

MANUAL

ANAGATE

TCP/IP

COMMUNICATION

ANALYTICA GmbH

Vorholzstrasse 36
D-76137 Karlsruhe
[gmbh.de](http://www.analytica-gmbh.de)

Tel. +49 721 35043-0
Fax: +49 721 35043-20

E-mail: info@analytica-gmbh.de
WWW: <http://www.analytica-gmbh.de>

Revision History

| Version | Date | Changes |
|---------|------------|---|
| 1.0 | 14.06.2004 | Initial version |
| 1.1 | 04.08.2004 | Integration of AnaGate I2C in the document |
| 1.2 | 15.08.2005 | Integration of AnaGate CAN and AnaGate DigitalIO in the document, revision of AnaGate I2C |
| 1.2.1 | 04.05.2006 | Integration of AnaGate SPI. |
| 1.2.2 | 08.11.2007 | Integration of AnaGate A/D |
| 1.2.3 | 04.05.2006 | Some protocol changes for AnaGate CAN. |
| 1.2.4 | 13.05.2008 | Some protocol extensions for AnaGate CAN uno/duo/Quattro. |
| 1.2.5 | 20.04.2009 | New timestamp command for AnaGate CAN. |
| 1.2.6 | 15.05.2009 | New restart command for AnaGate CAN. |
| 1.2.6.1 | 05.08.2015 | Invalid codes for GetGlobals/SetGobals (I2C) fixed. |
| | | |

Contents

| | | |
|--------|--------------------------------------|----|
| 1 | Introduction | 9 |
| 1.1 | Information about this manual..... | 9 |
| 1.2 | Restrictions..... | 9 |
| 2 | General Interface..... | 10 |
| 2.1 | Setting up a connection..... | 10 |
| 2.2 | Data transmission | 11 |
| 2.2.1 | Telegram structure..... | 11 |
| 2.3 | Closing a connection | 14 |
| 3 | Specific Telegrams..... | 15 |
| 3.1 | All AnaGate devices..... | 15 |
| 3.1.1 | OP_ANAGATE_XX_OPEN_REQ..... | 15 |
| 3.1.2 | OP_ANAGATE_XX_OPEN_CNF..... | 15 |
| 3.1.3 | OP_ANAGATE_XX_CLOSE_REQ..... | 16 |
| 3.1.4 | OP_ANAGATE_XX_CLOSE_CNF..... | 16 |
| 3.1.5 | OP_ANAGATE_XX_GET_INFO_REQ..... | 16 |
| 3.1.6 | OP_ANAGATE_XX_GET_INFO_CNF..... | 17 |
| 3.1.7 | OP_ANAGATE_XX_DIO_WRITE_REQ..... | 18 |
| 3.1.8 | OP_ANAGATE_XX_DIO_WRITE_CNF..... | 18 |
| 3.1.9 | OP_ANAGATE_XX_DIO_READ_REQ..... | 19 |
| 3.1.10 | OP_ANAGATE_XX_DIO_READ_CNF..... | 19 |
| 3.2 | AnaGate I2C..... | 21 |
| 3.2.1 | OP_ANAGATE_I2C_OPEN_REQ..... | 22 |
| 3.2.2 | OP_ANAGATE_I2C_OPEN_CNF..... | 22 |
| 3.2.3 | OP_ANAGATE_I2C_CLOSE_REQ..... | 22 |
| 3.2.4 | OP_ANAGATE_I2C_CLOSE_CNF..... | 23 |
| 3.2.5 | OP_ANAGATE_I2C_RESET_REQ..... | 23 |
| 3.2.6 | OP_ANAGATE_I2C_RESET_CNF..... | 23 |
| 3.2.7 | OP_ANAGATE_I2C_SET_GLOBALS_REQ..... | 23 |
| 3.2.8 | OP_ANAGATE_I2C_SET_GLOBALS_CNF..... | 24 |
| 3.2.9 | OP_ANAGATE_I2C_GET_GLOBALS_REQ..... | 24 |
| 3.2.10 | OP_ANAGATE_I2C_GET_GLOBALS_CNF..... | 25 |
| 3.2.11 | OP_ANAGATE_I2C_WRITE_REQ..... | 25 |
| 3.2.12 | OP_ANAGATE_I2C_WRITE_CNF..... | 26 |
| 3.2.13 | OP_ANAGATE_I2C_READ_REQ..... | 27 |
| 3.2.14 | OP_ANAGATE_I2C_READ_CNF..... | 28 |
| 3.2.15 | OP_ANAGATE_I2C_EEPROM_WRITE_REQ..... | 29 |
| 3.2.16 | OP_ANAGATE_I2C_EEPROM_WRITE_CNF..... | 31 |
| 3.2.17 | OP_ANAGATE_I2C_EEPROM_READ_REQ..... | 31 |
| 3.2.18 | OP_ANAGATE_I2C_EEPROM_READ_CNF..... | 33 |
| 3.2.19 | OP_ANAGATE_I2C_SEQUENCE_REQ..... | 34 |
| 3.2.20 | OP_ANAGATE_I2C_SEQUENCE_CNF..... | 35 |
| 3.2.21 | OP_ANAGATE_I2C_GET_INFO_REQ..... | 36 |
| 3.2.22 | OP_ANAGATE_I2C_GET_INFO_CNF..... | 36 |
| 3.2.23 | OP_ANAGATE_I2C_STATUS_REQ..... | 36 |
| 3.2.24 | OP_ANAGATE_I2C_STATUS_CNF..... | 37 |
| 3.2.25 | OP_ANAGATE_I2C_DIO_WRITE_REQ..... | 37 |
| 3.2.26 | OP_ANAGATE_I2C_DIO_WRITE_CNF..... | 37 |
| 3.2.27 | OP_ANAGATE_I2C_DIO_READ_REQ..... | 37 |
| 3.2.28 | OP_ANAGATE_I2C_DIO_READ_CNF..... | 37 |

| | | |
|--------|---|----|
| 3.3 | AnaGate CAN / AnaGate CAN uno, duo, quattro | 38 |
| 3.3.1 | OP_ANAGATE_CAN_OPEN_REQ | 40 |
| 3.3.2 | OP_ANAGATE_CAN_OPEN_CNF | 40 |
| 3.3.3 | OP_ANAGATE_CAN_CLOSE_REQ | 40 |
| 3.3.4 | OP_ANAGATE_CAN_CLOSE_CNF | 40 |
| 3.3.5 | OP_ANAGATE_CAN_DATA_REQ | 40 |
| 3.3.6 | OP_ANAGATE_CAN_DATA_CNF | 42 |
| 3.3.7 | OP_ANAGATE_CAN_DATA_IND | 42 |
| 3.3.8 | OP_ANAGATE_CAN_DATA_RSP | 44 |
| 3.3.9 | OP_ANAGATE_CAN_GET_INFO_REQ | 44 |
| 3.3.10 | OP_ANAGATE_CAN_GET_INFO_CNF | 45 |
| 3.3.11 | OP_ANAGATE_CAN_SET_CONFIG_REQ | 45 |
| 3.3.12 | OP_ANAGATE_CAN_SET_CONFIG_CNF | 45 |
| 3.3.13 | OP_ANAGATE_CAN_GET_CONFIG_REQ | 46 |
| 3.3.14 | OP_ANAGATE_CAN_GET_CONFIG_CNF | 46 |
| 3.3.15 | OP_ANAGATE_CAN_SET_GLOBALS_REQ | 47 |
| 3.3.16 | OP_ANAGATE_CAN_SET_GLOBALS_CNF | 49 |
| 3.3.17 | OP_ANAGATE_CAN_GET_GLOBALS_REQ | 50 |
| 3.3.18 | OP_ANAGATE_CAN_GET_GLOBALS_CNF | 50 |
| 3.3.19 | OP_ANAGATE_CAN_SET_FILTER_REQ | 51 |
| 3.3.20 | OP_ANAGATE_CAN_SET_FILTER_CNF | 53 |
| 3.3.21 | OP_ANAGATE_CAN_GET_FILTER_REQ | 53 |
| 3.3.22 | OP_ANAGATE_CAN_GET_FILTER_CNF | 53 |
| 3.3.23 | OP_ANAGATE_CAN_SET_TIME_REQ | 54 |
| 3.3.24 | OP_ANAGATE_CAN_SET_TIME_CNF | 55 |
| 3.3.25 | OP_ANAGATE_CAN_RESTART_REQ | 55 |
| 3.3.26 | OP_ANAGATE_CAN_RESTART_CNF | 56 |
| 3.3.27 | OP_ANAGATE_CAN_DIO_WRITE_REQ | 56 |
| 3.3.28 | OP_ANAGATE_CAN_DIO_WRITE_CNF | 56 |
| 3.3.29 | OP_ANAGATE_CAN_DIO_READ_REQ | 57 |
| 3.3.30 | OP_ANAGATE_CAN_DIO_READ_CNF | 57 |
| 3.3.31 | „Z-CARD Process Image“ mode | 57 |
| 3.4 | AnaGate SPI | 62 |
| 3.4.1 | OP_ANAGATE_SPI_OPEN_REQ | 63 |
| 3.4.2 | OP_ANAGATE_SPI_OPEN_CNF | 63 |
| 3.4.3 | OP_ANAGATE_SPI_CLOSE_REQ | 63 |
| 3.4.4 | OP_ANAGATE_SPI_CLOSE_CNF | 63 |
| 3.4.5 | OP_ANAGATE_SPI_GET_INFO_REQ | 63 |
| 3.4.6 | OP_ANAGATE_SPI_GET_INFO_CNF | 64 |
| 3.4.7 | OP_ANAGATE_SPI_DIO_WRITE_REQ | 64 |
| 3.4.8 | OP_ANAGATE_SPI_DIO_WRITE_CNF | 64 |
| 3.4.9 | OP_ANAGATE_SPI_DIO_READ_REQ | 64 |
| 3.4.10 | OP_ANAGATE_SPI_DIO_READ_CNF | 64 |
| 3.4.11 | OP_ANAGATE_SPI_DATA_REQ | 64 |
| 3.4.12 | OP_ANAGATE_SPI_DATA_CNF | 65 |
| 3.4.13 | OP_ANAGATE_SPI_SEQUENCE_REQ | 65 |
| 3.4.14 | OP_ANAGATE_SPI_SEQUENCE_CNF | 66 |
| 3.4.15 | OP_ANAGATE_SPI_SET_GLOBALS_REQ | 67 |
| 3.4.16 | OP_ANAGATE_SPI_SET_GLOBALS_CNF | 68 |
| 3.4.17 | OP_ANAGATE_SPI_GET_GLOBALS_REQ | 69 |
| 3.4.18 | OP_ANAGATE_SPI_GET_GLOBALS_CNF | 69 |
| 3.5 | AnaGate RS232 | 71 |
| 3.6 | AnaGate DigitalIO | 72 |

| | | |
|--------|----------------------------------|----|
| 3.6.1 | OP_ANAGATE_DIO_OPEN_REQ..... | 72 |
| 3.6.2 | OP_ANAGATE_DIO_OPEN_CNF..... | 73 |
| 3.6.3 | OP_ANAGATE_DIO_CLOSE_REQ..... | 73 |
| 3.6.4 | OP_ANAGATE_DIO_CLOSE_CNF..... | 73 |
| 3.6.5 | OP_ANAGATE_DIO_GET_INFO_REQ..... | 73 |
| 3.6.6 | OP_ANAGATE_DIO_GET_INFO_CNF..... | 73 |
| 3.6.7 | OP_ANAGATE_DIO_WRITE_REQ..... | 73 |
| 3.6.8 | OP_ANAGATE_DIO_WRITE_CNF..... | 73 |
| 3.6.9 | OP_ANAGATE_DIO_READ_REQ..... | 74 |
| 3.6.10 | OP_ANAGATE_DIO_READ_CNF..... | 74 |
| 3.7 | AnaGate Audio..... | 75 |
| 3.8 | AnaGate A/D..... | 76 |
| 3.8.1 | OP_ANAGATE_AD_OPEN_REQ..... | 77 |
| 3.8.2 | OP_ANAGATE_AD_OPEN_CNF..... | 77 |
| 3.8.3 | OP_ANAGATE_AD_CLOSE_REQ..... | 77 |
| 3.8.4 | OP_ANAGATE_AD_CLOSE_CNF..... | 77 |
| 3.8.5 | OP_ANAGATE_AD_GET_INFO_REQ..... | 77 |
| 3.8.6 | OP_ANAGATE_AD_GET_INFO_CNF..... | 77 |
| 3.8.7 | OP_ANAGATE_AD_DIO_WRITE_REQ..... | 77 |
| 3.8.8 | OP_ANAGATE_AD_DIO_WRITE_CNF..... | 78 |
| 3.8.9 | OP_ANAGATE_AD_DIO_READ_REQ..... | 78 |
| 3.8.10 | OP_ANAGATE_AD_DIO_READ_CNF..... | 78 |
| 3.8.11 | OP_ANAGATE_AD_READ_REQ..... | 78 |
| 3.8.12 | OP_ANAGATE_AD_READ_CNF..... | 78 |
| 3.9 | AnaGate Phone..... | 80 |

Charts

| | | |
|-------------|--|----|
| Chart 2-1: | Overview of the various AnaGate ports | 10 |
| Chart 2-2: | General structure of a telegram..... | 11 |
| Chart 2-3: | Length..... | 11 |
| Chart 2-4: | Command code | 13 |
| Chart 2-5: | Command ID | 13 |
| Chart 2-6: | Examples of telegrams..... | 14 |
| Chart 3-1: | Telegram data for OP_ANAGATE_XX_OPEN_CNF..... | 15 |
| Chart 3-2: | Telegram data for OP_ANAGATE_XX_CLOSE_CNF | 16 |
| Chart 3-3: | Telegram data for OP_ANAGATE_XX_GET_INFO_CNF | 17 |
| Chart 3-4: | Telegram data for OP_ANAGATE_XX_DIO_WRITE_REQ..... | 18 |
| Chart 3-5: | Telegram data for OP_ANAGATE_XX_DIO_WRITE_CNF | 18 |
| Chart 3-6: | Telegram data for OP_ANAGATE_XX_DIO_READ_CNF | 20 |
| Chart 3-7: | Telegram data for OP_ANAGATE_I2C_RESET_CNF | 23 |
| Chart 3-8: | Telegram data for OP_ANAGATE_I2C_SET_GLOBALS_REQ..... | 24 |
| Chart 3-9: | Telegram data for OP_ANAGATE_I2C_SET_GLOBALS_CNF | 24 |
| Chart 3-10: | Telegram data for OP_ANAGATE_I2C_GET_GLOBALS_CNF..... | 25 |
| Chart 3-11: | Telegram data for OP_ANAGATE_I2C_WRITE_REQ..... | 26 |
| Chart 3-12: | Telegram data for OP_ANAGATE_I2C_WRITE_CNF | 27 |
| Chart 3-13: | Telegram data for OP_ANAGATE_I2C_READ_REQ..... | 28 |
| Chart 3-14: | Telegram data for OP_ANAGATE_I2C_READ_CNF | 29 |
| Chart 3-15: | Telegram data for OP_ANAGATE_I2C_EEPROM_WRITE_REQ | 30 |
| Chart 3-16: | Telegram data for OP_ANAGATE_I2C_EEPROM_WRITE_CNF..... | 31 |
| Chart 3-17: | Telegram data for OP_ANAGATE_I2C_EEPROM_READ_REQ | 32 |
| Chart 3-18: | Telegram data for OP_ANAGATE_I2C_EEPROM_READ_CNF..... | 33 |
| Chart 3-19: | Telegram data for OP_ANAGATE_I2C_SEQUENCE_REQ..... | 34 |
| Chart 3-20: | Read command of the OP_ANAGATE_I2C_SEQUENCE_REQ..... | 34 |
| Chart 3-21: | Write command of the OP_ANAGATE_I2C_SEQUENCE_REQ..... | 35 |
| Chart 3-22: | Telegram data for OP_ANAGATE_I2C_SEQUENCE_CNF | 36 |
| Chart 3-23: | Telegram data for OP_ANAGATE_I2C_STATUS_CNF..... | 37 |
| Chart 3-24: | Telegram data for OP_ANAGATE_CAN_DATA_REQ..... | 41 |
| Chart 3-25: | Telegram data for OP_ANAGATE_CAN_DATA_CNF | 42 |
| Chart 3-26: | Telegram data for OP_ANAGATE_CAN_DATA_IND with activated timestamp | 43 |
| Chart 3-27: | Telegram data for OP_ANAGATE_CAN_DATA_RSP | 44 |
| Chart 3-28: | Telegram data for OP_ANAGATE_CAN_SET_CONFIG_REQ..... | 45 |
| Chart 3-29: | Telegram data for OP_ANAGATE_CAN_SET_CONFIG_CNF | 46 |
| Chart 3-30: | Telegram data for OP_ANAGATE_CAN_GET_CONFIG_CNF | 47 |
| Chart 3-31: | Telegram data for OP_ANAGATE_CAN_SET_GLOBALS_REQ..... | 49 |
| Chart 3-32: | Telegram data for OP_ANAGATE_CAN_SET_GLOBALS_CNF | 49 |
| Chart 3-33: | Telegram data for OP_ANAGATE_CAN_GET_GLOBALS_CNF..... | 51 |
| Chart 3-34: | Telegram data for OP_ANAGATE_CAN_SET_FILTER_CNF | 52 |
| Chart 3-35: | Telegram data for OP_ANAGATE_CAN_SET_FILTER_CNF | 53 |
| Chart 3-36: | Telegram data for OP_ANAGATE_CAN_GET_FILTER_CNF..... | 54 |
| Chart 3-37: | Telegram data for OP_ANAGATE_CAN_SET_TIME_REQ..... | 55 |
| Chart 3-38: | Telegram data for OP_ANAGATE_CAN_SET_TIME_CNF | 55 |
| Chart 3-39: | Telegram data for OP_ANAGATE_CAN_RESTART_CNF | 56 |
| Chart 3-40: | Telegram data for OP_ANAGATE_CAN_ZCARD_PI_DATA_IND | 58 |
| Chart 3-41: | Telegram data for OP_ANAGATE_CAN_ZCARD_PI_DATA_RSP | 58 |
| Chart 3-42: | Telegram data for OP_ANAGATE_CAN_SET_ZCARD_PI_DATA_REQ | 59 |

| | | |
|-------------|---|----|
| Chart 3-43: | Telegram data for OP_ANAGATE_CAN_SET_ZCARD_PI_CNF..... | 59 |
| Chart 3-44: | Telegram data for OP_ANAGATE_CAN_GET_ZCARD_PI_DATA_CNF | 61 |
| Chart 3-45: | Telegram data for OP_ANAGATE_SPI_DATA_REQ..... | 64 |
| Chart 3-46: | Telegram data for OP_ANAGATE_SPI_DATA_CNF | 65 |
| Chart 3-47: | Telegram data for OP_ANAGATE_SPI_SEQUENCE_REQ..... | 66 |
| Chart 3-48: | Data request command of the OP_ANAGATE_SPI_SEQUENCE_REQ | 66 |
| Chart 3-49: | Telegram data for OP_ANAGATE_SPI_SEQUENCE_CNF | 67 |
| Chart 3-50: | Telegram data for OP_ANAGATE_SPI_SET_GLOBALS_REQ..... | 68 |
| Chart 3-51: | Telegram data for OP_ANAGATE_SPI_SET_GLOBALS_CNF | 69 |
| Chart 3-52: | Telegram data for OP_ANAGATE_SPI_GET_GLOBALS_CNF | 70 |
| Chart 3-53: | Telegram data for OP_ANAGATE_AD_READ_CNF | 79 |

Tables

| | | |
|-------------|--|----|
| Table 3-1: | Return value for OP_ANAGATE_XX_OPEN_CNF | 15 |
| Table 3-2: | Return value for OP_ANAGATE_XX_CLOSE_CNF | 16 |
| Table 3-3: | Return value for OP_ANAGATE_XX_GET_INFO_CNF | 18 |
| Table 3-4: | Return value für OP_ANAGATE_XX_DIO_WRITE_CNF | 19 |
| Table 3-5: | Return value für OP_ANAGATE_XX_DIO_READ_CNF | 20 |
| Table 3-6: | Command IDs for AnaGate I2C devices | 22 |
| Table 3-7: | Return value for OP_ANAGATE_I2C_RESET_CNF | 23 |
| Table 3-8: | Return value for OP_ANAGATE_I2C_SET_GLOBALS_CNF | 24 |
| Table 3-9: | Return value for OP_ANAGATE_I2C_GET_GLOBALS_CNF | 25 |
| Table 3-10: | Return value for OP_ANAGATE_I2C_WRITE_CNF | 27 |
| Table 3-11: | Return value for OP_ANAGATE_I2C_READ_CNF | 29 |
| Table 3-12: | Return value for OP_ANAGATE_I2C_EEPROM_WRITE_CNF | 31 |
| Table 3-13: | Return value for OP_ANAGATE_I2C_EEPROM_READ_CNF | 33 |
| Table 3-14: | Return value for OP_ANAGATE_I2C_SEQUENCE_CNF | 36 |
| Table 3-15: | Return value for OP_ANAGATE_I2C_STATUS_CNF | 37 |
| Table 3-16: | Command-IDs for AnaGate CAN / CAN uno/ CAN duo / CAN quattro | 39 |
| Table 3-17: | Return value for OP_ANAGATE_CAN_DATA_CNF | 42 |
| Table 3-18: | Return value for OP_ANAGATE_CAN_DATA_RSP | 44 |
| Table 3-19: | Return value for OP_ANAGATE_CAN_SET_CONFIG_CNF | 46 |
| Table 3-20: | Return value for OP_ANAGATE_CAN_GET_CONFIG_CNF | 47 |
| Table 3-21: | Return value for OP_ANAGATE_CAN_SET_GLOBALS_CNF | 50 |
| Table 3-22: | Return value for OP_ANAGATE_CAN_GET_GLOBALS_CNF | 51 |
| Table 3-23: | Return value for OP_ANAGATE_CAN_SET_FILTER_CNF | 53 |
| Table 3-24: | Return value for OP_ANAGATE_CAN_GET_FILTER_CNF | 54 |
| Table 3-25: | Return value for OP_ANAGATE_CAN_SET_TIME_CNF | 55 |
| Table 3-26: | Return value for OP_ANAGATE_CAN_RESTART_CNF | 56 |
| Table 3-27: | Return value for OP_ANAGATE_CAN_ZCARD_PI_DATA_RSP | 58 |
| Table 3-28: | Return value for OP_ANAGATE_CAN_SET_ZCARD_PI_CNF | 60 |
| Table 3-29: | Command-IDs for AnaGate SPI | 63 |
| Table 3-30: | Return value for OP_ANAGATE_SPI_DATA_CNF | 65 |
| Table 3-31: | Return value for OP_ANAGATE_SPI_SEQUENCE_CNF | 67 |
| Table 3-32: | Return value for OP_ANAGATE_SPI_SET_GLOBALS_CNF | 69 |
| Table 3-33: | Return value for OP_ANAGATE_SPI_GET_GLOBALS_CNF | 70 |
| Table 3-34: | Command-IDs for AnaGate DigitalIO | 72 |
| Table 3-35: | Command-IDs for AnaGate A/D | 76 |
| Table 3-36: | Return value for OP_ANAGATE_AD_READ_CNF | 79 |

1 Introduction

This manual describes the basic communication between an **AnaGate** device and any other system¹ (e.g. a PC) via TCP/IP. First of all, you will find a general description of the basic functions and structures of how the system communicates. Later chapters will introduce you to the specific parameters and methods (I²C, CAN, etc.).

We will just be concentrating on specific data exchange (e.g.. socket API). TCP/IP-specific methods are not discussed here.

1.1 Information about this manual

For detailed information on the specification and programming of interfaces 2C/CAN/RS232/..., refer to the relevant technical literature (see also "Literature").

Basic knowledge in programming TCP/IP interfaces (e.g. Socket Interface) is required.

1.2 Restrictions

All **AnaGate** series devices basically support communication with several partners at any one time. The number of supported connections is restricted and depends on the AnaGate device type.

Users and software developers are advised to keep possible side-effects in mind. Applications must be realised without any risk of negative effects.

¹ Referred to in the following as "partners"

2 General Interface

2.1 Setting up a connection

Communication between an **AnaGate** device and a partner is triggered by an active system connection initiated by the partner. The partner has to use the IP address of the AnaGate and the corresponding port of the specific application (see also [Chart 2-1](#)). Which local port is used is immaterial, provided it is only used once.

| AnaGate Type | TCP Port |
|---|----------|
| AnaGate I2C | 5000 |
| AnaGate CAN | 5001 |
| AnaGate CAN uno/duo/quattro interface A | 5001 |
| AnaGate CAN duo/quattro interface B | 5101 |
| AnaGate CAN quattro interface C | 5201 |
| AnaGate CAN quattro interface D | 5301 |
| AnaGate SPI | 5002 |
| AnaGate RS232 | 5003 |
| AnaGate DigitalIO | 5004 |
| AnaGate Audio | 5005 |
| AnaGate A/D | 5006 |
| AnaGate Phone | 5007 |

Chart 2-1: Overview of the various AnaGate ports

2.2 Data transmission

Data is exchanged between an **AnaGate** device and a partner by sending and receiving TCP telegrams with the following structures:

2.2.1 Telegram structure

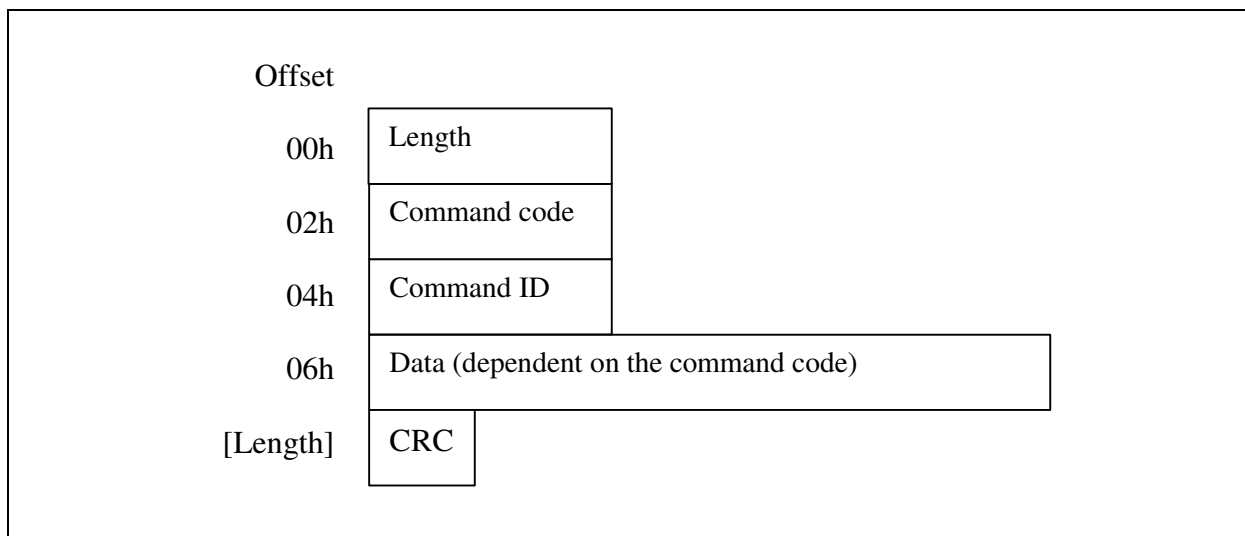


Chart 2-2: General structure of a telegram

2.2.1.1 Length

The length of the command is represented as a 16-bit value ("Little Endian" format), which contains all the subsequent data from the command code to the CRC.

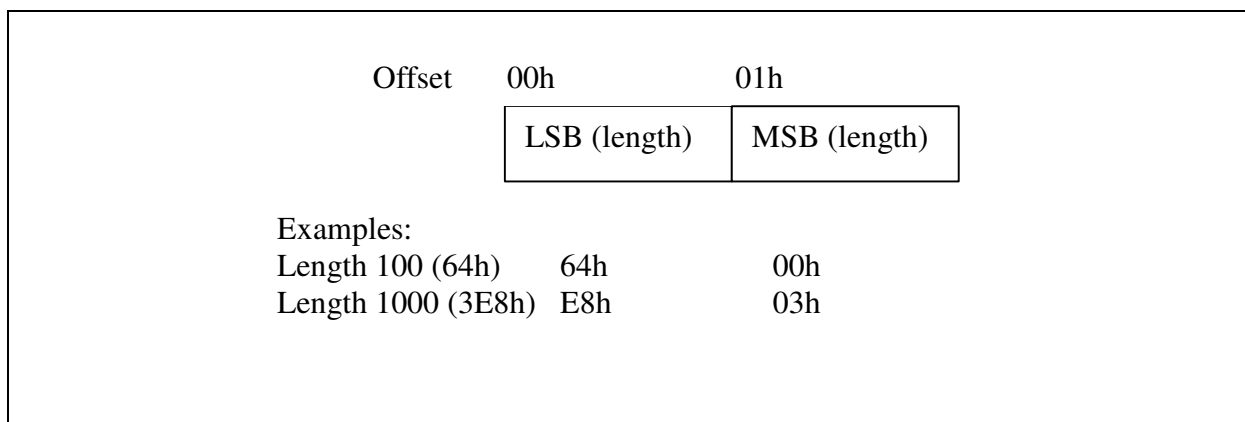


Chart 2-3: Length

2.2.1.2 Command code

The length of the command is represented as a 16-bit value ("Little Endian" format), which contains the following information:

- Bit 15:

Defines the type of command:

- Request / Indication (Bit 15 = 0)

A "request" defines an order to **AnaGate** to perform an action (e.g. **AnaGate** should send data).

An "indication" defines a message from **AnaGate** to display a piece of information (e.g. **AnaGate** has received data).

- Confirm / Response (Bit 15 = 1)

A "Confirm" confirms a "Request".

A "Response" confirms an "Indication".

- Bit 8 – 14

Unambiguously identifies a specific **AnaGate** and contains the following values:

- **AnaGate** I2C (1)
- **AnaGate** CAN / CAN uno/ CAN duo / CAN Quattro (2)
- **AnaGate** SPI(3)
- **AnaGate** RS232 (4)
- **AnaGate** DigitalIO (5)
- **AnaGate** Audio (6)
- **AnaGate** A/D (7)
- **AnaGate** Phone (8)

- Bit 0 – 7

Unambiguously identifies a specific command.

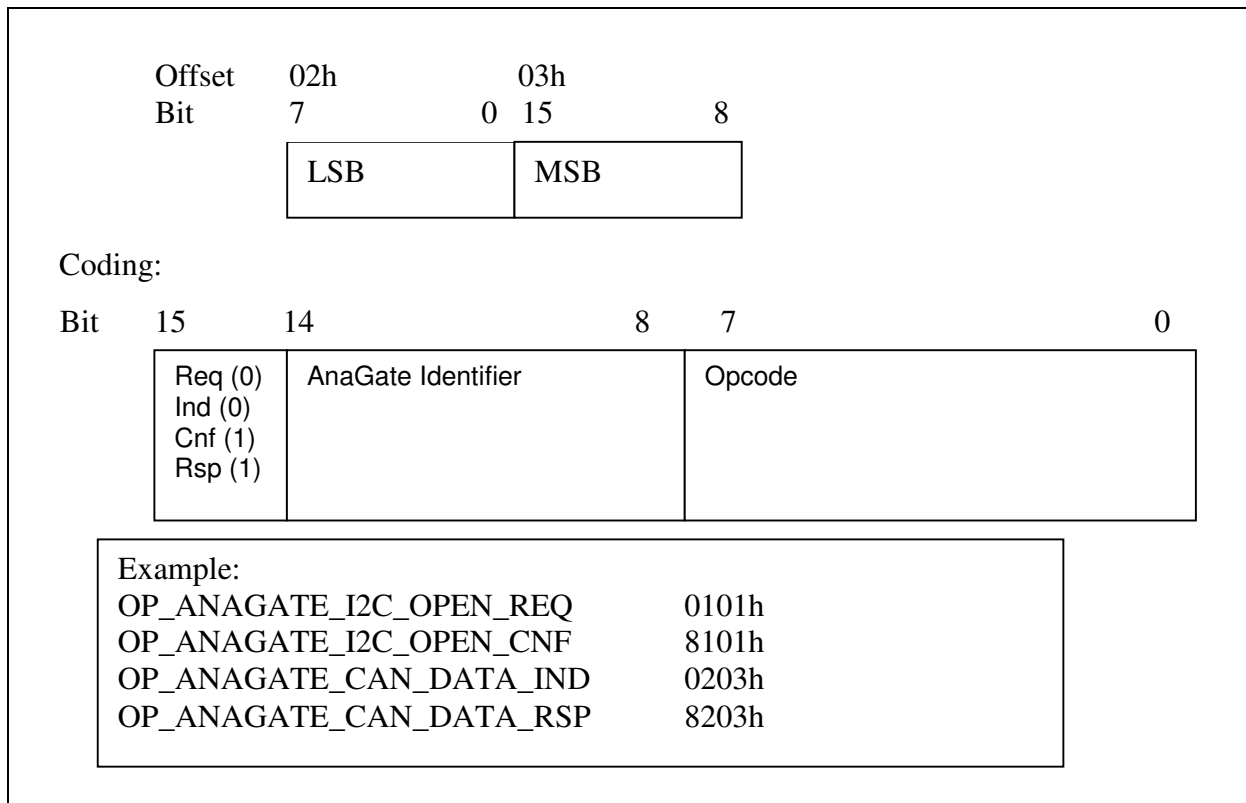


Chart 2-4: Command code

2.2.1.3 Command ID

The command ID is specified by a partner sending a message (**AnaGate** or partner) and has to be returned when acknowledging in order to coordinate "Request/Confirm" and "Indication/Response".

The command ID is represented as a 16-bit value ("Little Endian" format) as follows:

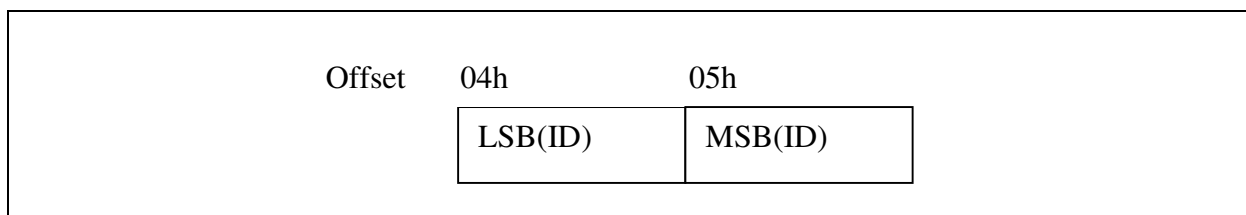


Chart 2-5: Command ID

AnaGate devices use their own sequential command IDs for requests and indications for each TCP session. The ID is initially set to "1" when the connection is set up.

2.2.1.4 Data

All the information required for each command code is handed over as data. The structures of the data are outlined in the descriptions of the respective command codes.

2.2.1.5 Checksum CRC

A byte calculated by XOR from all the existing bytes, without the length or CRC, is used as a checksum.

2.2.1.6 Examples of telegrams

| OP_ANAGATE_I2C_OPEN_CNF | | | | | | | | | | | |
|-------------------------|--------|-----|--------------|-----|------------|-----|----------|-----|-----|-----|-----|
| Offset | 00h | 01h | 02h | 03h | 04h | 05h | 06h | 07h | 08h | | |
| | 07h | 00h | 01h | 81h | 02h | 00h | 01h | 00h | 83h | | |
| | Length | | Command code | | Command ID | | Cnf code | | | | |
| OP_ANAGATE_CAN_DATA_IND | | | | | | | | | | | |
| Offset | 00h | 01h | 02h | 03h | 04h | 05h | 06h | 07h | 08h | 09h | 0Ah |
| | 09h | 00h | 03h | 02h | 02h | 00h | 01h | 02h | 03h | 04h | 54h |
| | Length | | Command code | | Command ID | | Data | | | | |

Chart 2-6: Examples of telegrams

2.3 Closing a connection

An existing connection between an **AnaGate** device and a partner can be closed by the partner at any time.

Under some circumstances, however, any commands that are still pending could be discarded and no longer executed by **AnaGate**.

The connection must be reopened to transmit data again.

3 Specific Telegrams

3.1 All AnaGate devices

In the following all telegrams are described which are supported by all AnaGate devices.

3.1.1 OP_ANAGATE_XX_OPEN_REQ

The Open command initiates the setting up of a logical connection to an AnaGate device. No further useful data is transmitted in this command.

3.1.2 OP_ANAGATE_XX_OPEN_CNF

The Confirmation acknowledges the previously performed Open command. An 8-bit return value is returned as useful data.

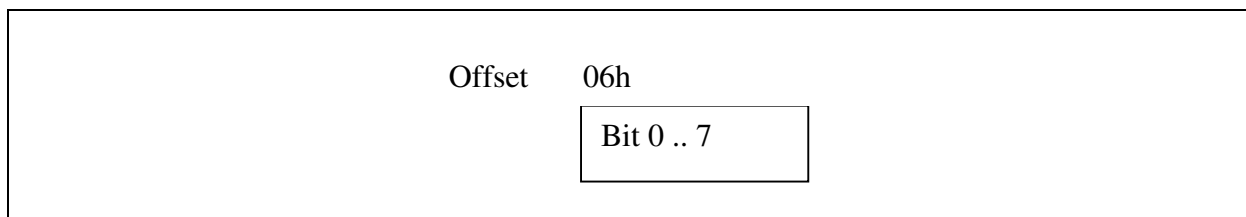


Chart 3-1: Telegram data for OP_ANAGATE_XX_OPEN_CNF

The following Return value may be given:

| Return value | Result |
|--------------|---|
| 00h | Open command was successful |
| 01h | Maximum number of simultaneous connections reached. |
| FFh | Open command was not successful |

Table 3-1: Return value for OP_ANAGATE_XX_OPEN_CNF

Remark:

If the error „maximum number of connections reached“ is raised, the AnaGate send immediately an OP_ANAGATE_XX_OPEN_CNF telegram. The TCP connection is cancelled to unblock the device for further requests. It could happen that the confirmation arrives at the partner before the OP_ANAGATE_XX_OPEN_REQ telegram is sended.

3.1.3 OP_ANAGATE_XX_CLOSE_REQ

The Close command terminates the logical connection to an AnaGate device. This closes down the TCP connection after sending the confirmation. No further useful data is transmitted in this command.

3.1.4 OP_ANAGATE_XX_CLOSE_CNF

The Confirmation acknowledges the previously executed Close Request command. An 8-bit return value is returned as useful data.

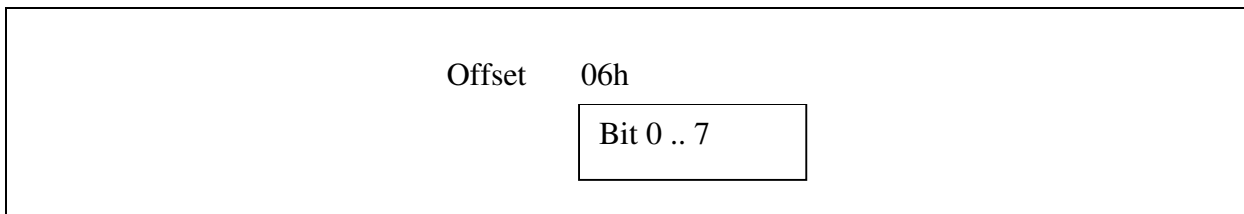


Chart 3-2: Telegram data for OP_ANAGATE_XX_CLOSE_CNF

The following Return value may be given:

| Return value | Result |
|--------------|----------------------------------|
| 00h | Close command was successful |
| FFh | Close command was not successful |

Table 3-2: Return value for OP_ANAGATE_XX_CLOSE_CNF

3.1.5 OP_ANAGATE_XX_GET_INFO_REQ

The Get Info Request reads back the global information from the AnaGate device. No further useful data is transmitted in this command.

3.1.6 OP_ANAGATE_XX_GET_INFO_CNF

The Confirmation acknowledges the previously executed Get Info Request command. The following data is returned as useful data:

- Return Code
The Return value given in Table 3-3 can be returned.
- Software-Version
The version number consists of 3 numbers (major.minor.revision), which are stored in a 4-byte integer value.
- Hardware-Version
The version number consists of 3 numbers (major.minor.revision), which are stored in a 4-byte integer value.
- Serial number
Serial number of the AnaGate device (4 Byte).
- MAC-Address
MAC-Address of the AnaGate device (6 Byte).

All returned values of the Get Info Requests can be also queried via the HTTP interface of the AnaGate device.

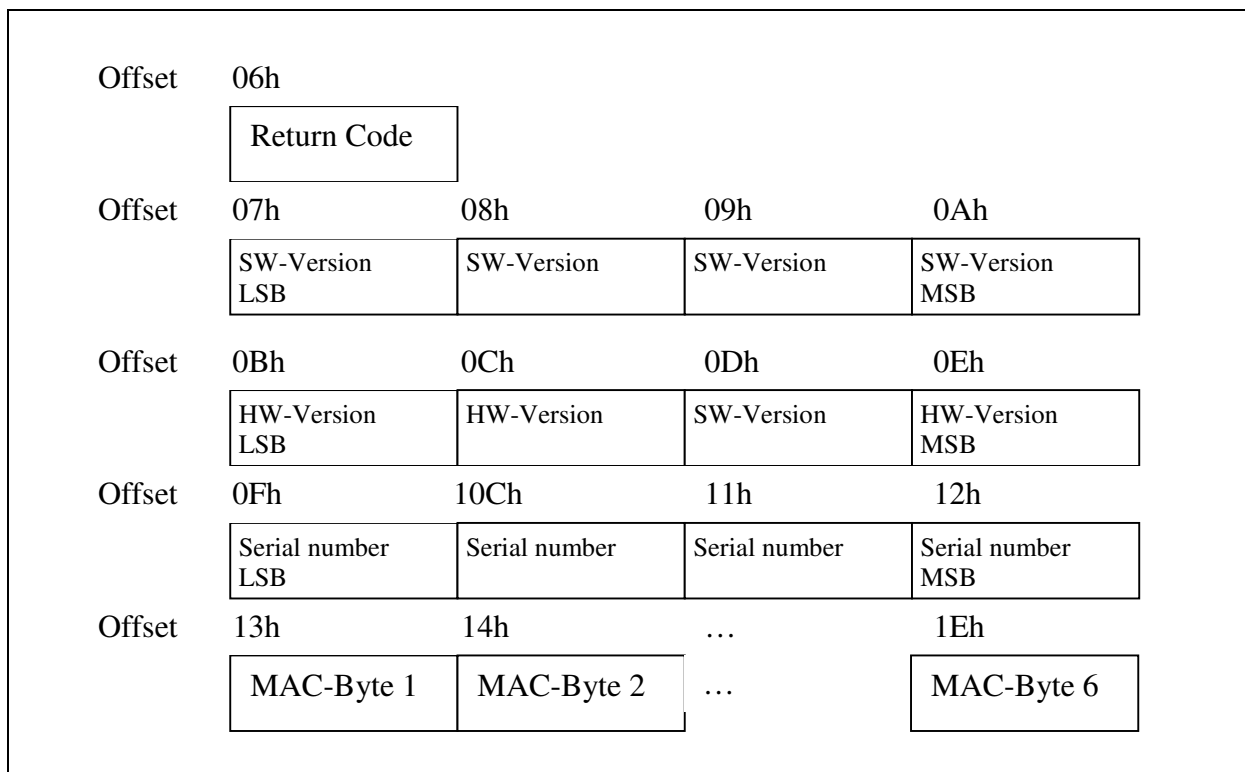


Chart 3-3: Telegram data for OP_ANAGATE_XX_GET_INFO_CNF

The following Return value may be given:

| Return value | Result |
|--------------|----------------------------------|
| 00h | Get Info command was successful. |
| 02h | Error reading internal memory. |

Table 3-3: Return value for *OP_ANAGATE_XX_GET_INFO_CNF*

3.1.7 OP_ANAGATE_XX_DIO_WRITE_REQ

The Write Digital Request writes new values to the digital output register of the AnaGate device. The following data is returned as useful data:

- Digital-Out Register

The new register value is passed on as a 32-bit value in Little Endian format. Only the bits 0 to 3 are used, the other bits are reserved.

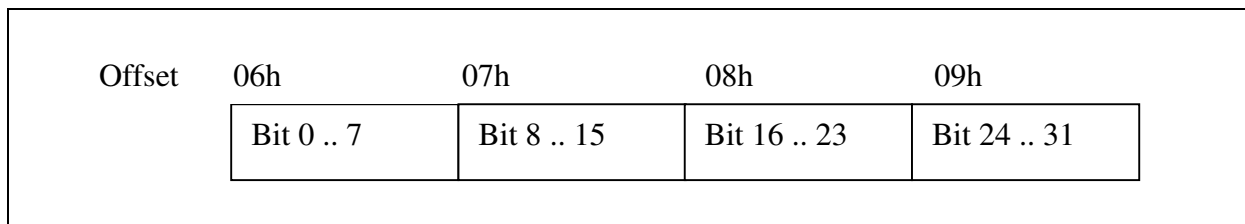


Chart 3-4: Telegram data for *OP_ANAGATE_XX_DIO_WRITE_REQ*

3.1.8 OP_ANAGATE_XX_DIO_WRITE_CNF

The Confirmation acknowledges the previously executed Write Digital Request command. An 8-bit return value is returned as useful data.

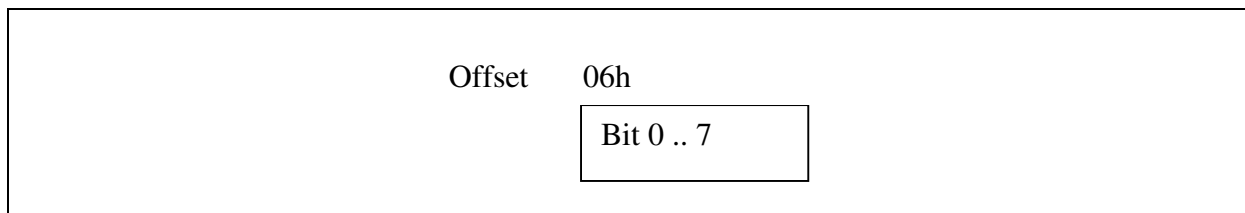


Chart 3-5: Telegram data for *OP_ANAGATE_XX_DIO_WRITE_CNF*

The following Return value may be given:

| Return Value | Result |
|---------------------|------------------------------------|
| 00h | Write command was successful. |
| FFh | Write command was not successful.h |

Table 3-4: Return value für OP_ANAGATE_XX_DIO_WRITE_CNF

3.1.9 OP_ANAGATE_XX_DIO_READ_REQ

The Read Digital Request reads the current values of the digital input and output registers of the AnaGate device. No further useful data is transmitted in this command.

3.1.10 OP_ANAGATE_XX_DIO_READ_CNF

The Confirmation acknowledges the previously executed Read Digital Request command. The following data is returned as useful data:

- Return Code
The Return value given in Table 3-5 can be returned.
- Digital-In Register
The current input register value is passed on as a 32-bit value in Little Endian format. Only the bits 0 to 3 are used, the other bits are reserved and are set to 0.
- Digital-Out Register
The current output register value is passed on as a 32-bit value in Little Endian format. Only the bits 0 to 3 are used, the other bits are reserved and are set to 0.

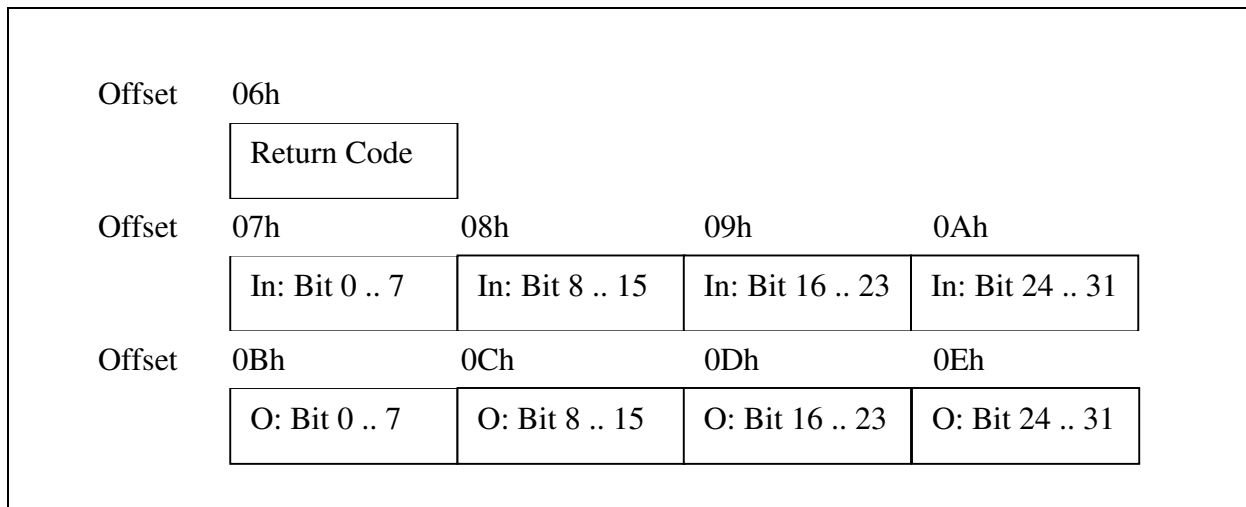


Chart 3-6: Telegram data for OP_ANAGATE_XX_DIO_READ_CNF

The following Return value may be given:

| Return Value | Result |
|--------------|-----------------------------------|
| 00h | Read command was successful. |
| FFh | Read command was not successful.. |

Table 3-5: Return value für OP_ANAGATE_XX_DIO_READ_CNF

3.2 AnaGate I2C

In order to set up a connection to an AnaGate I2C device, it is necessary to open a TCP connection with TCP port 5000.

The first command sent to AnaGate I2C must be a OP_ANAGATE_I2C_OPEN_REQ request. The remaining requests (Read/Write/ Reset/Close) can be performed once receipt has been acknowledged (OP_ANAGATE_I2C_OPEN_CNF).

A OP_ANAGATE_I2C_CLOSE_REQ request must be sent to close down the connection. The AnaGate I2C then returns the confirmation and independently closes the TCP connection.

| Command ID | Value |
|--------------------------------|-------|
| OP_ANAGATE_I2C_OPEN_REQ | 0101h |
| OP_ANAGATE_I2C_OPEN_CNF | 8101h |
| OP_ANAGATE_I2C_CLOSE_REQ | 0102h |
| OP_ANAGATE_I2C_CLOSE_CNF | 8102h |
| OP_ANAGATE_I2C_RESET_REQ | 0104h |
| OP_ANAGATE_I2C_RESET_CNF | 8104h |
| OP_ANAGATE_I2C_SET_GLOBALS_REQ | 0107h |
| OP_ANAGATE_I2C_SET_GLOBALS_CNF | 8107h |
| OP_ANAGATE_I2C_GET_GLOBALS_REQ | 0108h |
| OP_ANAGATE_I2C_GET_GLOBALS_CNF | 8108h |
| OP_ANAGATE_I2C_GET_INFO_REQ | 0109h |
| OP_ANAGATE_I2C_GET_INFO_CNF | 8109h |
| OP_ANAGATE_I2C_STATUS_REQ | 010ah |
| OP_ANAGATE_I2C_STATUS_CNF | 810ah |
| OP_ANAGATE_I2C_READ_REQ | 0110h |

| Command ID | Value |
|---------------------------------|-------|
| OP_ANAGATE_I2C_READ_CNF | 8110h |
| OP_ANAGATE_I2C_WRITE_REQ | 0111h |
| OP_ANAGATE_I2C_WRITE_CNF | 8111h |
| OP_ANAGATE_I2C_EEPROM_READ_REQ | 0112h |
| OP_ANAGATE_I2C_EEPROM_READ_CNF | 8112h |
| OP_ANAGATE_I2C_EEPROM_WRITE_REQ | 0113h |
| OP_ANAGATE_I2C_EEPROM_WRITE_CNF | 8113h |
| OP_ANAGATE_I2C_SEQUENCE_REQ | 0114h |
| OP_ANAGATE_I2C_SEQUENCE_CNF | 8114h |
| OP_ANAGATE_I2C_DIO_WRITE_REQ | 0141h |
| OP_ANAGATE_I2C_DIO_WRITE_CNF | 8141h |
| OP_ANAGATE_I2C_DIO_READ_REQ | 0140h |
| OP_ANAGATE_I2C_DIO_READ_CNF | 8140h |

Table 3-6: Command IDs for AnaGate I2C devices

3.2.1 OP_ANAGATE_I2C_OPEN_REQ

For further information see 3.1.1 OP_ANAGATE_XX_OPEN_REQ.

3.2.2 OP_ANAGATE_I2C_OPEN_CNF

For further information see 3.1.2 OP_ANAGATE_XX_OPEN_CNF.

3.2.3 OP_ANAGATE_I2C_CLOSE_REQ

For further information see 3.1.3 OP_ANAGATE_XX_CLOSE_REQ.

3.2.4 OP_ANAGATE_I2C_CLOSE_CNF

For further information see 3.1.4 OP_ANAGATE_XX_CLOSE_CNF.

3.2.5 OP_ANAGATE_I2C_RESET_REQ

The Reset command resets the internal I2C Master component without altering the baud rate. No further useful data is transmitted in this command.

3.2.6 OP_ANAGATE_I2C_RESET_CNF

The Confirmation acknowledges the previously executed Reset Request command. An 8-bit return value is returned as useful data.

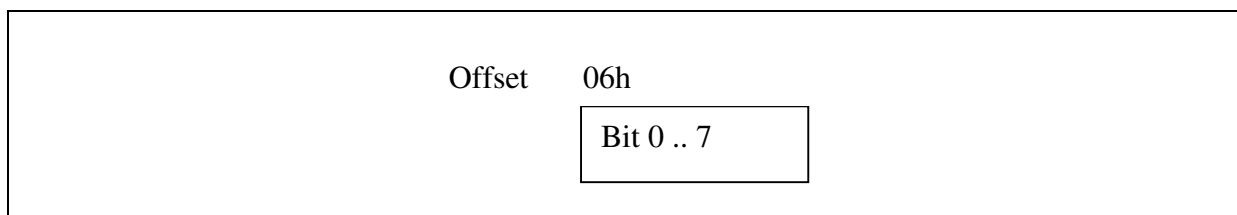


Chart 3-7: Telegram data for OP_ANAGATE_I2C_RESET_CNF

The following Return value may be given:

| Return value | Result |
|--------------|------------------------------|
| 00h | Reset command was successful |

Table 3-7: Return value for OP_ANAGATE_I2C_RESET_CNF

3.2.7 OP_ANAGATE_I2C_SET_GLOBALS_REQ

The Set Globals command configures the global device settings of the AnaGate I2C.

The baud rate is passed on as a 32-bit value in Little Endian Format:

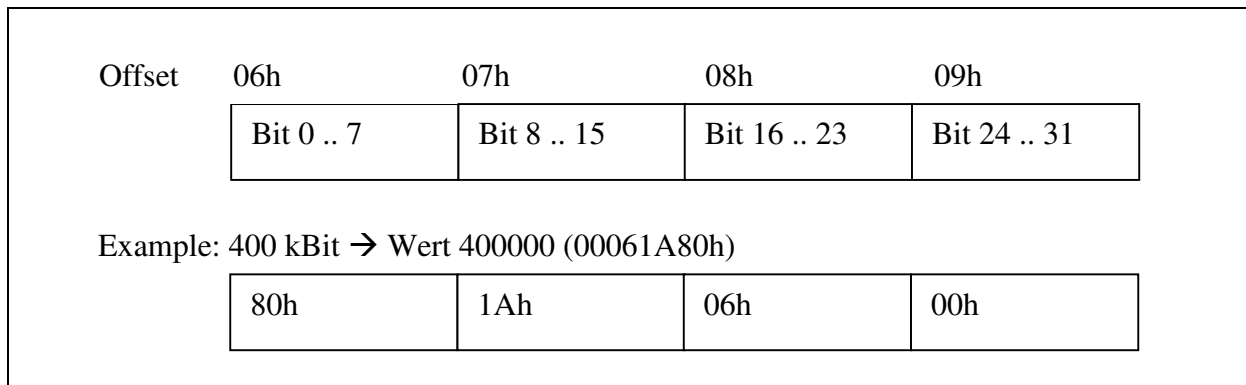


Chart 3-8: Telegram data for *OP_ANAGATE_I2C_SET_GLOBALS_REQ*

The currently valid baud rates are 100000 for 100 kBit and 400000 for 400 kBit.

3.2.8 OP_ANAGATE_I2C_SET_GLOBALS_CNF

The Confirmation acknowledges the previously executed Set Globals Request command. An 8-bit return value is returned as useful data.

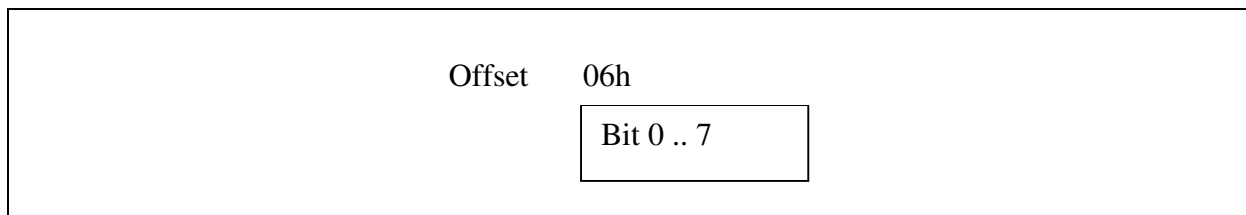


Chart 3-9: Telegram data for *OP_ANAGATE_I2C_SET_GLOBALS_CNF*

The following Return value may be given:

| Return value | Result |
|--------------|---|
| 00h | Set Globals command was successful. |
| FFh | Set Globals command was not successful. |

Table 3-8: Return value for *OP_ANAGATE_I2C_SET_GLOBALS_CNF*

3.2.9 OP_ANAGATE_I2C_GET_GLOBALS_REQ

The Get Globals command retrieves the global AnaGate device settings. No further useful data is transmitted in this command.

3.2.10 OP_ANAGATE_I2C_GET_GLOBALS_CNF

The Confirmation acknowledges the previously executed Get Globals Request command. The following data is returned as useful data:

- Return code
The Return value given in Table 3-11 can be returned.
- Baud rate
The baud rate currently used on the I2C bus is returned.

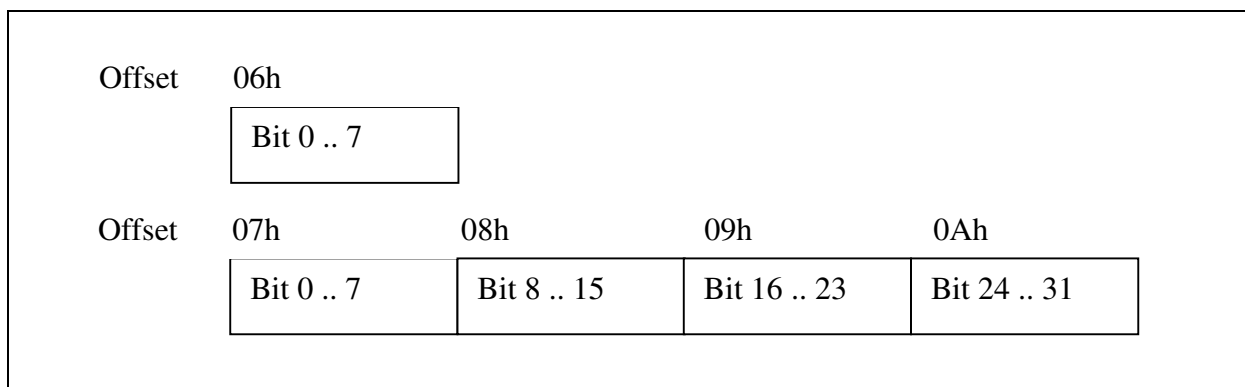


Chart 3-10: Telegram data for OP_ANAGATE_I2C_GET_GLOBALS_CNF

The following Return value may be given:

| Return value | Result |
|--------------|-------------------------------------|
| 00h | Get Globals command was successful. |

Table 3-9: Return value for OP_ANAGATE_I2C_GET_GLOBALS_CNF

3.2.11 OP_ANAGATE_I2C_WRITE_REQ

The Write command writes data on the I2C bus. It passes on the address of the Slave and all the bytes which are to be written on the bus.

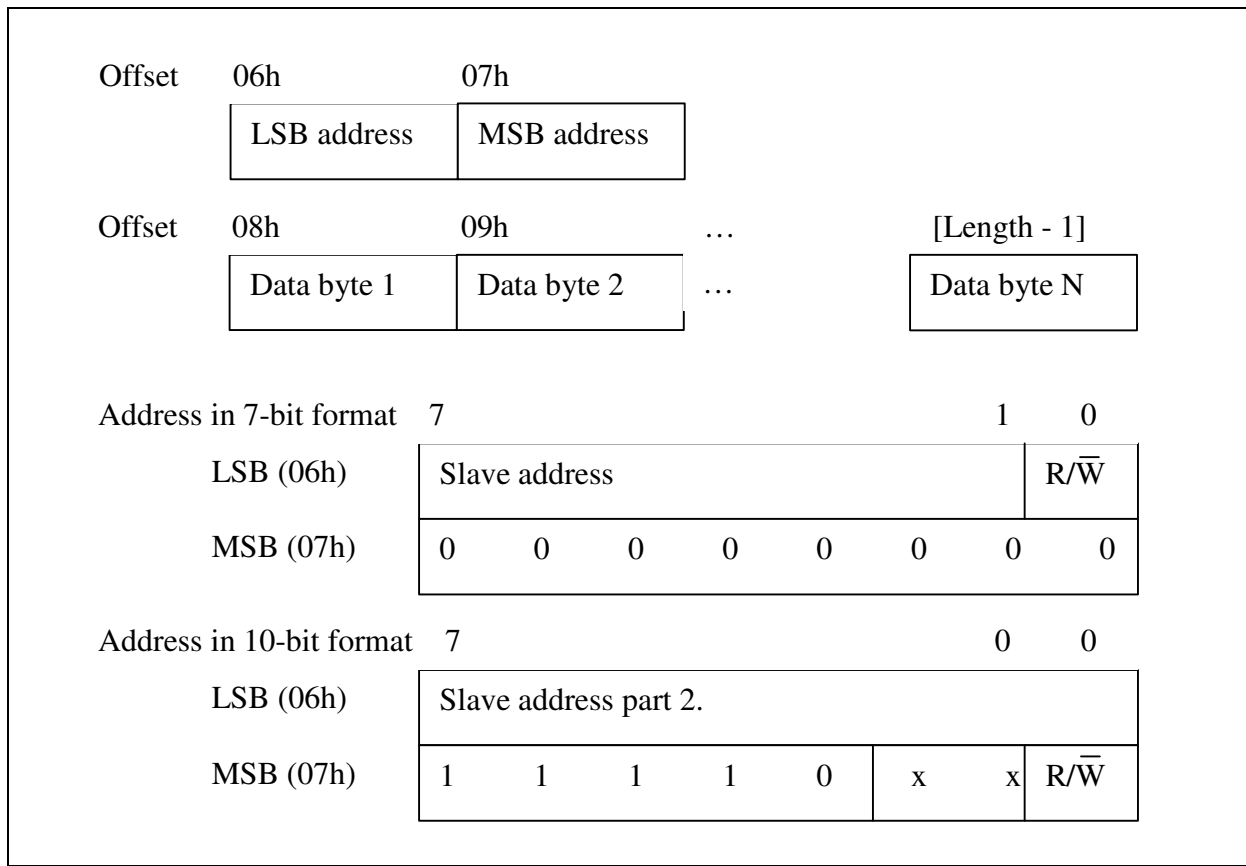


Chart 3-11: Telegram data for *OP_ANAGATE_I2C_WRITE_REQ*

With the 10-bit format, the MSB is the first byte to be transmitted on the I2C bus. Only then is the LSB passed on. With the 7-bit format, only the LSB is transmitted on the I2C bus.

The R/ \bar{W} bit in the address is automatically reset to 0 by this command.

The useful data to be written is passed on after the address as additional data. This is given out to the address on the I2C bus immediately afterwards.

3.2.12 **OP_ANAGATE_I2C_WRITE_CNF**

The Confirmation acknowledges the previously executed Write Request command.

The following data is returned as useful data:

- Return code

The Return value given in Table 3-10 is returned.

- Error byte (optional)

If the write failed, this parameter returns the byte number, where the error has occurred.

For example if the forth byte is NACK'd, the return code is 0x20 and the error byte 0x04. If the I2C address is NACK'd, the error byte is set to 0.

If the write was successful, this parameter was not transmitted.

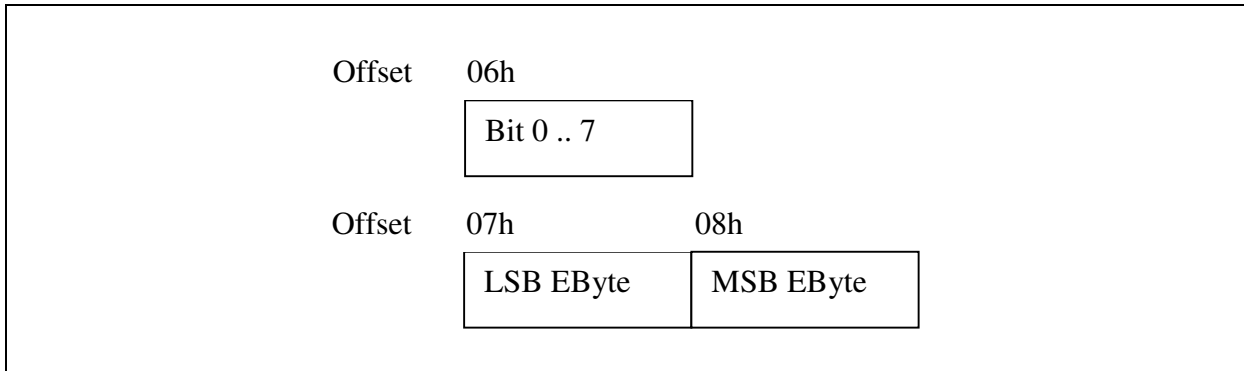


Chart 3-12: Telegram data for `OP_ANAGATE_I2C_WRITE_CNF`

The following Return value may be given:

| Return value | Result |
|--------------|--------------------------------------|
| 00h | Write command was successful |
| 20h | The I2C Slave has returned an NAK |
| 21h | The I2C Slave did not respond at all |

Table 3-10: Return value for `OP_ANAGATE_I2C_WRITE_CNF`

3.2.13 `OP_ANAGATE_I2C_READ_REQ`

The Read command is used to read data on the I2C bus. The address of the Slave and the number of bytes which are to be read by the Slave are passed on.



Chart 3-13: Telegram data for `OP_ANAGATE_I2C_READ_REQ`

With the 10-bit format, the MSB is the first byte to be transmitted on the I2C bus. Only then is the LSB passed on. With the 7-bit format, only the LSB is transmitted on the I2C bus.

The R/ \bar{W} bit in the address is automatically set to 1 by this command.

The number of bytes to be read out is then transmitted by the I2C Slave to the address as additional data. The AnaGate I2C then waits for this number of characters.

3.2.14 `OP_ANAGATE_I2C_READ_CNF`

The Confirmation acknowledges the previously executed Read Request command. The following data is returned as useful data:

- Return code
The Return value given in Table 3-11 can be returned.
- Read data bytes
The data received by the I2C Slave is returned here. No data is returned if the Read command failed.

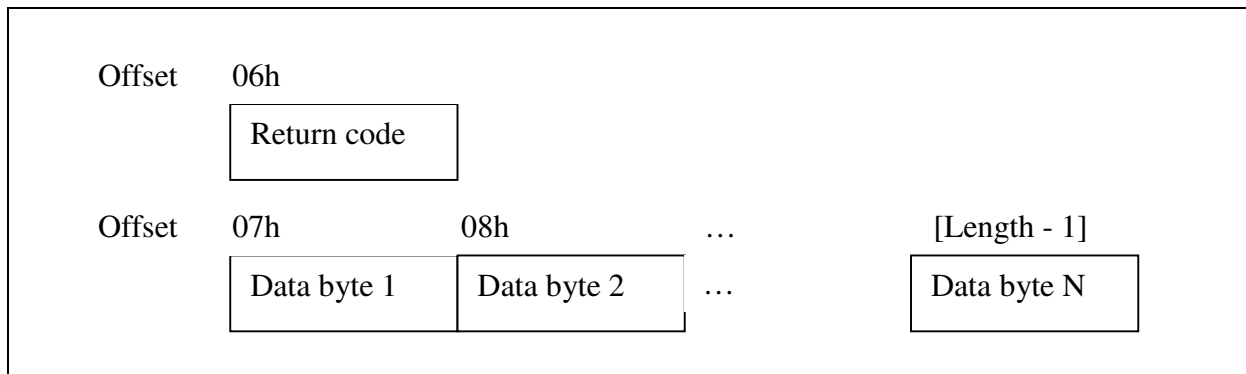


Chart 3-14: Telegram data for `OP_ANAGATE_I2C_READ_CNF`

The following Return value may be given:

| Return value | Result |
|--------------|--------------------------------------|
| 00h | Read command was successful |
| 20h | The I2C Slave has returned an NAK |
| 21h | The I2C Slave did not respond at all |

Table 3-11: Return value for `OP_ANAGATE_I2C_READ_CNF`

3.2.15 `OP_ANAGATE_I2C_EEPROM_WRITE_REQ`

The Write command is used to write data on the I2C bus. The address of the Slave, the length of the address (1-4 bytes), the address itself, and all data which is to be written on the bus are passed on.

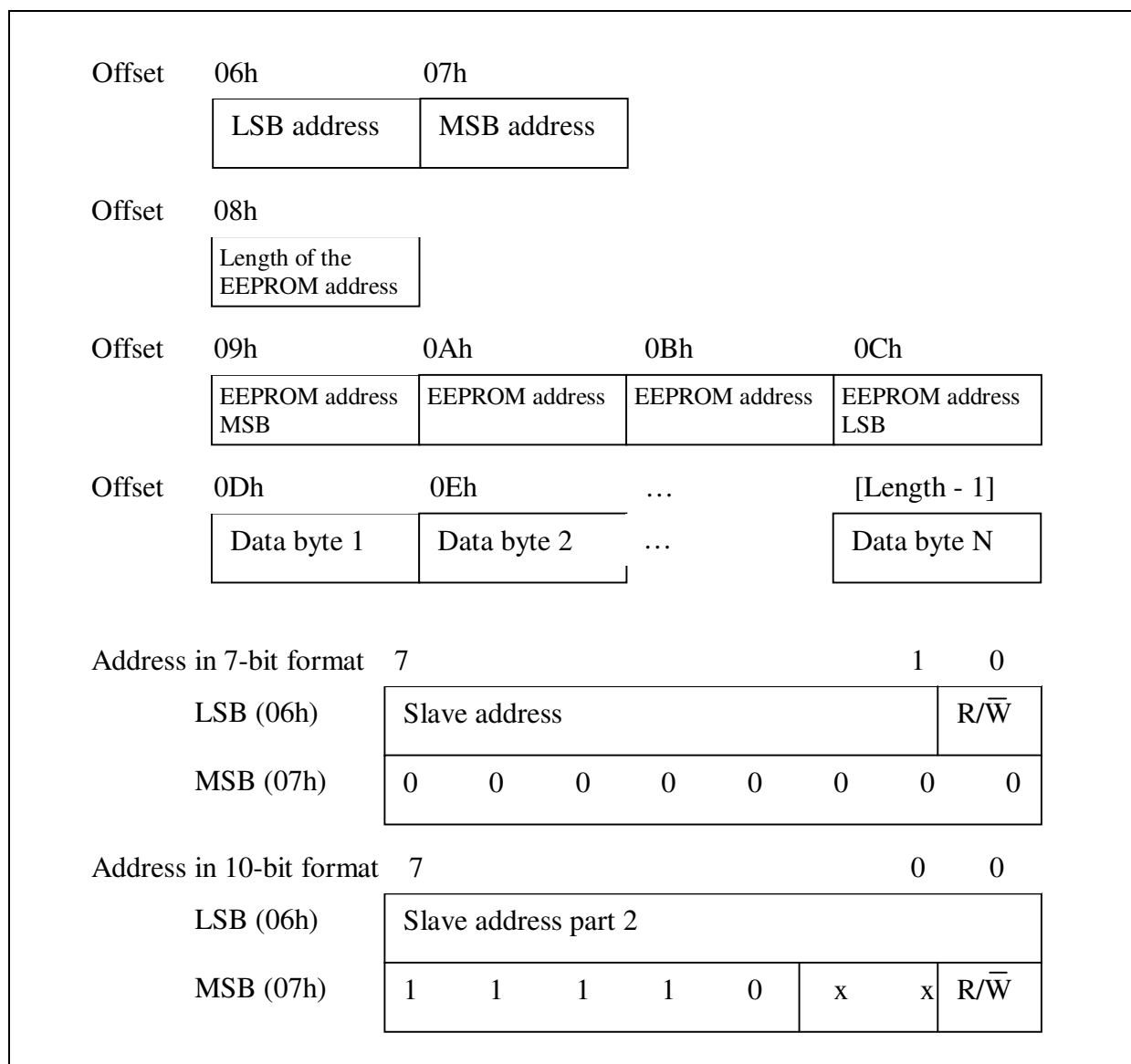


Chart 3-15: Telegram data for OP_ANAGATE_I2C_EEPROM_WRITE_REQ

With the 10-bit format, the MSB is the first byte to be transmitted on the I2C bus. Only then is the LSB transmitted. With the 7-bit format, only the LSB is transmitted on the I2C bus.

The R/ \bar{W} bit in the address is automatically reset to 0 by this command.

The length of the EEPROM address indicates the number of bytes necessary for the address itself. Values from 1 to 4 can be used.

The EEPROM address gives the address of the EEPROM on which all the data is to be written. Please note that the address begins with the MSB and ends with the LSB. This implies that the address 8000 (=1F40h) starting at Offset 09h as 00-00-1F-40 is passed on.

The useful data to be written to the EEPROM address is then transmitted as additional data. This is then passed on to the EEPROM in a single write cycle.

The user must pay attention to any restrictions relating to the specific EEPROM regarding addresses, page sizes, etc.

3.2.16 OP_ANAGATE_I2C_EEPROM_WRITE_CNF

The Confirmation acknowledges the previously executed OP_ANAGATE_I2C_EEPROM_WRITE_REQ command. An 8-bit return value is returned as useful data.

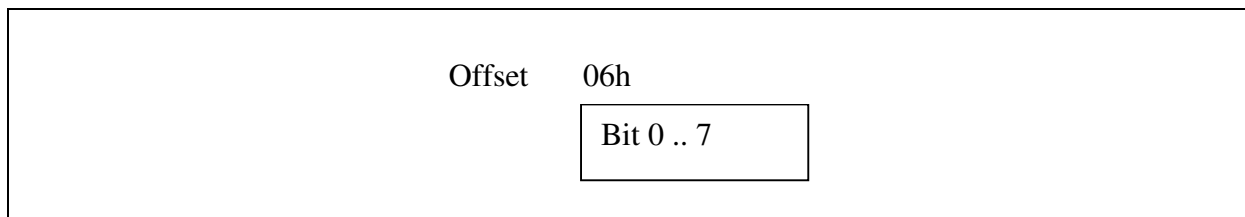


Chart 3-16: Telegram data for OP_ANAGATE_I2C_EEPROM_WRITE_CNF

The following Return value may be given:

| Return value | Result |
|--------------|--------------------------------------|
| 00h | EEPROM Write command was successful |
| 20h | The I2C Slave returned an NAK |
| 21h | The I2C Slave did not respond at all |

Table 3-12: Return value for OP_ANAGATE_I2C_EEPROM_WRITE_CNF

3.2.17 OP_ANAGATE_I2C_EEPROM_READ_REQ

The Read command is used to read data from the I2C EEPROM device. The address of the Slave and the number of bytes to be read by the Slave are passed on.

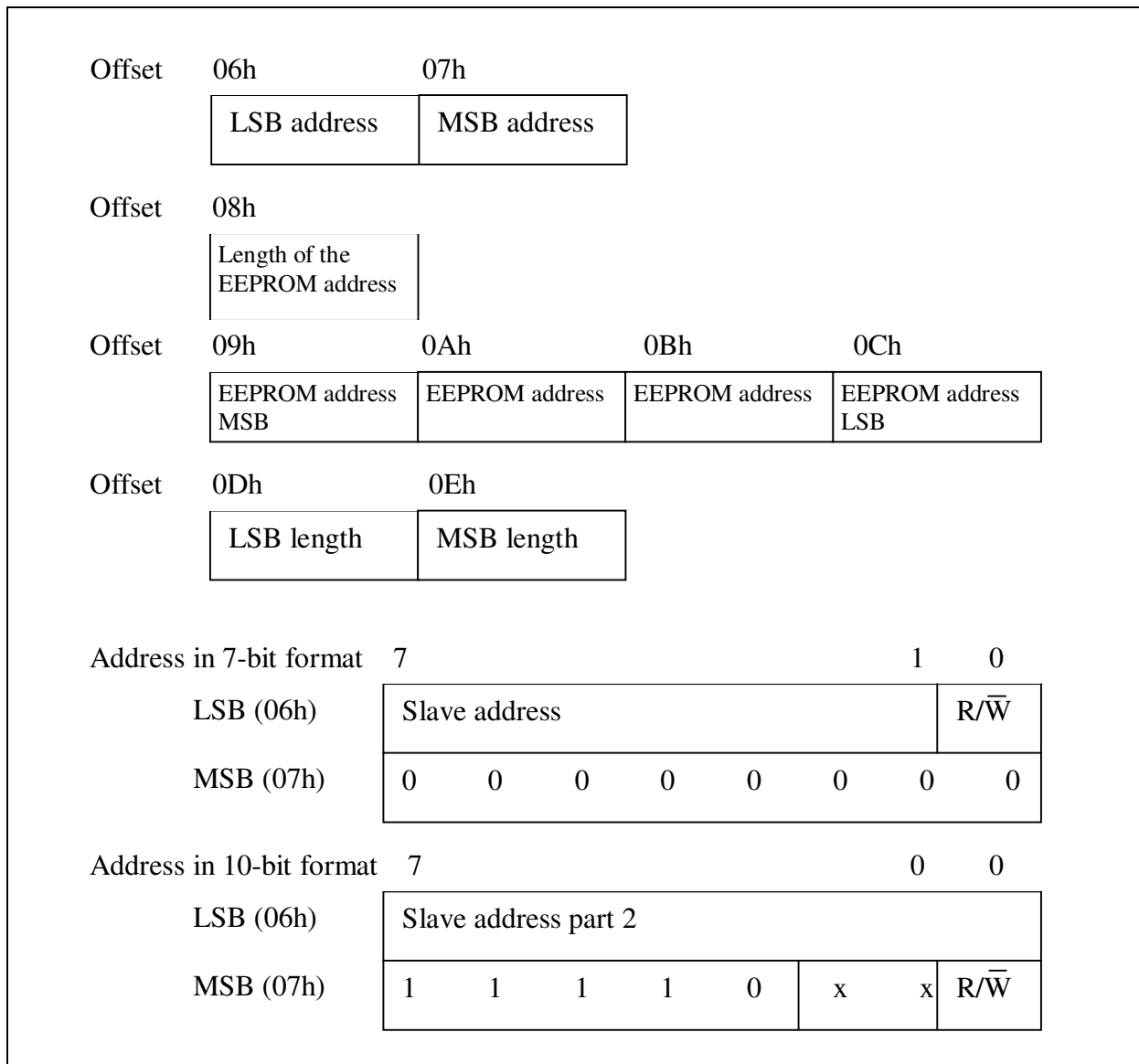


Chart 3-17: Telegram data for OP_ANAGATE_I2C_EEPROM_READ_REQ

With the 10-bit format, the MSB is the first byte to be transmitted on the I2C bus. Only then is the LSB transmitted. With the 7-bit format, only the LSB is transmitted on the I2C bus.

The R/ \bar{W} bit in the address is automatically reset to 0 by this command.

The length of the EEPROM address indicates the number of bytes necessary for the address itself. Values from 1 to 4 can be used.

The EEPROM address gives the address of the EEPROM on which all the data is to be written. Please note that the address begins with the MSB and ends with the LSB. This implies that the address 8000 (=1F40h) starting at Offset 09h as 00-00-1F-40 is passed on.

The number of bytes to be read by the I2C Slave is then transmitted to the EEPROM address as additional data. The AnaGate I2C device reads this number of bytes out of the EEPROM.

The user must pay attention to any restrictions relating to the specific EEPROM regarding addresses, page sizes, etc.

3.2.18 OP_ANAGATE_I2C_EEPROM_READ_CNF

The Confirmation acknowledges the previously executed Read Request command. The following data is returned as useful data:

- Return code
The Return value given in Table 3-10 can be returned.
- Read data bytes
The data received by the I2C Slave is returned here. No data is returned if the Read command failed.

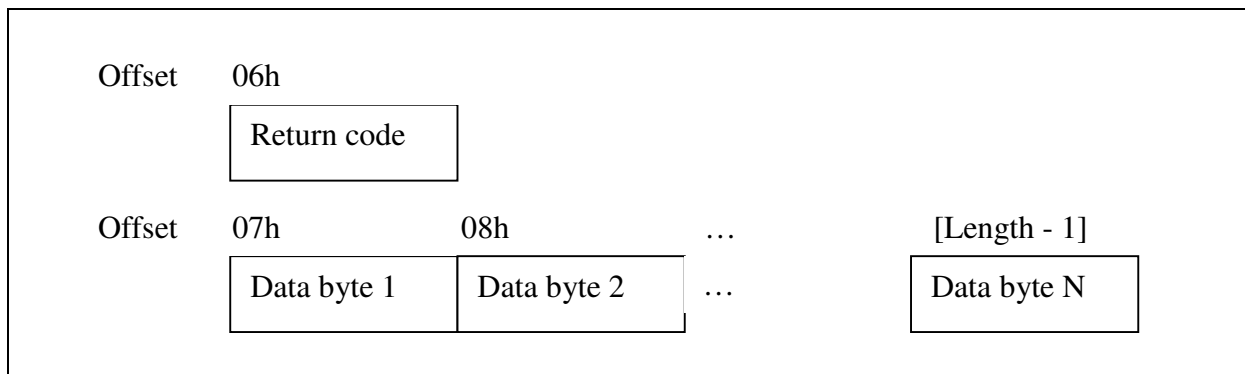


Chart 3-18: Telegram data for `OP_ANAGATE_I2C_EEPROM_READ_CNF`

The following Return value may be given:

| Return value | Result |
|--------------|--------------------------------------|
| 00h | EEPROM Read command was successful |
| 20h | The I2C Slave returned an NAK |
| 21h | The I2C Slave did not respond at all |

Table 3-13: Return value for `OP_ANAGATE_I2C_EEPROM_READ_CNF`

3.2.19 OP_ANAGATE_I2C_SEQUENCE_REQ

The Read command is used to write a sequence of write and read commands to an I2C device.

- Number of data bytes to read (sum of all single read requests)
- Command sequence

Read or/and write commands can be send to different I2C devices. The single commands are simply placed in the message buffer

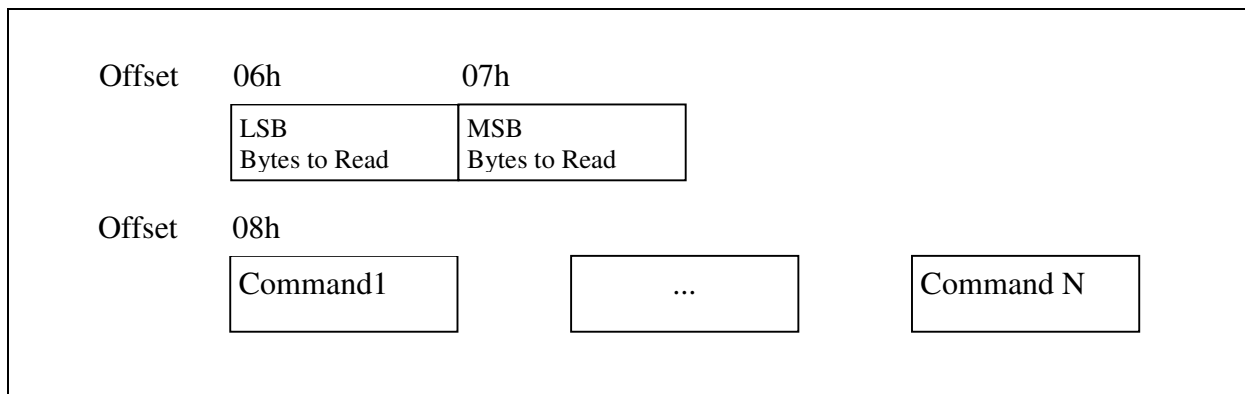


Chart 3-19: Telegram data for OP_ANAGATE_I2C_SEQUENCE_REQ

A read command has the following structure:

- Slave address
The slave address can be in 7 or 10 bit address format (see Chart 3.13), the R/W ibit in the address must be explicitly set to 1.
- Number of data bytes to read
Number of bytes, which are to be read from the I2C device. If the data is read successfully, it is returned by the confirmation message.

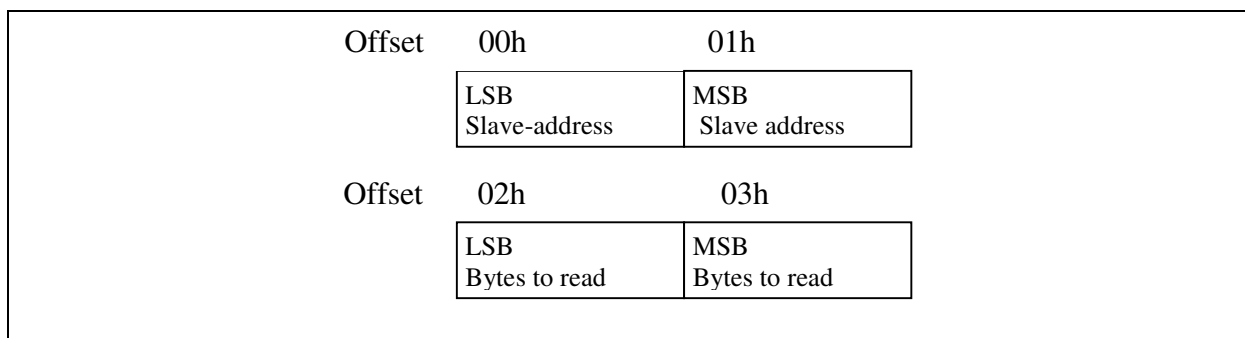


Chart 3-20: Read command of the OP_ANAGATE_I2C_SEQUENCE_REQ

A write command has the following structure:

- Slave address

The slave address can be in 7 or 10 bit address format (see Chart 3.13), the R/W ibit in the address must be explicitly set to 0.

- Number of data bytes
Number of the following data bytes, which are to write to the I2C device.
- Data bytes

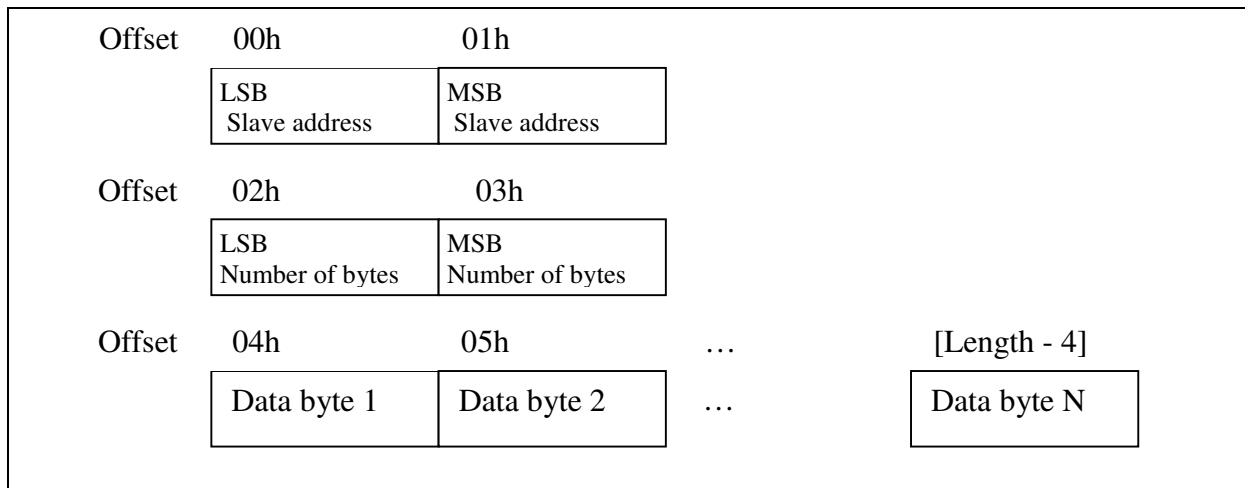


Chart 3-21: Write command of the `OP_ANAGATE_I2C_SEQUENCE_REQ`

3.2.20 `OP_ANAGATE_I2C_SEQUENCE_CNF`

The Confirmation acknowledges the previously executed Read Request command. The following data is returned as useful data:

- Return code
The Return value given in Table 3-14 can be returned.
- Error byte
If the write failed, this parameter returns the byte number, where the error has occurred.
For example if the forth byte is NACK'd, the return code is 0x20 and the error byte 0x04. If the I2C address is NACK'd, the error byte is set to 0.
If the write was successful, this parameter was not transmitted
- Read data bytes
The data received by the I2C Slave is returned here. No data is returned if the Read command failed.

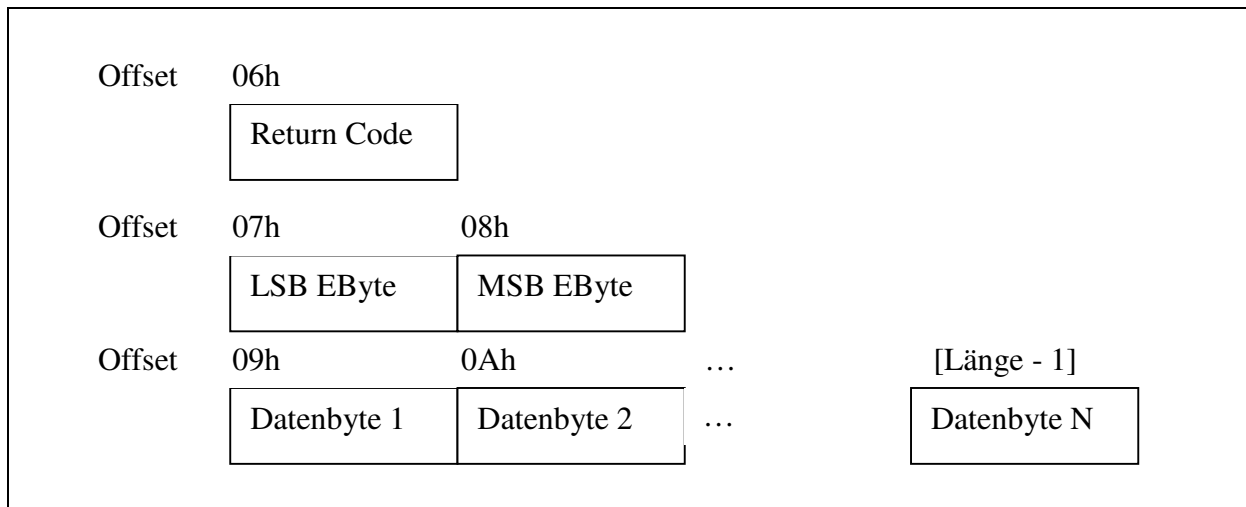


Chart 3-22: Telegram data for *OP_ANAGATE_I2C_SEQUENCE_CNF*

The following Return value may be given:

| Return value | Result |
|--------------|--------------------------------------|
| 00h | Sequence command was successful |
| 20h | The I2C Slave returned an NAK |
| 21h | The I2C Slave did not respond at all |

Table 3-14: Return value for *OP_ANAGATE_I2C_SEQUENCE_CNF*

3.2.21 OP_ANAGATE_I2C_GET_INFO_REQ

For further information see 3.1.5 *OP_ANAGATE_XX_GET_INFO_REQ*.

3.2.22 OP_ANAGATE_I2C_GET_INFO_CNF

For further information see 3.1.6 *OP_ANAGATE_XX_GET_INFO_CNF*.

3.2.23 OP_ANAGATE_I2C_STATUS_REQ

The Status Request reads back the status information from the AnaGate I2C. No further useful data is transmitted in this command.

3.2.24 OP_ANAGATE_I2C_STATUS_CNF

The Confirmation acknowledges the previously executed Status Request command. An 8-bit return value is returned as useful data.

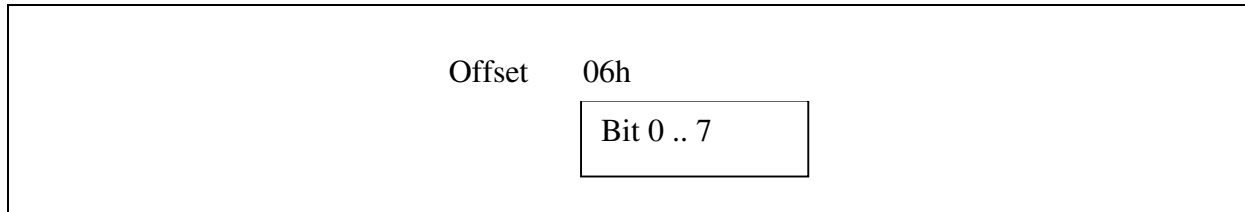


Chart 3-23: Telegram data for OP_ANAGATE_I2C_STATUS_CNF

The following Return value may be given:

| Return value | Result |
|--------------|-------------------------------|
| 00h | Status command was successful |

Table 3-15: Return value for OP_ANAGATE_I2C_STATUS_CNF

3.2.25 OP_ANAGATE_I2C_DIO_WRITE_REQ

For further information see 3.1.7 OP_ANAGATE_XX_DIO_WRITE_REQ.

3.2.26 OP_ANAGATE_I2C_DIO_WRITE_CNF

For further information see 3.1.8 OP_ANAGATE_XX_DIO_WRITE_CNF.

3.2.27 OP_ANAGATE_I2C_DIO_READ_REQ

For further information see 3.1.9 OP_ANAGATE_XX_DIO_READ_REQ.

3.2.28 OP_ANAGATE_I2C_DIO_READ_CNF

For further information see 3.1.10 OP_ANAGATE_XX_DIO_READ_CNF.

3.3 AnaGate CAN / AnaGate CAN uno, duo, quattro

In order to set up a connection to an AnaGate CAN or AnaGate CAN uno device, it is necessary to open a TCP connection with TCP port 5001. For the CAN interface A of a AnaGate CAN duo and AnaGate CAN quattro use TCP port 5001 and for the CAN interface B use TCP port 5101. For the CAN interface C of a AnaGate CAN quattro use TCP port 5201 and for the CAN interface D use TCP port 5301.

The first command sent to AnaGate CAN must be a OP_ANAGATE_CAN_OPEN_REQ request. The remaining requests (Read/Write/ Reset/Close) can be performed once receipt has been acknowledged (OP_ANAGATE_CAN_OPEN_CNF).

An OP_ANAGATE_CAN_CLOSE_REQ request must be sent to close down the connection. The AnaGate CAN / CAN duo then returns the confirmation and independently closes the TCP connection.

An AnaGate CAN / CAN duo can accept up to 4 simultaneous TCP connections.

| Befehls-ID | Wert |
|--|-------|
| OP_ANAGATE_CAN_OPEN_REQ | 0201h |
| OP_ANAGATE_CAN_OPEN_CNF | 8201h |
| OP_ANAGATE_CAN_CLOSE_REQ | 0202h |
| OP_ANAGATE_CAN_CLOSE_CNF | 8202h |
| OP_ANAGATE_CAN_DATA_REQ OP_ANAGATE_CAN_DATA_IND | 0203h |
| OP_ANAGATE_CAN_DATA_CNF OP_ANAGATE_CAN_DATA_RSP | 8203h |
| OP_ANAGATE_CAN_SET_CONFIG_REQ | 0205h |
| OP_ANAGATE_CAN_SET_CONFIG_CNF | 8205h |
| OP_ANAGATE_CAN_GET_CONFIG_REQ | 0206h |
| OP_ANAGATE_CAN_GET_CONFIG_CNF | 8206h |
| OP_ANAGATE_CAN_SET_GLOBALS_REQ | 0207h |

| Befehls-ID | Wert |
|----------------------------------|-------------|
| OP_ANAGATE_CAN_SET_GLOBALS_CNF | 8207h |
| OP_ANAGATE_CAN_GET_GLOBALS_REQ | 0208h |
| OP_ANAGATE_CAN_GET_GLOBALS_CNF | 8208h |
| OP_ANAGATE_CAN_GET_INFO_REQ | 0209h |
| OP_ANAGATE_CAN_GET_INFO_CNF | 8209h |
| OP_ANAGATE_CAN_SET_FILTER_REQ | 0220h |
| OP_ANAGATE_CAN_SET_FILTER_CNF | 8220h |
| OP_ANAGATE_CAN_GET_FILTER_REQ | 0221h |
| OP_ANAGATE_CAN_GET_FILTER_CNF | 8221h |
| OP_ANAGATE_CAN_SET_TIME_REQ | 020Bh |
| OP_ANAGATE_CAN_SET_TIME_CNF | 820Bh |
| OP_ANAGATE_CAN_RESTART_REQ | 020Ch |
| OP_ANAGATE_CAN_RESTART_CNF | 820Ch |
| OP_ANAGATE_CAN_DIO_WRITE_REQ | 0241h |
| OP_ANAGATE_CAN_DIO_WRITE_CNF | 8241h |
| OP_ANAGATE_CAN_DIO_READ_REQ | 0240h |
| OP_ANAGATE_CAN_DIO_READ_CNF | 8240h |
| OP_ANAGATE_CAN_ZCARD_PI_DATA_IND | 0250h |
| OP_ANAGATE_CAN_ZCARD_PI_DATA_RSP | 8250h |
| OP_ANAGATE_CAN_SET_ZCARD_PI_REQ | 0251h |
| OP_ANAGATE_CAN_SET_ZCARD_PI_CNF | 8251h |
| OP_ANAGATE_CAN_GET_ZCARD_PI_REQ | 0252h |
| OP_ANAGATE_CAN_GET_ZCARD_PI_CNF | 8252h |

Table 3-16: Command-IDs for AnaGate CAN / CAN uno/ CAN duo / CAN quattro

3.3.1 OP_ANAGATE_CAN_OPEN_REQ

For further information see 3.1.1 OP_ANAGATE_XX_OPEN_REQ.

3.3.2 OP_ANAGATE_CAN_OPEN_CNF

For further information see 3.1.2 OP_ANAGATE_XX_OPEN_CNF.

3.3.3 OP_ANAGATE_CAN_CLOSE_REQ

For further information see 3.1.3 OP_ANAGATE_XX_CLOSE_REQ.

3.3.4 OP_ANAGATE_CAN_CLOSE_CNF

For further information see 3.1.4 OP_ANAGATE_XX_CLOSE_CNF.

3.3.5 OP_ANAGATE_CAN_DATA_REQ

The Data Request command is used to send data to the CAN bus.

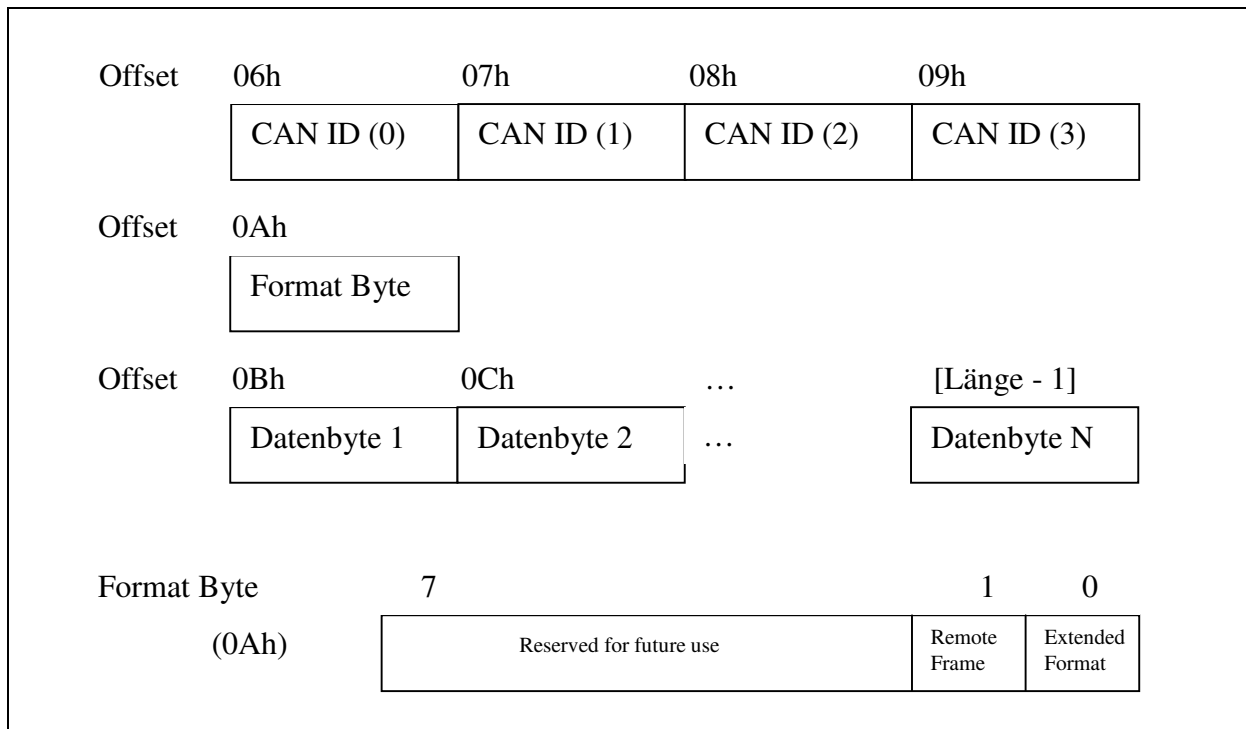


Chart 3-24: Telegram data for OP_ANAGATE_CAN_DATA_REQ

The following data has to be passed on:

- CAN-ID

The CAN ID is passed on as a 16-bit value in Little Endian format.

- Format byte

The format byte contains two bits which controls the message format.

- Remote Frame (Bit 1)

Is this bit set, the telegram is sent as remote frame without data. The number of data bytes is zero.

Is this bit unset, the telegram is sent as normal CAN telegram including data bytes.

- Extended Format (Bit 0)

Is this bit set, the CAN ID is interpreted as extended (29 Bit) and sended accordingly. Bits 5..7 in byte CAN ID (3) are ignored.

Is this bit unset, the CAN ID is in standard format (11 Bit). Bits 3..7 in CAN ID (1) and CAN ID (2) und (3) are ignored.

- Data bytes

The data bytes of the CAN telegram.

3.3.6 OP_ANAGATE_CAN_DATA_CNF

The Confirmation acknowledges the previously executed Data Request command. An 8-bit return value is returned as useful data. If the timestamp option (OP_ANAGATE_CAN_GET_GLOBALS_REQ) is activated, then additional 8 bytes are returned for system time, at which the telegram is acknowledged on the CAN bus.

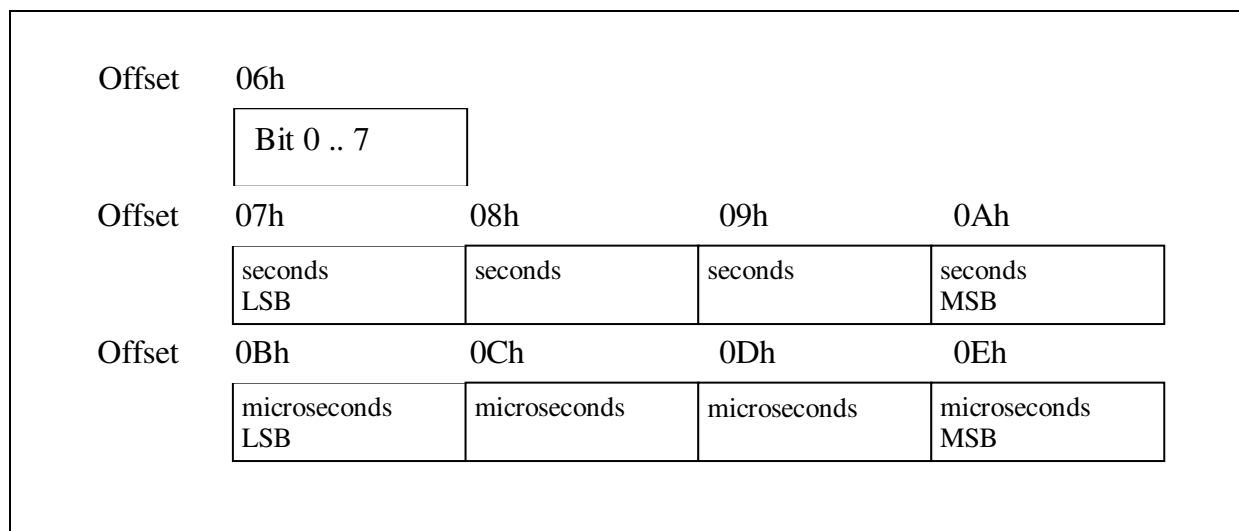


Chart 3-25: Telegram data for OP_ANAGATE_CAN_DATA_CNF

The following Return value may be given:

| Return value | Result |
|--------------|-------------------------------|
| 00h | Telegram was send successful. |
| 21h | Transmit error. |
| 22h | Internal buffer overflow. |
| 23h | Lost of arbitration. |

Table 3-17: Return value for OP_ANAGATE_CAN_DATA_CNF

3.3.7 OP_ANAGATE_CAN_DATA_IND

This command is created automatically by the AnaGate CAN, if a CAN telegram is received. The received CAN telegram is transmitted to the PC partner with this message. The Data Indication can occur at any time after a successful OP_ANAGATE_CAN_OPEN_REQ command.

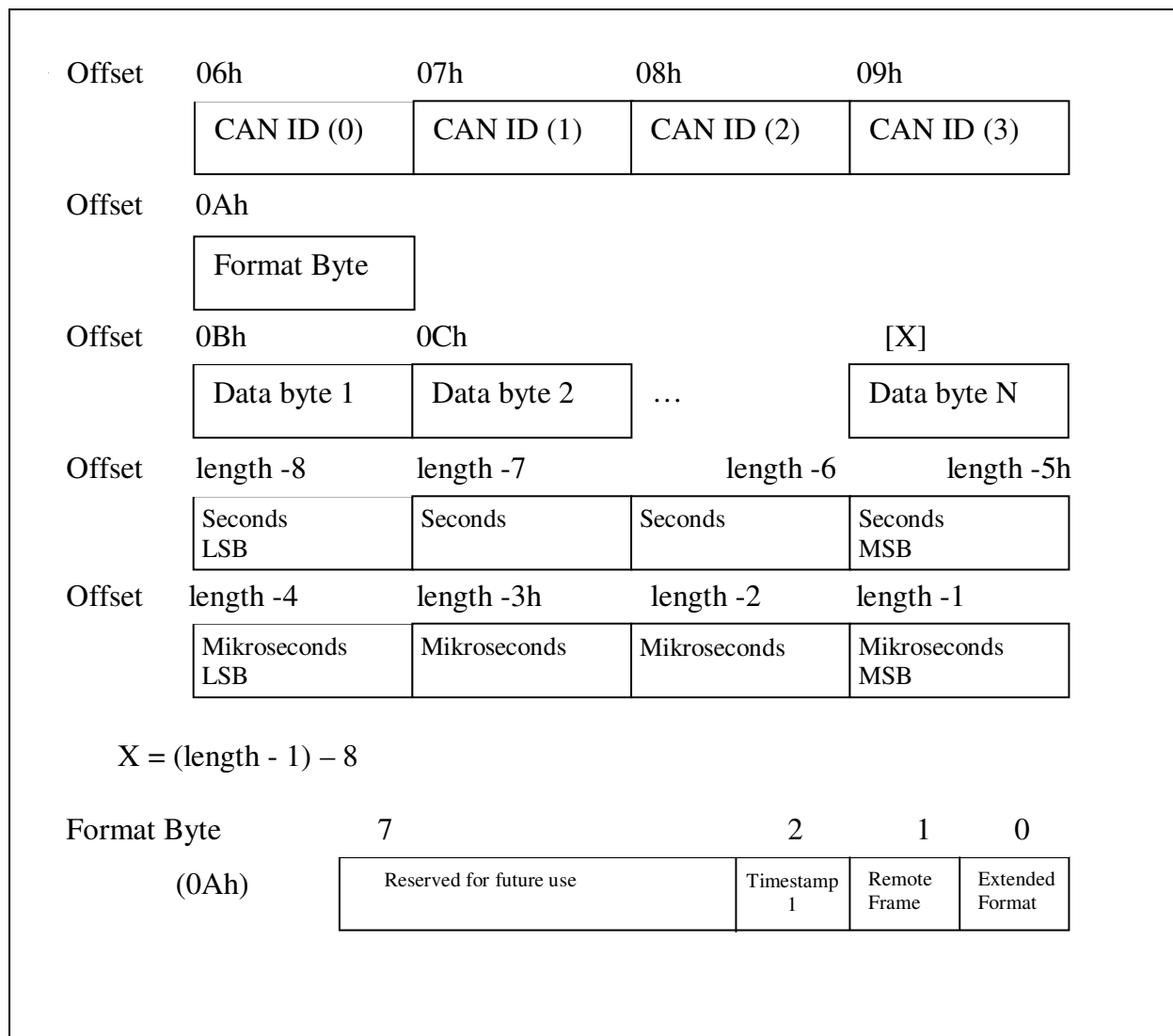


Chart 3-26: Telegram data for OP_ANAGATE_CAN_DATA_IND with activated timestamp

The following data is passed on:

- CAN-ID

The CAN ID is passed on as a 16-bit value in Little Endian format.

- Format byte

The format byte contains three bits which controls the message format.

- Timestamp (Bit 2)

Is this bit set, the additional 8 bytes are passed on in the message containing the system time at which the telegram is acknowledged by the CAN controller.

Is this bit unset, the 8 bytes for the system time are not passed on and the telegram structure is identical to the description in the OP_ANAGATE_CAN_DATA_REQ command.

- Remote Frame (Bit 1)
 - Is this bit set, the telegram is sent as remote frame without data.
 - Is this bit unset, the telegram is sent as normal CAN telegram including data bytes.
- Extended Format (Bit 0)
 - Is this bit set, the CAN ID is interpreted as extended (29 Bit) and sended accordingly. Bits 5..7 in byte CAN ID (3) are ignored.
 - Is this bit unset, the CAN ID is in standard format (11 Bit). Bits 3..7 in CAN ID (1) and CAN ID (2) und (3) are ignored.
- Data bytes
 - The data bytes of the CAN telegram.
- Timestamp (optional)
 - The time at which the data is acknowledged by the CAN controller. It is specified in seconds and microseconds since 01.01.1970 00:00 Uhr (Unix time).

3.3.8 OP_ANAGATE_CAN_DATA_RSP

The Response acknowledges the previously executed Data Indication command. An 8-bit return value is returned as useful data.

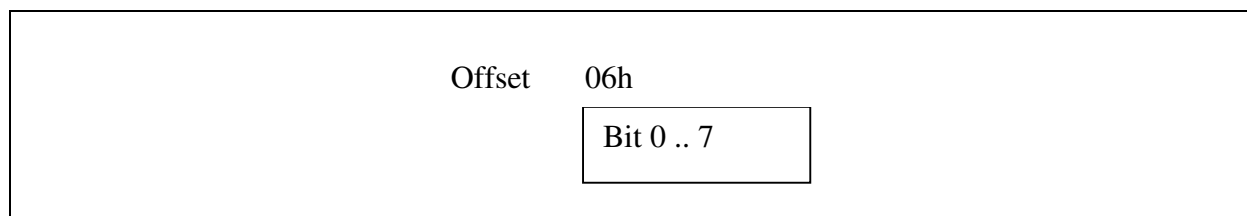


Chart 3-27: Telegram data for OP_ANAGATE_CAN_DATA_RSP

The following Return value may be given:

| Return value | Result |
|--------------|-------------------------------------|
| 00h | Telegram is processed successfully. |

Table 3-18: Return value for OP_ANAGATE_CAN_DATA_RSP

3.3.9 OP_ANAGATE_CAN_GET_INFO_REQ

For further information see 3.1.5 OP_ANAGATE_XX_GET_INFO_REQ.

3.3.10 OP_ANAGATE_CAN_GET_INFO_CNF

For further information see 3.1.6 OP_ANAGATE_XX_GET_INFO_CNF.

3.3.11 OP_ANAGATE_CAN_SET_CONFIG_REQ

The Set Config command configures the connection specific settings of the AnaGate CAN. The following data has to be passed on:

- DataConfirm
 - 0: The AnaGate device do not confirm Data Requeste. Likewise the master (PC) may not send Data Responses to Data Indications from the AnaGate device. This setting should be made for performance reasons only.
 - 1: Data Confirmations/Responses are exchanged.
- DataIndication
 - 0: The AnaGate device do generally not create Data Indications.
 - 1: The AnaGate device create Data Indications dependent on the active filter masks.

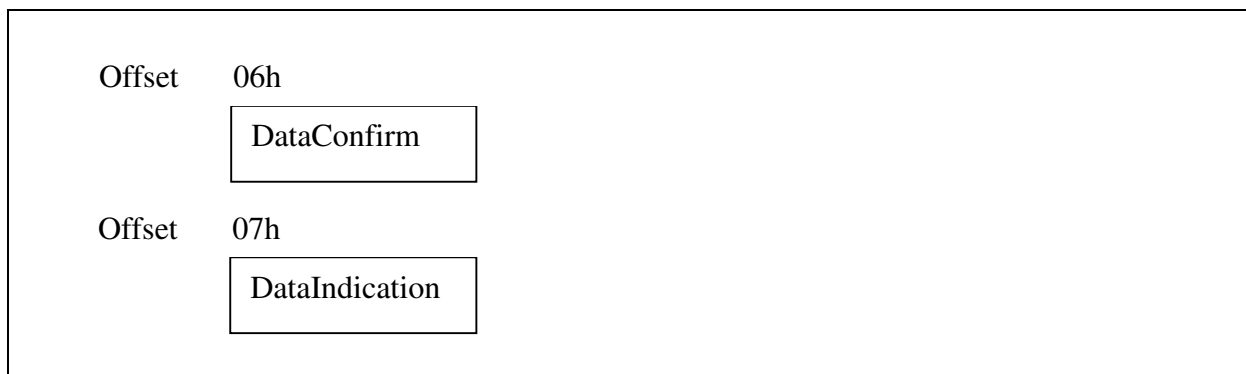


Chart 3-28: Telegram data for OP_ANAGATE_CAN_SET_CONFIG_REQ

3.3.12 OP_ANAGATE_CAN_SET_CONFIG_CNF

The Confirmation acknowledges the previously executed Set Config Request command. An 8-bit return value is returned as useful data.

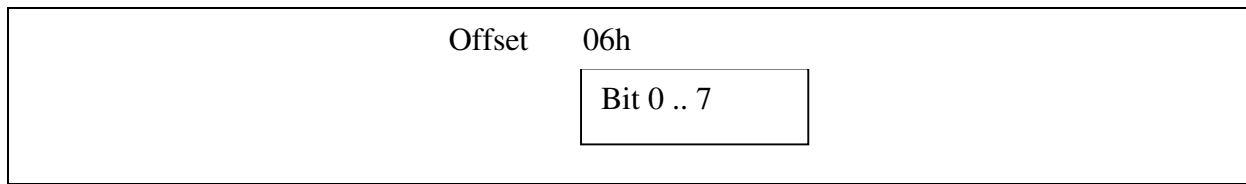


Chart 3-29: Telegram data for OP_ANAGATE_CAN_SET_CONFIG_CNF

The following Return value may be given:

| Return value | Result |
|--------------|--|
| 00h | Set Config command was successful. |
| FFh | Set Config command was not successful. |

Table 3-19: Return value for OP_ANAGATE_CAN_SET_CONFIG_CNF

3.3.13 OP_ANAGATE_CAN_GET_CONFIG_REQ

The Get Config command retrieves the connection specific AnaGate device settings. No further useful data is transmitted in this command.

3.3.14 OP_ANAGATE_CAN_GET_CONFIG_CNF

The Confirmation acknowledges the previously executed Get Config Request command. The following data is returned as useful data:

- Return code
The Return value given in Table 3-20 can be returned.
- DataConfirm
 - 0: The AnaGate device do not confirm Data Requeste. Likewise the master (PC) may not send Data Responses to Data Indications from the AnaGate device. This setting should be made for performance reasons only.
 - 1: Data Confirmations/Responses are exchanged.
- DataIndication
 - 0: The AnaGate device do generally not create Data Indications.
 - 1: The AnaGate device create Data Indications dependent on the active filter masks.

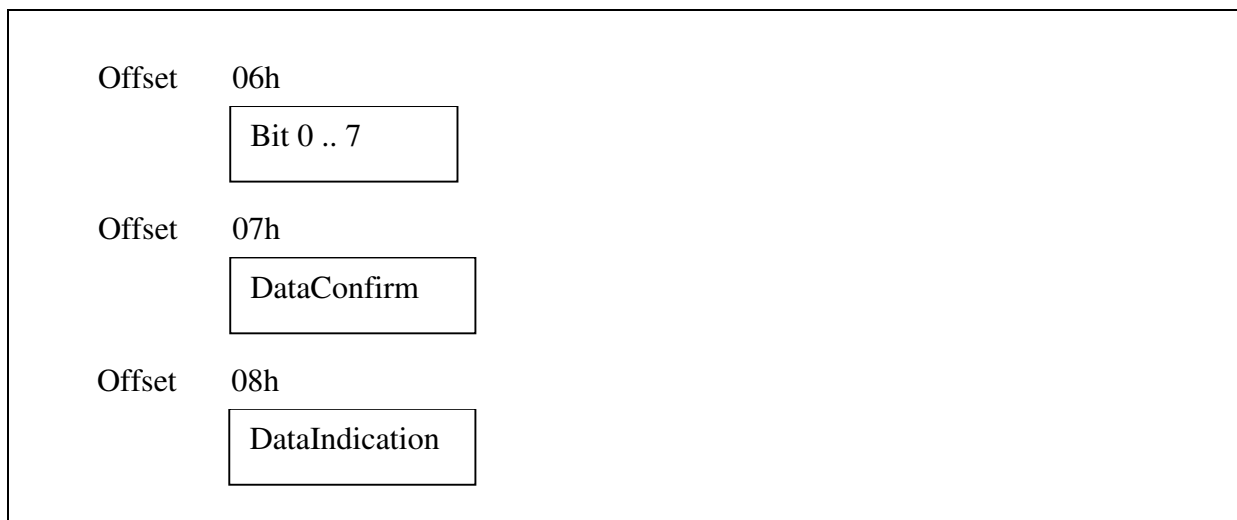


Chart 3-30: Telegram data for *OP_ANAGATE_CAN_GET_CONFIG_CNF*

The following Return value may be given:

| Return value | Result |
|--------------|------------------------------------|
| 00h | Get Config command was successful. |

Table 3-20: Return value for *OP_ANAGATE_CAN_GET_CONFIG_CNF*

3.3.15 OP_ANAGATE_CAN_SET_GLOBALS_REQ

The Set Globals command configures the global device settings of the AnaGate CAN. The following data has to be passed on:

- Operating mode

The operating mode is passed on as a 8-bit value Following operating modes are supported:

0 = default

1 = loop back All CAN telegrams sent to the AnaGate CAN via the TCP/IP protocol, are send back to all connected pc partners.

2 = listen The AnaGate CAN operates as an absolute passive CAN device. This means that the AnaGate CAN does not ACK incoming CAN telegrams, and it is not possible to send telegrams via the device.

- Baud rate

The baud rate is passed on as a 32-bit value in Little Endian Format.

Currently the following valid baud rates are allowed: 20000 for 20kBit, 50000 for 50kBit, 62500 for 62,5kBit, 100000 for 100kBit, 250000 for 250kBit, 500000 for 500kBit and 1000000 for 1MBit.

It is also possible, to set individual baud rates on the device. In this case the registers CNF1, CNF2 and CNF3 of the CAN controller MCP2515 have to be set directly. If bit 24 of the baudrate parameter is set, the lower 3 bytes of the parameter value are copied directly in the control registers in the following way:

| CAN bus baudrate | MCP2515 CNF1 | MCP2515 CNF2 | MCP2515 CNF3 | Baudrate parameter |
|---------------------|--------------|--------------|--------------|--------------------|
| | AA | BB | CC | 0x01 AA BB CC |
| For example: 1MHz | 0x41 | 0x89 | 0x00 | 0x01 41 89 00 |
| For example: 250KHz | 0x43 | 0xA1 | 0x03 | 0x01 43 A1 03 |

Note: Please refer to the manual of the MCP2515 for a description of the configuration registers CNF1, CNF2, CNF3. The oscillator frequency of the AnaGate CAN is $F_{OSC}=20\text{MHz}$, all other models use the frequency $F_{OSC}=24\text{MHz}$.

- Termination

The termination parameter is passed on as 8-bit value. Following values are allowed:

1: Integrated CAN bus termination is to be switched on.

0: Integrated CAN bus termination is to be switched off.

REMARK: Termination of the CAN bus is not supported by devices of hardware version 1.1.A. In this case value is ignored.

- Highspeed

The Highspeed parameter is passed on as 8-Bit value Following values are allowed:

1: Highspeed mode is to be switched on.

0: Highspeed mode is to be switched off.

The Highspeed mode was introduced to do not loose CAN telegrams with big data continuous loads and high baud rates. In this mode transmitted and received telegrams are not longer confirmed on protocol layer. Additionally all previously defined software filters (OP_ANAGATE_CAN_SET_FILTER_REQ) are deactivated to get a better system performance.

- Timestamp

The Timestamp parameter is passed on as 8-Bit value Following values are allowed:

1: Timestamp mode is to be switched on.

0: Timestamp mode is to be switched off.

If the timestamp mode is activated, data indications (OP_ANAGATE_CAN_DATA_IND) and data confirmations (OP_ANAGATE_CAN_DATA_CNF) are sent with an additional timestamp containing the system time when the telegram is acknowledged by the CAN controller.

Remark: The Timestamp imode s available first in Firmware version 1.3.0.

Remark: The parameters Termination, Highspeed and Timestamp are not supported by the devices with hardware version 1.1.A (AnaGate CAN). In this case the parameters are completely ignored.

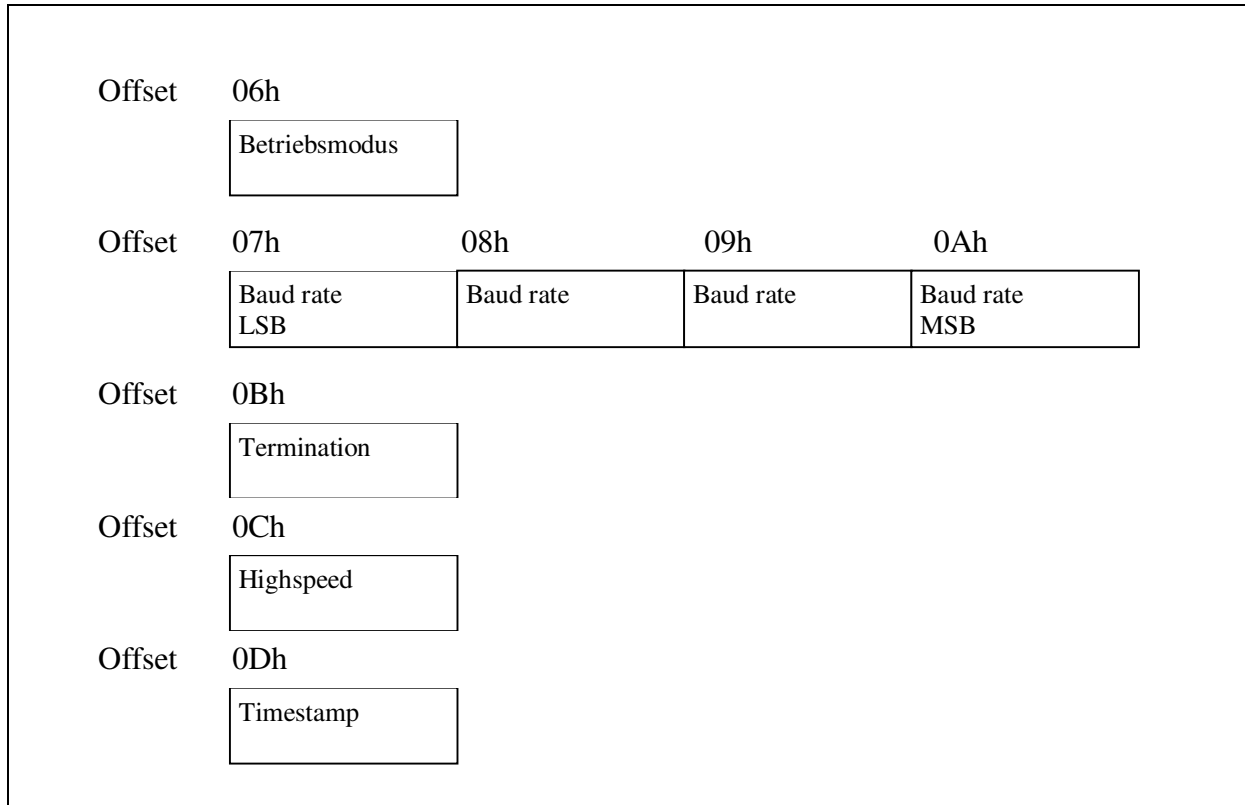


Chart 3-31: Telegram data for `OP_ANAGATE_CAN_SET_GLOBALS_REQ`

Attention!!

Baud rate, mode, termination, highspeed and timestamp are global settings for all TCP/IP connections.

3.3.16 `OP_ANAGATE_CAN_SET_GLOBALS_CNF`

The Confirmation acknowledges the previously executed Set Globals Request command. An 8-bit return value is returned as useful data.

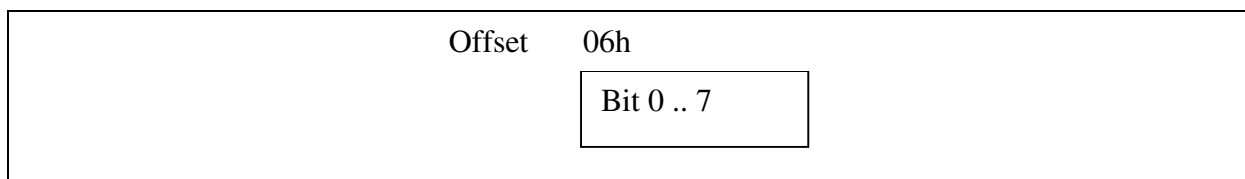


Chart 3-32: Telegram data for `OP_ANAGATE_CAN_SET_GLOBALS_CNF`

The following Return value may be given:

| Return value | Result |
|--------------|---|
| 00h | Set Globals command was successful. |
| FFh | Set Globals command was not successful. |

Table 3-21: Return value for *OP_ANAGATE_CAN_SET_GLOBALS_CNF*

3.3.17 OP_ANAGATE_CAN_GET_GLOBALS_REQ

The Get Globals command retrieves the global AnaGate device settings. No further useful data is transmitted in this command.

3.3.18 OP_ANAGATE_CAN_GET_GLOBALS_CNF

The Confirmation acknowledges the previously executed Get Globals Request command. The following data is returned as useful data:

- Return code
The Return value given in Table 3-22 can be returned.
- Operating mode
The operating modes is passed on as a 8-bit value (see 3.3.15 for details).
- Baud rate
The baud rate currently used on the CAN bus is returned. The baud rate is passed on as a 32-bit value in Little Endian Format.
- Termination
The termination mode is passed on as a 8-bit value. AnaGate devices of hardware version 1.1.A do not support this parameter. In this case a value of 0 is always returned.
- Highspeed
The Highspeed mode is passed on as a 8-bit value. AnaGate devices of hardware version 1.1.A do not support this parameter. In this case a value of 0 is always returned.
- Timestamp
The timestamp mode is passed on as a 8-bit value. AnaGate devices of hardware version 1.1.A do not support this parameter. In this case a value of 0 is always returned.

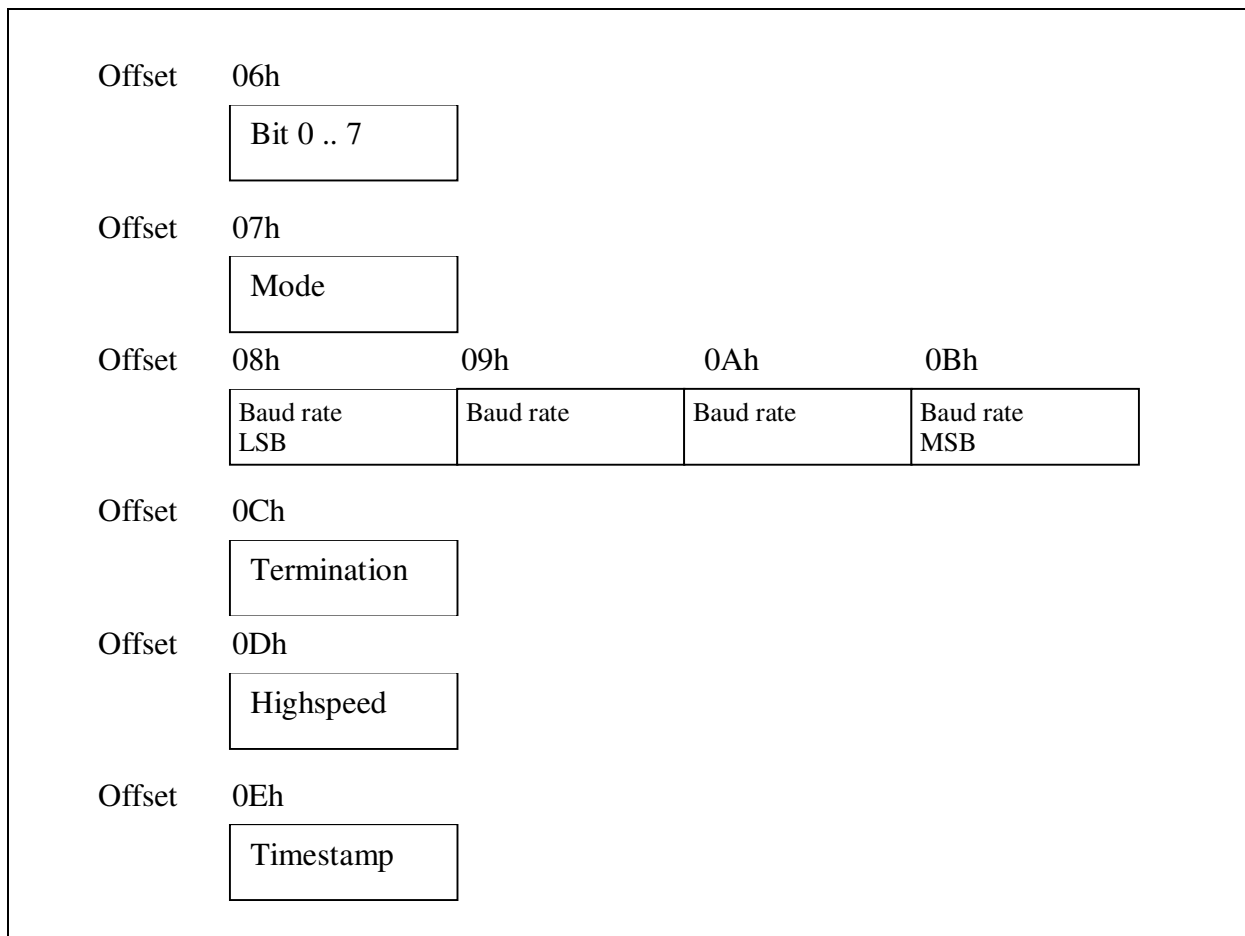


Chart 3-33: Telegram data for `OP_ANAGATE_CAN_GET_GLOBALS_CNF`

The following Return value may be given:

| Return value | Result |
|--------------|-------------------------------------|
| 00h | Get Globals command was successful. |

Table 3-22: Return value for `OP_ANAGATE_CAN_GET_GLOBALS_CNF`

3.3.19 `OP_ANAGATE_CAN_SET_FILTER_REQ`

The Set Filter Request defines connection specific messages filters. A maximum of 8 filters are available, which are linked with a logical AND operation. The following data has to be passed on:

- 4 mask filters

A mask filter contains of a mask value, which define the bits of the CAN identifier to examine, and the appropriate filter value.

If the CAN identifier matches in the indicated filter mask with the filter value, the incoming CAN telegram is sent to the partner, otherwise not.

- 4 range filters

A range filter defines an address range with a appropriate start and end address.

If the CAN identifier do not lie in the indicated filter range, the incoming CAN telegram is not sent to the partner.

| | | | | |
|--------|--------------------------------|----------------|----------------|-----------------------|
| Offset | 06h | 07h | 08h | 09h |
| | Filter 1 mask LSB | Filter 1 mask | Filter 1 mask | Filter 1 mask MSB |
| Offset | 0Ah | 0Bh | 0Ch | 0Dh |
| | Filter 1 value LSB | Filter 1 value | Filter 1 value | Filter 1 value MSB |
| | ... 3 additional mask filters | | | |
| Offset | 26h | 27h | 28h | 29h |
| | Filter 1 start LSB | Filter 1 start | Filter 1 start | Filter 1 start MSB |
| Offset | 2Ah | 2Bh | 2Ch | 2Dh |
| | Filter 1 end LSB | Filter 1 end | Filter 1 end | Filter 1 end MSB |
| | ... 3 additional range filters | | | |

Chart 3-34: Telegram data for OP_ANAGATE_CAN_SET_FILTER_CNF

The baud rate is passed on as a 32-bit value in Little Endian Format.

All filter masks, filter values and range values are passed on as a 32-bit values in Little Endian Format. Not used filter entries must be initialized with 0 values.

Remarks

All filters are implemented as software filters. Hardware filtering through the CAN controller of the AnaGate CAN device is not used, since only global filters for all TCP/IP connections are possible then.

3.3.20 OP_ANAGATE_CAN_SET_FILTER_CNF

The Confirmation acknowledges the previously executed Set Filter Request command. An 8-bit return value is returned as useful data.

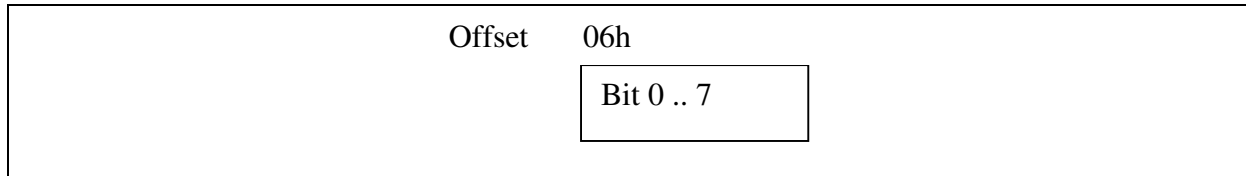


Chart 3-35: Telegram data for OP_ANAGATE_CAN_SET_FILTER_CNF

The following Return value may be given:

| Return value | Result |
|--------------|--|
| 00h | Set Filter command was successful. |
| FFh | Set Filter command was not successful. |

Table 3-23: Return value for OP_ANAGATE_CAN_SET_FILTER_CNF

3.3.21 OP_ANAGATE_CAN_GET_FILTER_REQ

The filters for the current TCPIP session are return by the Get Filter Request. No further useful data is transmitted in this command.

3.3.22 OP_ANAGATE_CAN_GET_FILTER_CNF

The Confirmation acknowledges the previously executed Get Filter Request command. The following data is returned as useful data:

- Return Code
The Return value given in Table 3-24 can be returned.
- 4 mask filters
See OP_ANAGATE_CAN_SET_FILTER_REQ for further details.
- 4 range filters
See OP_ANAGATE_CAN_SET_FILTER_REQ for further details.

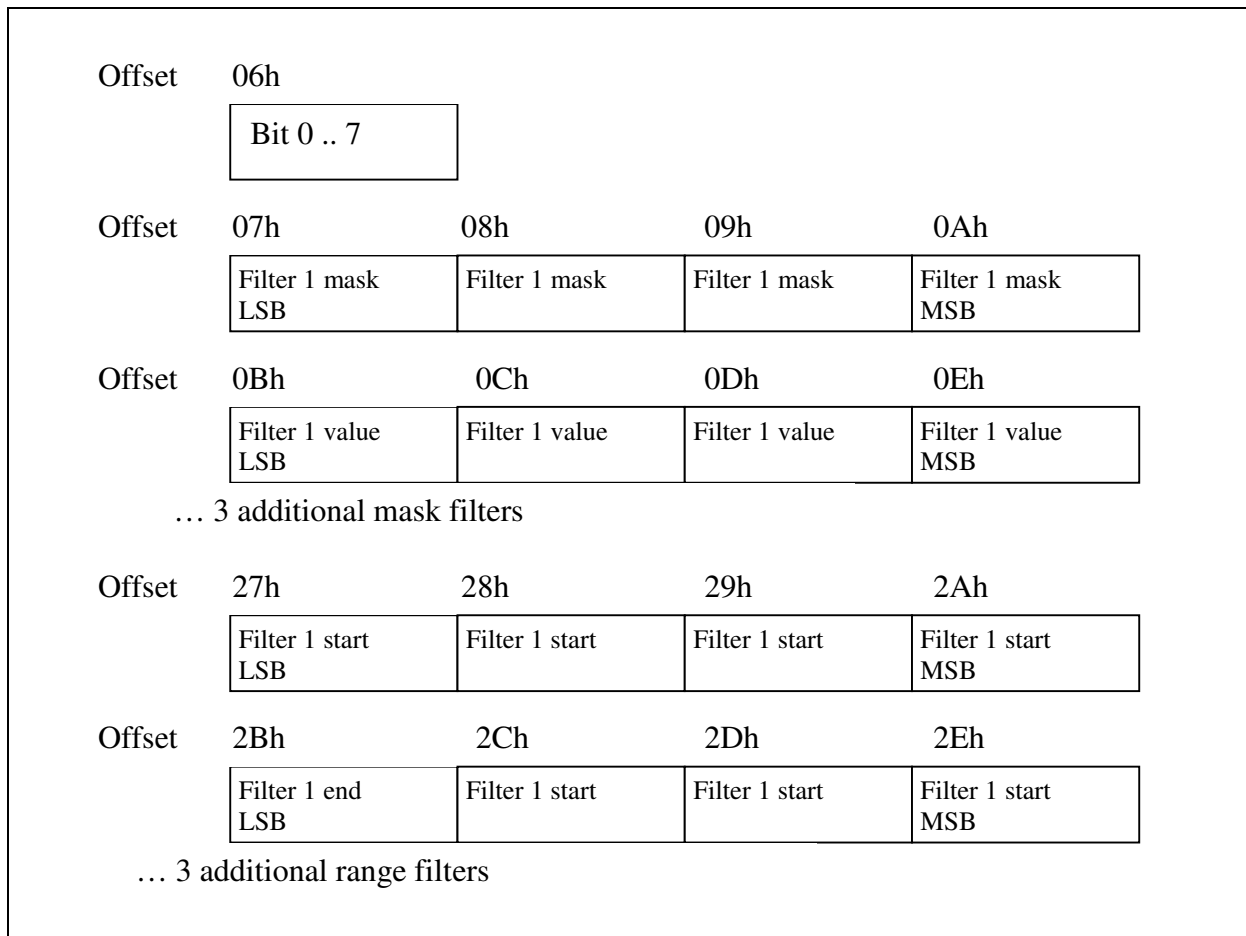


Chart 3-36: Telegram data for OP_ANAGATE_CAN_GET_FILTER_CNF

The following Return value may be given:

| Return value | Result |
|--------------|------------------------------------|
| 00h | Get Filter command was successful. |

Table 3-24: Return value for OP_ANAGATE_CAN_GET_FILTER_CNF

3.3.23 OP_ANAGATE_CAN_SET_TIME_REQ

The Set Time command changes the system time of the AnaGate device, which is used for the timestamp mode. This command has to be executed after each restart of the device. It is recommend to synchronize the device regularly with the Set Time command.

The system time is passed on as seconds and microseconds passed by since 01.01.1970 00:00 (unix time) in the following data format (struct timeval).

| | | | | |
|--------|---------------------|--------------|--------------|---------------------|
| Offset | 06h | 07h | 08h | 09h |
| | Seconds LSB | Seconds | Seconds | Seconds MSB |
| Offset | 0Ah | 0Bh | 0Ch | 0Dh |
| | Mikroseconds LSB | Mikroseconds | Mikroseconds | Mikroseconds MSB |

Chart 3-37: Telegram data for OP_ANAGATE_CAN_SET_TIME_REQ

3.3.24 OP_ANAGATE_CAN_SET_TIME_CNF

The Confirmation acknowledges the previously executed Set Time Request command. An 8-bit return value is returned as useful data.

| | |
|--------|------------|
| Offset | 06h |
| | Bit 0 .. 7 |

Chart 3-38: Telegram data for OP_ANAGATE_CAN_SET_TIME_CNF

The following Return value may be given:

| Return value | Result |
|--------------|-------------------------------------|
| 00h | Set Time command was successful. |
| FFh | Set Time command was not supported. |

Table 3-25: Return value for OP_ANAGATE_CAN_SET_TIME_CNF

3.3.25 OP_ANAGATE_CAN_RESTART_REQ

The Restart request terminates the running firmware on the AnaGate device und restart it. No further useful data is transmitted in this command.

The AnaGate confirms the request und restarts the firmware afterwards. This means that all open TCP-IP connections are immediately cancelled (including the current connection, which has requested the OP_ANAGATE_CAN_RESTART_REQ) and are available for new incoming connection requests.

This command is accepted by the AnaGate independent of former OP_ANAGATE_CAN_OPEN_REQ commands. This means that all open connections can be cancelled, even if the maximum of allowed connections is already reached.

Restarting the Firmware take about 1 second. During the restart the device does not accept incoming connection requests.

3.3.26 OP_ANAGATE_CAN_RESTART_CNF

The Confirmation acknowledges the previously executed Restart Request command. An 8-bit return value is returned as useful data.

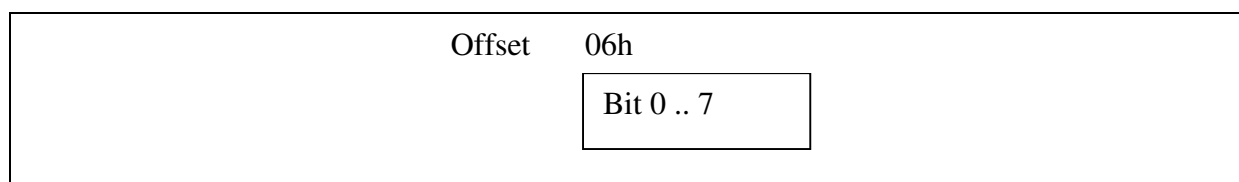


Chart 3-39: Telegram data for OP_ANAGATE_CAN_RESTART_CNF

The following Return value may be given:

| Return value | Result |
|--------------|-------------------------------------|
| 00h | Restart command was successful. |
| FFh | Restart command was not successful. |

Table 3-26: Return value for OP_ANAGATE_CAN_RESTART_CNF

3.3.27 OP_ANAGATE_CAN_DIO_WRITE_REQ

For further information see 3.1.7 OP_ANAGATE_XX_DIO_WRITE_REQ.

3.3.28 OP_ANAGATE_CAN_DIO_WRITE_CNF

For further information see 3.1.8 OP_ANAGATE_XX_DIO_WRITE_CNF.

3.3.29 OP_ANAGATE_CAN_DIO_READ_REQ

For further information see 3.1.9 OP_ANAGATE_XX_DIO_READ_REQ.

3.3.30 OP_ANAGATE_CAN_DIO_READ_CNF

For further information see 3.1.10 OP_ANAGATE_XX_DIO_READ_CNF.

3.3.31 „Z-CARD Process Image“ mode

The „Z-CARD Process Image“-Mode is a specific operating mode of the Anagate CAN duo, which is optimized for roller conveying systems using Z-CARD controller modules.

The Z-CARD unit sends a state telegram, called the CAN state word of the Z-CARD, for each internal state change via CAN bus. Through interpretation of this certain CAN telegrams a software application can create a process image of the whole conveyor. This process image can be used to build a parent controlling system, for example a tracking system.

This functionality can be shifted to the AnaGate with the „Z-CARD Process Image“ operating mode, so that the process image is cached on the AnaGate. The process image is sent in a single TCP telegram of type OP_ANAGATE_CAN_ZCARD_PI_DATA_IND at a configurable interval.

The Z-CARD units within a conveying system have a so called unique Z-CARD ID from 0 to 999. When activating the „Z-CARD Process Image“ mode, a contiguous range of Z-CARD IDs for the recorded process image has to be preset. As a result, the process image can be reduced to the actual number of Z-CARD units.

The state telegrams of the Z-CARD are then not longer forwarded with the OP_ANAGATE_CAN_DATA_IND command, but regardless from the selected Z-CARD ID range completely suppressed.

3.3.31.1 OP_ANAGATE_CAN_ZCARD_PI_DATA_IND

This command is only supported by AnaGate CAN duo devices.

IMPORTANT: This special functionality, the so called “Z-CARD Process Image” mode, must be explicitly activated with the OP_ANAGATE_CAN_SET_ZCARD_PI_REQ command. The mode is basically deactivated when the device is switched on.

The Z-CARD PI Data command sends the current process image of a for roller conveying system using Z-CARD controller modules (see also 3.3.31 „Z-CARD Process Image“).

The following data is passed on:

- CAN status word (4 bytes each Z-CARD)

For all selected Z-CARD units the CAN state words are returned in a sequential array. If, for example, the range from 10 to 20 is selected, 11 CAN state words are returned (4 * 11 = 44 bytes). The CAN state word includes in particular information about conditions of sensors, motors and PLC inputs and outputs.

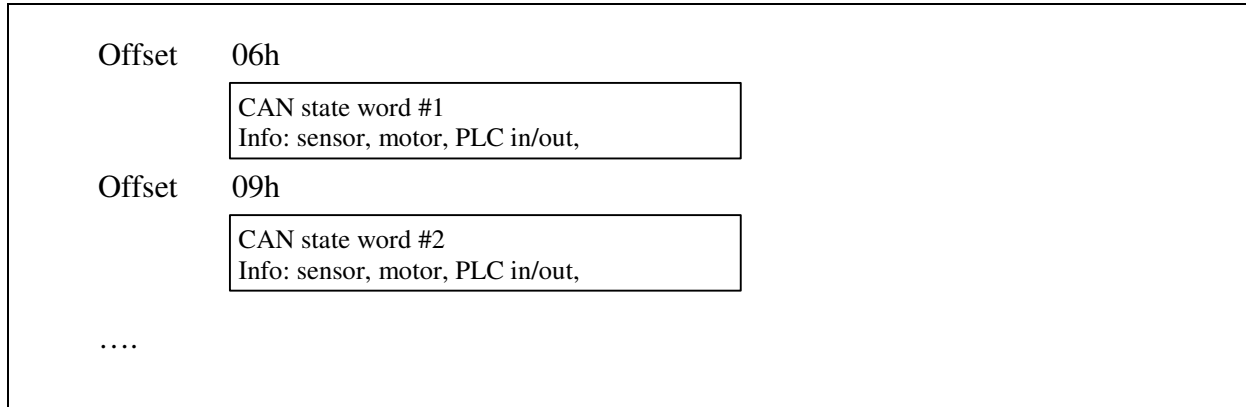


Chart 3-40: Telegram data for `OP_ANAGATE_CAN_ZCARD_PI_DATA_IND`

3.3.31.2 OP_ANAGATE_CAN_ZCARD_PI_DATA_RSP

The Confirmation acknowledges the previously executed Z-CARD PI Data Indication command. An 8-bit return value is returned as useful data.

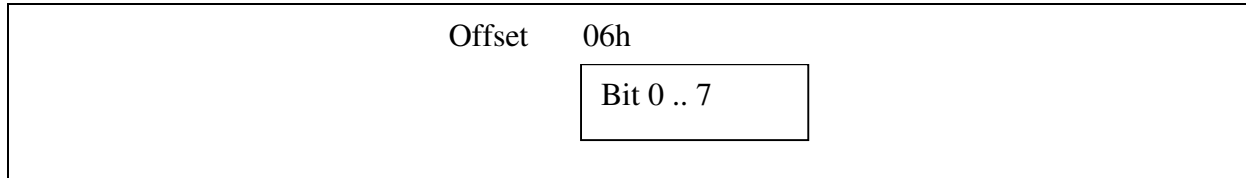


Chart 3-41: Telegram data for `OP_ANAGATE_CAN_ZCARD_PI_DATA_RSP`

The following Return value may be given:

| Return value | Result |
|--------------|---|
| 00h | Set Z-CARD PI Data command was received . |

Table 3-27: Return value for `OP_ANAGATE_CAN_ZCARD_PI_DATA_RSP`

3.3.31.3 OP_ANAGATE_CAN_SET_ZCARD_PI_REQ

This command is only supported by AnaGate CAN duo devices.

The Set Z-CARD PI command configures the settings of the "Z-CARD Process Image" mode of the AnaGate CAN duo. The following data has to be passed on:

- Z-CARD intervall

This parameter indicates a time interval in milliseconds from 30 to 255. The AnaGate sends the current Z-CARD process image regularly in the indicated time interval via the command `OP_ANAGATE_CAN_ZCARD_PI_DATA_IND`. By indication of "0" as a time interval, "Z-CARD process Image" mode on the AnaGate is deactivated. The values 1 to 29 are not allowed.

- Z-CARD ID minimum and Z-CARD ID maximum

The minimum and maximum values for Z-CARD ID defines the range of Z-CARD units, which states are to be cached in the process image. The range must be in the interval from 0 to 999.

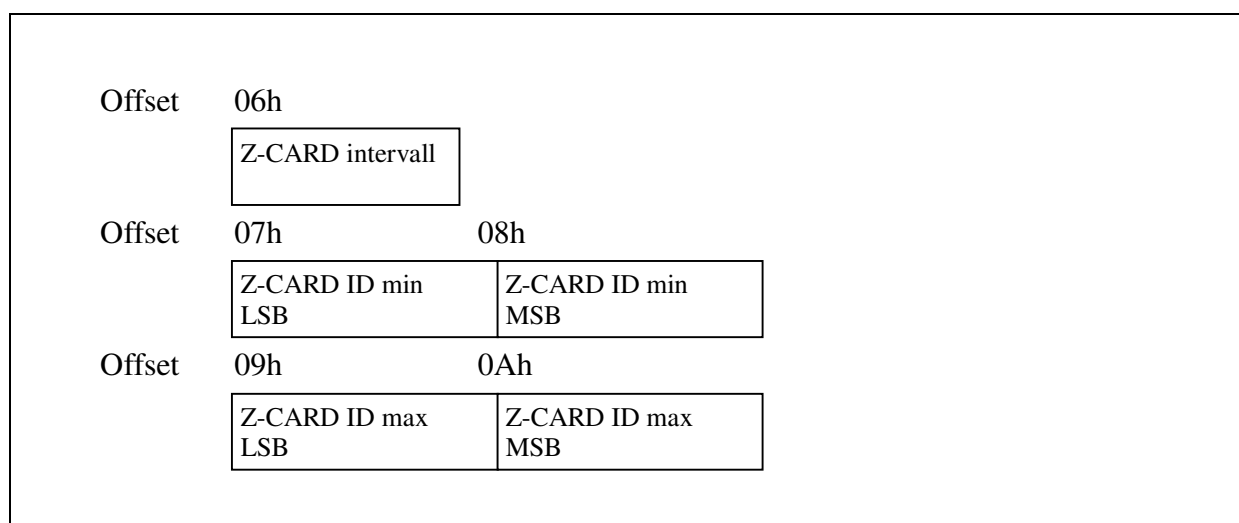


Chart 3-42: Telegram data for `OP_ANAGATE_CAN_SET_ZCARD_PI_DATA_REQ`

REMARK: The `OP_ANAGATE_CAN_SET_ZCARD_PI_REQ` command is not supported by AnaGate CAN devices.

3.3.31.4 `OP_ANAGATE_CAN_SET_ZCARD_PI_CNF`

The Confirmation acknowledges the previously executed Set Z-CARD PI Request command. An 8-bit return value is returned as useful data.

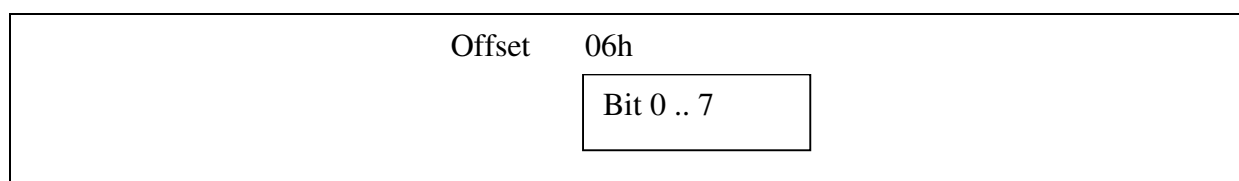


Chart 3-43: Telegram data for `OP_ANAGATE_CAN_SET_ZCARD_PI_CNF`

The following Return value may be given:

| Return value | Result |
|--------------|--------|
|--------------|--------|

| Return value | Result |
|--------------|---|
| 00h | Set Z-CARD PI command was successful. |
| FFh | Set Z-CARD PI command was not successful. |

Table 3-28: Return value for OP_ANAGATE_CAN_SET_ZCARD_PI_CNF

3.3.31.5 OP_ANAGATE_CAN_GET_ZCARD_PI_REQ

This command is only supported by AnaGate CAN duo devices.

The current settings of the "Z-CARD Process Image" mode are return by the Get Z-CARD PI Request. No further useful data is transmitted in this command.

3.3.31.6 OP_ANAGATE_CAN_GET_ZCARD_PI_CNF

The Confirmation acknowledges the previously executed Get Z-CARD PI Request command. The following data is returned as useful data:

- Return code
The Return value given in **Fehler! Verweisquelle konnte nicht gefunden werden.** can be returned.
- Z-CARD intervall
The time interval in milliseconds for sending the current Z-CARD of process image. The value "0" indicates that the "Z-CARD process Image" mode is not activated.
- Z-CARD ID minimum and maximum
The selected range of the Z-CARD units, which are to be cached in the process image.

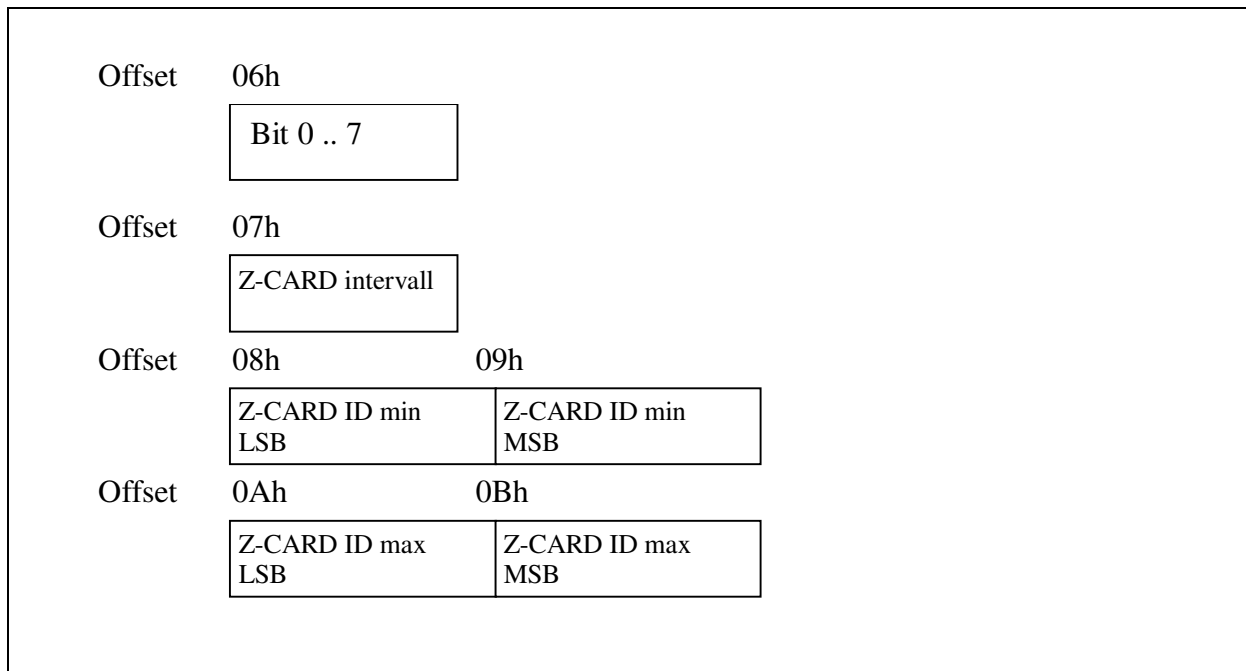


Chart 3-44: Telegram data for OP_ANAGATE_CAN_GET_ZCARD_PI_DATA_CNF

The following Return value may be given:

| Return value | Result |
|--------------|--|
| 00h | Get Z-CARD PI Data command was successful . |
| FFh | Get Z-CARD PI Data command was not supported . |

3.4 AnaGate SPI

In order to set up a connection to an AnaGate SPI device, it is necessary to open a TCP connection with TCP port 5003.

The first command sent to AnaGate SPI must be a `OP_ANAGATE_SPI_OPEN_REQ` request. The remaining requests (Read/Write/Close) can be performed once receipt has been acknowledged (`OP_ANAGATE_SPI_OPEN_CNF`).

An `OP_ANAGATE_SPI_CLOSE_REQ` request must be sent to close down the connection. The AnaGate SPI then returns the confirmation and independently closes the TCP connection.

An AnaGate SPI can accept up to 1 simultaneous TCP connection.

| Command-ID | Value |
|---|-------|
| <code>OP_ANAGATE_SPI_OPEN_REQ</code> | 0301h |
| <code>OP_ANAGATE_SPI_OPEN_CNF</code> | 8301h |
| <code>OP_ANAGATE_SPI_CLOSE_REQ</code> | 0302h |
| <code>OP_ANAGATE_SPI_CLOSE_CNF</code> | 8302h |
| <code>OP_ANAGATE_SPI_GET_INFO_REQ</code> | 0309h |
| <code>OP_ANAGATE_SPI_GET_INFO_CNF</code> | 8309h |
| <code>OP_ANAGATE_SPI_DIO_WRITE_REQ</code> | 0340h |
| <code>OP_ANAGATE_SPI_DIO_WRITE_CNF</code> | 8340h |
| <code>OP_ANAGATE_SPI_DIO_WRITE_CNF</code> | 0341h |
| <code>OP_ANAGATE_SPI_DIO_READ_CNF</code> | 8341h |
| <code>OP_ANAGATE_SPI_DATA_REQ</code> | 0730h |
| <code>OP_ANAGATE_SPI_DATA_CNF</code> | 8730h |
| <code>OP_ANAGATE_SPI_SEQUENCE_REQ</code> | 0730h |
| <code>OP_ANAGATE_SPI_SEQUENCE_CNF</code> | 8730h |
| <code>OP_ANAGATE_SPI_SET_GLOBALS_REQ</code> | 0730h |

| Command-ID | Value |
|--------------------------------|-------|
| OP_ANAGATE_SPI_SET_GLOBALS_CNF | 8730h |
| OP_ANAGATE_SPI_GET_GLOBALS_REQ | 0730h |
| OP_ANAGATE_SPI_GET_GLOBALS_CNF | 8730h |

Table 3-29: Command-IDs for AnaGate SPI

3.4.1 OP_ANAGATE_SPI_OPEN_REQ

For further information see 3.1.1 OP_ANAGATE_XX_OPEN_REQ.

3.4.2 OP_ANAGATE_SPI_OPEN_CNF

For further information see 3.1.2 OP_ANAGATE_XX_OPEN_CNF.

3.4.3 OP_ANAGATE_SPI_CLOSE_REQ

For further information see 3.1.3 OP_ANAGATE_XX_CLOSE_REQ.

3.4.4 OP_ANAGATE_SPI_CLOSE_CNF

For further information see 3.1.4 OP_ANAGATE_XX_CLOSE_CNF.

3.4.5 OP_ANAGATE_SPI_GET_INFO_REQ

For further information see 3.1.5 OP_ANAGATE_XX_GET_INFO_REQ.

3.4.6 OP_ANAGATE_SPI_GET_INFO_CNF

For further information see 3.1.6 OP_ANAGATE_XX_GET_INFO_CNF.

3.4.7 OP_ANAGATE_SPI_DIO_WRITE_REQ

For further information see 3.1.7 OP_ANAGATE_XX_DIO_WRITE_REQ.

3.4.8 OP_ANAGATE_SPI_DIO_WRITE_CNF

For further information see 3.1.8 OP_ANAGATE_XX_DIO_WRITE_CNF.

3.4.9 OP_ANAGATE_SPI_DIO_READ_REQ

For further information see 3.1.9 OP_ANAGATE_XX_DIO_READ_REQ.

3.4.10 OP_ANAGATE_SPI_DIO_READ_CNF

For further information see 3.1.10 OP_ANAGATE_XX_DIO_READ_CNF.

3.4.11 OP_ANAGATE_SPI_DATA_REQ

The Data Request command is used to send data to and receive data from the SPI bus.

Since the SPI Bus always works in duplex mode, for every byte send, a byte is received too. For every byte which is to read a 0x00 must be send to the partner.

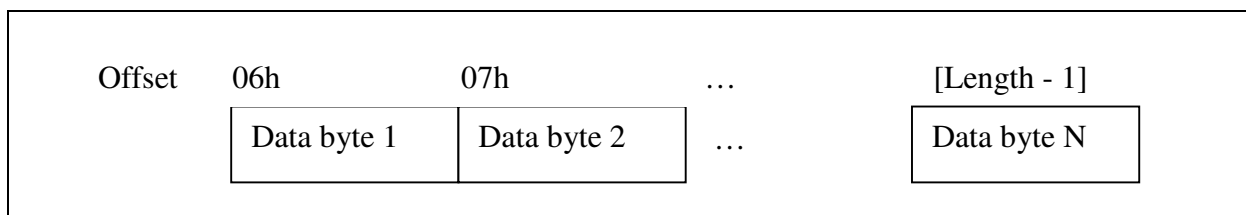


Chart 3-45: Telegram data for OP_ANAGATE_SPI_DATA_REQ

3.4.12 OP_ANAGATE_SPI_DATA_CNF

The Confirmation acknowledges the previously executed Data Request command. The following data is returned as useful data:

- Return Code
The Return value given in Table 3-30 can be returned.
- Data bytes

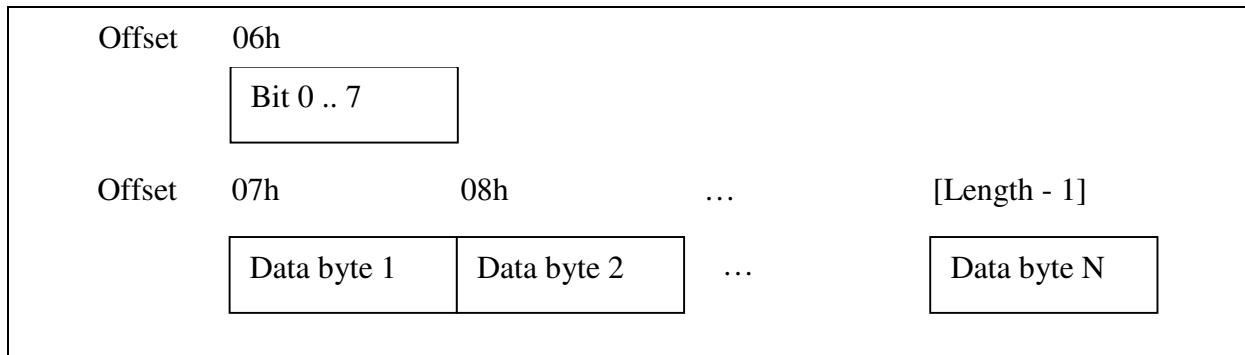


Chart 3-46: Telegram data for `OP_ANAGATE_SPI_DATA_CNF`

The following Return value may be given:

| Return value | Result |
|--------------|--------------------------------------|
| 00h | Data request command was successful. |
| 21h | An error during the send is raised. |

Table 3-30: Return value for `OP_ANAGATE_SPI_DATA_CNF`

3.4.13 OP_ANAGATE_SPI_SEQUENCE_REQ

The Sequence Request command is used to send a sequence of data requests to the SPI bus. The following data has to be passed on:

- Total number of bytes to read
- Command sequence

Several data request commands can be sent to the SPI partner. The commands are simply put into the message one behind the other.

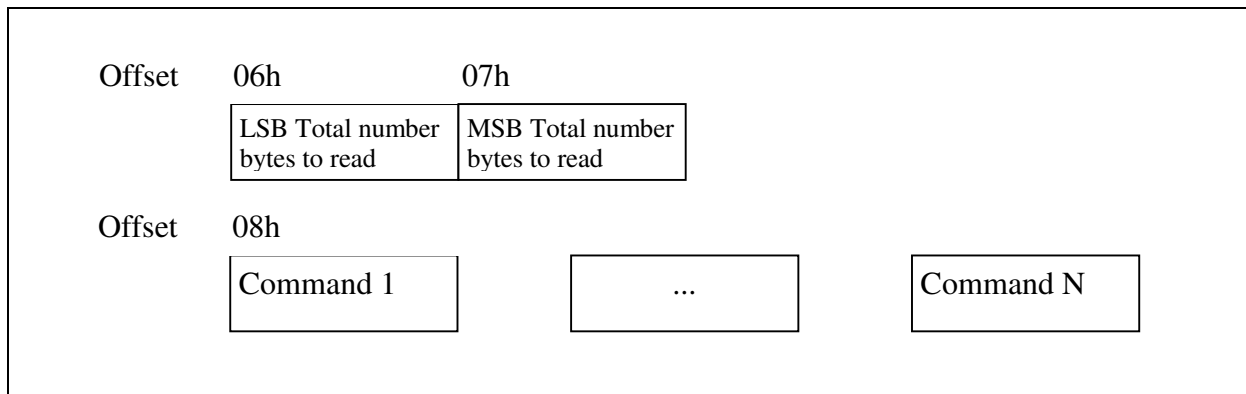


Chart 3-47: Telegram data for OP_ANAGATE_SPI_SEQUENCE_REQ

The data request command has following structure:

- Number of data bytes
Number of following bytes which are to send to the SPI Partner.
- Data bytes

While processing a data request command, the external partner is selected by the AnaGate SPI and de-selected afterwards again.

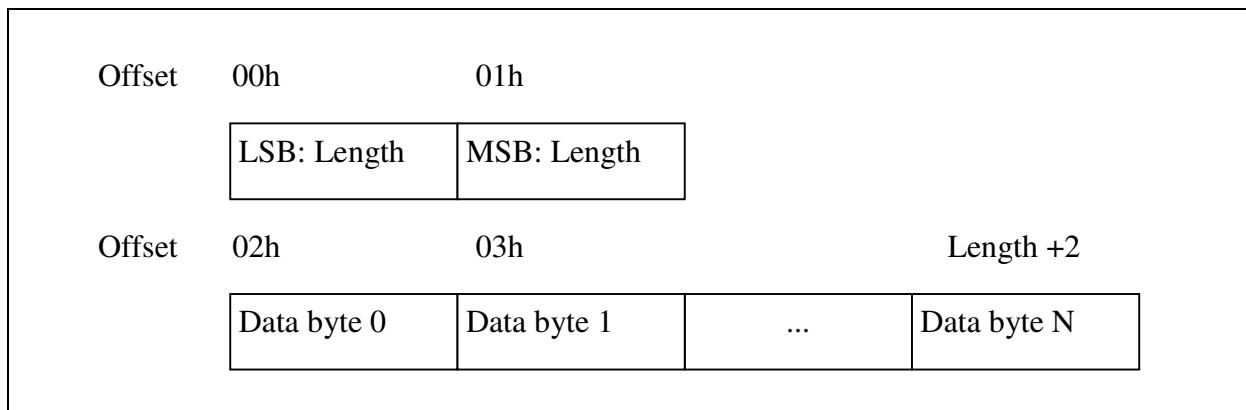


Chart 3-48: Data request command of the OP_ANAGATE_SPI_SEQUENCE_REQ

3.4.14 OP_ANAGATE_SPI_SEQUENCE_CNF

The Confirmation acknowledges the previously executed Sequence Request command. The following data is returned as useful data:

- Return code
The Return value given in Table 3-31 can be returned.
- Read data bytes

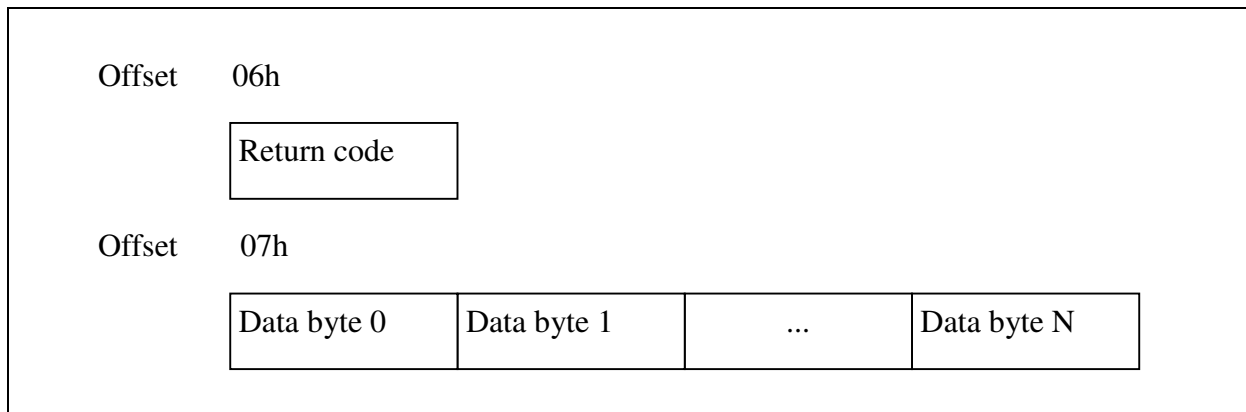


Chart 3-49: Telegram data for *OP_ANAGATE_SPI_SEQUENCE_CNF*

The following Return value may be given:

| Return value | Result |
|--------------|--------------------------------------|
| 00h | Sequence command was successful |
| 20h | The SPI Slave returned an NAK |
| 21h | The SPI Slave did not respond at all |

Table 3-31: Return value for *OP_ANAGATE_SPI_SEQUENCE_CNF*

3.4.15 **OP_ANAGATE_SPI_SET_GLOBALS_REQ**

The Set Globals command configures the global device settings of the AnaGate SPI. The following data has to be passed on:

- Baud rate

The baud rate is passed on as a 32-bit value in Little Endian Format.

Currently all values between 100000 for 100kBit and 6250000 for 6250 kBit are allowed.

- Voltage level

The voltage level for SPI signals is passed on as a 8-bit value Following operating modes are supported:

- 0 = Standard Output in High Impedance Mode
- 1 +5.0V level for signals
- 2 +3.3V level for signals
- 3 +2.5V level for signals

- Support level

The voltage level of the support voltage is passed on as a 8-bit value Following operating modes are supported:

- 0 Support voltage level is +3.3V
- 1 Support voltage level is +2.5V

- Clock mode

The clock mode (phase and polarity of the clock line for data transfer) is passed on as a 8-bit value. Following operating modes are supported:

- 0 CPHA = 0 and CPOL = 0
- 1 CPHA = 0 and CPOL = 1
- 2 CPHA = 1 and CPOL = 0
- 3 CPHA = 1 and CPOL = 1

| | | | | |
|--------|-----------------------------|-------------------------|---------------------------|---------------------------|
| Offset | 06h | 07h | 08h | 09h |
| | Bit 0 .. 7 Baud rate | Bit 8.. 15 Baud rate | Bit 16 .. 23 Baud rate | Bit 24 .. 31 Baud rate |
| Offset | 0Ah | | | |
| | Bit 0 .. 7 Voltage level | | | |
| Offset | 0Bh | | | |
| | Bit 0 .. 7 Support level | | | |
| Offset | 0Ch | | | |
| | Bit 0 .. 7 Clock mode | | | |

Chart 3-50: Telegram data for *OP_ANAGATE_SPI_SET_GLOBALS_REQ*

3.4.16 OP_ANAGATE_SPI_SET_GLOBALS_CNF

The Confirmation acknowledges the previously executed Set Globals Request command. An 8-bit return value is returned as useful data.

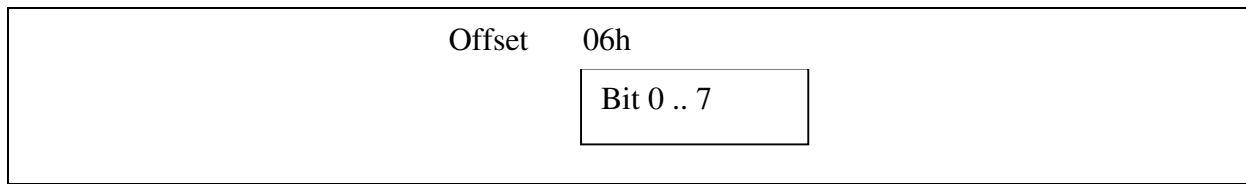


Chart 3-51: Telegram data for OP_ANAGATE_SPI_SET_GLOBALS_CNF

The following Return value may be given:

| Return value | Result |
|--------------|---|
| 00h | Set Globals command was successful. |
| FFh | Set Globals command was not successful. |

Table 3-32: Return value for OP_ANAGATE_SPI_SET_GLOBALS_CNF

3.4.17 OP_ANAGATE_SPI_GET_GLOBALS_REQ

The Get Globals command retrieves the global AnaGate device settings. No further useful data is transmitted in this command.

3.4.18 OP_ANAGATE_SPI_GET_GLOBALS_CNF

The Confirmation acknowledges the previously executed Get Globals Request command. The following data is returned as useful data:

- Return code
The Return value given in Table 3-33 can be returned.
- Baud rate
- Voltage level
- Support level
- Clock mode (phase and polarity of clock signal).

See 3.4.15 for details of the parameters.

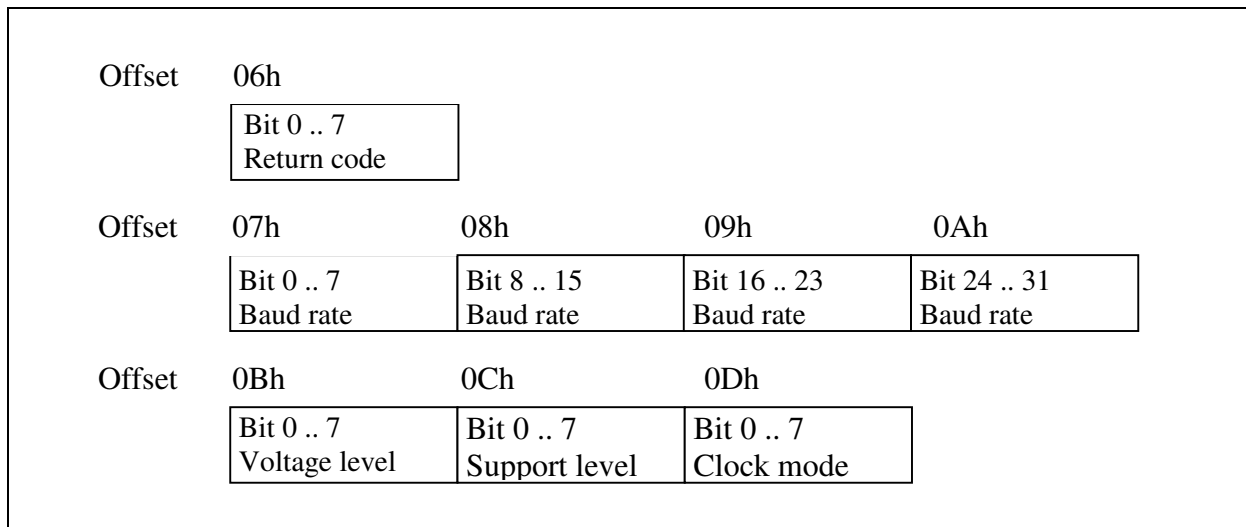


Chart 3-52: Telegram data for OP_ANAGATE_SPI_GET_GLOBALS_CNF

The following Return value may be given:

| Return value | Result |
|--------------|-------------------------------------|
| 00h | Get Globals command was successful. |

Table 3-33: Return value for OP_ANAGATE_SPI_GET_GLOBALS_CNF

3.5 AnaGate RS232

[Reserved for future use]

3.6 AnaGate DigitalIO

In order to set up a connection to an AnaGate DigitalIO device, it is necessary to open a TCP connection with TCP port 5003.

The first command sent to AnaGate DigitalIO must be a `OP_ANAGATE_DIO_OPEN_REQ` request. The remaining requests (Read/Write/ Reset/Close) can be performed once receipt has been acknowledged (`OP_ANAGATE_DIO_OPEN_CNF`).

An `OP_ANAGATE_DIO_CLOSE_REQ` request must be sent to close down the connection. The AnaGate DigitalIO then returns the confirmation and independently closes the TCP connection.

An AnaGate DigitalIO can accept up to 4 simultaneous TCP connections.

| Command-ID | Value |
|--|-------|
| <code>OP_ANAGATE_DIO_OPEN_REQ</code> | 0401h |
| <code>OP_ANAGATE_DIO_OPEN_CNF</code> | 8401h |
| <code>OP_ANAGATE_DIO_CLOSE_REQ</code> | 0402h |
| <code>OP_ANAGATE_DIO_CLOSE_CNF</code> | 8402h |
| <code>OP_ANAGATE_DIO_GET_INFO_REQ</code> | 0409h |
| <code>OP_ANAGATE_DIO_GET_INFO_CNF</code> | 8409h |
| <code>OP_ANAGATE_DIO_WRITE_REQ</code> | 0440h |
| <code>OP_ANAGATE_DIO_WRITE_CNF</code> | 8440h |
| <code>OP_ANAGATE_DIO_READ_REQ</code> | 0441h |
| <code>OP_ANAGATE_DIO_READ_CNF</code> | 8441h |

Table 3-34: Command-IDs for AnaGate DigitalIO

3.6.1 `OP_ANAGATE_DIO_OPEN_REQ`

For further information see 3.1.1 `OP_ANAGATE_XX_OPEN_REQ`.

3.6.2 OP_ANAGATE_DIO_OPEN_CNF

For further information see 3.1.2 OP_ANAGATE_XX_OPEN_CNF.

3.6.3 OP_ANAGATE_DIO_CLOSE_REQ

For further information see 3.1.3 OP_ANAGATE_XX_CLOSE_REQ.

3.6.4 OP_ANAGATE_DIO_CLOSE_CNF

For further information see 3.1.4 OP_ANAGATE_XX_CLOSE_CNF.

3.6.5 OP_ANAGATE_DIO_GET_INFO_REQ

For further information see 3.1.5 OP_ANAGATE_XX_GET_INFO_REQ.

3.6.6 OP_ANAGATE_DIO_GET_INFO_CNF

For further information see 3.1.6 OP_ANAGATE_XX_GET_INFO_CNF.

3.6.7 OP_ANAGATE_DIO_WRITE_REQ

For further information see 3.1.7 OP_ANAGATE_XX_DIO_WRITE_REQ.

3.6.8 OP_ANAGATE_DIO_WRITE_CNF

For further information see 3.1.8 OP_ANAGATE_XX_DIO_WRITE_CNF.

3.6.9 OP_ANAGATE_DIO_READ_REQ

For further information see 3.1.9 OP_ANAGATE_XX_DIO_READ_REQ.

3.6.10 OP_ANAGATE_DIO_READ_CNF

For further information see 3.1.10 OP_ANAGATE_XX_DIO_READ_CNF.

3.7 AnaGate Audio

[Reserved for future use]

3.8 AnaGate A/D

In order to set up a connection to an AnaGate A/D device, it is necessary to open a TCP connection with TCP port 5006.

The first command sent to AnaGate A/D must be a OP_ANAGATE_AD_OPEN_REQ request. The remaining requests (Read/Write/Close) can be performed once receipt has been acknowledged (OP_ANAGATE_AD_OPEN_CNF).

An OP_ANAGATE_AD_CLOSE_REQ request must be sent to close down the connection. The AnaGate A/D then returns the confirmation and independently closes the TCP connection.

An AnaGate A/D can accept up to 1 simultaneous TCP connection.

| Command-ID | Value |
|-----------------------------|-------|
| OP_ANAGATE_AD_OPEN_REQ | 0701h |
| OP_ANAGATE_AD_OPEN_CNF | 8701h |
| OP_ANAGATE_AD_CLOSE_REQ | 0702h |
| OP_ANAGATE_AD_CLOSE_CNF | 8702h |
| OP_ANAGATE_AD_GET_INFO_REQ | 0709h |
| OP_ANAGATE_AD_GET_INFO_CNF | 8709h |
| OP_ANAGATE_AD_DIO_WRITE_REQ | 0740h |
| OP_ANAGATE_AD_DIO_WRITE_CNF | 8740h |
| OP_ANAGATE_AD_DIO_READ_REQ | 0741h |
| OP_ANAGATE_AD_DIO_READ_CNF | 8741h |
| OP_ANAGATE_AD_READ_REQ | 0730h |
| OP_ANAGATE_AD_READ_CNF | 8730h |

Table 3-35: Command-IDs for AnaGate A/D

3.8.1 OP_ANAGATE_AD_OPEN_REQ

For further information see 3.1.1 OP_ANAGATE_XX_OPEN_REQ.

3.8.2 OP_ANAGATE_AD_OPEN_CNF

For further information see 3.1.2 OP_ANAGATE_XX_OPEN_CNF.

3.8.3 OP_ANAGATE_AD_CLOSE_REQ

For further information see 3.1.3 OP_ANAGATE_XX_CLOSE_REQ.

3.8.4 OP_ANAGATE_AD_CLOSE_CNF

For further information see 3.1.4 OP_ANAGATE_XX_CLOSE_CNF.

3.8.5 OP_ANAGATE_AD_GET_INFO_REQ

For further information see 3.1.5 OP_ANAGATE_XX_GET_INFO_REQ.

3.8.6 OP_ANAGATE_AD_GET_INFO_CNF

For further information see 3.1.6 OP_ANAGATE_XX_GET_INFO_CNF.

3.8.7 OP_ANAGATE_AD_DIO_WRITE_REQ

For further information see 3.1.7 OP_ANAGATE_XX_DIO_WRITE_REQ.

3.8.8 OP_ANAGATE_AD_DIO_WRITE_CNF

For further information see 3.1.8 OP_ANAGATE_XX_DIO_WRITE_CNF.

3.8.9 OP_ANAGATE_AD_DIO_READ_REQ

For further information see 3.1.9 OP_ANAGATE_XX_DIO_READ_REQ.

3.8.10 OP_ANAGATE_AD_DIO_READ_CNF

For further information see 3.1.10 OP_ANAGATE_XX_DIO_READ_CNF.

The return value of the Digital-In-Register differs from the value from the standard **Fehler! Verweisquelle konnte nicht gefunden werden.** command.

The return value of the parameter *Digital-In Register* is defined as follows:

- Bit 0 to 3 Bit 0 to 3 correspond to the actual values of the digital inputs
- Bit 8 to 11 are set to 1, if on the corresponding digital input a positive clock edge has occurred since last read command
- Bit 17 to 20 are set to 1, if on the corresponding digital input a negative clock edge has occurred since last read command
- Bit 24 to 31 are reserved and set to 0

The internal memory, which holds the clock edge changes, are reset to zero after every read command.

3.8.11 OP_ANAGATE_AD_READ_REQ

The Read Request reads the current values of the four analog inputs of the AnaGate A/D device. No further useful data is transmitted in this command.

3.8.12 OP_ANAGATE_AD_READ_CNF

The Confirmation acknowledges the previously executed Read Digital command. The following data is returned as useful data:

- Return Code

The Return value given in Table 3-36 can be returned.

- 4 Analog Registers

The current register values of the analog inputs A1 to A4 are passed on as a 32-bit value in Little Endian format.

Due to the technical structure a linear transformation of the values of the input interval from 0 to 20,48mA takes place to 0 to 65536.

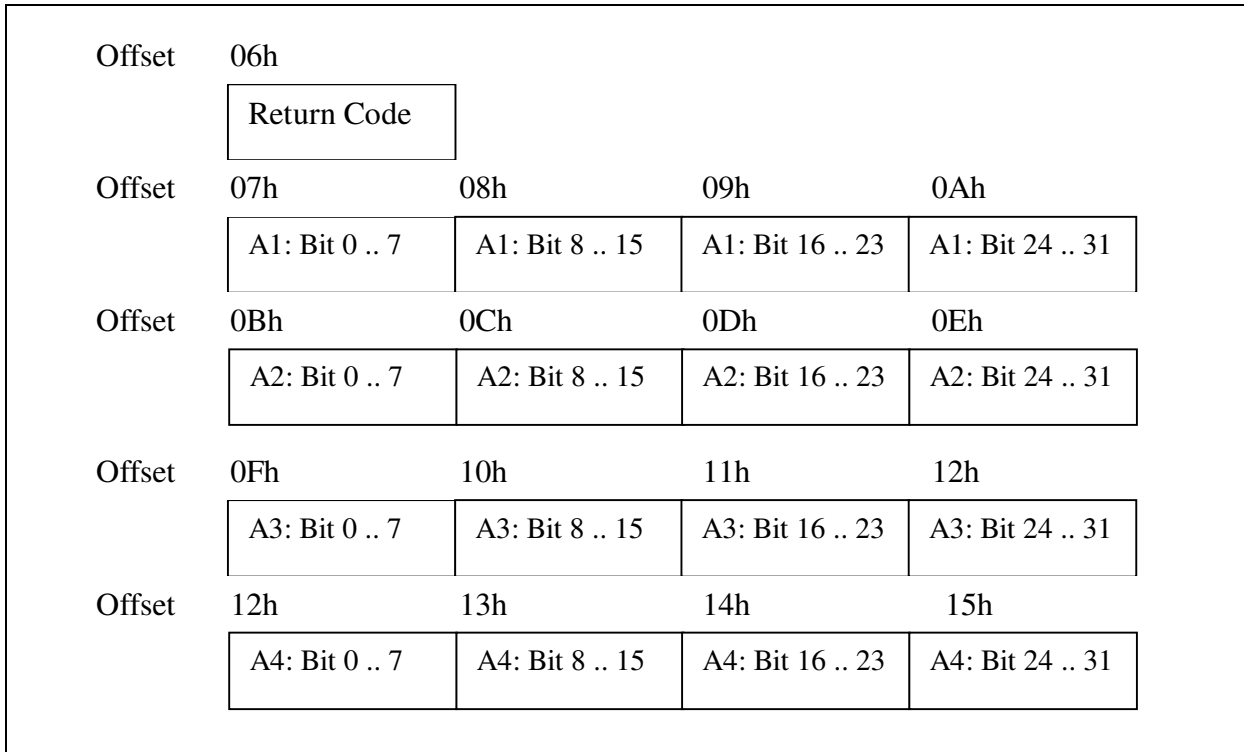


Chart 3-53: Telegram data for OP_ANAGATE_AD_READ_CNF

The following Return value may be given:

| Return value | Result |
|--------------|----------------------------------|
| 00h | Read command was successful. |
| FFh | Read command was not successful. |

Table 3-36: Return value for OP_ANAGATE_AD_READ_CNF

3.9 AnaGate Phone

[Reserved for future use]

Literature

- [1] I2C Bus <http://www.standardproducts.philips.com/products/collateral/i2c/pdf/spec-i2cbus21.pdf>
- [2] CAN Bus <http://www.can.bosch.com/>

Abbreviations

| | |
|----------------------------|---|
| CAN | <u>C</u> ontroller <u>A</u> rea <u>N</u> etwork |
| I2C / I ² C Bus | <u>I</u> nter <u>I</u> C Bus |
| SPI | <u>S</u> erial <u>P</u> eripheral <u>I</u> nterface |
| LSB | <u>L</u> east <u>S</u> ignificant <u>B</u> yte |
| MSB | <u>M</u> ost <u>S</u> ignificant <u>B</u> yte |