STart168

Evaluation Kit for

ST10F168 Microcontroller

FORTH-SYSTEME GmbH

P.O. Box 11 03 ● D-79200 Breisach, Germany
Kueferstrasse 8 ● D-79206 Breisach, Germany
☎+49 (7667) 908-0 ● Fax +49 (7667) 908-200 ● e-mail: sales@fsforth.de

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

Release of Document:	October 30, 2000
Filename:	STRT168d.DOC
Author:	Jochen Florath

Table of Contents

1. Introduction

The product line of FS FORTH-SYSTEME got a brand new member of user-friendly, easy-to-use and powerful evaluation kits which is offered as ”STart168”. It is shipped with a ST10F168 controller which provides a 256 KByte on-chip Flash memory, a 6 KByte on-chip extension RAM (XRAM), a 2 KByte internal SRAM (IRAM) a CAN controller which works according to ISO11898 (profile 2.0B) and with many additional I/O features. The board supports a up to 1 MByte external Flash memory and 256 KByte Fast-SRAM which allow to build a large number of sophisticated applications.

The ST10F168 is a derivative component based on an Infineon C167 architecture but with many additional features. Please refer to the ST10F168 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. An intelligent battery management supports the assembly of an optional, low power 256 KByte SRAM working as non-volatile memory. Furthermore a set of user selectable switches (DIP) offer a very flexible handling of the CPU specific start-up configuration. Therefore the STart168 Evaluation Kit promises a successful start-up development for beginners and experienced users. The EVA168 Evaluation Board 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 this evaluation board to fulfill a lot of actual and future user requirements. Therefore this kit has been designed to allow the assembly of ST10 controllers with 2 CAN modules. Thus the board comes with two pre-assembled CAN transceivers. Both can easily be used without the need of additional hardware or a redesign of the board’s printed circuit. Furthermore the external memories have been chosen to run the board with a high speed ST10 / 25 MHz controller. For fast prototyping the EVA kit includes a small wire wrap field, a potentiometer for an analog input signal, serial EEPROM, a set of LEDs and a push button to make development easy and simple. Please refer to the following chapters to get into more details about the ST10F168, the board’s capabilities and the tools which are coming with the STart168 Evaluation Kit.

2. Features of STart168 Evaluation Kit

- Evaluation Kit based on the EVA168 Evaluation Board
- 256 kByte high speed SRAM
- 512 kByte external Flash
- 256 KByte, low powered SRAM for optional memory expansion
- Optional battery back-up circuit for low powered SRAM (see BQ2201SNN)
- Additional 512 kByte Flash for optional program memory extension
- Two CAN transceiver (PCA82C250T, ±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 EPROM (24LC02, I2C-Interface) for user purposes
- Including a RTC-72423 realtime clock
- The EVA168 Evaluation Board is designed for running a ST10F168 at 25 MHz and future controllers with two integrated CAN modules, according to ISO11898 (2.0B)
- Power supply 90..264VAC/5VDC enclosed. Input voltage to the EVA168 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
- Flash programming utility for the internal CPU Flash memory provided by Lauterbach
- Stand alone flash utility to program the embedded on-chip Flash of the ST10F168.
- C compiler (demo version only) provided by TASKING and KEIL
- Including a socket to carry a four pin crystal oscillator device
- Reserved pads to hold an optional ”Quad-Connector” for Lauterbach emulator systems.

3. Getting started with EVA168 Evaluation Board

As already mentioned above, this paragraph intents 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 cerate a project path which will keep the demonstration software, as well as the encapsulated bootstrap and monitor program, both provided on CD-ROM. 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 uploaded to any memory location. For more details please refer to chapter 5.1 ”Memory-Configuration”. By the way: the user may select both, the external FLASH or the external SRAM memory to downloaded his individual application software.

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

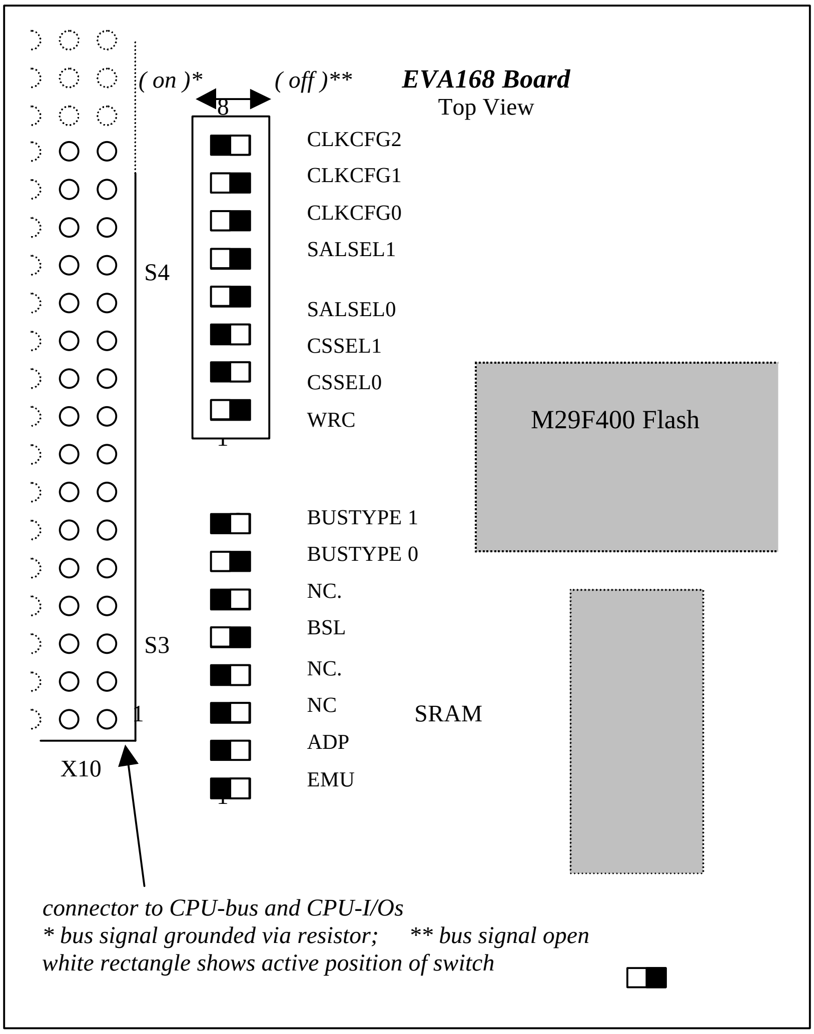
- Setup the board's DIP switches as shown in figure 1 (the board is pre-configured by FS FORTH-SYSTEME according to figure 1). Refer to chapter 6 ”System Start-up Configuration” for detailed information about these settings.

- Make sure that the BSL switch is set to left position (S4-4=ON / P0L.4=low). This will force the CPU to go into bootstrap mode right after reset or power on. Bootstrap mode is always entered by the CPU whenever pin P0L.4 has been grounded prior to a reset.

- Connect the board's serial RS232 interface 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 a sample program from the attached CD-ROM into your project path. For example take ”....\EXAMPLES\RTC168_Release\RTC.BIN” which will show you how the realtime clock works on the EVA168 evaluation board.

Figure 1: Factory setting



Example continued :

- 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.

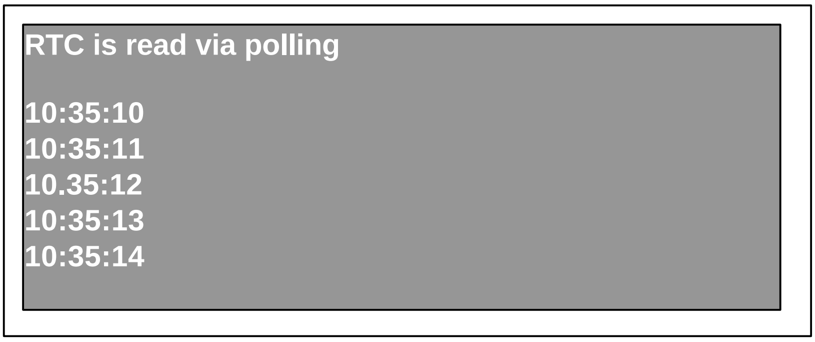
- Make sure that the auto-execution path does include a reference to the new path directory from where to load ”FLASH166”.

- Now start the download:
  Type ”FLASH166 /P EPROM.BIN” at the DOS prompted line and press <RETURN> key. The bootstrap loader tries to connect to COM1 by default. If this does not work properly try to connect to COM2.

- *Type ”FLASH166 /P RTC.BIN /COM2” and see what happens. If you still have problems to start downloading, please revise the settings of S4 once more. Check if you have chosen the RS232 cable delivered with the Start168 Evaluation Kit.*

- Now move the BSL switch to the right side (opened) and press the reset button S1.

Figure 2: Output of the example RTC168



```
RTC is read via polling

10:35:10
10:35:11
10.35:12
10:35:13
10:35:14
```

If everything is working properly the measured realtime of the RTC-72421 device will appear on your PC screen as shown in figure 2.

4. Software Tools for EVA168 Evaluation Board

The STart168 Evaluation Kit always includes the bootstrap loader and monitor program (FLASH166) from FS FORTH-SYSTEME. That’s everything you need to run your individual software on a ST10F168 controller. This software is a part of the STart168 Evaluation Kit.

FLASH166 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 more things

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 ST10F168 as a standalone controller you may use the Flash utilities which are designed and provided by Lauterbach. It allows a fast download of your program into the internal memory section of the CPU. A second stand alone utility is provided on the CD allowing programming/erasing of the on-chip Flash.

5. EVA168 Memory-Mapping

The EVA168 evaluation board supports a number of memory architectures according to the capability of the ST10F168 controller.

The board is marketed by FS FORTH-SYSTEME with the following components:

Table 1: Memory

	Memory	Type	Chip-Select	Size	Comment
1	FLASH, Boot	M29F400BB90N1 or M29F800BB90N1	CSFLSHA	256k * 16 or 512k * 16	assembled or optional
2	FLASH, Extension	M29F400BB90N1 or M29F800BB90N1	CSFLSHB	256k * 16 or 512k * 16	optional
3	SRAM, High Speed	2 * µPD431008LE or 2 *µPD434008LE	CSRAM	2 * 128k * 8 or 2 * 512k * 8	assembled or optional
4	SRAM, Low Power	2 * M5M51008AFP or 2 * M5M5408LE	CSRAM_B	2 * 128k * 8 or 2 * 512k * 8	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 (both Flash parts must be a 29F800).

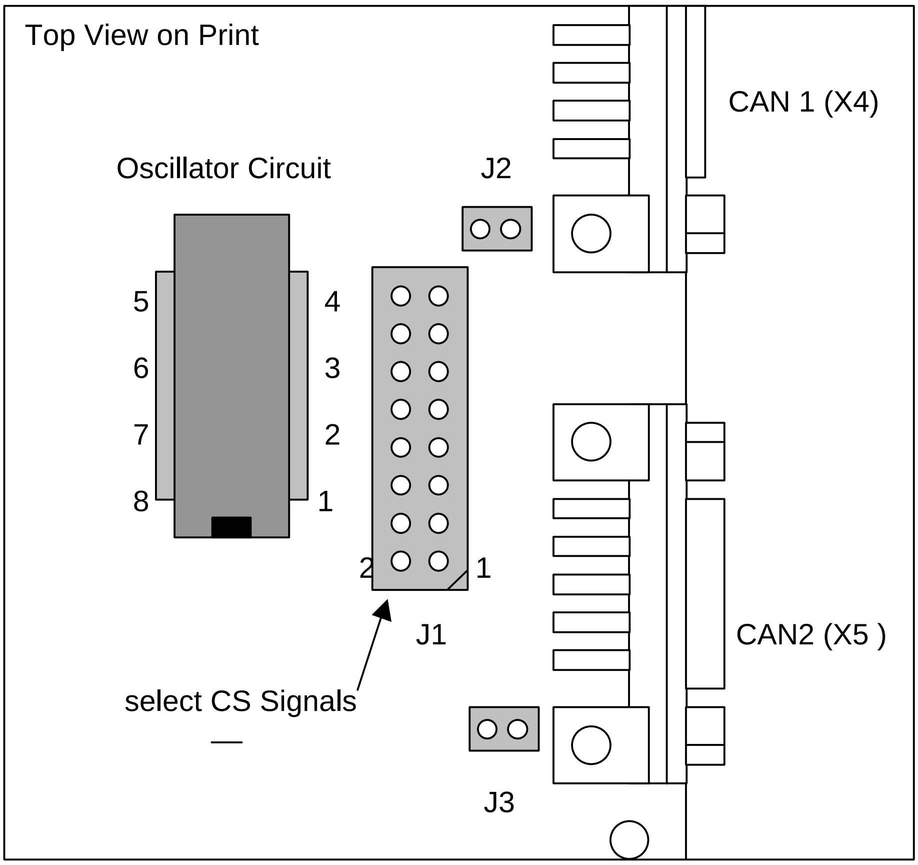
The optional static RAM (line 4 in table 1) can be assembled as an alternative memory block which provides lower power characteristics. This memory is interfaced to an external battery back-up and supervisory circuit which prevents loss of data. The circuit with the part number U6 (BQ2201) has to be added to the printed board if you want to use the low power RAM block with a maximum size of 1 Mbyte.

This circuit generates the /CSRAM_B signal which is derived from the origin CSRAM signal controlled by the CPU’s internal address logic. If power fails this integrated circuit takes care about the power level at /CSRAM_B. To prevent damage of the SRAM block both, the supply voltage and CS_RAM_B must be stable (3V) in case of a power fail situation.

Since the selection signal CSRAM_B is derived from the CSRAM signal both memory blocks (3 or 4) may never work at the same time. If you want to have both memory blocks working together you may connect the last chip select signal (/CS4) from the CPU’s address logic. This signal is not used yet and can be interfaced to the BQ2201 through a short wire.

5.1. Memory-Configuration

Figure 3: Chip Select selection



5.2. Settings of Jumper J 1

Table 2:

PIN	A	B	C	Chip-Selects	Pre-configured
1 2	▌			CS0 -> CSFLSHA	Yes
3 4	▌			CS1 - > CSRAM	Yes
5 6		▌		CS0 -> CSRAM	No
7 8		▌		CS1 - > CSFLSHA	No
9 10			▌	CS2 - > CSRTC	Yes
11 12			▌	CS3 -> CSFLSHB	No

▌ *= connected pins*

To make sure that your system works properly you may combine the setting shown in column (A) with column (C) provided that the system boots from an external Flash device. Otherwise if the program resides within an external SRAM you should select column (B) in combination with column (C). The jumper setting (C) works independent of the other selections. Open jumper J1(9)-J1(10) (see figure above) for disabling the chip-select signal on the RTC unit. Since the board comes without the second Flash device J1(11)-J1(12) are removed. 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 ST10F168 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 are active on the PORT0 lines which results into a high level state, if the respective pin is left open. Otherwise the pin must be pulled down by an external resistor (4.7k..8.2k) if the user wants to select any other CPU option at this certain pin.

The following paragraphs explain how to set all of these start-up configuration pins. Please refer to chapter 2 figure 1 to find out the corresponding switches located at S3 or S4. If a bit has to be configured as 0 move the switch to the left position (see 0 in figure 1). A logical 1 appears when the switch is set to the right location (see 1 in figure 1).

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 is in general 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 ST10F168 goes into a passive, floating state, which is similar to its state during reset. This mode allows switching a ST10 that is mounted on board virtually off, so an emulator may control the circuitry since the pins are floating even if the original controller remains in its place. This mode is used for special emulator purposes and is generally not used in basic applications.

Default: Emulation Mode Off (P0L.1=1 / S3-2=off)

6.3. Bootstrap Mode

PIN BSL, S3-5

Pin P0L.4 activates the on-chip bootstrap loader when switch S3-5 is on during reset. The bootstrap loader allows moving a tiny start code (about 32 Byte) into the internal RAM of the ST10F168 via the serial controller interface ASC0 (asynchronous, serial mode). The controller remains in BSL mode until a hardware reset with P0L.4 set high occurs. Bootstrap loader mode may also be exited through a software reset instruction. In this case P0L.4 is ignored and program execution starts fetching code from 0x00000 (internal or external). Refer to the user manual for more details.

Default: Bootstrap Mode On (P0L.4=0 / S3-5=on)

6.4. Bus-Access-Types

PIN BUSTYPE0, S3(7)
PIN BUSTYPE1, S3(8)

Pins P0L.7 (BUSTYPE1) and P0L.6 (BUSTYPE0)

Table 3: 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; Bus Mode = Demultiplexed

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, i.e. /WRH and /WRL.

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

6.6. Chip-Select-Lines

PIN CSSEL0, S4-2
PIN CSSEL1, S4-3

Defines the number of active chip select lines after reset. This allows the selection which pins of port 6 drive external chip select signals and which may be used for general purpose I/O lines.

Table 4: Chip Select Selection

S4-3	**S4-2**	**active chip select**	**Note**
off	off	/CS0.../CS4	Presetting of EVA168
off	on	none	Port 6 Pins free for I/O
on	off	/CS0.../CS1	
on	on	/CS0.../CS2	

6.7. Segment Address Lines

PIN SALSEL0, S4-4
PIN SALSEL1, S4-5

During reset this 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.

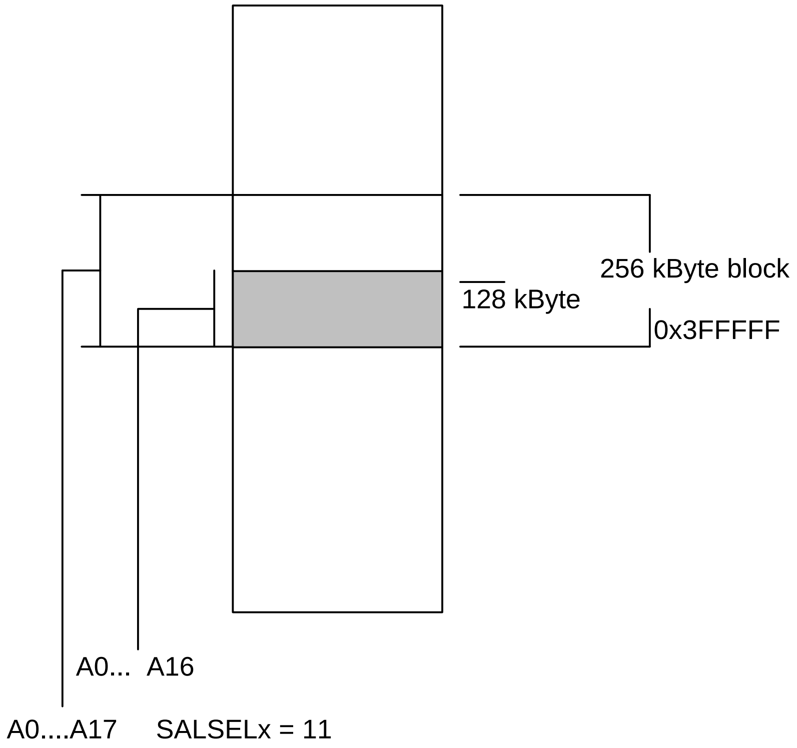
Table 6: Segment Address Lines Selection

S4-5	S4-4	Segment Address Lines	Directly addressable space
off	off	A17.....A16	256 kByte
off	on	A23.....A16	16 Mbyte, CAN module disabled !
off	on	A20.....A16	2 Mbyte, CAN 1 module enabled
on	off	None	64 kByte
on	on	A19.....A16	1 Mbyte; Default state of EVA168

Example:

Assume that your application requires the 128 kByte SRAM block at the absolute address 0x3FFFFF. 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 (SALSELx = 11).

Example cont'd



Define BUSCON1 Register:

CSWEN = CSREN = 1, RDYEN1 = 0; BUSACT1 = 1
ALECTL1 = 0; BTYP = 10; etc....

Define ADDRSEL1 Register

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

6.8. RANGE START ADDRESS

Depending on the chosen block size (see previous example) the range start address 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:

0x3FFFFF = 0011.1111.1111.1111.1111.1111

*

RGSAD = 0011.111*0.0000* **

ADRSEL1 = 0011.1110.0000.0101

* controlled by bus address lines A0..A16
**cursive bits are discarded since they are already controlled by address bus

6.9. EVA168 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 register and the configuration registers as already described in a previous chapter to setup your individual memory layout.

Table 7: Typical Memory Layout

Address Range	Chip-Select	Bus Width	Function
0x4F.FFFF 0x40.0000	/CS4	?	Not connected yet
0x3F.FFFF 0x30.0000	/CS3	16 bit	Flash Extension, up to1 MByte
0x20.0FFF 0x20.0000	/CS2	8 bit	RTC; 4 kByte
0x1F.FFFF 0x10.0000	/CS1	16 bit	SRAM, up to 1 MByte
0x0F.FFFF 0x00.0000	/CS0	16 bit	Boot-Flash, up to 1 MByte

6.10. Clock Generation

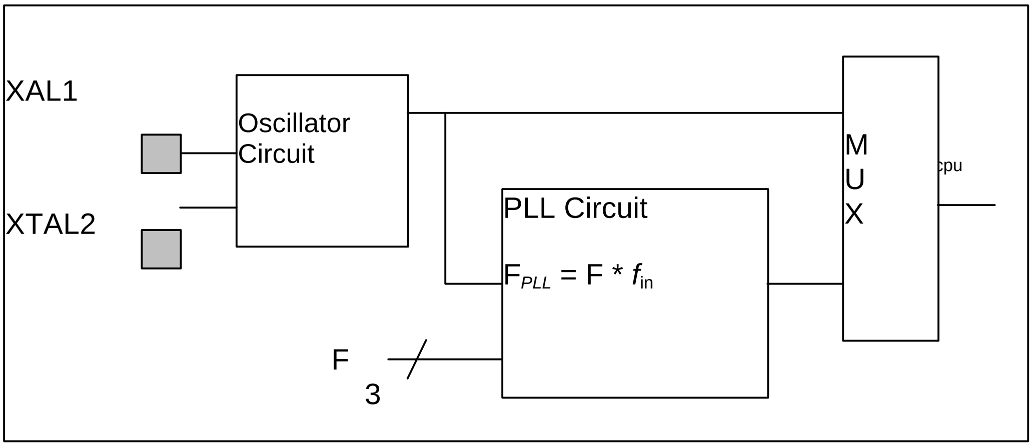
PIN CLKCFG0, S4-6
PIN CLKCFG1, S4-7
PIN CLKCFG2, S4-8

Since the ST10F168 includes an internal PLL circuit, the external oscillator device does not need to be driven by a high frequency crystal (Quartz). The external oscillator output either directly feeds the CPU and peripherals or it is connected to the on-chip PLL circuit which then provides the CPU clock.

Table 8: PLL selection

S4-8	S4-7	S4-6	CPU-Clock Fcpu = fxtal * F	External Clock Input Range in MHz	PLL mode
off	off	off	fxtal * 4	2.5 to 6.25	active
off	off	on	fxtal * 3	3.33 to 8.33	active
off	on	off	fxtal * 2	5.0 to 12.5	active
off	on	on	fxtal * 5	2.0 to 5.0	active (*)
on	X	X	fxtal * 1	1.0 to 25	inactive

(*) The EVA168 Evaluation Board comes with an external 5 MHz oscillator circuit. The CLKCFG pins are set by default to a factor of 5 (CPU internal = 25 MHz !).



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 EVA168 Evaluation Board.

Single chip mode is entered when pin /EA is set high during reset. Switch S2-3 to ON position before restarting the evaluation board. If the switch is set OFF a pull-down resistor configures the CPU to run in external bus mode. Refer to figure 4 on page 25.

6.12. User Switches and analog Components

The EVA168 Evaluation Board includes a wire-wrap field, four LEDs and analog components for user specific purposes like self-training and trials. Most of them are directly wired with connector X10 and X1 whereby this 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 cords 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 EVA168 Evaluation Board.

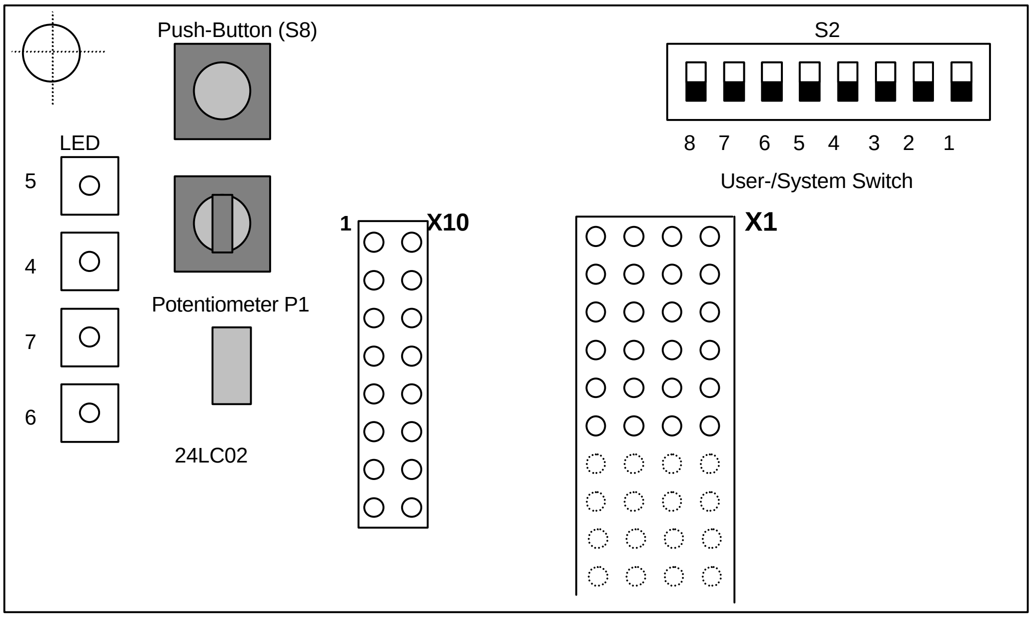


Figure 4.

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

6.13. Switch S2

PIN	Function description	X1	X10
1	If **ON** the VPP programming voltage is controlled by P2.0 Switch S2.2 **OFF to prevent a short circuit**	16c	---
2	If **ON** the VPP programming voltage is automatically controlled by the reset signal RSTIN. Switch S2.1 **OFF to prevent a short circuit**	11a	---
3	If **ON** the /EA pin of the CPU is set to +5V starting the device in single chip mode. A pull down resistor terminates the /EA pin at the CPU part if switch is off !	12d	
4	If **ON** disable oscillator Watchdog	13d	
5	USER-Switch 1: If **ON** pulls down signal at X10.1 otherwise the pin becomes high active (5V)		X10.1
6	USER-Switch 2: If **ON** pulls down signal at X10.2 otherwise the pin becomes high active (5V)		X10.2
7	USER-Switch 3: If **ON** pulls down signal at X10.3 otherwise the pin becomes high active (5V)		X10.3
8	USER-Switch 4: If **ON** pulls down signal at X10.4 otherwise the pin becomes high active (5V)		X10.4

Table 9.

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**
5	6
4	5
7	8
6	7

Table 10.

6.15. LED 1, 2, 3

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

LED	Name	Function
1	POWER	Power On/Off signal
2	VPP	On if programming voltage VPP is switched on
3	EINIT	Off after Reset, On after execution of EINIT

Table 11

6.16. Push button

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

6.17. Potentiometer

This variable resistor allows 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.

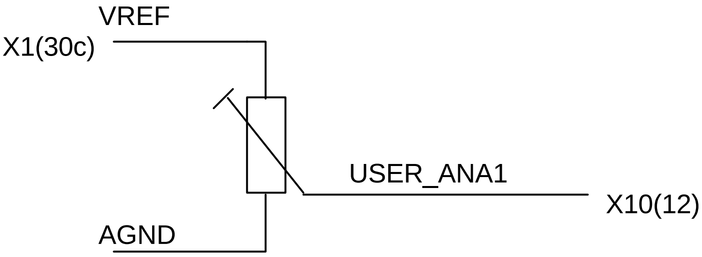


Figure 5.

6.18. 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.

You can use microcontroller with a second CAN interface. For that the user may wish to interface the controller CAN signal lines to the second CAN connector which is already a part of the EVA168 Evaluation Board!

6.19. RS232 Interface

The RS232 is interfaced to the board through connector X3 (see figure 6.0 on next page). It allows the download of user 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.

6.20. Battery Back-up

A Lithium Battery (3V) is also included on the EVA168 Evaluation Board when shipped to our customers. It supplies both low-powered SRAMs, provided they exist on board (not mounted on EVA168). According to the position of jumper J4 the battery can be removed from supplying the SRAMs.

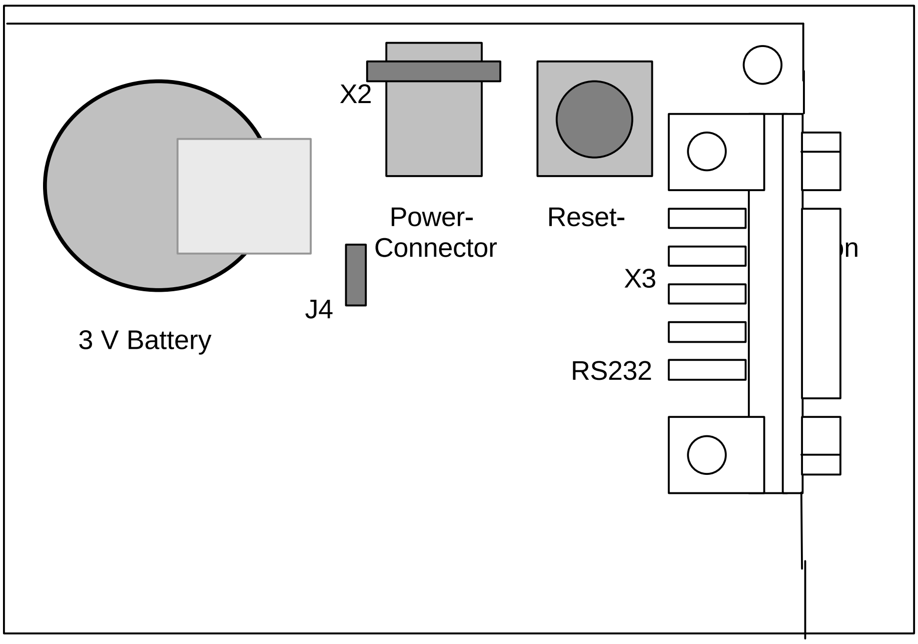


Figure 6.

7. Connectors

7.1. CPU signal connector X1

Row	PIN A	PIN B	PIN C	PIN 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	/RESTIN	/INT_RTC	/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	VPP
31	PORT 5.12	PORT 5.13	GND	GND
32	PORT 5.14	PORT 5.15	VDD (+5V)	VDD (+5V)

7.2. User hardware connector X10

PIN	FUNCTION
1	User Switch 1
2	User Switch 2
3	User Switch 3
4	User Switch 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
12	Analog Voltage from potentiometer
13	NC
14	NC
15	NC
16	NC

7.3. X4, CAN Connector 1

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	VCC (5V), optional

7.4. X5, CAN Connector 2

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	VCC (5V), optional

Pin 2 and 7 may be connected via a terminating resistor (120 Ω) via jumpers J2 and J3. Refer to chapter 6.18 ”CAN Bus-Termination”

7.5. 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

The housings of all DSUB connectors are grounded.

8. Power-Supply

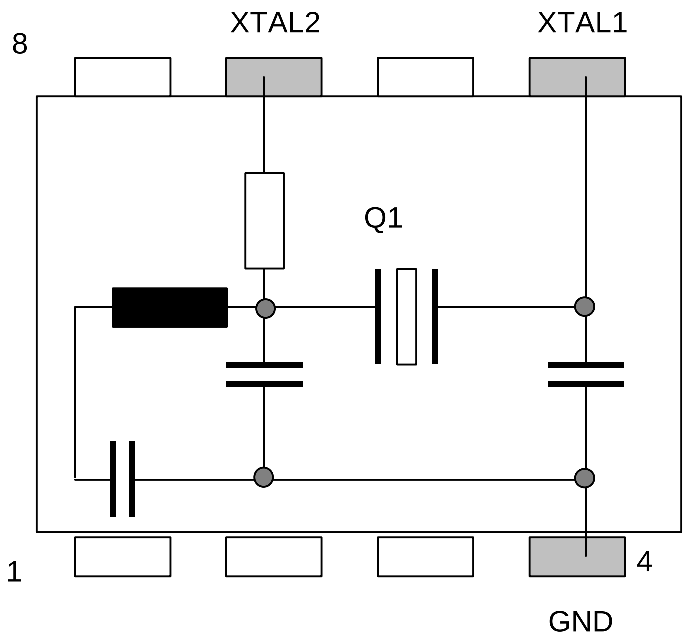
The EVA168 Evaluation Board expects a stabilized 5 VDC (±0.25V) voltage at the X2 power connector. An external power mains unit is also part of the STart168 Evaluation Kit.

CAUTION:
DON'T USE ANY OTHER POWER SUPPLY THAN THAT ONE COMING WITH THE START168 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 (5MHz) is already plugged into an 8 DIP socket module (refer to the G2 part on top view schematics) when shipped to our customers. It is pretty easy to replace this part 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, which keeps everything to run the CPU through an external crystal.



10. CD-ROM

Within the STart168 Evaluation Kit 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 life span (30 days) and 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 more detailed description.

DC166 This directory contains the Tasking demo, board configuration for the monitor and examples.

Keil Keil demo, monitor, examples.

Manual Manual of this Evaluation Kit and datasheets for the ST10F168.

Tools This directory contains tools for programming the external/internal Flash of the F168 and the PLS demo debugger.

10.2. DEMO\TASKING

Examples which can be build using a TASKING C compiler (full or demo version).

RTC168_Debug:	Shows how the realtime clock works and how to make a project for a debug session.
RTC168_Release:	Shows how the realtime clock works and how to make a project for the on-board Flash.

10.3. DEMO\KEIL

Examples which can be build using a KEIL C compiler (full or demo version).

RTC168_Debug:	Shows how the realtime clock works and how to make a project for debug session.
RTC168_Release:	Shows how the realtime clock works and how to make a project for the on-board Flash.

10.4. TOOLS\FLASH

In this folder you will find the FLASH166 tool which have already been mentioned in this manual. Furthermore this directory also keeps the technical documentation of the Flash tool where you can find all information you need for debugging and downloading binary dumps 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 (doit.bat or prog.bat) which makes it easy for you to create a binary dump. 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 ST10F168 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.

10.5. Board configuration

For each demonstration program refer to the associated text file, like rtc.txt or abstract.txt. Herein you find all technical details about the board configuration, CPU speed and other parameters. The PDF file, provided by STMicroelectronics, gives you a well documented overview about what this certain demo program is intended to do.

10.6. Error conditions

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 switch is into bootstrap mode (BSL=0). If you are using a serial interface different than COM1 you must revise the ”doit.bat” file according to: ”flash166 /P <file.bin> /COM2”.

10.7. Lauterbach Flash Tool

This tool allows programming the internal Flash memory of the ST10F168 controller. The tool is provided within the tool directory on CD-ROM.

10.8. Make a project with the TASKING C-Compiler

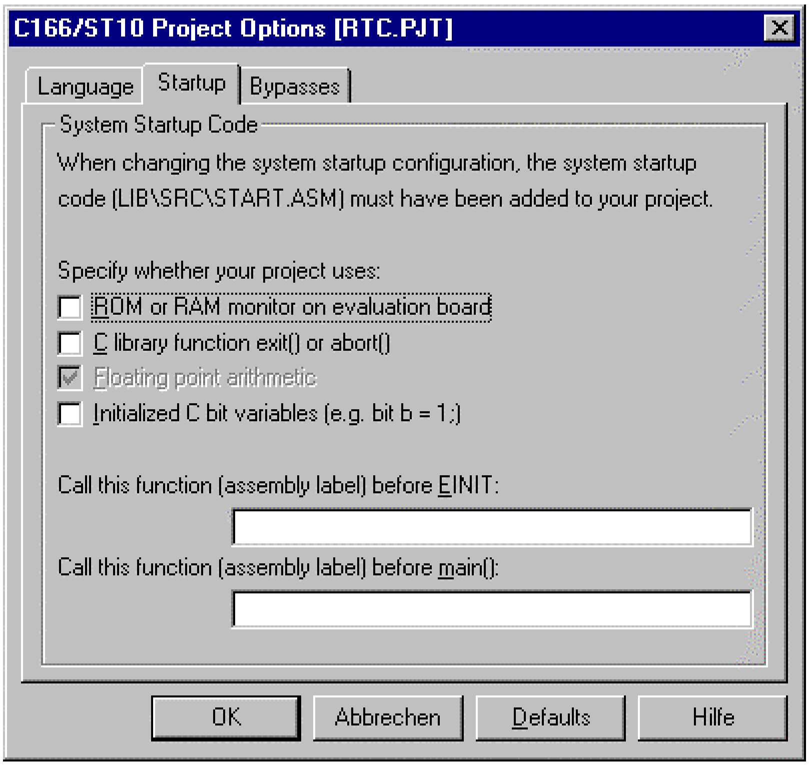
According to the sample project ”\DC166\EXAMPLES\RTC168_RELEASE \RTC.PJT” we at FS FORTH-SYSTEME would like to show you how to create a fast start-up program. For the Keil compiler we have written the start-up file ”START168.A66”. It is not intended by this chapter to explain the C compiler nor the CrossView debugger in detail. For that please refer to the online help files coming with the compiler demo package. The RTC sample initializes and reads the RTC72423A realtime clock while running in a loop. The registers which are read, are written to the serial interface ASC0 (RS232). Therefore you should connect your PC to connector X3 through an appropriate standard cable. To watch the data coming from the EVA168 evaluation board run a standard terminal program, like HyperTerminal, and look what happens.

10.8.1. Memory Mapping

Since we do not want to run the program by the CrossView debugger we have to generate an Intel-formatted hex file without carrying any debug information. Furthermore the program memory should be located at 0x00.0000 and is controlled by /CS0. The SRAM is located as a word oriented memory (2 * 128 kByte) which follows right after this first program section (/CS1). At last the RTC uses a 4 kByte memory block and should be supervised through /CS2.

Address Range	Chip-Select	Bus Width	Function
0x20.0FFF 0x20.0000	/CS2	8 bit	RTC; 4 kByte
0x13.0FFF 0x10.0000	/CS1	16 bit	256KSRAM
0x0F.FFFF 0x08.0000	/CS0	16 bit	512K Program, optional
0x07.FFFF 0x00.0000			512K Program

Use the EDE setup menu to define all CPU and memory specific parameters. Type EDE → Project Options and *deselect* ”ROM or RAM monitor”. Otherwise the EDE will reserve a certain amount of memory which is requested to run the CrossView debugger in the SRAM or Flash memory.



System Startup Code

When changing the system startup configuration, the system startup code (LIB\SRC\START.ASM) must have been added to your project.

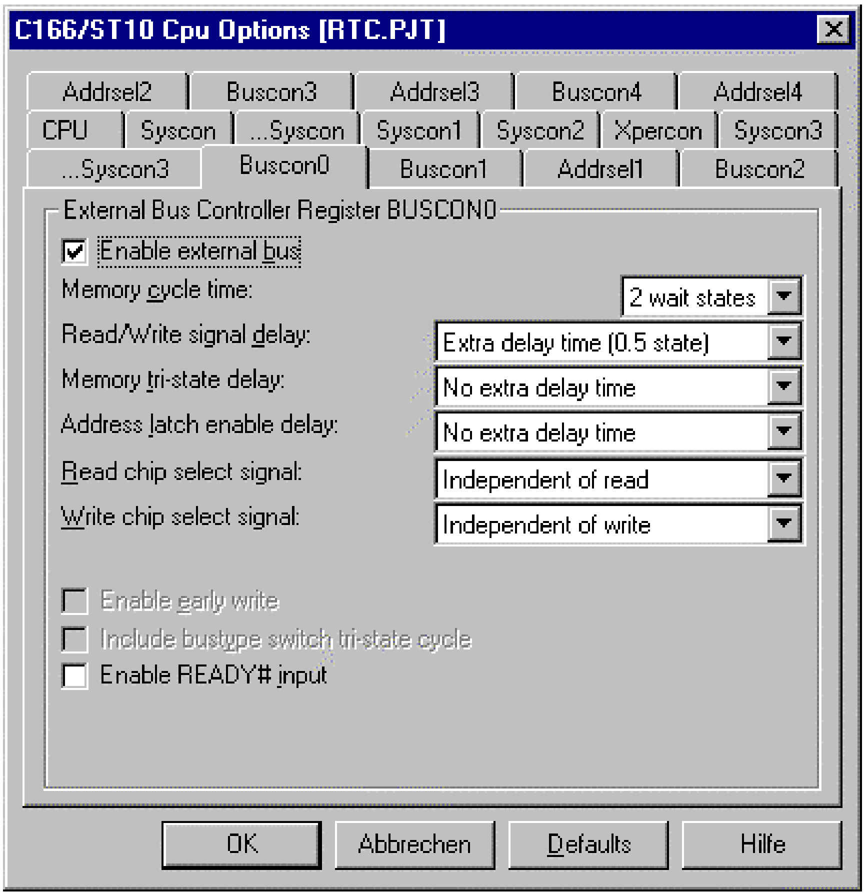
Specify whether your project uses:

- ☐ ROM or RAM monitor on evaluation board
- ☐ C library function exit() or abort()
- ☑ Floating point arithmetic
- ☐ Initialized C bit variables (e.g. bit b = 1;)

Call this function (assembly label) before EINIT:

Call this function (assembly label) before main():

Set parameters for bus mode and address selection. All settings will be automatically updated within the startup.asm file whenever you build the project. There is no need to modify the start-up file manually. Just take a start-up file (template) from the INCLUDE directory of your C/C++ compiler CD-ROM and copy it into your project path (see EDE->Project properties). If you like to change the start-up file by your own, use a make utility and don't run the compiler's build option!



External Bus Controller Register BUSCON0

☑ Enable external bus

Memory cycle time: 2 wait states

Read/Write signal delay: Extra delay time (0.5 state)

Memory tri-state delay: No extra delay time

Address latch enable delay: No extra delay time

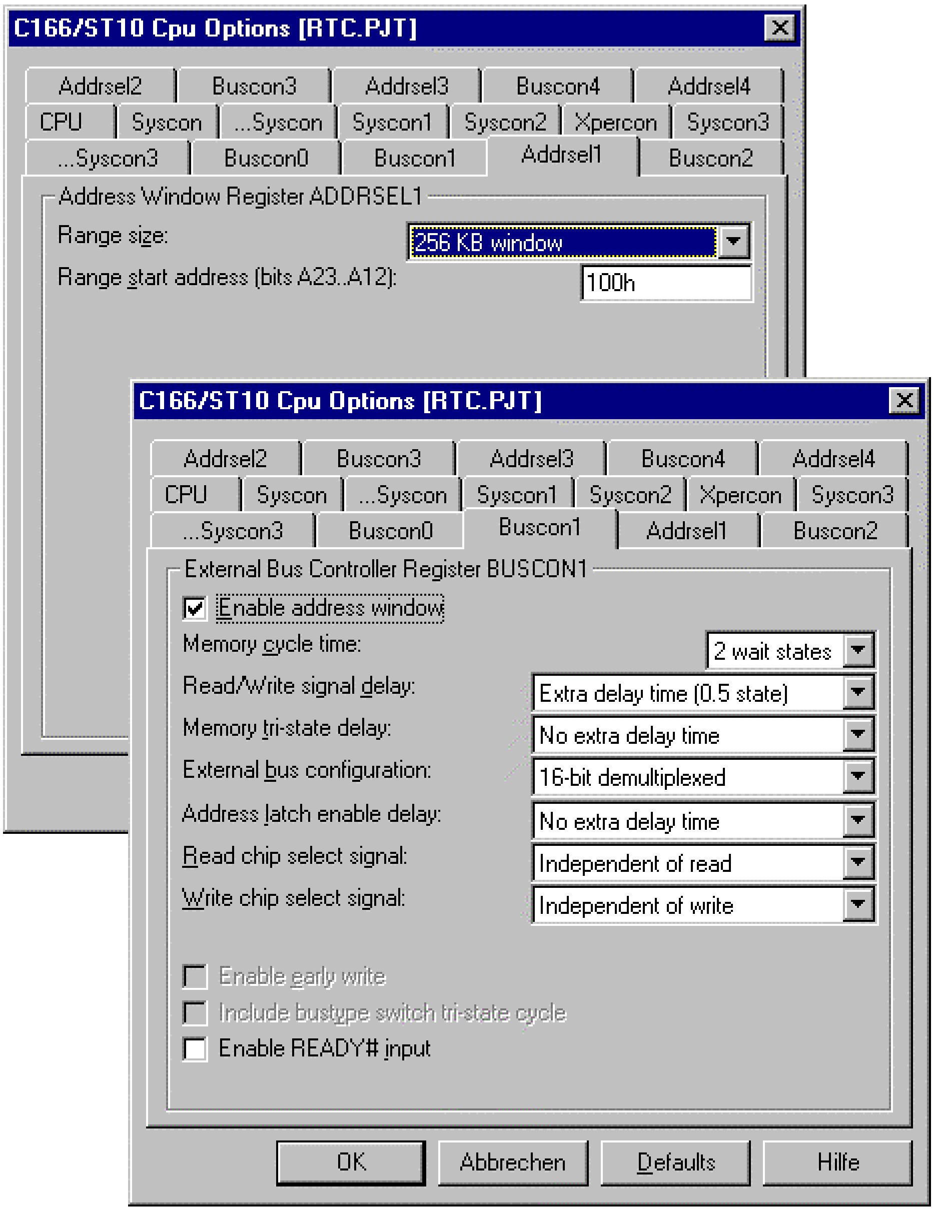
Read chip select signal: Independent of read

Write chip select signal: Independent of write

☐ Enable early write

☐ Include bustype switch tri-state cycle

☐ Enable READY# input



C166/ST10 Cpu Options [RTC.PJT]

Address Window Register ADDRSEL1

Range size: 256 KB window

Range start address (bits A23..A12): 100h

C166/ST10 Cpu Options [RTC.PJT]

External Bus Controller Register BUSCON1

☑ Enable address window

Memory cycle time: 2 wait states

Read/Write signal delay: Extra delay time (0.5 state)

Memory tri-state delay: No extra delay time

External bus configuration: 16-bit demultiplexed

Address latch enable delay: No extra delay time

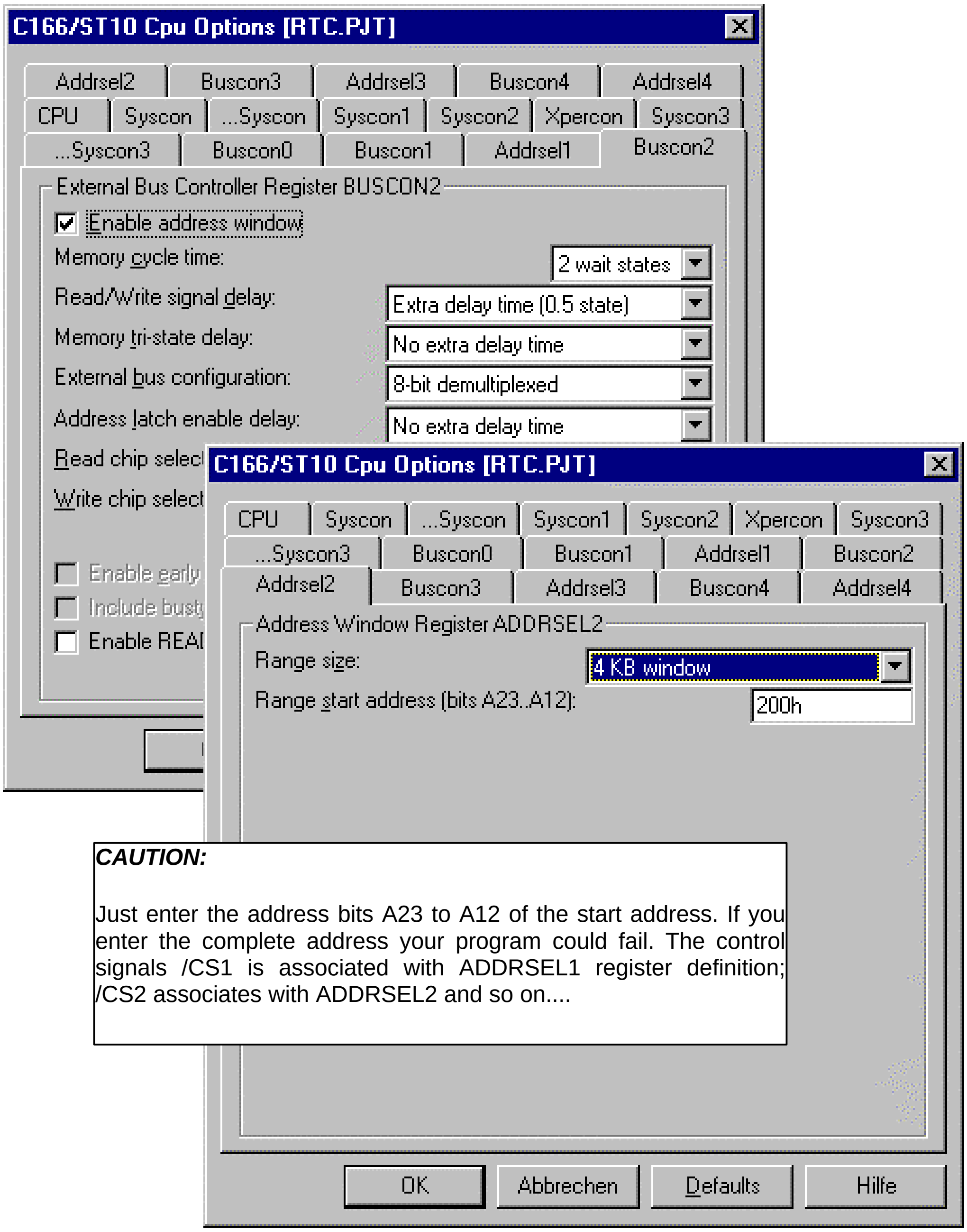
Read chip select signal: Independent of read

Write chip select signal: Independent of write

□ Enable early write

□ Include bustype switch tri-state cycle

□ Enable READY# input



C166/ST10 Cpu Options [RTC.PJT]

External Bus Controller Register BUSCON2

☑ Enable address window

Memory cycle time: 2 wait states

Read/Write signal delay: Extra delay time (0.5 state)

Memory tri-state delay: No extra delay time

External bus configuration: 8-bit demultiplexed

Address latch enable delay: No extra delay time

Read chip select

Write chip select

□ Enable early

□ Include busty

□ Enable REAI

C166/ST10 Cpu Options [RTC.PJT]

Address Window Register ADDRSEL2

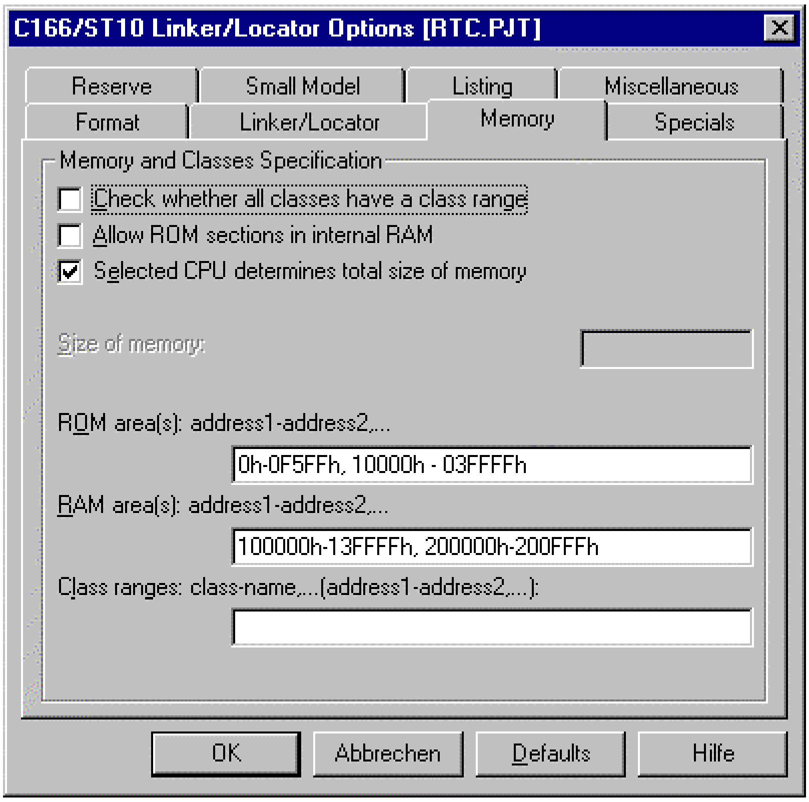
Range size: 4 KB window

Range start address (bits A23..A12): 200h

CAUTION:

Just enter the address bits A23 to A12 of the start address. If you enter the complete address your program could fail. The control signals /CS1 is associated with ADDRSEL1 register definition; /CS2 associates with ADDRSEL2 and so on....

Define where RAM and ROM (Flash) sectors are located within your individual application. If you wish to use internal components like CAN or XRAM you must consider these parts through a declaration within this menu section.



C166/ST10 Linker/Locator Options [RTC.PJT]

Memory and Classes Specification

□ Check whether all classes have a class range

□ Allow ROM sections in internal RAM

☑ Selected CPU determines total size of memory

Size of memory:

ROM area(s): address1-address2,...

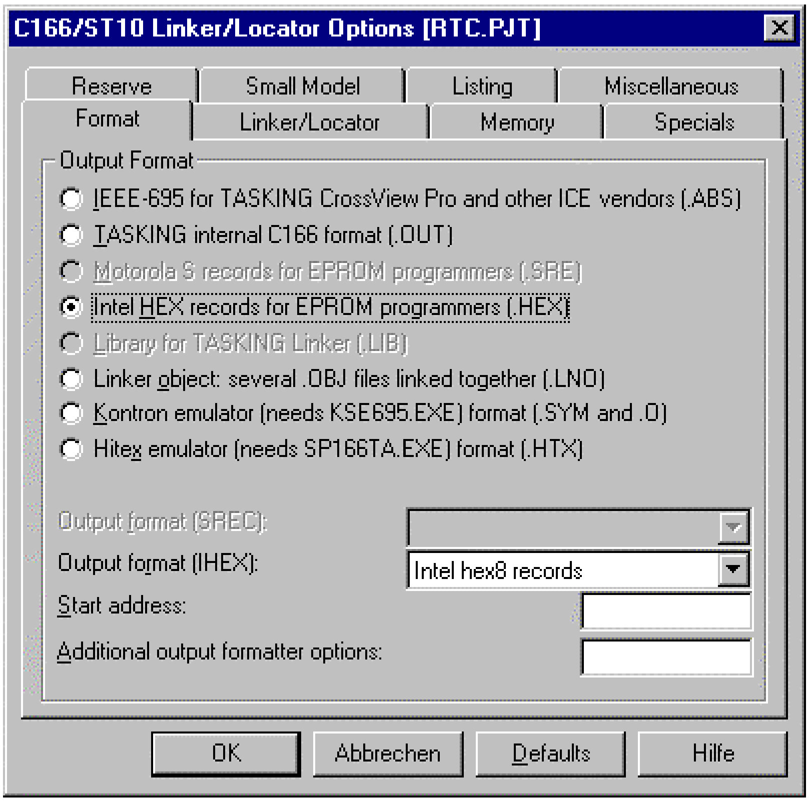
0h-0F5FFh, 10000h - 03FFFFh

RAM area(s): address1-address2,...

100000h-13FFFFh, 200000h-200FFFh

Class ranges: class-name,...(address1-address2,...):

Generate an Intel-formatted hex dump which is converted into a binary output afterwards and download it into the Flash with the FLASH166 tool. If you like to run the CrossView debugger select ”IEEE-695” format including debug information.



C166/ST10 Linker/Locator Options [RTC.PJT]

Output Format

○ IEEE-695 for TASKING CrossView Pro and other ICE vendors (.ABS)
○ TASKING internal C166 format (.OUT)
○ Motorola S records for EPROM programmers (.SRE)
● Intel HEX records for EPROM programmers (.HEX)
○ Library for TASKING Linker (.LIB)
○ Linker object: several .OBJ files linked together (.LNO)
○ Kontron emulator (needs KSE695.EXE) format (.SYM and .O)
○ Hitex emulator (needs SP166TA.EXE) format (.HTX)

Output format (SREC):

Output format (IHEX): Intel hex8 records

Start address:

Additional output formatter options:

11. Know-how for handling C166/ST10F168 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.

For 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. Please have a look on \dc166\etc\EVA168.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!

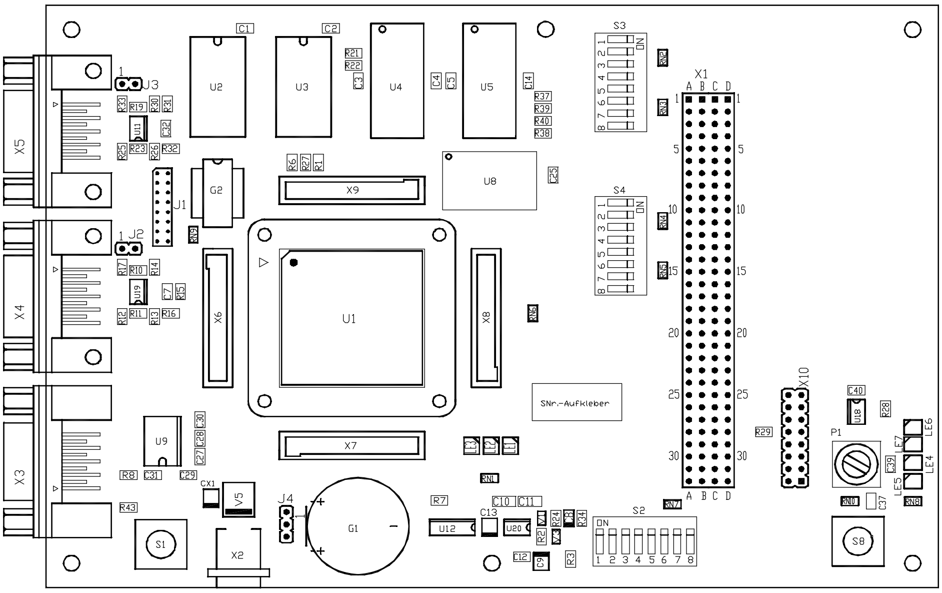
For release, you have to link a ”start-up” to your application, which is automatically started after RESET. The start-up is usually an assembler program linked to your application (Keil example **RTC168_Release**) or is automatically generated via the project settings (Tasking example **RTC168_Release**). 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!

12. Appendix

12.1. Top View of the EVA168 Evaluation Board



EVA168_1 Placement TOP 24.11.99