Programming manual
API Description

Byte Command Protocol

GB

1520792 / 00 11 / 2023

Inhaltsverzeichnis

1 Preliminary note . . . 3
1.1 Legal and copyright information . . . 3
1.2 General . . . 3
1.3 Symbols used . . . 4
1.4 History of the instructions . . . 4
1.4.1 2023/11 - 11520792 / 00 . . . 4
1.4.2 2017/08 - 7391131 / 01 . . . 4

2 Byte Mode Commands . . . 5
2.1 Command Format (Byte Mode) . . . 6
2.1.1 Example – Command Format . . . 6
2.2 Extended Command Format (Byte Mode) . . . 7
2.2.1 Example – Extended Command . . . 7
2.3 Process Data Messages . . . 8
2.3.1 CAN data frame reception . . . 8
2.3.1.1 Example – Receive . . . 8
2.3.2 Transmit CAN data frame . . . 9
2.3.3 CAN remote frame reception . . . 9
2.3.4 Transmit CAN remote frame . . . 10
2.4 Feedback messages . . . 11
2.4.1 CAN message transmission feedback . . . 11
2.5 Diagnosis data messages . . . 12
2.5.1 Device ID . . . 12
2.5.2 Hardware / Software version . . . 12
2.5.2.1 Example – Reading Hardware / Software version . . . 12
2.5.3 Error status . . . 13
2.5.4 Interfaces version . . . 13
2.5.5 Device busy / command not supported . . . 14
2.6 CAN Controller setup messages . . . 15
2.6.1 Change / request receive channel ID settings . . . 15
2.6.2 Change / request receive channel mask settings . . . 15
2.6.3 CAN baud rate . . . 16
2.6.4 CAN controller reset . . . 16
2.6.5 Set / request active / passive mode . . . 17
2.6.6 Change / request receive channel cycle time . . . 17
2.7 Host interface configuration messages . . . 19
2.7.1 Feedback / output settings . . . 19
2.8 Device parameter settings . . . 20
2.8.1 Change / request Timestamp settings . . . 20
2.8.2 CANopen object dictionary access . . . 20
2.9 Hardware access commands . . . 22
2.9.1 Reset device . . . 22
2.9.2 Request analog channel value . . . 22
2.9.3 Set Communication password . . . 22
2.9.4 Reset Communication password . . . 23

3 Commands and Devices . . . 24

1 Preliminary note

1.1 Legal and copyright information

© All rights reserved by ifm electronic gmbh. No part of these instructions may be reproduced and used without the consent of ifm electronic gmbh.

All product names, pictures, companies or other brands used on our pages are the property of the respective rights owners.

- AS-i is the property of AS-International Association, (→ www.as-interface.net)
- CAN is the property of Robert Bosch GmbH, Germany (→ www.bosch.de)
- CAN is the property of CiA (CAN in Automation e.V.), Germany (→ www.can-cia.org)
- CODESYS™ is the property of CODESYS GmbH, Germany (→ www.codesys.com)
- DeviceNet™ is the property of ODVA™ (Open DeviceNet Vendor Association), USA (→ www.odva.org)
- EtherNet/IP® is the property of → ODVA™
- EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.
- IO-Link® is the property of PROFIBUS Nutzerorganisation e.V., Germany (→ www.io-link.com)
- ISOBUS is the property of AEF - Agricultural Industry Electronics Foundation e.V., Germany (→ www.aef-online.org)
- Microsoft® is the property of Microsoft Corporation, USA (→ www.microsoft.com)
- Modbus® is the property of Schneider Electric SE, France (→ www.schneider-electric.com)
- PROFIBUS® is the property of PROFIBUS Nutzerorganisation e.V., Germany (→ www.profibus.com)
- PROFINET® is the property of → PROFIBUS Nutzerorganisation e.V., Deutschland
- Windows® is the property of → Microsoft Corporation, USA

1.2 General

This manual describes the structure and commands used by the Byte Command Protocol. This protocol is supported by all CAN to wireless gateways from ifm electronic. It is meant for communication with a remote host via the interface supported by the device. This manual extends the device specific manuals. In case of uncertainties please also refer to the device manuals of your specific gateway device.

The CAN to wireless gateways connect a CAN network to a remote host using a specific interface like Bluetooth or WLAN. The Byte Command Protocol serves for bi-directional transparent CAN message transmission between the host and the CAN network as well as for the transfer of commands between host and gateway. Since all CAN to wireless gateways work on the same protocol any application software can be easily extended to use different communication interfaces by just exchanging the basic communication routines.

Most of the commands are supported by all gateways, in some cases special commands are not supported by single devices. The table in chapter Commands and Devices (→ 🗋 24) gives you an overview on the commands supported by each single gateway-type.

Normally all commands which influence devices settings are only stored in the volatile RAM memory of the gateway, after a reset the settings stored in the devices non-volatile memory are used again. You can change these settings with the device specific configurator software or – if supported by the device – via CANopen.

Exception: The command CAN baud rate (→ 🗋 16) is stored permanently.

The latest versions of the drivers, software, firmware, documentations and further information and support are available on our website: www.ifm.com

1.3 Symbols used

✓	Requirement
▶	Instructions
▷	Reaction, result
[...]	Designation of keys, buttons or indications
→	Cross-reference
	Important note Non-compliance may result in malfunction or interference.
	Information Supplementary note

1.4 History of the instructions

1.4.1 2023/11 - 11520792 / 00

- Update of the document

1.4.2 2017/08 - 7391131 / 01

- New creation of the document

2 Byte Mode Commands

2.1 Command Format (Byte Mode) 6
2.1.1 Example – Command Format 6
2.2 Extended Command Format (Byte Mode) 7
2.2.1 Example – Extended Command 7
2.3 Process Data Messages 8
2.3.1 CAN data frame reception 8
2.3.2 Transmit CAN data frame 9
2.3.3 CAN remote frame reception 9
2.3.4 Transmit CAN remote frame 10
2.4 Feedback messages 11
2.4.1 CAN message transmission feedback 11
2.5 Diagnosis data messages 12
2.5.1 Device ID 12
2.5.2 Hardware / Software version 12
2.5.3 Error status 13
2.5.4 Interfaces version 13
2.5.5 Device busy / command not supported 14
2.6 CAN Controller setup messages 15
2.6.1 Change / request receive channel ID settings 15
2.6.2 Change / request receive channel mask settings 15
2.6.3 CAN baud rate 16
2.6.4 CAN controller reset 16
2.6.5 Set / request active / passive mode 17
2.6.6 Change / request receive channel cycle time 17
2.7 Host interface configuration messages 19
2.7.1 Feedback / output settings 19
2.8 Device parameter settings 20
2.8.1 Change / request Timestamp settings 20
2.8.2 CANopen object dictionary access 20
2.9 Hardware access commands 22
2.9.1 Reset device 22
2.9.2 Request analog channel value 22
2.9.3 Set Communication password 22
2.9.4 Reset Communication password 23

2.1 Command Format (Byte Mode)

Byte(s)	Value	Description
SOF (Start of Frame)	'C'	SOF marks the beginning of the command. 'C' (0x43) is the default setting. For some devices this value can be changed. This manual always shows the default setting.
Length	0x00-0xFF	The length byte includes the number of data bytes + the number of command bytes following.
Command	0x00-0xFF	Command byte, values 0x00 to 0xFE are direct commands, the value 0xFF means that the next byte contains a command extension
Data	xx	Data content
Checksum	xx	Contains the XOR checksum of the SOF, Length, Command and the Data-bytes.
EOF (End of Frame)	<CR>	EOF byte marks the end of the command. 0x0D (Carriage Return) is the default setting. For some devices this value can be changed. This manual always shows the default setting

"--" means that this value is not of interest

"xx" means that the field contains the configuration / reply value

Each message which changes or requests the settings of the CAN device is replied by a message with the same command byte, containing the new settings.

Request commands normally do not require additional parameters (no "Data" field).

Exceptions are mentioned in the command description.

2.1.1 Example – Command Format

Byte(s)	Value	Description
SOF (Start of Frame)	'0x43'	SOF marks the beginning of the command.
Length	0x0B	The length byte includes the number of data bytes + the number of command bytes following.
Command	0x00	11 bit ID CAN message
Data	0x07, 0x89, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17, 0x18	ID (0x789) (MSB first) CAN-message Data content
Checksum	0xCE	Contains the XOR checksum of the SOF, Length, Command and the Data-bytes.
EOF (End of Frame)	0x0D	EOF byte marks the end of the command.

2.2 Extended Command Format (Byte Mode)

The Extended Command Format is only available for certain devices. The table in chapter Commands and Devices (→ 24) gives an overview on the commands supported by each device type.

Byte(s)	Value	Description
SOF (Start of Frame)	'C'	SOF marks the beginning of the command. 'C' (0x43) is the default setting. For some devices this value can be changed. This manual always shows the default setting.
Length	0x00-0xFF	The length byte includes the number of data bytes + the number of command bytes following.
Extended Command	0xD0-0xDF	Command byte, values 0xD0 to 0xDF are extended commands
Data	xx	Byte 0: CAN-Channel (starting at 0 for CAN 1) / 128 … 255 reserved for module (WLAN / GSM / GPS…) Byte 1: Command Byte 2: Data content
Checksum	xx	Contains the XOR checksum of the SOF, Length, Command and the Data-bytes.
EOF (End of Frame)	<CR>	EOF byte marks the end of the command. 0x0D (Carriage Return) is the default setting. For some devices this value can be changed. This manual always shows the default setting

"--" means that this value is not of interest

"xx" means that the field contains the configuration / reply value

2.2.1 Example – Extended Command

Byte(s)	Value	Description
SOF (Start of Frame)	'0x43'	SOF marks the beginning of the command.
Length	0x0D	The length byte includes the number of data bytes + the number of command bytes following.
Command	0xD0	Extended Command
Data	0x01 0x00 0x07, 0x89, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17, 0x18,	CAN-Channel: 2 Command: 11 bit ID CAN message ID (0x789) (MSB first) CAN-message Data content
Checksum	0x19	Contains the XOR checksum of the SOF, Length, Command and the Data-bytes.
EOF (End of Frame)	0x0D	EOF byte marks the end of the command.

2.3 Process Data Messages

These messages are used to transmit received CAN messages to the host or to send CAN messages from the host to the CAN network.

2.3.1 CAN data frame reception

CAN data frame messages are received by the gateway from the CAN bus.

These CAN data frame messages are transferred to the host interface in the following format.

Byte(s)		Value	Description
SOF (Start of Frame)		'C'	SOF marks the beginning of the command.
Length		xx	The length byte contains the number of data bytes + the number of command bytes following.
Command		0x00 0x01 0x02 0x03	11 bit ID CAN message received 11 bit ID CAN message received, message contains an additional 32 bit timestamp value. 29 bit ID CAN message received 29 bit ID CAN message received, message contains an additional 32 bit timestamp value.
Data	For 11 bit ID messages :	0x00-0xFF	**Byte 0-1:** 11 bit CAN ID (MSB first) **additional data bytes:** Up to 8 CAN data bytes + an additional 32 bit timestamp value (only if timestamp option is enabled, see below)
	For 29 bit ID messages:	0x00-0xFF	**Byte 0-3:** 29 bit CAN ID (MSB first) **additional data bytes:** Up to 8 CAN data bytes + an additional 32 bit timestamp value (only if timestamp option is enabled, see below)
Checksum		xx	Contains the XOR checksum of the SOF, Length, Command and the Data-bytes.
EOF (End of Frame)		<CR>	EOF byte marks the end of the command.

If a timestamp is desired, this option must be enabled with the corresponding command message. (→ Change / request Timestamp settings 20)

The usage as extended command is possible. (→ Extended Command Format (Byte Mode) 7)

2.3.1.1 Example – Receive

Byte #	Byte(s)		Value	Description
1	SOF (Start of Frame)		'C'	SOF marks the beginning of the command.
2	Length		= 1 + 14 = 0xF	The length byte contains the number of command bytes following + the number of data bytes.
3	Command		0x01	11 bit ID CAN message received, message contains an additional 32 bit timestamp value.
4, 5	Data	11 bit ID	0x07, 0x89	ID: 0x789
6 ... 13		CAN data	0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17, 0x18	Data: 0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17, 0x18
14 ... 17		Timestamp	0x23, 0x22, 0x21, 0x20	Timestamp: 0x23222120
	Checksum		0xCB	Contains the XOR checksum of the SOF, Length, Command and the Data-bytes.

Byte #	Byte(s)	Value	Description
	EOF (End of Frame)	<CR>	EOF byte marks the end of the command.

2.3.2 Transmit CAN data frame

In order to transmit CAN data frame messages to the CAN bus, the host interface sends data to the gateway in the following format.

Byte(s)		Value	Description
SOF (Start of Frame)		'C'	SOF marks the beginning of the command.
Length		xx	The length byte contains the number of data bytes + the number of command bytes following.
Command		0x00 0x02	Transmit 11 bit ID CAN data frame Transmit 29 bit ID CAN data frame
Data	For 11 bit ID messag-es:	0x00-0xFF	**Byte 0-1:** 11 bit CAN ID (MSB first) **additional data bytes:** Up to 8 CAN data bytes
	For 29 bit ID messag-es:	0x00-0xFF	**Byte 0-3:** 29 bit CAN ID (MSB first) **additional data bytes:** Up to 8 CAN data bytes
Checksum		xx	Contains the XOR checksum of the SOF, Length, Command and the Data-bytes.
EOF (End of Frame)		<CR>	EOF byte marks the end of the command.

The usage as extended command is possible. (→ Extended Command Format (Byte Mode) 7)

2.3.3 CAN remote frame reception

CAN remote frame messages received by the gateway from the CAN bus are transferred to the host interface in the following format.

Byte(s)		Value	Description
SOF (Start of Frame)		'C'	SOF marks the beginning of the command.
Length		xx	The length byte contains the number of data bytes + the number of command bytes follow-ing.
Command		0x04 0x05 0x06 0x07	11 bit ID CAN remote frame received 11 bit ID CAN remote frame received, message contains an additional 32 bit timestamp val-ue. 29 bit ID CAN remote frame received 29 bit ID CAN remote frame received, message contains an additional 32 bit timestamp val-ue.
Data	For 11 bit ID re-mote frames:	0x00-0xFF	**Byte 0-1:** 11 bit CAN ID (MSB first) **additional data bytes:** DLC (CAN remote frame Data Length Code) + an additional 32 bit timestamp value (only if timestamp option is enabled, see below)
	For 29 bit ID re-mote frames:	0x00-0xFF	**Byte 0-3:** 29 bit CAN ID (MSB first) **additional data bytes:** DLC (CAN remote frame Data Length Code) + an additional 32 bit timestamp value (only if timestamp option is enabled, see below)

Byte(s)	Value	Description
Checksum	xx	Contains the XOR checksum of the SOF, Length, Command and the Data-bytes.
EOF (End of Frame)	<CR>	EOF byte marks the end of the command.

If a timestamp is desired, this option must be enabled with the corresponding command message. (→ Change / request Timestamp settings 20)

The usage as extended command is possible. (→ Extended Command Format (Byte Mode) 7)

2.3.4 Transmit CAN remote frame

In order to transmit CAN remote frame messages to the CAN bus, the host interface sends data to the gateway in the following format.

Byte(s)		Value	Description
SOF (Start of Frame)		'C'	SOF marks the beginning of the command.
Length		xx	The length byte contains the number of data bytes + the number of command bytes following.
Command		0x04 0x06	Transmit 11 bit ID CAN remote frame Transmit 29 bit ID CAN remote frame
Data	For 11 bit ID remote frames:	0x00-0xFF	**Byte 0-1:** 11 bit CAN ID (MSB first) **additional data byte:** DLC (CAN remote frame Data Length Code)
	For 29 bit ID remote frames:	0x00-0xFF	**Byte 0-3:** 29 bit CAN ID (MSB first) **additional data byte:** DLC (CAN remote frame Data Length Code)
Checksum		xx	Contains the XOR checksum of the SOF, Length, Command and the Data-bytes.
EOF (End of Frame)		<CR>	EOF byte marks the end of the command.

The usage as extended command is possible. (→ Extended Command Format (Byte Mode) 7)

2.4 Feedback messages

These messages contain feedback data which allows the user to check the correct transmission of CAN messages.

2.4.1 CAN message transmission feedback

Byte(s)		Value	Description
SOF (Start of Frame)		'C'	SOF marks the beginning of the command.
Length		xx	The length byte contains the number of data bytes + the number of command bytes following.
Command		0x20 0x21 0x22 0x23 0x24 0x25 0x26 0x27	11 bit ID CAN message transmitted 11 bit ID CAN message transmitted, feedback contains an additional timestamp of the transmission time 29 bit ID CAN message transmitted 29 bit ID CAN message transmitted, feedback contains an additional timestamp of the transmission time 11 bit ID CAN remote frame transmitted 11 bit ID CAN remote frame transmitted, feedback contains an additional timestamp of the transmission time 29 bit ID CAN remote frame transmitted 29 bit ID CAN remote frame transmitted, feedback contains an additional timestamp of the transmission time
Data	For 11 bit ID messages / remote frames:	0x00-0xFF 0x00-0xFF	**Byte 0-1:** 11 bit CAN ID (MSB first) **additional data bytes (not on remote frame feedback):** Up to 8 CAN Data bytes + an additional 32 bit timestamp value (only if timestamp option is enabled, see below)
	For 29 bit ID messages / remote frames:	0x00-0xFF 0x00-0xFF	**Byte 0-3:** 29 bit CAN ID (MSB first) **additional data bytes(not on remote frame feedback):** Up to 8 CAN Data bytes + an additional 32 bit timestamp value (only if timestamp option is enabled, see below)
Checksum		xx	Contains the XOR checksum of the SOF, Length, Command and the Data-bytes.
EOF (End of Frame)		<CR>	EOF byte marks the end of the command.

If feedback messages are desired, the feedback option has to be enabled by the corresponding command. (→ Feedback / output settings 19)

If a timestamp is desired, this option must be enabled with the corresponding command message. (→ Change / request Timestamp settings 20)

The usage as extended command is possible. (→ Extended Command Format (Byte Mode) 7)

2.5 Diagnosis data messages

2.5.1 Device ID

Byte(s)	Value	Description
SOF (Start of Frame)	'C'	SOF marks the beginning of the command.
Length	xx	The length byte contains the number of data bytes + the number of command bytes following.
Command	0x40	Device ID string (request: no additional data bytes)
Data (only on reply)	0x00-0xFF	ID string (only on response)
Checksum	xx	Contains the XOR checksum of the SOF, Length, Command and the Data-bytes.
EOF (End of Frame)	<CR>	EOF byte marks the end of the command.

The last 2 data bytes of the ID String contain a 16 bit value for individual device identification. This value can be manipulated with the devices individual configurator software.

2.5.2 Hardware / Software version

Byte(s)	Value	Description
SOF (Start of Frame)	'C'	SOF marks the beginning of the command.
Length	xx	The length byte contains the number of data bytes + the number of command bytes following.
Command	0x41	Hardware / Software version (request: no additional data bytes)
Data (only on reply)	0x00-0xFF	**Byte 0-1 :** Hardware version of the device **Byte 2-3 :** Software version of the firmware **Byte 4-5 :** Software version of the bootloader
Checksum	xx	Contains the XOR checksum of the SOF, Length, Command and the Data-bytes.
EOF (End of Frame)	<CR>	EOF byte marks the end of the command.

2.5.2.1 Example – Reading Hardware / Software version

Read Request

Byte #	Byte(s)	Value
1	SOF (Start of Frame)	'C'
2	Length	0x01
3	Command	0x41
4	Checksum	0x03
5	EOF (End of Frame)	<CR>

Device Response

Byte #	Byte(s)	Value
1	SOF (Start of Frame)	'C'
2	Length	0x07

Byte #	Byte(s)		Value
3	Command		0x41
4, 5	Data	Hardware version	0x02, 0x01
6, 7		Software version firmware	0x02, 0x00
7, 8		Software version bootloader	0x00, 0x00
9	Checksum		0x04
10	EOF (End of Frame)		<CR>

2.5.3 Error status

Byte(s)	Value	Description
SOF (Start of Frame)	'C'	SOF marks the beginning of the command.
Length	xx	The length byte contains the number of data bytes + the number of command bytes following.
Command	0x42	Error status (request: no additional data bytes)
Data (only on reply)	0x00-0xFF	Byte 0: bit 0: CAN receive buffer overflow bit 1: CAN transmit timeout bit 2: CAN error counter overflow bit 3: CAN bus-off error bit 4: Host interface syntax error bit 5: Host interface format error bit 6: Host interface transmit buffer overflow bit 7: not used Byte 1 (Last error code of the CAN controller): '0': No Error '1': More than 5 equal bits in a sequence have occurred '2': Received message does not have the right format '3': Transmitted Message was not acknowledged '4': Not possible to set recessive level during transmission '5': Not possible to set dominant level during transmission '6': received CRC check sum was incorrect
Checksum	xx	Contains the XOR checksum of the SOF, Length, Command and the Data-bytes.
EOF (End of Frame)	<CR>	EOF byte marks the end of the command.

If a timestamp is desired, this option must be enabled with the corresponding command message. (→ Change / request Timestamp settings 20)

Each time the error status changes, an error status message is generated automatically.

The usage as extended command is possible. (→ Extended Command Format (Byte Mode) 7)

2.5.4 Interfaces version

Byte(s)	Value	Description
SOF (Start of Frame)	'C'	SOF marks the beginning of the command.
Length	xx	The length byte contains the number of data bytes + the number of command bytes following.

Byte(s)	Value	Description
Command	0x44	Interfaces version (e.g. Bluetooth module, WLAN-module, …) (request: no additional data bytes)
Data (only on reply)	0x00-0xFF	ASCII coded version string
Checksum	xx	Contains the XOR checksum of the SOF, Length, Command and the Data-bytes.
EOF (End of Frame)	<CR>	EOF byte marks the end of the command.

The usage as extended command is possible. (→ Extended Command Format (Byte Mode) 7)

Modul is selectable.

2.5.5 Device busy / command not supported

Byte(s)	Value	Description
SOF (Start of Frame)	'C'	SOF marks the beginning of the command.
Length	xx	The length byte contains the number of data bytes + the number of command bytes following.
Command	0x48	Device cannot execute command (only response on other commands)
Data	0x00-0xFF	Reserved for future use
Checksum	xx	Contains the XOR checksum of the SOF, Length, Command and the Data-bytes.
EOF (End of Frame)	<CR>	EOF byte marks the end of the command.

The usage as extended command is possible. (→ Extended Command Format (Byte Mode) 7)

2.6 CAN Controller setup messages

2.6.1 Change / request receive channel ID settings

Byte(s)	Value	Description
SOF (Start of Frame)	'C'	SOF marks the beginning of the command.
Length	xx	The length byte contains the number of data bytes + the number of command bytes following.
Command	0x50 0x51	Request receive channel ID settings (needs one additional data byte = channel number) Change receive channel ID settings (needs at least two additional data bytes, the channel number and the channel settings)
Data	0x00-0xFF 0x00-0x07	Byte 0: Receive object number (range depends on gateway device) 0…15 Byte 1: bit 0: 0 = channel disable, 1 = channel enable bit 1: 0 = receive 11 bit ID, 1 = receive 29 bit ID bit 2: 0 = change 11 bit ID, 1 = change 29 bit ID Byte 2-3: 11 bit ID if byte1, bit 2 = 0 Byte 2-5: 29 bit ID if byte1, bit 2 = 1
Checksum	xx	Contains the XOR checksum of the SOF, Length, Command and the Data-bytes.
EOF (End of Frame)	<CR>	EOF byte marks the end of the command.

If the change message contains no ID information (length byte < 4) the last ID setting remains. In this case byte 1, bit 2 is ignored.

The usage as extended command is possible. (→ Extended Command Format (Byte Mode) 7)

2.6.2 Change / request receive channel mask settings

Byte(s)	Value	Description
SOF (Start of Frame)	'C'	SOF marks the beginning of the command.
Length	xx	The length byte contains the number of data bytes + the number of command bytes following.
Command	0x52 0x53	Request receive channel mask settings (needs one additional data byte = channel number) Change receive channel mask settings (needs at least two additional data bytes, the channel number and the channel settings)
Data	0x00-0xFF 0x00-0x07	Byte 0: Receive object number (range depends on gateway device) 0…15 Byte 1: bit 0: 0 = channel disable, 1 = channel enable bit 1: 0 = receive 11 bit mask, 1 = receive 29 bit mask bit 2: 0 = change 11 bit mask, 1 = change 29 bit mask Byte 2-3: 11 bit mask if byte1, bit 2 = 0 Byte 2-5: 29 bit mask if byte1, bit 2 = 1
Checksum	xx	Contains the XOR checksum of the SOF, Length, Command and the Data-bytes.
EOF (End of Frame)	<CR>	EOF byte marks the end of the command.

If the change message contains no mask information (length byte < 4) the last mask setting remains. In this case byte 1, bit 2 is ignored.

The usage as extended command is possible. (→ Extended Command Format (Byte Mode) 🗋 7)

2.6.3 CAN baud rate

Byte(s)	Value	Description
SOF (Start of Frame)	'C'	SOF marks the beginning of the command.
Length	xx	The length byte contains the number of data bytes + the number of command bytes following.
Command	0x56 0x57	Request CAN baud rate parameters (no additional data bytes) Set CAN baud rate Parameters
Data	0x00-0xFF	Byte 0: 0xFF if no CIA conform standard baud rate, else: 0x00 : 10 kBit / sec 0x01 : 20 kBit / sec 0x02: 50 kBit / sec 0xFE: 100 kBit / sec 0x03 : 125 kBit / sec 0x04: 250 kBit / sec 0x05: 500 kBit / sec 0x06: 800 kBit / sec 0x07: 1 MBit / sec 0xFF: BTR values are used Byte 1: BTR0 Value (device dependent, only if Byte 0 = 0xFF) Byte 2: BTR1 Value (device dependent, only if Byte 0 = 0xFF) Byte 3: BTR2 Value (device dependent, only if Byte 0 = 0xFF) Byte 4: BTR3 Value (device dependent, only if Byte 0 = 0xFF)
Checksum	xx	Contains the XOR checksum of the SOF, Length, Command and the Data-bytes.
EOF (End of Frame)	<CR>	EOF byte marks the end of the command.

If data byte 0 is set to 0xFF, the BTR values are used to set nonstandard baud rates. If a standard baud rate is used, the BTR values are not necessary, if they are used anyway, they are ignored!

If you need to set a baud rate that is not supported by the standard settings of the device, please contact our support for further information: www.ifm.com

The usage as extended command is possible. (→ Extended Command Format (Byte Mode) 🗋 7)

2.6.4 CAN controller reset

Byte(s)	Value	Description
SOF (Start of Frame)	'C'	SOF marks the beginning of the command.
Length	xx	The length byte contains the number of data bytes + the number of command bytes following.
Command	0x58	CAN controller reset (no additional data bytes)

Byte(s)	Value	Description
Checksum	xx	Contains the XOR checksum of the SOF, Length, Command and the Data-bytes.
EOF (End of Frame)	<CR>	EOF byte marks the end of the command.

The "CAN Controller Reset" command performs a complete reset of the CAN controller, including the timestamp value, receive and transmit buffers.

The CAN controller reset also resets the error status of the device. For this reason, after the CAN controller reset, an error status message is generated, to update the error status.

The usage as extended command is possible. (→ Extended Command Format (Byte Mode) 7)

2.6.5 Set / request active / passive mode

Byte(s)	Value	Description
SOF (Start of Frame)	'C'	SOF marks the beginning of the command.
Length	xx	The length byte contains the number of data bytes + the number of command bytes following.
Command	0x5A 0x5B	Request active / passive mode (no additional data bytes) Set active passive mode
Data	0x00-0x01	0x00 device = active mode 0x01 device = passive mode
Checksum	xx	Contains the XOR checksum of the SOF, Length, Command and the Data-bytes.
EOF (End of Frame)	<CR>	EOF byte marks the end of the command.

The usage as extended command is possible. (→ Extended Command Format (Byte Mode) 7)

2.6.6 Change / request receive channel cycle time

Byte(s)	Value	Description
SOF (Start of Frame)	'C'	SOF marks the beginning of the command.
Length	xx	The length byte contains the number of data bytes + the number of command bytes following.
Command	0x5C 0x5D	Request CAN cycle time parameters (no additional data bytes) Set CAN cycle time parameters
Data	0x00-0xFF 0x00-0xFF	Byte 0: Receive object number (range depends on gateway device) 0…15 Byte 1: Cycle Time in 10 ms
Checksum	xx	Contains the XOR checksum of the start byte, the length byte, the command byte and the data bytes.

Byte(s)	Value	Description
EOF (End of Frame)	<CR>	EOF byte marks the end of the command.

The usage as extended command is possible. (→ Extended Command Format (Byte Mode) 7)

2.7 Host interface configuration messages

2.7.1 Feedback / output settings

Byte(s)	Value	Description
SOF (Start of Frame)	'C'	SOF marks the beginning of the command.
Length	xx	The length byte contains the number of data bytes + the number of command bytes following.
Command	0x60 0x61	Request feedback settings (no additional data bytes) Set feedback settings
Data	0x00-0x07	Byte 0: bit 0: 1 = CAN output on / 0 = CAN output off bit 1: 1 = Transmit can message feedback on / 0 = feedback off bit 2: 1 = Server feedback on / 0 = server feedback off bit 3: 1 = RS232 interface on / 0 = RS232 interface off
Checksum	xx	Contains the XOR checksum of the SOF, Length, Command and the Data-bytes.
EOF (End of Frame)	<CR>	EOF byte marks the end of the command.

If the serial output is disabled, the received CAN messages are stored in the RAM of the CAN device. If this buffer is full, a buffer overflow error is indicated. If the output is enabled, all stored messages are transmitted immediately to the host interface.

The usage as extended command is possible. (→ Extended Command Format (Byte Mode) 🗋 7)

2.8 Device parameter settings

2.8.1 Change / request Timestamp settings

Byte(s)	Value	Description
SOF (Start of Frame)	'C'	SOF marks the beginning of the command.
Length	xx	The length byte contains the number of data bytes + the number of command bytes following.
Command	0xA0 0xA1	Request timestamp settings (request: no additional data bytes) Set timestamp settings
Data	0x00-0x13	Byte 0: bit 0: 1 = Timestamp on, 0 = Timestamp off bit 1: 1 = Relative timestamp, 0 = Absolute timestamp bit 4: 1 = Error and Echo Feedback timestamp on 0 = Error and Echo Feedback timestamp off
Checksum	xx	Contains the XOR checksum of the SOF, Length, Command and the Data-bytes.
EOF (End of Frame)	<CR>	EOF byte marks the end of the command.

If the "Timestamp-Mode" is selected, a 32 bit counter (100 µs resolution) value is inserted after the last data byte of each received CAN message. All messages on the host interface side which contain a timestamp are in the correct chronological order.

If a timestamp should be inserted after the echoed messages ("Transmit Server Feedback" or "Send CAN Message Feedback") and the error messages, The 'Error and Echo Feedback timestamp' has to be activated additionally.

The difference between "relative-" and "absolute-mode" is that the counter value is reset after each received message in "relative-mode".

The "Error and Echo Feedback" timestamp is only transmitted if the timestamp mode is enabled.

The usage as extended command is possible. (→ Extended Command Format (Byte Mode) 7)

2.8.2 CANopen object dictionary access

Byte(s)	Value	Description
SOF (Start of Frame)	'C'	SOF marks the beginning of the command.
Length	xx	The length byte contains the number of data bytes + the number of command bytes following.

Byte(s)	Value	Description
Command	0xA5	Read / Write in the object dictionary
Data	0x00-0xFF	Request: Byte 0: Command: 0x01 write ,0x00 read (no additional data bytes) Byte 1-2: Index (LSB first) Byte 3: Sub-index additional data bytes: Data (LSB first) Response Byte 0: Result: 0x40 read success 0x80 read failure (no additional data bytes) 0x41 write success (no additional data bytes) 0x81 write failure (no additional data bytes) Byte 1-2: Index (LSB first) Byte 3: Sub-index additional data bytes: Data (LSB first)
Checksum	xx	Contains the XOR checksum of the SOF, Length, Command and the Data-bytes.
EOF (End of Frame)	<CR>	EOF byte marks the end of the command.

2.9 Hardware access commands

2.9.1 Reset device

Byte(s)	Value	Description
SOF (Start of Frame)	'C'	SOF marks the beginning of the command.
Length	xx	The length byte contains the number of data bytes + the number of command bytes following.
Command	0xC0	Set reset device
Data	--	
Checksum	xx	Contains the XOR checksum of the SOF, Length, Command and the Data-bytes.
EOF (End of Frame)	<CR>	EOF byte marks the end of the command.

The devices respond to this command with the Command Device ID (→ 12).

For some devices this command does not lead to a hardware reset since the communication to the host would be lost otherwise:

- CANwireless

2.9.2 Request analog channel value

Byte(s)	Value	Description
SOF (Start of Frame)	'C'	SOF marks the beginning of the command.
Length	xx	The length byte contains the number of data bytes + the number of command bytes following.
Command	0xC2	Request analog channel value (needs 1 additional data byte containing the channel number)
Data	0x00- 0x03 0x00- 0xFF	Byte 0 (channel number): 0x00: Supply voltage Byte 1: analog value (for channel 0…2, see description below)
Checksum	xx	Contains the XOR checksum of the SOF, Length, Command and the Data-bytes.
EOF (End of Frame)	<CR>	EOF byte marks the end of the command.

2.9.3 Set Communication password

Byte(s)	Value	Description
SOF (Start of Frame)	'C'	SOF marks the beginning of the command.
Length	xx	The length byte contains the number of data bytes + the number of command bytes following.
Command	0xC6	Request communication password check

Byte(s)	Value	Description
Data	xx	Command **Byte 0…n:** Password string (default: "GSMONLIN") Response **Byte 0:** 0 = password checked and OK If password check failed, there is no response from the device
Checksum	xx	Contains the XOR checksum of the SOF, Length, Command and the Data-bytes.
EOF (End of Frame)	<CR>	EOF byte marks the end of the command.

This message is very important to unlock the whole functionality of byte mode interface at CANremote GSM/UMTS. A session is only possible with a check of the correct communication password!

2.9.4 Reset Communication password

Byte(s)	Value	Description
SOF (Start of Frame)	'C'	SOF marks the beginning of the command.
Length	xx	The length byte contains the number of data bytes + the number of command bytes following.
Command	0xC7	Reset communication password check
Checksum	xx	Contains the XOR checksum of the SOF, Length, Command and the Data-bytes.
EOF (End of Frame)	<CR>	EOF byte marks the end of the command.

This message locks the byte mode interface. This command should be sent at the end of a communication session.

3 Commands and Devices

The table below gives a summary of all existing commands and indicates which device supports them.

The usability of a command for a device is marked with a "S", "E" or "S / E".

These acronyms stand for:

Acronym	Description
S	The device supports this command only in the standard command format (→ Command Format (Byte Mode) 🗋 6).
E	The device supports this command only in the extended command format (→ Extended Command Format (Byte Mode) 🗋 7).
S / E	The device supports this command in the standard command format and in the extended command format.
empty	The device does not support this command.

Message Type	Command	CR3130 CR3131	CR3132 CR3133	CR3146 CR3158
Process Data Messages (→ 🗋 8)	0x00	S / E	S / E	S
	0x01	S / E	S / E	S / E
	0x02	S / E	S / E	S / E
	0x03	S / E	S / E	S / E
	0x04	S / E	S / E	S / E
	0x05	S / E	S / E	S / E
	0x06	S / E	S / E	S / E
	0x07	S / E	S / E	S / E
	0x08			S / E
	0x09			
Feedback messages (→ 🗋 11)	0x20	S / E	S / E	S / E
	0x21	S / E	S / E	S / E
	0x22	S / E	S / E	S / E
	0x23	S / E	S / E	S / E
	0x24	S / E	S / E	S / E
	0x25	S / E	S / E	S / E
	0x26	S / E	S / E	S / E
	0x27	S / E	S / E	S / E
	0x28			
	0x29			
	0x2A			
	0x2B			
Diagnosis data messages (→ 🗋 12)	0x40	S	S	S
	0x41	S	S	S
	0x42	S / E	S / E	S / E
	0x43			
	0x44	S / E	S / E	S / E
	0x47			
	0x48	S / E	S / E	S / E
	0x49		S / E	S / E

Message Type	Command	CR3130 CR3131	CR3132 CR3133	CR3146 CR3158
CAN Controller setup messages (→ 🗋 15)	0x50	S / E	S / E	S / E
	0x51	S / E	S / E	S / E
	0x52	S / E	S / E	S / E
	0x53	S / E	S / E	S / E
	0x56	S / E	S / E	S / E
	0x57	S / E	S / E	S / E
	0x58	S / E		S / E
	0x59			
	0x5A	S / E		S / E
	0x5B	S / E		S / E
	0x5C	S / E	S / E	S / E
	0x5D	S / E	S / E	S / E
Host interface configuration messages (→ 🗋 19)	0x60	S / E	S / E	S / E
	0x61	S / E	S / E	S / E
Transmit server interface commands	0x80			
	0x81			
	0x82			
	0x83			
	0x84			
	0x85			
	0x86			
	0x87			
	0x88			
	0x89			
Device parameter settings (→ 🗋 20)	0xA0	S / E	S / E	S / E
	0xA1	S / E	S / E	S / E
	0xA2			
	0xA3			
	0xA4			
	0xA5	S	S	
Hardware access commands (→ 🗋 22)	0xC0	S	S	S
	0xC2	S		
	0xC4			
	0xC5			
	0xC6	S		S
	0xC7	S		S