

# FF\_PROGFLASH Module Documentation

CSM GmbH, Filderstadt, Germany  
[www.csm.de/unicom/](http://www.csm.de/unicom/)

June 2, 2023

Date	Version	Name	Changes
2013-03-25	1.00	CSM/RN	first release
2016-03-08	1.02	CSM/RN	MultiModule support
2017-01-26	1.03	CSM/RN	New form for Config Module
2017-09-05	1.04	CSM/RN	Config Module command corrected
2023-06-02	1.10	CSM/RN	SWITCH_VERIFY command

All concepts and procedures introduced in this document are intellectual properties of CSM GmbH. Copying or usage by third parties without written permission of CSM GmbH is strictly prohibited. All trademarks mentioned in this document are properties of their respective owners. **This document is subject to changes without notice!**



CSM GmbH Computer-Systeme-Messtechnik  
Raiffeisenstrasse 36 70794 Filderstadt-Bonlanden  
Phone ++49 711 77964 0 Fax ++49 711 77964 40  
mailto:unicom@csm.de http://www.csm.de

Copyright © 2023 by CSM GmbH

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Overview</b>	<b>4</b>
<b>3</b>	<b>Loading and Configuration</b>	<b>5</b>
3.1	MODULE Command . . . . .	5
3.2	CONFIG_INTERFACE Command . . . . .	7
<b>4</b>	<b>FASTFLASH</b>	<b>9</b>
<b>5</b>	<b>FF_PROGFLASH Commands</b>	<b>10</b>
5.1	FF_PROGFLASH::READ_VERSION (2) . . . . .	10
5.2	FF_PROGFLASH::SWITCH_VERIFY (117) . . . . .	12
5.3	FF_PROGFLASH::ErrorCodes . . . . .	13

# Chapter 1

## Introduction

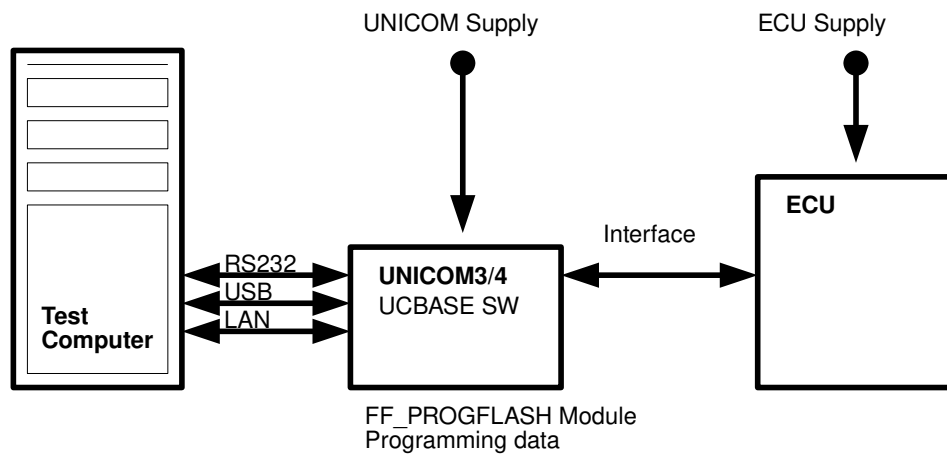
*FF\_PROGFLASH* is a module for extending the *UCBASE* software running on *UNICOM3* and *4*. It implements a special *FASTFLASH* type that uses *PROGRAM\_FLASH(116,122,...)* commands for programming the data.

## Chapter 2

# Overview

To use UNICOM device with FF\_PROGFLASH module, UCBASE software version 1.21 or newer must be installed.

The figure below shows the components of the system.



## Chapter 3

# Loading and Configuration

### 3.1 MODULE Command

This command downloads and runs the FF\_PROGFLASH module.

#### Command, form 1 (unload module)

byte 0	byte 1	byte 2	byte 3
len	ecu	cmd	cks
3	0xC0	20,40..43	

#### Command, form 2 (load module)

byte 0	byte 1	byte 2	byte 3	...	byte N-2	byte N-1	byte N
len	ecu	cmd	mod 1	...	mod m	EOS mod	cks
N=m+4	0xC0	20,40..43				0	

<b>len</b>	length of telegram
<b>ecu</b>	target address
<b>cmd</b>	command code
<b>mod</b>	filename of module (here: ff_progflash.mod)
<b>EOS mod</b>	end-of-string of module filename (0)
<b>cks</b>	checksum of telegram

#### Response

byte 0	byte 1	byte 2	byte 3
len	ecu	status	cks
3	0xC0		

<b>len</b>	length of telegram
------------	--------------------

<b>ecu</b>	source address
<b>status</b>	result status
<b>cks</b>	checksum of telegram

### Remarks

- Before FF\_PROGFLASH module can be used, one of the interface slots must be configured for a matching interface type according to the target communication interface, using the CONFIG\_UNICOM(1) command. Every possible interface of UCBASE software can be used except MODULE(15) interface.
- After MODULE command, FF\_PROGFLASH is configured as follows:
  - use slot 0 for target communication
  - command code of program telegram is set to 116
  - *address* parameter of program telegrams is 4 bytes in size
  - no option byte is used
  - max. portion size per telegram is 256. If such portion size is used, the communication protocol of UNICOM must be set to *XSTP* with the CONFIG\_UNICOM(1) command.
  - type of file that contains programming data is *BINARY*
  - suppressing of sending of data portions that consist completely of 0x00 is activated
- The configuration of FF\_PROGFLASH module can be changed in a wide range using the CONFIG\_INTERFACE(4) command of UCBASE software (s. chapter 3.2 on page 7).

## 3.2 CONFIG\_INTERFACE Command

The *CONFIG\_INTERFACE* command can configure an interface that is activated at the specified slot. If slot is configured for *MODULE* interface and the *FF\_PROGFLASH* module is loaded, the module can configured in a wide range this way.

### Command (form 1)

byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7	
len	ecu	cmd	slot	c-slot	cmd	opt	addrsz	
12	0xC0	4	0..3	0..3			2,3,4	

	byte 8	byte 9	byte 10	byte 11	byte 12
	psize		ftype	stype	cks
	MSB	LSB	0,1,2	0,1,2	

### Command (form 2)

byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7	
len	ecu	cmd	slot	c-slot	cmd	opt	opt2	
13	0xC0	4	0..3	0..3				

	byte 8	byte 9	byte 10	byte 11	byte 12	byte 13
	addrsz	psize		ftype	stype	cks
	2,3,4	MSB	LSB	0,1,2	0,1,2	

<b>len</b>	length of telegram
<b>ecu</b>	target address
<b>cmd</b>	command code
<b>slot</b>	interface slot where <i>MODULE</i> interface is configured
<b>c-slot</b>	interface slot which should be used for communication with target device
<b>cmd</b>	command code of the programming telegrams. Default: 116
<b>opt</b>	option byte that is inserted between command code and address in the programming telegrams. if it is set to 0xFF, no option byte is inserted. That is the default adjustment.
<b>opt2</b>	(optional) another option byte which is placed behind the first one if necessary. Default: not applied.
<b>addrsz</b>	number of bytes that is used for the target address. can be 2,3 or 4 bytes. Default: 4
<b>psize</b>	maximum number of data bytes that can be transferred with one programming telegram. Consider that communication protocol of <i>UNICOM</i> must be switched to <i>XSTP</i> when the entire size of programming telegram is greater then 255 bytes.



With XSTP, the maximum telegram length can be 4095 bytes.  
Default: 256

**f<sub>type</sub>** type of file that contains the programming data:  
0: BINARY (default)  
1: MOTOROLA SRECORD  
2: INTEL HEX

**s<sub>type</sub>** Suppression type:  
0: No suppression  
1: suppression of data portions that consist of 0xFF  
2: suppression of data portions that consist of 0x00 (default)

**cks** checksum of telegram

### Response

byte 0	byte 1	byte 2	byte 3
len	ecu	status	cks
3	0xC0		

**len** length of telegram

**ecu** source address

**status** result status

**cks** checksum of telegram

## Chapter 4

# FASTFLASH

If `FF_PROGFLASH` module is loaded and configured properly, the *FASTFLASH* process can be started by sending an *X\_FASTFLASH(14)* or an *X\_FASTFLASH\_TAB(15)* command to UNICOM.

FASTFLASH can either be used for programming the flash memory, or for verify. Which action is being performed by FASTFLASH can be determined by the `SWITCH_VERIFY` command (ref. chapter 5.2 on page 12). If verify mode is selected, FASTFLASH doesn't transfer data to target device but let target compute a cumulative CRC over the flash ranges which have been programmed before, and computes the same CRC over the ranges of file in UNICOM. As result, both CRCs are reported in the response telegram, first actual CRC which was computed by target, behind nominal CRC, computed by UNICOM. The command reports `VERIFY_ERROR` if these CRCs doesn't fit.

Please refer `ucbase.pdf` for more information about FASTFLASH.

## Chapter 5

# FF\_PROGFLASH Commands

### 5.1 FF\_PROGFLASH::READ\_VERSION (2)

This command reports about the module version information.

#### Command

byte 0	byte 1	byte 2	byte 3
len	ecu	cmd	cks
3	xx	2	

**len** length of telegram  
**ecu** target address  
**cmd** command code  
**cks** checksum of telegram

#### Response

byte 0	byte 1	byte 2	byte 3	...	byte 18	byte 19
len	ecu	status	ver 1	...	ver 16	cks
19	xx					

**len** length of telegram  
**ecu** source address  
**status** result status  
**ver 1..16** version string  
**cks** checksum of telegram

**Remarks**

- As version string `progflash_□□Vx.yy` should be reported.

## 5.2 FF\_PROGFLASH::SWITCH\_VERIFY (117)

This command selects the behaviour of *X\_FASTFLASH(14)* command, either *programming mode* (default setting) or *verify mode*.

### Command

byte 0	byte 1	byte 2	byte 3	byte 4
len	ecu	cmd	mode	cks
4	xx	117	0,1	

<b>len</b>	length of telegram
<b>ecu</b>	target address
<b>cmd</b>	command code
<b>mode</b>	0: programming mode (default) 1: verify mode
<b>cks</b>	checksum of telegram

### Response

byte 0	byte 1	byte 2	byte 3
len	ecu	status	cks
3	xx		

<b>len</b>	length of telegram
<b>ecu</b>	source address
<b>status</b>	result status
<b>cks</b>	checksum of telegram

### Remarks

- In *program mode*, FASTFLASH transfers the programming data portion by portion to the target device and let it be programmed into the flash memory.
- In *verify mode*, FASTFLASH will not transfer any data. It analyzes the file for contiguous data blocks, computes the CRC over them and let the target device do the same on the corresponding flash areas.
- This method is the suggested only if the flash data are contained by an SRECORD or INTEL HEX file where no CRC can be computed easily.
- To let that work, target device must understand the CRC\_FLASH check command (111) with applying the start value can be disabled (opt = 0x80)

### 5.3 FF\_PROGFLASH::ErrorCodes

The following table describes possible error codes reported by the *status* of the response telegrams, and their meanings.

Error	Code	Description
NO_ERROR	0xA0	No error occurred
NOT_CONFIGURED_ERROR	0x90	not matching file type or programming telegram too long with the current protocol
PARAMETER_ERROR	0xB0	Wrong parameter in command telegram
LENGTH_ERROR	0xB3	Wrong command telegram length
FILE_ERROR	0xB9	Toolbox file not found or not readable
FILE_SYNTAX_ERROR	0xBA	Syntax error in SRECORD or INTELHEX file
FILE_CHECKSUM_ERROR	0xBB	Checksum error in SRECORD or INTELHEX file
UNKNOWN_COMMAND_ERROR	0xFF	Command code not supported by the module