

SpectraAlyzer MIA AddOn Board

Serial Communication

October 25, 2023

This document explains the serial communication of the SpectraAlyzer MIA AddOn Board. The AddOn Board of the SpectraAlyzer MIA is the interface between the NIRscanNano from Texas Instruments and the individual hardware components (stepper motor, fan, display knob, lamps, etc.). To communicate with this AddOn Board the ZEUTEC serial protocol standard is used. On the AddOn Board there is a ATmega2560 microcontroller which receives and handles the commands.

1 ZEUTEC serial protocol standard

This is the standard protocol for serial communication that shall be used for all devices developed by ZEUTEC and replaces the old simple protocol inherited by Bran+Luebbe. The protocol was developed in July 2016 and implemented for the first time in fan controller board of SpectraAlyzer 2.0.

1.1 Protocol

This is the structure of ZEUTEC's standardised protocol in revision 1:

Name	Length in bytes	Description	Value
Start	1	Start byte	0x16
Receiver	1	Address of receiver	0x01 to 0xFE
Sender	1	Address of sender	0x01 to 0xFE
Package ID	1	Package number (increasing with each package)	0x00 to 0xFF
Command ID	1	Command identifier to be executed	0x00 to 0xFF
Load data length	4	Length of load data in bytes	0x00 to 0xffffffff
Load data	0 to 4294967295	Arbitrary load data	-
End	1	End byte	0x15
Checksum	1	ZEUTEC specific CRC8 check sum	0x00 to 0xFF

For each commands received, the device sends an acknowledgement response and then an answer to the host.

If the command is not clear enough or something goes wrong during the transmission the device just sends a response advising the user that the command was not acknowledged. The not acknowledged response is only send if the device gets enough information to enable the device to send a response (start byte, receiver address, sender address, command number and command ID).

1.2 Checksum

To validate data communicated by means of ZEUTEC serial protocol standard a special checksum of 1 byte length is used.

1.2.1 Algorithm

The checksum S_{crc} for a byte stream of length n is calculated iteratively as follows: First of all S_{crc} is initialised to zero. Next by every byte in the stream starting from index $i = 0$ to index

$i = n - 1$ the checksum is modified. Thereby, the exclusive or combination of the checksum and the current value of S_{crc} is given to the following function as variable `val` (of type `uint8_t`)

```
uint8_t CalculateCRC8ForByte (uint8_t val)
{
    for (int j = 0; j < 8; j++)
    {
        if (val & 0x01)
            val ^= CRC8MASK;
        val >>= 1;
    }

    return val;
}
```

and is returned as new checksum S_{crc} . In the serial ZEUTEC protocol **CRC8MASK = 0x91**.

2 Commands AddOnBoard

The mainboard has a microcontroller and thus a firmware. The serial communication protocol is based on ZEUTEC's serial protocol standard.

- Baudrate: 115,200
- Device adress: 0x20

0x00: Communication alive

Parameters: any
Response: 1 byte
 uint8_t 0x00 when successful
Timeout: 100 ms
Notes:

0x01: Get firmware version

Parameters: any
Response: 3 bytes
 uint8_t 0x00 when successful
 uint8_t for decimal version number
 uint8_t for decimal place version number
Timeout: 100 ms
Notes: E.g. 0x000319 for version 3.25

0x02: Get serial number of microcontroller

Parameters: any
Response: 3 bytes
 uint8_t 0x00 when successful
 uint16_t for serial number of the device
Timeout: 100 ms
Notes: E.g. 0x0003ED for serial number 1005

0x03: Set serial number of microcontroller

Parameters: 2 bytes
 uint16_t for serial number of the device
Response: 1 byte
 uint8_t 0x00 when successful
Timeout: 250 ms
Notes: E.g. 0x0003ED for serial number 1005

0x04: Set device address

Parameters: 1 byte
uint8_t for new device address

Response: 1 byte
uint8_t 0x00 when successful

Timeout: 250 ms

Notes:

0x05: Trigger bootloader for firmware update (not implemented)

Parameters: any

Response: ZEUTEC Bootloader version

Notes: The microcontroller waits for 'pars' for two seconds. If 'pars' are not received, the microcontroller resets and starts again. If 'pars' are received, the microcontroller waits for the hex-file. The hex-file has to be transferred with XON/XOFF control sets. ATTENTION: If used incorrectly, this can destroy the microcontroller software.

0x06:

Parameters: any

Response: 1 byte
uint8_t 0x00 when successful

Timeout: 100 ms

Notes:

0x07: Get stepper parameter

Parameters: any

Response: 6 bytes
uint8_t 0x00 when successful
uint8_t for status (0x00 stepper is not running, 0x01 stepper is running)
uint8_t for direction (0x00 clockwise, 0x01 counterclockwise)
uint8_t for drive selection (0x00 drive by time, 0x01 drive by steps, 0x02 drive by slits)
uint16_t for remaining time, steps or slits or 0x0000 if stepper is not running

Timeout: 250 ms

Notes:

0x08: Set stepper parameter

Parameters: 5 bytes
uint8_t for direction (0x00 clockwise, 0x01 counterclockwise)
uint8_t for drive selection (0x00 drive by time, 0x01 drive by steps, 0x02 drive by slits)
uint16_t for time, steps or slits
uint8_t for speed (0x00 for off to 0x0A for 100%)

Response: 1 byte
uint8_t 0x00 when successful

Timeout: various

Notes:

0x09: Disable stepper

Parameters: any
Response: 1 byte
uint8_t 0x00 when successful
Timeout: 250 ms
Notes:

0x0A: Get stepper optical switch

Parameters: any
Response: 3 bytes
uint8_t 0x00 when successful
uint8_t 0x00 optical switch 0 off, 0x01 optical switch 0 on
uint8_t 0x00 optical switch 1 off, 0x01 optical switch 1 on
Timeout: 250 ms
Notes:

0x0B: Move stepper

Parameters: 3 bytes
uint16_t for steps
uint8_t for reference position (0x00 no homing before moving steps,
0x01 first homing then steps)
Response: 1 byte
uint8_t 0x00 when successful
Timeout: various
Notes: E.g. 0x0107D0 First rotate to the reference position the rotate 2,000 steps.

0x0C: Stepper initialisation

Parameters: any
Response: 1 byte
uint8_t 0x00 when successful
Timeout: 10000 ms
Notes: Move to the home position.

0x0D: Get ADC TEC value

Parameters: any
Response: 3 bytes
uint8_t 0x00 when successful
uint16_t for ADC value (10-bit)
Timeout: 100 ms
Notes:

0x0E: Set TEC

Parameters: 2 bytes
uint8_t for enable (0x00 off, 0x01 on)
int8_t for temperature
Response: 1 byte
uint8_t 0x00 when successful
Timeout: 250 ms
Notes: E.g. 0x01F6 for TEC on with a temperature of -10°C

0x0F: Get TEC status

Parameters: any
Response: 4 bytes
uint8_t 0x00 when successful
uint8_t 0x00 status of TEC
Bit 0: 0 for on, 1 for off
Bit 1: for IFLAG
Bit 2: for TFLAG
int8_t for actual temperature
int8_t for set temperature
Timeout: 250 ms
Notes:

0x10: Set lamp

Parameters: 4 bytes
uint8_t for enable lamp 1 (0x00 off, 0x01 on)
uint8_t for enable lamp 2 (0x00 off, 0x01 on)
uint8_t for enable lamp 3 (0x00 off, 0x01 on)
uint8_t for enable lamp 4 (0x00 off, 0x01 on)
Response: 1 byte
uint8_t 0x00 when successful
Timeout: 250 ms
Notes: E.g. 0x01000100 for switching lamp 1 and lamp 3 on

0x11: Get lamp status

Parameters: any
Response: 5 bytes
uint8_t 0x00 when successful
uint8_t for lamp 1 (0x00 off, 0x01 on)
uint8_t for lamp 2 (0x00 off, 0x01 on)
uint8_t for lamp 3 (0x00 off, 0x01 on)
uint8_t for lamp 4 (0x00 off, 0x01 on)
Timeout: 250 ms
Notes:

3 Commands Display Knob

0x40: Communication alive (Display knob)

Parameters: any
Response: 1 byte
 uint8_t 0x00 when successful
Timeout: 250 ms
Notes:

0x41: Get stepper firmware version (Display knob)

Parameters: any
Response: 3 bytes
 uint8_t 0x00 when successful
 uint8_t for main version
 uint8_t for minor version
Timeout: 250 ms
Notes: E.g. 0x000319 for version 3.25

0x42: Get serial number

Parameters: any
Response: 3 bytes
 uint8_t 0x00 when successful
 uint16_t for serial number of the device
Timeout: 250 ms
Notes:

0x43: Set serial number

Parameters: 2 bytes
 uint16_t for serial number of the device
Response: 1 byte
 uint8_t 0x00 when successful
Timeout: 250 ms
Notes:

0x44: Set device address

Parameters: 1 byte
 uint8_t for new device address
Response: 1 byte
 uint8_t 0x00 when successful
Timeout: 250 ms
Notes:

0x45: Trigger stepper bootloader for firmware update (not implemented)

Parameters: any
Response: ZEUTEC Bootloader version
Notes: The microcontroller waits for 'pars' for two seconds. If 'pars' are not received, the microcontroller resets and starts again. If 'pars' are received, the microcontroller waits for the hex-file. The hex-file has to be transferred with XON/XOFF control sets. ATTENTION: If used incorrectly, this can destroy the microcontroller software.

0x46:

Parameters: any
Response: 1 byte
uint8_t 0x00 when successful
Timeout: 250 ms
Notes:

0x47: Set bluetooth address

Parameters: 6 bytes
uint48_t for bluetooth address
Response: 1 byte
uint8_t 0x00 when successful
Timeout: 250 ms
Notes: E.g. 0x123456789ABC for bluetooth address 12:34:56:78:9A:BC

0x48: Set bluetooth connection

Parameters: 1 byte
uint8_t 0x00 for not connected, 0x01 for connected
Response: 1 byte
uint8_t 0x00 when successful
Timeout: 250 ms
Notes:

0x49: Get display knob status

Parameters: any
Response: 1 byte
uint8_t 0x00 when successful
Timeout: 250 ms
Notes:

0x4A: Set language

Parameters: 1 byte
uint8_t for language (0x00 for Chinese, 0x01 for English, 0x02 for German)
Response: 1 byte
uint8_t 0x00 when successful
Timeout: 250 ms
Notes:

0x4B: Set measurement status

Parameters: 2 bytes
uint8_t for status (0x00 for idle, 0x01 for measurement in progress)
uint8_t for progress in percent

Response: 1 byte
uint8_t 0x00 when successful

Timeout: 250 ms

Notes:

0x4C: Set measurement results

Parameters: variable
uint8_t for each ASCII character for parameter name
uint8_t 0x2C for separate name and value
uint8_t for each ASCII character for parameter value with two decimal places
uint8_t 0x3B if more parameters should be displayed

Response: 1 byte
uint8_t 0x00 when successful

Timeout: 250 ms

Notes: E.g. 0x4F696C2C323334353B 0x4D6F6973747572652C303730313B
0x4646412C30303834 for "Oil,23.45%;Moisture,07.01%;FFA,00.84%"

4 Error list

CODE	ERROR	NOTE
0x00	SUCCESSFUL	Command was sucessful
0x02	COMMAND_NOT_OK	Command length not ok
0x03	COMMAND_NOT_OK	Checksum not ok
0x04	COMMAND_NOT_OK	No stop bit received
0x3A	UNKNOWN_COMMAND	Command is not in the command list
0x3B	WRONG_DATA	
0x3C	VALUE_LESS_THAN_MIN	Value is below the minimum value
0x3D	VALUE_MORE_THAN_MAX	Value is above the maximum value
0x3E	WRONG_DATA_LENGTH	Length of the values is not correct
0x3F	UNDEFINED_ERROR	
0x51	STEPPER_UNDEFINED_HOME	Stepper has no position
0x52	STEPPER_NO_ROTATION	Stepper is not rotating
0x60	DISPLAY_KNOB_NO_COMM	No connection to display knob

5 Revision

Revision	Date	Author	Comments
1	09/23	Michael Grelcke	Initial version
1.1	09/23	Michael Grelcke	Fixed error in the protocol
1.2	09/23	Michael Grelcke	Added display knob commands
1.3	10/23	Michael Grelcke	Added lamp commands

The information contained in or attached to this document is non-public and the confidential, proprietary information of ZEUTEC Opto-Elektronik GmbH. It may not be reproduced or disclosed to any third party or used by the recipient and/or the recipient's employer (hereinafter "recipient") for other than the intended purposes for which it was provided to the recipient without the prior written approval of ZEUTEC Opto-Elektronik GmbH. These restrictions are in addition to any restrictions that may apply pursuant to the terms of any Confidentiality or Non -Disclosure Agreement(s) between ZEUTEC Opto-Elektronik GmbH, its affiliates, or the recipient.

Copyright ©2023, ZEUTEC Opto-Elektronik GmbH
Friedrich-Voss-Strasse 11, 24768 Rendsburg, Germany
(+49) 4331 –136650
www.spectraalyzer.com, www.zeutec.de