SIEMENS

Microcontrollers

ApNote AP1642

☑ : Additional file
AP1642xx.EXE available

In-System Programming of C167CS-32F, C161CI-32F and C161SI-32F Flash Devices (all Ax-steps)

Siemens C167CS-32F/ C161CI-32F/ C161SI-32F 16-bit microcontrollers provide 256 KBytes Flash memory on-chip. In-system programming of C167CS-32F/ C161CI-32F/ C161SI-32F Flash devices is supported by the OTP/ Flash/ EEPROM memory programming tool “Memtool”. Additionally included is a tested bootstrap loader routine.

Author : Peter Kliegelhöfer / HL MD AE Microcontroller Application Support

Contents **Page**

AP1642 ApNote - Revision History		
Actual Revision : 02.99 Previous Revision : 07.98		
Page of actual Rev.	Page of prev.Rel.	
all	all	C161CI-32F added
5	5	Chapter “Handling of Flash Addresses” added
6-9	6-9	`MEMTOOL.EXE` and C167CS-32F/ C161CI-32F drivers updated
6-9	6-9	`MEMTOOL.EXE` and several driver files updated; formal text changes

1. Flash Memory Overview

The C167CS-32F/ C161CI-32F/ C161SI-32F devices provide 256 KBytes of flash memory on-chip for both instruction code and constant data. The flash memory is segmented in 8 flash sectors of 32 KBytes each (all Ax steps).
The lower 32 KBytes of the on-chip flash memory of the C167CS-32F/ C161CI-32F/ C161SI-32F devices can be mapped to either segment 0 (00’0000$_H$ to 00’7FFF$_H$) or segment 1 (01’0000$_H$ to 01’7FFF$_H$) during the initialization phase to allow external memory to be used for additional system flexibility. The upper 224 KBytes of the on-chip flash memory are assigned to locations 01’8000$_H$ to 04’FFFF$_H$ (figure 1).

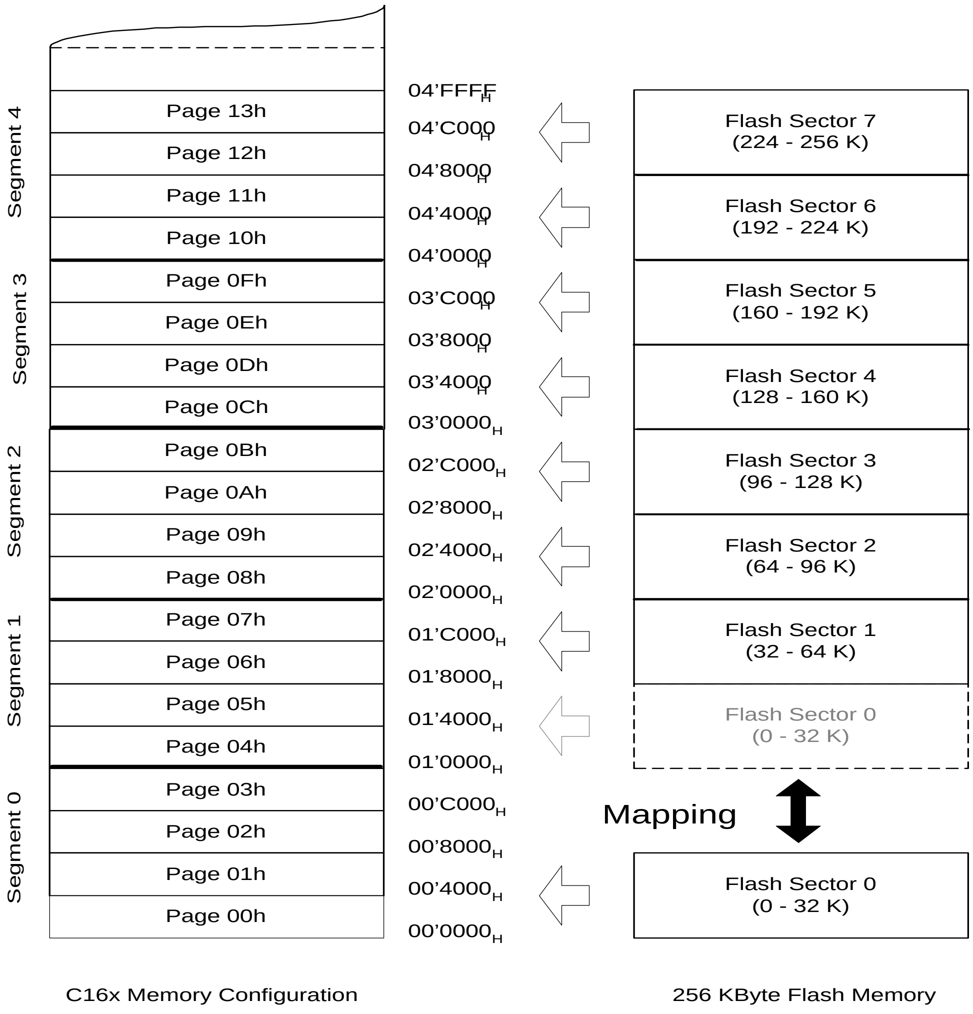

Figure 1
Flash Memory Configuration

2 Flash Memory Configuration

Upon reset the default memory configuration of the C167CS-32F/ C161CI-32F/ C161SI-32F is determined by the state of its $\overline{\text{EA}}$ pin. When $\overline{\text{EA}}$ is low the internal flash memory is disabled and the startup code is fetched from external memory.

In order to access the on-chip flash memory after booting from external memory the internal flash memory must be enabled via software by setting bit ROMEN in register SYSCON. The lower 32 KBytes of the flash memory can be mapped to segment 0 or segment 1, controlled by bit ROMS1 in register SYSCON. Mapping to segment 1 preserves the external memory containing the startup code, while mapping to segment 0 replaces the lower 32 KBytes of the external memory with on-chip flash memory. In this case a valid vector table must be provided.
As the on-chip flash memory covers more than segment 0, segmentation should be enabled (by clearing bit SGTDIS in register SYSCON) in order to map the whole internal flash into the address space.

Whenever the internal memory configuration of the C167CS-32F/ C161CI-32F/ C161SI-32F is changed (mapping, enabling, disabling) the following procedure must be used to ensure correct operation:

- Configure the internal flash memory as required
- Execute an inter-segment branch (JMPS, CALLS, RETS)
- Reload all four DPP registers

Note:
Register SYSCON can only be modified **before** the execution of the EINIT instruction.

Note:
For detailed informations concerning the handling of internal non-volatile memory please refer to the users manual, chapter „System Programming“ (Handling the Internal ROM/ Pits, Traps and Mines).

2.1 Handling of Flash Addresses

When performing command sequence operations to the 256 Kbyte on-chip Flash of the C167CS-32F/ C161CI-32F/ C161SI-32F (all Ax steps) one has to consider that the flash physically consists of **two separate** 128 Kbyte blocks which have to be addressed *individually*.

Example:

Flash address 0AAAAh in **lower** 128K flash block:

PAG = 00 0000 0110b
POF = 0010 1010 1010 1010b => final address: 01'AAAAh (pointer to segment 1)

Flash address 0AAAAh in **upper** 128K flash block:

PAG = 00 0000 1110b
POF = 0010 1010 1010 1010b => final address: 03'AAAAh (pointer to segment 3)

All flash command, sector and data register addresses have to be located within the active flash memory space. The active space is that address range to which the physical flash addresses are mapped as defined by the user.
When using data page pointers (DPPs) for command or sector addresses one has to make sure that both MSBs of the command or sector address are reflected in both LSBs of the selected DPP.

Only register-*indirect* addressing like "mov [Rx], Ry", "mov Rx, [Ry]" can be used for command, sector or write-data accesses. Direct addressing is **not allowed**.

When performing *read* cycles within flash commands (e.g. "read Flash Status Register") it is **not allowed** to use indexed addressing mode, which is only available with instructions "mov Rx, [Ry+#data16]" and "movb Rx, [Ry+#data16]". Make sure that the C-compiler doesn't use these instructions.

3 Memtool - The OTP/ Flash/ EEPROM Memory Programming Tool

In-system programming of on-chip OTP- Flash- and EEPROM memory is supported by the Windows-based programming tool „Memtool“ which is freely available. Siemens provides driver updates on the Internet; please ask for the current status. Memtool is an application example as well as a programming tool for on-chip OTP- Flash- and EEPROM memory, supporting C161CI-32F, C161SI-32F, C167CR-16F, C167CS-32F and C163-16F Flash devices and C164CI-8E OTP devices.

Thanks to the modular structure Memtool is easily expandable for future C16x devices. For an interested user all driver sources are also available (figure 2). Along with the drivers come readme.txt files which contain informations about the hardware requirements and latest informations about Memtool and each driver.

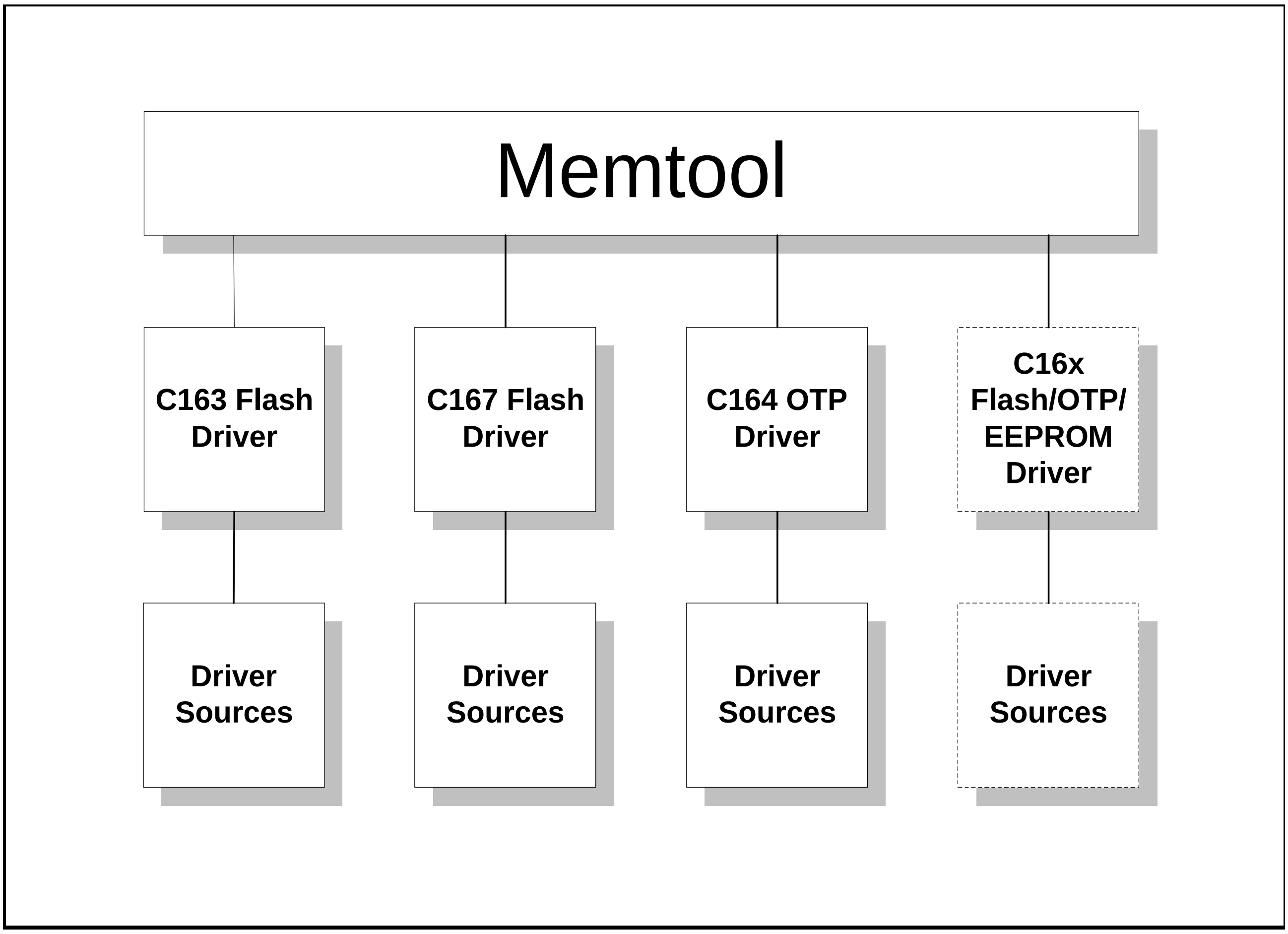

Figure 2
The structure of Memtool

Memtool allows programming and erasing the C167CS-32F/ C161CI-32F/ C161SI-32F flash memory employing an Phytec C167 starter kit, an Ertec EVA167 evaluation board (in case of the C161CI-32F/ C161SI-32F an additional adaptor is needed) or a customer's application (free ASC0 needed). The process of programming is based on the bootstrap loader (BSL). The C167CS-32F/ C161CI-32F/ C161SI-32F enters bootstrap mode if pin P0L.4 is sampled low at the end of a hardware reset. In this case the BSL is activated, independent on the selected bus mode.

All software including the programming data is downloaded from a host PC into the internal RAM of the microcontroller. The application requires only internal RAM, external RAM is not required. Since the C167CS-32F/ C161CI-32F/ C161SI-32F is a 5 Volt-only device, no additional voltage for flash programming or erase is required.

In order to download the application from a PC, a serial link has to be established. The C167CS-32F/ C161CI-32F/ C161SI-32F already provides an asynchronous serial interface that only has to be connected to COM1 or COM2 of your PC. Supposing you are using a Phytec C167 starter kit, you can directly connect its serial connector with the COM1/ COM2 interface of your PC.

3.1 Special Memtool Options/ Hints

For default Memtool expects serial communication via RS232. Some drivers also support single-wire transmission (K-Line). All drivers which support K-Line protocol are marked with "KL" (see box "Target"/ "Change"/ "Select Controller Type"). Select the desired mode (RS232 or K-Line) in box "Interface" / "Setup Interface".

When using the K-Line option transmission speed can be significantly increased by deactivating the FIFO of the used communication port of the PC. This concerns Windows 95 as well as Windows NT.

In some cases it is useful to get a LOG-file of the transmitted data. For LOG-file generation switch "DiagLog" in the Memtool configuration file (memtool.ini/ siemtool.ini; put down under "[main]") has to be set to "1". The LOG-file will be closed by leaving Memtool. A set box for the LOG switch is currently not implemented.

4 C167CS-32F/ C161CI-32F Flash Memory Programming Drivers Overview

Several C167CS-32F/ C161CI-32F Flash programming drivers are available. This is founded on different demands of different applications. The following overview is just a snapshot of the status quo. In the future new versions will be available which will partly replace the current ones. Please pay attention to the readme.txt files which come along with the drivers. For C161SI-32F flash devices you can use the C161CI-32F drivers; there are no differences concerning the flash module.

Due to the fact that with some steps it is not possible to perform flash read accesses as long as the test ROM bootstrap loader is still active (this also includes read accesses to the Flash Status Register “FSR”), the success of a flash programming/ erase operation cannot be verified.
In case you are using the internal test ROM bootstrap loader for booting we provide the drivers “C167CS-32F Blind 1st Sil. ”, “C167CS-32F KL Blind 1st Sil.” and “C161CI-32F Blind 1st Sil. ” which have some restrictions (see below).

Flash read accesses (this also includes read accesses to the Flash Status Register “FSR”) are possible when booting from external memory (EPROM, Flash) or when booting (=starting a BSL routine as a user program) out of the pre-programmed internal flash memory.
In case you do not use the internal test ROM bootstrap loader we provide the drivers “C167CS-32F Full 1st Sil.”, “C167CS-32F KL Full 1st Sil.” and “C161CI-32F Full 1st Sil.”.

All C167CS-32F/ C161CI-32F Flash drivers are properly tested with 9600 Baud transmission rate. You may also try a higher baudrate (>= 19200 Bd) for faster data transmission. For single wire transmission (K-Line) we recommend to use lower baudrates (<= 19200 Bd).

The readme.txt files in the sub-directories (memtool/ siemdrv/ C16x...) contain important informations about each driver:

Note:
Please also refer to the dedicated errata sheet of the used C167CS-32F/ C161CI-32F/ C161SI-32F device. Later steps may have not the restrictions described above.

C167CS-32F Full 1st Sil. / C167CS-32F KL Full 1st Sil./ C161CI-32F Full 1st Sil.

Please use these drivers when...

- booting out of external memory (which contains a BSL routine; in this case make sure that pin $\overline{EA}$ (external access) is low and pin P0L.4 is not low during reset (which would activate the internal bootstrap loader)

- booting out of the pre-programmed on-chip user flash (the user flash must contain a bootstrap loader routine which will be executed as a normal user program after reset; in this case make sure that pin $\overline{EA}$ (external access) is high and pin P0L.4 is not low during reset)

Note:
Pin P0L.4 low during reset activates the bootstrap loader which resides in the internal test ROM. When the internal bootstrap loader is selected (which means that the internal test ROM is active) no read accesses to the flash memory (which also includes the flash status register) are possible. Read accesses to the flash memory as well as to the flash status register FSR will always deliver value FFFFh, independent on the real value.
This version supports all available flash functions, performs verification of programming and erase and evaluates the flags of the FSR in case of a flash operation.

Note:
A tested bootstrap loader (target: external memory/ internal flash memory) is attached.

C167CS-32F Blind 1st Sil. / C167CS-32F KL Blind 1st Sil./ C161CI-32F Blind 1st Sil.

These versions are recommended when programming the flash memory the first time or when using the internal test ROM bootstrap loader. The internal test ROM bootstrap loader can be activated with P0L.4 low during reset. The state of pin $\overline{\text{EA}}$ (external access) will be ignored.

Restrictions
Read accesses to the flash memory (which also includes the flash status register FSR) are not possible (the read value is always FFFFh, independent on the real value). This is the reason why this version neither performs a verification of programming and erase nor evaluates the flags of the FSR in case of a flash operation. Error flags (if set in case of a failure during a flash operation) will not being recognized and therefore not evaluated.
Due to this fact the success of a flash operation cannot be controlled.
The success messages of the programming tool are no reliable indicators !

Wait routines for each flash operation replace polling of the busy bit (FSR).

Note:
When booting with the internal bootstrap loader all read accesses to the flash memory (and to the flash status register FSR) deliver the same dummy value FFFFh, independent on the real content of the flash memory/ FSR.
Please consider this e.g. when working with an Ertec EVA167 evaluation board (BTL switch in position „ON“): a memory dump of the flash memory will always deliver FFFFh value !

Note:
Please also refer to the dedicated errata sheet of the used C167CS-32F/ C161CI-32F/ C161SI-32F device. Later steps may have not the restrictions described above.