CAN-TALK
Monitor Protocol for CAN Bus

Protocol Description

Ablage: \UserProducts\Sw\LinMon

Document changes:

Document	Date	Author	Status, Remarks
TALK-CAN.DOC	12.6.1996	eb, Rei	V0.5 (preliminary)
Dokumentation_CAN-Talk.doc	16.5.1997	Rei	V1.0
DevDoc_CAN-Talk.doc	13.5.2002	Ro	V1.1

Table of Contents

1 Introduction

1.1 Overview

The CAN-TALK Protocol was developed by NTI for the communication between a bus master (Host PC) and one or more Slave Processor Systems (Target System). The protocol permits both the control of the target system and the data exchange between the bus master and the target system.

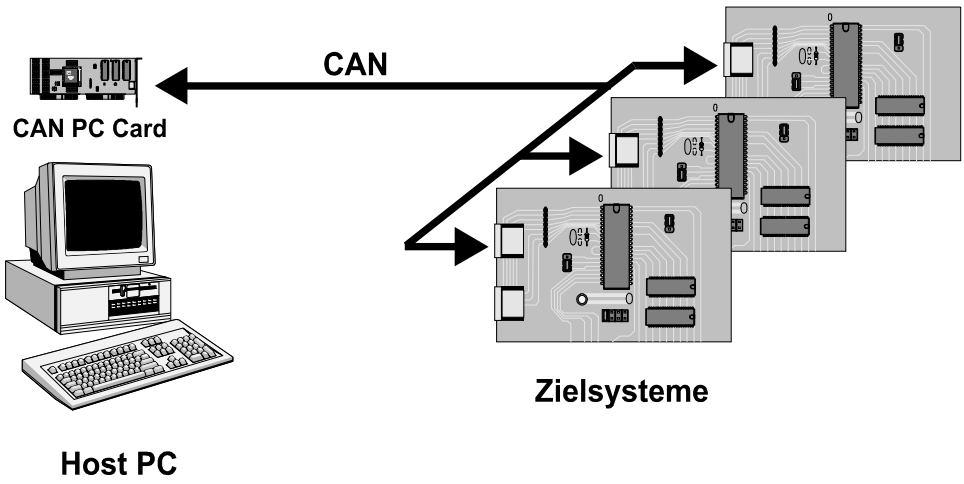

Figure 1.1: Communication with the host PC

A CAN interface card on the Host PC permits an easy communication between the host PC and one or more target system(s).

1.2 Goals and specifications

The protocol must fulfil the following specifications:

- Data exchange on the CAN interface of the Host PC in both directions (Only the Host PC can start the data transfer).
- The target system should have the possibility to send voluntary messages to the bus master.
- The protocol should permit both the data exchange between master and target system and the control of the target system.
- Transmission of single data word (16 bit) at any address of the target system.
- Transmission of big data blocks for the downloading of programs and data (ex.: tables, motion profiles, etc.)
- The protocol should permit an easy evaluation on the host PC so the PC's CPU load is as low as possible.

1.3 References

Ref	Title	Author	Date	ID	Source
[1]	Selectron MAS-DP Systemhandbuch SeleCAN		19.09.96	393.0060	Selectron AG CH-3250 Lyss
[2]	RS232-Talk				
[3]					
[4]					

1.4 Definitions, Terms and Abbreviations

Abbreviation	Meaning

2 CAN Link

2.1 Link Specification

The CAN link for data transmission between the host PC and the target system ist specified as follows:

Baud Rate: Max: 1 MBaud

Data Format: CAN 2.0A (11 Bit Identifier)

1	Start Bit
11	Identifier Bits
1	RTR Bit
1	IDE Bit
1	r0 Bit
4	DLC Bits
0…8*8	0 bis 8 Data Byte
15	CRC Bits
2	ACK Bits
10	EOF + IFS Bits

3 Structure of a CAN Message

3.1 Structure of the CAN Identifier

The CAN identifier consists of 11 bits. The CAN identifier defines the address of the CAN node and the priority of the message. The identifier is specified for the CAN Talk protocol as follows:

ID 10	ID 9	ID 8	ID 7	ID 6	ID 5	ID 4	ID 3	ID 2	ID 1	ID 0
Res	Dir	Addr 5	Addr 4	Addr 3	Addr 2	Addr 1	Addr 0	Spec 2	Spec 1	Spec 0

Res — The first bit of the identifier is reserved for future purpose. As long as this bit is unused, it must be set to 1. This keeps the possibility of sending messages with a higher prority.

Dir — The direction bit defines the direction of the CAN message:

0 Master Slave message (high priority)
1 Slave Master message (low priority)

Addr 0 - Addr 5 — The five address bits define the address of the sender or receiver (see chapter 3.3). With these five bits can be addressed up to 64 nodes.

Spec 0 – Spec 2 — These three bits are always set to 110 in the CAN talk protocol. This enables the parallel use of the CAN talk protocol and the Selectron CAN protocol [1]. For a parallel use please make sure the system communication group 2 of the Selectron protocol is not used.

3.2 Structure of the CAN data block

Die CAN Nutzdaten bestehen aus bis zu acht Datenbytes:

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service ID	Data	Data	Data	Data	Data	Data	Data

The first byte of the CAN data block is used as service ID. This service Id specifies the command which is to be executed on the target system. This is analog to the control byte of the RS232 talk protocol [2]. The rest of the data block contains command specific data.

3.3 Node address

The node address is the physical address of a bus node. The node address is selected with hex switches on the controller and defines also the priority of the node. The CAN talk protocol allows up to 64 bus nodes, for which the following addresses can be used:

Node address	**CAN-Identifier (Addr 5 … Addr 0)**	
Node address 0	00000	Noode with highest priority
Node address 1	00001	
Node address 2	00010	
:		
Node address 63	11111	Node with lowest priority

Make sure each address is only used once in a bus system.

The CAN header containing address specifies, according to the direction of the message, the address of the sender or receiver:

- Messages from host to node contain the receiver address.
  If the host uses the address 0000h, a broadcast message is sent, which is addressed to all bus nodes at once.
- Messages from node to host contain the sender address.

It is not necessary that a host node has the address 0000h. A host receives all messages going over the net and behaves according to its program. The direction bit in the CAN identifier defines if the message is vrom host to node or from node to hose. If the host address is not set to 0000h, this address should not be used for any other node because this node could only be address via broadcast messges.

4 CAN Talk Protocol

The CAN talk protocol supports two data communication ways for transferring data between host and target system.

- Standard transfer
- Block transfer from host to node

The standard transfer enables the communication of single 16 bit values. The block transfer is for transferring bigger blocks and will be used for downloading programs e.g.

4.1 Command Overview

The following table lists all the CAN-Talk commands. Afterwards all commands are described with more details.

Command ID	Description
10h	Read Memory Word
11h	Write Memory Word
12h	Write Flash Word
20h	Start Program
21h	Stop Program
22h	Reset Device
23h	Erase Flash Sector
24h	Broadcast Enable/Disable
30h	Read Memory Block
31h	Write Memory Block
32h	Download to Flash

4.2 Standard Transfer

The standard transfer supports reading and writing of 16 bit words from or to the entire memory space of the target system. In addition it is used for controlling the monitor (start, stop of application programs, erasing FLASH sectors, etc. ..)

The host sends a request message to a node (node adddress in the CAN id). The addressed node sends back a response, which contains, according to the command, the requested data or an error code.

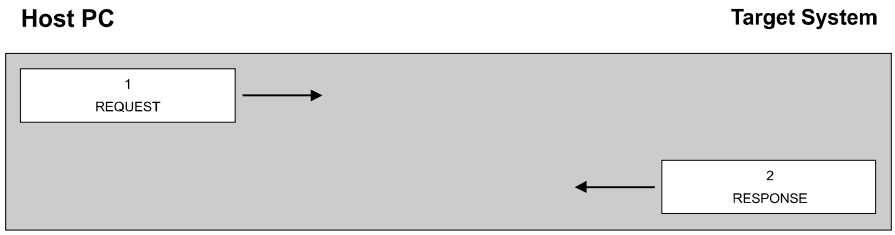

Fig. 4.1: Standard transfer

Even broadcast messages to all bus nodes can be sent as standard transfers (address 0000h in the CAN id). All bus nodes will send back a response to the host:

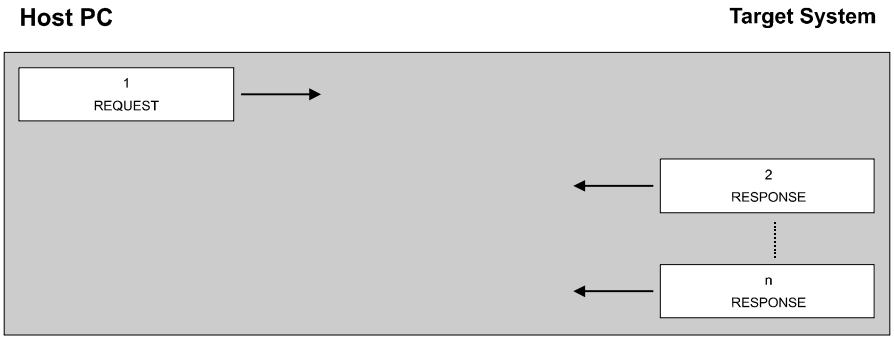

Fig. 4.2: Standard transfer broadcast

If two nodes try to send their response at the same time, the node with the lower address will be prioritized.

4.2.1 Read Memory

Reads a 16 bit word from the specified address.

Read Memory, REQUEST							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID=10h**	Address Bit 0-7	Address Bit 8-15	Address Bit 16-23	-	-	-	-

Read Memory, RESPONSE							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID=10h**	Result Ok = 00	Data Bit 0-7	Data Bit 8-15	-	-	-	-

Result	Description
00	Ok
10	Error

4.2.2 Write Memory

Writes a 16 bit wort to the specified address.

Write Memory, REQUEST							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID=11h**	Address Bit 0-7	Address Bit 8-15	Address Bit 16-23	Data Bit 0-7	Data Bit 8-15	-	-

Write Memory, RESPONSE							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID=11h**	Result Ok = 00			-	-	-	-

Result	Description
00	Ok
10	Error

4.2.3 Write Flash

Writes a 16 bit word into the flash EPROM

Write Flash, REQUEST							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID =12h**	Address Bit 0-7	Address Bit 8-15	Address Bit 16-23	Data Bit 0-7	Data Bit 8-15	-	-

Write Flash, RESPONSE							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID=12h**	Result Ok = 00			-	-	-	-

Result	**Description**
00	Ok
01	Acknowledge
40	Flash error
41	Flash timeout
42	Flash not empty
43	Flash protected

4.2.4 Start Program

Starts a user program.

Start Program, REQUEST							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID=20h**	Address Bit 0-7	Address Bit 8-15	Address Bit 16-23	-	-	-	-

Start Program, RESPONSE							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID=20h**	Result Ok = 00			-	-	-	-

Result	**Description**
00	Ok
10	Error

4.2.5 Stop Program

Stops the user program.

Stop Program, REQUEST							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID=21h**	-	-	-	-	-	-	-

Stop Program, RESPONSE							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID=21h**	Result Ok = 00			-	-	-	-

Result	**Description**
00	Ok
10	Error

4.2.6 Reset Device

A software reset will be executed on the target system. This command can only be executed, if the monitor software on the target is still running (if a complete breakdown has occurred only a hardware reset on the target system can help).

Reset, REQUEST							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID=22h**	-	-	-	-	-	-	-

Reset, RESPONSE							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID=22h**	Result Ok = 00			-	-	-	-

Result	**Description**
00	Ok
10	Error

4.2.7 Erase Flash

Erases the flash EPROM sector, which contains the specified address (REQUEST byte 1 to byte 3).

Erase Flash, REQUEST							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID = 23**	Address Bit 0-7	Address Bit 8-15	Address Bit 16-23	-	-	-	-

Erase Flash, RESPONSE 1 (enter sector erase)							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID = 23**	Result Ok = 00			-	-	-	-

Erase Flash, RESPONSE 2							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID = 23**	Result Ok = 00			-	-	-	-

Result	**Description**
00	Ok
01	Acknowledge
40	Flash error
41	Flash timeout
43	Flash protected

4.2.8 Broadcast Enable/Disable

Enables or disables the reception of broadcast messages. A “Broadcast Enable” message must also be responded even if the broadcast enable is cleared.

Broadcast Disable, REQUEST							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID = 24**	00	-	-	-	-	-	-

Broadcast Enable, REQUEST							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID = 24**	01	-	-	-	-	-	-

4.3 Block transfer from Host to Knoten

The block transfer from host to nodes enables writing of bigger data blocks (such as programs or curves, ...) from the host to one or all nodes.

The host sends a request task 1 to a node (node address in the CAN id), which contains the start address of the target range. Then, data will be transferred in parts of three words with the reqest task 2 messages. The request task 3 finishes the block transfer. After having received the request task 3, the addressed node sends back a response which contains a possible error message.

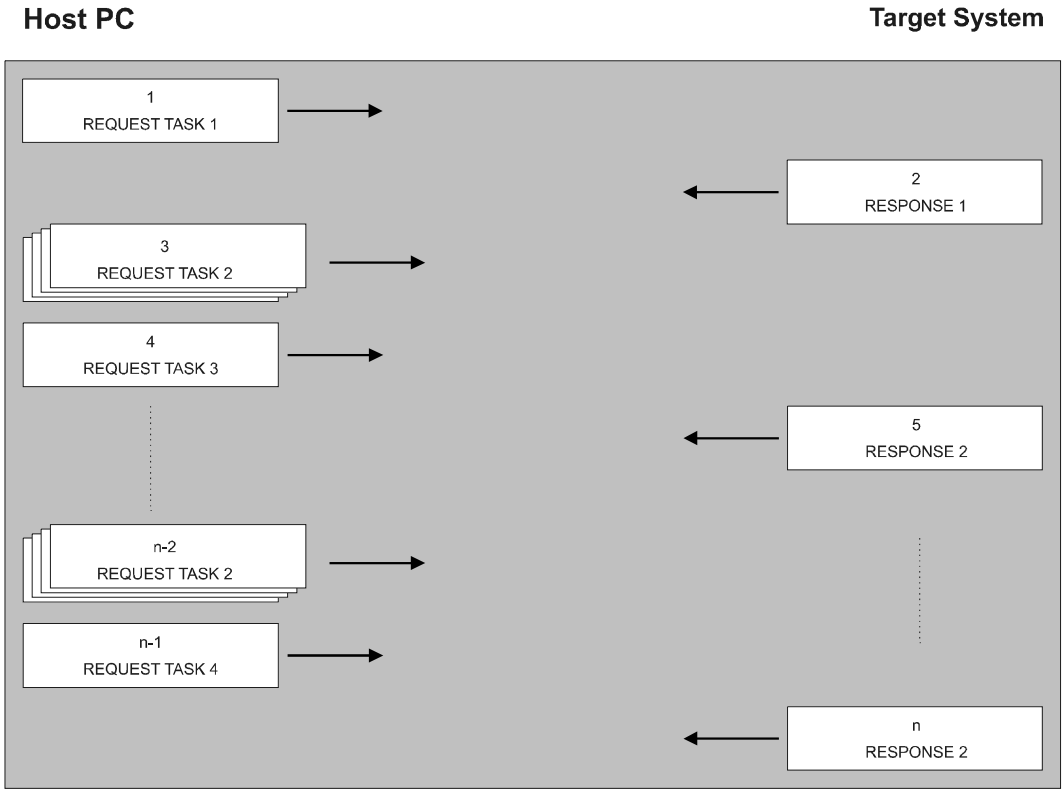

Abb. 4.1: Block Transfer Host Knoten

Der Blocktransfer Host - Knoten Transfer gestattet auch das Senden von Broadcast Datenblocks an alle Bussteilnehmer (Adresse 0000h in der CAN-ID). Nach dem Empfang des REQUEST TASK 3 senden alle Bussteilnehmer die RESPONSE an den Host zurück.

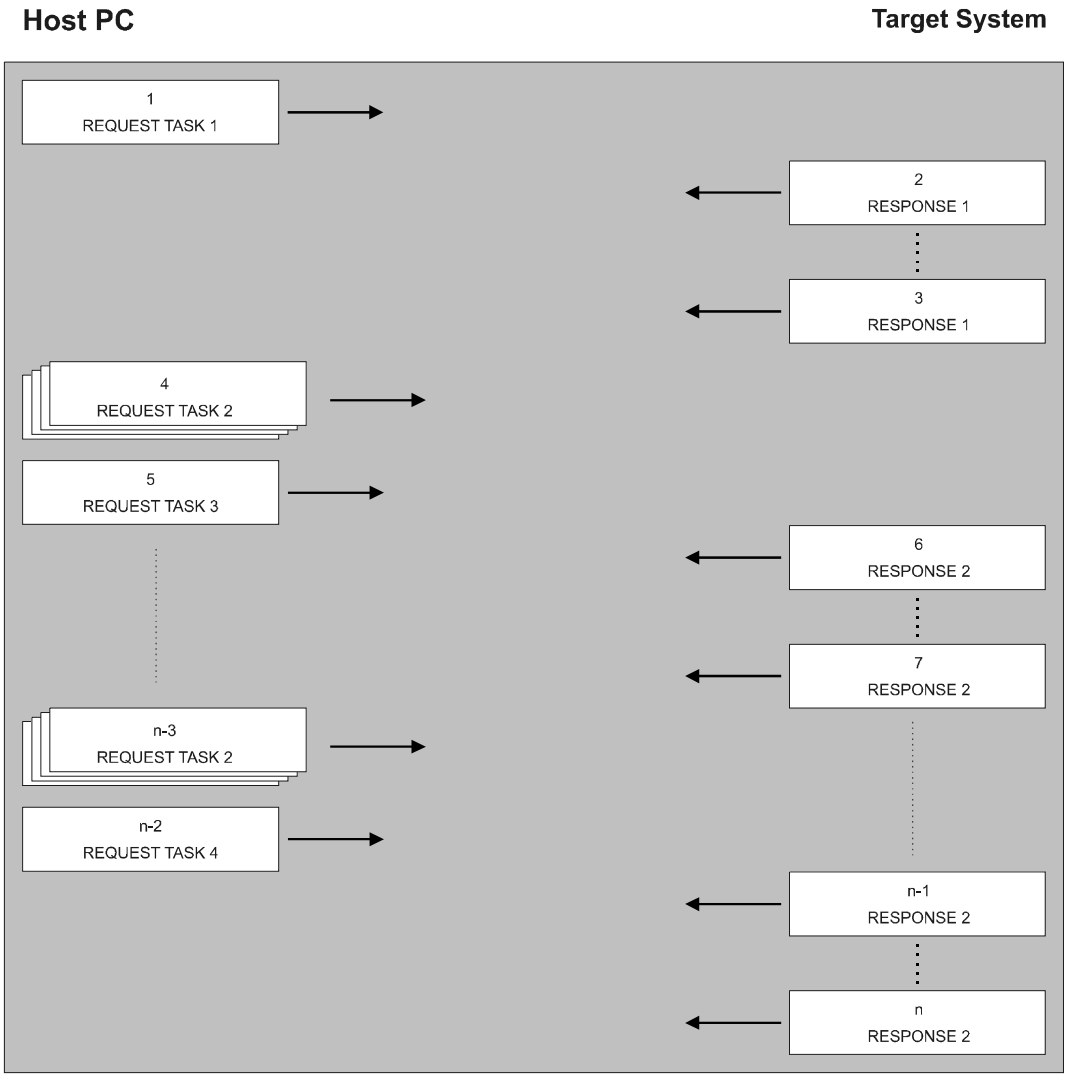

Fig. 4.2: Standard transfer host node broadcast

If two nodes try to sends their response at the same time, the node with the lower address is prioritized.

4.3.1 Read Memory Block

This command is used to read whole memory blocks and needs much less communication overhead than the word wise access. This command is available since monitor version 1.7.2.

Read Memory Block, REQUEST							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID = $30**	Request = 00	Address Bit 0-7	Address Bit 8-15	Address Bit 16-23	Size in Bytes	-	-

Read Memory Block, Response 1							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID = $30**	Result Ok = 01	Data Byte 0	Data Byte 1	Data Byte 2	Data Byte 3	Data Byte 4	Data Byte 5

Read Memory Block, Response n (n > 1)							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID = $30**	Result Ok = n	Data Byte 6*(n-1)	Data Byte 6*(n-1)+1	Data Byte 6*(n-1)+1	Data Byte 6*(n-1)+1	Data Byte 6*(n-1)+1	Data Byte 6*(n-1)+1

Result	**Description**
01	Ok
10	Error

Note: The Address and the Size have to be an even number.
Make sure this command is completely finished (request and all responses) before starting the next command.

4.3.2 Write Memory Block

This command is used to write whole memory blocks into RAM location. This command needs much less communication overhead than the word wise access. This command is available since monitor version 1.7.2.

Write Memory Block, Request 0 (set address and size)							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID = $31**	Request = 00	Address Bit 0-7	Address Bit 8-15	Address Bit 16-23	Size in Bytes	-	-

Write Memory Block, Response 0 (answer to request 0)							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID = $31**	Result Ok = 00	-	-	-	-	-	-

Write Memory Block, Request n (n > 0)							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID = $31**	Request = n	Data Byte 6*(n-1)	Data Byte 6*(n-1)+1	Data Byte 6*(n-1)+1	Data Byte 6*(n-1)+1	Data Byte 6*(n-1)+1	Data Byte 6*(n-1)+1

Write Memory Block, Response 1 (answer to request 0)							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID = $31**	Result Ok = 01	-	-	-	-	-	-

Result	**Description**
00	Ok
01	Ok, last telegram received
10	Error

Note: The Address and the Size have to be an even number.
Make sure this command is completely finished (all requests and responses) before starting the next command.

4.3.3 Download to Flash

This is used for loading data block into the flash EPROM. If an error occurs on the target system, the following data will not be stored in the flash EEPROM.

Download to Flash, REQUEST TASK 1 (set address & enter blockwrite)							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID = 32**	Task **01**	Address Bit 0-7	Address Bit 8-15	Address Bit 16-23	-	-	-

Download to Flash, REQUEST TASK 2 (add data to buffer)							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID = 32**	Task **02**	Data 0 Bit 0-7	Data 0 Bit 8-15	[Data 1] Bit 0-7	[Data 1] Bit 8-15	[Data 2] Bit 0-7	[Data 2] Bit 8-15

Download to Flash, REQUEST TASK 3 (add data to buffer and write buffer)							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID = 32**	Task **03**	Data 0 Bit 0-7	Data 0 Bit 8-15	[Data 1] Bit 0-7	[Data 1] Bit 8-15	[Data 2] Bit 0-7	[Data 2] Bit 8-15

Download to Flash, REQUEST TASK 4 (add data, write buffer and exit)							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID = 32**	Task **04**	Data 0 Bit 0-7	Data 0 Bit 8-15	[Data 1] Bit 0-7	[Data 1] Bit 8-15	[Data 2] Bit 0-7	[Data 2] Bit 8-15

Download to Flash, RESPONSE 1 (to set address & enter blockwrite)							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID = 32**	Result ok = 00	Length of buffer in bytes		-	-	-	-

Download to Flash, RESPONSE 2 (send result)							
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Service **ID = 32**	Result			-	-	-	-

Result	**Description**
00	Ok
10	Device busy
11	Service protected
20	Protocol error
22	Frame size error
24	Buffer overflow
31	Unknown service
40	Flash error
41	Flash timeout
42	Flash not empty
43	Flash protected

5 Protocol of changes

changed version	new version	Short description / remarks	date	name
	1.0	CAN-TALK protocol description	25.10.1996	eb / 0150
1.0	1.1	Block Read/Write added	13.5.2002	Ro