

STart276 Evaluation Kit

for ST10F276 Microcontroller, Art-Nr.: 995

STart27x Development Kit

for ST10F27x Microcontroller, Art-Nr.: 996



P.O: Box 1103
Kueferstrasse 8
Tel. +49 (7667) 908-0
sales@fsforth.de

- D-79200 Breisach, Germany
- D-79206 Breisach, Germany
- Fax +49 (7667) 908-200
- <http://www.fsforth.de>

© Copyright 2003:

FS FORTH-SYSTEME GmbH
Postfach 1103, D-79200 Breisach a. Rh., Germany

Release of Document: February 02, 2004
Filename: STRT27xc.doc
Author: Joachim Jaeger

All rights reserved. No part of this document may be copied or reproduced in any form or by any means without the prior written consent of FS FORTH-SYSTEME GmbH.

Table of Contents

1. Introduction.....	5
2. Features of STart27x Kits.....	7
2.1. Evaluation Kit based on the Evaluation Board	7
2.2. Development Kit based on the Development Board	8
3. Getting started.....	9
4. Software Tools	13
5. EVA27x Memory Mapping	14
5.1. Memory-Configuration	15
5.2. Settings of Jumper J1 (X15).....	16
6. System Start-up Configuration.....	17
6.1. EMULATION Mode.....	17
6.2. Adapt Mode	18
6.3. Bootstrap Mode	18
6.4. Bus-Access-Types.....	19
6.5. Write Configuration.....	20
6.6. Chip-Select-Lines	20
6.7. Segment Address Lines	21
6.8. RANGE START ADDRESS.....	23
6.9. EVA27x Memory Layout	24
6.10. Clock Generation	25
6.11. Single-Chip-Mode	26
6.12. User Switches and analog Components	27
6.13. Switch S2	28
6.14. LED 4, 5, 6, 7	28
6.15. LED 1, 3.....	29
6.16. Push button S8	29
6.17. Potentiometer P1, P2.....	29
6.18. Buzzer H1	30
6.19. 7 Segment Display.....	30
6.20. CAN Bus-Termination	30
6.21. RS232 Interface, K-Line Interface	31

7. Connectors	32
7.1. CPU signal connector X1	32
7.2. User hardware connector X10	33
7.3. X4, CAN1 Connector (bottom), CAN2 Connector (top)	34
7.4. X3, RS232 Interface.....	34
7.5. X12, K-Line Interface	35
7.6. Jumper J5, J6	35
7.7. X6, X7, X8, Debug Interface	36
8. Power-Supply.....	39
9. User Oscillator	40
10. CD-ROM	41
10.1. Content of CD-ROM	41
10.2. DEMO for KEIL.....	42
10.3. DEMO for TASKING.....	42
10.4. FLASH166 / WINFLASH166.....	43
10.5. Board configuration	43
10.6. Error conditions FLASH166 / WINFLASH.....	44
10.7. ST10Flasher	45
10.8. PLS UDE.....	45
11. Building the sample project with the TASKING C-Compiler	46
11.1. Project for the debugger	47
11.2. Project for the external FLASH	48
11.3. Project for the internal Flash with ST10Flasher	49
11.4. Project for the internal Flash with UDE	50
12. Building the sample project with the KEIL C-Compiler	51
12.1. Project for the debugger	52
12.2. Project for the external FLASH	52
12.3. Project for the internal Flash with ST10Flasher	54
12.4. Project for the internal Flash with UDE	55
13. Know-how for handling ST10F27x IDEs.....	56
14. Appendix.....	57
14.1. Top View of the EVA27x boards.....	57

1. Introduction

The product line of FS FORTH-SYSTEME got brand new members of user-friendly, easy-to-use and powerful evaluation kits which are offered as "STart276 Evaluation Kit" (995) and "STart27x Development Kit" (996). The kits differ from each other by their boards, a development board and an evaluation board. The development board can be used for different CPUs from the ST10F27x series and is equipped with debug connectors, external SRAM and external FLASH memory. The evaluation board is a subset of the development board and has no debug connectors, no external FLASH memory and the CPU is soldered, while the development board only has a socket for different CPUs. At the moment there is only the ST10F276 CPU available for the boards. Therefore all of the features mentioned in this manual, depend on the ST10F276 CPU.

The evaluation board is shipped with a ST10F276 controller which provides a 512 kByte on-chip IFLASH memory with 32bit access, a 320 kByte on-chip XFLASH memory with 16bit access, a 66 kByte on-chip extension RAM (XRAM), a 2 kByte internal SRAM (IRAM) two CAN controller which are working according to ISO11898 (profile 2.0B) and with many additional I/O features. The board supports up to 2MByte external Flash memory and up to 1MByte Fast-SRAM which allows to build a large number of sophisticated applications.

The ST10F276 is a derivative component of the ST10 series with many features. Please refer to the ST10F276 datasheet on the CD-ROM shipped with this package. Check the web page of STMicroelectronics (www.st.com) for updated versions of the device datasheet. A set of user selectable switches (DIP) offers a very flexible handling of the CPU specific start-up configuration. Therefore the STart27x Kits promises a successful start-up development for beginners and experienced users. The kit comes with a Flash programming tool and a well prepared demo software which helps you to become an experienced specialist for ST10 controllers.

The engineers of FS FORTH-SYSTEME created these boards to fulfill a lot of current and future user requirements. Therefore the boards come with two pre-assembled CAN transceivers. Both can easily be used without the need of additional hardware. Furthermore the controllers can run up to their maximum CPU-clock of 64 MHz (ST10F276). For fast prototyping the boards include a small wire wrap field, two potentiometers for analog input signals, a serial

EEPROM, a set of LEDs, a 7-segment display, a buzzer and a push button to make development easy and simple. Please refer to the following chapters to get more details about the ST10F276, the board's capabilities and the tools which are coming with the STart276 Evaluation Kit and STart27x Development Kit.

2. Features of STart27x Kits

2.1. Evaluation Kit based on the Evaluation Board

- ◆ CPU soldered on the board (e.g. ST10F276)
- ◆ 256 kByte high speed SRAM, expandable up to 1MByte
- ◆ no external Flash fitted, but expandable up to 2MByte
- ◆ Real Time Clock on chip
- ◆ Two CAN transceiver (PCA82C250T, $\pm 15V$ common mode offset voltage)
- ◆ Two CAN plugs (DSUB-9)
- ◆ Selectable bus configuration via DIP switch
- ◆ Selectable bootstrap mode via DIP switch
- ◆ 128 pin connector providing all CPU signals for user purposes
- ◆ Additional wire-wrap field for individual hardwired applications
- ◆ External, serial EEPROM (24LC02, I2C-Interface) for user purposes
- ◆ The Evaluation Board is designed for running ST10F27x CPUs (e.g. ST10F276 at 64 MHz)
- ◆ Power supply 90..264VAC/5VDC enclosed. Input voltage to the Evaluation Board (X2) connector is 5VDC only
- ◆ RS232-Interface cable enclosed
- ◆ Flash programming utility for easy downloading user specific code to external Flash memory.
- ◆ Hex-to-binary converter tool provided on CD-ROM as freeware
- ◆ Stand alone Flash programming utility to program the embedded on-chip Flash of the ST10F27x, provided by STMicroelectronics
- ◆ C compiler (demo version only) provided by TASKING and KEIL
- ◆ Demo version of the PLS-debugger
- ◆ Including a socket to carry a four pin crystal oscillator device

2.2. Development Kit based on the Development Board

- ◆ Socket for CPU soldered on the board, no CPU is shipped
- ◆ 256 kByte high speed SRAM, expandable up to 1MByte
- ◆ 512 kByte external Flash, expandable up to 2MByte
- ◆ Connectors for a logic analyzer
- ◆ Two CAN transceiver (PCA82C250T, $\pm 15V$ common mode offset voltage)
- ◆ Two CAN plugs (DSUB-9)
- ◆ Selectable bus configuration via DIP switch
- ◆ Selectable bootstrap mode via DIP switch
- ◆ 128 pin connector providing all CPU signals for user purposes
- ◆ Additional wire-wrap field for individual hardwired applications
- ◆ External, serial EEPROM (24LC02, I2C-Interface) for user purposes
- ◆ The Development Board is designed for running ST10F27x CPUs (e.g. ST10F276 at 64 MHz)
- ◆ Power supply 90..264VAC/5VDC enclosed. Input voltage to the Evaluation Board (X2) connector is 5VDC only
- ◆ RS232-Interface cable enclosed
- ◆ Flash programming utility for easy downloading user specific code to external Flash memory.
- ◆ Hex-to-binary converter tool provided on CD-ROM as freeware
- ◆ Stand alone Flash programming utility to program the embedded on-chip Flash of the ST10F27x, provided by STMicroelectronics
- ◆ C compiler (demo version only) provided by TASKING and KEIL
- ◆ Demo version of the PLS-debugger
- ◆ Including a socket to carry a four pin crystal oscillator device

3. Getting started

As already mentioned above, this paragraph intends to explain how you can run the board without writing own software. You will learn how the controller's internal bootstrap mode can be used to initiate a simple download of a binary program file to the embedded on-chip and external Flash memories. If you would like to use the environments of KEIL or TASKING you should refer to chapter 10.

First we recommend to create a project path to copy the demonstration software FLASH166, from CD-ROM. This software includes a bootstrap and a monitor program. The bootstrap module is a tiny piece of FS FORTH-SYSTEME firmware which is downloaded to the internal 2 kByte SRAM (IRAM) memory of the controller. It makes sure that other programs, like the monitor can be loaded to any memory location. For more details please refer to chapter 5.1 "Memory-Configuration". By the way: the user may select either, the external FLASH (if equipped on board), the external SRAM memory, or the internal Flash memory to download his individual application software.

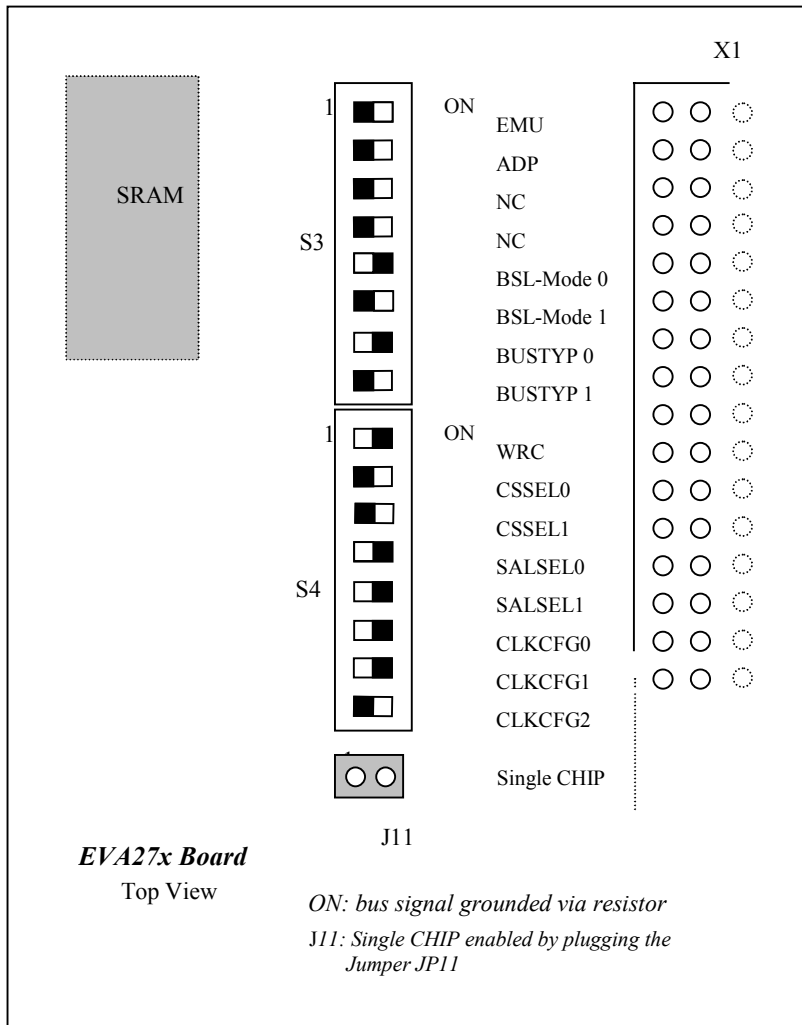
To setup your board successfully, please take care of the following steps:

- ◆ Setup the board's DIP switches as shown in Figure 1 (the board is pre-configured by FS FORTH-SYSTEME according to Figure 1) and the jumper J1 (X15). Refer to chapter 6 "System Start-up Configuration" for detailed information about these settings.
- ◆ Check the connections of the cables at the X1-and X10-connector:
C16(X1) to 5(X10), D16(X1) to 8(X10), C17(X1) to 7(X10) , D17(X1) to 6(X10), A25(X1) to 12(X10), B25(X1) to 13(X10), A17(X1) to 15(X10).
These connections are necessary to toggle the LEDs LE4 to LE7, to read the potentiometers P1, P2 and to send a PWM signal to the buzzer H1.
- ◆ Make sure that the BSL-Mode 0 switch is set to ON (S3-5=ON / P0L.4=low) and the BSL-Mode 1 switch is set to OFF (S3-6=OFF / P0L.5=high). This will force the CPU to bootstrap mode right after reset or power on. The Standard Bootstrap mode is always entered by the CPU whenever pin P0L.4 has been grounded prior to a reset.

- ◆ Connect the board's serial interface (X3) to COM1 or COM2 of your Personal Computer. Caution: CTS and RTS signals are not supervised by the board's serial interface and the bootstrap loader software.
- ◆ Copy the sample project, depending on your board, from the attached CD-ROM into your project path:
"STart27x\Keil\C166\Examples\Boards\FORTH_EVA27x\RTC27X\Release\RTC27X.BIN"
- ◆ Copy "FLASH166.EXE" and "FLASH166.OVL" onto your hard disk. Both files allow you to run the bootstrap loader and monitor in an opened DOS box.
- ◆ Make sure that the path of the "FLASH166" directory is included in your path environment, if you choose another directory than for your project before. To choose between SRAM and FLASH, you have to remark the corresponding command in the file "PROG.BAT".
- ◆ The tool ST10Flasher is provided within the tool directory on CD-ROM. Just unzip the file and run Setup.exe to install it. This tool allows programming the internal Flash memory of the ST10F27x controllers.
Now you can run the ST10Flasher and select the Hex file to download into the internal Flash:
"STart27x\Keil\C166\Examples\Boards\FORTH_EVA27x\RTC27X\Intern\obj\RTC27X.h86"
- ◆ Another tool to program the internal Flash memory is the UDE from PLS, which is also provided on CD-ROM.
- ◆ Now move the BSL-Mode 0 switch to OFF (S3-5=OFF), and press the reset button S1. Run a terminal program and select the line parameters: 9600Baud, No parity, 8 bit data, 1stopbit.

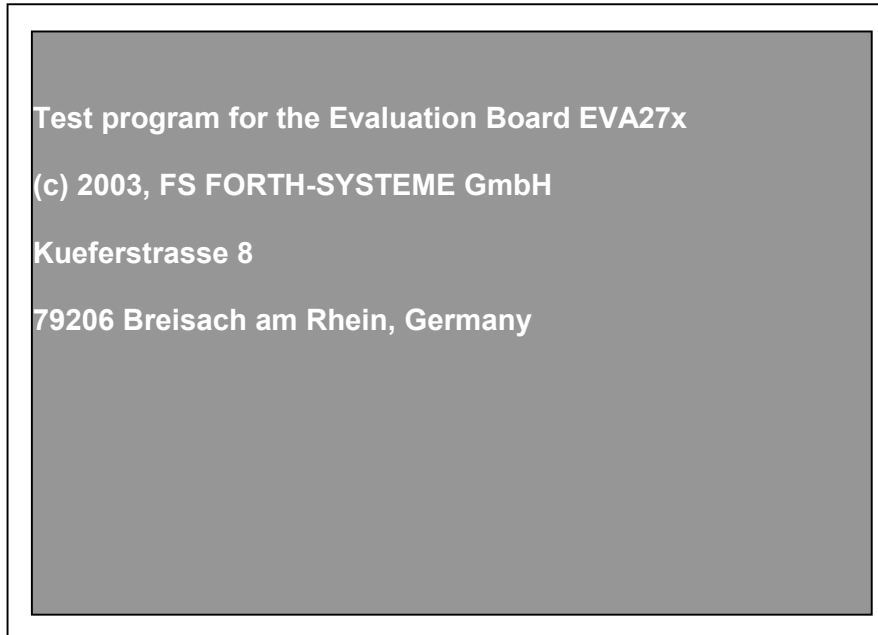
If everything is working properly a small text will appear on your PC screen as shown in Figure 2. Also the LEDs LE4 to LE7 will represent a flashing light from LE4 to LE7.

Figure 1: Factory setting



Single CHIP is enabled at evaluation board by default.

Figure 2: Output of the example RTC27x



4. Software Tools

The *STart27x* Kits always includes the bootstrap loader and monitor program (FLASH166) from FS FORTH-SYSTEME, as well as the ST10Flasher from STMicroelectronics and the UDE from PLS. That's everything you need to run your individual software on a ST10F27x controller. This software is a part of the *STart27x* Kits.

FLASH166 / WINFLASH166 package allows you

- ◆ to download your program to an external Flash or SRAM
- ◆ to request a memory dump from the target
- ◆ to test serial interface, CPU-I/Os and a lot of more things

ST10Flasher allows you

- ◆ to download your program to the internal Flash

UDE allows you

- ◆ to download your program to the internal Flash
- ◆ to debug your application

If you would like to use a sophisticated programming tool which provides you the advantages and performance of "C", we would appreciate to make you an interesting offer at any time. FS FORTH-SYSTEME is distributor for KEIL and TASKING. Both companies are manufacturer of well known "C/C++" compilers and debugging tools. For the very first beginning you may use a free demo of the KEIL and TASKING "C" compilers which allow you to create individual programs with a restricted size of program code.

A detailed and well documented description of writing software with TASKING and KEIL environments is attached to this manual. If you want to run the ST10F27x as a standalone controller you may use the ST10Flasher utility which is designed and provided by STMicroelectronics, or the UDE from PLS. They allow a fast download of your program into the internal memory section of the CPU.

5. EVA27x Memory Mapping

The EVA27x board supports a number of memory architectures according to the capability of the ST10F27x controller.

The evaluation board is provided with the following components:

Table 1: Memory evaluation board

	Memory	Type	Chip-Select	Size	Comment
1	FLASH, Boot	M29F400BB90N1 or M29F800BB90N1 or M29F160D	CSFLSHA	256k * 16 or 512k * 16 or 1024k * 16	optional or optional or optional
2	SRAM, High Speed	2 * μ PD431008LE or 2 * μ PD434008LE	CSRAM	2 * 128k * 8 or 2 * 512k * 8	assembled or optional

The development board is provided with the following components:

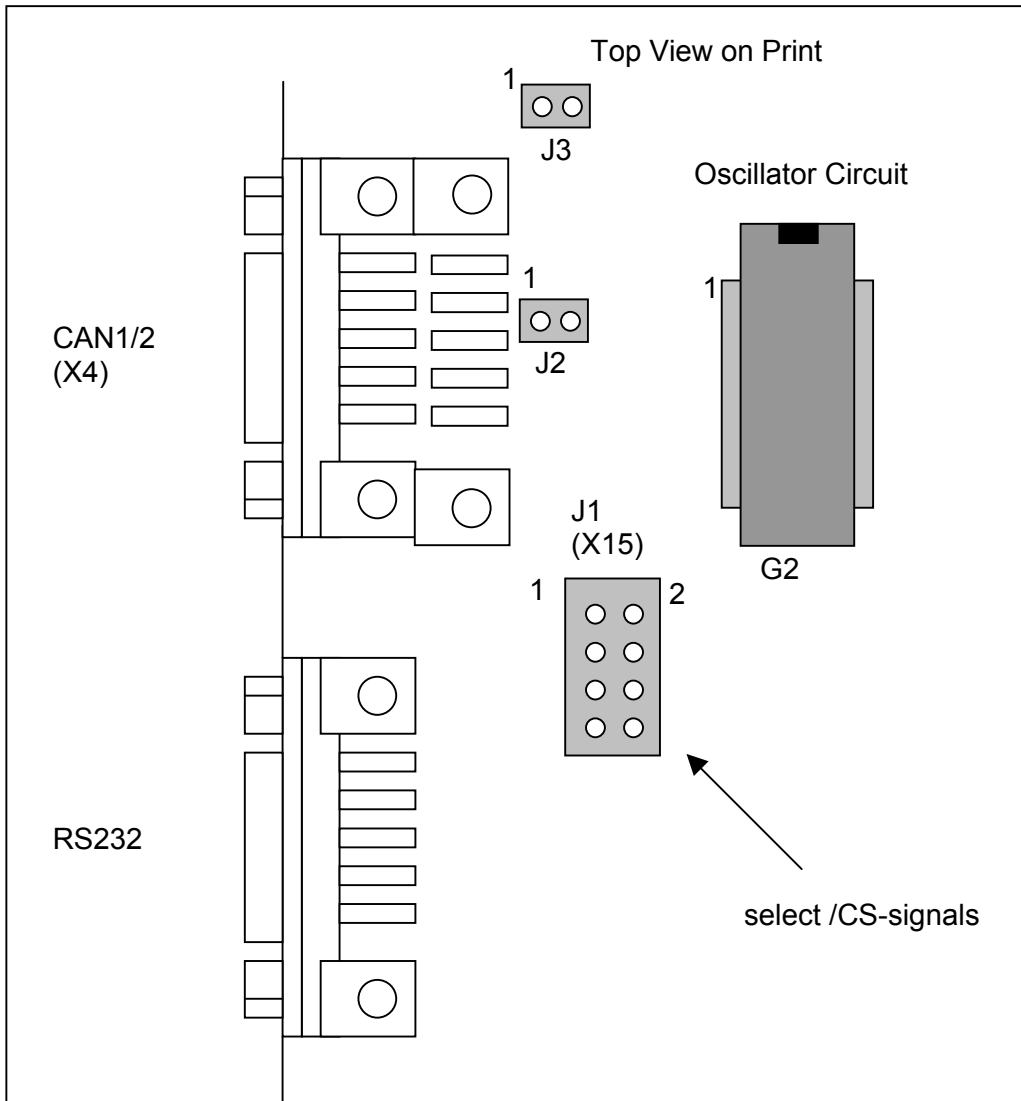
Table 2: Memory development board

	Memory	Type	Chip-Select	Size	Comment
1	FLASH, Boot	M29F400BB90N1 or M29F800BB90N1 or M29F160D	CSFLSHA	256k * 16 or 512k * 16 or 1024k * 16	assembled or optional or optional
2	SRAM, High Speed	2 * μ PD431008LE or 2 * μ PD434008LE	CSRAM	2 * 128k * 8 or 2 * 512k * 8	assembled or optional

All components marked "optional" are not mounted when the board is leaving FS FORTH-SYSTEME. You may expand the board's addressable and non-volatile program memory up to a maximum of 2 MByte (Flash part must be a 29F160D).

5.1. Memory-Configuration

Figure 3: Selection of Chip Select



5.2. Settings of Jumper J1 (X15)

Table 3: Chip Select

PIN	A	B	Chip Select	Pre-configured evaluation board	Pre-configured development board
1	■		/CS0 -> CSFLSHA	Yes	Yes
2					
3	■		/CS1 -> CSRAM	Yes	Yes
4					
5		■	/CS0 -> CSRAM	No	No
6					
7		■	/CS1 -> CSFLSHA	No	No
8					

■ = connected pins

To make sure that your system works properly you may choose the setting shown in column (A), this provides that the system boots from an external Flash device, if mounted. Otherwise if the program resides within an external SRAM you should select column (B). All jumpers which are marked as pre-configured are set by FS FORTH-SYSTEME prior to delivery.

6. System Start-up Configuration

Although most of the programmable features of the ST10F27x are either selected during the initialization phase (software) or repeatedly done during the program execution (software), there are some configurations that **MUST** be selected earlier, because they are used for the very first access of the CPU (hardware). For example, the CPU has to know how to deal with the external bus interface since the external memory may be organized as an 8 bit multiplexed/non multiplexed or a 16 bit multiplexed/non multiplexed bus. These selections are made during reset at the pins of PORT0, which are read at the end of the internal reset sequence. During reset, CPU-internal pull-up devices at the PORT0 lines are active, so a high level will be read, if the respective pin is left open. To get a low level, the pin must be pulled down by an external resistor (4.7k..8.2k). These resistors are already provided on the board.

The following paragraphs explain how to set all of these start-up configuration pins. Please refer to Figure 1 to find the corresponding location at the S3 and S4 switches. If a signal has to be configured as “0” (low level), move the switch to the ON-position. A “1” (high level) appears when the switch is left open (OFF-position).

6.1. EMULATION Mode

PIN EMU, S3-1

Pin P0L.0 selects the Emulation Mode, when low during reset. This mode allows the access to the integrated XBUS peripherals via the external bus interface. This mode is used for special emulator purposes and, in general, is of no use in basic applications.

Default: Emulation Mode off (P0L.0=1 / S3-1=OFF)

6.2. Adapt Mode

PIN ADP, S3-2

Pin P0L.1 selects the ADAPT Mode when low during reset. In this mode, the ST10F27x goes into a passive, floating state, which is similar to the state during reset. This mode allows, to switch the ST10F27x virtually off, so that an emulator may control the circuitry, since the pins are floating. This mode is used for special emulator purposes only.

Default: Adapt Mode off (P0L.1=1 / S3-2=OFF)

6.3. Bootstrap Mode

PIN BSL-Mode 0, S3-5

PIN BSL-Mode 1, S3-6

With the BSL-Mode pins it is possible to select between user code, standard Bootstrap Loader and alternate Bootstrap Loader:

Table 4: Bootstrap Loader Mode Selection

S3-6	S3-5	Bootstrap Loader Mode
OFF	OFF	User Code execution
OFF	ON	Standard Bootstrap Loader Mode
ON	OFF	Alternate Boot Mode
ON	ON	Reserved

The Standard Bootstrap loader allows moving a tiny start code (32 Bytes) into the internal RAM of the ST10F27x via the serial interface ASC0 (asynchronous, serial mode), or 64 Bytes via the CAN (P4.5 CAN1_RXD). The controller remains in BSL mode until a hardware reset with S3-5=OFF occurs. Bootstrap loader mode may also be exited through a software reset instruction. In this case

S3-5 is ignored and program execution starts, fetching code from 0x00000 (internal or external).

Refer to the user manual for more details.

Default: Standard Bootstrap Mode (P0L.4=0 / S3-5=ON, P0L.5=1 / S3-6=OFF)

6.4. Bus-Access-Types

PIN BUSTYP0, S3-7
PIN BUSTYP1, S3-8

With the pins P0L.6 (BUSTYP0) and P0L.7 (BUSTYP1) the bus mode can be selected for /CS0.

Table 5: Boot Bus Mode Selection

S3-8	S3-7	External Data Bus Width	External Address Bus Mode
ON	ON	8-bit Data	Demultiplexed Addresses
ON	OFF	8-bit Data	Multiplexed Addresses
OFF	ON	16-bit Data	Demultiplexed Addresses
OFF	OFF	16-bit Data	Multiplexed Addresses

In multiplexed bus modes PORT0 drives both, the 16-bit intra-segment address and the 8 (16) bit output data, while PORT1 remains in high impedance state. In demultiplexed bus modes PORT1 drives the 16-bit intra-segment address, while PORT0 (P0H, P0L) drives the 16-bit Data output. If a 8-bit data width has been selected, only P0L is driving the data output.

Default: Data-Bus-Width = 16-bit; Bus Mode = Demultiplexed (P0L.6=0 / S3-7=ON, P0L.7=1 / S3-8=OFF)

6.5. Write Configuration

Pin WRC, S4-1

Pin P0H.0 selects the initial operation of the control pins /WR and /BHE. When set high (P0H.0=1 / S4-1=OFF), this pin selects the standard mode, i.e. /WR and /BHE. When set low (P0H.0=0 / S4-1=ON) the alternate function is selected: /WR = /WRL and /BHE = /WRH

Default: /WRL and /WRH selected (P0H.0=0 / S4-1=ON)

6.6. Chip-Select-Lines

PIN CSSEL0, S4-2

PIN CSSEL1, S4-3

These signals choose the number of chip select lines, which are active after reset. Unused chip select lines may be used as general purpose I/O.

Table 6: Active Chip Select

S4-3	S4-2	active chip select	Note
OFF	OFF	/CS0.../CS4	Default
OFF	ON	none	5 Port 6 Pins free for I/O
ON	OFF	/CS0.../CS1	3 Port 6 Pins free for I/O
ON	ON	/CS0.../CS2	2 Port 6 Pins free for I/O

Default: All /CS signals used (P0H.1=1 / S4-2=OFF, P0H.2=1 / S4-3=OFF)

6.7. Segment Address Lines

PIN SALSEL0, S4-4

PIN SALSEL1, S4-5

During reset these control lines define the number of active segment address lines. This allows to select pins of port 4 to work as general I/O lines as well as to become a part of the controller address logic. Depending on the selection of SALSELx, the required address space is chosen right after the system start-up and the program may address all locations without prior programming. Even if not all segment address lines are enabled on port 4, the ST10 uses it's complete 24-bit addressing mechanism. This allows to generate all CS signals over the entire address space even if the external bus width is less in size.

CAUTION!

The selected number of segment address lines **cannot** be changed by software after reset.

Table 7: Segment Address Lines

S4-5	S4-4	Segment Address Lines	Directly addressable space
OFF	OFF	A17.....A16	256 kByte
OFF	ON	A23.....A16	16 Mbyte, CAN modules disabled !
ON	OFF	None	64 kByte
ON	ON	A19.....A16	1 Mbyte; Default

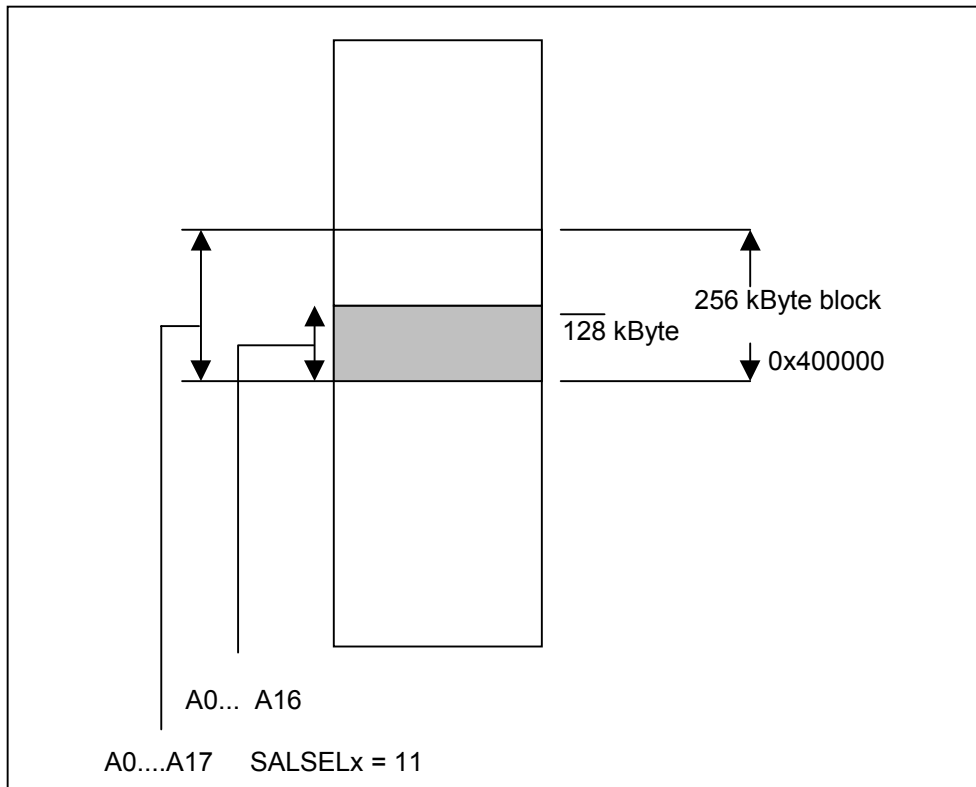
Default: 1MByte address space (P0H.3=0 / S4-4=ON, P0H.4=0 / S4-5=ON)

Example:

Assume that your application requires the 128 kByte SRAM block at the absolute address 0x400000. Furthermore the /CS1 signal should control both, reading from and writing to the SRAM. The controller must at least supervise the address

lines A0 to A16 in the direct addressing mode (SALSEL0 = OFF, SALSEL1 = OFF).

Figure 4: Example



Define BUSCON1 Register:

/CSWEN = /CSREN = 1, RDYPOL = 0; RDYEN = 0; BUSACT = 1;
 ALECTL = 0; BTYP = 10; MTTC = 1; RWDC = 1; MCTC = 1111;

Define ADDRSEL1 Register

RGSZ (Range-Size-Selection) = 0101 (128 kByte block)
 RGSAD (Range-Start-Address) = 0100000

6.8. RANGE START ADDRESS

Depending on the chosen block size (see previous example) the range start address "RRRRRRRxxxxx" specifies the upper bits (A23....A12) of the respective address area. An "R" stands for a bit which has to be set within the RGSAD register if this address line is not supervised by the controller bus logic. All bits expressed as an "x" are exclusively maintained by the bus address signals.

Referring to the example, the register configuration "RRRRRRRxxxxx" lets A12 to A16 being controlled by the controller address bus. The address lines from A17 to A23 are configured within the RGSAD bits which will be taken by the CPU to compute the internal, absolute 24-bit address.

All memory areas which are not handled by BUSCON(x) and ADRSEL(x) (x = 1...4) are maintained by BUSCON0 and ADRSEL0.

Now specify ADRSEL1 register:

```
0x400000 = 0100.0000.0000.0000.0000.0000
           |         |         |         |         |
           |         |         |         |         *
RGSAD =   |0100.0000.0.0000 **
```

ADRSEL1 = 0100.0000.0000.0101

* controlled by bus address lines A0..A16

**cursive bits are discarded since they are already controlled by address bus

6.9. EVA27x Memory Layout

The following table shows some address selections which are only suggestions by FS FORTH-SYSTEME. The memory areas may be changed to fulfill different user requirements. Use BUSCONx, ADRSELx registers and the configuration registers as already described in a previous chapter to setup your individual memory layout.

Table 8: Typical Memory Layout

Address Range	Chip-Select	Bus Width	Function
0x4F.FFFF 0x40.0000	/CS4	?	Not connected yet
0x3F.FFFF 0x30.0000	/CS3	?	Not connected yet
0x20.0FFF 0x20.0000	/CS2	?	Not connected yet
0x1F.FFFF 0x10.0000	/CS1	16 bit	SRAM, up to 1 MByte
0x0F.FFFF 0x00.0000	/CS0	16 bit	Boot-Flash, up to 2 MByte at development board only

6.10. Clock Generation

PIN CLKCFG0, S4-6

PIN CLKCFG1, S4-7

PIN CLKCFG2, S4-8

The external clock input range refers to a CPU clock range of 1..64 MHz. Besides, the PLL usage is limited to a 4-8MHz input frequency range. All configurations need a crystal to generate the CPU clock through the internal oscillator amplifier, except the Direct Drive mode (oscillator amplifier disabled, so no crystal or resonator can be used). Vice versa, the clock can be forced through an external clock source only in Direct Drive mode.

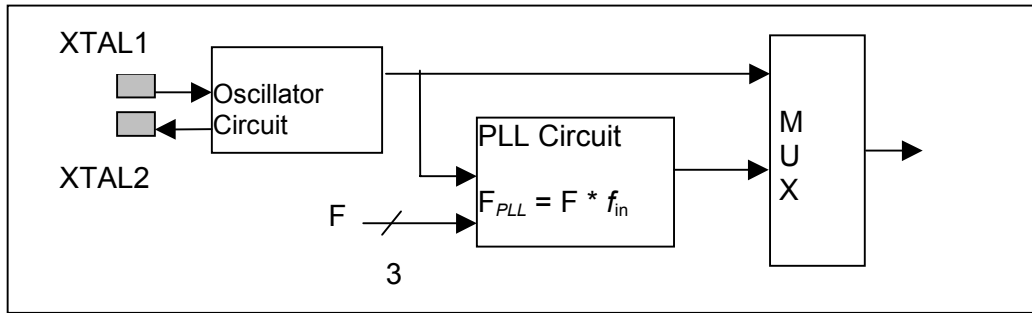
Table 9: Selection of CPU clock

S4-8	S4-7	S4-6	CPU-Clock $F_{cpu} = f_{xtal} * F$	External Clock Input Range in MHz (1)	PLL mode
ON	ON	ON	$f_{xtal} * 16$	4	Active
ON	ON	OFF	$f_{xtal} * 0.5$	4 to 8	Active
ON	OFF	ON	$f_{xtal} * 10$	4 to 6.4	Active
ON	OFF	OFF	$f_{xtal} * 1$	1 to 64 (direct drive)	Inactive
OFF	ON	ON	$f_{xtal} * 5$	4 to 8 (default)	Active
OFF	ON	OFF	$f_{xtal} * 8$	4 to 8	Active
OFF	OFF	ON	$f_{xtal} * 3$	4 to 8	Active
OFF	OFF	OFF	$f_{xtal} * 4$	4 to 8	Active

(1) The STart27x comes with an external 8 MHz quartz. The CLKCFG pins are set by default to a factor of 5 (CPU internal = 40 MHz!).

Default: $f_{xtal} * 5$ (P0H.5=0 / S4-6=ON, P0H.6=0 / S4-7=ON, P0H.7=1 / S4-8=OFF)

Figure 5: Clock generation



6.11. Single-Chip-Mode

In single chip mode the program is fetched from the on-chip Flash memory. Therefore the embedded Flash memory must contain a valid program to execute correctly. A utility for this can be found on the CD shipped with your STart27x Kit.

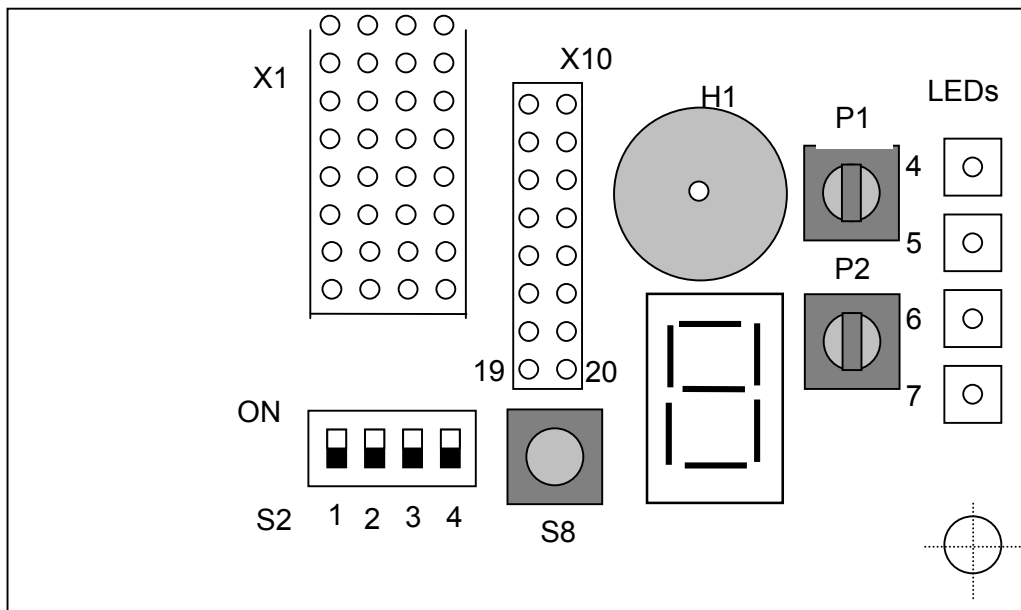
Single chip mode is entered when pin /EA is set high during reset. Place JP11 to Jumper J11 (Default at evaluation board) before restarting the CPU. If the jumper is not plugged, a pull-down resistor configures the CPU to run in external bus mode. Refer to Figure 1.

6.12. User Switches and analog Components

The EVA27x boards include a wire-wrap field, four LEDs, a 7 segment display and analog components for user specific purposes like self-training and trials. Most of them are directly wired to connector X10 and X1 whereby these components can be accessed by the CPU's port lines. A small serial EEPROM (24LC01) allows to store and restore user data through a standard I2C-Bus interface. To create a simple application the evaluation board comes with a set of junction wires which are provided to interface the CPU-connector X1 with X10.

To access these user components, you must connect your selected component from connector X10 to the CPU connector X1 with external wire, e.g. like the wires shipped with your STart27x Evaluation Kit/Development Kit.

Figure 6: User switch and analog components



Refer also to the attached top view schematics in the appendix.

6.13. Switch S2

PIN	Function description	Pin of X10
1	USER-Switch 1: If ON pulls down signal at X10.1 otherwise the pin becomes high active (5V)	X10.1
2	USER-Switch 2: If ON pulls down signal at X10.2 otherwise the pin becomes high active (5V)	X10.2
3	USER-Switch 3: If ON pulls down signal at X10.3 otherwise the pin becomes high active (5V)	X10.3
4	USER-Switch 4: If ON pulls down signal at X10.4 otherwise the pin becomes high active (5V)	X10.4

Table 10: User Switch S2

6.14. LED 4, 5, 6, 7

All user LEDs are pulled into a high active level (5V) through a series resistor (1K5). Switching one of these LEDs **ON**, the CPU signal, interfaced to X10, must be pulled down by software.

LED	Pin of X10
4	X10.5
5	X10.6
6	X10.7
7	X10.8

Table 11: LEDs (LE4..LE7)

6.15. LED 1, 3

Please refer to the top view schematic attached in the manual's appendix!

LED	Name	Function
1	POWER	Power On/Off signal
3	EINIT	Off after Reset, On after execution of EINIT

Table 12: LEDs (LE1, LE3)

6.16. Push button S8

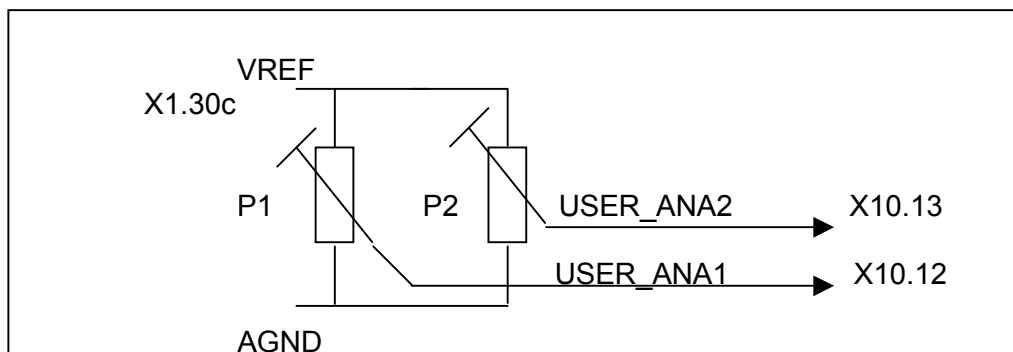
Gets into high level state when key is being pressed. Signal is interfaced to X10.11 connector.

6.17. Potentiometer P1, P2

These variable resistors allow the adjustment of the reference voltage provided by the CPU (connector X1.30C). The voltage can be adapted to the user specific application via X10.12 (P1) or via X10.13 (P2) to the desired port pin (e.g. X1.25A analog channel 0, or X1.25B analog channel 1).

Refer to Figure 6 for the correct position of the potentiometers at the EVA27x board.

Figure 7: Potentiometer (P1, P2)



6.18. Buzzer H1

The buzzer H1 is connected via a 100nF capacitor to the X10.15 connector. Connect a cable from this pin to the desired port pin (e.g. X1.17A XPWM output 0).

6.19. 7 Segment Display

This display V3 is connected to an I/O-extender, which is connected to the I2C interface. Each output of this I/O-extender is connected to a segment of the display. A low level at the output switches on the corresponding segment.

Remove jumpers J2 and J3, when using the I2C bus!

6.20. CAN Bus-Termination

A CAN bus system has to be terminated at both ends of the network with two resistors. The resistors should have a value of nominal 124 Ω (typically 120 Ω). The terminating resistor can be activated through jumpers J2 (CAN1) and J3 (CAN2).

Caution: Never use more than two termination resistors for a network. This might result into any unexpected bus transmission errors or may force the transceiver circuits to drop down their operation.

The connector X4 provides two D-Sub male connectors for the CAN-interface. The connector at the bottom is used for the CAN1-interface and the connector at the top is used for the CAN2-interface (see Figure 3).

You cannot use the I2C-bus, when using the CAN-bus!

6.21. RS232 Interface, K-Line Interface

The RS232 is interfaced to the board through connector X3 (see Figure 3). It allows the download of the monitor and bootstrap loader as well as the user application file. If the monitor is not in use, this interface can be taken as a standard RS232 for individual purposes. The RS232 hardware does not support any handshake signals, like RTS, CTS or DTR.

The same UART of the ST10F27x can also be used to realize a K-Line interface (ISO 9141). Herefore the jumpers J9 and J10 are used. The connector X12 provides the signals of the K-Line interface.

PIN	A	B	Serial interface	Pre-configured
1	■		K-Line	No
2	■	■		
3		■	RS232	Yes

Table 13: Jumper settings for J9

■ = connected pins

PIN	A	B	Serial interface	Pre-configured
1		■	RS232	Yes
2	■	■		
3	■		K-Line	No

Table 14: Jumper settings for J10

■ = connected pins

Choose column (A), if you want to use the K-Line interface, or column (B), if you want to use the RS232 interface.

7. Connectors

7.1. CPU signal connector X1

Pin	Row A	Row B	Row C	Row D
1	Address A0	Address A1	Data D0	Data D1
2	Address A2	Address A3	Data D2	Data D3
3	Address A4	Address A5	Data D4	Data D5
4	Address A6	Address A7	Data D6	Data D7
5	Address A8	Address A9	Data D8	Data D9
6	Address A10	Address A11	Data D10	Data D11
7	Address A12	Address A13	Data D12	Data D13
8	Address A14	Address A15	Data D14	Data D15
9	Address A16	Address A17	NC	NC
10	Address A18	Address A19	TXD0, serial	RXD0, serial
11	/RSTIN	NC	/RSTOUT	/READY
12	/CS0	/CS1	ALE	/EA
13	/CS2	/CS3	/NMI	/RD
14	/CS4	PORT 6.5	CAN_RXD1, P4.5	/WRH, P3.12
15	PORT 6.6	PORT 6.7	CAN_RXD2, P4.4	/WRL,
16	CAN_TXD1, P4.6	CAN_TXD2, P4.7	PORT 2.0	PORT 2.1
17	PORT 8.0	PORT 8.1	PORT 2.2	PORT 2.3
18	PORT 8.2	PORT 8.3	PORT 2.4	PORT 2.5
19	PORT 8.4	PORT 8.5	PORT 2.6	PORT 2.7
20	PORT 8.6	PORT 8.7	PORT 2.8	PORT 2.9
21	PORT 7.0	PORT 7.1	PORT 2.10	PORT 2.11
22	PORT 7.2	PORT 7.3	PORT 2.12	PORT 2.13
23	PORT 7.4	PORT 7.5	PORT 2.14	PORT 2.15
24	PORT 7.6	PORT 7.7	PORT 3.15	PORT 3.13
25	PORT 5.0	PORT 5.1	PORT 3.9	PORT 3.8
26	PORT 5.2	PORT 5.3	PORT 3.7	PORT 3.6
27	PORT 5.4	PORT 5.5	PORT 3.5	PORT 3.4
28	PORT 5.6	PORT 5.7	PORT 3.3	PORT 3.2
29	PORT 5.8	PORT 5.9	PORT 3.1	PORT 3.0
30	PORT 5.10	PORT 5.11	VAREF	RPD
31	PORT 5.12	PORT 5.13	GND	GND
32	PORT 5.14	PORT 5.15	VDD (+5V)	VDD (+5V)

Table 15: Connector X1

7.2. User hardware connector X10

PIN	FUNCTION
1	User Switch 1 (S2-1)
2	User Switch 2 (S2-2)
3	User Switch 3 (S2-3)
4	User Switch 4 (S2-4)
5	User LED 4
6	User LED 5
7	User LED 6
8	User LED 7
9	I2C Bus Clock Line
10	I2C Data Read/Write Line
11	User Pushbutton S8
12	Analog Voltage from potentiometer P1
13	Analog Voltage from potentiometer P2
14	Analog GND
15	Buzzer H1
16	I2C Interrupt from I/O-extender
17	NC
18	NC
19	VStby Standby Voltage
20	GND

Table 16: Connector X10

7.3. X4, CAN1 Connector (bottom), CAN2 Connector (top)

PIN	Signal
1	NC
2	CAN_L, differential Low signal
3	GND, optional
4	NC
5	NC
6	GND, optional
7	CAN_H, differential High signal
8	NC
9	NC

Table 17: Connector X4

Pins 2 and 7 may be connected via a terminating resistor (120 Ω) via jumpers J2 and J3. Refer to chapter 6.20 CAN Bus-Termination.

7.4. X3, RS232 Interface

PIN	Signal
1	NC
2	TXD
3	RXD
4	NC
5	GND
6	DTR0, static 10V, no handshake
7	NC
8	NC
9	NC

Table 18: Connector X3

The housings of all DSUB connectors are grounded.

7.5. X12, K-Line Interface

PIN	Signal
1	NC
2	VS
3	K
4	NC
5	LI
6	NC
7	NC
8	GND

Table 19: Connector X12

7.6. Jumper J5, J6

The jumpers J5 and J6 are only fitted at the development board. There it could be chosen to connect the quartz Q2 to the ST10F27x CPUs or to VDD_CPU/GND

PIN	A	B	Pin 143/144 connected to:	Pre-configured
1	█		VDD_CPU/GND	No
2		█		
3			quartz Q2	Yes

Table 20: Jumper settings for J5,J6

█ = connected pins

At the evaluation board the jumpers are replaced by two resistors: R51 and R52, which make a permanent connection between the CPU and the quartz.

7.7. X6, X7, X8, Debug Interface

The EVA27x boards provide pads for 3 Mictor connectors, which are fitted only at the development board.

These connectors are necessary, if a logic analyzer should be used.

Pin	Signal	Pin	Signal
1	nc	2	nc
3	/RD	4	Address A15
5	Address A14	6	Address A13
7	Address A12	8	Address A11
9	Address A10	10	Address A9
11	Address A8	12	Address A7
13	Address A6	14	Address A5
15	Address A4	16	Address A3
17	Address A2	18	Address A1
19	Address A0	20	Data D0
21	Data D1	22	Data D2
23	Data D3	24	Data D4
25	Data D5	26	Data D6
27	Data D7	28	Data D8
29	Data D9	30	Data D10
31	Data D11	32	Data D12
33	Data D13	34	Data D14
35	Data D15	36	/WRL
37	nc	38	nc

Table 21: Connector X6

Pin	Signal	Pin	Signal
1	nc	2	nc
3	ALE	4	CAN_TXD2, P4.7
5	CAN_TXD1, P4.6	6	CAN_RXD1, P4.5
7	CAN_RXD2, P4.4	8	Address A19
9	Address A18	10	Address A17
11	Address A16	12	PORT 6.7
13	PORT 6.6	14	PORT 6.5
15	/CS4	16	/CS3
17	/CS2	18	/CS1
19	/CS0	20	PORT 8.0
21	PORT 8.1	22	PORT 8.2
23	PORT 8.3	24	PORT 7.0
25	PORT 7.1	26	PORT 7.2
27	PORT 7.3	28	PORT 7.4
29	PORT 7.5	30	PORT 7.6
31	PORT 7.7	32	/READY
33	/RSTOUT	34	/RSTIN
35	/NMI	36	nc
37	nc	38	nc

Table 22: Connector X7

Pin	Signal	Pin	Signal
1	nc	2	nc
3	nc	4	RXD0, P3.11
5	PORT 2.15	6	PORT 3.13
7	PORT 2.14	8	PORT 3.15
9	PORT 2.13	10	PORT 3.6
11	PORT 2.12	12	PORT 3.7
13	PORT 2.11	14	PORT 3.8
15	PORT 2.10	16	PORT 3.9
17	PORT 2.9	18	TXD0, P3.10
19	PORT 2.8	20	PORT 3.5
21	PORT 2.7	22	PORT 3.4
23	PORT 2.6	24	PORT 3.3
25	PORT 2.5	26	PORT 3.2
27	PORT 2.4	28	PORT 3.1
29	PORT 2.3	30	PORT 3.0
31	PORT 2.2	32	PORT 8.6
33	PORT 2.1	34	PORT 8.7
35	PORT 2.0	36	/WRH
37	nc	38	nc

Table 23: Connector X8

8. Power-Supply

The EVA27x boards expect a stabilized 5 VDC ($\pm 0.25V$) voltage at the X2 power connector (see Figure 8). An external power mains unit is also part of the STart27x Evaluation Kit/Development Kit.

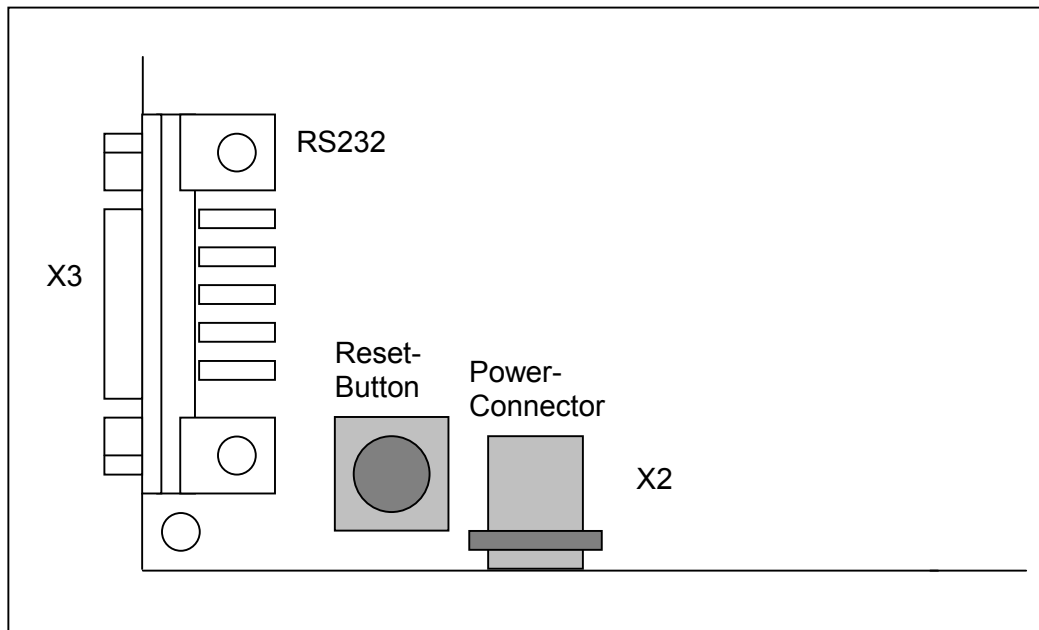


Figure 8: Power Supply

CAUTION:

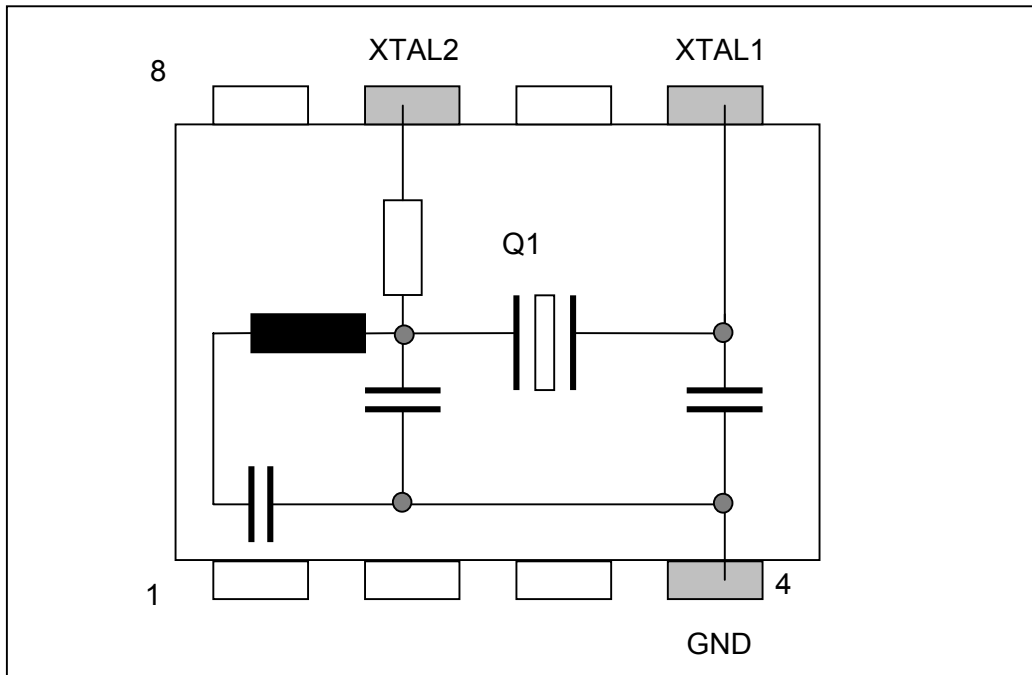
DON'T USE ANY OTHER POWER SUPPLY THAN THAT ONE COMING WITH THE START27x EVALUATION KIT. OTHERWISE YOU TOLERATE THE DAMAGE OF THE ENTIRE BORD !

IN THIS CASE WE CANNOT GRANT ANY WARRANTY!

9. User Oscillator

The oscillator unit (8MHz crystal) is already plugged into a DIP8 socket module (refer to the G2 component on top view schematics) when shipped. It is pretty easy to replace this component by your own, individual circuitry. Just remove the oscillator from the DIP socket and replace it by your discrete solution. This could be a small, 3-pin printed circuit board, which keeps all parts, to run the CPU through an external crystal.

Figure 9: External crystal



10. CD-ROM

Within the STart27x Kits you will find everything you need for a fast start-up development. Two C compilers (demo versions) coming from KEIL and TASKING allow you to write small sample programs or to run the examples provided on the CD-ROM. The use of the compiler is reduced in memory size. You will find datasheets of the devices on the CD-ROM as well as a utility to program and erase the embedded Flash memory.

10.1. Content of CD-ROM

On your CD, you can find four main directories. The content is described below. Please refer to the TXT files in the sub-folders for detailed information.

DC166	Tasking demo, board configuration for the monitor and examples.
Keil	Keil demo, monitor, examples.
Manual	Manual of the STart27x Kits and datasheets of the ST10F27x.
Tools	This directory contains tools for programming the external/internal Flash of the ST10F27x and the PLS demo debugger.

10.2. DEMO for KEIL

The example RTC27x can be built, using a KEIL C compiler (full or demo version). Within the project options you can choose between a Debug-version, a version running from the internal Flash and a version, running from the external SRAM/Flash. For this the following switches have been used:

- _KEIL, _DEBUG: Shows how to make a project for a debug session.
- _KEIL, _INTERN: Shows how to make a project for the internal Flash-memory of the ST10F27x.
- _KEIL: Shows how to make a project for the external Flash-memory.

10.3. DEMO for TASKING

The example RTC27x can be built, using a TASKING C compiler (full or demo version). Within the workspace you can choose between a Debug-version, a version running from the internal Flash and a version, running from the external Flash.

- RTC_deb: Shows how to make a project for a debug session.
- RTC_int: Shows how to make a project for the internal Flash-memory of the ST10F27x.
- RTC_rel: Shows how to make a project for the external Flash-memory.

10.4. FLASH166 / WINFLASH166

In this folder you will find the FLASH166 tool which has already been mentioned in this manual. Furthermore this directory keeps the technical documentation of the Flash tool, where you can find all needed information for debugging and downloading binaries into the Flash circuit. FLASH166 supports a lot of different Flash memory types.

Each sample folder for the release version includes a certain batch execution file (prog.bat) which makes it easy for you to create a binary file. FCONV is used to generate a binary output file, based on a hex file typically generated by all C-compilers. Since FLASH166 expects a binary formatted file you should use FCONV and FLASH166 to program the ST10F27x controller. You must not pay attention to these subjects if you are using any other programming tool.

We are permanently upgrading the FLASH166 tool. So we actually support a large number of different standard FLASH memories. If you are planning to use a Flash type which is not supported by FLASH166 yet, you should not hesitate to contact us.

Furthermore, if you are planning to use FLASH166 for your own application later on, please contact us by phone (+49 7667 908 122), or email (sales@fsforth.de) We will make you an interesting offer for the tool and additional licences.

In case of problems with the serial link under Windows XP or WIN2000, we provide the tool WINFLASH166 on the CD-ROM.

10.5. Board configuration

For each demonstration program refer to the associated text file, like readme.txt or abstract.txt. Herein you will find all technical details about the board configuration, CPU speed and other parameters. Also have a look at the PDF file, provided by STMicroelectronics, which represents the datasheet of the ST10F27x.

10.6. Error conditions FLASH166 / WINFLASH

Batch file does not run:	Make sure that you have adapted the batch file according to your individual path structure. You may be forced to edit the batch file slightly.
Wrong Flash number:	Please note that FLASH166 expects a Flash device at the address 0x00.0000 (/CS0). If you have changed the chip selection lines this error message occurs. The switch "/CSx" changes the chip select line.
Bootstrap mode timeout.	Make sure that you have connected a serial (RS232) cable to COM1 (COM2) and X3 of your board. Then press the RESET push button again. Check if the BSL switches are in bootstrap mode (BSL-Mode0=ON, BSL-Mode1=OFF). If you are using a serial interface different than COM1 you must revise the "prog.bat" file according to: "flash166 /P <file.bin> /COM2".

10.7. ST10Flasher

This tool allows programming the internal Flash memory of the ST10F27x controller. The tool is provided within the tool directory on CD-ROM. Just unzip the file and run Setup.exe to install the tool.

Before running the ST10Flasher, establish the RS232 connection between your PC and the EVA27x board, switch the board into BSL-mode (S3-5 = ON, S3-6 = OFF) and switch on the power supply, or press the Reset button S1.

Now you can run the ST10Flasher and select the Hex-file to download into the internal Flash:

```
"\STart27x\Keil\C166\Examples\Boards\FORTH_EVA27x\RTC27x\Intern\obj\RTC27x.h86"
```

After the programming was successful, switch S3-5 OFF and plug on the jumper JP11. Then reset the controller by the Reset push button S1.

10.8. PLS UDE

This Debugger allows also programming the internal Flash memory of the ST10F27x controller. The tool is provided within the tool directory on CD-ROM.

Take notice to run "SETUP.BAT" to install the tool!

Before running UDE, establish the RS232 connection between your PC and the EVA27x board, switch the board into BSL-mode (S3-5 = ON, S3-6 = OFF) and switch on the power supply, or press the Reset button S1.

Now you can run UDE, open the workplace EVA276.WSP of your board and select the Hex-file to download into the internal Flash:

```
"\STart27x\Keil\C166\Examples\Boards\FORTH_EVA27x\RTC27x\Intern\obj\RTC27x.h86"
```

After the programming was successful, switch S3-5 OFF and plug on the jumper JP11. Then reset the controller by the Reset push button S1.

11. Building the sample project with the TASKING C-Compiler

First of all you have to install the Tasking C166 Compiler:

Copy the file DC166-801.zip to a free directory on your harddisk and unzip this file. Then run the setup.exe to install the Tasking Compiler. We recommend to use the default directories and to copy the files from the \TOOLS\FLASH166-directory of the CD-ROM into a directory on your Hard disk, which is known by your system.

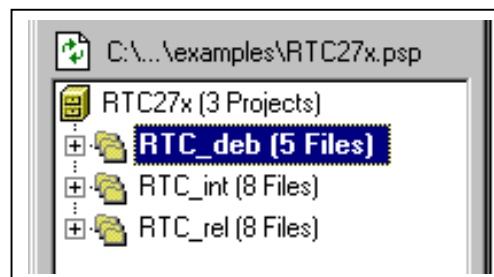
Now copy the files from the structure \DC166\... from the CD-ROM to the corresponding directories on your harddisk, remove the "Read-only"-attribute from all of these files.

Then run the Tasking Compiler and select from the menu:

Project -> Project Space -> Open

From the next menu select the file "RTC27X.PSP" from the directory \DC166\EXAMPLES\.

Now you will see the project space from the RTC27X-sample project:



Go to the menu Build and select the item "Scan all dependencies"

If you have the license for the full version of EDE, you can remove the macro _TASKING_DEMO from the preprocessing project options of the C-compiler options.

11.1. Project for the debugger

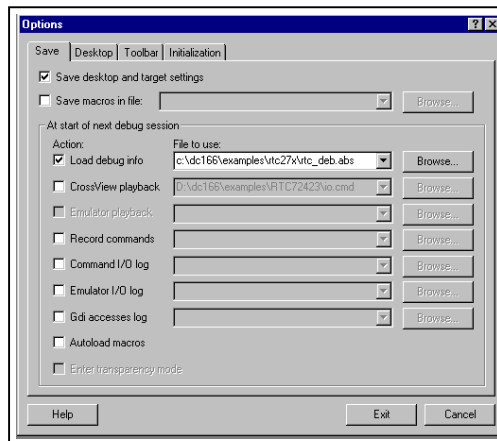
To build the project for the debugger, choose the project “RTC_deb” as current project and “Rebuild” the project. The results can be found in the RTC27X-directory.

Connect the delivered cable for the RS232-interface between your PC and the connector X3 on the EVA27x-board. Switch the BSL-switch S3-5 to ON, select CS1 for SRAM and switch on the power supply of the board.

Now you can start the CrossView debugger by choosing the item “Debug” at the menu “Project”. After the correct download of the project you will see the environment of the debugger. Here you can start the application by choosing the item “Run” from the menu “Run”.

Now you will see the output of the target in the Terminal window.

If you will get an error message from the CrossView debugger, belonging to a directory, you have to quit the debugger and enter the correct drive letter into the appearing window:



11.2. Project for the external FLASH

To build the project, running from the external FLASH, select the project "RTC_rel" as current project and "Rebuild" the project. The results can be found in the RTC27X-directory.

Connect the delivered cable for the RS232-interface between your PC and the connector X3 on the EVA27x-board. Switch the BSL-switch S3-5 to ON and switch on the power supply of the board.

Start the batch-file "PROG.BAT", which converts the hex-file of the project into a binary file, which will be downloaded into the external FLASH, by the FLASH166 program.

Start the batch-file "PROGW.BAT", which converts the hex-file of the project into a binary file, which will be downloaded into the external FLASH, by the WINFLASH166 program.

After the correct download, you have to switch OFF the BSL-switch S3-5, press the Reset push button S1 and start a terminal program on your PC with the line parameters 9600Baud, no parity, 8 bit data and 1 stop bit.

At the terminal program you will receive the output from the EVA27x-board.

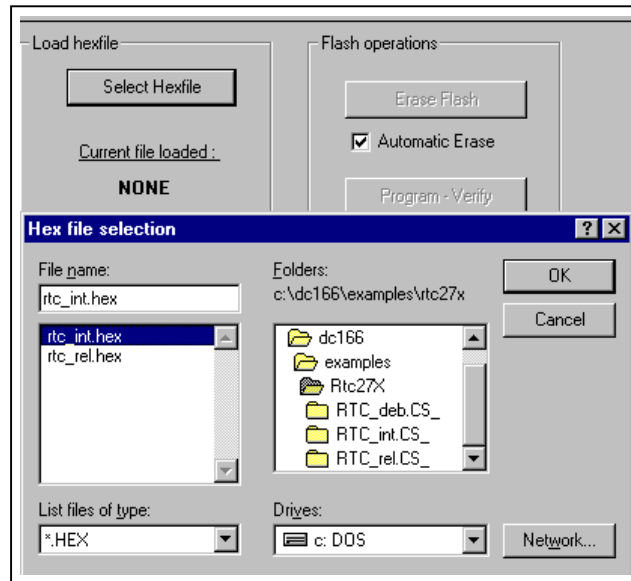
This project will only run at the development board.

11.3. Project for the internal Flash with ST10Flasher

To build the project, running from the internal Flash, select the project “RTC_int” as current project and “Rebuild” the project.

Connect the delivered cable for the RS232-interface between your PC and the connector X3 on the EVA27x-board. Switch the BSL-switch S3-5 to ON and switch on the power supply of the board.

Run the software ST10Flasher from STMicroelectronics and load the hex-file “RTC_INT.HEX” from the RTC27X-directory:



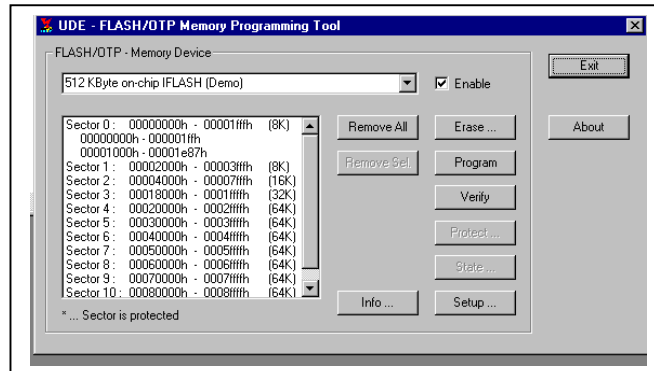
Now select the Flash operation Program – Verify, to download the project into the internal Flash. After a successful download, you have to switch OFF the BSL-switch S3-5, enable the single chip mode by plugging on the jumper JP11, press the Reset push button S1 and start a terminal program on your PC with the line parameters 9600Baud, no parity, 8 bit data and 1 stop bit.

11.4. Project for the internal Flash with UDE

To build the project, running from the internal Flash, select the project “RTC_int” as current project and “Rebuild” the project.

Connect the delivered cable for the RS232-interface between your PC and the connector X3 on the EVA27x-board. Switch the BSL-switch S3-5 to ON, enable the single chip mode by plugging on jumper JP11 and switch on the power supply of the board.

Run the software UDE from PLS and load the hex-file “RTC_INT.OUT” from the RTC27X-directory:



Now select the Flash operation Program, to download the project into the internal Flash. After a successful download, you have to switch OFF the BSL-switch S3-5, press the Reset push button S1 and start a terminal program on your PC with the line parameters 9600Baud, no parity, 8 bit data and 1 stop bit.

12. Building the sample project with the KEIL C-Compiler

First of all you have to install the Keil C166 Compiler:

Copy the file ek166v427.exe to a free directory on your harddisk and run it to install the Keil Compiler. We recommend to use the default directories and to copy the files from the \TOOLS\FLASH166-directory of the CD-ROM into a directory on your Hard disk, which is known by your system.

Now copy the files from the structure \Keil\... from the CD-ROM to the corresponding directories on your harddisk, remove the "Read-only"-attribute from all of these files.

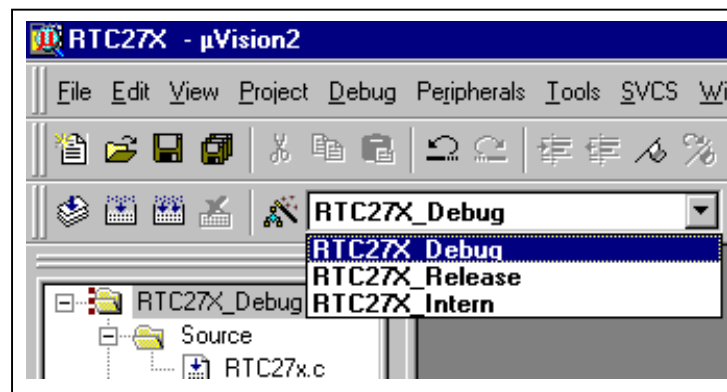
Before starting to work with the debugger, you have to choose the correct monitor for the debug session:

Run the Keil Compiler and open the project "MONITOR.UV2" from the directory: "\Keil\C166\MONITOR\FORTH_EVA27x\"

Choose the target "Bootstrap" and rebuild the project. Now you have built the Monitor for the EVA27x-board and you can leave this project

Open the project "RTC27X.UV2" from the directory: "\Keil\C166\EXAMPLES\BOARDS\FORTH_EVA27x\RTC27X\"

Select RTC27X_Debug as target.



12.1. Project for the debugger

To build the project for the debugger, select "RTC27X_Debug" as target and "Rebuild" the project. The results can be found in the directories:

\\RTC27X\\Debug\\List and \\RTC27X\\Debug\\Obj

Connect the delivered cable for the RS232-interface between your PC and the connector X3 on the EVA27x-board. Switch the BSL-switch S3-5 to ON, select CS1 for SRAM and switch on the power supply of the board.

To start the debugger, Select Debug -> Start/Stop Debug Session. After the correct download of the project you will see the environment of the debugger. Here you can start the application by choosing the item Go from the menu Debug.

After you have started the application, you will see the output from the target in the serial #1-window.

12.2. Project for the external FLASH

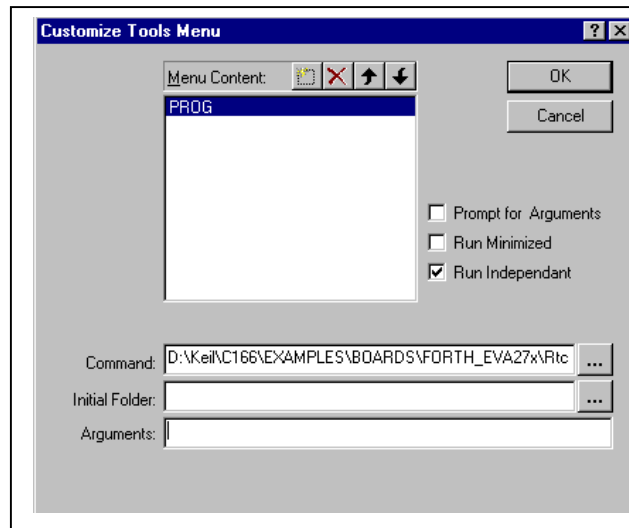
To build the project, running from the external FLASH, select "RTC27X_Release" as target and "Rebuild" the project. The results can be found in the directories:

\\RTC27X\\Release\\List and \\RTC27X\\Release\\Obj

Connect the delivered cable for the RS232-interface between your PC and the connector X3 on the EVA27x-board. Switch the BSL-switch S3-5 to ON and switch on the power supply of the board.

To make the project ready to download you can either open a DOS-box to start the batch-file "PROG.BAT" in your project-directory, or you can enter the call of the batch-file "PROG.BAT" at "Customize Tools Menu" from the menu Tools:

Don't forget to select the option "Run independant" !



The batch-file "PROG.BAT" converts the hex-file of the project into a binary file, which will be downloaded into the external FLASH, by the FLASH166 program.

The batch-file "PROGW.BAT" converts the hex-file of the project into a binary file, which will be downloaded into the external FLASH, by the WINFLASH166 program.

After the correct download, you have to switch OFF the BSL-switch S3-5, press the Reset push button S1 and start a terminal program on your PC with the line parameters 9600Baud, no parity, 8 bit data and 1 stop bit.

At the terminal program you will receive the output from the EVA27x-board.

This project will only run at the development board.

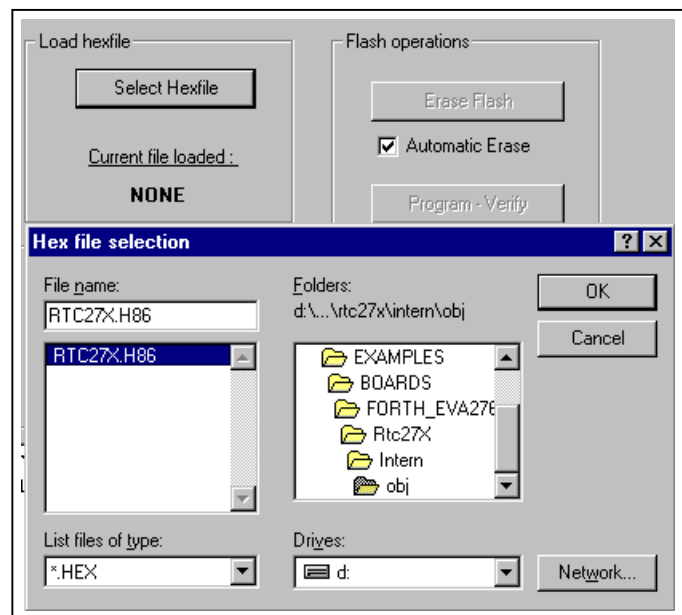
12.3. Project for the internal Flash with ST10Flasher

To build the project, running from the internal Flash, select “RTC27X_Intern” as target and “Rebuild” the project. The results can be found in the directories:

\\RTC27X\\Intern\\List and \\RTC27X\\Intern\\Obj

Connect the delivered cable for the RS232-interface between your PC and the connector X3 on the EVA27x-board. Switch the BSL-switch S3-5 to ON and switch on the power supply of the board.

Run the software ST10Flasher from STMicroelectronics and load the hex-file “RTC27X.H86” from the Obj-directory:



Now select the Flash operation Program – Verify, to download the project into the internal Flash. After a successful download, you have to switch OFF the BSL-switch S3-5, enable the single chip mode by plugging on jumper JP11, press the Reset push button S1 and start a terminal program on your PC with the line parameters 9600Baud, no parity, 8 bit data and 1 stop bit.

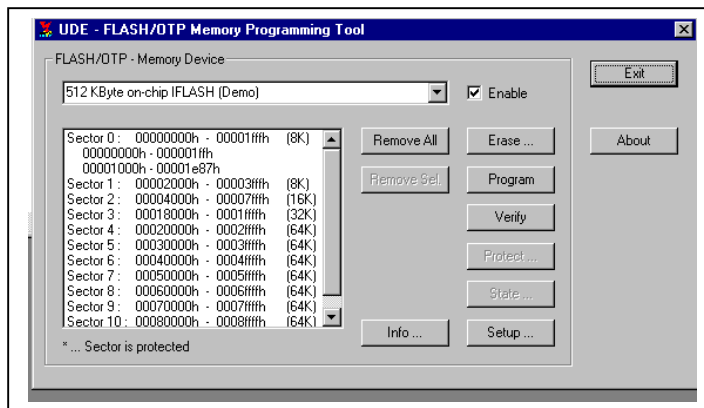
12.4. Project for the internal Flash with UDE

To build the project, running from the internal Flash, select “RTC27X_Intern” as target and “Rebuild” the project. The results can be found in the directory:

\\RTC27X\\Intern\\List and \\RTC27X\\Intern\\Obj

Connect the delivered cable for the RS232-interface between your PC and the connector X3 on the EVA27x-board. Switch the BSL-switch S3-5 to ON, enable the single chip mode by plugging on jumper JP11 and switch on the power supply of the board.

Run the software UDE from PLS and load the hex-file “RTC27X.H86” from the Obj-directory:



Now select the Flash operation Program, to download the project into the internal Flash. After a successful download, you have to switch OFF the BSL-switch S3-5, press the Reset push button S1 and start a terminal program on your PC with the line parameters 9600Baud, no parity, 8 bit data and 1 stop bit.

13. Know-how for handling ST10F27x IDEs

There is no general way how to handle IDEs to get fast and good results but we can give some useful hints to avoid mistakes.

There are a few differences in handling projects for debugging the application and for making a release version for the ROM (Flash).

For a debug session, you need a RAM or ROM monitor. This is a small program which initializes registers and provides functions in order to make your hardware able to communicate with a debugger, usually via serial line. The most important point is, that the registers for RAM memory are set to a correct value by the monitor, because the remote debugger loads your application into RAM and has to modify it for setting breakpoints. The configuration for the Keil monitor can be found in the project MONITOR.UV2 and the configuration for the Tasking monitor can be found, when opening the file \DC166\ETC\EVA27x.CFG

A RAM monitor is loaded via bootstrap loader, the ROM monitor has to be programmed into the ROM (Flash) and is automatically activated after RESET. This Evaluation Kit works mainly with RAM monitors because the download time is so short that a direct start from ROM yields no advantages.

Please make sure that your application doesn't overwrite the monitor in the RAM!

For a release session, you have to link a "start-up" to your application, which is automatically started after RESET. The start-up is usually an assembler program, linked to your application (Keil example **RTC27X_Release**: Start276.a66), or is automatically generated via the project settings (Tasking example **RTC_rel**). The memory settings for linker/locator has to match with the start-up settings! I.e., if you set the address selection register for RAM to 100 000h and locate your data to 10 000h your program won't run.

Generally, the start-up initializes your hardware. Please check the settings for memory access times for RAM and ROM (Flash) to avoid your hardware running slower than it could. **If you have any questions, please don't hesitate to contact the technical support of FS FORTH-SYSTEME!**

14. Appendix

14.1. Top View of the EVA27x boards

