

FF_MPC Module Documentation

CSM GmbH, Filderstadt, Germany
www.csm.de/unicom/

September 2, 2019

Date	Version	Name	Changes
2012-10-04	1.0	CSM/RN	first release
2012-10-05	1.01	CSM/RN	some more remarks with the MODULE command
2014-05-27	1.09	CSM/RN	AutoBaud, AutoRetry
2015-07-08	1.14	CSM/RN	adjustable file type and supression mode
2019-09-02	1.30	CSM/RN	logical toolbox parameters

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 © 2019 by CSM GmbH

Contents

1	Introduction	3
2	Overview	4
3	Loading and Configuration	5
3.1	MODULE Command	5
3.2	CONFIG_INTERFACE Command	9
4	FASTFLASH	11
5	FF_MPC Commands	12
5.1	FF_MPC::CONFIG_MODULE (1)	12
5.2	FF_MPC::READ_VERSION (2)	14
5.3	FF_MPC::ErrorCodes	15

Chapter 1

Introduction

FF_MPC is a module for extending the *UCBASE* software running on *UNICOM3*. It realizes downloading and running of toolbox code into RAM of MPC/SPCxxxx based target devices using the CAN download feature of this uC.

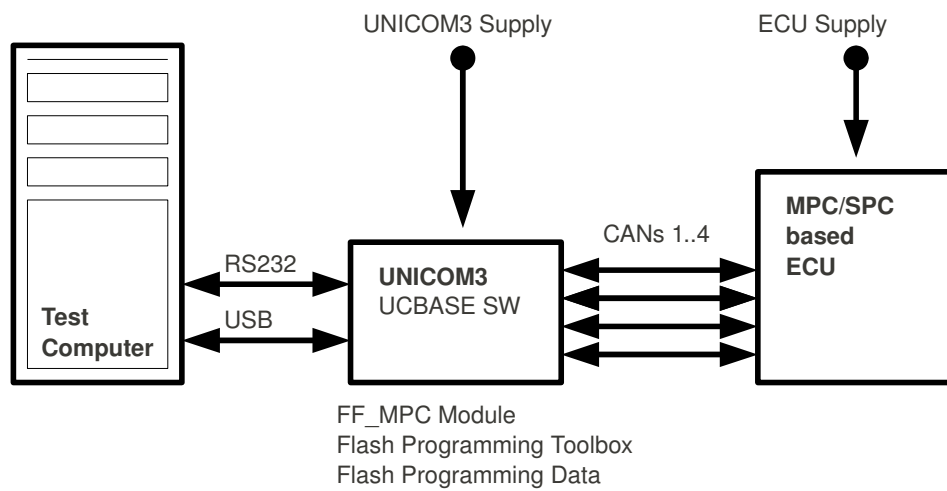
Furthermore, it realizes fast flash programming (*FASTFLASH*) using up to 4 CAN buses in parallel.

Chapter 2

Overview

To use UNICOM3 device with FF_MPC module, UCBASE software version 1.03 or newer must be installed on UNICOM3.

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_MPC module.

Command, form 1 (unload module)

byte 0	byte 1	byte 2	byte 3
len	ecu	cmd	cks
3	0xC0	20	

Command, form 2 (load module, without downloading a toolbox)

byte 0	byte 1	byte 2	byte 3	...	byte N-2	byte N-1	byte N
len	ecu	cmd	mod 1	...	mod n	EOS mod	cks
N=4+n	0xC0	20				0	

Command, form 3 (load module, with downloading a toolbox)

byte 0	byte 1	byte 2	byte 3	...	byte x	byte x	
len	ecu	cmd	mod 1	...	mod n	EOS mod	
N	0xC0	20				0	

	byte x	...	byte x	byte x	byte x	...	byte x
	tbx 1	...	tbx n	EOS tbx	pw1	...	pw 8
				0			

	byte x	byte x	byte x	byte x	byte x	byte x	...	byte N-1	byte N
	TBX Addr				conf	param	...	param	cks
	MSB			LSB					

len	length of telegram
ecu	target address
cmd	command code
mod	filename of module (here: ff_mpc.mod)
EOS mod	end-of-string of module filename (0)
tbx	filename of toolbox
EOD tbx	end-of-string of toolbox filename (0)
pw	password for entering CAN download (8 bytes)
TBX Addr	(optional) destination and start address of toolbox, default: 0x40000000
conf	(optional) configuration flags: bit 2 = 1: enable AutoBaud feature (s. remarks, default: off) bit 3 = 1: enable AutoRetry feature (s. remarks, default: off) bit 7 = 1: VLE encoding mode (default), 0: BOOK_E encoding mode
param	(optional) parameters for toolbox, see below
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

Toolbox Parameters

byte 0	byte 1	
XTAL	PLL	
8..40	0,1	

	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7	byte 8	byte 9	
	Pin1		Pin2		Pin3		Pin4		
	MSB	LSB	MSB	LSB	MSB	LSB	MSB	LSB	

	byte 10	byte 11	...	byte 14	
	CAN		CAN BR		
		MSB	...	LSB	

	byte 15	...	byte 17	byte 18	...	byte 21	
	UC ID			ECU ID			
	MSB	...	LSB	MSB	...	LSB	

	byte 22	byte 32	byte 24	byte 25	
	WD Type		WD Pin		WD SPI
		MSB	LSB		

	byte 26	...	byte 29	byte 30
	CAN FD BR			FF Prot
	MSB	...	LSB	0,1

XTAL	Crystal oscillator frequency in MHz (default: 8)
PLL	0: off, 1: on (default: on)
PinX	Port pins to be set: bit 0..8: number of GPIO, bit 15: level. 0xFFFF means no pin to set (default)
CAN	Code for primary CAN, depends on toolbox, default: 0
CAN BR	CAN bitrate in bits/s, default: 1000000 (0x000F4240)
UC ID	UNICOM CAN ID, default: 0x000007E0
ECU ID	ECU CAN ID, default: 0x000007E8
WD Type	Type of external watchdog to service, depends on toolbox, 0: no watchdog (default)
WD Pin	If a port pin is needed for watchdog service, this defines the GPIO number of this pin, see <i>PinX</i>
WD SPI	If an SPI is used for watchdog service, this defines the num- ber of SPI, default: 0
CAN FD BR	Data bitrate in case of CAN FD able targets in bits/s, default: 0 (0x00000000)
FF Prot	0: legacy FASTFLASH protocol, 1: new FASTFLASH pro-

TOCOL, default value depends on toolbox. Not all toolboxes support new FASTFLASH protocol

Remarks

- To let download work, CAN A/0 of target device must be connected to CAN1 of UNICOM3.
- Mode pins of uC (FAB, ABS0, ABS1, RSTCFGx, ...) must be set to the level that forces running the CAN download program of BAM, have a look at corresponding user manual of uC).
- CAN1 of UNICOM3 must be configured for the download bitrate using the INIT_CAN(98) command of UCBASE software.
- Module command with form 2 only starts the module without using the CAN download interface of target uC. That may useful if a CSM boot loader that starts out of reset runs on target device.
- After successful execution of the MODULE command, the interface slot 0 must be configured for *STPonCAN* (6), the interface slot 1 must be configured for *Module* (15), using the CONFIG_UNICOM(1) command of the UCBASE software. So, a telegram with ECU number = 0x00 will be redirected to toolbox via STPonCAN, and a telegram with ECU number = 0x10 will be executed by the FF_MPC module.
- To let STPonCAN and FASTFLASH work, CAN1 controller must be configured for the STPonCAN CAN parameters of toolbox, and the virtual CAN0 controller must be configured for the FASTFLASH CAN parameters of toolbox, using the CAN_CONFIG(98) command of UCBASE software.
- With *AutoBaud* feature enabled, a special CAN message is sent before the doanload to let uC figure out the current CAN bitrate. Higher bitrates are possible as with the standard CAN download this way. Not all uCs support this feature. Please refer to the user manual of the current uC, chapter *BAM* for more information.
- With *AutoRetry* feature enabled, the module tries up to 5 times to start the CAN download mode in case of fail. That only works if a watchdog chip is included into the target that resets the uC after ap. 300 ms without service (e.g. ATIC155).

3.2 CONFIG_INTERFACE Command

FF_MPC can be configured using the *CONFIG_INTERFACE(4)* command of the *UCBASE* software. To let it work, the slot (0..3) that is specified with the *CONFIG_INTERFACE* telegram must be set to *MODULE* interface (15) using the *CONFIG_UNICOM(1)* command. Please refer *ucbase.pfd* for more information about this command.

The command configures the FF_MPC module. The suppression of data portions that completely consists of 0xFF data bytes while the FASTFLASH process is running can be enabled or disabled, and, the file type of flash data can be adjusted: BINARY, SRECORD or INTEL HEX.

Command

byte 0	byte 1	byte 2	byte 3	byte 4	byte 5
len	ecu	cmd	slot	par	cks
5	0xC0	4	0..3		

len length of telegram

ecu target address

cmd command code

slot interface slot that is configured for MODULE interface

par file and suppression type.

- bit 0+1 define the suppression type:
0: nothing suppressed (default)
1: portions that consist completely of 0xFF bytes will be suppressed from transfer while FASTFLASH.
else: error
- bit 2+3 define the source file type for FASTFLASH:
0: BINARY file (default)
1: SRECORD file
2: INTEL HEX file
else: error

all other bits are ignored.

CAN number of CAN to be used for FASTFLASH (default: 1)

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

Remarks

- 0xFF suppression means that every data portion that should be sent to the target device while FASTFLASH process is running and that consists completely of 0xFF data bytes is suppressed from sending. Since data bits of flash memory of target device are 1 in erased state, such data portions need not to be transferred. That may lead to more or less increasing of programming speed depending on the content of the data file that is to be programmed.
- The 0xFF suppression is not enabled per default after MODULE(20) command.

Chapter 4

FASTFLASH

The FF_MPC module realizes a FASTFLASH procedure that can use up to 4 CAN buses of UNICOM3 for fast data transfer. Data to be programmed into the flash memory of target must reside on UNICOM3's storage medium.

After the module is loaded successfully by the *MODULE(20)* command (s. chapter 3.1 on page 5) and well configured by *CONFIG_MODULE(1)* (s. chapter 5.1 on page 12), the FASTFLASH can be triggered by executing a *FASTFLASH(14)* or a *FASTFLASH_MULTI(15)* command of the UCBASE software. Refer to `ucbase.pdf` for more information about these commands.

Chapter 5

FF_MPC Commands

5.1 FF_MPC::CONFIG_MODULE (1)

With this command, the FASTFLASH procedure of the module can be configured.

Command

byte 0	byte 1	byte 2	byte 3	byte 4	byte 5
len	ecu	cmd	cans	type	cks
N	0x10	1	1..4		

len	length of telegram
ecu	target address
cmd	command code
cans	Number of can buses that are used for FASTFLASH, 1: CAN1, 2: CAN1+CAN2, 3: CAN1+CAN2+CAN3 4: CAN1+CAN2+CAN3+CAN4
type	type of file that contains flash programming data which resides on UNICOM3's storage medium, and type of portion suppression: 0: BINARY, no suppression 1: BINARY, FF suppression enabled 4: SRECORD, no suppression 5: SRECORD, FF suppression enabled 8: INTELHEX, no suppression 9: INTELHEX, FF suppression enabled

cks checksum of telegram

Response

byte 0	byte 1	byte 2	byte 3
len	ecu	status	cks
3	0x10		

len length of telegram
ecu source address
status result status
cks checksum of telegram

Remarks

- This command must be executed before using FASTFLASH.
- If the command is not executed, a default configuration is active
 - 1 CAN used
 - BINARY file type
 - no suppression

5.2 FF_MPC::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	0x10	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	0x10					

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

Remarks

- As version string FF_MPC_Vx.yy should be reported.

5.3 FF_MPC::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	function/command not available
PARAMETER_ERROR	0xB0	Wrong parameter specified in command telegram
LENGTH_ERROR	0xB3	Wrong command telegram length
WRONG_ECHO_ERROR	0xE1	Wrong echo message received while downloading the toolbox
TOOBOX_FILE_ERROR	0xE3	error while reading toolbox from expansion meory
UNKNOWN_COMMAND_ERROR	0xFF	Command code not supported by the module