for some reason, but did not change anything in source code.

See typical screen content in Debug mode.



Debug mode toolbar

In Debug mode additional toolbar and hot keys is available. Also in source and listing windows additional pop-up menu with frequently used functions is available on right mouse click.

 **Close Debug Session (ctrl+C).** This key will close debug connection with SX microcontroller and return to Edit mode.

Reset (F10). This key emulates hardware reset of the target SX microcontroller. During the Run or Walk process, first Reset will do the same, as Stop function, second Reset will do hardware reset.

 **Snapshot (F11).** This key allows interrupting microcontroller in Run mode for short time and updating all register data on screen. Usually this helps to understand, what currently happens in the target microcontroller and where user program executed. Please take into account, that program execution will be stopped for the following time:

- For SX18/28 – about 200 ms
- For SX48/52 – about 500 ms

Data Injection is another useful feature of the Snapshot function. When SX microcontroller are in Run mode, you can change one byte in FuncRegs or Banks window, or one variable up to 4 bytes in Watch window or ASCII up to 16 symbols. These changes will be injected into target SX, when you press Snapshot button, and data in the windows will be replaced with data loaded from target SX. You can see your changes on the screen, when you press Snapshot button next time, if target SX did not modify their values.

 **Stop (ctrl+F2).** Stop execution of Run and Walk modes.

 **Run (F9).** This key can activate different run modes, depended of Realtime module and Breakpoint/ Watchpoint settings.

If Realtime module not present or Real Time Run button switched off, SX-TIPS will clock target SX microcontroller on the fixed speed about 170 kHz. When Realtime module present and Real Time Run button switched on, target SX clock will be defined by user selectable clock source (see Chapter 2 for details).

If Breakpoint present, target SX will run on the defined speed till the Breakpoint, than user program execution will be terminated and all register windows and watches will be updated with new content.

If Watchpoint present, target SX will run on the defined speed till the Watchpoint, than user program execution will be interrupted on the same time, as in the Snapshot function execution, all register windows and watches will be updated and user program execution continues.

If the target SX program did not pass through Breakpoint or Watchpoint, commands Stop, Reset or hardware reset on the target board will help to stop target program execution.

Walk (F6). Start low-speed step-by-step target program execution with the same timing, as for the Snapshot command. During Walk process you can change any registers and variables and therefore immediately change program flow.

Step (F7). This key activates one command execution.

Step Over (F8). This key adds dynamic breakpoint on the next to the current PC command and therefore omit one command. It allows step-by-step debug without subroutines or unwanted branches execution. This function will not affect watchpoint or breakpoint already set in program.

Real Time Run. Switch on this key, when you use Realtime module and want to define target SX clock source and frequency (see Chapter 2 for details). Switch off this key, if you use only SX-TIPS debugger/ programmer without Realtime module.

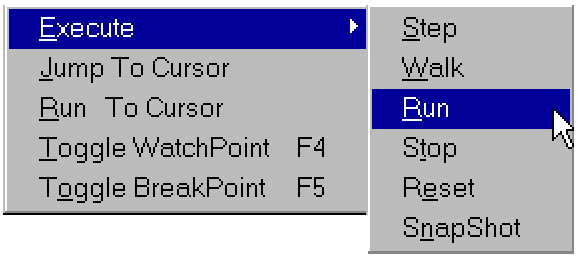
New Watch Window (ctrl+N). Open Watch Constructor window to create new Watch Window. See Chapter 7 for detailed Watch Constructor description.

Run Time Measurement. This is tool for code fragment run time counting. To measure time, Jump to Cursor and Run to Cursor commands from the pop-up menu can be used. Set target SX program counter (PC) with the command Jump to Cursor to the first command of the measured fragment. Than place cursor to the next command after the last executed command of the measured fragment and activate Run to Cursor command. Last executed command will be Breakpoint command; this command execution time did not counted.

Real time measurement can work only with Real Time Run button switched off, with low-speed clock about 170 kHz, generated by SX-TIPS debugger/programmer.

Popup window

In Debug mode popup window on the right mouse click available in the source and listing windows. Most functions are the same, as in the Debug mode toolbar. Some functions were available only in the popup window.



Jump to Cursor. This function load new value into the target SX microcontroller program counter (PC). This is address of the command, currently pointed in the source or listing window. All other registers content did not changed.

Run to Cursor. This function set dynamic breakpoint at cursor position in the source or listing window and activate Run command. Target SX microcontroller will run from its previous state till the new breakpoint. This function will not affect watchpoint or breakpoint already set in program.

Toggle Watchpoint (F4). This function set or reset watchpoint. When target SX run to the Watchpoint, user program execution will be interrupted on the same time interval, as in the Snapshot function execution; all register windows and watches will be updated and user program execution continues. Watchpoint displayed as blue color selected line only in the listing window, as the source window can have macro commands, and one line in the source window will be related to many lines in the listing window. Only one watchpoint or breakpoint can be set at one time. When new watchpoint set, previous watchpoint or breakpoint will be cleared.

Toggle Breakpoint (F5). This function set or reset Breakpoint. When target SX run to the Breakpoint, user program execution will be terminated and all register windows and watches will be updated. Breakpoint displayed as red color selected line only in the listing window, as the source window can have macro commands, and one line in the source window will be related to many lines in the listing window. Only one watchpoint or breakpoint can be set at one time. When new breakpoint set, previous watchpoint or breakpoint will be cleared.

Debug menu options

Debug menu have two additional Debug mode options.

Show BP/WP location. This option moves cursor position in the source and listing windows. Source and listing text will be positioned to show current breakpoint or watchpoint.

Clear BP/WP. This option clear breakpoint or watchpoint, if it set somewhere in the program.

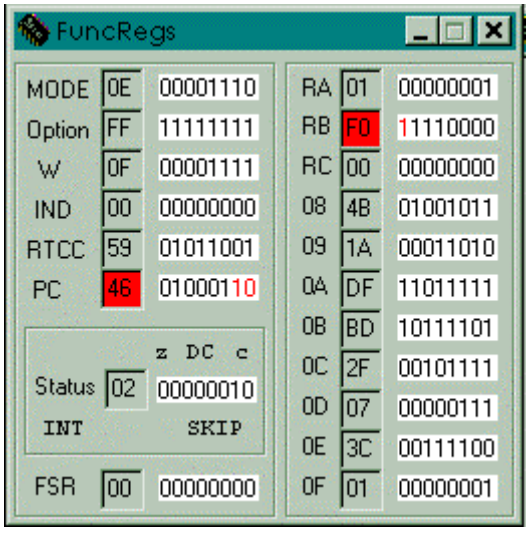
Source edit

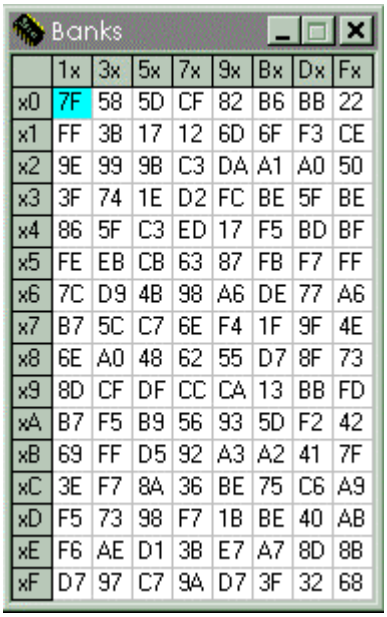
You can do minor changes in program source in Debug mode, however, if you add or delete lines in source window, synchronization between source and listing window will not work properly. Listing window will not reflect your changes in source window. When you make some changes in program source, start Debug mode once more with Make/Debug Start option from the drop-down menu or use (**ctrl+H**) hotkey.

Register windows

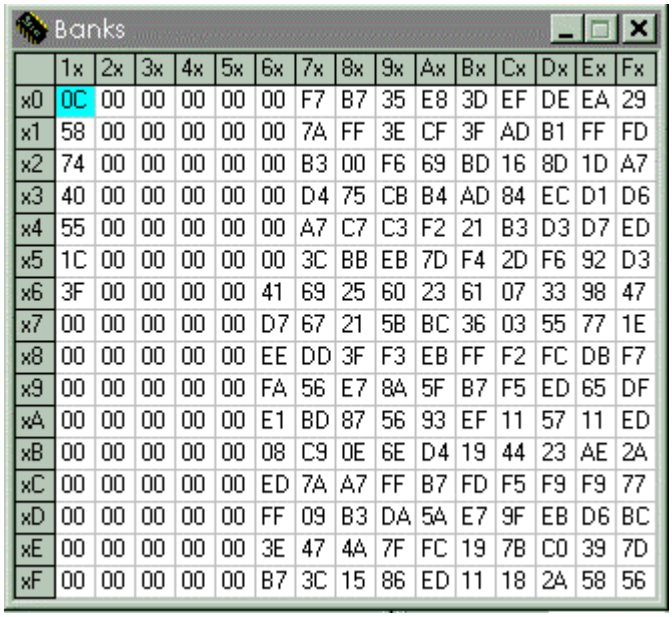
Two register windows available by default. FuncRegs window display work registers in hex and binary form. Last changes displayed as red color. Any value can be changed manually in binary or hex form with left mouse click. Status flags also allow modification with symbol names.

Banks window display data memory in hex form and also allow modification. Last changes in the data registers have different color. FSR address pointer value indicates register in color frame. Current FSR value also displayed as the red frame in the banks window. You will see exact memory register, pointed with FSR address pointer.





SX18 (20)/28 Register Window



SX48/52 Register Window

Chapter 7

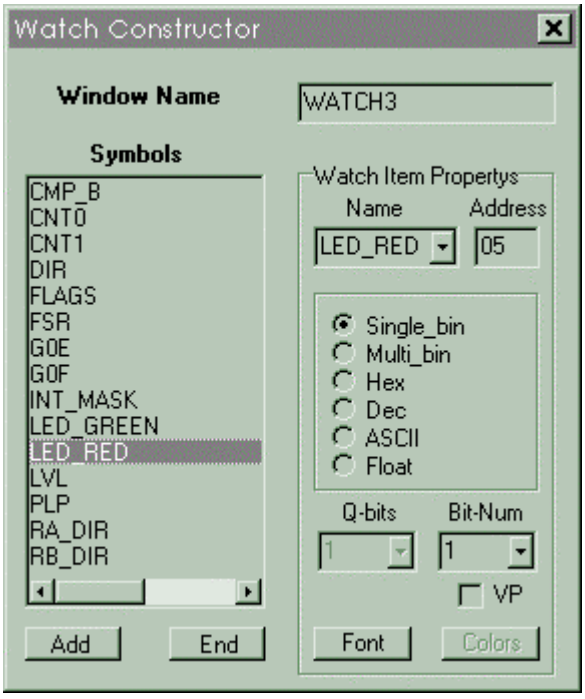
Watch Constructor

Unlimited watch variables can be created in many different views for the quick access to any program variable.

New watch window creation start with  button or (ctrl+N) hotkey. You will get Watch Constructor window.

Create watch window

There are several steps to create new watch window:



1) Assign name for new watch window in **Window Name** field (optional).

2) Choose necessary symbol name to display from the **Symbols** window. This list display all symbols, created in your program, include constants and variables. When you move on symbols list, Name and Address fields display each symbol parameters.

3) Choose proper visualization method for chosen symbol name:

- **Single_bin** – one bit, choose bit number in Bit-Num field
- **Multi_bin** – binary from 8 till 128 bit, choose variable length in Q-bits field (8, 16, 24, 32, 40, 48, 56, 64, 128)
- **Hex** – hexadecimal, from 1 to 16 bytes, choose variable length in Q-bits field (1, 2, 3, 4, 5, 6, 7, 8, 16)
- **Dec** – decimal, from 1 to 16 bytes
- **ASCII** – alphabet characters, one symbol in byte, from 1 to 16 chars

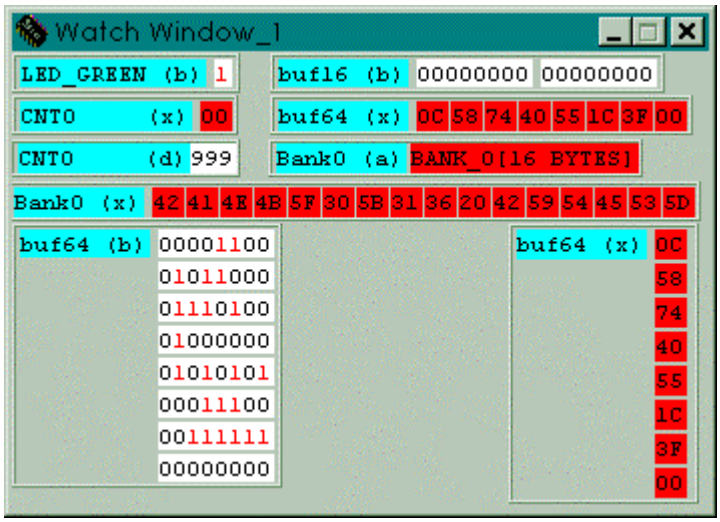
- **Float** – 24-bit float

4) For binary and Hex watches enabling **VP checkbox** will place variables vertically in watch window.

5) Optionally choose preferred **font** for new variable.

6) Double-click on the symbol name or use **Add** button to create Watch window for the chosen symbol.

7) Repeat this steps for each necessary variable and than use **End** button to finish new watch window creation.



Almost all operations in this procedure are optional: you can simply choose necessary symbol from the Symbol window, double click on it's name and use End button – this will create watch window for one byte in Hex mode with default font. You can also use popup window at right mouse click to add new variable into watch window and to end watch window creation.

Watch options

You can display not only symbols, used in your program, but also work registers and special registers with watch variables. Choose necessary register name from the **Name** field in Watch Constructor window and use the same visualization options, as with the symbols.

If for any reason some variables not displayed in Symbol window, you can add them manually, entering their name and address in the appropriate fields.

Registers update

To **update Hex** value in registers or watch window, simply enter new hex digits in the value field. Register content in the target SX microcontroller will be updated for each byte immediately after last digit entering, no need to press Enter key. If you update registers in Walk mode, do not update it faster, than Walk speed. For each Walk step only one register can be updated.

To **update binary** value in registers or watch window, use left mouse click. Register content in the target SX will be update immediately for each bit.

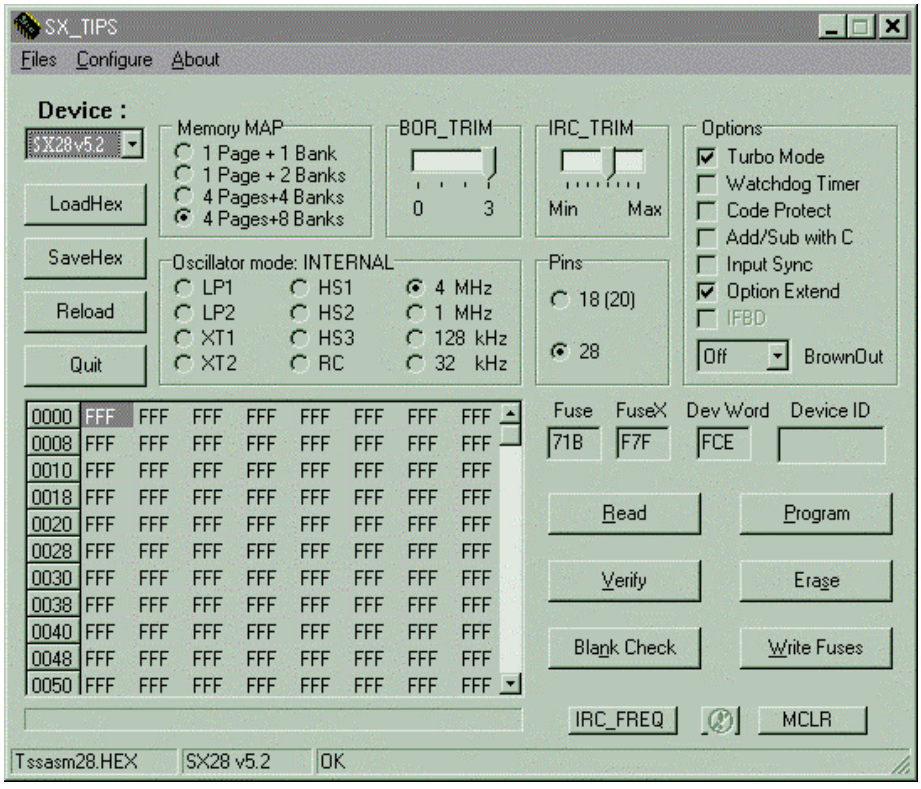
To **update decimal and ASCII** watch variables, delete previous value in the field with Del or Backspace keys, than enter new value and press Enter key. Register content in the target SX will be updated after Enter key press.

Chapter 8

Programmer

Built-in programmer described here.

Built-in programmer supports all features of SX microcontrollers. Most options can be set in source file (see SASM manual for details). SX-TIPS programmer allows all options manual change.



LoadHex button allow Hex file loading. INHX8M .hex file and Parallax .sxh Hex file formats supported.

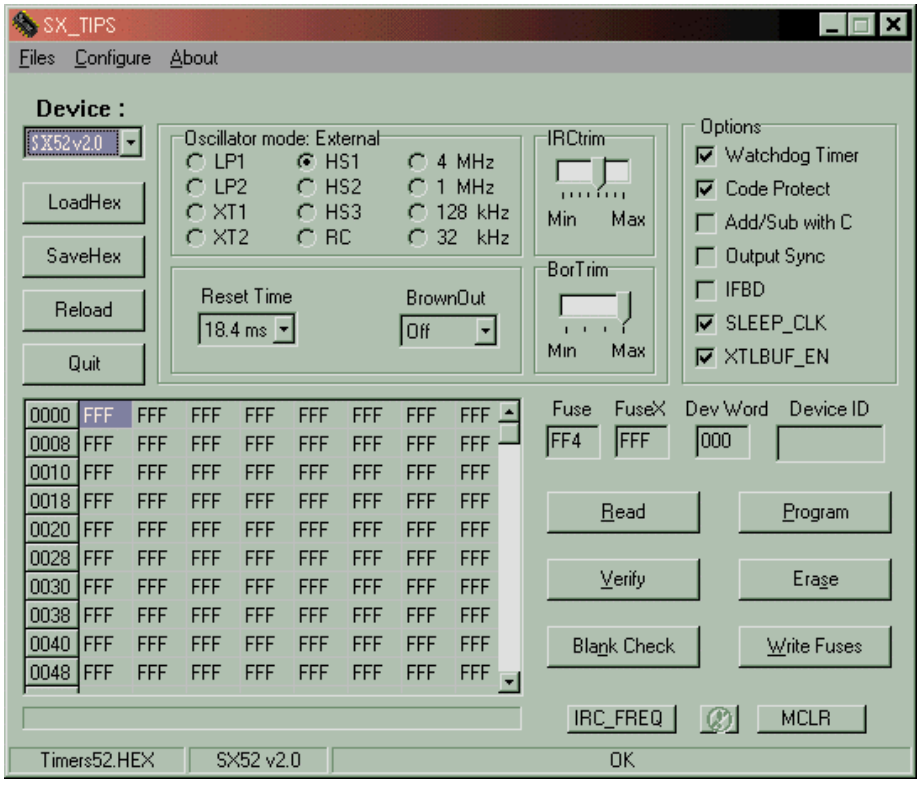
Target microcontroller type can be changed in Device field. Please ensure, that you enter right microcontroller type (see Chapter 4 for details). Programmer can also detect connected microcontroller type automatically, when **Read** function is performed.

SaveHex button allow save hex file, read from the target SX. File will be saved in INHX8M format.

Reload button will load the same file, as previously loaded with LoadHex button. This option is useful in debug applications.

Built-in **Hex editor** allow manual change of program memory before programming SX microcontroller. This can be useful for hot on-the-fly code changes or some constants changing.

Device options



Memory Map – Configured Memory Size (SX18/28 only). These factory-configured options should not be changed unless you want to reduce the configured amount of program memory in the device.

Bor Trim – Brown Out trim bits. These bits allow fine tune of brownout level. Parts are shipped from factory untrimmed.

IRC Trim – Internal RC Oscillator trim bits. These bits adjust the operation of the internal RC oscillator to make it operate within the target frequency range of 4.0 MHz (typical) plus or minus 8%. Parts are shipped from factory untrimmed. To measure current microcontroller frequency, press **IRC_FREQ** button. Measured internal RC oscillator frequency will be displayed in the status bar. You can adjust IRC trim value, program target SX and measure resulting frequency.

Oscillator Mode – This combination of bits sets up the device to operate with a particular type of external oscillator or sets the divide-by factor for generating the instruction clock from the internal oscillator. Eight external oscillator options determined as follows:

LP1 – low-power crystal (32 kHz)

LP2 – low-power crystal/resonator (32 kHz to 1 MHz)

XT1 – low power crystal/resonator (32 kHz to 10 MHz)

XT2 – normal crystal/resonator (1 MHz to 24 MHz)

HS1 – normal crystal/resonator (1 MHz to 50 MHz)

HS2 – normal crystal/resonator (1 MHz to 50 MHz)

HS3 – normal crystal/resonator (1 MHz to 50 MHz)

RC – external RC, OSC2 is pulled high with a weak pullup, no CLKOUT output

Four nominal instructions rates with internal oscillator determined as 4 MHz, 1 MHz, 128 kHz and 32 kHz.

Pins – Selects the number of pins for SX18 (20)/28. Set it to the correct value, according to your target microcontroller.

Reset Time – Delay Reset Timer (DRT) timeout period (SX48/52 only). The SX18 (20)/28 devices have a fixed startup time of 18 ms when the wakeup reset occurs. The SX48/52 devices have a programmable startup time. Timeout period can be one of the following: 0.06 ms, 18 ms, 60 ms, 960 ms.

BrownOut – Set the Brown Out Reset threshold voltage. When the supply voltage to the SX device drops below a specified value but remains above zero volts, it is called as "brown-out" condition. The SX device has a brownout detection circuit that puts the device into the reset state when the brownout occurs, and allows the device to re-start when the brownout condition ends. This feature prevents the device from producing abnormal results when the supply voltage falls to unreliable levels. The brownout threshold voltage can be programmed to the following three levels – 4.2 V, 2.6 V, 2.2 V. If the supply voltage drops below this level but remains above zero, the brownout circuit holds the SX device in the reset state. When the voltage rises above this threshold, the device start operating again, starting at the reset address. Fine-tuning of the brownout threshold voltage can be done with brownout trim bits.

If the brownout detection circuit disabled, device will still operate below the brownout threshold voltage, but will produce unreliable results if the supply voltage falls too low.

Turbo mode – SX18 (20)/28 only option. If set, the instruction clock rate is equal to the oscillator clock rate. If cleared, the instruction rate operates at one-fourth the oscillator clock rate. See also Chapter 4, option 7).

Watchdog Timer – Enable Watchdog Timer. If set, a watchdog timeout occurs when the watchdog timer overflows. This feature provides an escape mechanism from an infinite loop or other abnormal program conditions. When a watchdog timeout occurs, it resets the device just like assertion of the MCLR input. The watchdog oscillator has a nominal frequency of 14 kHz; at this rate 8-bit watchdog timer overflows in 18 ms. the watchdog period can be increased under program control up to 2.34 s with OPTION register.

Code Protect – Code Protection. If set, the program code and configuration registers read back as scrambled data. This prevents reverse engineering of your proprietary code and configuration options.

Add/Sub with C – Carry bit Input. If set, carry bit will be add into all addition operations (ADD fr, W means fr=fr+W+C); and to subtract the complement of the carry bit from all subtraction operations (SUB fr, W means fr=fr-W-/C). If clear, carry bit will be ignored as an input for addition and subtraction operations.

Input Sync – Synchronous Input Mode. If set, input signals will be synchronized with internal clock through two internal flip-flops. If clear, input signals will go directly to the port inputs.

IFBD – Internal Feedback Disable. External oscillator mode only. If set, the crystal/oscillator can rely on the internal feedback resistor between the OSC1 and OSC2 pins. If clear, an external feedback resistor is required between the OSC1 and OSC2 pins.

Option Extend – OPTION register Extension and Stack Extension, SX18 (20)/28 only option. If set, enable programmability of bit 6 and 7 in the OPTION register, the RTW and RTE_IE bits and to extend the stack

size to eight locations. If clear, disable programming of the RTW and RTE_IE bits and to limit the program stack size to two locations.

SLEEP_CLK – Sleep Clock Disable, SX48/52 only option. If set, enable clock operation during power down mode to allow fast start-up. If clear, disable clock operation during power down mode to reduce power consumption.

XTLBUF_EN – Crystal Buffer enable, SX48/52 only option. Set, if use external resonator/crystal. Clear to reduce power consumption, if use internal oscillator or external clock signal.

Fuse window – This window display all Fuse Word Register Configuration bits. It is possible to modify it content directly; however selecting options with as described above will be more easy and friendly.

FuseX window – This window display all FuseX Word Register Configuration bits. It is possible to modify it content directly; however selecting options with as described above will be more easy and friendly.

Dev Word window – This window display Device Word: hard-wired read-only Part ID. The following Part ID exist:

FEE	SX18 (20)/28 v2.5
FDE	SX28 v4.1
FCE	SX28 v5.2
001	SX48/52 v1.0
002	SX48/52 v2.1

Device ID window – This window allow to add user identification to the programmed device. This identification will be programmed into special user code ID locations, not accessible from the user program. Up to 8 ASCII symbols can be written. This ID can help identify the code-protected device, when read it.

Table of Contents