MACH-AIRCAN1
Communication Protocol Specification

For firmware v1.0

MACH SYSTEMS s.r.o.
www.machsystems.cz

Changes

Date	Firmware Version	Change	Changed by
26.3.2025	1.0	Initial release	PK

Contents

1. Introduction 3
2. Communication Protocol 4
   2.1. Message Overview 5
   2.2. Error Codes 7
3. Message Specification 8
   3.1. General Response 8
   3.2. Error Response 8
   3.3. Device Messages 8
   3.4. Wi-Fi Messages 9
   3.5. LIN Messages 14
   3.6. CAN and CAN FD Messages 22
   3.7. IO Control Messages 44
   3.8. Miscellaneous Messages 46
4. Communication Examples 47
4.1. Device Settings 47
   4.2. LIN 48
   4.3. USB – CAN interface 48
   4.4. USB – CAN FD interface 49
   4.5. LIN Diagnostics 49
   4.6. CAN Diagnostics 50
5. Contact 51

1. Introduction

The MACH-AIRCAN1 is a versatile Wi-Fi and Bluetooth interface designed to serve as a wireless gateway for CAN(/FD) and LIN buses.
The wireless interface features two CAN channels with CAN FD support, one LIN channel, 2.4 GHz Wi-Fi with support for both access point and client modes, Bluetooth 5, a USB 2.0 port, an analogue input, and a digital output. Its embedded web server allows for easy configuration and real-time monitoring.
The device offers an open communication protocol via Wi-Fi, Bluetooth, and USB, providing seamless access to CAN(/FD) and LIN channels. It simplifies integration of wireless connectivity into existing systems. Additionally, two MACH-AIRCAN1 devices can be paired to create a CAN(/FD)-Wi-Fi-CAN(/FD) bi-directional bridge, enabling the connection of two separate CAN(/FD) networks through a wireless tunnel.
For custom applications, a free C-language SDK is available for firmware development. This enables user applications to run directly on the MACH-AIRCAN1, and to use all the device's peripherals.



The communication protocol allows to:

- Configure CAN(/FD) channels and transmit and receive frames
- Configure LIN channel and transmit and receive frames
- Transmit and receive LIN diagnostic messages over LIN transport protocol (ISO 17987-2 / LIN 2 Transport Layer)
- Transmit and receive CAN(/FD) diagnostic messages over CAN ISO-TP protocol (ISO 15765-2)
- Change the device's configuration
- Set up Wi-Fi configuration and read status
- Read analogue input
- Write digital output

2. Communication Protocol

The communication between the MACH-AIRCAN1 and a user system is based upon a binary protocol over Wi-Fi or USB. The same message structure is used for both directions - to and from the device.

The protocol consists of Start Byte, Message Id, Data length, Data bytes, Checksum, and End Byte.

USB configuration is fixed: Virtual COM port (VCP), 115200 Baud, 8 data bits, no parity, 1 stop bit.

Wi-Fi default: Access Point mode, SSID: MACH-AIRCAN1, Password: machsystems, IP address 192.168.1.100, subnet mask 255.255.255.0, port 8000
The server runs TCP/IP and UDP and its configuration can be reconfigured.
This can be changed directly from the protocol or, as there is a web server running, also via a web browser (we recommend Google Chrome).

STX (1B)	ID (1B)	DATALEN (2B)	DATA (X B)	CHECKSUM (1B)	ETX (1B)
0x02	Message ID	Number of data bytes (LSB first)	Data bytes Number of bytes = DATALEN	1-byte sum (e.g. mod 256) of ID, DATALEN and all DATA bytes	0x03

The rest of the documentation refers to DATA part only. The user is then responsible for encapsulating it with the rest of the protocol fields, namely STX, ID, DataLen, Checksum, and ETX.
Checksum calculation example:

STX	ID	DATELEN		DATA		CHECKSUM	ETX
0x02	0x01	0x02	0x00	0xFF	0x01	**0x03** = (0x01 + 0x02 + 0x00 + 0xFF + 0x01) mod 256	0x03

2.1. Message Overview

The following tables describes message of the communication protocol over Ethernet / USB.

Message ID	Name	Request Data Length (bytes)	Response Data Length (bytes)	Description
0x01	BOOT_UP	- (no request needed)	0B ACK	A notification that the gateway was powered up. Sent to USB only.
Product information				
0x11	READ_SN	0	4	Read device serial number
0x12	READ_HW_INFO	0	6	Read device HW info
0x13	READ_SW_INFO	0	2	Read device SW info
Device configuration				
0x14	DEVICE_RESET_CONFIGURATION	0	0B ACK	Restore the default device configuration (AP, Client, CAN and LIN) and save it to EEPROM
0x15	WIFI_CONNECTED	0	1	Check is device is connected to network in Client mode
0x16	WIFI_RSSI	0	1	Read the RSSI value in Client mode
0x17	WIFI_READ_IP_ADDRESS_AP	0	5	Read IP address and mask of AP
0x18	WIFI_WRITE_IP_ADDRESS_AP	5	0B ACK	Write IP address and mask of AP
0x19	DEVICE_READ_PORT	0	2	Read communication port
0x1A	DEVICE_WRITE_PORT	2	0B ACK	Write communication port
0x1B	WIFI_READ_MAC_ADDRESSES	0	18	Read MAC addresses
0x1C	WIFI_READ_DEFAULT_GW_AP	0	4	Read default gateway of AP
0x1D	WIFI_WRITE_DEFAULT_GW_AP	4	0B ACK	Write default gateway of AP
0x1E	WIFI_DHCP	1	0B ACK or 1	Enable / disable the DHCP in Client mode
0x1F	WIFI_MODE	1	0B ACK or 1	Set the Wi-Fi Access Point or Client mode
0x82	WIFI_READ_IP_ADDRESS_CLIENT	0	5	Read IP address of Client
0x83	WIFI_WRITE_IP_ADDRESS_CLIENT	5	0B ACK	Write IP address of Client
0x84	WIFI_READ_DEFAULT_GW_CLIENT	0	4	Read default gateway Client
0x85	WIFI_WRITE_DEFAULT_GW_CLIENT	4	0B ACK	Write default gateway Client
0x86	WIFI_READ_SSID_AP	0	1 to 32	Read SSID Access Point
0x87	WIFI_WRITE_SSID_AP	1 to 32	0B ACK	Write SSID Access Point
0x88	WIFI_READ_SSID_CLIENT	0	1 to 32	Read SSID Client
0x89	WIFI_WRITE_SSID_CLIENT	1 to 32	0B ACK	Write SSID Client
0x8A	WIFI_WRITE_PASSWORD_AP	8 to 63	0B ACK	Write password Client
0x8B	WIFI_WRITE_PASSWORD_CLIENT	8 to 63	0B ACK	Write password Access Point
LIN configuration				
0x20	LIN_WRITE_CONFIGURATION	1	0B ACK	Configure LIN channel
0x21	LIN_READ_CONFIGURATION	0	1	Read LIN channel configuration
0x22	LIN_SAVE_CONFIGURATION	0	0B ACK	Save LIN configuration to EEPROM
0x23	LIN_LOAD_CONFIGURATION	0	0B ACK	Load LIN configuration from EEPROM
0x24	LIN_DEFAULT_CONFIGURATION	0	0B ACK	Load LIN default configuration
0x30	LIN_START	0	0B ACK	Start LIN channel

0x31	LIN_STOP	0	0B ACK	Stop LIN channel
0x32	LIN_ECHO_CONF	1	0B ACK	Configure LIN echo
0x33	LIN_ERROR	N/A	2	LIN bus error
0x40	LIN_MASTER_RESPONSE_TX	3 to 10	0 to 10	Transmit LIN Header + Response *Available when the device is configured as a Master*
0x41	LIN_MASTER_REQUEST_TX	1	0B ACK	Transmit LIN Header *Available when the device is configured as a Master*
0x42	LIN_MASTER_REQUEST_RX	N/A	4 to 10	Receive Slave Response *Available when the device is configured as a Master*
0x45	LIN_DIAG_TIMING	4	0B ACK	Set LIN UDS timing
0x46	LIN_DIAG_REQUEST	1 to 20	0B ACK	Transmit LIN UDS request
0x47	LIN_DIAG_RESPONSE	-	1 to 20	Asynchronous LIN UDS response
0x50	LIN_SLAVE_RESPONSE_CONFIG	2 to 10	0B ACK	Configure Slave Response buffer
0x51	LIN_SLAVE_RESPONSE_TX	N/A	2 to 11	A Slave response transmitted onto the bus.
0x52	LIN_SLAVE_RESPONSE_ RX	N/A	2 to 11	A Slave response received from the bus.
0x53	LIN_EVENT	N/A	1	Event on LIN bus. Currently, the wake up signal only.
CAN configuration				
0x60	CAN_WRITE_CONFIGURATION	6	1B ACK	Configure CAN channel
0x61	CAN_WRITE_CONFIG_TIM	9	1B ACK	Configure CAN channel set time quanta
0x62	CAN_READ_CONFIGURATION	1	13	Read CAN channel configuration
0x63	CAN_SAVE_CONFIGURATION	1	1B ACK	Save CAN configuration to EEPROM
0x64	CAN_LOAD_CONFIGURATION	1	1B ACK	Load CAN configuration from EEPROM
0x65	CAN_DEFAULT_CONFIGURATION	1	1B ACK	Load CAN default configuration
0x66	CAN_ECHO_CONF	2	1B ACK	Enable/Disable Tx echo
0x67	CAN_START_CHANNEL	1	1B ACK	Start CAN channel
0x68	CAN_STOP_CHANNEL	1	1B ACK	Stop CAN channel
0x69	CAN_GET_TIMESTAMP	1	9	Get time in microsecond from startup of channel
0x6A	CAN_TRANSMIT_FRAME	5 to 71	1B ACK	Send CAN message
0x6B	CAN_RECEIVED_FRAME	N/A	13 to 79	Received CAN message
0x6C	CAN_ERROR_FRAME	N/A	10	Some error on CAN bus
0x6D	CAN_TUNNEL	2	1B ACK or 2	Enable / disable CAN tunnel
0x6E	CAN_TUNNEL_SETTINGS_READ	0	8	Read CAN tunnel setting
0x6F	CAN_TUNNEL_SETTINGS_WRITE	8	1B ACK	Write CAN tunnel setting
0x70	CAN_DIAG_RX_CONFIG	8	1B ACK	Set DoCAN RX configuration
0x71	CAN_DIAG_TX_CONFIG	8	1B ACK	Set DoCAN TX configuration
0x72	CAN_DIAG_CHANGE_RX_STATE	2	1B ACK	Enable / disable parsing of DoCAN messages and set $P2_{CAN_Client}$

0x73	CAN_DIAG_REQUEST	3 to 400	1B ACK	Transmit a diagnostic request
0x74	CAN_DIAG_RESPONSE	N/A	3 to 400	Diagnostic response was received
0x75	CAN_DIAG_TIMEOUT	N/A	2	Diagnostic response timeout
I/O control				
0xE0	IO_WRITE	1	0B ACK	Toggle digital output
0xE1	IO_READ	0	2	Read analog input
Miscellaneous				
0xFD	RESTART	0	0	Restart the device
0xFE	RESTART_BOOT	1	0	Restart gateway to USB
0xFF	GENERAL_ERROR	N/A	2 to 3	An error occurred, see Error Codes for description.

2.2. Error Codes

The following tables describes error codes

Error Code	Data length	Comment
Communication protocol errors Messages contain: Error Code and Message ID		
0xA0	2	Incorrect end byte on the Ethernet protocol
0xA1	2	Bad checksum on the protocol
0xA2	2	Unknown message ID
0xA3	2	Too large or incorrect data length
LIN bus errors Messages contain: Error Code, Message ID and Channel Number		
0xB3	2	Buffer Full
General Bus errors Messages contain: Error Code, Message ID and Channel Number		
0xF0	2-3	Configuration Error
0xF1	2-3	Channel running, channel should be stopped when is configure
0xF2	2-3	Invalid channel selected – index out of bounds
0xF3	2-3	Channel is not running
0xF4	2-3	Hardware FIFO is full (should not happen in normal operation)

3. Message Specification

3.1. General Response

Device responds with a message acknowledgment after receiving a valid message. The acknowledgment does not contain any data. If there is some problem, the response is Error Response.

3.2. Error Response

MessageId = 0xFF

Device responds with an error if the command could not be processed correctly. Message ID contains ID of the requesting message. If Channel number is relevant, error response is three-byte and last byte is number of the relevant channel.

Response:

DATA 0	DATA 1
Error code	Message ID

When Channel number is not relevant.

OR

DATA 0	DATA 1	DATA 2
Error code	Message ID	Channel number

When both Message ID and Channel number are relevant.

See the table above to determine which error message contains what information.

3.3. Device Messages

3.3.1.Device Serial Number

MessageId = 0x11

This command is used for reading device serial number.

Request:

No data

Response:

DATA 0 – DATA 3
Device serial number

Example S/N: 02030106

DATA 0	DATA 1	DATA 2	DATA 3
06	01	03	02

3.3.2.Device Hardware Information

MessageId = 0x12

This command is used for reading device hardware number.

Request:

No data

Response:

DATA 0 – DATA 5
Device hardware number

Example HW Info: 000400030002

DATA 0	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5

02	00	03	00	04	00

3.3.3. Device Software Information

MessageId = 0x13

This command is used for reading software version number.

Request:

No data

Response:

DATA 0	DATA 1
VERSION MINOR	VERSION MAJOR

3.3.4. Restore Default Configuration

MessageId = 0x14

Reset the default device configuration – IP address, mask, port, default gateway, Wi-Fi configuration, CAN and LIN settings. After issuing this command, you must restart the device for Wi-Fi and IP changes to apply. The CAN and LIN changes are applied immediately.

Request:

No data

Response:

No data

3.4. Wi-Fi Messages

3.4.1. Wi-Fi Connected

MessageId = 0x15 for read

This command shows if the device is connected to the network. If the Wi-Fi is in Client mode, the response is 1 when device is successfully connected to the network. If connection to network failed, the response is 0. In Access Point the response is 1 when Wi-Fi server runs correctly and 0 otherwise.

Request 0x15:

No data

Response:

DATA 0
Wi-Fi connected

3.4.2. Wi-Fi RSSI Client

MessageId = 0x16 for read

This command is for reading of RSSI in Wi-Fi client mode when the device in connected to the network when the device is not connected or in Access point mode is shows 0.

Request 0x16:

No data

Response:

DATA 0
Wi-Fi RSSI

3.4.3. Read and Write IP Address Configuration Access Point

MessageId = 0x17 for read, 0x18 for write

This command is for reading or writing device IP address and subnet mask in Access Point mode. IP address octets are in the order first to last, e.g. 192.168.1.100 is encoded as C0 A8 01 64. Mask is in the one number format, which determines how many non-zero bits there is. For example, 24

corresponds with mask 255.255.255.0 (1111 1111.1111 1111.1111 1111.0000 0000). After issuing this command, you must restart the device for changes to apply.
Request 0x18:

DATA 0 – DATA 3	DATA 4
IP address	Mask

Response:
No data
Response to ID 0x17 has same structure as request 0x18 above.

3.4.4.Read and Write Communication Port

MessageId = 0x19 for read, 0x1A for write
This command is for changing the application communication port. Default port is 8000. After issuing this command, you must restart the device for changes to apply. The communication port is common for both Access Point and Client modes.

Request 0x1A:

DATA 0 – DATA 1
Port

Response:
No data
Response to 0x19 has the same structure as request for 0x1A above.

3.4.5.Read MAC Address

MessageId = 0x1B
This command is for reading device MAC addresses. Each device has three unique MAC address which cannot be changed by the user. The first MAC address is used in Wi-Fi Access Point mode, second in Wi-Fi Client mode and the third MAC address is used for Bluetooth. MAC address octets are in the order first to last, e.g. A0:19:6E:C2:A5:FC is encoded as A0 19 6E C2 A5 FC.

Request 0x1B:
No data

Response:

DATA 0 – DATA 5	DATA 6 – DATA 11	DATA 12 – DATA 17
MAC address Access Point	MAC address Client	MAC address Bluetooth

3.4.6. Read and Write Default Gateway Access Point

MessageId = 0x1C for read, 0x1D for write

Used for reading / changing the default gateway in the Access Point mode. Value in default configuration is 0.0.0.0 (in standard environment, default gateway is not needed as it is assumed that communication is running in single network segment). IP address octets are in the order first to last, e.g. 192.168.1.100 is encoded as C0 A8 01 64.
After issuing this command, you must restart the device for changes to apply.

Request 0x1D:

DATA 0 – DATA 3
Default gateway IP address

Response:
No data

Response to 0x1C has the same structure as request for 0x1D above.

3.4.7. Read and Write DHCP Enable / Disable Client mode

MessageId = 0x1E

Reading and writing of DHCP client enable. This is only applied in Wi-Fi Client mode. I When DHCP client is enabled, issuing IP address, mask or default gateway read means reading the currently active network value. In the Access Point mode the DHCP server is always running and connected clients can obtain the IP configuration from it.
After issuing this command, you must restart the device for changes to apply.

Request for read:

DATA 0
0x0

Response:

DATA 0
DHCP enable 0x00 – Disabled 0x01 – Enabled

Request for write:

DATA 0
0x01 – disable 0x02 – enable

Response: No data

3.4.8. Read and Write Wi-Fi mode Access Point / Client

MessageId = 0x1F

Reading and writing of Wi-Fi mode. When the Wi-Fi mode is selected the values (IP address, SSID, password etc.) associated with this mode, are used for the Wi-Fi configuration.
After issuing this command, you must restart the device for changes to apply.

Request for read:

DATA 0

0x0

Response:

DATA 0
Wi-Fi mode 0x00 – Access Point 0x01 – Client

Request for write:

DATA 0
0x01 – Access Point 0x02 – Client

Response: No data

3.4.9.Read and Write IP Address Configuration Client

MessageId = 0x82 for read, 0x83 for write

This command is for reading or writing device IP address and subnet mask in Client mode. IP address octets are in the order first to last, e.g. 192.168.1.100 is encoded as C0 A8 01 64. Mask is in the one number format, which determines how many non-zero bits there is. For example, 24 corresponds with mask 255.255.255.0 (1111 1111.1111 1111.1111 1111.0000 0000). After issuing this command, you must restart the device for changes to apply.

Request 0x83:

DATA 0 – DATA 3	DATA 4
IP address	Mask

Response:

No data

Response to ID 0x82 has the same structure as request 0x83 above.

3.4.10. Read and Write Default Gateway Client

MessageId = 0x84 for read, 0x85 for write

Used for reading / changing the default gateway in the Client mode. Value in default configuration is 0.0.0.0 (in standard environment, default gateway is not needed as it is assumed that communication is running in single network segment). IP address octets are in the order first to last, e.g. 192.168.1.100 is encoded as C0 A8 01 64.

After issuing this command, you must restart the device for changes to apply.

Request 0x85:

DATA 0 – DATA 3
Default gateway IP address

Response:

No data

Response to 0x84 has the same structure as request for 0x85 above.

3.4.11. Read and Write Wi-Fi SSID Access Point

MessageId = 0x86 for read, 0x87 for write

Used for reading / changing the Wi-Fi SSID in the Access Point mode. Default SSID is “MACH-ARCAN1”.

After issuing this command, you must restart the device for changes to apply.

Request 0x87:

DATA 0 – DATA 3
Default gateway IP address

Response:
No data

Response to 0x86 has the same structure as request for 0x87 above.

3.4.12. Read and Write Wi-Fi SSID Client

MessageId = 0x88 for read, 0x89 for write
Used for reading / changing the Wi-Fi SSID in the Client mode. Default SSID is not set.
After issuing this command, you must restart the device for changes to apply.

Request 0x89:

DATA 0 – DATA n
Wi-Fi SSID

Response:
No data

Response to 0x88 has the same structure as request for 0x89 above.

3.4.13. Write Wi-Fi Password Access Point / Client

MessageId = 0x8A Access Point, 0x8B Client
Used for changing the Wi-Fi Password. The default password is “machsystems” for Access Point. For the Client mode the default value is not set. The minimum password length is 8 bytes. The maximum password length is 63 bytes. The password cannot be read.
After issuing this command, you must restart the device for changes to apply.

Request 0x8A, 0x8B:

DATA 0 – DATA n
Wi-Fi Password

Response:
No data

3.5. LIN Messages

The LIN gateway can act as:

- LIN Master
- LIN Slave
- LIN bus Sniffer (receives all LIN communication and forwards into onto the Ethernet / USB)

The gateway can be controlled by a binary protocol over Ethernet / USB. This allows the user to:

- Configure LIN channel (Master/Slave, Baud Rate)
- Transmit and Receive LIN frames
- Acts as a sniffer - all LIN communication is forwarded onto Ethernet / USB

The sniffing mode does not actively communicate over the LIN bus. Instead, it forwards all incoming LIN communication onto the Ethernet / USB port.

The LIN channel configuration can be stored into the internal EEPROM. The configuration is then automatically loaded on power-up.

The LIN Id is a 6-bit LIN message identifier with the range of 0 – 63 (0x00 – 0x3F). The *LIN Id* keyword used in the following chapters is **always** 6-bit, even though it is not explicitly stated in next chapters.

LIN Frame Naming Convention

LIN frame consists of a header and a response.
Header = Synch. Break + Sync. Field + Id Field
Response = Data bytes + Checksum

Name	Meaning
Master Response	a complete LIN frame which contains both header and response
Master Request	a LIN header only e.g. Master transmits a header and expects a Slave to answer by a Slave response
Slave Response	a Slave Response only e.g. Databytes + Checksum

The following table summarizes TX/RX possibilities of the gateway for both LIN Master and Slave mode:

LIN Message Action	LIN Mode	
	Master	**Slave**
Transmit Master Response	yes	
Transmit Master Request	yes	

Transmit Slave Response		yes
Receive Slave Response	yes	
Receive LIN Frame	yes	yes

When the gateway is configured as LIN Master, an internal 1 kOhm pull-up resistor between Vbat and LIN bus is automatically enabled.

3.5.1.LIN Channel Configuration

MessageId=0x20

This command is for configuration of LIN interface. The Enhanced Checksum can't be selected when **AMLR == 0**. The LIN interface supports three baud rates (9600, 10417 and 19200).

Request:

DATA 0
Configuration Register

Configuration Register:

bit 7							bit 0
TX_ECHO	CHECKSUM	AMLR	AUTOSTART	MODE1	MODE0	BAUD1	BAUD0

Bit 7 **TX echo**
1 – TX echo enabled
0 – TX echo disabled

Bit 6 **Checksum Type**
0 – Classical Checksum
1 – Enhanced Checksum (except for 0x3C and 0x3D identifiers)

Bit 5 **AMLR - Automatic Message Length Recognition**
0 – Message length is taken from LIN Id field (as defined in LIN v1.x)
1 – Message length is recognized automatically (variable datalength as defined in LIN v2.x)

Bit 4 **AutoStart**
0 – LIN channel is NOT automatically started on power-up
1 – LIN channel is automatically started on power-up

Bit 2..3 **Mode**
00 – Slave
01 – LIN Master
10 – Sniffing Mode
11 – Reserved

Bit 0..1 **Baud rate**
00 – Reserved
01 – 9600 Bd
10 – 19200 Bd
11 – 10417 Bd

Response:
No data
General error message in case of error. Gateway cannot be reconfigured.

Reasons for error: Wrong baud rate type selected.

Default configuration of gateway

- Master
- Enhanced checksum
- 19200 Baud
- Auto-start disabled
- Automatic Message Length Recognition

3.5.2.Read Configuration

MessageId=0x21

This command reads LIN interface configuration register.

Request:
No data

Type of response:

DATA 0
Configuration Register

3.5.3.Save Configuration

MessageId=0x22

This command saves LIN interface configuration register to EEPROM.

Request:
No data
Response:
No data

3.5.4.Load Configuration

MessageId=0x23

This command loads saved configuration register from EEPROM.

Request:
No data
Response:
No data

3.5.5.Default Configuration

MessageId=0x24

This command loads default interface parameters.

Request:
No data
Response:
No data

Default configuration of gateway

- Master

- Enhanced checksum
- 19200 Baud
- Auto-start disabled
- Automatic Message Length Recognition

3.5.6.Start LIN

This command starts LIN interface.
MessageId=0x30

Request:
No data
Response:
No data in case of successful LIN start, general error message in case of error.

3.5.7.Stop LIN

This command stops LIN interface.
MessageId=0x31

Request:
No data
Response:
No data

3.5.8.Configure Echo LIN

This command setup LIN echo. When TX echo is on, all transmitted messages will be echoed back. If is TX echo off, no TX message will be echoed back. Same with RX echo for received messages. Default setting is TX echo enabled; RX echo enabled.
MessageId=0x32

Request:

DATA 1
Echo configuration

bit 7							bit 0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	TXECHO	RXECHO

Bit 1 **TX Echo on/off**

0 – Echo Off

1 – Echo On (default)

Bit 0 **RX Echo on/off**

0 – Echo Off

1 – Echo On (default)

Response:
No data

3.5.9.LIN Error

MessageId = 0x33

This message is sent asynchronously when there is some error on LIN.

Response:

DATA 0	DATA 1
Error type	LIN ID related to error

Error type:

- 0: Checksum error
- 1: Bus error
- 2: Timeout overrun
- 3: Error data too long

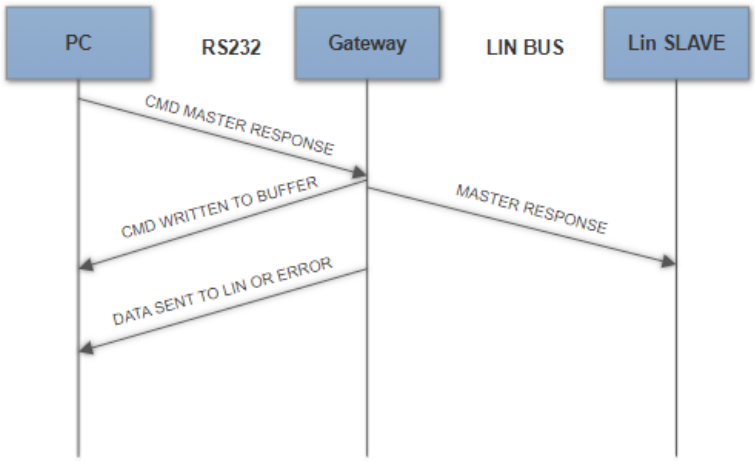
3.5.10. Transmit Master Response

MessageId=0x40

The gateway will transmit a LIN frame (both LIN Header and Response) onto the LIN bus.

Request:

DATA 0	DATA 1	DATA 2	DATA 3	DATA n
LIN ID	LIN DATALEN	LIN DATA 0	LIN DATA 1	DATA n



Response:
No data

If echo enabled after the data are sent the message with same data structure as request is send.

Error response: General error message or LIN Error message in case of error.
Reasons for error: Channel is in stop state or buffer is full.

3.5.11. Transmit Master Request

MessageId=0x41

The gateway will transmit a LIN Header onto the LIN bus and will expect a Slave to send a response.

Request:

DATA 0
SLAVE LIN ID

DATA0 – LIN ID of slave device

Response:
No data

If echo enabled after the data are sent the message with same data structure as request is send.

Error response:
General error message or LIN Error message in case of error.
Reasons for error: Request cannot be sent, or slave doesn't respond.

3.5.12. Receive Master Response

MessageId=0x42
The gateway received a Slave response onto previously sent Master request.

Response:

DATA 0	DATA 1	DATA 2	DATA n
LIN ID	LIN DATALEN	LIN DATA 0	LIN DATA n

Error response: General error message or LIN Error message in case of error. Request cannot be sent, or slave doesn't respond.

3.5.13. Set LIN Diagnostic Timing (UDS Transport)

MessageId = 0x45

Set values for P2 and STmin parameters. P2 is time between last diagnostic response and first request (basically delay for the slave to be able to provide response). STmin is a minimum time between consecutive diagnostic requests or diagnostic responses. Note that those values are specified in the slave device's LDF file. Both parameters are 16-bit millisecond values. This message can be issued any time.
Default values for both parameters are 50 ms.

DATA 0	DATA 1	DATA 2	DATA 3
STmin LSB	STmin MSB	P2 LSB	P2 MSB

Response:
No data

If data length is correct, this command always succeeds.

3.5.14. LIN Diagnostic Request (UDS Transport)

MessageId = 0x46

Create and send LIN UDS request on the bus. Request has LIN Id 0x3C. It may be transmitted using multiple LIN frames depending on data length of the request. Device then automatically polls the slave for response on LIN Id 0x3D.
Note that this command can be used only when LIN is started and when device is configured as LIN Master.

DATA 0	DATA 1	DATA 2	…	DATA N
Device NAD	SID	UDS Data 1	…	UDS Data N – 1

- Device NAD: UDS NAD of the addressed device. This is also used to match the device response with the original request.
- SID: Service Identifier (e.g., 0x22 for Read Data by Identifier)
- UDS Data 0 … N – 1: UDS data to be sent on the bus

Data length does not have to be specified as it is determined from the protocol message length.

Response:
No data

Error response: General error message or LIN Error message in case of error

3.5.15. LIN Diagnostic Response (UDS Tranport)

MessageId = 0x47

Transmitted asynchronously when slave UDS response on LIN Id 0x3D was received. Response can span multiple LIN frames depending on data length. Note that our device listens only for NAD that was specified in the request message. When no slave responds, LIN error frame is echoed.

Format:

DATA 0	DATA 1	DATA 2	…	DATA N
UDS NAD	RSID	Raw UDS Data 0	…	Raw UDS Data N – 2

3.5.16. Slave Response Configuration

MessageId=0x50

When the gateway is configured as LIN Slave, it provides message buffers for Slave Responses. These message buffers can be used for both direction - transmission and reception of Slave Response.

The following describes how the message buffers can be set up.

Request:

DATA 0	DATA 1	DATA 2	DATA 3	DATA n
BUFFER CONFIG	LIN DATALEN	LIN DATA 0	LIN DATA 1	LIN DATA n

- LIN ID has to be unique
- Slave response buffers are empty after channel start
- LIN DATALEN can be up to 8
- LIN DATA bytes are present for TX direction only

DATA 0 – BUFFER CONFIG:

bit 7			bit 0
Reserved	BUFFER DIR	LIN ID	

Bit 7 Reserved

Bit 6 **Buffer Direction**

0 – RX, the gateway will receive data from LIN frame into this buffer. Length of received message depends on AMLR bit from configuration register. If AMLR bit is 0, message length is hardcoded in LIN ID, otherwise length of the message is recognized automatically. See 3.5.18 for notification about this event.

1 – TX, the gateway will transmit a Slave Response when the corresponding LIN Id is pooled by the Master. See 3.5.18 for notification about this event.

Bit 0..5 LIN ID (including message length coding if AMLR bit is 0)

DATA 1 – LIN DATALEN:

Number of LIN data bytes (0-8)

DATA 2, 3, 4… - LIN DATA (message data bytes)

The data bytes shall only be present for TX Buffer Direction.

Slave response buffer configuration example:

DATA 0		**DATA 1**	**DATA 2, DATA 3, DATA 4, DATA n**	**Comment:**
BUFFER DIR + LIN ID		*LIN DataLen*	*LIN Data Bytes*	
1	0x01	4	0x01, 0x02, 0x03, 0x04	This data will be sent to LIN BUS if slave receives MASTER REQUEST with LIN ID 0x01
0	0x02	0	-	Slave will receive data from master and send them to PC.
1	0x05	2	0x22,0x23	This data will be sent to LIN BUS if slave receives MASTER REQUEST with ID 0x05

Response:
Data written to the Slave Response buffer
No data

Error response: General error message in case of error. Data cannot be written to the Slave Response buffer
Reasons for error: Buffer Direction is 1 (TX) but the DataLen is 0.

3.5.17. Slave Response TX

MessageId=0x51

When the gateway is configured as a Slave and it transmits a Slave Response onto the bus, it sends a notification to the user, if TX echo is enabled.

Response:
Transmitted Slave Response onto the bus

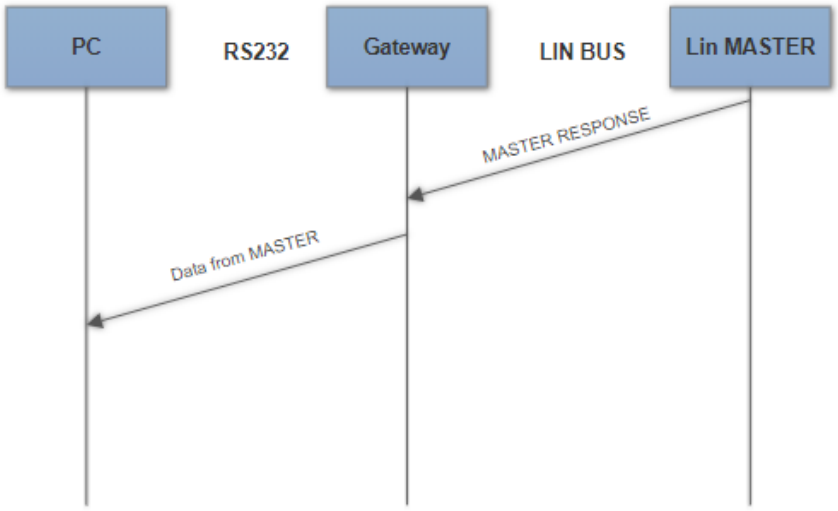
DATA 0	**DATA 1**	**DATA 2**	**DATA n**
LIN ID	LIN DATALEN	LIN DATA 0	LIN DATA n

Error response: General error message or LIN Error message in case of error.

3.5.18. Slave Response RX

MessageId=0x52

When the gateway is configured as a Slave and it receives a Master Response from the bus, it sends a



notification to the user, if RX echo is enabled.

Response:

Received data from Slave Response from the bus

DATA 0	DATA 1	DATA 2	DATA n
LIN ID	LIN DATALEN	LIN DATA 0	LIN DATA n

Error response: General error message or LIN Error message in case of error.

3.5.19. LIN Event

MessageId=0x53

When the gateway is configured as a Master, it sends a notification to the user when event occur.

Response:

Transmitted Response onto the bus

DATA 0
LIN EVENT

DATA 0	LIN event
0x0	LIN wake up signal
0x1 – 0xFF	Reserved

3.6. CAN and CAN FD Messages

3.6.1.Channel Configuration

MessageId=0x60

This message configures a CAN(/FD) channel. The time quanta for CAN FD controller are chosen by given sample point and baud rate. Sample point can be set exactly for baud rates up to 2 MBd. For 4 MBd, sample point is rounded to nearest lower multiple of 5 %; for 8 MBd, sample point is rounded to nearest lower multiple of 10 %. Note that for Data baud rate of 8 MBaud, Arbitration baud rate 1 MBaud should be used. The actual time quanta setting can be obtained by **Read Configuration** command. The CAN FD controller clock is 80 MHz.

Request:
Data section of a packet:

DATA 0	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5
Channel and SAVE bit	Configuration Register 1	Configuration Register 2	Configuration Register 3	Configuration Register 4 (**CAN FD mode**)	Configuration Register 5 (**CAN FD mode**)

Channel number and SAVE bit:

bit 7							
SAVE	Reserved	Reserved	Reserved	Reserved	Reserved	CHNL1	CHNL0

Bit 7 **Save**
Flag to tell if configuration will be saved right away.
0 – Do not save the configuration
1 – Store device configuration to EEPROM immediately after reconfiguration.

Bit 0..1 **Channel**
00 – CAN 1
01 – CAN 2
10 – Reserved
11 – Reserved

Configuration CHANNEL N Register 1:

bit 7							bit 0
PROTOCOL1	PROTOCOL0	AUTOSTART	ACK	ASP3	ASP2	ASP1	ASP0

Bit 6..7 **Protocol**
00 – CAN 2.0B
01 – ISO CAN FD
10 – Reserved
11 – Reserved

Bit 5 **AutoStart**
0 – CAN channel is NOT automatically started on power-up
1 – CAN channel is automatically started on power-up

Bit 4 **Acknowledge mode**
0 – Normal mode
1 – Silent mode

Bit 0..3 **Arbitration Sample Point**
0000 – 60%
0001 – 62,5%
0010 – 65%
0011 – 67,50%
0100 – 70%
0101 – 72,50%
0110 – 75%
0111 – 77,50%

1000 – 80%
1001 – 82,50%
1010 – 85%
1011 – 87,50%
1100 – 90%
1101 – Reserved
1110 – Reserved
1111 – Reserved

Configuration CHANNEL N Register 2:

bit 7							bit 0
Reserved	Reserved	Reserved	Reserved	Reserved	ABAUD2	ABAUD1	ABAUD0

Bit 7..3 **Reserved**

Bit 0..2 **Arbitration baud rate**
000 – 125 kBd
001 – 250 kBd
010 – 500 kBd
011 – 1 MBd
100..111 – Reserved

Configuration CHANNEL N Register 3:

bit 7							bit 0
Reserved	ASJW6	ASJW5	ASJW4	ASJW3	ASJW2	ASJW1	ASJW0

Bit 7 **Reserved**

Bit 0..6 **Arbitration jump width**
0000000 – 1
0000001 – 2
0000010 – 3
0000011 – 4
…
1111111 – 128

Configuration CHANNEL N Register 4 (**relevant for CAN FD mode only**):

bit 7							bit 0
Reserved	DBAUD2	DBAUD1	DBAUD0	DSJW3	DSJW2	DSJW1	DSJW0

Bit 4..6 **Data baud rate**
000 – 1 MBd
001 – 2 MBd
010 – 4 MBd
011 – 8 MBd
100..111 – Reserved

Bit 0..3 **Data Synchronization jump width**

0000 – 1
0001 – 2
0010 – 3
0011 – 4
…
1111 – 16

Configuration CHANNEL N Register 5 (**relevant for CAN FD mode only**):

bit 7							bit 0
Reserved	Reserved	Reserved	Reserved	DSP3	DSP2	DSP1	DSP0

Bit 0..3 **Data Sample Point**

0000 – 60%
0001 – 62,5%
0010 – 65%
0011 – 67,50%
0100 – 70%
0101 – 72,50%
0110 – 75%
0111 – 77,50%
1000 – 80%
1001 – 82,50%
1010 – 85%
1011 – 87,50%
1100 – 90%
1101 – Reserved
1110 – Reserved
1111 – Reserved

Response:

DATA 0
Channel number

Possibilities for error: CAN channel cannot be reconfigured - wrong arbitration or data jump width. Acknowledgment when configuration was loaded, general error message in case of error.

Default configuration

Channel 1

- ISO CAN FD
- Normal mode
- Arbitration speed 500 kBd
- Arbitration SJW 8
- Arbitration Sample Point 80%
- Data speed 2 MBd
- Data SJW 4
- Data Sample Point 80 %
- Autostart disable

Channel 2

- ISO CAN FD
- Normal mode
- Arbitration speed 500 kBd
- Arbitration SJW 8
- Arbitration Sample Point 80%
- Data speed 2 MBd
- Data SJW 4
- Data Sample Point 80 %
- Autostart disable

3.6.2.Channel Configuration Time Quanta Timing

MessageId=0x61

Same as channel configuration, except in this case also exact time quanta sizes are set.

Request:
Data section of a packet:

DATA 0	**DATA 1**	**DATA 2**	**DATA 3**	**DATA 4**
Channel and SAVE bit	Configuration Register 1	Configuration Register 6	Configuration Register 7	Configuration Register 8
DATA 5	**DATA 6**	**DATA 7**	**DATA 8**	
Configuration Register 9	Configuration Register 10 **(CAN FD mode)**	Configuration Register 11 **(CAN FD mode)**	Configuration Register 12 **(CAN FD mode)**	

Channel number and SAVE bit:

bit 7							
SAVE	Reserved	Reserved	Reserved	Reserved	Reserved	CHNL1	CHNL0

Bit 7 **Save**
Flag to tell if configuration will be saved right away.
0 – Do not save the configuration
1 – Store device configuration to EEPROM immediately after reconfiguration.

Bit 0..1 **Channel**
00 – CAN 1
01 – CAN 2
10 – Reserved
11 – Reserved

Configuration CHANNEL N Register 1:

bit 7							bit 0
PROTOCOL1	PROTOCOL0	AUTOSTART	ACK	Reserved	Reserved	Reserved	Reserved

Bit 6..7 **Protocol**
00 – CAN 2.0B
01 – ISO CAN FD

10 – Reserved
11 – Reserved

Bit 5 **AutoStart**
0 – CAN channel is NOT automatically started on power-up
1 – CAN channel is automatically started on power-up

Bit 4 **Acknowledge mode**
0 – Normal mode
1 – Silent mode

Bit 0..3 **Reserved**

Configuration CHANNEL N Register 6:

bit 7							bit 0
ATSEG1_7	ATSEG1_6	ATSEG1_5	ATSEG1_4	ATSEG1_3	ATSEG1_2	ATSEG1_1	ATSEG1_0

Bit 0..7 **Arbitration time segment 1**
0000 0000 – 1
0000 0001 – 2
0000 0010 – 2
0000 0011 – 3
…
1111 1111 – 256

Configuration CHANNEL N Register 7:

bit 7							bit 0
Reserved	ATSEG2_6	ATSEG2_5	ATSEG2_4	ATSEG2_3	ATSEG2_2	ATSEG2_1	ATSEG2_0

Bit 7 **Reserved**

Bit 0..6 **Arbitration time segment 2**
000 0000 – 1
000 0001 – 2
000 0010 – 3
000 0011 – 4
…
111 1111 – 128

Configuration CHANNEL N Register 8:

bit 7							bit 0
APRESC_7	APRESC_6	APRESC_5	APRESC_4	APRESC_3	APRESC_2	APRESC_1	APRESC_0

Bit 0..7 **Arbitration prescaler**
0000 0000 – 1
0000 0001 – 2
0000 0010 – 2
0000 0011 – 3
…

1111 1111 – 256

Configuration CHANNEL N Register 9:

bit 7							bit 0
Reserved	ASJW6	ASJW5	ASJW4	ASJW3	ASJW2	ASJW1	ASJW0

Bit 7 **Reserved**

Bit 0..6 **Arbitration jump width**
000 0000 – 1
000 0001 – 2
000 0010 – 3
000 0011 – 4
…
111 1111 – 128

Configuration CHANNEL N Register 10 (**relevant for CAN FD mode only**):

bit 7							bit 0
Reserved	Reserved	Reserved	DTSEG1_4	DTSEG1_3	DTSEG1_2	DTSEG1_1	DTSEG1_0

Bit 5..7 **Reserved**

Bit 0..4 **Data time segment 1**
0 0000 – 1
0 0001 – 2
0 0010 – 2
0 0011 – 3
…
1 1111 – 32

Configuration CHANNEL N Register 11 (**relevant for CAN FD mode only**):

bit 7							bit 0
DSJW3	DSJW2	DSJW1	DSJW0	DTSEG2_3	DTSEG2_2	DTSEG2_1	DTSEG2_0

Bit 4..7 **Data Synchronization jump width**
0000 – 1
0001 – 2
0010 – 3
0011 – 4
…
1111 – 16

Bit 0..3 **Data time segment 2**
0000 – 1
0001 – 2
0010 – 3
0011 – 4

…
1111 – 16

Configuration CHANNEL N Register 12 (**relevant for CAN FD mode only**):

bit 7							bit 0
Reserved	Reserved	Reserved	DPRESC4	DPRESC3	DPRESC2	DPRESC1	DPRESC0

Bit 5..7 **Reserved**

Bit 0..4 **Data prescaler**
0 0000 – 1
0 0001 – 2
0 0010 – 3
0 0011 – 4
…
1 1111 – 32

DATA 0
Channel number

Acknowledgment when configuration was loaded, general error message in case of error.
Possibilities for error: CAN channel cannot be reconfigured - wrong arbitration or data jump width

3.6.3.Read Configuration

This command reads CAN interface settings. If configuration is set by precise timing message, 0xF values are set instead of Sample point and Baud rate values.

MessageId=0x62

Request:

DATA 0
Channel number

Bit 0..1 **Channel**
00 – CAN 1
01 – CAN 2
10 – Reserved
11 – Reserved

Response:

DATA 0	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5
Channel	Configuration Register 1	Configuration Register 2	Configuration Register 3	Configuration Register 6	Configuration Register 7
DATA 6	**DATA 7**	**DATA 8**	**DATA 9**	**DATA 10**	**DATA 11**
Configuration Register 8	Configuration Register 4 **(CAN FD mode)**	Configuration Register 5 **(CAN FD mode)**	Configuration Register 10 **(CAN FD mode)**	Configuration Register 11 **(CAN FD mode)**	Configuration Register 12 **(CAN FD mode)**
DATA 12					
Echo configuration Register 13					

Channel number

bit 7							
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	CHNL1	CHNL0

Bit 2..7 **Reserved**

Bit 0..1 **Channel**
00 – CAN 1
01 – CAN 2
10 – Reserved
11 – Reserved

Configuration CHANNEL N Register 1:

bit 7							bit 0
PROTOCOL1	PROTOCOL0	AUTOSTART	ACK	ASP3	ASP2	ASP1	ASP0

Bit 6..7 **Protocol**
00 – CAN
01 – ISO CAN FD
10 – Reserved
11 – Reserved

Bit 5 **AutoStart**
0 – CAN channel is NOT automatically started on power-up
1 – CAN channel is automatically started on power-up

Bit 4 **Acknowledge mode**
0 – Normal mode
1 – Silent mode

Bit 0..3 **Arbitration sample point**
0000 – 60%
0001 – 62,5%
0010 – 65%
0011 – 67,50%
0100 – 70%
0101 – 72,50%
0110 – 75%
0111 – 77,50%
1000 – 80%
1001 – 82,50%
1010 – 85%
1011 – 87,50%
1100 – 90%
1101 – Reserved
1110 – Reserved
1111 – Reserved

Configuration CHANNEL N Register 2:

bit 7							bit 0
Reserved	Reserved	Reserved	Reserved	Reserved	ABAUD2	ABAUD1	ABAUD0

Bit 3..7 **Reserved**

Bit 0..2 **Arbitration baud rate**
000 – 125 kBd
001 – 250 kBd
010 – 500 kBd
011 – 1 MBd
100..111 – Reserved

Configuration CHANNEL N Register 3:

bit 7							bit 0
Reserved	ASJW6	ASJW5	ASJW4	ASJW3	ASJW2	ASJW1	ASJW0

Bit 7 **Reserved**

Bit 0..6 **Arbitration jump width**
000 0000 – 1
000 0001 – 2
000 0010 – 3
000 0011 – 4
…
111 1111 – 128

Configuration CHANNEL N Register 6:

bit 7							bit 0
ATSEG1_7	ATSEG1_6	ATSEG1_5	ATSEG1_4	ATSEG1_3	ATSEG1_2	ATSEG1_1	ATSEG1_0

Bit 0..7 **Arbitration time segment 1**
0000 0000 – 1
0000 0001 – 2
0000 0010 – 2
0000 0011 – 3
…
1111 1111 – 256

Configuration CHANNEL N Register 7:

bit 7							bit 0
Reserved	ATSEG2_6	ATSEG2_5	ATSEG2_4	ATSEG2_3	ATSEG2_2	ATSEG2_1	ATSEG2_0

Bit 7 **Reserved**

Bit 0..6 **Arbitration time segment 2**
000 0000 – 1
000 0001 – 2
000 0010 – 3
000 0011 – 4

…
111 1111 – 128

Configuration CHANNEL N Register 8:

bit 7							bit 0
APRESC_7	APRESC_6	APRESC_5	APRESC_4	APRESC_3	APRESC_2	APRESC_1	APRESC_0

Bit 0..7 **Arbitration prescaler**
0000 0000 – 1
0000 0001 – 2
0000 0010 – 2
0000 0011 – 3
…
1111 1111 – 256

Configuration CHANNEL N Register 4 (**relevant for CAN FD mode only)**:

bit 7							bit 0
Reserved	DBAUD2	DBAUD1	DBAUD0	DSJW3	DSJW2	DSJW1	DSJW0

Bit 7 **Reserved**

Bit 4..6 **Data baud rate**
000 – 1 MBd
001 – 2 MBd
010 – 4 MBd
011 – 8 MBd
100..111 – Reserved

Bit 0..3 **Data Synchronization jump width**
0000 – 1
0001 – 2
0010 – 3
0011 – 4
…
1111 – 16

Configuration CHANNEL N Register 5 (**relevant for CAN FD mode only)**:

bit 7							bit 0
Reserved	Reserved	Reserved	Reserved	DSP3	DSP2	DSP1	DSP0

Bit 4..7 **Reserved**

Bit 0..3 **Data sample point**
0000 – 60%
0001 – 62,5%
0010 – 65%
0011 – 67,50%
0100 – 70%
0101 – 72,50%
0110 – 75%
0111 – 77,50%
1000 – 80%

1001 – 82,50%
1010 – 85%
1011 – 87,50%
1100 – 90%
1101 – Reserved
1110 – Reserved
1111 – Reserved

Configuration CHANNEL N Register 10 (**relevant for CAN FD mode only**):

bit 7							bit 0
Reserved	Reserved	Reserved	DTSEG1_4	DTSEG1_3	DTSEG1_2	DTSEG1_1	DTSEG1_0

Bit 5..7 **Reserved**

Bit 0..4 **Data time segment 1**
0 0000 – 1
0 0001 – 2
0 0010 – 2
0 0011 – 3
…
1 1111 – 32

Configuration CHANNEL N Register 11 (**relevant for CAN FD mode only**):

bit 7							bit 0
Reserved	Reserved	Reserved	Reserved	DTSEG2_3	DTSEG2_2	DTSEG2_1	DTSEG2_0

Bit 4..7 **Reserved**

Bit 0..3 **Data time segment 2**
0000 – 1
0001 – 2
0010 – 3
0011 – 4
…
1111 – 16

Configuration CHANNEL N Register 12 (**relevant for CAN FD mode only**):

bit 7							bit 0
Reserved	Reserved	Reserved	DPRESC4	DPRESC3	DPRESC2	DPRESC1	DPRESC0

Bit 5..7 **Reserved**

Bit 0..4 **Data prescaler**
0 0000 – 1
0 0001 – 2
0 0010 – 3
0 0011 – 4
…
1 1111 – 32

Echo configuration CHANNEL N register 13:

bit 7							bit 0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	TXECHO	RXECHO

Bit 2..7 **Reserved**

Bit 1 **TX Echo on/off**

0 – Echo Off

1 – Echo On (default)

Bit 0 **RX Echo on/off**

0 – Echo Off

1 – Echo On (default)

3.6.4.Save Configuration

MessageId=0x63

Saves CAN configuration to the non-volatile memory.

Request 0x63:

DATA 0
Channel number

Bit 0..1	**Channel**
	00 – CAN 1
	01 – CAN 2
	10 – Reserved
	11 – Reserved

Response:

DATA 0
Channel number

Acknowledgment when configuration was saved, general error message in case of error.

3.6.5.Load Configuration

MessageId=0x64

Request 0x64:

DATA 0
Channel number

Bit 0..1	**Channel**
	00 – CAN 1
	01 – CAN 2
	10 – Reserved
	11 – Reserved

Response:

DATA 0
Channel number

Acknowledgment when configuration was loaded, general error message in case of error.

3.6.6.Default Configuration

MessageId=0x65

Apply CAN default configuration. For default configuration values see 3.6.1 Channel Configuration.

Request 0x65:

DATA 0
Channel number

Bit 0..1	**Channel**
	00 – CAN 1
	01 – CAN 2
	10 – Reserved
	11 – Reserved

Response:

DATA 0
Channel number

Acknowledgment when configuration was loaded, general error message in case of error.

Reasons for error: wrong channel selected, CAN channel is already running.

3.6.7.Frame Echo Configuration

MessageId=0x66

Request 0x66:

DATA 0	DATA 1
Channel number	Echo configuration

bit 7							bit 0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	TXECHO	RXECHO

Bit 1 **TX Echo on/off**

0 – Echo Off

1 – Echo On (default)

Bit 0 **RX Echo on/off**

0 – Echo Off

1 – Echo On (default)

Response:

DATA 0
Channel number

Acknowledgment when echo configuration was changed, general error message in case of error.
Reasons for error: wrong channel selected, CAN channel is already running.

3.6.8.Start Channel

MessageId=0x67

Request 0x67:

DATA 0
Channel number

DATA 0

Bit 0..7 **Channel**

0x0 – CAN 1
0x1 – CAN 2
0x2 – Reserved
…
0xFE – Reserved
0xFF – All Channels

Response:

DATA 0
Channel number

Acknowledgment when channel was successfully stopped, error message when error occurred (for example wrong channel number).

Note: if all channels are selected for starting and one of the channels is running, no error is returned. Otherwise, if start of a running channel is requested, it is considered an error.

3.6.9.Stop Channel

MessageId=0x68

Stop CAN channel.

Request:

DATA 0
Channel number

Bit 0..7 **Channel**
0x0 – CAN 1
0x1 – CAN 2
0x2 – Reserved
…
0xFE – Reserved
0xFF – All Channels

Response:

DATA 0
Channel number

Acknowledgment when channel was successfully stopped, error message when error occurred (for example wrong channel number).

3.6.10. Get Channel Timestamp

MessageId=0x69

Timestamp is 64bit number that represent the time from startup of CAN(/FD) channel in microseconds. The bit order in message is LSB.

Request:

DATA 0
Channel number

Bit 0..7 **Channel**
0x0 – CAN 1
0x1 – CAN 2
0x2 – Reserved
…
0xFE – Reserved
0xFF – All Channels

Response:

DATA 0	DATA 1…8
Channel number	Timestamp byte 0…7

If all channels option is set

Response:

DATA 0	DATA 1…8	DATA 9	DATA 10…17
Channel 0	Timestamp byte 0…7	Channel 1	Timestamp byte 0…7

3.6.11. Transmit Frame

MessageId=0x6A

This message transmits CAN frame. The structure of frame is different when Extended ID is set. Without extended ID frame is header is 5 bytes long. With Extended ID is 7 bytes long. The format of ID is LSB.

Request:
IF EXTId==0

DATA 0	DATA 1	DATA 2	DATA 3	DATA 4	DATA n
Channel number	MESSAGE_INFO	ID0	ID1	DLC	DATA

IF EXTId==1

DATA 0	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5	DATA 6	DATA n
Channel number	MESSAGE_INFO	ID0	ID1	ID2	ID3	DLC	DATA

Channel:

Bit 0..1 **Channel**
00 – CAN 1
01 – CAN 2
10 – Reserved
11 – Reserved

MESSAGE_INFO:

bit 7							bit 0
Reserved	Reserved	Reserved	FDF	ESI	BRS	RTR	EXTId

Bit 5..7 **Reserved**

Bit 4 **FDF**
0 – Frame transmitted in Classic CAN format
1 – Frame transmitted in FDCAN format

Bit 3 **ESI**
0 – Transmitting node is error active
1 – Transmitting node is error passive

Bit 2 **BRS**
0 – FDCAN frames transmitted/received without bit rate switching
1 – FDCAN frames transmitted/received with bit rate switching

Bit 1 **RTR**
0 – Data frame
1 – Remote frame

Bit 0 **EXTId**
0 – Standard ID
1 – Extended ID

Timestamp

Timestamp is 64bit number that represents the time from startup of CAN(/FD) channel in microseconds. The bit order in message is LSB.

Response:

DATA 0
Channel number

Acknowledgment if the frame was successfully passed to the controller for transmission, general error message in case of error.
Possible reasons for error: wrong bit configuration.

IF the TX echo is enabled this message is received after the CAN frame is transmitted (EXTId==0):

DATA 0	DATA 1	DATA 2...9	DATA 10	DATA 11	DATA 12	DATA 13 - n
Channel number	MESSAGE_INFO	Timestamp byte 0…7	ID0	ID1	DLC	DATA

IF the TX echo is enabled this message is received after the CAN frame is transmitted (EXTId==1):

DATA 0	DATA 1	DATA 2...9	DATA 10	DATA 11	DATA 12	DATA 13
Channel number	MESSAGE_INFO	Timestamp byte 0…7	ID0	ID1	ID2	ID3
DATA 14	**DATA 15 - n**					
DLC	DATA					

3.6.12. Receive Frame

MessageId=0x6B

Message response has similar structure as Transmit Frame. It only differs with added timestamp bytes (bytes 2…9). Timestamp is 64bit number that represents the time from startup of CAN(/FD) channel in microseconds. The bit order in message is LSB. For this message no request is needed, it appears when frame from other CAN unit arrived.). Upon reception of CAN error frame, protocol error frame message is sent 0
CAN Error Frame.

Response:
IF EXTId==0

DATA 0	DATA 1	DATA 2...9	DATA 10	DATA 11	DATA 12	DATA 15 - n
Channel number	MESSAGE_INFO	Timestamp byte 0…7	ID0	ID1	DLC	DATA

IF EXTId==1

DATA 0	DATA 1	DATA 2...9	DATA 10	DATA 11
Channel number	MESSAGE_INFO	Timestamp byte 0…7	ID0	ID1
DATA 12	**DATA 13**	**DATA 14**	**DATA n**	
ID2	ID3	DLC	DATA	

3.6.13. CAN Error Frame

MessageId = 0x6C
This message is sent asynchronously when there is some error on CAN.

Response:

DATA 0	DATA 1	DATA 2…9
Channel number	Error type	Timestamp byte 0 – 7

Channel number: Always 0

Error type:

- 0: Bit Stuff Error
- 1: Form Error
- 2: Acknowledge Error
- 3: Bit Error
- 4: CRC Error

Timestamp: 64bit number representing duration in microseconds from channel start.

3.6.14. CAN Tunnel

MessageId = 0x6D

Reading and writing of CAN tunnel. After enabling the CAN tunnel the messages received on CAN 1 channel are automatically transmitted via UDP protocol on second device and send to CAN1 channel (same for CAN2 channel). The IP address and port of second device can be setup by *Write CAN Tunnel Setting*.

Request for read:

DATA 0	DATA 1
Channel number	0x0

Response:

DATA 0	DATA 1
Channel number	Tunnel enable 0x00 – Disabled 0x01 – Enabled

Request for write:

DATA 0	DATA 1
Channel number	Tunnel enable 0x01 – Disabled 0x02 – Enabled

Response:

DATA 0
Channel number

3.6.15. Read / Write CAN Tunnel Setting

MessageId = 0x6E for read, 0x6F for write

This command is for reading or writing CAN tunnel setting. The target device IP address octets are in the order first to last, e.g. 192.168.1.101 is encoded as C0 A8 01 65. Target port is encoded as Little Endian.

Request 0x6F:

DATA 0	DATA 1–4	DATA 5–6	DATA 7
Channel number	IP Address	Port	AutoStart

Channel number: CAN channel index.

IP Address: IP address of target device.

Port: Receive port of the target device.

AutoStart: Set to 1 if you want the CAN Tunnel to start automatically after the device boots up.

Response:

DATA 0
Channel number

Response to 0x6E has the same structure as request for 0x6F above.

3.6.16. CAN Diagnostic RX Configuration (ISO-TP)

MessageId = 0x70

Request 0x70:

DATA 0	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5	DATA 6	DATA 7
Channel number	ID LSB	ID MidB 0	ID MidB 1	ID MSB	Config byte	N_Br LSB	N_Br MSB

Channel number: CAN channel index.

ID: UDS CAN ID

Config byte:

bit 7							bit 0
Reserved	Reserved	Reserved	Reserved	Reserved	Mixed Addressing	Extended Addressing	ExtID

Bits 7..3	**Reserved**
Bit 2	**Mixed Addressing**– Byte [0] of all the diagnostic messages on the bus is treated as Address Extension
Bit 1	**Extended Addressing**– Byte [0] of all the diagnostic messages on the bus is treated as Target Address
Bit 0	**Extended ID** – Use extended CAN ID

N_Br: Time until transmission of the next Flow Control frame when First Frame is received. When set to 0, Flow Control is sent right away. Maximum value is 900 ms, if large value is provided, 900 ms is used.

See below for more details of the communication.

Response:

DATA 0
Channel number

3.6.17. CAN Diagnostic TX Configuration (ISO-TP)

MessageId = 0x71

Request 0x71:

DATA 0	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5	DATA 6	DATA 7
Channel number	ID LSB	ID MidB 0	ID MidB 1	ID MSB	Config byte	Address Extension	CAN padding

Channel number: CAN channel index.

ID: UDS CAN ID

Config byte:

bit 7							bit 0
Reserved	Padding	BRS	FD	Echo	Mixed Addressing	Extended Addressing	ExtID

Bit 7	**Reserved**
Bit 6	**Padding** – Enable or disable CAN frame data padding (CAN frame data length always set to 8, unused bytes are filled with 0xCC)
Bit 5	**BRS** – Use BRS for the requests (when FD is 0, this must be 0)
Bit 4	**FD** – Use CAN FD frames (64-byte data length). Possibility of CAN FD frames is determined by general CAN channel configuration.
Bit 3	**Echo** – Enable or disable TX echo
Bit 2	**Mixed Addressing** – Address Extension byte from configuration is inserted to byte [0] of all the diagnostic messages on the bus (when Extended Addressing is 1, this must be 0)
Bit 1	**Extended Addressing** – Target Address from the request is inserted to byte [0] of all the diagnostic messages on the bus (when Mixed Addressing is 1, this must be 0)
Bit 0	**Extended ID** – Use extended CAN ID

Address extension: When Mixed Addressing is enabled, this byte is inserted at the beginning of the diagnostic messages according to DoCAN specification. Otherwise, this is ignored.

CAN padding: When Padding in the config byte above is enabled, value of this byte is used for CAN padding. User should choose 0xCC as the default value. When padding is disabled, this parameter is irrelevant.

Response:

DATA 0
Channel number

3.6.18. CAN Diagnostic Change RX State (ISO-TP)

MessageId = 0x72

This command must be issued to enable parsing of diagnostic communication. Note that device transmits echo only from valid frames, there is no error information available.
When this is enabled, device filters out CAN traffic that uses the configured RX CAN ID and transmits only echo from the valid communication (meaning that 3.6.12 Receive Frame is not used for those CAN IDs). Furthermore, when this is enabled, device automatically transmits needed Flow Control

frame when it detects it can receive some data.
When this is enabled, $P2_{CAN_Client}$ timing parameter can be set – timeout from successful transmission of a DoCAN request message to the start of incoming response message. If set to 0, it is not used.

Request:

DATA 0	DATA 1	DATA 2	DATA 3
Channel number	Enable	$P2_{CAN_Client}$ LSB	$P2_{CAN_Client}$ MSB

Channel number: CAN channel index.
Enable: 1 for enabling parsing of diagnostic traffic, 0 for disabling it.
$P2_{CAN_Client}$: Millisecond timeout from successful transmission of request message to reception of response message. Max value: 32767

Response:

DATA 0
Channel number

3.6.19. CAN Diagnostic Request (ISO-TP)

MessageId = 0x73

Transmit a diagnostic request on the bus. Note that CAN channel must be started beforehand. Transmission configuration is set using message CAN Diagnostic TX Configuration (ISO-TP). When diagnostic communication is active, CAN frame echo on specified CAN IDs is disabled.

Request 0x73:

DATA 0	DATA 1	DATA 2	…	DATA N
Channel number	TA	Diagnostic data [0]	…	Diagnostic data [N – 2]

Channel number: CAN channel index.
TA: Target address. When Extended Addressing is enabled in the configuration, this is inserted to the messages per DoCAN specification; otherwise, it is ignored.
Diagnostic data [0..N – 2]: Diagnostic data to be sent on bus

When TX echo is enabled, after transmission, this is echoed:

DATA 0	DATA 1	DATA 2	DATA 3	…	DATA N
Channel number	TA	AE	Diagnostic data [0]	…	Diagnostic data [N – 3]

Channel number: Set per request.
TA: Target address from the request.
AE: Configured Address Extension, when Mixed Addressing is not used, this is set to 0xFF.
Diagnostic data: data from the request.

Response:

DATA 0
Channel number

3.6.20. CAN Diagnostic Response (ISO-TP)

MessageId = 0x74

This message is transmitted asynchronously when device receives a diagnostic response. It is similar

to Diagnostic Request, only difference is that there is added Address Extension (note that its value is valid only when Mixed Addressing is used). Reception must be enabled using message CAN Diagnostic Change RX State (ISO-TP).

Request: Not possible

Response 0x74:

DATA 0	DATA 1	DATA 2	DATA 3	…	DATA N
Channel number	TA	AE	Diagnostic data [0]	…	Diagnostic data [N – 3]

Channel number: CAN channel index
TA: Target Address. Valid when Extended Addressing is enabled in the configuration, otherwise always set to 0xFF.
AE: Address Extension. Valid when Mixed Addressing is enabled in the configuration, otherwise always set to 0xFF.
Diagnostic data [0..N – 2]: Diagnostic data received on bus

3.6.21. CAN Diagnostic Timeout (ISO-TP)

MessageId = 0x75

This message is transmitted asynchronously when diagnostic timeout occurs.

Request: Not possible

Response 0x75:

DATA 0	DATA 1
Channel number	Reason

Channel number: CAN channel index
Reason: when the timeout occurred

- 0x01: No Consecutive Frame was received after Flow Control was transmitted (N_Cr timeout)
- 0x02: No Flow Control frame received after First Frame was transmitted (N_Bs timeout)
- 0x03: No diagnostic message reception after request was transmitted (configurable timeout $P2_{CAN_Client}$)

3.7. IO Control Messages

3.7.1. Write Digital Output

MessageId = 0xE0

Message for controlling 5V push-pull output. Least significant bit of data byte 0 controls the output (0 off, 1 on).

Request:

DATA 0	
X [7:1]	Status [0]
7 b	1 b

Response:
No data

3.7.2.Read Analogue Input

MessageId = 0xE1

Message for reading voltage value of the analogue input. Maximum input voltage is 5 V. Value is two-byte voltage measurement in millivolts and it is transmitted LSB first.

Request:

No data

Response:

DATA 0	DATA 1
Value LSB	Value MSB

3.8. Miscellaneous Messages

3.8.1.Restart Device

MessageId = 0xFD

Issuing this command makes the device restart. Restart is needed after changing IP address or port.

Request, response:
No data

3.8.2.Restart Device to Bootloader

MessageId = 0xFE

This command restarts device to System Bootloader, so that new firmware can be loaded. The new firmware can be flashed via USB and ESP tool or Arduino IDE.

Request:

DATA 0
Bootloader selection

- Bootloader selection: 0 = System bootloader

Response
No data

4. Communication Examples

4.1. Device Settings

Command	Bytes [hex]
Read SN	02 11 00 00 11 03 Example response: 02 11 04 00 00 01 02 03 1B 03 Serial number is 03020100 Bytes explanation - request: 02 – STX 11 – ID 00 00 – DATALEN 11 – Checksum 03 – ETX Response: 02 – STX 11 – ID 04 00 – DATALEN 00 01 02 03 – DATA 1B – Checksum 03 – ETX
Read MAC address	02 1B 00 00 1B 03 Example response: 02 1B 06 00 A7 19 6E C2 A5 FC B2 03 MAC address is A7:19:6E:C2:A5:FC
Write Ethernet settings IP address: 192.168.1.101 Mask: 24 (255.255.255.0) Port: 8001	02 16 07 00 C0 A8 01 65 18 41 1F 63 03 Response: 02 16 00 00 16 03
Write default gateway IP address: 192.168.1.100	02 1D 04 00 C0 A8 01 64 EE 03 Response: 02 1D 00 00 1D 03
Restart device	02 FD 00 00 FD 03

4.2. LIN

Command	Bytes [hex]
Configure LIN channel 19200, Master, Enhanced Checksum, Auto-Message Length, TX echo enabled	02 20 01 00 66 87 03 Gateway response: 02 20 00 00 20 03
Start LIN channel	02 30 00 00 30 03 Gateway response: 02 30 00 00 30 03
Transmit Master Response Frame LIN Id = 0x21 with 3 data bytes:0x01 0x02 0x03, TX echo enabled	02 40 05 00 21 03 01 02 03 6F 03 Gateway response: 02 40 00 00 40 03 – Written to buffer 02 40 05 00 21 03 01 02 03 6F 03 – LIN frame has been sent onto the LIN bus
Stop LIN channel	02 31 00 00 31 03 Gateway response: 02 31 00 00 31 03

Command	Bytes [hex]
Start LIN channel	02 30 00 00 30 03 Gateway response: 02 30 00 00 30 03
UDS request • Device NAD 0x14 • SID 0x22	02 46 04 00 14 22 64 1F 03 03 Gateway response: 02 46 00 00 46 03
Asynchronous UDS response (from an example device) • Device NAD 0x14 • RSID 0x62	02 47 08 00 14 62 64 1F 30 30 31 30 09 03

4.3. USB – CAN interface

Command	Bytes [hex]
Configure CAN channel Channel 0, CAN, AutoStart, Normal mode, Arbitration SP = 80%, Arbitration Baud = 500 kBd, Arbitration SJW = 2, Data Baud = 2MBd, Data SJW= 1, Data SP = 80%	02 60 06 00 00 28 02 01 10 08 A9 03 Gateway response: 02 60 01 00 00 61 03
Configure CAN channel timing Channel 0, CAN, Normal mode, Arbitration T_seg1 = 15, Arbitration T_seg2 = 4, Arbitration Prescaler = 4, Arbitration SJW = 2, Data T_seg1 = 5, Data T_seg2 = 1, Arbitration SJW = 1, Data Prescaler = 1	02 61 09 00 00 00 0E 03 03 01 04 00 00 83 03 Gateway response: 02 61 01 00 00 62 03
Start CAN channel Channel 0	02 67 01 00 00 68 03 Gateway response: 02 67 01 00 00 68 03
Transmit CAN Frame Channel 0, format = CAN, ID = 0x01FF, DLC = 7, Data (hex) = 05 04 50 06 06 08 14 , Tx echo disable	02 6A 0C 00 00 00 FF 01 07 05 04 50 06 06 08 14 FE 03 Gateway response: 02 6A 10 00 00 6B 03
Stop CAN channel Channel 0	02 68 01 00 00 69 03 Gateway response: 02 68 01 00 00 69 03

4.4. USB – CAN FD interface

Command	Bytes [hex]
Configure CAN FD channel Channel 0, CAN FD, AutoStart, Normal mode, Arbitration SP = 80%, Arbitration Baud = 500 kBd, Arbitration SJW = 8, Data Baud = 2MBd, Data SJW= 4, Data SP = 80%	02 60 06 00 00 68 02 07 13 08 F2 03 Gateway response: 02 60 01 00 00 61 03
Configure CAN FD channel timing Channel 0, CANFD, AutoStart, Normal mode, Arbitration T_seg1 = 15, Arbitration T_seg2 = 4, Arbitration Prescaler = 8, Arbitration SJW = 6, Data T_seg1 = 5, Data T_seg2 = 1, Arbitration SJW = 1, Data Prescaler = 1	02 61 09 00 00 60 0E 03 03 01 0E 13 00 00 03 Gateway response: 02 61 01 00 00 62 03
Start CAN FD channel Channel 0	02 67 01 00 00 68 03 Gateway response: 02 67 01 00 00 68 03
Transmit CAN FD Frame Channel 0, format = CAN FD, BRS, ID = 0x01FF, DLC = 7, Data (hex) = 05 04 50 06 06 08 14, Tx echo disable	02 6A 0C 00 00 14 FF 01 07 05 04 50 06 06 08 14 12 03 Gateway response: 02 6A 01 00 00 6B 03
Stop CAN FD channel Channel 0	02 68 01 00 00 69 03 Gateway response: 02 68 01 00 00 69 03

4.5. LIN Diagnostics

Suppose that on the LIN bus, there is connected a device with address 0x14 which responds on SID 0x22. Device is sleeping, so first request timeouts as the device wakes up, then it responds on the second request. This is how the communication could look like.

Command	Bytes [hex]
Start LIN channel	02 30 00 00 30 03 Gateway response: 02 30 00 00 30 03
Set UDS timing **Setting STmin = 50, P2 = 50**	02 45 04 00 32 00 32 00 AD 03 Gateway response: 02 45 00 00 45 03
UDS request Device wakeup	02 46 04 00 14 22 64 1F 03 03 Gateway response: 02 46 00 00 46 03
Asynchronous LIN error Error due to n**o response as the device was sleeping**	Gateway response: 02 33 02 00 02 3D 74 03
UDS request • Device NAD 0x14 • SID 0x22	02 46 04 00 14 22 64 1F 03 03 Gateway response: 02 46 00 00 46 03
Asynchronous UDS response Response from an example device. • Device NAD 0x14 • RSID 0x62	02 47 08 00 14 62 64 1F 30 30 31 30 09 03

4.6. CAN Diagnostics

Example of DoCAN communication. Presume that CAN channel was previously configured as needed.

Command	Bytes [hex]
Start CAN channel **Channel 0**	02 67 01 00 00 68 03 Gateway response: 02 67 01 00 00 68 03
Disable CAN echo Channel 0	02 66 02 00 00 00 68 03 Gateway response: 02 66 01 00 00 67 03
DoCAN RX configuration Channel 0 CAN ID 0x72C No mixed addressing, no extended addressing N_Br = 100 ms	02 70 08 00 00 2C 07 00 00 00 64 00 0F 03 Gateway response: 02 70 00 00 70 03
DoCAN TX configuration Channel 0 CAN ID 0x724 Enable TX echo Enable padding to 8 bytes	02 71 07 00 00 24 07 00 00 48 00 EB 03 Gateway response: 02 71 00 00 71 03
DoCAN configure parsing Channel 0 Enable parsing of diagnostic messages $P2_{CAN_Client}$ = 1000 ms	02 72 04 00 00 01 E8 03 62 03 Gateway response: 02 72 00 00 72 03
DoCAN request Channel 0 TA = 0 (not used) Data = 22 F1 88 (SID 0x22)	02 73 05 00 00 00 22 F1 88 13 03 Gateway response: 02 73 01 00 00 74 03
Asynchronous response – request was transmitted Channel 0 TA = 0x00 (not used, filled by the request) AE = 0xFF (not used, was not set) Data = 0x22 0xF1 0x88	02 73 06 00 00 00 FF 22 F1 88 13 03
Asynchronous response – response was received Channel 0 TA = 0xFF (Extended addressing is disabled) AE = 0xFF (Mixed addressing is disabled) Diagnostic data: 0x62 0xF1 0x88 0x4D 0x41 0x43 0x48 0x20 0x53 0x59 0x53 0x54 0x45 0x4D 0x53 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00	02 74 1E 00 00 FF FF 62 F1 88 4D 41 43 48 20 53 59 53 54 45 4D 53 00 00 00 00 00 00 00 00 00 00 00 DE 03

5. Contact

MACH SYSTEMS s.r.o.
www.machsystems.cz
info@machsystems.cz
Czech Republic

Company Registration: 29413893
VAT no.: CZ29413893