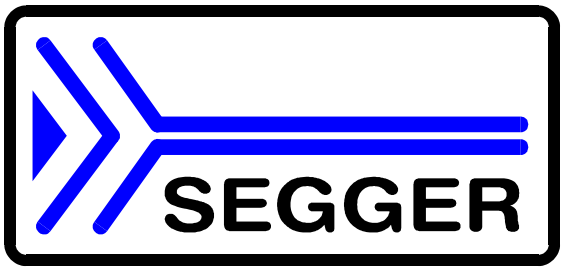
FLASHER 5 PRO

**Programming tool for
serial in circuit programming
of microcontrollers
with on-chip flash**

**Software version 2.00x
Hardware rev. 1.2
Manual rev. 1**



www.segger.com

A product of SEGGER Microcontroller GmbH & Co. KG

Content

Applicable microcontrollers

Serial mode:

M16C/1N series
M16C/26 series
M16C/24 series
M16C/26; M16C26A series
M16C/28; M16C28B series
M16C/29 series
M16C/56 series
M16C/57 series
M16C/57 series
M16C/62 series
M16C/62P series
M16C/64 series
M16C/65 series
M16C/6N series
M16C/6S series
M16C/6V series
M16C/80 series
M32C/83; M32C/84; M32C/85; M32C/86; M32C/88; M32C/95 series
R8C/14; R8C/15; R8C/18; R8C/19; R8C/22; R8C/23; R8C/24; R8C/25; R8C/26; R8C/27; R8C/28; R8C/29; R8C/2C; R8C/2D, R8C/3x series
R32C series
M79 series

For detailed information on supported flash chips please check www.segger.com/flashmcus.htm

Features

- Easy to use windows program
- Serial in target programming supported
- Programming / Clearing / Verifying / Read back supported; High speed programming
- User or boot area selectable (read only in serial mode)
- 64 MByte FLASH to store target program
- Can be used in a production environment
- PC Program for batch mode processing, allowing usage in automated test systems.
- Remote control functions for automated testers

Working environment

General

Flasher has been designed for use in a lab only.

Host System

IBM PC/AT or compatible. CPU: 486 (or better) with at least 8Mb of RAM, running Windows 95 / 98 / 2000 / Vista / Windows 7 or NT. It needs to have an RS232 interface available for communication with FLASHER.

Power supply

Flasher requires a 5V power supply over its USB connector, minimum current consumption is about 80mA. You may use the power supply which comes with the tool or may power it from a USB port from your PC. Flasher is NOT protected against polarity reversion on the supply input. Please avoid excess voltage over 6.0V.

FLASHER can be supplied from the target, if the target is operated from 4.5 to 5V.

Installing FLASHER PC-software

The latest PC software FLASHER.EXE is available from the download pages on our website: http://www.segger.com. In order to use the software, simply copy it onto your hard drive and start the executable which will guide you through the installation process.

Connecting FLASHER to the PC

PC <-> FLASHER interface cable

A standard serial 1 by 1 interface cable with one male and one female 9 pin SUB-D9 connector can be used to connect FLASHER to the PC. The pin assignment of the 9 pin SUB-D female RS-232 interface connector is as follows:

Pin no.	Signal	Function	Host-Signal
2	TxD	Serial async. data input	Serial data output (TxD)
3	RxD	Serial async. data output	Serial data input (RxD)
5	GND	Signal ground	Signal ground

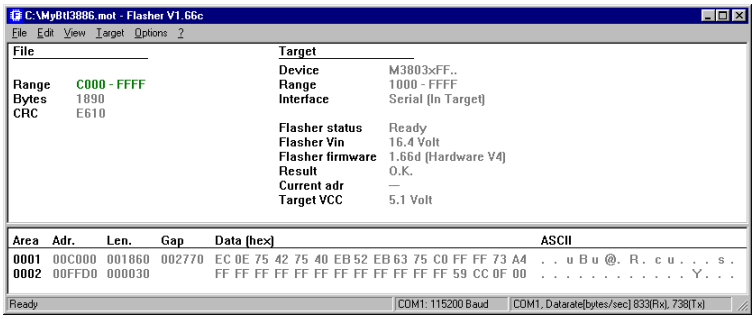
Getting started:

1. Connect the FLASHER to a PC running Windows the 1 by 1 interface cable and run the FLASHER software FLASHER.EXE
2. Connect the FLASHER to the power supply.
3. Set up the device via Options|Device menu of PC program.
4. For in-circuit programming: Connect the FLASHER to target system via the standard 10pin or customized interface cable.

Using the FLASHER PC program

General

Flasher comes with an easy to use Windows program. It allows reading of program files in motorola, intel hex or binary format. The following is a screenshot of FLASHER.EXE with loaded target program

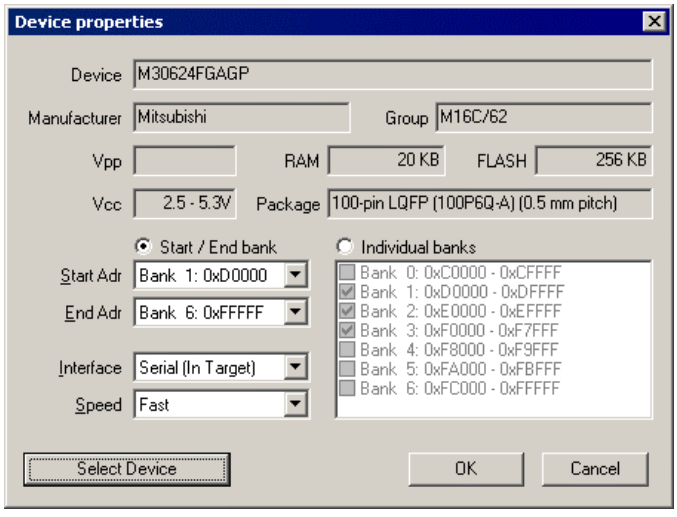


Communication between PC and FLASHER

Make sure the power supply is connected (one of FLASHERs LEDs should be illuminated) and FLASHER is connected to your PC with the 9pin 1 by 1 interface cable as supplied. If the PC-program displays anything other than "No communication" under flasher status, the communication between Flasher and your PC is functioning.

First time setup of FLASHER

When using Flasher for the first time, please select the menu point Options|Device. You will see the following dialog box:



The device properties dialog allows selection of the chip area you would like to access, the sectors of the on chip-flash and the interface you would like to use. The serial interface requires a cable to connect FLASHER to your target.

For targets running at low frequencies, it may be necessary to set the speed option to Slow.

To select an other device, press the „Select Device“ button.

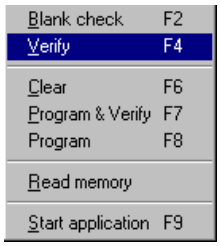
The device selection dialog will open.

You can select the group in a dropdown list and the specific device from the list below

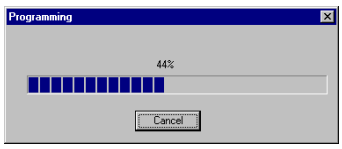
Now you should be able to blank check, clear, program, verify or read the target chip in serial mode (if your target is properly connected to FLASHER). The first time you program or verify, the PC downloads your target program to FLASHER, where it is stored in the on board RAM chip for programming or verification. The PC-Program stores all setup information in the registry; when you start the program the next time, it will start with the same settings.

Programming / clearing / verifying / blank check

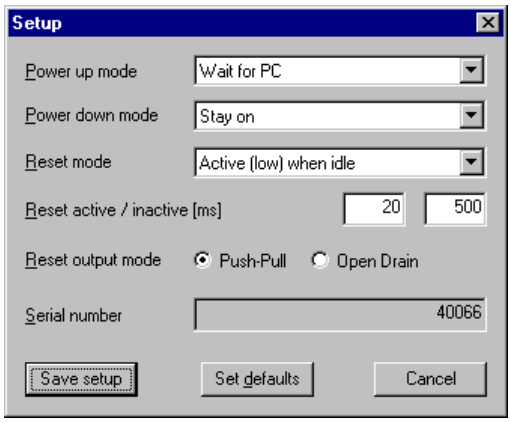
Select one of the commands in the TARGET menu to start the operation. Note that some of the menu points may be grayed if you have no connection to the target or no file loaded.



A modal dialog box will indicate the status and progress of the operation; the operation can be canceled hitting the CANCEL button.



Setup



The operating mode of FLASHER may be changed using the setup dialog from the Options menu.

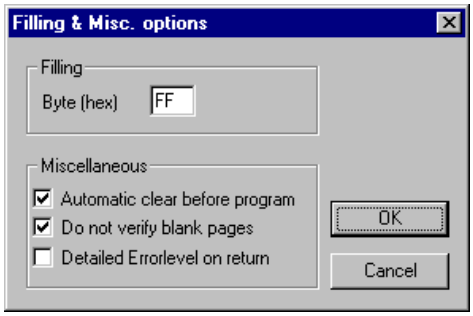
Power up mode, Power down mode and Reset mode should not be changed for normal operation. Setting of Power down mode has no effect on FLASHER 4.

You may change the reset active and reset inactive time, if required by your target hardware.

You may select reset output mode as Push-Pull output or Open Drain.

All setup settings are stored permanently in FLASHER after pressing ‘Save setup’ button.

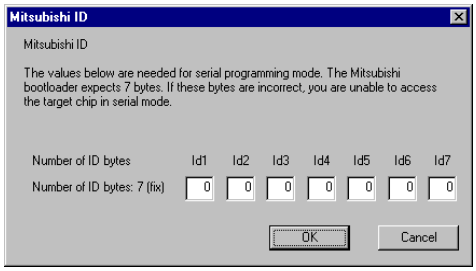
Additional options



The Filling & Misc. Options from the Options menu may be altered if required.
Normally there is no need to change any of these settings
Improper setting of fill byte may lock your target CPU!
When programming blank (virgin) CPUs ‘Automatic clear before program’ is not required, so this feature can be disabled to speed up programming procedure.
Detailed errorlevel on return option may be used to return a detailed errorlevel to the calling program when Flasher is used in batchmode.

ID check

When programming Renesas CPUs in serial mode (in target), an identification of upto 15 bytes has to be supplied. If the target MCUs user program area is blank, this ID-value does not matter. However, after programming, these values need to be set correctly, because otherwise FLASHER will be unable to communicate with the target CPU. These ID-values can be set using the menu point Options|Passcode. With a standard program, these values should be 0, as the high bytes of the interrupt vectors which are used to store the values are usually 0. For more detailed information, please consult the Renesas users manual. The menu point "Edit | Copy passcode into loaded file" can be used to copy these ID bytes into your application program.

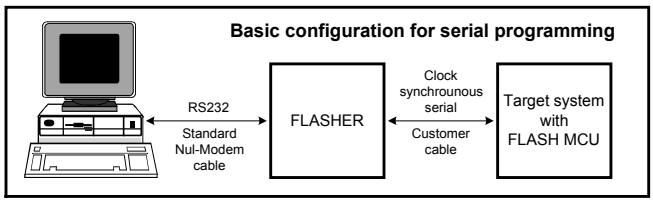


Problems with ID check

You should act carefully when programming ID bytes. If you do not know the ID-value programmed into a target chip, there is no way to erase, read or reprogram the chip in-circuit later. We recommend not to use this feature during the development process.

Using the serial link to program in circuit

FLASHER can be used for in circuit programming of supported CPUs, which incorporate built in firmware for serial update of user flash. The target system has to be designed to support this mode of operation. Refer to target specific connection diagrams or Users manuals of your target CPU.



Operating FLASHER in stand-alone mode

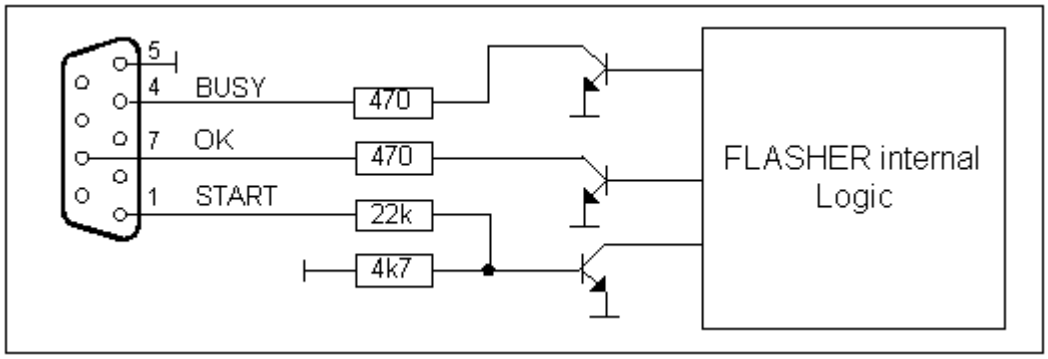
After download the target program and all settings are stored in FLASHERs on board FLASH memory and remain valid until new settings or data are sent to FLASHER.
Any number of microcontrollers may now be programmed by FLASHER (one at a time) without the need of a host PC, by simply pressing the start button. FLASHER will use the settings which have been made in the PC-program. This includes the selection of target address range as well as any options. Whether the target CPU will be erased before programming depends on setting of option “Automatic clear before program”.
Progress and result of the operation is indicated by FLASHERs LEDs:

Status of LED	Meaning
GREEN, flashing	Erasing / Programming / Verifying operation in progress
GREEN	Programming operation successful
RED	Programming operation failed

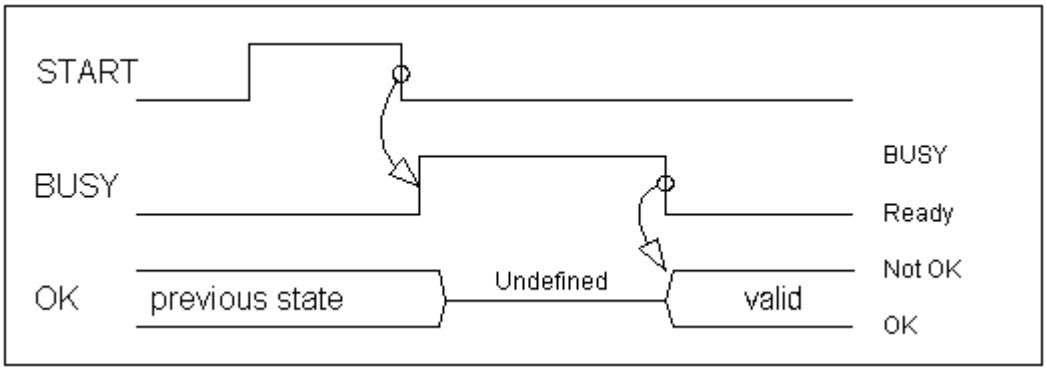
Remote control of FLASHER 5 PRO

FLASHER 5 can be remote controlled by automated testers without the need of a connection to PC and Flashers PC program. Therefore FLASHER 5 is equipped with additional hardware control functions, which are connected to the SUBD9 male connector, normally used as RS232 interface to PC.

The following diagram shows the internal remote control circuitry of FLASHER:



Pin No.	Function	Description
1	START	A positive pulse of any voltage between 5 and 30V with duration of min. 30 ms starts “Auto” function (Clear / Program / Verify) on falling edge of pulse. Whether Clear is executed depends on Options | Filling & misc. | Automatic clear before program.
4	BUSY	As soon as Auto-Function is started, BUSY becomes active, which means that transistor is switched OFF.
7	OK	This output reflects result of last action. It is valid after BUSY turned back to passive state. The output transistor is switched ON to reflect OK state.
5	GND	Common Signal ground



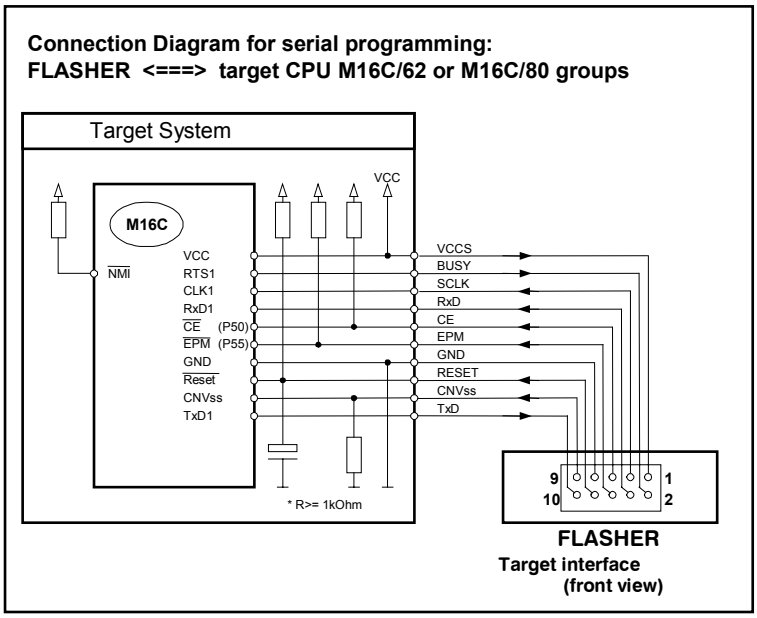
Target interface for M16C/62, M16C26P, M16C/80, M32C and R32C

The clocked synchronous interface from Flasher to the target system is built with a 10 pin dual in line pin connector,(pin1 is marked at the connector) at the front of FLASHER. The function depends on selected target.

Pin no.	Signal	Function for M16C/62, M16C/80	Specification / remarks
1	VCCS	Positive supply voltage of target	Input 3.0 .. 5.5V to supply the interface
2	BUSY	Target CPU Busy signal output	FLASHER Input with Pull-Up to internal 3.3V
3	SCLK	Target CPU Serial clock (input)	FLASHER Output, CMOS driver via 220 Ohms
4	RxD	Target CPU Serial data input	FLASHER Output, CMOS driver via 220 Ohms
5	CE	Chip enable signal of target CPU	FLASHER Input / Output
6	EPM	EPM signal of target CPU	FLASHER Input / Output
7	GND	Common signal ground	---
8	RESET	RESET signal of target system	FLASHER Output, CMOS driver via 220 Ohms
9	CNVss	Target CPU CNVss signal	FLASHER analog Output
10	TxD	Target CPU Serial data output	FLASHER Input / Output

If the RESET of the target system is driven by a reset circuitry with active high driver, the RESET output of FLASHER must not be connected directly to the CPU reset of the target. For M16C/62 or M16C/80 targets you do not have to connect RESET to FLASHER; you can always manually reset your target system after connecting FLASHER.

Target system interface for M16C/62, M16C/80, M32C, R32C

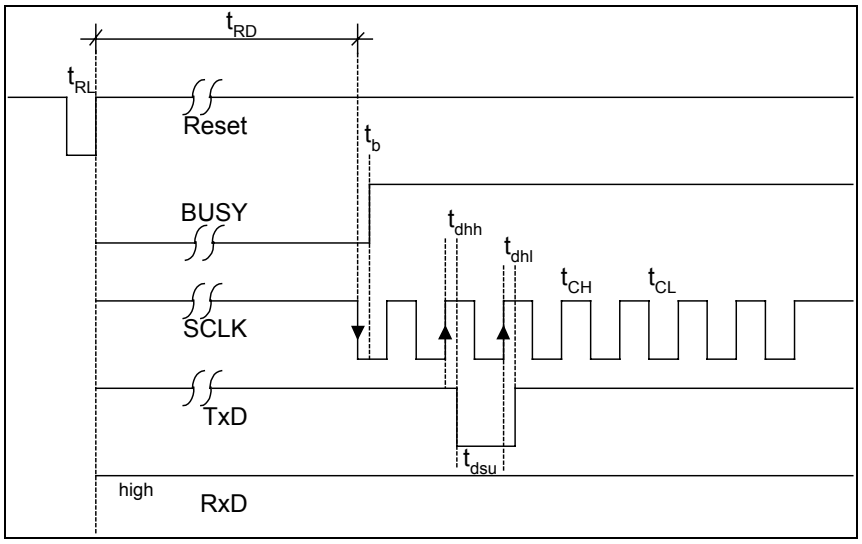


Caution:
Before connecting the target with Flasher, ensure that there is NO difference in the ground potential between Flasher and target.
When the Flasher is connected to a PC via RS232C cable, ensure that the PC and the target operate on the same ground potential.
Connect the ground lines from PC and your target before connecting the Flasher to the target.
If a ground potential difference between Flasher and target exists, the Flasher may be damaged.

Serial programming, technical details for M16C/6x, M16C/80, R32C

Serial programming uses a clock synchronous interface. 8 bits of data (1 byte) is transferred at a time. The commands which are used are described in the Renesas manuals. In general, the sequence is as follows:

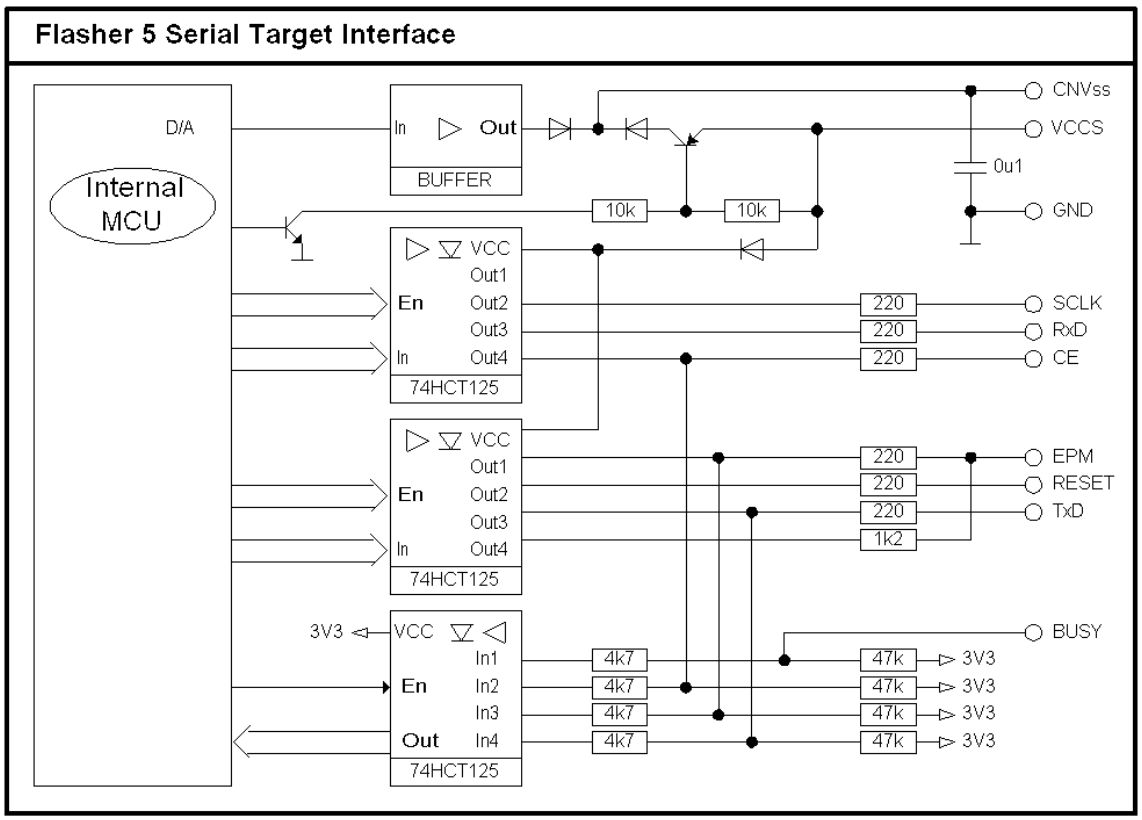
FLASHER resets the target system by pulling the /Reset line low for a period of time which is set as Reset active time in Options|Setup dialog of PC program.
FLASHER waits for the Reset inactive time, nominal 500 ms, (t_{RD}) in order to allow the target system to recover from reset. This time can also be set in Options|Setup dialog via PC program.
FLASHER checks the BUSY line. If it is active (high level) FLASHER stops with error message 40: `Target chip says "BUSY"` because it can not communicate with the target system
FLASHER outputs one clock (clock changes from high to low and back). BUSY should now be active (high). If it is not active, FLASHER stops with error message 41: `Target chip: Busy does not react`
FLASHER outputs 7 more data bits (7 clock cycles) and waits for BUSY to go low.
More data bytes are output (or read) the same way.



t_{RL} : nominal 20 ms
t_{RD} : nominal 500 ms

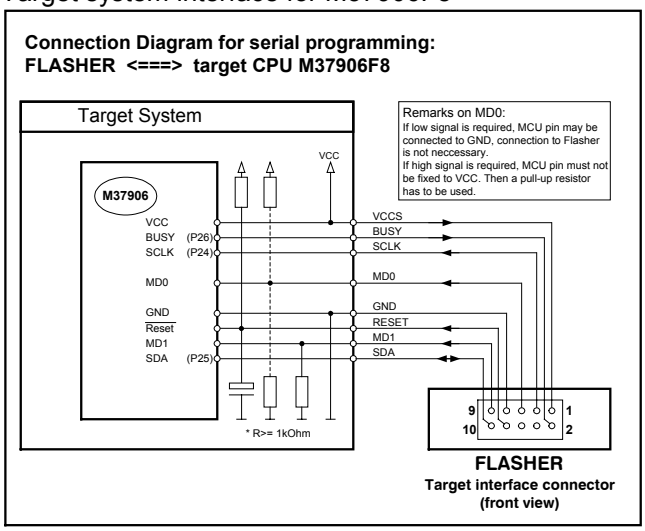
The reset active time (t_{RL}) and reset inactive (eg reset delay) time (t_{RD}) can be set in Options|Setup dialog if required.

Serial target interface circuitry

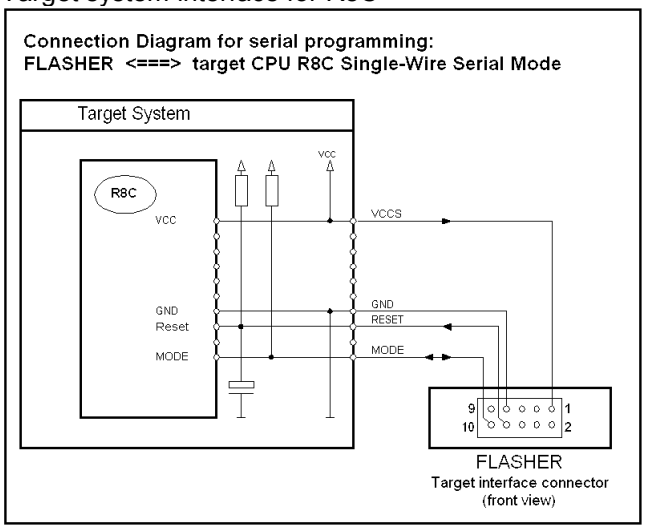


Target interface for other systems

Target system interface for M37906F8



Target system interface for R8C



CRC calculation used in Flasher and PC program.

The Flasher PC program and Flasher calculate a CRC over all data downloaded to the Flasher. The CRC is used to verify correct data transfer as well as an integrity check of target-data stored in the on board non volatile memory.
The CRC is calculated over all bytes of all selected flash sectors of the target device. The calculation is compatible to the algorithm used in the CRC generator circuit inside a Renesas M16C/62P device.
The Renesas application note REU05B0007-0100Z describes how the CRC is calculated.

To calculate the CRC of a target application you have to perform the following steps:

- Create a buffer, large enough to hold all bytes of all selected flash banks of the target.
- Fill the whole buffer with the fill-byte which is defined as fill byte in the Flasher Filling & misc. options.
- Parse your hex-file and fill all defined bytes into the buffer. Bytes not defined in the hex-file were set to the fill byte in step 2.
- Calculate the CRC over the whole buffer following the algorithm from the application note, or the following program example.

Alternatively, the CRC over all bytes may be calculated without a buffer, when a function delivers all bytes of all selected flash banks, including fill bytes, in ascending address order from a stream.

Program example for CRC calculation

```
//
// CRC_Calc1_M16() calculates the CRC for one byte.
// Before calling the function the first time, set the CRC to 0x0000
//
void CRC_Calc1_M16(unsigned int* pCRC, unsigned char Data) {
  unsigned int crc;
  unsigned int x16;
  int i;

  crc = *pCRC;

  for (i = 0; i < 8; i++) {
    if ((crc & 0x0001) ^ (Data & 0x01)) {
      x16 = 0x8408;
    } else {
      x16 = 0x0000;
    }
    crc  = crc >> 1;
    crc ^= x16;
    Data = Data >> 1;
  }

  *pCRC = crc;
}

//
// CRC_Calc() calculates the CRC over NumBytes bytes in a buffer
//
unsigned int CRC_Calc(unsigned char* pBuffer, unsigned long NumBytes) {
  unsigned int   CRC_M16;
  unsigned long  i;

  //
  // Initialize CRC and calculate the CRC over all bytes
  //
  CRC_M16 = 0x0000;
  for (i = 0; i < NumBytes; i++) {
    CRC_Calc1_M16(&CRC_M16, ((char) pBuffer[i]));
  }
  return CRC_M16;
}
```

```
//
// The following example shows how to calculate the CRC over all bytes
// in a buffer as described above using the CRC_Calc() function
//
unsigned long  NumBytes;
unsigned long  i;
unsigned char* pBuffer;
unsigned char  Fillbyte;
unsigned int   CRC;

NumBytes = SUM_OF_ALL_BYTES;  // Number of bytes of all selected flash sectors
//
// Create a buffer for all bytes
//
pBuffer  = (unsigned char*) malloc(NumBytes);
//
// Initialize the buffer, fill up with the fill bytes
//
if (pBuffer != NULL) {
  Fillbyte = FLASHER_FILL_BYTE;  // The fill byte set in Flasher options
  memset(pBuffer, Fillbyte, NumBytes);
  //
  // Fill the buffer with data. This has to be done by a function that
  // parses the Hexfile and addresses the buffer according the address
  // offset which depends on the selected flash sectors
  //
  ParseFile(pBuffer, NumBytes);
  //
  // Initialize CRC and calculate CRC over all bytes in the buffer
  //
  CRC = CRC_Calc(pBuffer, NumBytes);
}
```

Remarks:

The first byte in the buffer has to be the first byte of the first selected flash sector, regardless the address of the sector.

For example, if the first selected sector has address 0x3000, the first byte in the buffer (offset 0) is the byte at address 0x3000 in the target device.

If the hex file addresses only some of the bytes in a flash sector, all the other bytes have to be filled up with the fill byte.

For example, if the selected flash sector has 4096 bytes, starting from address 0x3000 and the hex-file only contains data for the first 16 bytes, the whole area from 0x3010 to 0x3FFF has to be filled up with the fill byte.

All selected sectors are stored in ascending address order without any gap, regardless the start address of the flash sectors.

For example, if the first sector starts at address 0x3000 and has a total size of 0x1000, the second selected sector starts at 0x8000 and has a total size of 0x1000, the buffer to hold the data needs a size of 0x2000 bytes.

The first byte from the first sector is stored at offset 0 in the buffer, the first byte of sector 2 is stored at offset 0x1000.

Error messages

The following error messages can occur during operation of FLASHER (shown in red on your PC) or returned as errorlevel when operated from batch file and “Detailed errorlevel on return” option is set.

Code	Error message	Meaning / remedy
1	Erase failed	Erase operation has failed
2	Write failed	Write operation has failed
3	Verify failed	Verification failed. Loaded Program and contents of the flash-memory are not identical.
4	Blank check failed	Chip is not blank
5	Flash write/erase timed out	Could not write into flash memory, the max. waiting time has been exceeded
6	Can not write into this memory area	In serial mode, the boot area of Renesas CPUs can be read out, but can not be written to.
7	Canceled	Last operation has been canceled by user.
21	Version read failure: Rx line problem ?	In serial mode, FLASHER reads out the version of Renesas target CPUs bootloader. If this is not the right format (VER….), a read line failure is most likely.
30	No ID	Renesas target CPU has no valid ID (This error should not occur)
31	ID mismatch	Renesas bootloader (target CPU) requires the correct ID. Without it, you will be unable to access, read out, clear, or program the contents of the flash memory. The requested ID depends on ID previously written into the target CPU. For more information, please refer to the section “ID code check function” of the CPUs users manual.
32	ID mismatch	Same as Error 31
40	Target chip says "BUSY"	Renesas target chip has BUSY signal high (active). Most likely CPU did not enter bootmode. Check all signals to target. Refer to CPUs users manual about conditions to enter bootmode.
41	Target chip: Busy does not react	Renesas target CPU should set BUSY after receiving the first bit or byte (depending on CPU). Some possible reasons for this error: a) RESET is not released b) Target did not enter the serial I/O mode because conditions to enter bootmode are not met. Check RESET, EPM, CE, CNVSS) c) No clock (check with oscilloscope)
42	Target chip says BUSY	Renesas target chip keep its Busy signal set (active) during communication. This inhibits communication between FLASHER and target. Some possible reasons: a) RESET is not released b) Target did not enter the serial I/O mode because the signals applied when RESET is released are not correct (EPM, CE, CNVSS) c) No clock (check with oscilloscope)
43	Timeout of target	Target command Clear, Program or Verify could not be finished within expected time. Please report this error.
44	Timeout during async. data reception	Target communication does not work. This error may be reported wit R8C targets, or other targets connected via asynchronous interface. Try another baud rate.
55	CRC check in programmer failed.	Integrity of target data held in FLASHERs internal memory is lost. Download new data to FLASHER.
56	Internal Vcc drop during operation	A voltage drop on Flashers internal supply was detected. Internal data may be damaged.
58	DAC for Vpp not calibrated	FLASHER lost calibration data for Vpp generation circuit. Please return FLASHER to factory, as calibration can not be done by user.
60	Unsupported interface, check device setting	FLASHER is requested to access a target CPU that is not supported by Flasher. Normally this fault should not occur. Ensure that Flasher PC software version fits to Flasher firmware version
61	Unsupported command	FLASHER received a command that is not supported. Ensure that Flasher PC software version fits to Flasher firmware version.
62	No target data, please download	FLASHER 5 never received valid data for target, therefore target can not be programmed. Download target data.
63	FLASHER lost Setup	Setup data for Flasher invalid. Please run Option | Setup. and Option | Device.
202	Blank check failed	Target CPU is not blank.
203	Verify failed	Verification failed. Loaded Program and contents of target CPUs flash-memory are not identical.
204	Clear target failed	Erase operation has failed.
205	Target auto func-	Any error occurred during “Auto” function (Clear / Program / Verify)

	tion failed	
206	Programming target failed	An error occurred during programming of target. If target was not blank before, ensure that target is erased before programming. Retry.
207	Reading target failed	An error occurred during reading of target. Retry operation. Check connection to target.
210	Could not open source file	The source file given as parameter could not be opened. Check file name or path.
211	Invalid Parameter.	Invalid command line parameter found.
212	Invalid parameter count.	Check parameter. Refer to command description.
213	Invalid command syntax.	Check parameter. Refer to command description.
214	Firmware mismatch	PC software version differs from firmware version of Flasher. Download new firmware via Options menu. Otherwise proper function can not be guaranteed.
215	The selected target chip is not supported by the connected programmer.	Check and setup device settings.
216	Session mismatch. Resetting connection	It seemed, that Flasher was disconnected during communication between Flasher and PC. Try an other COM port.
217	Flasher refuses connection..	Did you select a chip which is not supported ? Check device settings.
218	None of the selected com ports is available.	PC program could not open selected COM-Port. Select an other COM port.
219	Error in HEX file	The HEX file for target could not be read. Check file format.
220	Invalid file extension, use .MOT, .HEX or .BIN	You tried to open or save a Hex file with unsupported file extension.
221	Invalid file name	You tried to Open or merge a file that could not be found. Check file name or path.
222	Error in Protocol Manager	PC program internal error. Restart program.
223	VCCS > VCCSmax (Target supply voltage too high...)	Supply voltage of target exceeds max value defined for selected device. Check device settings, connection to Flasher and target supply. If everything is correct, Flashers voltage measurement circuitry seems to be damaged and has to be repaired.
224	VCCS < VCCSmin (Target supply voltage too low ...)	Supply voltage of target is lower than min value defined for selected device. Check device settings, connection to Flasher and target supply. If everything is correct, Flashers voltage measurement circuitry seems to be damaged and has to be repaired.
225	Flasher Input voltage too low	Flashers supply voltage is lower than required to generate programming voltage of connected target CPU. Check supply voltage.
226	Assertion	PC program internal error. Restart and try again.
227	Invalid default program settings found. Please setup device.	Your registry contained invalid data about Flasher PC program. Setup device and other options.
228	Flasher not ready !	Setup data could not be sent to Flasher. Is a Flasher connected to your PC? Check COM port.
229	Invalid commandline option	Your command line parameter contains a command which is not supported.
230	Flasher defect !!! Target interface does not work.	Target interface of FLASHER not found during initialization. Flasher has to be repaired.

Trouble shooting

Proper operation of FLASHER in serial mode depends on your target system. If you have any trouble operating FLASHER in serial mode, please:

a) Check your target hardware,
b) Check the connecting cable,
c) Use an oscilloscope to check the state of all signals on the target connector, especially to check if the target CPU is RESET properly and the target CPUs BUSY signal works properly.

Known limitations

Serial mode

Older versions of the Renesas M16C/62 bootloader sometimes do not start after RESET. If you experience problems in communicating with the target system, power down the target system, power it up and try again.

Support

For support questions, please consult our website at www.segger.com. If this does not answer your questions, please send an email to support@segger.com.

Using Flasher in batch mode

Introduction

Flasher.exe supports command line options to enable automated programming of targets. This document describes the supported commands and their respective parameters.
Flasher PC software version 1.72b or above replaces FlasherPro, which is not delivered anymore.

How to start Flasher program in batch mode

To use Flasher in batch mode, just call Flasher.exe directly from DOS-Box or start any batch file which calls Flasher.exe, To start any action, parameter may be passed as command line options.

General rules

- The first parameter specified must be the file to load, if download is required.
- The return code is 0 if all operations have been executed successfully, !=0 otherwise.
- All commands are identical to the corresponding commands in the menu bar.
- All commands are processed from left to right.
- If “-exit” is specified as the last command, the program will terminate as soon as any error occurs or after all commands have been executed.
- If one parameter contains a space use quotation marks for this parameter.

List of commands

The following commands are currently supported as parameter when Flasher is called in batch mode:

Command	Description
-download	Downloads the loaded hex file into Flashers memory without starting any additional action.
-checkblank	Checks if target is blank
-verify	Verifies loaded data with contents of target
-clear	Clears target area, selected memory banks
-clearchip	Clears entire chip. Clearing the entire chip might not be implemented for all devices. Then the command behaves the same as the clear command.
-programverify	Programs & verifies target
-program	Programs loaded data into target
-readback	Reads target area into PC
-start	Starts application program (serial mode only, only Flasher MV3)
-com<PORT>	Sets COM-port of PC(1..4).
-saveas<FILENAME.EXT>[,FIRST-LAST]	Saves the file currently in PC memory. The extension needs to be “MOT”, “HEX” or “BIN” and determines the filetype. The optional range is used for files in BIN-format.
-merge<FILENAME.EXT>	Merge specified file to current data
-delrange<FIRST-LAST>	Deletes the specified range of data
-relocate<OFFSET>	Relocates current data by offset
-selbanks<START,END>	Sets the numbers of start and end bank. The numbers have to be the same as in the selection box shown under Options/Device.
-selmultiple<BANK1, BANK2 … BANKn>	Selects individual banks. The numbers have to be the same as in the selection box shown under Options/Device.
-seldevice<DEVICENAME>	Selects the desired device. The name of the device has to be exactly the same as in the selection box shown under Options/Device.
-password<LENGTH,ID1,ID2,...,IDn>	Sets the ID-bytes
-setFILL<xx>	Sets the fill byte, the parameter is a two digit hex number.
-setIfSync	Selects the serial synchronous interface for the target.
-setIfAsync	Selects the serial asynchronous interface for the target communication. This command is available for Flasher 5 PRO only.

-setIfSpeed<Speed>	Sets the target interface communication speed. Valid speed settings are: 500000Baud 250000Baud 125000Baud 55555Baud 38400Baud 19200Baud 9600Baud Fast Medium Slow VerySlow. The parameter used should be a valid parameter as shown in the device selection dialog for the selected device. The command and parameter have to be passed as one word without any spaces. For example: -setIfSpeed500000Baud Passing wrong settings not allowed for the selected device, may result in unexpected interface speed settings.
-openfile<FILENAME.EXT>	opens a HEX file
-setACBP	Sets option 'Automatic clear before program'.
-setCOD	Sets option 'Clear on demand'.
-setDE	Sets option 'Detailed errorlevel on return'.
-setDNVBP	Sets option 'Do not verify blank pages'.
-clearACBP	Clears option 'Automatic clear before program'.
-clearCOD	Clears option 'Clear on demand'.
-clearDE	Clears option 'Detailed errorlevel on return'.
-clearDNVBP	Clears option 'Do not verify blank pages'.
-index<X>	Allows selection of an individual setup data set for Flasher. This is useful when multiple Flasher should be handled concurrently with one PC. After setting up all options together with the –index command, further calls of Flasher with the same index command will restore all previous settings for that index.
-exit	Finish execution after performing all commands
-help	Displays available commands.
-?	The same as –help

NOTE:

To open a HEX file, place the file name as first parameter just behind the call of Flasher.exe, the openfile command is not required when only one file should be opened.

Return values

The following return values are sent as errorlevel unless “Options | Filling & misc. | Detailed errorlevel on return” is selected:

Value	Meaning
2	Target not blank
3	Verify error, Contents of target data not identical to FLASHERs internal data.
4	Erase error. Target CPU could not be erased.
5	Error during Program & Verify function
6	Eror during programming of target CPU
7	Error during target readback.
8	Error during “Start application”.
9	Timeout error.
10	HEX file could not be opened.
>10	Corresponds to error number which would normally shown on PC screen when program was used in normal mode.

When “Detailed errorlevel on return” is set as option, the returned value equals to error codes described under chapter “Error messages”.

Examples

Program and verify

In the following example the software

- reads the file TEST.MOT
- tells the Flasher to program and verify,
- exits.

```
Flasher.exe test.mot -programverify  -exit
```

Programming using specified com port

In the following example the software

- reads the file TEST.MOT
- sets the COM-port of the PC to Port 2,
- tells the Flasher to program,
- exits.

```
Flasher.exe test.mot -com2 -program -exit
```

Usage in production environment

This example shows a batch file that executes a download to the Flasher once. After the first call of Flasher the batch file runs in a loop that programs the last downloaded file continuously after pressing a key. After every execution of Flasher the return code is evaluated.

NOTE: Evaluating the return code only works under WindowsNT or Windows2000. If you use Windows9x Flasher will be started in a new task and you can't evaluate the return code from the DOS-box from which you started Flasher, because your DOS-Box will not halt execution until FLASHER finished!

```
@echo off
rem The first call of Flasher loads the the HEX file
Flasher test.mot -program -verify -exit
goto checkerror
rem Loop for repeated programming without download
:loop
pause
Flasher -program -verify -exit
rem Check errorlevel if succeed
:checkerror
if errorlevel 1 goto ERROR_%errorlevel%
echo Operation successfully finished
goto loop
rem handle errors
:ERROR_1
echo Undefined error
goto end
:ERROR_2
echo ERROR_TARGET_CHECKBLANK
goto end
:ERROR_3
echo ERROR_TARGET_VERIFY
goto end
:ERROR_4
echo ERROR_TARGET_CLEAR
goto end
:ERROR_5
echo ERROR_TARGET_AUTO
goto end
:ERROR_6
echo ERROR_TARGET_PROGRAM
goto end
:ERROR_7
echo ERROR_TARGET_READBACK
goto end
:ERROR_8
echo ERROR_TARGET_STARTAPPLICATION
goto end
:ERROR_9
echo ERROR_TARGET_TIMEOUT
goto end
:ERROR_10
echo ERROR_OPENDOCUMENTFILE
:end
echo Operation canceled
```

Relocate, delete, merge and save

The following example shows a batch file, where the software

- reads file A.HEX,
- deletes the range 0xFC0000..0xFFFFFF,
- moves the remaining bytes 0x40000 bytes up,
- saves the memory contents as B.MOT,
- reads the file A.HEX again,
- deletes the range 0xC00000..0xCFFFFF,
- moves the remaining bytes 0x340000 bytes down,
- saves the memory contents as C.MOT,
- merges the files B.MOT and C.MOT
- and save the new file as RESULT.BIN.

```
Flasher A.HEX -delrangeFC0000-FFFFFF -relocate040000 -saveasB.MOT -exit
Flasher A.HEX -delrangeC00000-CFFFFF -relocate340000 -saveasC.MOT -exit
Flasher B.MOT -mergeC.MOT -saveasRESULT.BIN,C00000-CFFFFF -exit
```

Selecting device and setting ID bytes

The following example shows a batch file, where the software

- selects a device,
- sets the ID bytes to FF,FF,FF,FF,FF,FF,FF
- and reads the contents of the device.

```
Flasher -seldeviceM30201F6xx -password7,FF,FF,FF,FF,FF,FF,FF -readback
```