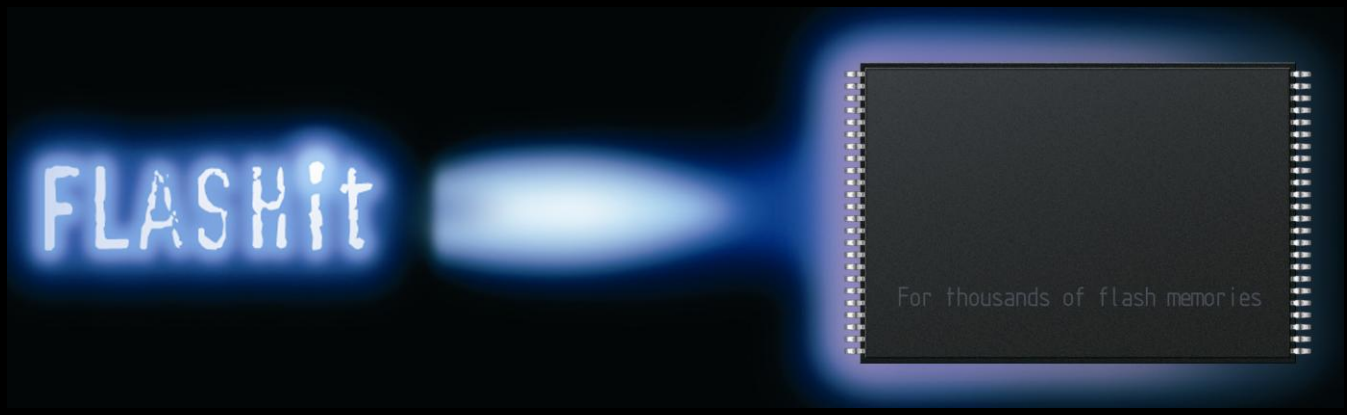


The FLASH Programming Tool

for easy programming of application software in a
STM32 ARM Cortex M3 System

LIMITED WARRANTY!

By using this FLASHit product and its associated products, you agree to the following conditions.
If you are unable to agree to these conditions, please notify us within two days of purchasing FLASHit.

LIMITED warranty:
hse-electronics guarantees that the **FLASHit** product will basically work as advertised and without defects if used properly for a period of 6 months after delivery.

CLAIMS of the customer:
At the discretion of **hse-electronics**, the entire liability of **hse-electronics** and your sole claim shall involve either
a) a refund of the paid price or
b) subsequent improvement or product placement.
This limited warranty does not apply if the failure of the product can be attributed to an accident or improper use.

NO additional warranty:
hse-electronics excludes any additional warranty claims related to the delivered product and associated manuals and written materials.

NO liability for consequential damage:
Neither **hse-electronics** nor the suppliers of **hse-electronics** are liable to pay compensation (included but not limited to damages for loss of profit, service interruption, loss of business information or data or any other financial loss) arising from the use of this **hse-electronics** product even if **hse-electronics** has been notified of the possibility of such damages.

LIMITED liability:
The liability of **hse-electronics** shall in any case be limited to the amount that the customer paid for the product. This exemption shall not apply to damages caused by willful intent or gross negligence on the part of **hse-electronics**.

JURISDICTION:
Disputes arising from this licensing agreement may only be resolved in the regional court of Kiel.

If you have any questions about this agreement, please contact your dealer or send us an e-mail at **info@hse-electronics.com**.

FLASHit 9-STM32 Manual

Table of contents

FLASHit 9-STM32 Manual

1 FLASHit

1.1 FLASHit

FLASHit allows you to quickly and easily program application software in a STM32 ARM Cortex-M3

1.1 How does FLASHit work?

FLASHit was designed for easy use. Once a connection to the hardware has been established via the RS232 interface and the interface parameters have been set, you only have to select an application for the upload / programming process (burning) to the target system to begin.

This process occurs as follows (without enabled options):
- Automatical detection of the MCU-type
- The bootstrap loaders of the target system are activated
- The configuration of the target system is determined
- The type of FLASH being used is detected automatically
- The FLASH data are displayed
- Hex file analysis: The sectors to be deleted are identified and selected and hex data are arranged in a binary array
- The FLASH memory is deleted (either individual sectors or the entire chip)
- FLASH in the target system is reprogrammed in blocks (programmed sectors are selected).

2 System Requirements

Target System
MCU: STM32 ARM Cortex M3

PC
Operating system: All current versions of Windows
Hardware: Pentium or higher

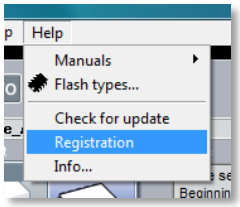
3 Program-Installation and Registration

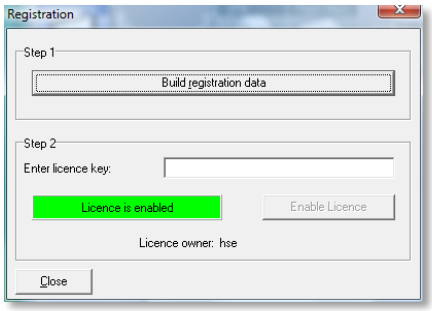
3.1 Installation

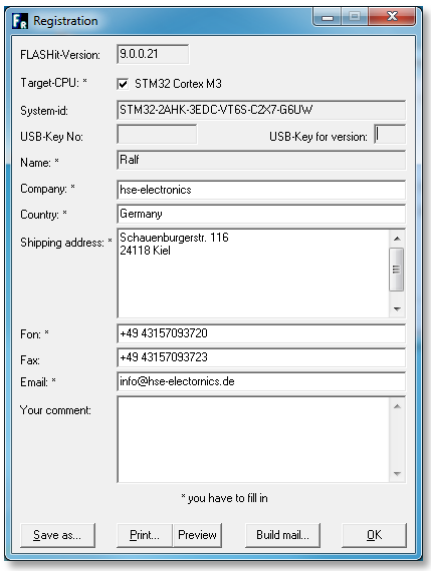
If you have a CD:
- Insert the program CD into the CD-ROM drive of your PC.
- Follow the instructions on your screen.

If you downloaded FLASHit from the hse-electronics website:
- Double-click `setupflashit_STM32.exe`
- Follow the instructions on your screen.

3.2 Registration







Two licensing methods are available:
1. PC-based licensing
2. USB dongle-based licensing

When FLASHit is launched for the first time, you will need to register it in the ***Help*/*Registration*** menu (otherwise FLASHit will run in restricted demo mode). If FLASHit was delivered with a dongle, you can skip the following steps. Insert the dongle into the USB port to unlock the program.
To register FLASHit, complete the following two steps:

Step 1:
Perform this step on the computer you want to register FLASHit for!
Click ***Build registration data***.
FLASHit will generate your own personal ***System Id***.
- Fill in the fields marked with "*".
- Now send us your registration data either by e-mail (***Build mail...***) or by fax (***Print...***).
- Once we receive the data we will send you your activation code (by e-mail or fax).

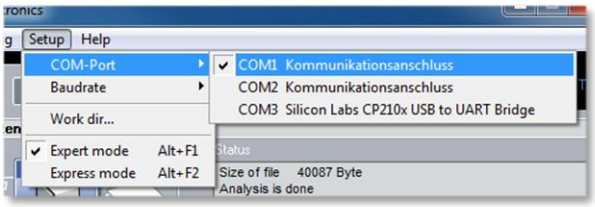
Step 2:
- Enter the activation code under ***Enter license key***.
- Click ***Enable license***.
- If the message ***License is enabled*** appears (in a green field), then FLASHit has been successfully registered.

If you have any questions about registration, e-mail us at **info@hse-electronics.com.**

Note: If you need a new activation code because your hardware or other conditions have changed, e-mail us the old license number and we will send you a new one.

4 Program Functions

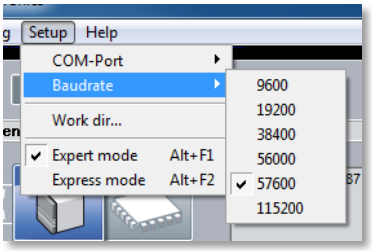
4.1 Establishing A Connection



- Connect your PC to the target system via the serial interface.
- Select the COM port you want to use in the ***Setup*** menu. you can choose from the first 10 installed COM devices whose COM numbers can range from 1 to 99.

Note: If you are using a USB RS232 adapter, you can also run FLASHit via a USB port. All settings in bold are the default or recommended settings.

4.2 Setting the Baud Rate



The adjustable baud rate between the target system and your PC depends on, among other factors, the clock rate of the target system (divisible by a standard baud rate). Just try out different baud rates
and select the one you want from the ***Baud rate*** menu

4.3 FLASHit Working Directory

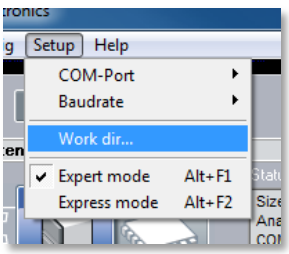
FLASHit creates several files during a session:
`flashit.ini` // program settings
`result.txt` // return values
Depending on which version of Windows is being used (in this case WinXP), these files are saved by default in the directory specified by Windows.
(e. g.: `c:\Documents and Settings\All Users\Applications Datas\FLASHit\*.*`)
Windows must be set to allow the creation and writing of files in this directory!
If you want to use a different working directory, you need to create one in advance.

Proceed as follows:
- Open the Properties dialog from the context menu and select the ***Connection*** tab.
- Enter the following in the ***Target*** input box (example):

```
c:\programme\flashit_STM32.exe WORK_DIR=j:\ini
```

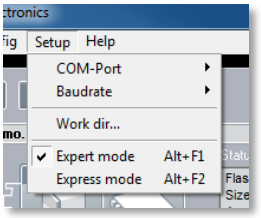
Result:
Double-click the FLASHit icon to launch the application and the files will then be saved in the "`j:\ini`" directory.
If the path name is enclosed in quotation marks, blank spaces are allowed (example):

```
c:\program files\flashit_STM32.exe WORK_DIR="c:\Program Files\FLASHit"
```



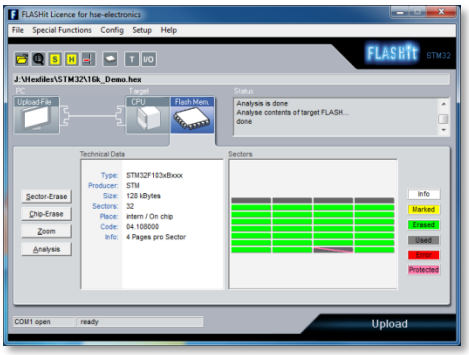
You can check the path by selecting ***Work dir...*** from the ***Setup*** menu.

4.4 Expert or Express Mode

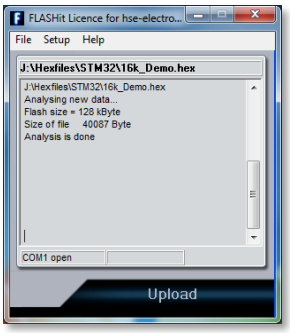


By selecting the ***Expert mode*** and ***Express mode*** menu items in the ***Setup*** menu you can choose between a basic or advanced FLASHit user interface.
FLASHit "remembers" the previous settings when restarted

„Expert mode“ (Default)

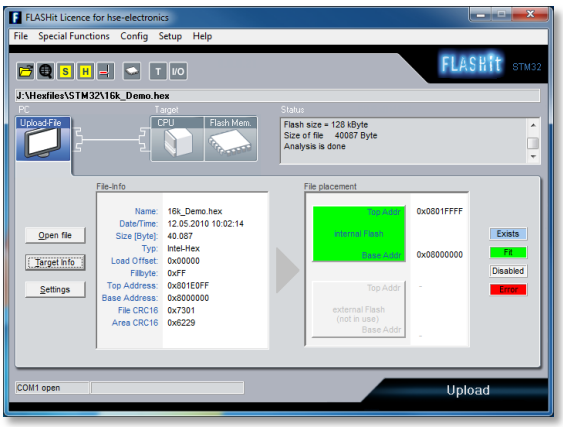


Basic user interface of FLASHit: "Express mode"

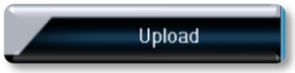


4.5 Uploading to the Target System ("burn FLASH")

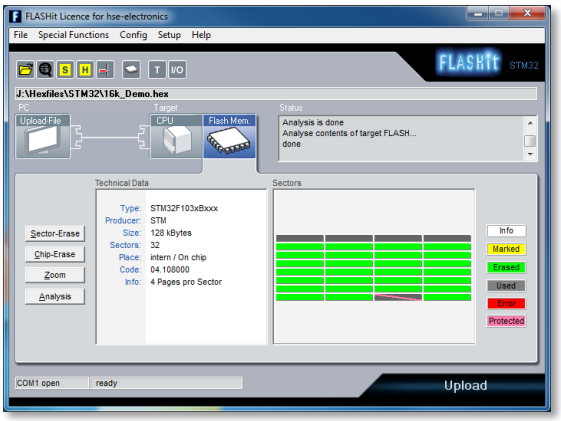
Having addressed the most important program settings in Items 4.1 and 4.2, we can now begin the process of uploading the application software to the target system.



The ***UploadFile*** tab lets you select a specific file (set format filter to *.hex). Information on the file you select will appear in the ***File-Info*** window. Click the ***Upload*** box in the bottom right corner to start uploading the file to the FLASH memory.

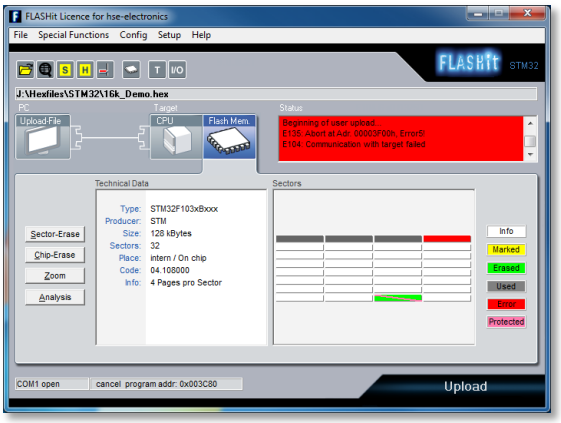


FLASHit will attempt to automate the settings described below.



During the upload FLASHit switches to the ***FlashMem*** tab and displays the data of the FLASH memory being used (FLASH type, manufacturer, memory capacity and the number of sectors). The ***Status*** window shows the progress of the upload.
You can cancel the upload at any time by clicking ***Cancel***.

The Status window will turn green when the upload is complete.

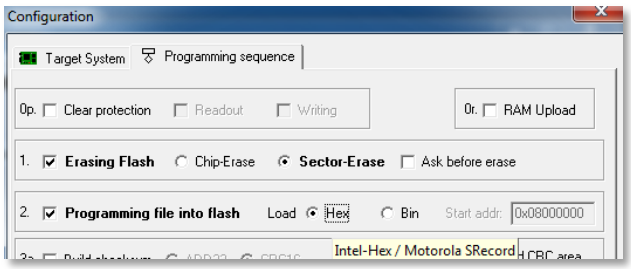


If the Status window turns red the upload was unsuccessful. The possible causes of the upload failure can be viewed in the ***Status*** window.
The appendix contains list of all error codes.

Note: If FLASHit was unable to automatically reset your target system before the upload began, you will be prompted to reset the target system. To do so, you need to activate the Bootstrap mode (BOOT1 auf High). Long upload file names are shown in abbreviated form in the upper window of FLASHit for space reasons. To view the entire path, place the cursor over the panel.

4.6 HEX or BIN?

FLASHit allows you to write Intel-HEX files (*.hex) as well as Motorola-HEX-files (*.s19) and binary files to the FLASH memory of the target system.

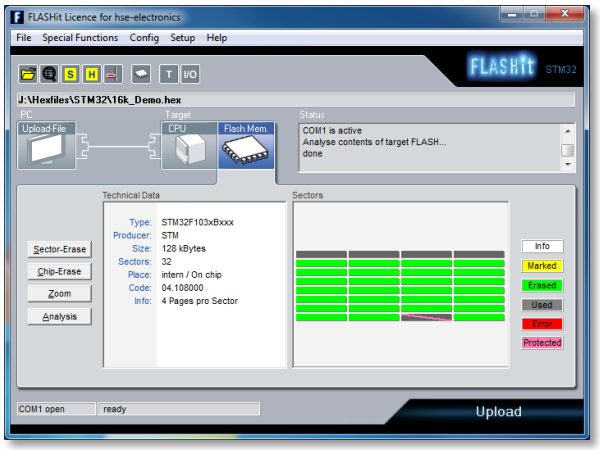


To do so, click the ***Program file into flash>Load Bin*** option in the ***Config>Target and program configuration*** menu. With this setting FLASHit will no longer interpret the file to be saved in FLASH. Even HEX files will **not (!)** be treated as HEX-files, but as binary files!
The start address in the target system where you want to save the binary file can also be set there.

If the ***Program file into flash/Load Hex*** option is enabled, you will not be able to flash binary files since they do not have the same structure as Intel hex files.
If the ***Program file into flash/Load Bin*** option is enabled and a hex file is selected to be flashed, FLASHit will ask you to confirm your selection again.

4.7 FLASH-Memory-Info

Regardless of whether or not an upload has been initiated, you can press the ***Analysis*** button on the ***FlashMem*** tab to display the following information on the FLASH type being used:

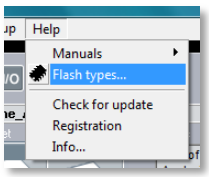


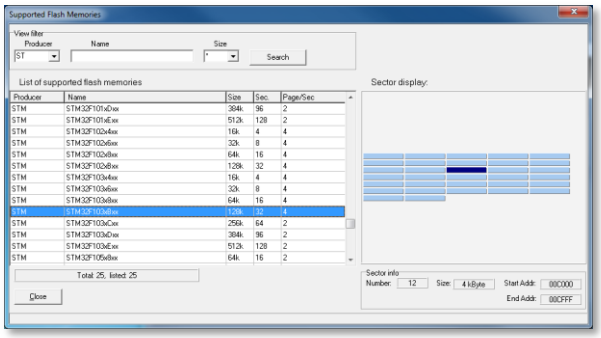
- MCU type
- Producer
- Memory size
- Number of sectors
- Place of the FLASH memory
- FLASH code
- Partitioning and size of the individual sectors

FLASHit automatically detects the MCU-type being used. A summary of the currently supported MCU- types can be found in the ***Flash types*** menu item of the ***Help*** menu.

You can also view all supported MCU-Types on our website at ***www.hse-electronics.com***.
Regardless of whether a program is being uploaded, you can use the ***Chip Erase*** option to delete the entire FLASH memory chip or the ***Sector Erase*** option to delete individual sectors of the FLASH memory by selecting them with the mouse.

4.8 Information about supported MCUs

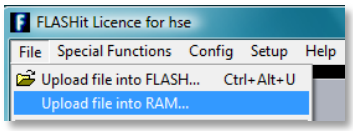


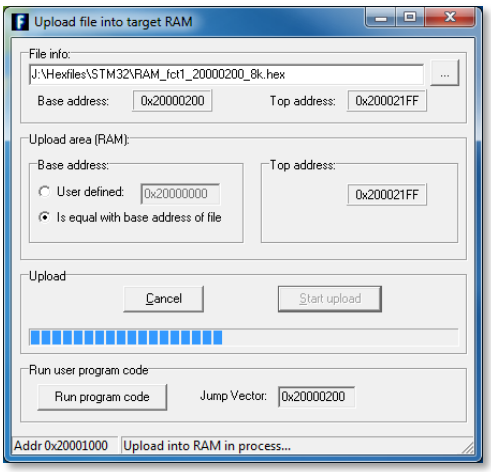


The ***Help>Flash types...*** menu lets you view the entire MCU database of FLASHit. If the particular MCU you want is not listed, hse may be able to quickly import the relevant software component.

With this function FLASHit also offers help in searching for supported MCU types.
You can enter keywords for the search in the ***View Filter*** menu (manufacturer, name and FLASH size).
All sectors of the FLASH are graphically represented. If you select a sector with the mouse, the number, size and start and end address of that sector will be shown.

4.9 Upload into the RAM of the Target



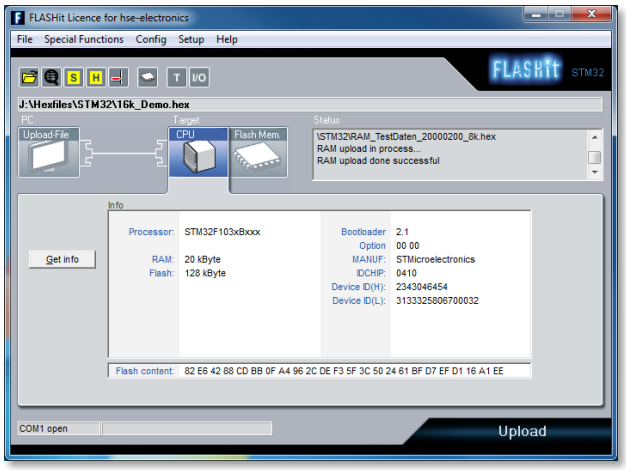


With the RAM Upload function you can transfer the content of a file to the RAM of the target system. Both the Intel hex and binary file formats are supported.

Select the RAM Upload file in the ***File>Upload file into RAM*** dialog. This file is selected independently of the FLASH Upload file selection. The ***Base address*** and ***Top address*** panels show the corresponding data of the Upload file. You may have to adjust the base address depending on the target system.
You can enter a new base address in the ***User defined*** field and the top address will then be automatically recalculated (***Top address)***.
In this example the basis address of the Upload file is 0x20000200. A new basis address, 0x20000000, is specified in the ***User defined*** field so that the data can be placed in the RAM.

4.10 Target System Info

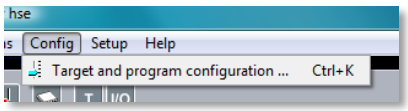
Click the ***Get info*** button to see information about the target system on the ***Target CPU*** tab.



Processor: Target CPU type (STM32F103 in this example)
RAM: Size of the RAM of the MCU
FLASH: Size of the FLASH of the MCU
Bootloader: Version of the internal bootloader
MANUF: Manufacturer
IDCHIP: MCU ID code
Device ID(H): Unique device ID register (96 bits)
Device ID(L): Unique device ID register (96 bits)
FLASH content: the first 24 bytes of the FLASH memory is displayed.

4.11 Configuring FLASHit – Target System

Although FLASHit determines most of the required data on its own, a number of parameters may have to be set manually.



You can configure the various settings of the target system using the ***Config>Target and program configuration*** menu and the ***Target System*** tab.

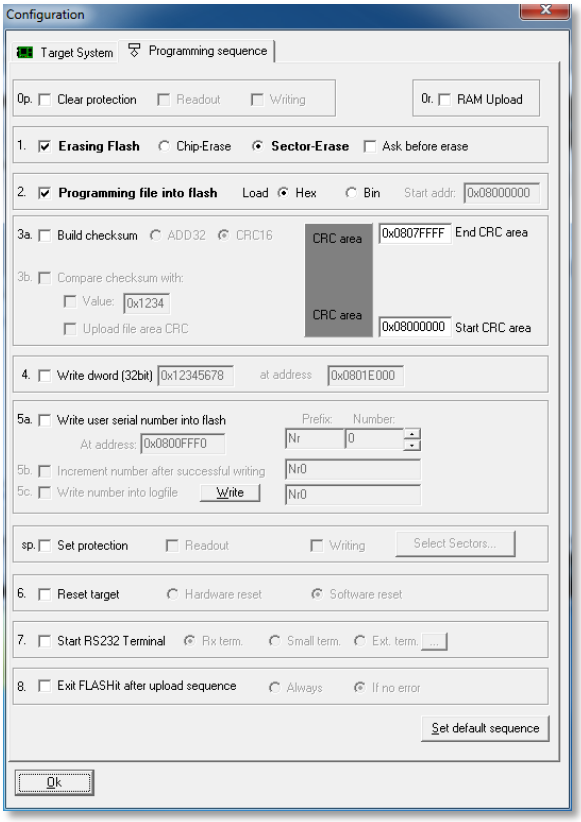
Target CPU
FLASHit normally detects the target CPU automatically.
If you select the ***Target Reset>Automatic*** option, FLASHit will try to reset the target system automatically.
If the ***User defined*** option is selected, you will be able to define the reset behavior of FLASHit (***Define***) manually.

Flash memory
FLASHit currently supports in this version only the internal OnChipFLASH-Memorys of the STM32 MCU

Basic address:
The basic address of the FLASH memory defines the address that FLASHit uses for the FLASH memory. This address is normally= 0x08000000

4.12 Configuring FLASHit – Programming Sequence

You can configure the programming sequence settings using the ***Config>Target and program configuration*** menu and the ***Program sequence*** tab. FLASHit performs parameter 0p through 8 in sequential order. The parameters highlighted in bold in the ***Configuration*** window are default settings.



0p. Clear protection
With the option Clear protection the possibility exists to deactivate ***Readout*** and or ***Writingprotection***
With the deactivation of the Readoutprotection the hole FLASHmemory will be erased!

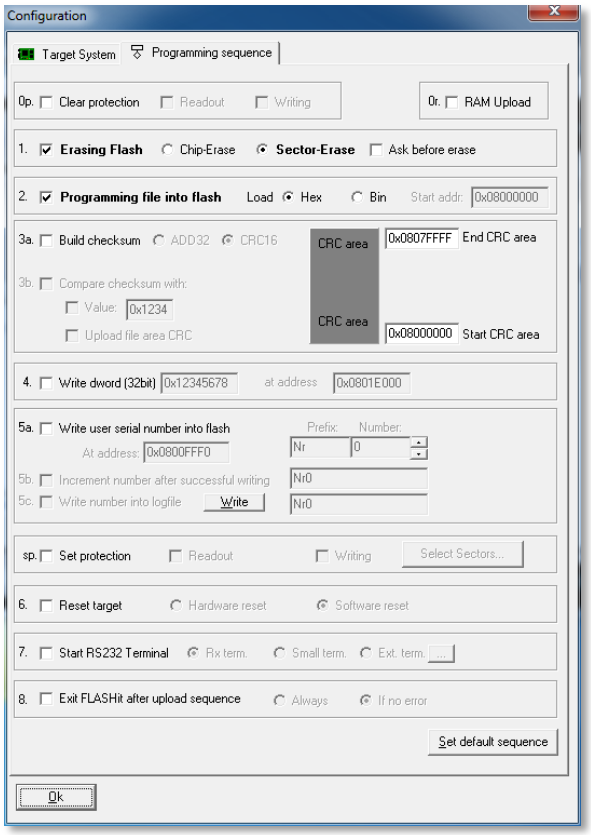
0r. Ram Upload
A chosen file is loaded into the internal RAM of the MCU with the option ***Ram Upload***.

1. ***Erasing Flash***
In this panel you can select either ***Chip Erase*** mode (the FLASH memory is completely erased before downloading) or ***Sector Erase*** mode. In Sector-Erase mode FLASHit analyses the Intel hex file that you want to upload to the target system and erases only those sectors where the program is to be saved. If the ***Erase Flash*** option is not selected, the FLASH memory will not be deleted before downloading. This is recommended if you plan on saving multiple hex files in a row. If the ***Ask before erase*** check box is selected, you will be asked to confirm the deletion of the FLASH memory.

2. ***Programming file into flash***
This option must be selected if you want to transfer a file into the FLASH memory of the target system.
FLASHit allows you to write HEX files (*.hex / *s19) as well as binary files to the FLASH memory of the target system.
To do so, click the ***Program file into flash>Load Bin*** option in the ***Config>Target and program configuration*** menu. With this setting FLASHit will no longer interpret the file to be saved in FLASH. Even Intel hex files will **not (!)** be treated as hex files, but as binary files! With this option you can "flash" data from a target system (or bitmaps, text files, etc.) into another target system.
The target system's start address where the binary file is to be saved can also be defined in this menu.

3a. Build checksum (CRC)
Select the ***Build Checksum*** option and FLASHit will build a checksum based on the content of the FLASH memory.
In the "Calculation area of checksum" section you can specify the address spaces to be used in building the checksum. The red area, for example, shows a space that contains the SFR or RAM and will therefore be hidden.
ADD32 (Add Bytes) requires about 2 Sec/256 Kbyte (lowest security level).
CRC16 (Cyclic Redundancy Check) requires about 16 Sec/256 Kbyte (highest security level). Since the system area is located in the linear address space (red field), it must be hidden during the checksum building process. You can specify the system area here.
*Note: The procedures described in this section are illustrated in Appendix **7.4**, **Sources of the Checksum Functions**.*

3b. Compare checksum with
Compares a fixed checksum with the calculated checksum or a checksum created from the upload file. You can also perform a checksum check of a specific area of the upload file (***Upload file area***).



4. Write dword (32Bit)
Selecting this option allows you to write a "dword" (4 bytes) in the variable address of the FLASH memory. The selected address of the FLASH memory must have been deleted before the word can be written.

5a. Write User serial number into flash
This option causes FLASHit to automatically generate a serial number and save it at a specified address (***At address***). The ***Prefix*** and ***Number*** input fields determine how the serial number will look like. The ***Increment...*** option specifies whether the ***Number*** field will be increased by 1 (incremented) each time the serial number is saved successfully. The two preview fields show how the following numbers will look like. If you select the ***Write number into logfile*** option, all assigned serial numbers will be saved in a log file. The name of the log file is composed of the prefix of the serial number and the *.txt extension (e.g., `hse1.txt`).

5b.Increment number after successful writing

5c. Write number into logfile

Sp .Set protection.
With the option ***Setprotection*** the possibility exists to activate Readout and or Writingprotection. With Select Sector the sectors to be protected, for Writeprotection can be selected.

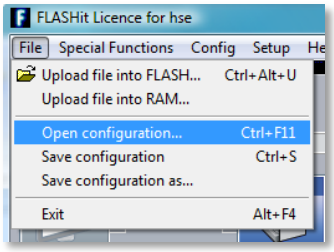
6. Reset target
The ***Software reset*** option generates a software reset in the target system once the download is complete. When the ***Hardware reset*** option is selected, FLASHit will generate a "bootstrap signal" on the DTR line of the COM interface and "reset signal" on the RTS line (see 7.5). This option allows you to reset the target system and switch to bootstrap mode provided that such a mode is recognized on the target hardware (see 7.5 and 7.6).

7. Start RS232 Terminal
Select the ***Start RS232 Terminal*** option to start a terminal program after the target system is reset that displays all the data your application transferred over the RS232 interface (the correct baud rate must be set!):

Rx term	Minimal terminal program in status window
Smal term	External terminal program (hse tool)
Ext. term	External terminal program

8. Exit FLASHit after upload sequence
With this option you can specify whether and in what way FLASHit will automatically quit after an upload.

Saving or loading Configuration Data

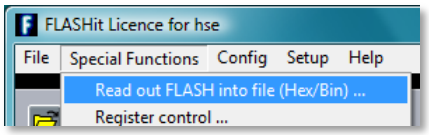


In the ***File>Open configuration*** menu you can load previously saved settings, saved them with ***Save configuration*** or save them under a new name with ***Save configuration as...***

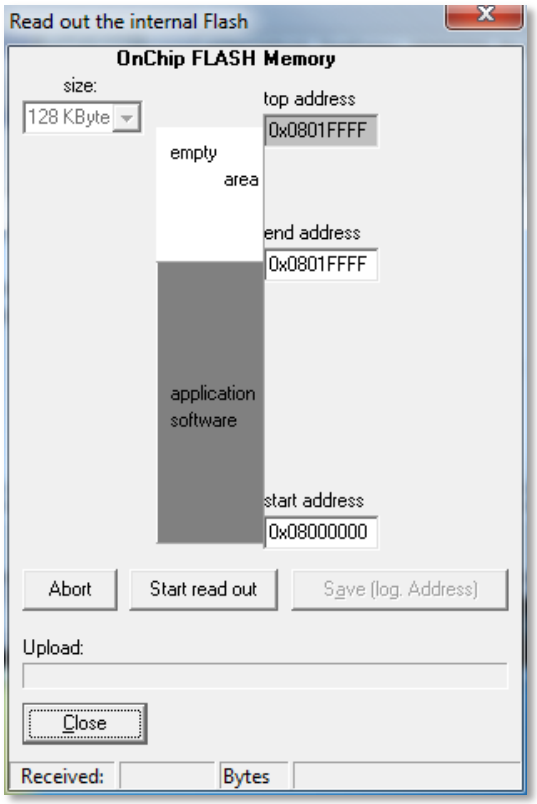
5 Special Functions

FLASHit provides an array of additional tools in the ***Special Functions*** menu item.

5.1 Reading Out the FLASH Memory



Reads out the FLASH memory and saves it as either a *.bin or *.hex file.



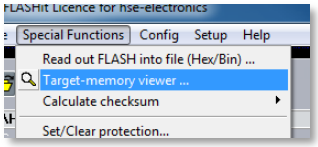
Size:
Displays the size of the FLASH memory

top address:
Shows the top memory address of the selected FLASH memory.

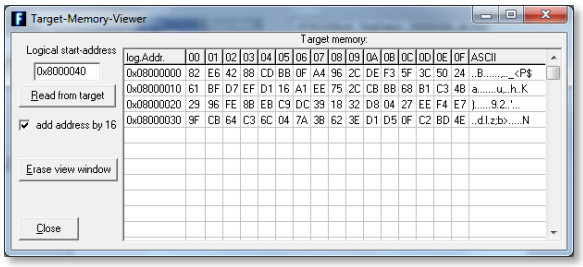
end address
End address of reading

start address:
Start address of reading

5.2 Reading Out the Content of Individual Addresses

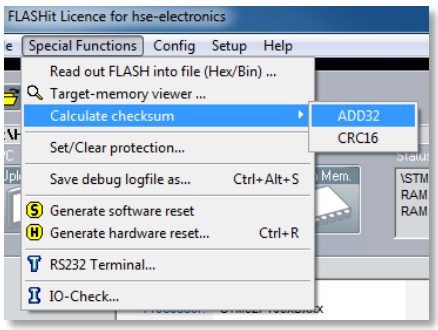


Click the ***Target-memory viewer*** menu item to read out the individual addresses from the FLASH module.



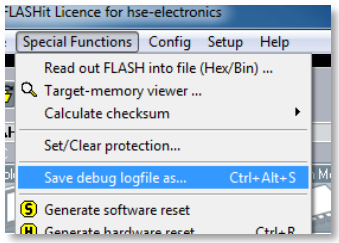
If the ***add address by 16*** check box is selected, every time you click the ***Read from target*** button the subsequent bytes will be displayed.
Click ***Erase view window*** to erase the data being displayed (but not the data in the FLASH memory!).

5.3 Build Checksum



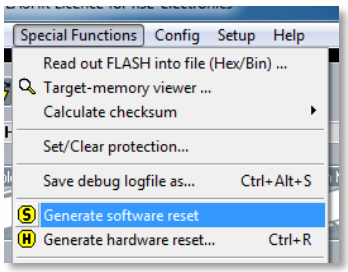
Based on the settings you selected in the ***Program sequence*** tab of the ***Config>Target and program configuration*** menu, you can click this menu item to determine the correct checksum.

5.4 Saving the Debug Log File



FLASHit logs "debug info" during the course of a session. If an error occurs, the log data are automatically saved in the `debugmemo.txt` file when you quit FLASHit.
If you click ***Save debug logfile as*** from the ***Special Function*** menu, the logged data will be saved in the directory of your choosing (for cases where no errors occur). This log file is used to perform a precise error analysis.

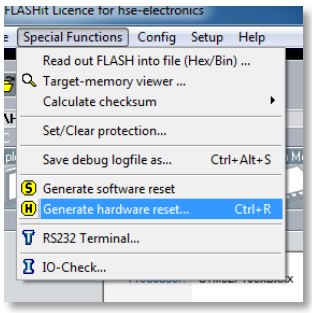
5.5 Generating a Software Reset



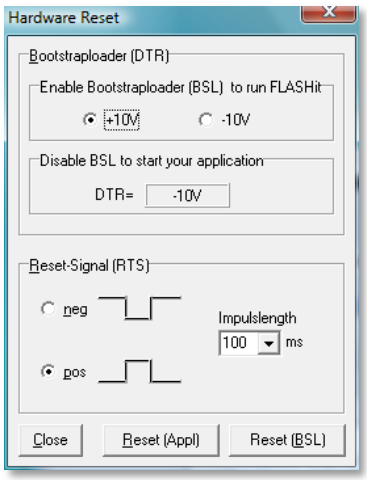
This option is a software command (SRST) that can be used to generate a reset in the target system. (Go_CMD)

Note: The software reset cannot be used to contact (boot) the target system! It can only start an application in the target system after the upload.

5.6 Generating a Hardware Reset



When the ***Generate hardware reset...*** option is selected, FLASHit will generate a "bootstrap signal" on the **DTR** line of the COM interface and a "reset signal" on the **RTS** line (see 7.5). This option allows you to reset the target system and switch to bootstrap mode provided that such a mode is recognized on the target hardware (see 7.5 and 7.6).



The form of the reset impulse can be adjusted in the ***Hardware Reset*** dialog.
The settings you make here do not affect the automatic reset mechanism of FLASHit.

5.7 Command Line Functions

The functionality of FLASHit can be controlled externally by means of command line parameters (scripts). Note that these parameters are upper and lower case sensitive! The CmdLine tool (see **6.4 Command Line Generator CmdLine**) can be used automatically generate the appropriate scripts. Command lines allow you to access and control FLASHit from a separate application with or without the desktop user interface.

Example:
1. FLASHit is to be launched via an icon on the Windows desktop.
2. A specific file is to be loaded.
3. The baud rate and the COM port are to be specified.
4. The file is to be programmed into the FLASH memory of the target system.
5. FLASHit exits automatically once the upload is complete.

Proceed as follows:
- Create a new FLASHit shortcut on the Windows desktop.
- Give the icon a name (e.g., `out.hex`).
- Open the Properties dialog from the context menu and select the ***Connection*** tab.
- Enter the following in the ***Target*** input box:

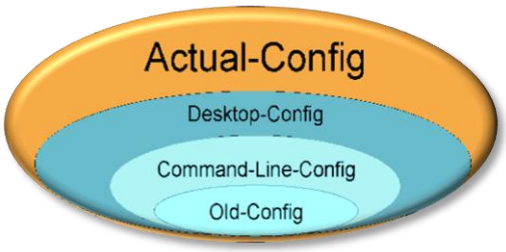
```
c:\programme\flashit_STM32.exe COM=1 BAUD=57600 HEX-FILE=e:\projekt\out.hex AUTOSTART EXIT
```

Result:
Double-click the **`out.hex`** program icon to launch FLASHit and the **`e:\projekt\out.hex`** file is transferred via COM1 at a baud rate of 57.600 to the target system where it is then programmed in the FLASH memory. FLASHit exits automatically.

Note: Chapter 7.2 provides an overview of the command line functions.

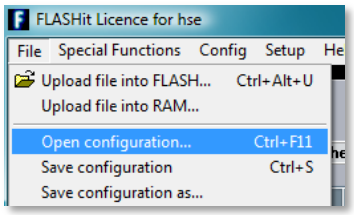
Priorities in the FLASHit Configuration

Alle vorgenommenen Parameter-Einstellungen (z. B. Baudrate, COM-Port usw.) von FLASHit werden in der Datei `flashit_STM32.ini` abgespeichert.



All the parameters you set (e.g., baud rate, COM port, etc.) in FLASHit are stored in the `flashit_STM32.ini` file.
Each valid FLASHit parameter (***Actual-Config***) is composed of:
- the "old data" from the `flashit_STM32.ini` file. (***Old-Config***)
- any "parameters" that are transferred when FLASHit starts up, for example, from another program (***Command-Line-Config***). These parameters have priority over the parameters saved in the `flashit_STM32.ini` file.
- the settings made by FLASHit directly on the desktop (***Desktop-Config***); these settings have priority over all other parameters.

5.8 Saving or loading the Configuration

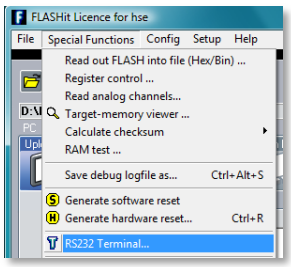


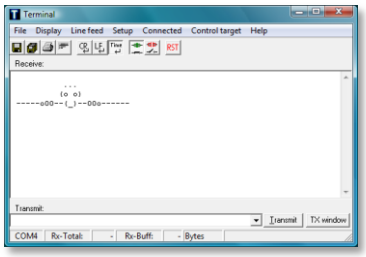
Select ***Save configuration*** to save all the settings you have made.
Select ***Open configuration...*** to load the settings.

6 The FLASHit Package

The tools described below are part of the FLASHit package and represent stand-alone programs that can be loaded externally or directly in FLASHit.

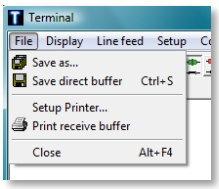
6.1 RS232 Terminal



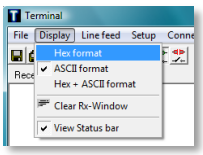


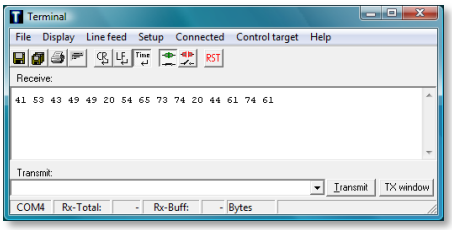
The ***RS232 terminal*** is a universal terminal program for logging data sent by the application on the target system via the RS232 interface.

Note: ***RS232 terminal*** can be launched either separately or in FLASHit.



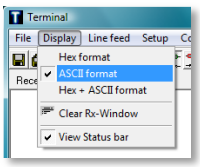
You can start all of the standard functions in the ***File*** menu.

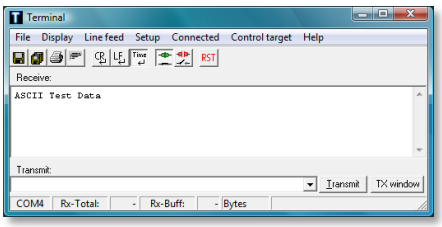




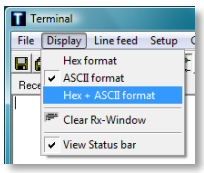
You can set the display mode in the ***Display*** menu.

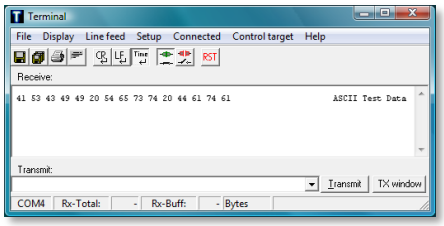
Hex format: All data are displayed in the hex format.





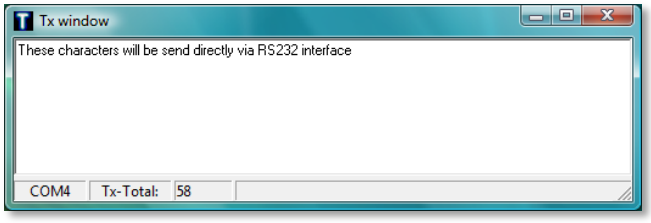
ASCII format: All data are displayed in the ASCII format.



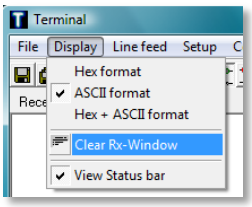


Hex+ASCII format: All data are displayed in the ASCII and hex formats.

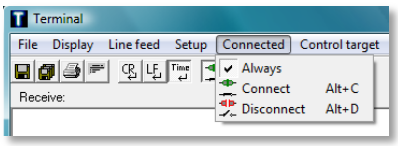
Transmit: In the Transmit list box you can select the string you want to send. Click the ***Transmit*** button to send the string via the RS232 interface.



Tx window: Click the ***TX window*** button to open a window where any text you enter will be sent "live" via the RS232 interface.

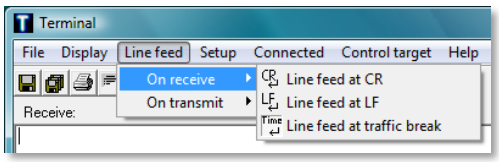


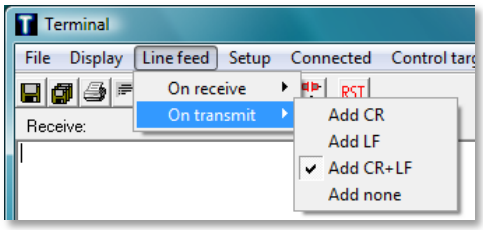
Click ***Clear Rx-Window*** to erase all data in the receive window.



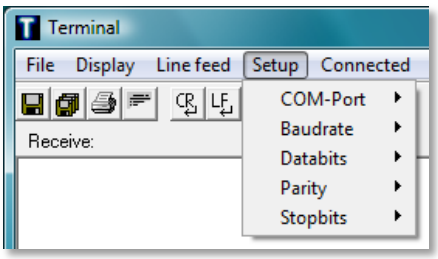
You can set the COM connection mode in the ***Connected*** menu.

The ***Line feed*** menu lets you adjust the line feed of the data in the terminal window, as well as define separate line breaks for ***receive*** and ***transmit***.

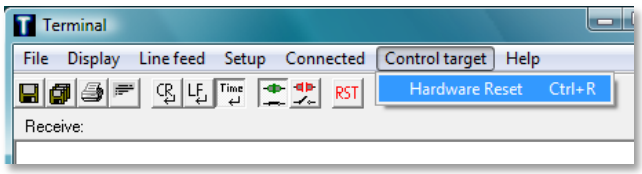




Line feed at CR:	Line feed at a carriage return.
Line feed at LF:	Line feed at a line feed.
Line feed at traffic break:	Line feed when data are absent
Add CR:	Adds a carriage return to the sent text.
Add LF:	Adds a line feed to the sent text.
Add CR+LF:	Adds a carriage return and a line feed to the sent text.
Add none:	Does not add a carriage return or a line feed to the sent text.



You can configure the standard COM port settings in the ***Setup*** menu.

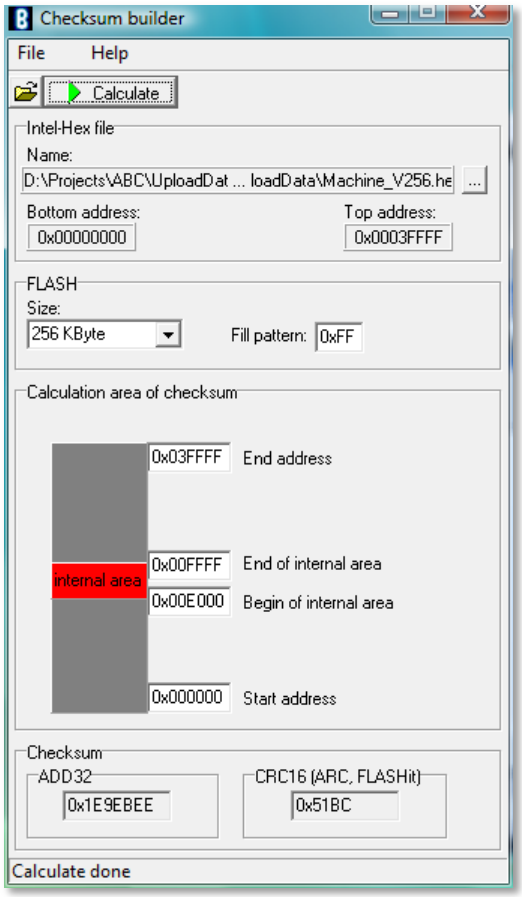


Target Reset via Terminal

The ***Control target*** menu can be used to generate a hardware reset of the target system. Before this can be done, however, the relevant connections of the RS232 interface must be wired based on the circuit suggestion (see Appendix 7.6, Reset and Bootstrap Interface).

6.2 Checksum Builder

The ***checksum builder*** is a universal program for calculating the checksum of a hex file.



You can select a hex file in the ***File*** menu. In the ***FLASH size*** list box you can specify the size of the FLASH memory.
The checksum builder defines the start and end addresses based on the ***FLASH size*** setting. You only need to specify the internal area of the target controller in the ***Begin of internal area*** und ***End of internal area*** fields since this area is factored out when calculating the checksum.
Click ***Calculate*** to start the calculation.
The resulting checksum will be indicated by the ***ADD32*** and ***CRC16*** values.

*Note: The **checksum builder** can only be launched externally.*

You can control the Checksum Builder using command line parameters.

Command	Function
EXIT	The program exits after the calculation.
A1=	Defines the start address of the checksum calculation.
A2=	Defines the start address of the system area.
A3=	Defines the end address of the system area.
A4=	Defines the end address of the checksum calculation.
FS=	Defines the flash size (0=auto, 1=64 k, 2=128 k, 3=256 k, 4=512 k, 5=1 M, 6=2 M, 7=4 M)
FP=	Defines the data content of a deleted flash module (fill pattern).
HEX-FILE=	Defines the file the checksum builder calculates.
LOCAL=	The result file is written to the program directory.

```
Hex-File =      j:\hexfiles\128kRandom.hex
CRC16 =         0x51EF
ADD32 =         0x00EDFE42l
Error Code =    000
Start-Adr =     0x000000
BegInt-Adr =    0x00E000
EndInt-Adr =    0x00FFFF
End-Adr =       0x01FFFF
```

Example of `Result_CRC.txt`

A result file (`Result_CRC.txt`) will be created when you exit the checksum builder. The file is located at `C:\Doku..Einstellungen\All users\Anwendungsdaten\Flashit` in Windows XP and `C:\PrgramData\FLASHit` in Windows Vista.

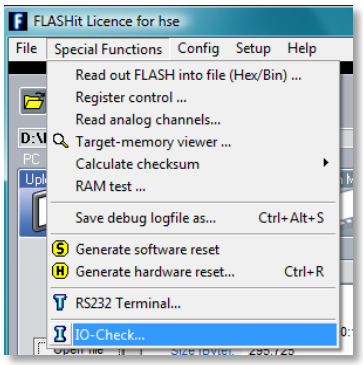
Error code	Meaning
0	Error-free execution.
14	Unable to read file.
36	The size of the flash module is unknown.
171	The file contains data outside of flash.
175	File not found

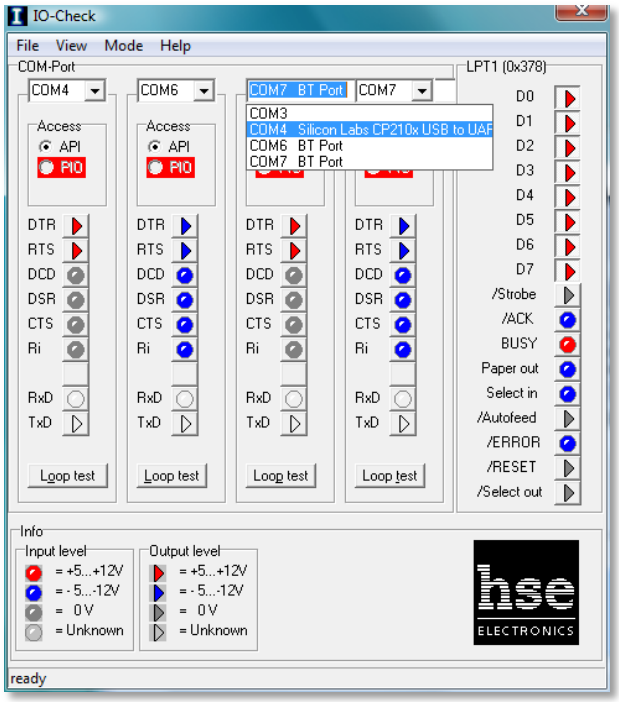
The checksum builder sends a return code back to the calling program. The return code is divided into two areas: the CRC17 checksum (Bit 0..15) and the error code (Bit 16..31).

6.3 IO Check

The ***IO check*** is an interface testing tool that is integrated in the FLASHit package and is used to check whether the COM port (the basis for working with FLASHit) is functioning properly.

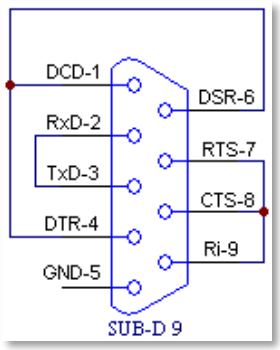
*Note: **IO check** can be launched either separately or in FLASHit.*





When the IO check function is executed, all installed COM parts will be checked to determine whether access via the Windows ***API*** function (API=Application Programming Interface) is possible. Afterwards the direct access to the PIO module (PIO mode) will be checked.
IO check shows which COM ports can be used in FLASHit. FLASHit can only use COM ports that are accessible via the API. IO check also lets you manually enable and disable individual port lines and test the LPT1port.

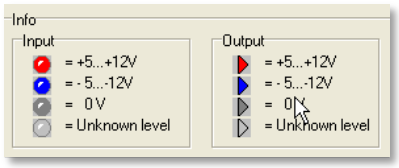
Access modes
Access to the COM port via the ***API*** is only possible if the port was free prior to starting the program.
The PIO mode is perfect for "observation" while the API mode is suited to performing function tests. The PIO mode can only be used for standard COM ports, which excludes COM ports based on UBS interfaces, for example (USB RS232 Adapter).
Loop test/line test:
Click the ***Test*** button to open a log window that displays the results of the automatic test.
To save the test results, click ***Save protocol as...*** in the ***File*** menu.
A physical test of the individual lines and signal runtimes will be performed.
To complete this test, however, you first need to insert a test plug into the COM port (see bottom left corner).



Example for line test log:

```
COM1: Start Loop Test (API mode)
TxD -> RxD Loop ok. Delay = 1.6 ms
DTR -> DSR Loop ok. Delay = 5.6 ms
DTR -> DCD Loop ok. Delay = 0.9 ms
RTS -> CTS Loop ok. Delay = 4.4 ms
RTS -> Ri Loop ok. Delay = 0.5 ms
```

The specified times are approximate values and depend on the speed of your computer. Signal times can be significantly longer with COM ports that are operated using a USB interface adaptor.



Manual check of port lines

- signals (*input*)

The colors of the symbol indicate the logical level. This means the logical level of "1" can range from approx. +5 V to approx. +12 V depending on the computer model. The corresponding negative level: approx. –5 V to approx. –12 V.

- outputs

By clicking the symbol, you can set a signal to the appropriate port.

The logical level is shown in place of the exact voltage.

WARNING: Any modification to the outputs may result in the destruction of hardware (PC and/or externally connected devices)!

Control Lines of the Printer Connection

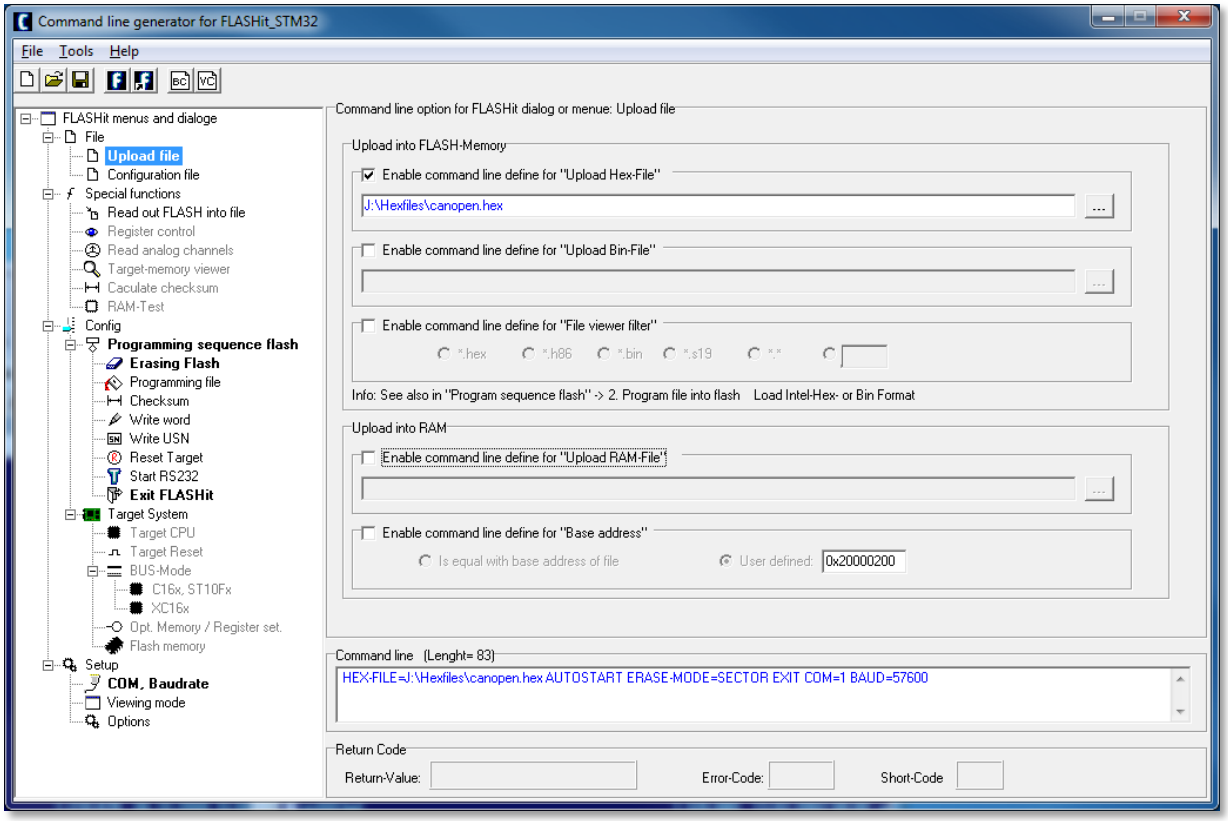
The control lines of the first default printer (address 0x378) can be modified and monitored manually.

WARNING: Any modification to the outputs may result in the destruction of hardware (PC and/or externally connected devices)!

6.4 Command Line Generator (CmdLine)

To facilitate the use of command line functions in FLASHit, the ***CmdLine*** program is included on the CD and can be used to automatically generate
- a command line and an ICON
- a C source code (for access from a separate application)

Note: The ***CmdLine*** can only be launched externally.



This window makes it extremely easy to adjust the various settings. In the example here, the upload hex file (***HEX-FILE=J:\Hexfiles\canopen.hex***) has been selected which

initiates the automatic start of the upload (***AUTOSTART***)
sets the FLASH memory erase mode to "Sector Erase" (***ERASE MODE=SECTOR***)
specifies that FLASHit will exit automatically (***EXIT***)
uses the COM port (***COM=1***)
sets the baud rate (***Baud=57600***)

The menus highlighted in bold contain settings that have been changed.
The sequence of the individual commands has no effect!


This button lets you launch FLASHit directly from the specified command line.


This button lets you create shortcuts with FLASHit and the specified command line.


This button lets you access FLASHit from a Visual C++ code.


This button lets you access FLASHit from a Borland C code.

FLASHit returns the "return code" which is then displayed by CmdLine.

With the following program lines you can extract the "error code" from the ***return value*** (see 7.1).

```
uiErrorCode = (unsigned int) ( ulReturnValue >> 8 );
```

With the following program lines you can extract the "short code" from the ***return value***.

```
cShortCode = (char)( ulReturnValue & 0x0F );
```

Short code equals bit 4 from the ***return value*** as shown below.

0	No error
1	Error in hex file
2	Error in target system
3	Flash was not found
4	Flash type not supported as yet
5	Error while deleting the flash module
6	Error while programming the flash module
7	Checksum error
8	Error in program flow of FLASHit
other	error in Windows

FLASHit can also generate a "return code file" (result.txt) for evaluation by another application.

Note: If you want to work with CmdLine, then it must be located in the FLASHit directory!

Subject to change!

"We hope your work is a success"

The hse-electronics Team

hse-electronics GmbH
Schauenburger Str. 116
D-24118 Kiel
fon: +49(0)431-570 937 20
fax: +49(0)431-570 937 23
email: info@hse-electronics.com
web: www.hse-electronics.com

7 Appendix

7.1 Status Messages (Errorcodes)

Versions			Error Number	Smal return Code	Message	Description Beschreibung
	ARM Cortex M3					
					File, Modul, Libarys Not found	
			E010	8	E010: Can't find file: *.mod	Modul-File not found Modul nicht gefunden
			E011	8	E011: Modul file <name> was not found	File is missing Datei fehlt
	V		E014	1	E014: Can't open upload file	Access to upload file is not possible Der Upload file konnte nicht geöffnet werden
	V		E015	8	E015: No FLASH data found	FLASH data missing Es fehlen Daten zum Flashen
	V		E016		E016: Can't generate Binray-Temp-File	Access to temp file (readout.bin) not possible Temporäre Datei readout.bin konnte nicht erstellt werden
	V		E017		E017: Can't open Binray-Temp-File	Read access to temp file not possible Temporäre Datei readout.bin konnte nicht geöffnet werden
	V		E018		E018: Can't generate Binray-File	Access to bin file not possible Binäre-Ziel-Datei konnte nicht geöffnet werden
	V		E019		E019: Can't generate Intel-hex-file	Access to hex file not possible Intel-Hex-Ziel-Datei konnte nicht geöffnet werden
			E020	8	E020: Library unit not found	FLASHitx.lib not found Flashit.lib konnte nicht geöffnet werden.
			E023		E023: Wrong in flashitx.lib	Error in Library x Fehler in Bibliothek x
	V		E024		E024: Instruction file not found	PDF files not found Die Datei doku/Anleitung.pdf fehlt
	V		E025		E025: Data base FLASHdat.LIB not found	File FLASHdat.LIB not found Die Datei FlashDat.lib wurde nicht gefunden
	V		E026		E026: Ini-File is write protected	Flashit.ini is write protection Die Ini-Datei ist schreibgeschützt
					Programmcycle	
			E030	8	E030: Modul-file <name> is too big!	File size is too big. Ein Modul ist zu groß
	V		E031	8	E031: No memory for FlashLib	No memory to read Flashitx.lib Kein dynamischer Speicher verfügbar um Flashdat.lib zu laden
	V		E032	8	E032: Not enough memory	No memory to read Flashitx.lib Kein Speicher für das virtuelles Flash
	V		E033	1	E033: Error in Hex-File: Line is too long!	Hex format error Aktuelle Zeile im Intel-Hex-File ist zu lang
	V		E034	1	E034: Wrong file format. Function abort	Hex format error *Unbekanntes Datenformat der Hex-Datei*
	V		E035	8	E035: Write mode n unknown	Flash write mode is not supported Flash-Write-Methode unbekannt
	V		E036	1	E036: Size of FLASH is unknown. Abort	Size of embedded flash is not found Größe des Flashspeichers ist unbekannt
	V		E038	6	E038: Can't write User-Serial-Number [..] at log. address ...	Error in embedded flash. User-Serial-Number konnte nicht geschrieben werden.
	V		E039	1	E039: Mirror address is unknown	Undefined size of embedded flash Größe des Flashspeichers ist unbekannt

Not supported						
	V		E040	8	E040: Sorry this CPU is not enabled at your version	Target CPU (step) is unknown CPU nicht freigeschaltet (nur bei limitierter Version)
	V		E042		E042: Internal error call hse	FLASHit access error Interner Fehler
	V		E043		E043: Wrong Date (1)	Date is out of format Datum kann nicht korrekt ermittelt werden.
	V		E046	8	E046: Please contact hse-elctronics	User access error Fehler der eine Kontaktaufnahme zu hse erforderlich macht
	V		E047	8	E047: Function in demo not available	Licence restriction Funktion ist in der Demoversion nicht verfügbar.
	V		E048	8	E048: Contact hse-elctronic	Upload data access error Fehler der eine Kontaktaufnahme zu hse erforderlich macht
Dataformat Error						
			E051	8	E051: Try to write odd number of data!	Try to write odd number of data Es wird versucht eine ungerade Anzahl von Bytes zu schreiben.
	V		E052	1	E052: Upload-file is no Intel-Hex format!	Upload file format error Die Upload-Datei entspricht nicht dem Intel-Hex Format
	V		E053	1	E053: Overwrite Address	Multiple define of data in Embedded upload file Fehler im Hexfile
	V		E056		E056: Data outside of FLASH at Address x	Address of embedded file is out of FLASH memory Daten außerhalb des Speicherbereiches
	V		E058	1	E058: Checksum error in Hexfile line	Checksum error in Hexfile line Prüfsummenfehler in Hexfile Zeile
	V		E059	1	E059: Wrong blocklenght in Hexfile line	Wrong blocklenght in Hexfile line Anzahl der Daten stimmt nicht mit Länge der Hex-Zeile überein
			E060		E060: No Debug/Toolstick Adapter found	No Debug/Toolstick Adapter found Toolstickadapter nicht gefunden
Target Error						
			E101	2	E101: Can't load modul	Can't load modul Fehler beim Laden eines Moduls ins Target
			E102	2	E102: Booting was not possible %d	Target booting was not possible Target konnte nicht gebootet werden
			E103	2	E103: No correct answer from target	Target does not answer correct Das Target antwortet nicht korrekt.
	V		E104	2	E104: Communication with target failed	Target gives an unknown response Das Taget meldet eine unbekannte CPU-Kennung.
			E109	2	E109: Target-bus length unknown!	Unknown bus mode Busbreite des Target-Systems konnte nicht erkannt werden
	V		E110	2	E110: Target crashed - rebooting...	Target crashed while erasing Target ist beim Sektor-Löschen abgestürtzt und wird neu gestartet.
	V		E111	2	E111: No response from target	Target does not answer correct Target antwortet nicht richtig
	V		E113	2	E113: Can't analyse contents of target FLASH	Can't analyse contents of target FLASH Die Analyse des Inhaltes des Target Flashes war nicht möglich.
			E114	2	E114: Can't load buffer	Can't load buffer Ein Bufferinhalt konnte nicht ins Target geladen werden
			E116	3	E116: FLASH-Read-Test failt	FLASH-Read-Test failt Der Flash-Lese-Test ergab ein ungültiges Ergebnis

Flashmemory Error						
			E130	6	E130: Timeout while writing FLASH!	Error in embedded flash.Timeout Fehler beim Schreiben des Flashbausteines. Timeout
			E131	6	E131: While writing FLASH (DQ7)	Error in embedded flash Fehler beim Schreiben: Flashbaustein meldet Fehler
			E132	6	E132: No. %d in modul Write_Buffer!	Error n in embedded flash Fehler beim Schreiben: Flash-Speicher meldet Fehler n
			E133	6	E133: No acknowledge while writing FLASH!	Target does not answer Fehler beim Schreiben: Es kommt keine Antwort von Target
			E134	6	E134: Abort at Adr. %08lXh, while Timeout-Error!	Programming stop at address x Abbruch beim Schreiben an Adresse x wegen Timeout
	V		E135	6	E135: Abort at Adr. %08lXh, Error%d!	Programming stop at address x Abbruch beim Schreiben an Adresse x wegen Fehler n
	V		E136		E136: Erase mode (Chip/Sector) unknown	Erase mode (Chip/Sector) unknown Löschmethode unbekannt
			E137	1	E137: Wrong page size	Wrong page size Blocklänge nicht zulässig
	V		E138	6	E138: Flash type is not found	No data about embedded flash Keine Daten über den FLASH-Speicher verfügbar
			E140	5	E140: FLASH-Erase abort while timeout!	Target does not answer: Timeout Fehler beim Löschen des Flashbausteines: Timeout
			E141	5	E141: Sector-Erase failt!	Error at erasing target flash Es trat ein Fehler beim Löschen des Flashes auf
			E142	5	E142: Sector-Erase abort while timeout!	Target does not answer: Timeout Fehler beim Löschen des Flashbausteines: Timeout
	V		E145	5	E145: FLASH-Erase abort after x s from flash	FLASH Erasing was canceled after x sec by target flash Flash hat nach x Sek einen Fehler erkannt
			E146	5	E146: Chip-Erase not possible	Mode Chip-Erase not possible Diese CPU kennt kein Chip-Erase
	V		E150	7	E150: Checksum compare error	CRC of target flash does not match Vorgegebene Checksumme stimmt nicht mit Flashinhalt überein
			E151	5	E151: Internal flash is not supported	Not support flash found Dieses interne Flash wird nicht unterstützt
	V		E153	8	E153: Abort at Adr. %08lXh, Error%d!	Error while writing a "Word/DWord" at address x Fehler beim Schreiben eines "Word/DWord" an Adresse x
Input Error						
			E170	8	E170: Odd address is not possible	To enter an odd address is not allowed Ungerade Adresseingabe ist nicht erlaubt
	V		E171	8	E171: Address combination is not possible (A1>A4)	Address combination is not allowed Adresskombination nicht erlaubt
	V		E172	8	E172: Command line: Pfad to hex-file does not exist	The upload file path of command line, does not exist Die Upload-Datei, ist nicht vorhanden
			E173	8	E173: Address combination is not possible A(n)>A(n+1)	Address combination is not allowed Adresskombination nicht erlaubt
	V		E175	1	E175: Hexfile not found	Upload file not found Upload Datei nicht gefunden
			E176	8	E176: Do not mapp ROM1 to seg 1	Bus setting of SYSCON.15 is wrong Bus Konfiguration SYSCON.15 ist falsch
	V		E177	8	E177: Command line: Pfad to RAM-file does not exist	The upload file path of command line, does not exist Die Datei ist nicht vorhanden
	V		E200	4	E200: sizeof(FLASHdat.LIB) too big	File FLASHdat.LIB size too big Die Datei FLASHdat.LIB ist zu gross
	V		E201		E201: Hex-Download Abort, lost Sync	Error while readout target FLASH memory. Beim Auslesen des Flashes ist ein Fehler aufgetreten
	V		E202:	8	E202: Lenght too high	Path of upload file is too long

						Der Upload Pfad ist zu lang
COM-Access						
	V		E301		Error Baudrate not possible	Baudrate can not be set Baudrate kann nicht eingestellt werden
RAM-Access						
			E410	2	E410: RAM-access test failed	Access to RAM failt RAM-Zugriffstest ist fehlgeschlagen
			E420	2	E420: RAM data line test failed	Access to RAM failt RAM-Test Fehler in Datenleitung
			E430	2	E430: RAM addr line test failed	Access to RAM failt RAM-Test Fehler in Adressleitung
			E440	2	E440: RAM-cell test failed	Access to RAM failt RAM-Test Fehler bei Zellen Test
			E450		E450: RAM upload failed, Ex	Error x while RAM upload Fehler x beim RAM upload
	V		E451	2	E451: Upload failed, CRC is wrong	CRC error while RAM upload CRC Fehler während RAM Upload
			E452	2	E452: RAM-Data compare error	Read back compare RAM-Rücklese-Daten stimmen nicht mit Upload Daten überein
	V		E453	2	E453: RAM CRC check failed	Read back CRC error RAM-CRC stimmt nicht mit Upload CRC überein
	V		E455	2	E455: can not open file	RAM-Upload file access error RAM-Upload Datei kann nicht geöffnet werden
			E456		E456: Data will destroy system area	Data will overwrite iRAM, SFR Versuch ins iRAM oder SFR zu schreiben
	V		E457		E457: Error Vector not possible	The jump vector is not possible Das Sprungziel ist nicht möglich
Problem with Licence						
	V		E814		E814: Licence not valid (version)	Licence does not match to program version Lizenz passt nicht zur Programm Version
	V		E815		E815:Licence not valid (timeout)	Validity periode expried Gültigkeitszeitraum abgelaufen
	V		E816		E816: Licence not valid (timeout)	Validity periode expried Gültigkeitszeitraum abgelaufen
	V		E817		E817: Licence not valid (timeout)	Validity periode expried Gültigkeitszeitraum abgelaufen
	V		E818		E818: Licence wrong (error)	Tiping error Tippfehler
	V		E819		E819: Empty USB-Key	USB-Key is empty (no licence data) USB-Dongle enthält keine Lizenz-Daten
	V		E820		E820: Null nicht erlaubt	Tiping error Tippfehler
Problem with Licence-Dongle						
	V		E900		E900	No licence found / Demo version Keine Lizenz gefunden / Demo Version
	V		E934	8	E934: USB-Dongle is broken!	USB-Dongle is broken USB-Dongle ist gebrochen

7.2 Supported MCUs

STM32F100x	STM32F101x	STM32F102x	STM32F103x	STM32F105x	STM32F107x
STM32F100C4	STM32F101C4	STM32F102C4	STM32F103C4	STM32F105R8	STM32F107RB
STM32F100C6	STM32F101C6	STM32F102C6	STM32F103C6	STM32F105RB	STM32F107RC
STM32F100C8	STM32F101C8	STM32F102C8	STM32F103C8	STM32F105RC	STM32F107VB
STM32F100CB	STM32F101CB	STM32F102CB	STM32F103CB	STM32F105V8	STM32F107VC
STM32F100R4	STM32F101R4	STM32F102R4	STM32F103R4	STM32F105VB	
STM32F100R6	STM32F101R6	STM32F102R6	STM32F103R6	STM32F105VC	
STM32F100R8	STM32F101R8	STM32F102R8	STM32F103R8		
STM32F100RB	STM32F101RB	STM32F102RB	STM32F103RB		
STM32F100V8	STM32F101RC		STM32F103RC		
STM32F100VB	STM32F101RD		STM32F103RD		
	STM32F101RE		STM32F103RE		
	STM32F101RF		STM32F103RF		
	STM32F101RG		STM32F103RG		
	STM32F101T4		STM32F103T4		
	STM32F101T6		STM32F103T6		
	STM32F101T8		STM32F103T8		
	STM32F101V8		STM32F103V8		
	STM32F101VB		STM32F103VB		
	STM32F101VC		STM32F103VC		
	STM32F101VD		STM32F103VD		
	STM32F101VE		STM32F103VE		
	STM32F101VF		STM32F103VF		
	STM32F101VG		STM32F103VG		
	STM32F101ZC		STM32F103ZC		
	STM32F101ZD		STM32F103ZD		
	STM32F101ZE		STM32F103ZE		
	STM32F101ZG		STM32F103ZF		
			STM32F103ZG		

7.3 Overview of Command Line Functions

Commands Befehle	Description Beschreibung	Adjustments Einstellungen	Examples Beispiele
	Project		
INI_FILE	Starts FLASHit with project file (*.ini). FLASHit mit Projekteinstellungen (*.ini) starten		INI_FILE=c:\demo.ini INI_FILE="c\program files\test.ini"
	Target-Connection		
COM	Defines COM-Port number Definiert COM-Port Nummer	1, 2, ..., 99	COM=2
BAUD	Defines baudrate Definition der Baudrate	9600,19200,38400, 57600,115200	BAUD=38400
	Target defines		
BASIC-ADDR	Flash basic address Basis Adresse des Flash-Speichers.	0x000000 - 0xFFFFFFFF	BASIC-ADDR=0x08000000
RDOUT-BEG	Beginn of Readout niedrigste Auslese-Adresse		RDOUT-BEG=0x08000000
RDOUT-TOP	End of Readout höchste Auslese-Adresse		RDOUT-TOP=0x0801FFFF
	Flash Upload		
FILTER	Defines filter for file view Definiert den Filter für die Hexfile-Anzeige	*.hex, *.h86, *.s19	FILTER=*.H86
HEX-FILE	Defines a path and file name for upload Pfad-Angabe zum Upload-Hexfile	Fullpath	HEX-FILE=c:\new\out.hex HEX-FILE="c:\new 1\out.s19"
BIN-FILE	Defines a path and file name for upload Pfad-Angabe zum Upload-Binary-File	Fullpath	HEX-FILE=c:\new\out.bin HEX-FILE="c:\new 1\out.jpg"
BIN-OFFSET	Load Bin-File with offset into Flash Lädt Binär-Datei mit Offset-Adresse in den FLASH-Speicher		BIN-OFFSET=0x08000000
FILELOAD	Loading mode (bin/Intel-/Mot.-Hex) Datei Lademethode	BIN, HEX	FILELOAD=HEX
	RAM Upload		
RAM-FILE	Defines a path and file name for RAM upload Pfad-Angabe RAM Uploaddatei	Fullpath	RAM-File="c:\Test\RAM.hex"
RAM-OFFSET	Offset for RAM upload Offset für Speicherung ins RAM		RAM-OFFSET=0x2000000
RAM-UPBASE	Set offset source Gibt die Quelle der Offset-Adresse an	0 = Hexfile base address 1 = User defines	RAM-UPBASE=1
	Visible		
ICONSIZE	Starts FLASHit as an icon Startet FLASHIt als Icon		ICONSIZE
EXPERTVIEW	Start full view Startet FLASHit mit voller Oberfläche		EXPERTVIEW
EXPRESSVIEW	Start smal view Startet FLASHIt mit reduzierter Oberfläche		EXPRESSVIEW
	User Serial Number (USN)		
USN_Prefix	Defines the prefix of the USN Definiert den Präfix der USN	max. 10digits	USN_Prefix=Version:
USN_Number	Defines the value of the USN Definiert die Wert der Seriennummer	max. 10digits	USN_Number=123
USN_Adress	Defines the start-adress of the USN Startadresse ab der die USN ins Flash schrieben wird	max. 10digits	USN_Adress=0x0803FFF0
USN_INC	Defines autoincrement of the USN Auto-linkrementierung der USN		USN_INC
USN_Log	Safes the USN into a file Speicherung der USN		USN_Log

	Program sequency / Programm-Schritte		
SEQUENCE	Clear Readout/Write protection	0p	SEQUENCE=0p
	Readout/Write Protection löschen		
	RAM upload	0r	SEQUENCE=0r
	RAM Upload		
	Erasing Flash	1	SEQUENCE=1
	FLASH-Speicher löschen		
	Programming file into flash	2	SEQUENCE=1;2;6;8
	Datei in den FLASH-Speicher programmieren		
	Build checksum (CRC)	3a	SEQUENCE=1;2;3a;8
	Checksumme bilden		
	Compare checksum with	3b	SEQUENCE=1;2;3a;3b;8
	Checksumme vergleichen mit einem festen Wert		
	Write dword (32 bit) at address	4	SEQUENCE=4;8
	Ein dword an Adresse schreiben		
	Write User serial number into flash (USN)	5a	SEQUENCE=1;2;5a;8
	Automatisch generierte Seriennummer schreiben (USN)		
	Increment USN-number after successful writing	5b	SEQUENCE=1;2;5a;5b;8
	USN-Nummer Inkrementieren nach erfolgreichem Schreiben		
	Write USN into logfile	5c	SEQUENCE=1;2;5a;5b;5c;8
	USN in log Datei speichern		
	Reset target	6	SEQUENCE=1;2;6;8
	Target reseten		
	Start RS232 Terminal	7	SEQUENCE=1;2;7
	RS232-Terminalprogramm starten		
	Exit FLASHit after upload sequence	8	SEQUENCE=1;2;6;8
	FLASHit beenden		
	Set Readout/write Protection	sp	SEQUNCE=0p;1;2;sp
	Setzen der Readout/Write Protection		
	Program setting		
ERASE-MODE	Defines SECTOR- or CHIP- NO-erase before programming	SECTOR, CHIP, NO	ERASE-MODE=SECTOR
	Definition des Lösch-Modus		
AskUseBeforeErase	Stops and ask user before sector erase	0, 1	AskUseBeforeErase=0 AskUseBeforeErase=1
	Abfrage ob der Anwender Änderungen durchführen will		
CHECKSUM=ADD	Calculates a 32 bit add up Checksum after download		CHECKSUM=ADD
	Als Prüfsumme wird die 32bit Summe aller Bytes verwendet		
CHECKSUM=CRC16	Calculates a CRC16 checksum after download		CHECKSUM=CRC16
	Als Prüfsumme wird die CRC16-Methode verwendet		
CMPCHK	Compares Checksum with value		CMPCHK=0x002345
	Definiert den Vergleichswert der Prüfsumme		
SOFTRESET	Generates a software reset after successful programming		SOFTRESET
	Definiert die Reset-Methode Software-Reset		
HARDRESET	Generates a hardware reset after successful programming		HARDRESET
	Definiert die Reset-Methode Hardware-Reset		
AutoEraseSec	Erases sector at address		AutoEraseSec=0x08007FF0
	Definiert den Sektor mit der Adresse zum Löschen		
AutoWriteWord	Writes a dword at address x		AutoWriteWord= 0x08007FF0,0x12345678
	Schreibt ein 32bit Wert, an die Adresse x		
	Program control		
EXIT	Exits FLASHit after successfull program sequence		EXIT
	Beendet FLASHit nach erfolgreicher Sequenz		
RETURN	Exits FLASHit allways after program sequence		RETURN
	Bendet FLASHit in jedem Fall. Bei Fehler siehe Returncode		
AUTOSTART	Starts automaticaly program sequence		AUTOSTART
	FLASHit startet automatisch die Sequenz		
WORK_DIR	Path to work directory (ini, result)	Fullpath	WORK_DIR=j:\ini WORK_DIR="c:\program files\flashit"
	Pfad zur ini-, result, Datei		

7.4 Sources of Checksum Function

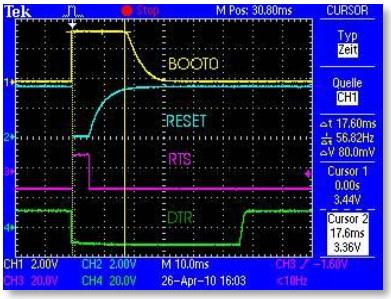
ADD32 (Add Bytes)

```
unsigned int ChecksumADD32( unsigned long ulA1, unsigned long ulA2,
unsigned long ulA3, unsigned long ulA4 )
{
unsigned int uiChecksumADD32 = 0;
unsigned long ulAddress;

// address range
for (ulAddress = ulA1; ulAddress <= ulA4; ulAddress++)
    {
    //Check for switch address to jump over system area
    if ( ulAddress == ulA2 ) ulAddress = ulA3 + 1;
    //Checksum function
    uiChecksumADD32 += *(volatile huge unsigned char*)  ulAddress;
    }
return( uiChecksumADD32 );
}
```

CRC16 (Cyclic Redundancy Check)

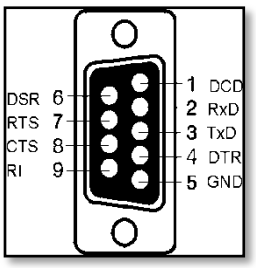
```
unsigned int ChecksumCRC16(unsigned long ulA1, unsigned long ulA2,
unsigned long ulA3, unsigned long ulA4 )
{
unsigned int uiCRC6 = 0;
unsigned char ucData;

for (ulAddress = ulA1; ulAddress <= ulA4; ulAddress++)
    {
     // Check for switch address to jump over system area
     if ( ulAddress == ulA2 ) ulAddress = ulA3 + 1;
     ucData = *(volatile huge unsigned char*)  ulAddress;

     // Checksum function
     for (i=0; i<8; i++)
         {
         if ((uiCRC6 ^ ucData) & 1) uiCRC6=(uiCRC6>>1)^0xA001;
         else uiCRC6 >>=1;
         ucData >>=1;
         }
     }
return( uiCRC6 );
}
```

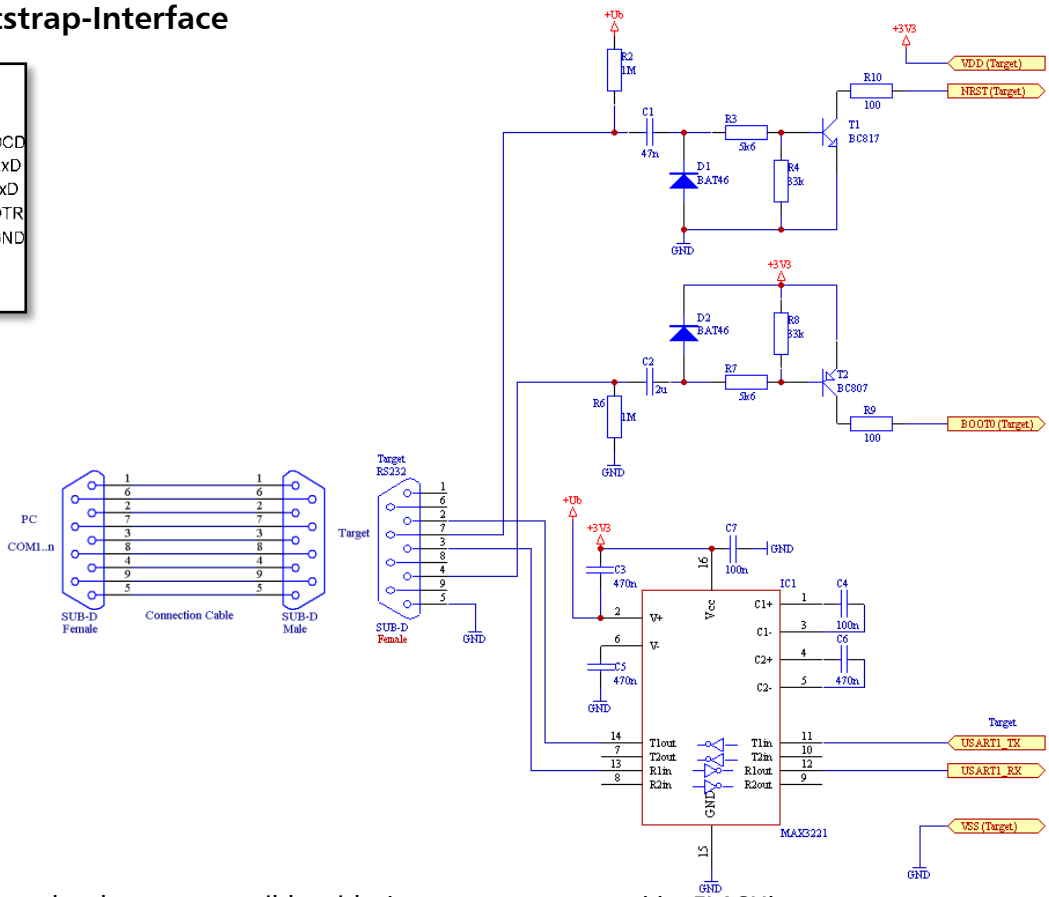
7.5 Reset- und Bootstrap-Signals



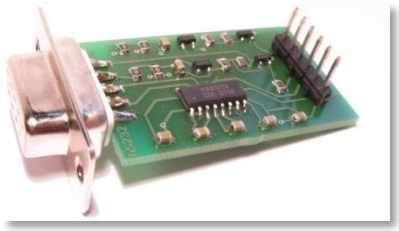
Reset and bootstrap signal on the COM interface Signal based on circuit in 7.6

7.6 Reset- und Bootstrap-Interface





Basic circuit! for making your hardware compatible with the reset concept used by FLASHit.



The reset and bootstrap interface can be purchased from hse-electronics and comes preassembled.

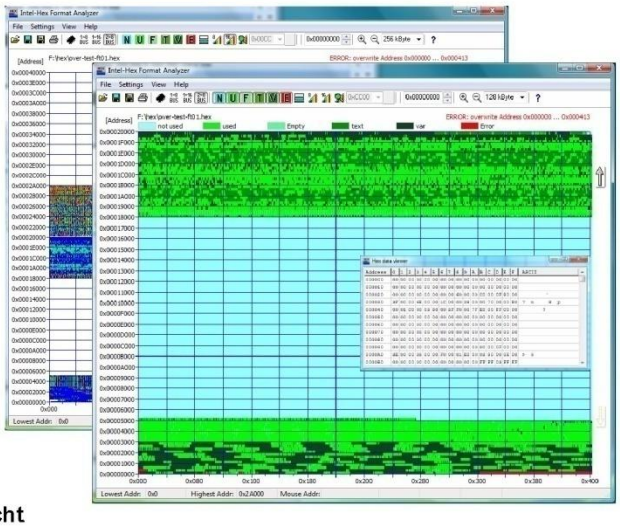
Subject to change!
No guarantee of completeness!

8 hse-elctronics product: HEXit the HEX-File Analysis-Tool

Das HEX-File Analyse-Tool

HEXit

Den Output von Compilern (HEX-Files) komfortabel verifizieren



Kommt Ihnen folgendes Problem bekannt vor? Sie haben ein Projekt kompiliert. 0 Errors und 0 Warnings. Im Debugger funktioniert alles. Nach dem Laden der Applikation in den Flash-Speicher des Targets funktioniert jedoch nichts.

Was ist passiert? Jetzt kommt HEXit ins Spiel: HEXit analysiert nun Ihre HEX-Datei auf Speicherbelegung, Adressverteilung, Verteilung von Programmcodes/-daten sowie auf mehrfach belegte Adressen. Das Ergebnis wird grafisch dargestellt. So können etwaige Probleme sehr schnell und deutlich erkannt werden. Im eingangs genannten Problem-Beispiel könnte sich z.B. herausstellen, dass der Reset-Vektor nicht belegt worden ist, oder dass der eingestellte Adressbereich nicht dem Hardware-Design entspricht

HEXit Features:

- **Windows-Versionen** von 9x bis Vista werden unterstützt
- **Intel-HEX-Dateien sowie Binär-Dateien** können analysiert werden
- **Bin2Hex-Wandlung** von Binär-Dateien in eine oder zwei Intel-HEX-Dateien, unter Berücksichtigung der Adressbereiche
- **Hex2Bin-Wandlung** von Intel-HEX-Dateien oder Teilen davon in Binär-Dateien
- **Grafische Analyse** der Speicher-belegung von Intel-Hex-Dateien. Diverse Filter ermöglichen eine übersichtliche grafische und textliche Darstellung der Daten (siehe Screenshot)
- **Prüfsummenbildung** über eine Polynomenauswahl ermöglicht die Berechnung verschiedener CRC über mehrere Adressbereiche
- **Splitten** von Datenbereichen aus einer Intel-Hex-Datei in zwei neue Dateien (z.B. zur Aufteilung der Datei für interne- und externe Flashspeicher)
- **Linken** von mehreren Intel-Hex-Dateien zu einer neuen Datei. (z.B. zur Pro-gramm- und Daten-Zusammenführung)
- **Datei-Generator** zum Erzeugen von Testdateien, wie Rampen, Pattern und Zufallszahlen. Kann auch zur Erzeugung von Jump-Tables eingesetzt werden
- **C-Include-Wandler** zum Einbinden von Dateien in C-Sourcen
- **Batch Betrieb** zur Steuerung der HEXit-Funktionen über Batch-Anweisungen
- **Weltweiter Einsatz** in der TV/Video -, Automobil-, Messgeräte- und Industrie-maschinen-Branche
- **Demo-Version** steht unter http://www.hse-electronics.de bereit

hse-electronics GmbH - the competence in embedded systems
Schauenburger Str. 116, 24118 Kiel, Germany
Fon +49(0) 431 57093720, FAX +49(0) 431 57093723, info@hse-electronics.de, www.hse-electronics.de