

# JTAG\_MPC Module Documentation

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

March 7, 2022

Date	Version	Name	Changes
2014-02-28	1.0	CSM/RN	first release
2014-05-07	1.1	CSM/RN	ECU number corrected in remarks of MODULE command
2014-04-30	1.2	CSM/RN	adjustable suppression and file type
2015-07-16	1.3	CSM/RN	MPC57xx support
2015-11-10	1.4	CSM/TO	EEPROM FASTFLASH support
2015-12-18	1.5	CSM/RN	Censor Password
2017-09-12	1.6	CSM/RN	SPC58XX support
2019-12-10	1.7	CSM/RN	Censor Password for SPC58/MPC57
2022-03-07	1.8	CSM/TO	MPC5744 Support, Config2

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 © 2015 by CSM GmbH

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Overview</b>	<b>4</b>
2.1	Requirements . . . . .	4
2.2	Connecting Target . . . . .	5
<b>3</b>	<b>Loading and Configuration</b>	<b>6</b>
3.1	MODULE Command . . . . .	6
3.2	CONFIG_INTERFACE Command . . . . .	11
<b>4</b>	<b>FASTFLASH</b>	<b>13</b>
<b>5</b>	<b>JTAG_MPC Commands</b>	<b>14</b>
5.1	JTAG_MPC::CONFIG_MODULE (1) . . . . .	14
5.2	JTAG_MPC::READ_VERSION (2) . . . . .	16
5.3	JTAG_MPC::WRITE_MEMORY (112) . . . . .	17
5.4	JTAG_MPC::READ_MEMORY (113) . . . . .	18
5.5	JTAG_MPC::ErrorCodes . . . . .	19

# Chapter 1

## Introduction

*JTAG\_MPC* is a module for extending the *UCBASE* software running on *UNI-COM3*. It implements communication with MPC/SPC based target devices thru its JTAG interface such as downloading toolbox code, communication with running toolbox, performing high speed flash programming or simply writing and reading memory.

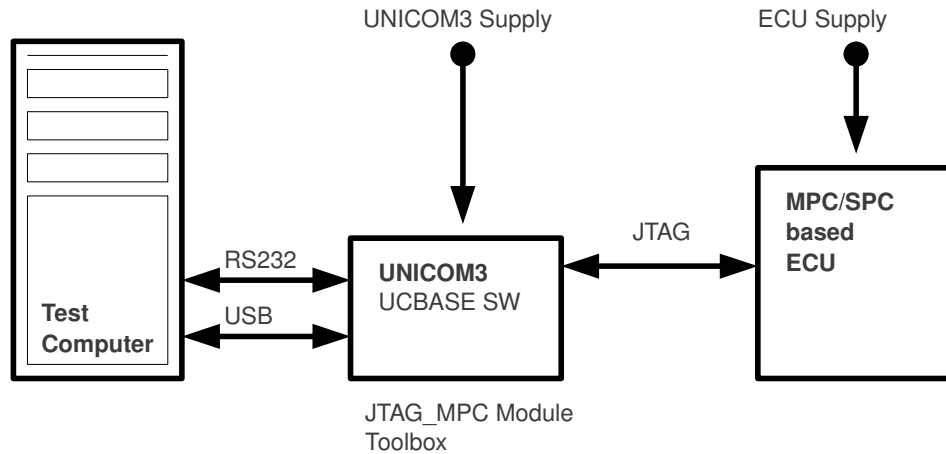
## Chapter 2

# Overview

### 2.1 Requirements

To use UNICOM3 device with JTAG\_MPC module, UCBASE software version 3.03 or newer must be installed on UNICOM3.

The figure below shows the components of the system.

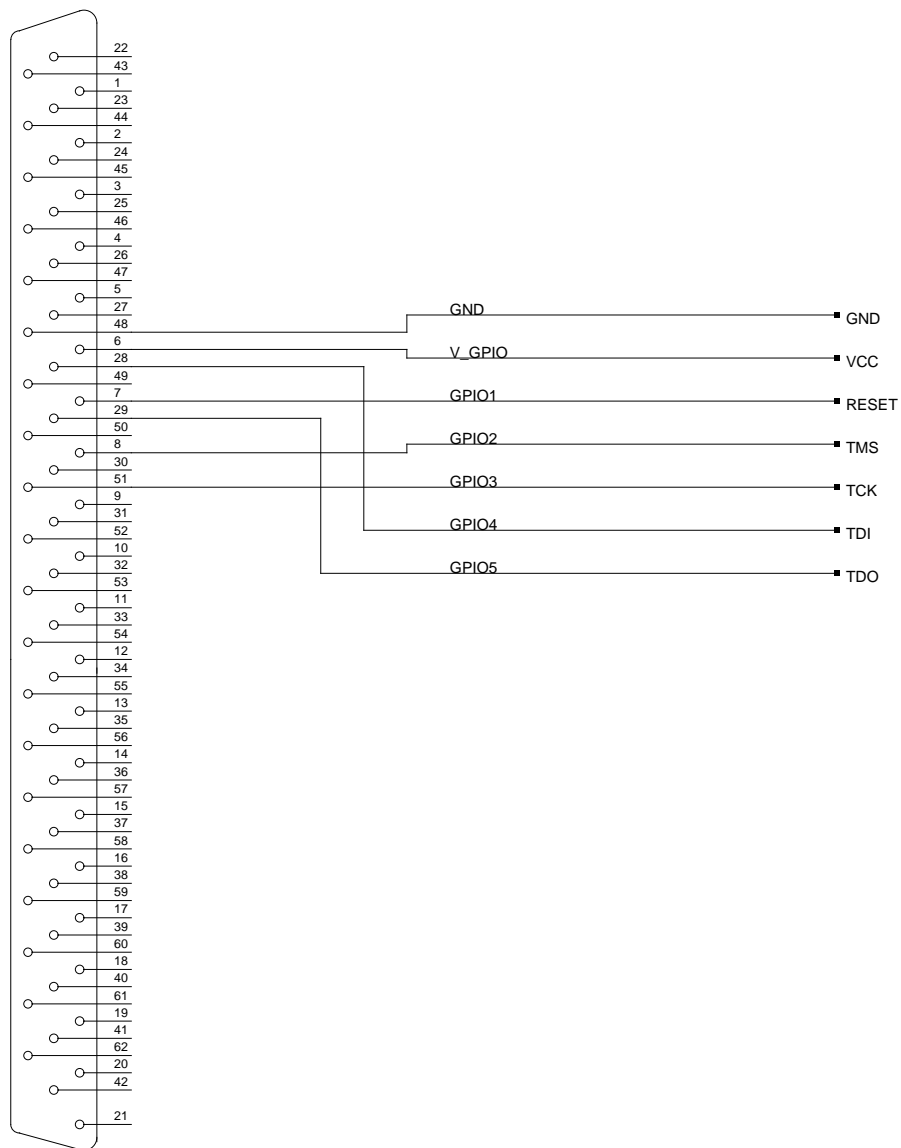


## 2.2 Connecting Target

The figure below shows connecting the target device to UNICOM3 as a matter of principle. Please refer additional information about connection that comes with the project specific software delivery.

UNICOM3  
DSUB62

JTAG interface of MPC/SPC



## Chapter 3

# Loading and Configuration

### 3.1 MODULE Command

This command downloads and runs the JTAG\_MPC 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 without downloading toolbox)

byte 0	byte 1	byte 2	byte 3	...	byte N-5	byte N-4	
len	ecu	cmd	mod 1	...	mod m	EOS mod	
N=m+6	0xC0	20,40..43				0	

byte N-3	byte N-2	byte N-1	byte N
SSC BR	config	config2	cks

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

byte 0	byte 1	byte 2	byte 3	...	byte x	byte x	
len	ecu	cmd	mod 1	...	mod m	EOS mod	
N=m+n+11	0xC0	20,40..43		...		0	

byte x	...	byte x	byte x	
tbx 1	...	tbx n	EOS tbx	
	...		0	

byte N-7	byte N-6	byte N-5	byte N-4	
tbxaddr				
MSB			LSB	

byte N-3	byte N-2	byte N-1	byte N
SSC BR	config	config2	cks

**Command, form 4 (load module with downloading toolbox and with toolbox parameters)**

byte 0	byte 1	byte 2	byte 3	...	byte x	byte x
len	ecu	cmd	mod 1	...	mod m	EOS mod
N	0xC0	20,40..43		...		0

byte x	...	byte x	byte x
tbx 1	...	tbx n	EOS tbx
	...		0

byte x	byte x	byte x	byte x	byte x	byte x
tbxaddr				SSC BR	config
MSB			LSB		

byte x	byte x	...	byte N-1	byte N
config2	tbxpar1	...	tbxpar i	cks
		...		

**Command, form 5 (load module with downloading toolbox censor password and with toolbox parameters)**

byte 0	byte 1	byte 2	byte 3	...	byte x	byte x
len	ecu	cmd	mod 1	...	mod m	EOS mod
N	0xC0	20,40..43		...		0

byte x	...	byte x	byte x
tbx 1	...	tbx n	EOS tbx
	...		0

byte x	byte x	byte x	byte x	byte x	byte x	byte x
tbxaddr				SSC BR	config	config2
MSB			LSB			

byte x	...	byte x	byte x	...	byte N-1	byte N
pw1	...	pw 8/32	tbxpar1	...	tbxpar i	cks
	...			...		



**Example of command without config2 byte (legacy support)**

byte 0	byte 1	byte 2	byte 3	...	byte x	byte x	
len	ecu	cmd	mod 1	...	mod m	EOS mod	
N	0xC0	20,40..43		...		0	

byte x	...	byte x	byte x	
tbx 1	...	tbx n	EOS tbx	
	...		0	

byte x	byte x	byte x	byte x	byte x	byte x	
tbxaddr				SSC BR	config	
MSB			LSB			

byte x	...	byte N-1	byte N
tbxpar1	...	tbxpar i	cks
	...		

<b>len</b>	length of telegram
<b>ecu</b>	target address
<b>cmd</b>	command code
<b>mod</b>	filename of module (here: jtag_mpc.mod)
<b>EOS mod</b>	end-of-string of module filename (0)
<b>tbx</b>	filename of toolbox to be downloaded
<b>EOS tbx</b>	end-of-string of toolbox filename (0)
<b>tbxaddr</b>	destination address of toolbox code
<b>SSC BR</b>	JTAG bitrate in units to 100 kBits/s (max. 6MBits/s), can be adjusted later on (ref. chapter 5.1 on page 14).
<b>config</b>	bit field that controls download:
<b>bit 0</b>	=1: without config2 byte (for downward compatibility) =0: config2 byte is used
<b>bit 1</b>	=1: MMU of target device is being initialized before download
<b>bit 2</b>	=1: SRAM is being initialized before download
<b>bit 3</b>	=1: RESET pin (GPIO1 of UNICOM3) is driven push-pull =0: RESET pin (GPIO1 of UNICOM3) is driven open drain
<b>bit 7</b>	0: without censor password 1: with censor password

<b>config2</b>	second bit field that controls download:
<b>bit 0</b>	has to be 1
<b>bit 4..7</b>	Type of uC: 0: MPC55XX, MPC56XX, SPC56XX, some MPC57XX 1: MPC574X 2: SPC58XX 3: SPC57xx 4: MPC5744
<b>pw1..pw8/32</b>	(optional, if bit 7 of config is 1) censor password. With MPC57xx and SPC58xx, this password must be 32 bytes in size, 8 bytes else.
<b>tbxpar</b>	(optional) up to 30 toolbox specific parameter bytes
<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

- After loading the module, at least one interface slot must be configured for *MODULE* interface using the *CONFIG\_UNICOM(1)* command of UCBASE software.
- The JTAG\_MPC module supports forwarding of STP commands to the toolbox thru the JTAG interface. To distinguish between commands for JTAG\_MPC itself and such for toolbox, the *ECU* number must be set accordingly. If, for instance, slot 0 is configured for *MODULE* interface and ECU number of command from test computer is 0x80, the command will be interpreted by JTAG\_MPC itself. If ECU number is 0x00, the command is forwarded to the toolbox.

Refer to *ucbase.pdf* to learn more about using ECU numbers in command telegrams.

- The *sensor password* must be specified if the device is in *secure mode*, means that the CW field of both NVSCC0 and NVSCC1 flash words are not equal or not 0x55AA. The password can even be specified if the device is not in secure mode, it is ignored in this case.

## 3.2 CONFIG\_INTERFACE Command

JTAG\_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 JTAG\_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: 3)

**cks** 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

- 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

JTAG\_MPC realizes *FASTFLASH* thru JTAG interface with help of running tool-box on target side. Format of programming data can be *BINARY*, *SRECORD* or *INTEL HEX*, refer to CONFIG\_INTERFACE(4) (chapter 3.2 on page 11).

As well as the *X\_FASTFLASH(14)* command and the *X\_FASTFLASH\_MULTI(15)* command is supported by the module.

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

## Chapter 5

# JTAG\_MPC Commands

### 5.1 JTAG\_MPC::CONFIG\_MODULE (1)

With this command, the JTAG bitrate can be modified. This is necessary because toolbox download bitrate is sometimes limited since PLL of MPC/SPC uC is off while this time. After toolbox is up and running, PLL is on and a higher JTAG bitrate is possible.

Additionally, the FASTFLASH command can be switched to program the EEPROM instead of the flash.

#### Command Form 1 (only bitrate)

byte 0	byte 1	byte 2	byte 3	byte 4
len	ecu	cmd	SSC BR	cks
4	xx	1		

#### Command Form 2 (bitrate, and FASTFLASH)

byte 0	byte 1	byte 2	byte 3	byte 4	byte 5
len	ecu	cmd	SSC BR	EEPROM	cks
5	xx	1		0,1	

<b>len</b>	length of telegram
<b>ecu</b>	target address
<b>cmd</b>	command code
<b>SSC BR</b>	new JTAG bitrate in units to 100 kBits/s. Ref. chapter 3.1 on page 6, too.
<b>EEPROM</b>	0: FASTFLASH programs the flash (default setting) 1: FASTFLASH programs the EEPROM
<b>cks</b>	checksum of telegram

### Response

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

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



## 5.2 JTAG\_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	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 `jtag_mpc_vx.yy` should be reported.

### 5.3 JTAG\_MPC::WRITE\_MEMORY (112)

With this command, all writable address ranges (RAM, registers...) can be modified directly thru the JTAG interface. No running toolbox is necessary.

#### Command

byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7	
len	ecu	cmd	opt	start				
N=8+n	xx	112	1,2,4	MSB			LSB	

byte 8	...	byte N-1	byte N
data 1	...	data n	check

<b>len</b>	length of telegram
<b>ecu</b>	target address
<b>cmd</b>	command code
<b>opt</b>	Type of data to be written: 1: byte(8bit), 2: word(16bit), 4: dword(32bit)
<b>start</b>	start destination address
<b>size</b>	size of address range to be read (in bytes)
<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

## 5.4 JTAG\_MPC::READ\_MEMORY (113)

With this command, all address locations can be read directly thru the JTAG interface. No running toolbox is necessary.

### Command

byte 0	byte 1	byte 2	byte 3	
len	ecu	cmd	opt	
10	xx	113	1,2,4	

byte 4	byte 5	byte 6	byte 7	byte 8	byte 9	byte 10
start				size		check
MSB			LSB	MSB	LSB	

<b>len</b>	length of telegram
<b>ecu</b>	target address
<b>cmd</b>	command code
<b>opt</b>	Type of data to be read: 1: byte(8bit), 2: word(16bit), 4: dword(32bit)
<b>start</b>	start source address
<b>data</b>	data to be written
<b>cks</b>	checksum of telegram

### Response

byte 0	byte 1	byte 2	byte 3	...	byte N-1	byte N
len	ecu	status	data 1	...	data n	check
N=3+n	xx					

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

## 5.5 JTAG\_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_INITIALIZED_ERROR	0x90	requested service not available
PARAMETER_ERROR	0xB0	Wrong parameter in command telegram
LENGTH_ERROR	0xB3	Wrong command telegram length
NO_VGPIO_ERROR	0xE0	No voltage spply detected on V_GPIO input of UNICOM3
JTAG_STATUS_ERROR	0xE1	Wrong status read over JTAG (communication problem?)
NEXUS_STATUS_ERROR	0xE2	Status error while reading or writing data thru JTAG interface
TOOLBOX_FILE_ERROR	0xE3	Wrong or corrupted toolbox file
VERIFY_ERROR	0xE4	Error while verifying toolbox code after download
UNKNOWN_COMMAND_ERROR	0xFF	Command code not supported by the module

More error codes are described in `ucbase.pdf`