MANUAL

ANAGATE

TCP/IP

COMMUNICATION

ANALYTICA GmbH

Vorholzstrasse 36 Tel. +49 721 35043-0 E-mail: info@analytica-gmbh.de

D-76137 Karlsruhe Fax: +49 721 35043-20 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

Contents

Charts

Tables

1 Introduction

This manual describes the basic communication between an ***AnaGate*** device and any other system[1] (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 (I^2C, 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 (up to 8) at any one time.

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 RS232	5002
AnaGate DigitalIO	5003
AnaGate Audio	5004
AnaGate Phone	5005

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

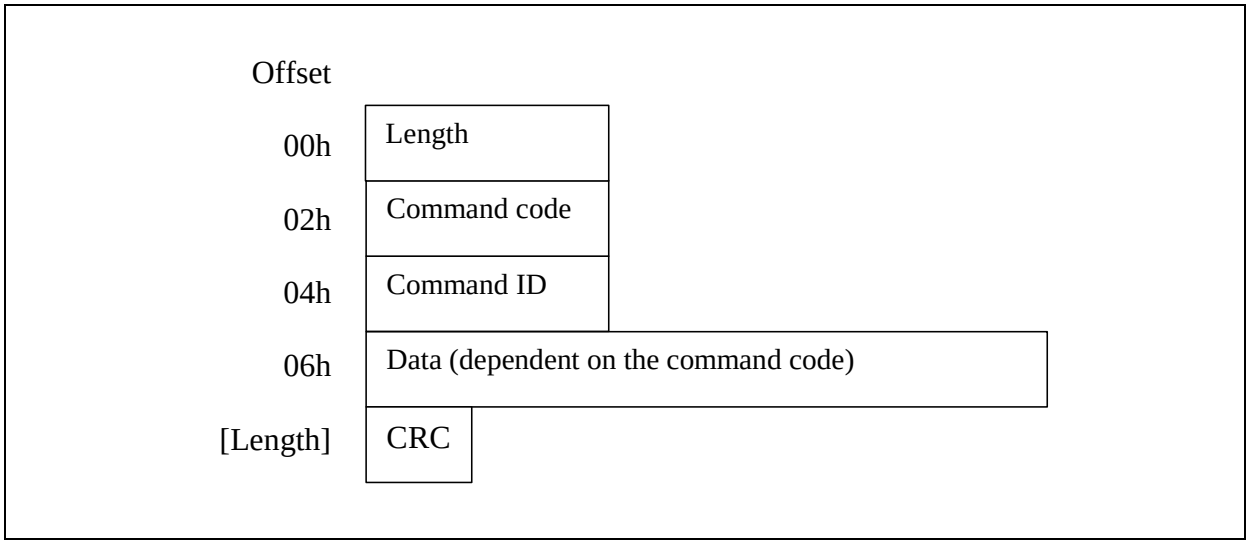

Chart 2-2: General structure of a telegram

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.

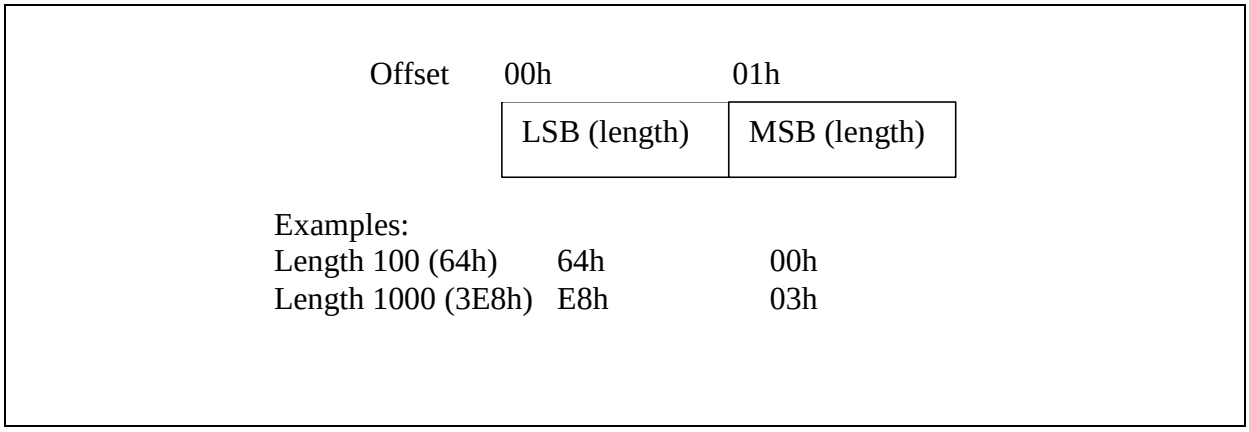

Chart 2-3: Length

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 (2)
  - ***AnaGate*** RS232 (3)
  - ***AnaGate*** DigitalIO (4)
  - ***AnaGate*** Audio (5)
  - ***AnaGate*** Phone (6)

- Bit 0 – 7

  Unambiguously identifies a specific command.

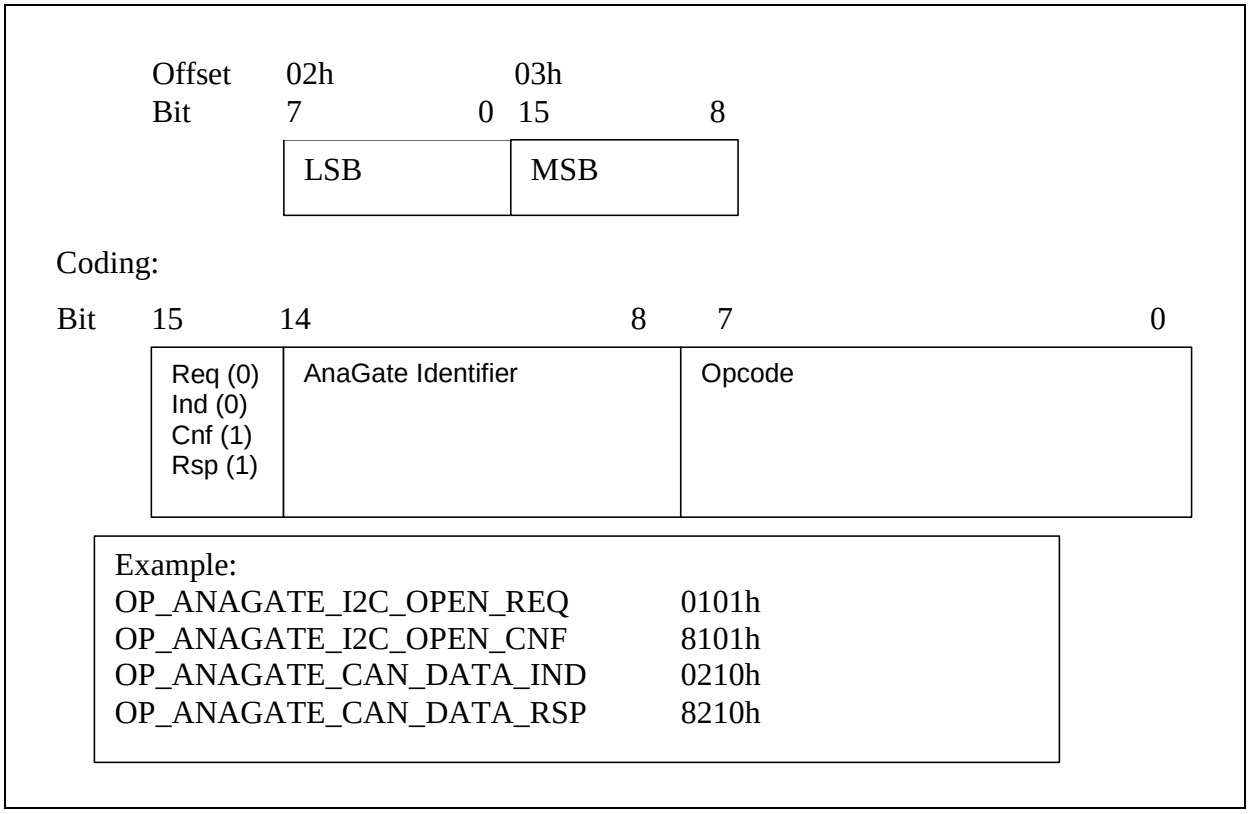

Chart 2-4: Command code

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:

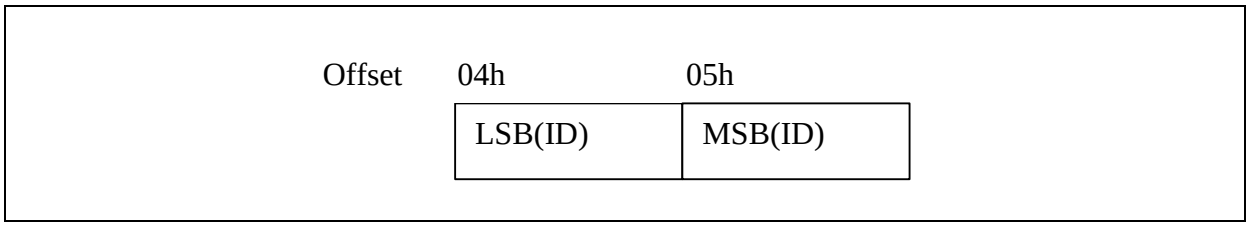

Chart 2-5: Command ID

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	10h	42h	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 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	0104h
OP_ANAGATE_I2C_CLOSE_CNF	8104h
OP_ANAGATE_I2C_CLOSE_CNF	0105h
OP_ANAGATE_I2C_RESET_CNF	8105h
OP_ANAGATE_I2C_WRITE_REQ	0103h
OP_ANAGATE_I2C_WRITE_CNF	8103h
OP_ANAGATE_I2C_READ_REQ	0102h
OP_ANAGATE_I2C_READ_CNF	8102h
OP_ANAGATE_I2C_EEPROM_WRITE_REQ	0106h
OP_ANAGATE_I2C_EEPROM_WRITE_CNF	8106h

Command ID	Value
OP_ANAGATE_I2C_EEPROM_READ_REQ	0107h
OP_ANAGATE_I2C_EEPROM_READ_CNF	8107h
OP_ANAGATE_I2C_STATUS_REQ	0108h
OP_ANAGATE_I2C_STATUS_CNF	8108h

Table 3-1: Command IDs for AnaGate I2C devices

3.1.1 OP_ANAGATE_I2C_OPEN_REQ

The Open command initiates the setting up of a logical connection to an AnaGate I2C device. The baud rate to be used on the I2C bus is transmitted as useful data.

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

Offset	06h	07h	08h	09h
	Bit 0 .. 7	Bit 8 .. 15	Bit 16 .. 23	Bit 24 .. 31

Example: 400 kBit → Value 400000 (00061A80h)

80h	1Ah	06h	00h

Chart 3-1: I2C baud rate

The currently valid baud rates are 100 kBit and 400 kBit.

3.1.2 OP_ANAGATE_I2C_OPEN_CNF

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

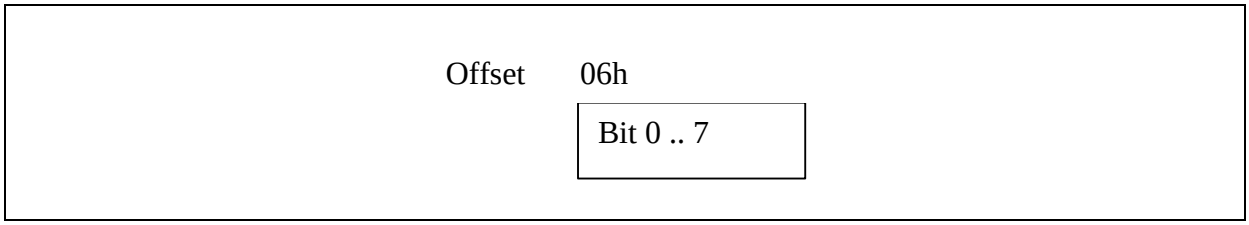

Chart 3-2: Return value for OP_ANAGATE_I2C_OPEN_CNF

The following return values may be given:

Return value	Result
00h	Open command was successful

Table 3-2: Return value for OP_ANAGATE_I2C_OPEN_CNF

3.1.3 OP_ANAGATE_I2C_CLOSE_REQ

The Close command terminates the logical connection to an AnaGate I2C 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_I2C_CLOSE_CNF

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

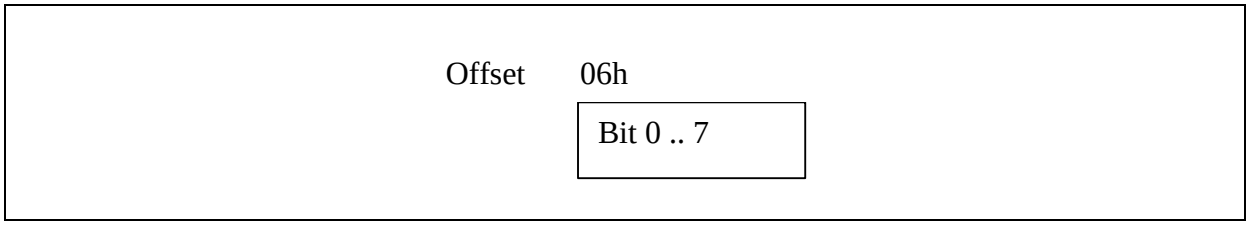

Chart 3-3: Return value for OP_ANAGATE_I2C_CLOSE_CNF

The following return values may be given:

Return value	Result
00h	Close command was successful

Table 3-3: Return value for OP_ANAGATE_I2C_CLOSE_CNF

3.1.5 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.

3.1.6 OP_ANAGATE_I2C_RESET_CNF

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

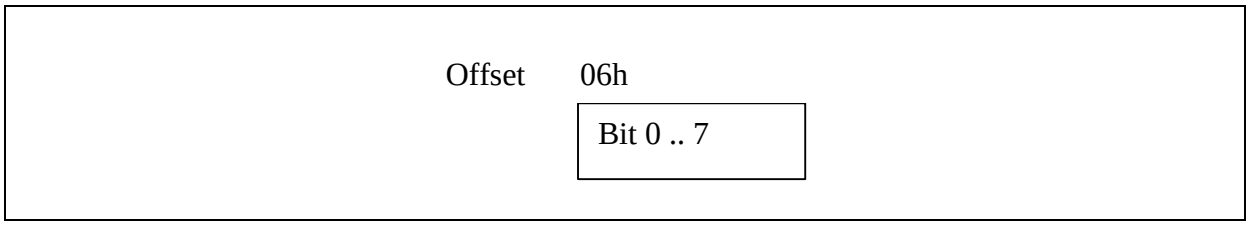

Chart 3-4: Return value for OP_ANAGATE_I2C_RESET_CNF

The following return values may be given:

Return value	Result
00h	Reset command was successful

Table 3-4: Return value for OP_ANAGATE_I2C_RESET_CNF

3.1.7 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.

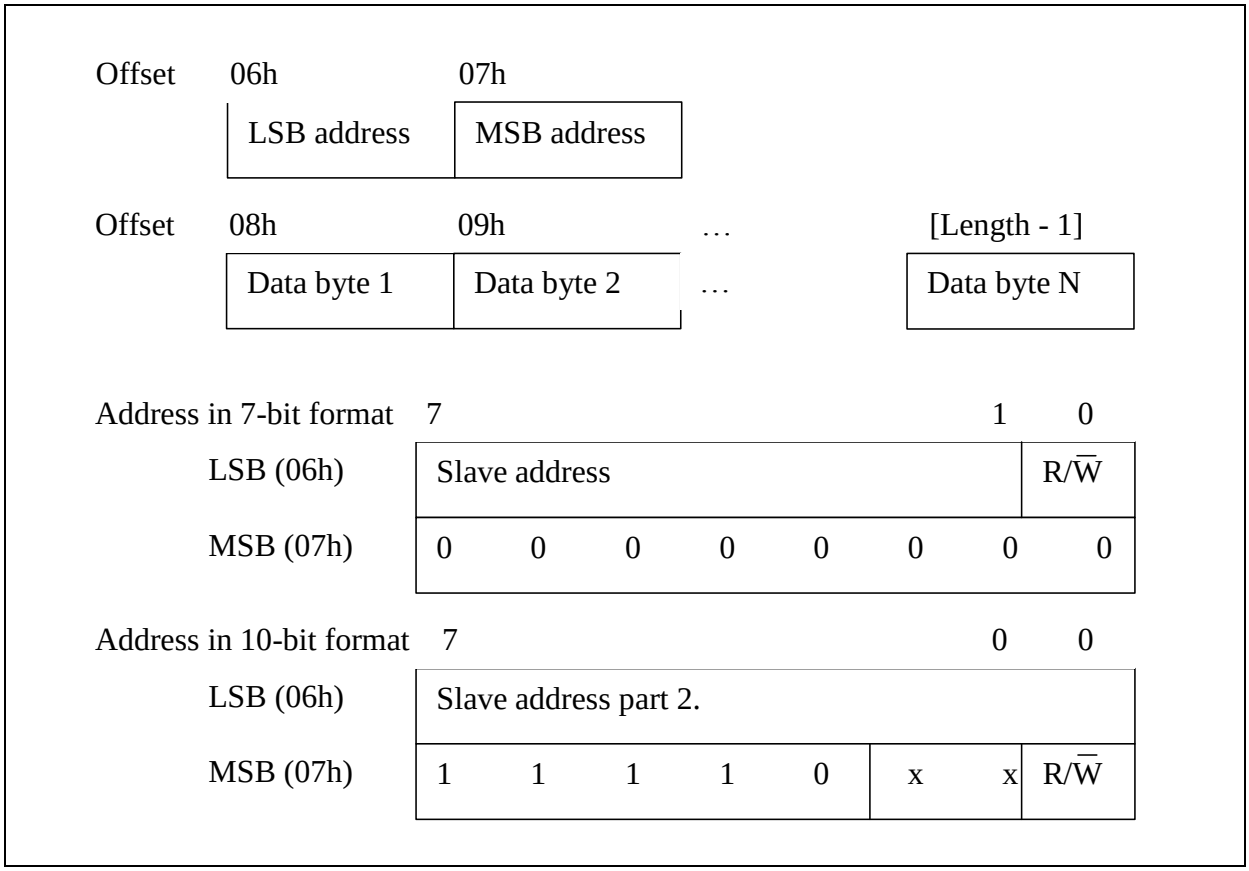

Chart 3-5: I2C Write request

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/W $\overline{\text{bit}}$ 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.

3.1.8 OP_ANAGATE_I2C_WRITE_CNF

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

:

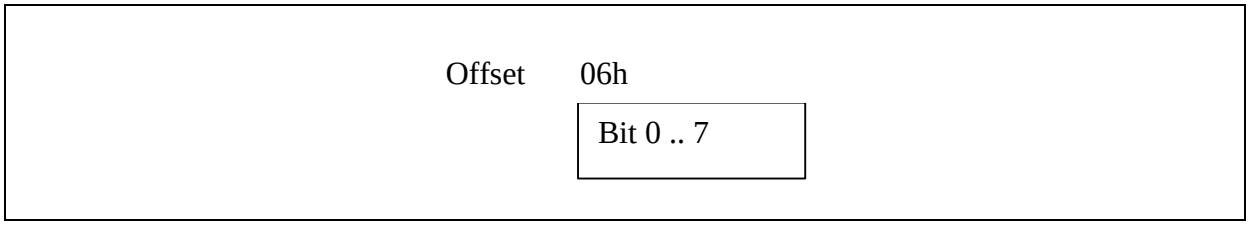

Chart 3-6: Return value for OP_ANAGATE_I2C_WRITE_CNF

The following return values may be given:

Return value	Result
00h	Write command was successful
01h	The I2C Slave has returned an NAK
02h	The I2C Slave did not respond at all

Table 3-5: Return values for OP_ANAGATE_I2C_WRITE_CNF

3.1.9 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.

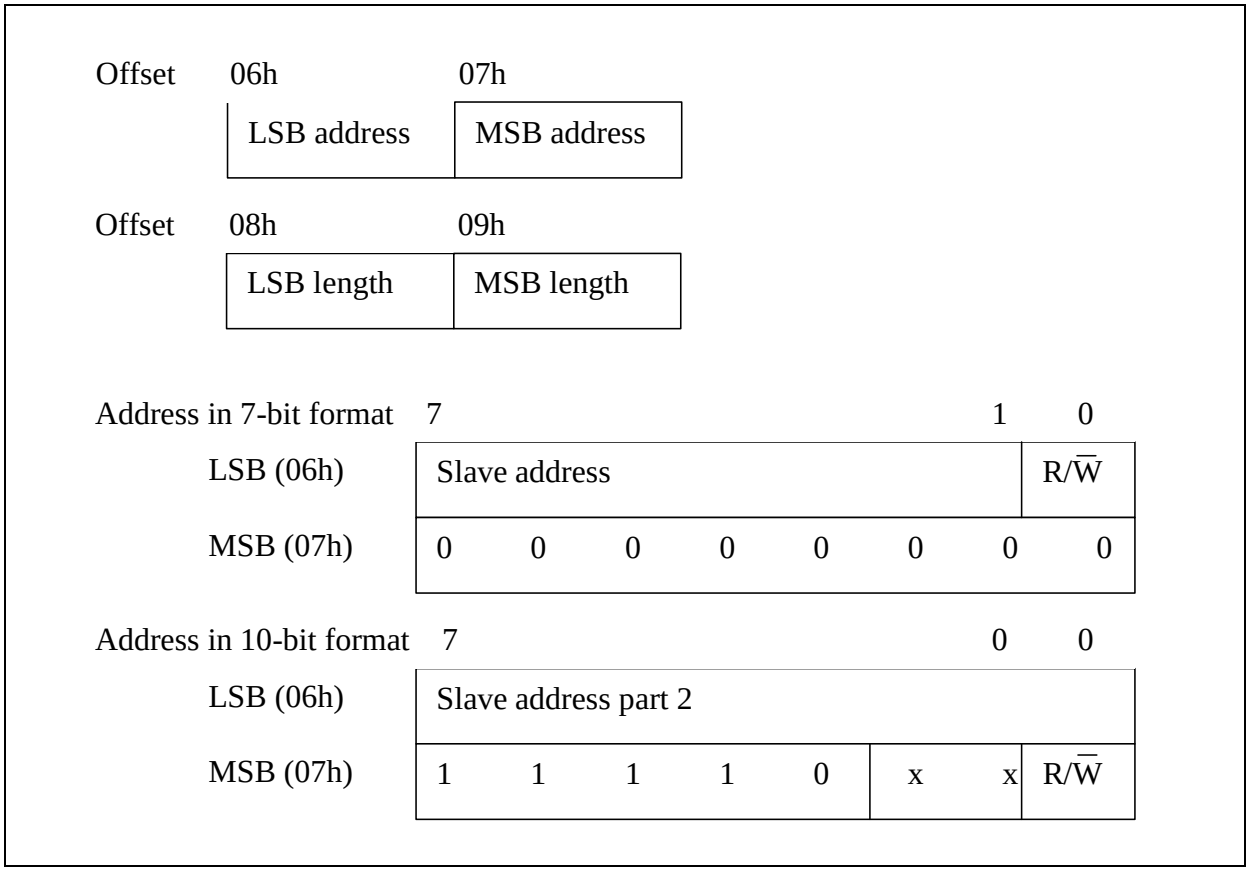

Chart 3-7: Read request

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/W $\overline{\text{bit}}$ 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.

3.1.10 OP_ANAGATE_I2C_READ_CNF

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

- Slave address

  The Slave address of the READ Request command is returned.

- Return code

  The Return values given in Table 3-6 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.

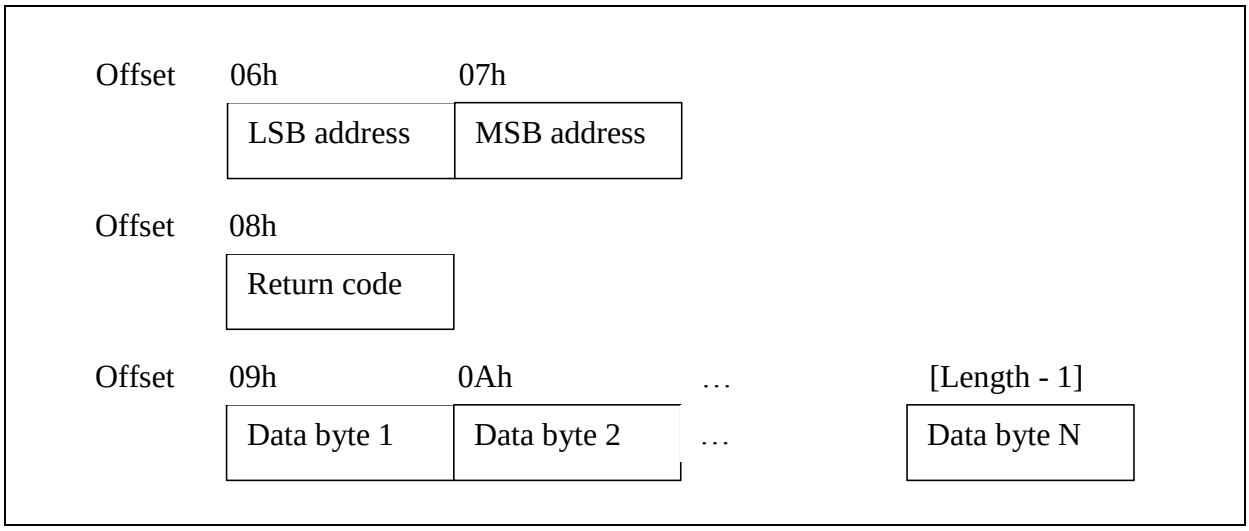

Chart 3-8: Return values for OP_ANAGATE_I2C_READ_CNF

The following return values may be given:

Return value	Result
00h	Read command was successful
01h	The I2C Slave has returned an NAK
02h	The I2C Slave did not respond at all

Table 3-6: Return values for OP_ANAGATE_I2C_READ_CNF

3.1.11 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.

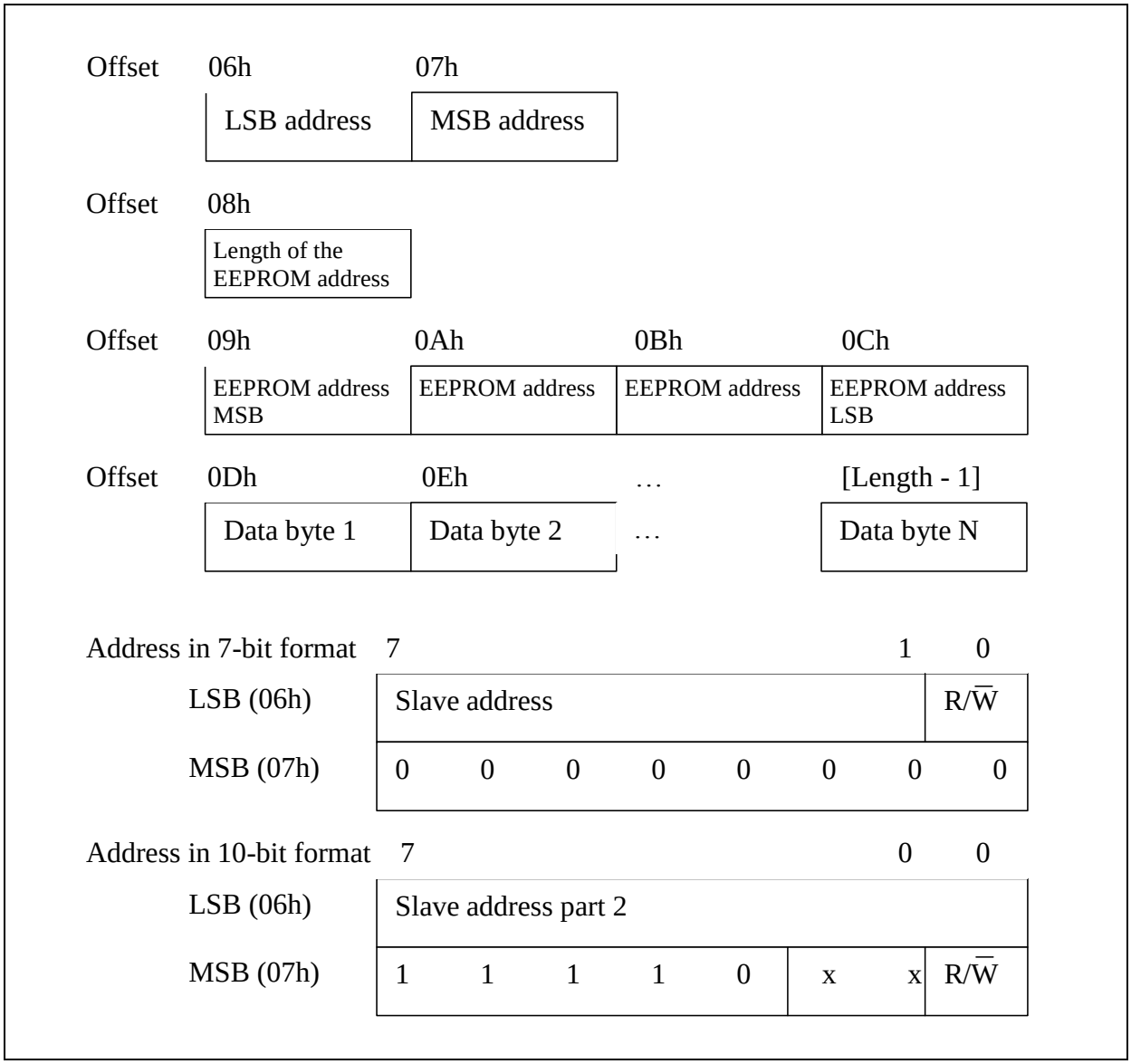

Chart 3-9: EEPROM write request

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/W $\overline{\text{bit}}$ 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.

3.1.12 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.

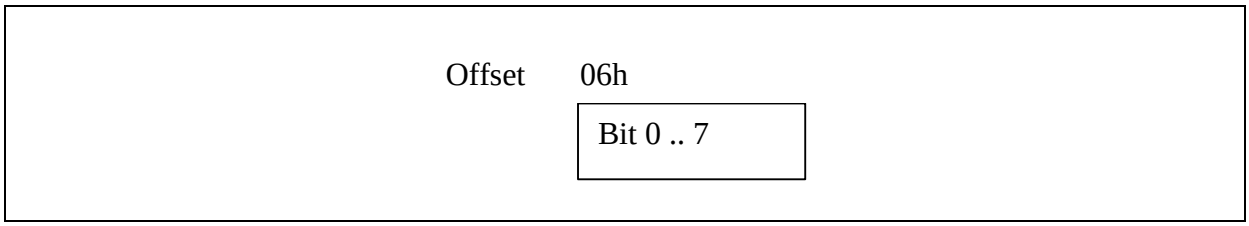

Chart 3-10: Return value for OP_ANAGATE_I2C_EEPROM_WRITE_CNF

The following return values may be given:

Return value	Result
00h	EEPROM Write command was successful
01h	The I2C Slave returned an NAK
02h	The I2C Slave did not respond at all

Table 3-7: Return values for OP_ANAGATE_I2C_EEPROM_WRITE_CNF

3.1.13 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.

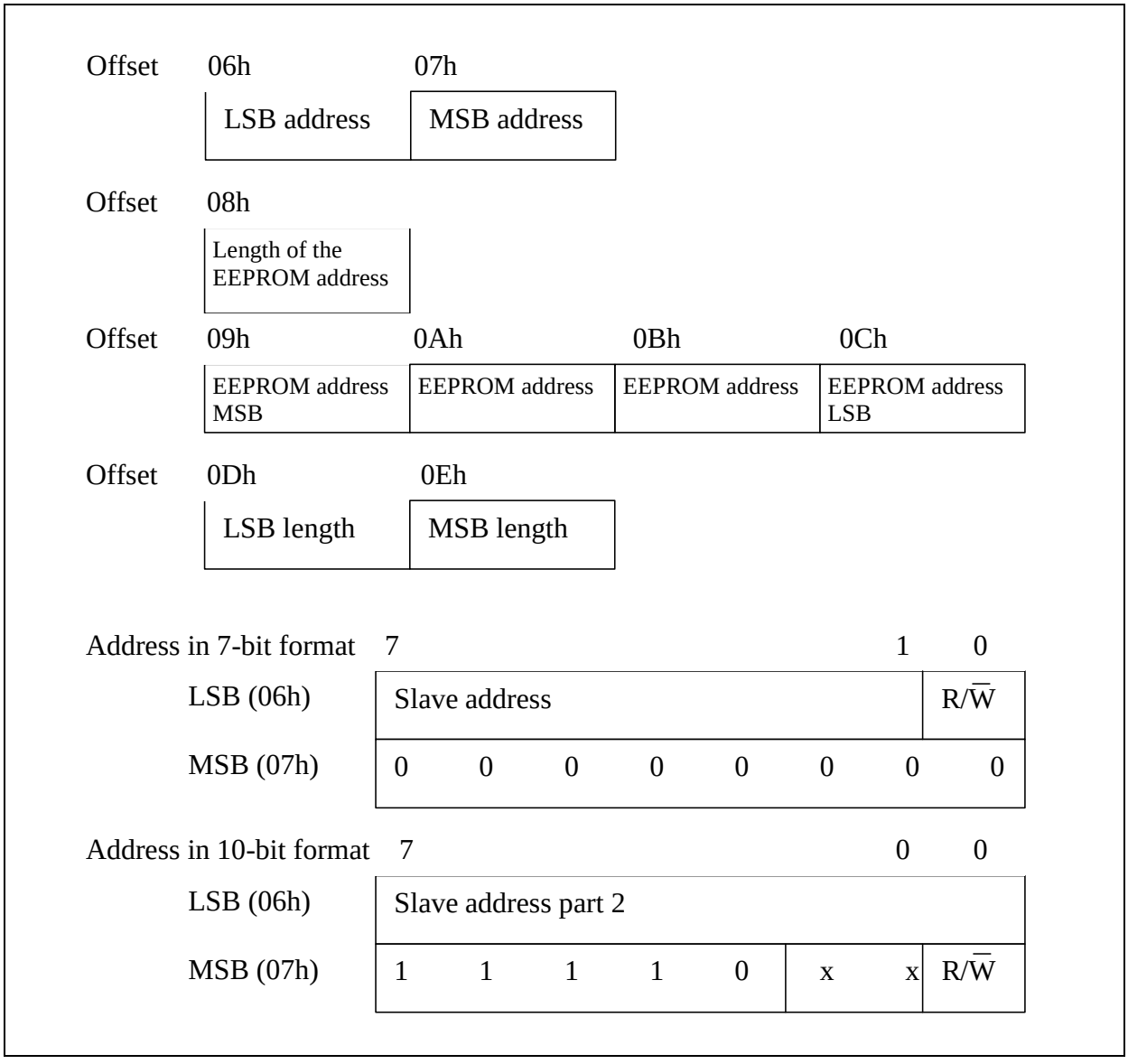

Chart 3-11: EEPROM read request

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/W $\overline{\text{bit}}$ 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.

3.1.14 OP_ANAGATE_I2C_EEPROM_READ_CNF

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

- Slave address

  The Slave address of the READ Request command is returned.

- Return code

  The Return values given in Table 3-8 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.

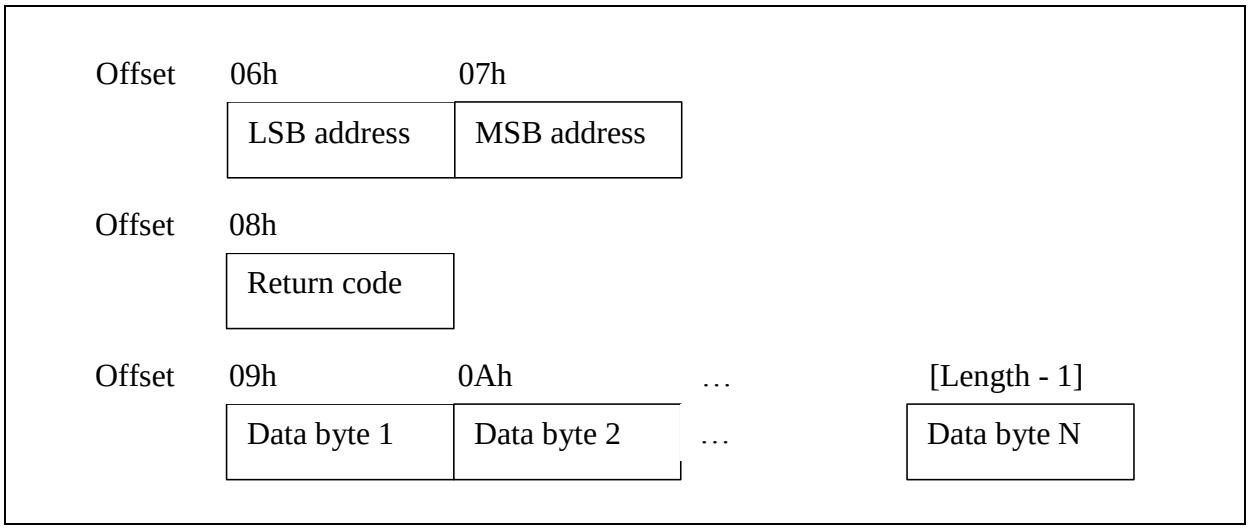

Chart 3-12: Return values for OP_ANAGATE_I2C_EEPROM_READ_CNF

The following return values may be given:

Return value	Result
00h	EEPROM Read command was successful
01h	The I2C Slave returned an NAK
02h	The I2C Slave did not respond at all

Table 3-8: Return values for OP_ANAGATE_I2C_EEPROM_READ_CNF

3.1.15 OP_ANAGATE_I2C_STATUS_REQ

The Status Request reads back the status information from the AnaGate I2C.

3.1.16 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 3-1: Return value for OP_ANAGATE_I2C_STATUS_CNF

The following return values may be given:

Return value	Result
00h	Status command was successful

Table 3-1: Return value for OP_ANAGATE_I2C_STATUS_CNF

3.2 AnaGate CAN

[Reserved for future use]

3.3 AnaGate RS232

[Reserved for future use]

3.4 AnaGate DigitalIO

[Reserved for future use]

3.5 AnaGate Audio

[Reserved for future use]

3.6 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	Controller Area Network
I2C / I^2C Bus	Inter IC Bus
LSB	Least Significant Byte
MSB	Most Significant Byte

