100BASE-T1 Media Gateway
Communication Protocol Specification

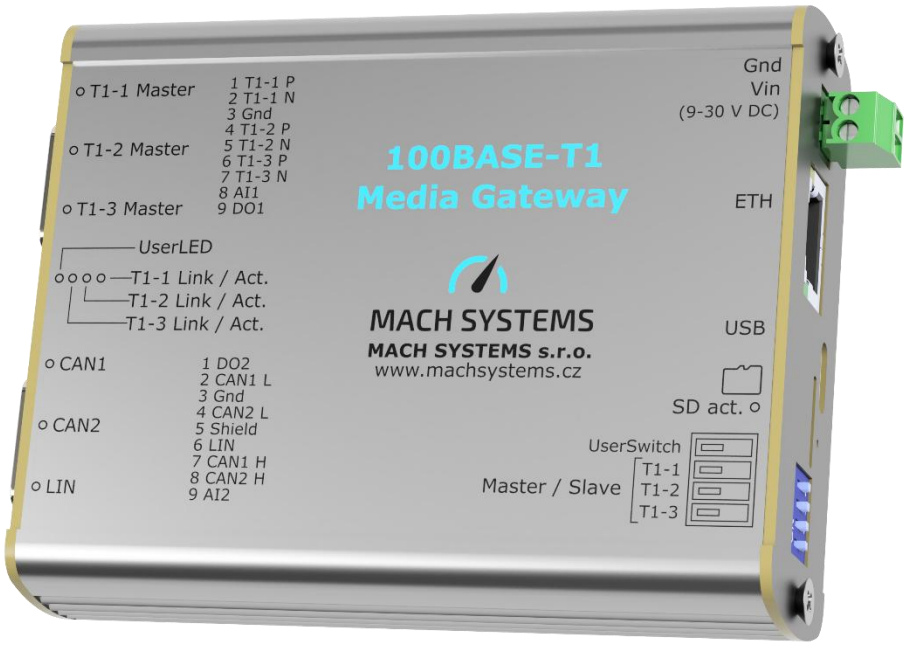
Changes

Date	Change	Changed by
24.06.2024	DHCP configuration added	PK
10.07.2023	Initial version	PK

Contents

1. Introduction

The **100BASE-T1 Media Gateway** (p/n: **100BASET1-MG)** is a device that acts as an Ethernet switch with three 100BASE-T1 ports, MCU port and a 1000BASE-T port. Also, it can be used as an interface for CAN(/FD) (2 channels) or LIN (1 channel), simultaneously to its switch function. The interface functionality is available through the USB or Ethernet.



The device offers an Ethernet port, USB port, two CAN(/FD) channel, and a LIN channel, and uses a standard RJ-45 connector and a USB Type-C connector (virtual serial port - Virtual COM port - VCP).
This document describes a communication protocol used by the 100BASE-T1 Media Gateway device so that it can be integrated into another system, such as HiL tester, simulator, PLC and others.

The 100BASE-T1 Media Gateway can operate 5 Ethernet ports. First, second and third port are 100BASE-T1 type and are available through a DSUB9 connector. Fourth port is connected to the internal MCU, and the fifth port is 1000/100/10 Ethernet port available through a RJ-45 connector. The device can be configured through them or it can be used as mirror port. The advance settings of the ports (etc. port mirroring, VLAN tagging, address resolution table modification…) can be done only through the embedded web server.
Configuration of device can be saved into a non-volatile memory, and is automatically loaded on power-up.

This document describes the communication protocol over Ethernet and CAN(/FD) that can be used to programmatically read the device status and write configuration.

2. Device Details

100BASE-T1 Media Gateway
p/n: 100BASET1-MG

2.1. Specification

- 3 100BASE-T1 ports
- Standard Ethernet (10/100/1000 Mbit) port with RJ-45 connector
- 2 CAN(/FD) channel
- LIN channel
- USB-Type-C 2.0
- 2 Digital outputs (high and low side)
- MicroSD card slot
- Web interface for easy configuration
- Open communication protocol over Ethernet, USB, and CAN(/FD) for integration
- Can be used as a USB-CAN(/FD) or Ethernet-CAN(/FD) interface
- It can be also used as a USB-LIN or Ethernet-LIN interface
- USB-powered or external 9 – 30 V DC
- Aluminium enclosure (80 x 82 x 32 mm)
- Firmware update over web
- 4 DIP switches for easy configuration
- DIN rail mounting (bracket sold separately)

2.2. Usage Examples

- Automotive Ethernet switch
- Media gateway
- Ethernet-CAN(/FD) interface
- Ethernet-LIN(/FD) interface
- USB - CAN(/FD) interface
- USB – LIN interface
- Logging communication between two or more units
- Create test specific address resolution table

3. Communication Protocol

The communication between the 100BASE-T1 Media Gateway and other system is based upon a binary protocol. The same message structure is used for both directions - to and from the device.
This protocol consists of Message ID and Data. For USB and Ethernet communication, the protocol is encapsulated by Start byte, Data length, Checksum and End byte. For CAN bus, the protocol is placed into the data bytes of a CAN frame.

3.1. USB and Ethernet

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

Ethernet default configuration:

IP address	**192.168.1.100**
Subnet mask	255.255.255.0
Default gateway	192.168.1.1
Port	**8000**
Protocol	TCP or UDP

The device offers communication over both TCP/IP and UDP and its Ethernet parameters can be changed. This can be done directly from the protocol or, as there is a web server running, also via a web browser (Google Chrome is recommended).

Communication protocol structure:

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 of ID, DATALEN and all DATA bytes	0x03

The rest of the document 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.

3.2. CAN bus

If the device CAN port is not used as a USB/Ethernet-CAN interface, it is possible to use the CAN bus for diagnostic purposes. The device receives via CANID_RX and transmits over CANID_TX. Both CAN identifiers can be changed per device – see 4.5.1, 4.5.2, 4.5.3.

Default configuration:
CANID_RX = 0x123 Std ID.
CANID_TX = 0x321 Std ID.
CAN Baud = 500 KBaud, sample point: 80%
Frame format: Classical CAN (CAN FD support can be enabled)

CAN frame – data part:

CAN DATA 0 (1 B)	**CAN DATA (0 to 7 B for CAN) (0 to 63 B for CAN FD)**
Message ID	Protocol data bytes

Data Byte 0 is always used as Message ID, the rest of the data bytes carry the message content.

3.3. Message Overview

Note that all the channel indexes in the configuration (CAN, Ethernet ports) are zero indexed. Even though the first Ethernet port is called T1-1 Port 1, its index is 0. The CAN setting messages (ID 0x60 – 0x6A) are not available over CAN, usage of those IDs over CAN will result in error response.

Message ID	Name	Request Data Length	Response Data Length	Description
0x01	BOOT_UP	No request needed	4	*A notification that the gateway was powered up. The data bytes of the response are CANID_BASE_RX. Sent via CAN and USB.*
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	ETH_RESET_CONFIGURATION	0	0B ACK	Restore the default communication configuration
0x15	ETH_READ_CONFIGURATION	0	13	Read configuration
0x16	ETH_WRITE_CONFIGURATION	7	0B ACK	Write configuration
0x17	ETH_READ_IP_ADDRESS	0	5	Read IP address and mask
0x18	ETH_WRITE_IP_ADDRESS	5	0B ACK	Write IP address and mask
0x19	ETH_READ_PORT	0	2	Read communication port
0x1A	ETH_WRITE_PORT	2	0B ACK	Write communication port
0x1B	ETH_READ_MAC_ADDRESS	0	6	Read MAC address
0x1C	ETH_READ_DEFAULT_GW	0	4	Read default gateway
0x1D	ETH_WRITE_DEFAULT_GW	4	0B ACK	Write default gateway
0x1E	ETH_DHCP	1	0B ACK or 1	Enable / disable the DHCP client
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*
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.
Device configuration – CAN protocol				
0x5A	CAN_WRITE_LOCK_TOGGLE	1	0B ACK	Unlock / lock CAN parameters change
0x5B	CAN_READ_RXID	1	5	Read CAN ID for protocol RX
0x5C	CAN_WRITE_RXID	5	1B ACK	Write CAN ID for protocol RX
0x5D	CAN_READ_TXID	1	5	Read CAN ID for protocol TX
0x5E	CAN_WRITE_TXID	5	1B ACK	Write CAN ID for protocol TX
0x5F	CAN_READ_STATUS	1	2	Read status of CAN interface
CAN communication control				
0x60	CAN_WRITE_CONFIG	6	1B ACK	Configure CAN channel
0x61	CAN_WRITE_CONFIG_TIM	9	1B ACK	Configure CAN channel (with time quanta setting)
0x62	CAN_READ_CONFIG	1	12	Read CAN channel configuration
0x63	CAN_SAVE_CONFIG	1	1B ACK	Save CAN configuration to non-volatile memory
0x64	CAN_LOAD_CONFIG	1	1B ACK	Load CAN configuration from non-volatile memory
0x65	CAN_DEFAULT_CONFIG	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 microseconds from startup of channel
0x6A	CAN_SEND_MESSAGE	5 to 71	1B ACK	Send CAN message / CAN message was sent
0x6B	CAN_RECEIVED_MESSAGE	N/A	13 to 79	Received CAN message
0x6C	CAN_ERROR_FRAME	N/A	10	Some error on CAN bus
100BASE-T1 Diagnostic				
0x70	READ_T1_STATUS	1	2	Read status information (port link status etc.)
0x71	READ_SQI	1	2	Read signal quality

0x72	DO_CABLE_TEST	1	2	Run the cable test
0x73	WRITE_MASTER_SLAVE	2	1B ACK	Override T1 Master/Slave
0x74	PHY_TEST_MODE	2	1B ACK	T1 test modes
I/O control				
0xE0	IO_WRITE	1	0B ACK	Toggle digital outputs
0xE1	IO_READ	0	4	Read analog inputs
Miscellaneous				
0xFD	RESTART	0	0	Restart the device
0xFE	RESTART_BOOT	1	0	Restart device to USB / web bootloader
0xFF	GENERAL_ERROR	N/A	1, 2 or 3	An error occurred, see Error Codes for description

3.4. Error Codes

The following table describes error codes. General structure of error message is described below.

Error Code	Data length	Comment
Communication protocol error 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
0xA4	2	Invalid data
0xA5	2	Attempt to change CAN configuration from CAN without unlock
0xA6	2	Configuration could not be saved – EEPROM error
General bus errors Messages contain: Error Code, Message ID and Channel Number		
0xF0	3	Configuration Error
0xF1	3	Channel running, channel should be stopped when it is being configured
0xF2	3	Invalid channel selected – index out of bounds
0xF3	3	Channel is not running
0xF4	3	Hardware FIFO is full (should not happen in normal operation)

4. Message Specification

4.1. General Response

Device responds with a message acknowledgment after receiving a valid message. The acknowledgement has the same ID as the original message. Acknowledgement is either without any data or it contains one data byte which signals which bus channel was relevant to the request. See Response Data Length in Message Overview for information about acknowledge length. If there is some problem, the response is Error Response.

Response:
No data when bus channel number is not relevant.

OR

DATA 0
Channel number

4.2. Error Response

Message ID = 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.

4.3. Device Message Specification

4.3.1. Device serial number

Message ID = 0x11
This command is used for reading device serial number.
Request 0x11:
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

4.3.2.Device hardware information

Message ID = 0x12

This command is used for reading device hardware number.

Request 0x12:

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

4.3.3.Device software information

Message ID = 0x13

This command is used for reading software version number.

Request 0x13:

No data

Response:

DATA 0	DATA 1
VERSION MINOR	VERSION MAJOR

4.4. ETH Message Specification

4.4.1.Restore Default Configuration

Message ID = 0x14

Reset the default communication configuration – IP address, mask and port.

Request: Message without data.

Response: acknowledge.

4.4.2.Read and Write Configuration

Message ID = 0x15 for read, 0x16 for write

This command is used for changing IP address, mask and TCP port all in one step. After issuing this command, you must restart the device for changes to apply.

The read variant is for reading IP address, mask, TCP port and MAC address. The difference is that you cannot write the MAC address. Note that when this Message ID is used over CAN, there is no MAC address.

Request 0x15:

No data

Response:

DATA0 – DATA 3	DATA 4	DATA 5 – DATA 6	DATA 7 – DATA 12

IP address	Mask	Port	MAC Address

Request for 0x16:

DATA0 – DATA 3	DATA 4	DATA 5 – DATA 6
IP address	Mask	Port

Response: acknowledge.

4.4.3.Read and Write IP Address Configuration

Message ID = 0x17 for read, 0x18 for write

This command is for reading or writing device IP address and subnet mask. 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:

DATA0 – DATA 3	DATA 4
IP address	Mask

Response: acknowledge.

Response to ID 0x17 has the same structure as request 0x18 above.

4.4.4.Read and Write TCP Port

Message ID = 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.

Request 0x19:

DATA0 – DATA 1
Port

Response: acknowledge.

Response to 0x1A has the same structure as request for 0x19 above.

4.4.5.Read MAC Address

Message ID = 0x1B

This command is for reading device MAC address. Each device has unique MAC address which cannot be changed by the user. MAC address octets are in the order first to last, e.g. A7:19:6E:C2:A5:FC is encoded as A7 19 6E C2 A5 FC.

Request 0x1B:

No data

Response:

DATA 0 – DATA 5
MAC address

4.4.6.Read and Write Default Gateway

Message ID = 0x1C for read, 0x1D for write

Used for reading / changing the default gateway. 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.

4.4.7.Read and Write DHCP Enable / Disable

MessageId = 0x1E

Reading and writing of DHCP client enable. When DHCP is enabled, issuing IP address, mask or default gateway read means reading the currently active network value.

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

4.4.8.LIN Channel Configuration

Message ID=0x20

This command is for configuration of LIN interface. The Enhanced Checksum can't be selected when **AMLR == 0**. The LIN interface supports two baud rates (9600 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**
  - **10 – 19200**
  - 11 – Reserved

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

4.4.9.Read Configuration

Message ID=0x21

This command reads LIN interface configuration register.

Request:

No data

Type of response:

DATA 0
Configuration Register

4.4.10. Save Configuration

Message ID=0x22

This command saves LIN interface configuration register to EEPROM.

Request:
No data
Response:
No data

4.4.11. Load Configuration

Message ID=0x23

This command loads saved configuration register from EEPROM.

Request:
No data
Response:
No data

4.4.12. Default Configuration

Message ID=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

4.4.13. Start LIN

This command starts LIN interface.
Message ID=0x30

Request:
No data

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

4.4.14. Stop LIN

This command stops LIN interface.
Message ID=0x31

Request:
No data
Response:
No data

4.4.15. 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.
Message ID=0x32

Request:

DATA 1
Echo configuration

bit 7	bit 2	bit 1	bit 0
Reserved		TXECHO	RXECHO

- Bit 7..2: **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)

Response:
No data

4.4.16. LIN Error

Message ID = 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

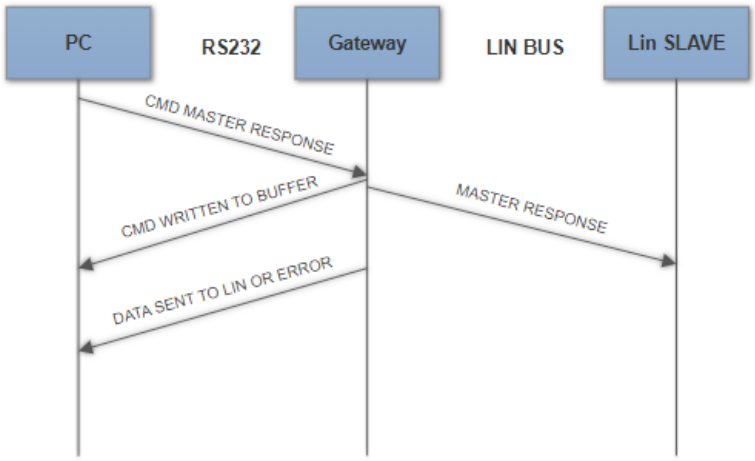
- 3: Error data too long

4.4.17. Transmit Master Response

Message ID=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.

4.4.18. Transmit Master Request

Message ID=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.

4.4.19. Receive Master Response

Message ID=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.

4.4.20. Slave Response Configuration

Message ID=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 4.4.22 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 4.4.22 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.

4.4.21. Slave Response TX

Message ID=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

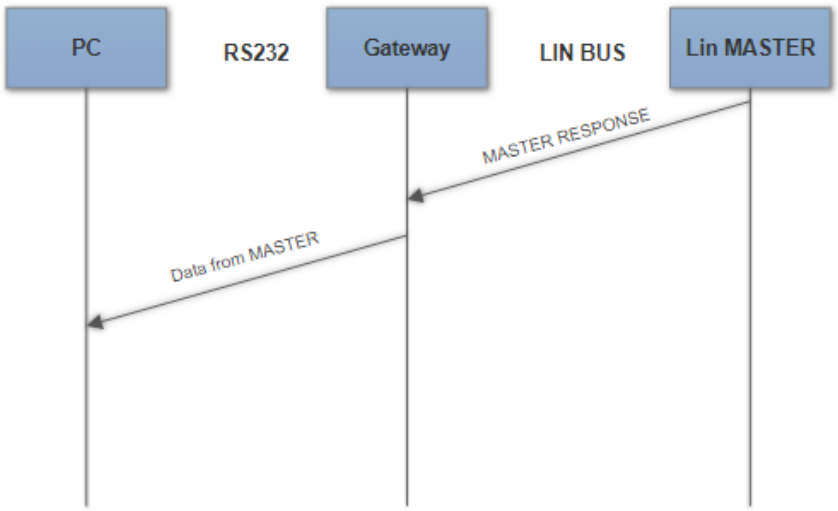
DATA 0	DATA 1	DATA 2	DATA n
LIN ID	LIN DATALEN	LIN DATA 0	LIN DATA n

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

4.4.22. Slave Response RX

Message ID=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.

4.5. CAN Communication Protocol Configuration

4.5.1. Unlock and Lock CAN configuration change

Message ID = 0x5A

Before changing CAN protocol Rx / Tx ID **via CAN**, you must issue this command to enable configuration change (other channels (TCP,…) can change those settings without issuing this command). If the data byte equals to one, configuration is enabled. If it is something else, it is disabled again. This command unlock/lock setting for both CAN channels.

Request 0x5A:

DATA 0
DATA = 1 for unlock DATA ≠ 1 for lock

4.5.2. Read and Write CAN protocol Rx ID

Message ID = 0x5B for read, 0x5C for write

If issued via CAN, command for unlocking CAN settings must be issued prior to write variant of this command. Reading and changing the CAN ID which is used for configuration messages. Default setting is 0x123 (standard ID). The Rx ID can be set for each CAN channel.

Request 0x5B:

DATA 0

Channel number

Request 0x5C:

DATA 0	DATA 1	DATA 2	DATA 3	DATA 4
Channel number	New CAN ID for Rx LSB	New CAN ID Byte 1	New CAN ID Byte 2	New CAN ID MSB + information

New CAN ID MSB + information:

bit 7							
EXT ID	FDF	Reserved	ID28	ID27	ID26	ID25	ID24

- Bit 7: **EXT ID** – determines if configuration messages are received with extended ID
  - 0: Protocol messages must have Standard ID
  - 1: Protocol messages must have Extended ID
- Bit 6: **FDF** – CAN or CAN FD frame
  - 0: Protocol messages must not have FDF flag set (CAN frame)
  - 1: Protocol messages must have FDF flag set (CAN FD frame). Valid only when channel operates in CAN FD mode
- Bit 5: Reserved
- Bits [4:0]: Bits [28:24] of extended CAN ID of configuration messages (if applicable)

Response: Channel number when success, error message if CAN configuration change was not previously unlocked.

Response to 0x5B has the same structure as request for 0x5C above.

4.5.3.Read and Write CAN Protocol Tx ID

Message ID = 0x5D for read, 0x5E for write

If issued via CAN, command for unlocking CAN settings must be issued prior to write variant. Reading and changing the CAN ID which is used for configuration messages. Default setting is 0x321. The Tx ID can be set for each CAN channel.

Request 0x5E:

DATA 0	DATA 1	DATA 2	DATA 3	DATA 4
Channel number	New CAN ID for Tx LSB	New CAN ID Byte 1	New CAN ID Byte 2	New CAN ID MSB + information

New CAN ID MSB + information:

bit 7							
EXT ID	FDF	BRS	ID28	ID27	ID26	ID25	ID24

- Bit 7: **EXT ID** – determines if configuration messages are sent with extended ID
  - 0: Protocol messages have Standard ID
  - 1: Protocol messages have Extended ID
- Bit 6: **FDF** – CAN or CAN FD frame

    - 0: Protocol messages do not have FDF flag set (CAN frame)
    - 1: Protocol messages have FDF flag set (CAN FD frame). Valid only when channel operates in CAN FD mode
- Bit 5: **BRS** – Bit Rate Switch
    - 0: Protocol messages do not have BRS flag set
    - 1: Protocol messages have BRS flag set (relevant when FDF = 1 and channel operates in CAN FD mode)
- Bits [4:0]: Bits [28:24] of extended CAN ID of configuration messages (if applicable)

Response: Channel number when success, error message if CAN configuration change was not previously unlocked.

Response to 0x5D has the same structure as request for 0x5E above.

4.5.4.Read CAN Channel Status

Message ID = 0x5F

Read status of the CAN channel.
If flag RUNNING is set: Channel is started. Frames can be received/transmit, forwarding to PC is enabled.

Request:

DATA 0
Channel number

Response:

DATA 0	DATA 1
Channel number	CAN status

CAN1 status:

bit 7	bit 2	bit 0
Reserved		RUNNING

- Bits [7:2]: Reserved
- Bits [1:0]:

    - 00 – Can running for protocol messages
    - 01 – Can running as interface (use the messages **0x6A** for transmission)
    - 10 – Can running as gateway (use the CAN gateway protocol for message transmission)
    - 11 – Reserved

4.6. CAN and CAN FD Messages

Messages in this section cannot be used over CAN. Attempt to use them over CAN will result in error response.

4.6.1. Channel Configuration

Message ID = 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 0	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5
Channel number 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	CHNL6	CHNL5	CHNL4	CHNL3	CHNL2	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.
- Bits [6:0]: **CHNL**: Always 0.

Configuration register 1:

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

- Bits [7:6]: **Protocol selection**
  - 00 – CAN 2.0B
  - 01 – ISO CAN FD
  - 10…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
- Bits [3:0]: **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…1111 – Reserved

Configuration register 2:

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

- Bits [2:0]: **Arbitration baud rate**
  - 000 – 125 kBd
  - 001 – 250 kBd
  - 010 – 500 kBd
  - 011 – 1 MBd
  - 100…111 – Reserved

Configuration register 3:

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

- Bits [6:0]: **Arbitration jump width**
  - 0000000 – 1
  - 0000001 – 2
  - 0000010 – 3
  - 0000011 – 4
  - …
  - 1111111 – 128

Configuration register 4 (**relevant for CAN FD mode only**):

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

- Bits [6:4]: **Data baud rate**
  - 000 – 1 MBd
  - 001 – 2 MBd
  - 010 – 4 MBd
  - 011 – 8 MBd
  - 100..111 – Reserved
- Bits [3:0]: **Data Synchronization jump width**
  - 0000 – 1
  - 0001 – 2
  - 0010 – 3
  - 0011 – 4
  - …
  - 1111 – 16

Configuration register 5 (**relevant for CAN FD mode only**):

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

- Bits [3:0]: **Data Sample Point**. Note that for baud rate 4 MBd, actual value will be rounded to nearest lower multiple of 5 %. For 8 MBd, it is nearest lower 10 %. Furthermore, 87.5 % will always be rounded to 85 % due to hardware constraints.
  - 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.

Default configuration

- 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 disabled

4.6.2.Channel Configuration with Time Quanta Timing

Message ID = 0x61

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

Request:

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	CHNL6	CHNL5	CHNL4	CHNL3	CHNL2	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.
- Bits [6:0]: **CHNL**: Always 0.

Configuration register 1: Has the same structure as in Channel Configuration except there is no ASP (arbitration selection) field. Those bits have no meaning here.

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

- Bits [7:6]: **Protocol**
  - 00 – CAN 2.0B
  - 01 – ISO CAN FD
  - 10…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
- Bits [3:0]: Reserved

Configuration register 6:

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

- Bits [7:0]: **Arbitration time segment 1**
  - 0000 0000 – 1
  - 0000 0001 – 2
  - 0000 0010 – 2
  - 0000 0011 – 3
  - …
  - 1111 1111 – 256

Configuration register 7:

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

- Bits [6:0]: **Arbitration time segment 2**
  - 000 0000 – 1
  - 000 0001 – 2

- 000 0010 – 3
- 000 0011 – 4
- …
- 111 1111 – 128

Configuration register 8:

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

- Bits [7:0]: **Arbitration prescaler**
  - 0000 0000 – 1
  - 0000 0001 – 2
  - 0000 0010 – 3
  - 0000 0011 – 4
  - …
  - 1111 1111 – 256

Configuration register 9:

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

- Bits [6:0]: **Arbitration jump width**
  - 000 0000 – 1
  - 000 0001 – 2
  - 000 0010 – 3
  - 000 0011 – 4
  - …
  - 111 1111 – 128

Configuration 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

- Bits [4:0]: **Data time segment 1**
  - 0 0000 – 1
  - 0 0001 – 2
  - 0 0010 – 2
  - 0 0011 – 3
  - …
  - 1 1111 – 32

Configuration register 11 (**relevant for CAN FD mode only**):

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

- Bits [7:4]: **Data Synchronization jump width**
  - 0000 – 1
  - 0001 – 2
  - 0010 – 3
  - 0011 – 4

- …
- 1111 – 16

- Bit [3:0]: **Data time segment 2**
  - 0000 – 1
  - 0001 – 2
  - 0010 – 3
  - 0011 – 4
  - …
  - 1111 – 16

Configuration register 12 (**relevant for CAN FD mode only**):

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

- Bits [4:0]: **Data prescaler**
  - 0 0000 – 1
  - 0 0001 – 2
  - 0 0010 – 3
  - 0 0011 – 4
  - …
  - 1 1111 – 32

Response:

DATA 0
Channel number

Possible errors: CAN channel cannot be reconfigured - wrong arbitration or data jump width.

4.6.3.Read Configuration

Message ID = 0x62

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.

Request:

DATA 0
Channel number

Response:

DATA 0	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5
Channel number	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 11 **(CAN FD mode)**
DATA 12					

Echo configuration

Configuration register 1: see Channel Configuration for this register's field description.

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

Configuration register 2: see Channel Configuration for this register's field description.

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

Configuration register 3: see Channel Configuration for this register's field description. Same as register 10 in Channel Configuration with Time Quanta Timing.

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

Configuration register 6: see Channel Configuration with Time Quanta Timing for this register's field description.

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

Configuration register 7: see Channel Configuration with Time Quanta Timing for this register's field description.

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

Configuration register 8: see Channel Configuration with Time Quanta Timing for this register's field description.

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

Configuration register 4 (**relevant for CAN FD mode only**): see Channel Configuration for this register's field description.

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

Configuration register 5 (**relevant for CAN FD mode only**): see Channel Configuration for this register's field description.

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

Configuration register 10 (**relevant for CAN FD mode only**): see Channel Configuration with Time Quanta Timing for this register's field description.

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

Configuration register 11 (**relevant for CAN FD mode only)**: see Channel Configuration with Time Quanta Timing for this register's field description. Only difference is that there is not the DSJW field (it can be seen in register 4).

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

Configuration CHANNEL N Register 12 (**relevant for CAN FD mode only)**: see Channel Configuration with Time Quanta Timing for this register's field description.

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

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)

4.6.4.Save configuration

Message ID = 0x63

Saves CAN configuration to the non-volatile memory. Note that configuration can be also saved in the moment of reconfiguration if SAVE bit is used (see above).

Request:

DATA 0
Channel number

Response:

DATA 0
Channel number

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

4.6.5.Load Configuration

Message ID = 0x64

Load the CAN configuration from non-volatile memory.

Request:

DATA 0
Channel number

Response:

DATA 0
Channel number

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

4.6.6.Default Configuration

Message ID = 0x65

Apply CAN default configuration. For default configuration values see 4.6.1 Channel Configuration.

Request:

DATA 0
Channel number and SAVE bit

Channel number and SAVE bit:

bit 7							
SAVE	CHNL6	CHNL5	CHNL4	CHNL3	CHNL2	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.
- Bits [6:0]: **CHNL**: Always 0.

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.

4.6.7.Frame Echo Configuration

Message ID = 0x66

Request:

DATA 0	DATA 1
Channel number	Echo configuration

Echo configuration:

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

- Bit 1: **TX Echo on/off**
  - 0 – Echo off (default)
  - 1 – Echo on
- 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.

4.6.8.Start Channel

Message ID = 0x67
Start CAN channel.

Request:

DATA 0
Channel number

Response:

DATA 0
Channel number

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

4.6.9.Stop Channel

Message ID = 0x68
Stop CAN channel.

Request:

DATA 0
Channel number

Response:

DATA 0
Channel number

Acknowledgment when channel was stopped, general error message in case of error.
Reasons for error: wrong channel selected, CAN channel is not running

4.6.10. Get Channel Timestamp

Message ID = 0x69
Get CAN channel timestamp. Timestamp is 64-bit number that represents the time from startup of CAN(/FD) channel in microseconds.

Request:

DATA 0
Channel number

Response:

DATA 0	DATA 1 - 8
Channel number	Timestamp byte 0 – 7

Channel number: always 0.
Timestamp: microsecond timestamp LSB first.

4.6.11. Transmit Frame / Frame Transmitted

Message ID = 0x6A

This message transmits CAN frame. The structure of frame is different when Extended ID is set. Without extended ID the header (protocol frame data before CAN data) is 5 bytes long. With extended ID it is 7 bytes long. The format of ID is LSB.
When the frame is sent on the bus, there is another message with this frame ID.

Request if EXT ID bit (see below) is 0:

DATA 0	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5 - n
Channel	MESSAGE_INFO	ID0	ID1	DLC	DATA

Request if EXT ID bit (see below) is 1:

DATA 0	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5	DATA 6	DATA 7 - n
Channel	MESSAGE_INFO	ID0	ID1	ID2	ID3	DLC	DATA

Channel must be set to 0.
MESSAGE_INFO:

bit 7							bit 0
Reserved	Reserved	Reserved	FDF	ESI	BRS	RTR	EXT ID

- Bit 4: **FDF**
  - 0 – Frame to be transmitted in Classic CAN format
  - 1 – Frame to be 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: **EXT ID**
  - 0 – Standard ID. Request without data is 5 bytes.
  - 1 – Extended ID. Request without data is 7 bytes.

DLC: Number of data bytes.

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 Tx echo is enabled, device sends another message with this ID after the CAN frame is really sent on the bus. Its format again depends on state of the EXT ID bit (standard or extended CAN ID).
Response with EXT ID equal to zero:

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

Response with EXT ID equal to one:

DATA 0	DATA 1	DATA 2 -9	DATA 10	DATA 11	DATA 12	DATA 13
Channel	MESSAGE_INFO	Timestamp bytes 0 - 7	ID0	ID1	ID2	ID3
DATA 14	**DATA 15 - n**					
DLC	CAN DATA					

Timestamp is 64-bit number that represents the time from startup of CAN(/FD) channel in microseconds. The bit order in message is LSB. Other parts of this message are the same as in the send request.

4.6.12. Frame Received

Message ID = 0x6B
Message response has similar structure as Transmit Frame. The only difference is in the added timestamp bytes (data bytes 2 to 9). Timestamp represents the time from startup of CAN(/FD) channel to reception of the frame in microseconds. For this message no request is needed, device sends it when it receives some CAN frame. Upon reception of CAN error frame, CAN Error Frame described in the next section is sent.

Response if frame with Standard ID was received (EXT ID bit is 0):

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

Response if frame with Extended ID was received (EXT ID bit is 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	

Meaning of the fields is exactly the same as in transmission request message.

4.6.13. CAN Error Frame

Message ID = 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: 64-bit number representing duration in microseconds from channel start.

4.7. 100BASE-T1 Diagnostic

4.7.1. T1 Status

Message ID = 0x70
Read status of T1 port.

Request:

DATA 0
Port number

Response:

DATA 0	DATA 1
Port number	Port Status

Port Status:

bit 7 bit 6	bit 5 bit 3	bit 2	bit 1	bit 0
Reserved	Operating mode	Polarity detection	Master / Slave	T1-Link

- Bits [7:6]: **Reserved**
- Bits [5:3]: **Operating Mode**
  - 000 – Normal operation
  - 001 – 100BASE-T1 test mode 1
  - 010 – 100BASE-T1 test mode 2
  - 011 – Test mode 3
  - 100 – 100BASE-T1 test mode 4
  - 101 – 100BASE-T1 test mode 5
  - 110 – Scrambler and descrambler bypassed
- Bit 2: **Polarity detection**
  Polarity is valid only if selected mode is slave.
  - 0 – Normal Polarity
  - 1 – Polarity Inverted
- Bit 1: **Master / Slave selection**
  - 0 – Slave
  - 1 – Master
- Bit 0: **T1-Link**
  - 0 – Link Down
  - 1 – Link Up

4.7.2. Read SQI

Message ID = 0x71

Read Signal Quality Indicator of T1 port.

Request:

DATA 0
Port number

Response:

DATA 0	DATA 1
Port number	SQI

SQI:

bit 7	bit 4	bit 3	bit 0
Reserved		SQI	

- Bits [7:3]: **Reserved**
- Bits [2:0]: **SQI**
  - 0000 – no link
  - 0001 – worse than class A SQI (unstable link)
  - 0010 – class A SQI (unstable link)
  - 0011 – class B SQI (unstable link)
  - 0100 – class C SQI (good link)
  - 0101 – class D SQI (good link; bit error rate < 1e-10)
  - 0110 – class E SQI (good link)
  - 0111 – class F SQI (very good link)
  - 1000 – class G SQI (very good link)

4.7.3. Cable Test

Message ID = 0x72

Run cable test on T1 port.

Request:

DATA 0
Port number

Response:

DATA 0	DATA 1
Port number	Cable test result

Cable test result:

bit 7	bit 2	bit 1	bit 0
Reserved		Cable test result	

- Bits [7:2]: **Reserved**
- Bits [1:0]: **Cable test result**
  - 00 – Ok
  - 01 – Open circuit

- 10 – Short circuit
- 11 – Test fail

4.7.4.T1 Master/Slave Overriding

Message ID = 0x73

Select the Master / Slave setting

Request:

DATA 0	DATA 1
Port number	Master / Slave selection

Cable test result:

bit 7	bit 2	bit 1	bit 0
Reserved		Master / Slave selection	

- Bits [7:2]: **Reserved**
- Bits [1:0]: **Master / Slave selection**
  - 00 – Value is set by a dip switch
  - 01 – T1 is set up as slave
  - 10 – T1 is set up as master
  - 11 – Reserved

Response:

DATA 0
Port number

4.7.5. Test Modes

Message ID = 0x74

Set T1 Test Mode

Request:

DATA 0	DATA 1
Port number	T1 test mode

Response:

DATA 0
Port number

Acknowledgment when configuration was loaded, general error message in case of error. Possible reason wrong test mode.

T1 test mode:

bit 7	bit 3	bit 2	bit 0
Reserved		T1 test mode	

- Bits [7:3]: **Reserved**
- Bits [2:0]: **T1 test mode**
  - 000 – Normal operation
  - 001 – 100BASE-T1 test mode 1
  - 010 – 100BASE-T1 test mode 2
  - 011 – Test mode 3
  - 100 – 100BASE-T1 test mode 4

- 101 – 100BASE-T1 test mode 5
- 110 – Scrambler and descrambler bypassed

4.8. IO Control Messages

4.8.1. Write Digital Output

Message ID = 0xE0

Message for controlling DO1 down side switch and DO2 5V high side switch. Two least significant bit of data byte 0 controls the output (0 off, 1 on).

Request:

DATA 0
Output control

Output control:

bit 7	bit 2	bit 1	bit 0
Reserved		DO2	DO1

- Bits [7:2]: Reserved
- Bit 1: **DO2**
  - 0 – Output low
  - 1 – Output high
- Bit 0: **DO1**
  - 0 – Output low
  - 1 – Output high Z

Response:

No data

4.8.2. Read Analogue Input

Message ID = 0xE1

Message for reading voltage value of the analogue inputs AD1 and AD2. Maximum input voltage is 30 V. Value is two-byte voltage measurement in millivolts and it is transmitted LSB first.

Request:

No data

Response:

DATA 0	DATA 1	DATA 2	DATA 3
Value AD1 LSB	Value AD1 MSB	Value AD2 LSB	Value AD2 MSB

4.9. Miscellaneous Messages

4.9.1. Restart Device

MessageID = 0xFD

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

Request, response: No data

4.9.2.Restart Device to Bootloader

Message ID = 0xFE

This command restarts device to System Bootloader, so that new firmware can be loaded. It can be chosen which bootloader will be started: System Bootloader for connection via USB and STM32CubeProgrammer or HTTP bootloader for upload from web browser.

System Bootloader: Only USB cable and STM32CubeProgrammer is needed for flashing the device.
HTTP bootloader: Recommended web browser for firmware upload is Google Chrome. Uploaded file must be in the binary format (.bin).
Request:

DATA 0
Bootloader selection

- Bootloader selection: 0 = System Bootloader, 1 = Web bootloader

Response:
No data

5. Communication Examples

5.1. 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 – SN 03020100
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 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

5.2. LIN communication

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 01 00 01 42 03 – Written to buffer 02 40 02 00 02 21 65 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

5.3. CAN communication (device acting as CAN interface)

Command	Bytes [hex]
Write CAN configuration Channel index: 0 (0x00) Protocol CAN 2.0, auto start false, normal ack, arbitration sample point: 80 % (0x08) Arbitration baud rate: 1 MBaud (0x03) Arbitration SJW: 1 (0x00)	02 60 06 00 00 08 03 00 FF FF 6F 03 Response: 02 60 01 00 00 61 03
Configure CAN echo Rx echo: ON Tx echo: ON	02 66 02 00 00 03 6B 03 Response: 02 66 01 00 00 67 03
Start CAN channel	02 67 01 00 00 68 03 Response: 02 67 01 00 00 68 03
Transmit CAN frame Channel index: 0 (0x00)	02 6A 0D 00 00 00 22 02 08 01 02 03 04 05 06 07 08 C7 03

FDF = ESI = BRS = RTR = EXT ID = 0 (0x00) ID: 0x222 DLC: 8 bytes Data: 0x01 0x02 0x03 0x04 0x05 0x06	Response: 02 6A 01 00 00 6B 03
Asynchronous response when frame transmitted Channel index: 0 (0x00) FDF = ESI = BRS = RTR = EXT ID = 0 (0x00) Timestamp: 2 115 042 µs (0x00000000002045E2) ID: 0x222 DLC: 8 bytes Data: 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 Note: in order to receive this message, you must have other CAN device connected to the bus (or you will receive Acknowledge error)	No request Response: 02 6A 15 00 00 00 E2 45 20 00 00 00 00 00 22 02 08 01 02 03 04 05 06 07 08 16 03

5.4. CAN FD configuration (device acting as CAN FD interface)

Command	Bytes [hex]
Write CAN configuration Channel index: 0 (0x00) Protocol ISO CAN FD, auto start false, normal ack, arbitration sample point: 80 % (0x48) Arbitration baud rate: 500 kBaud (0x02) Arbitration SJW: 1 (0x00) Data baud rate: 2 MBaud, data SJW: 1 (0x10) Data sample point: 80 % (0x08)	02 60 06 00 00 48 02 00 10 08 C8 03 Response: 02 60 01 00 00 61 03
Configure CAN echo Rx echo: OFF Tx echo: ON	02 66 02 00 00 02 6A 03 Response: 02 66 01 00 00 67 03
Start CAN channel	02 67 01 00 00 68 03 Response: 02 67 01 00 00 68 03
Transmit CAN FD frame Channel index: 0 (0x00) FDF = BRS = 1, ESI = RTR = EXT ID = 0 (0x14) ID = 0x333 DLC: 16 bytes Data: 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0x0B 0x00 0x00 0x00 0x00 0x00	02 6A 15 00 00 14 33 03 10 01 02 03 04 05 06 07 08 09 0A 0B 00 00 00 00 00 1B 03 Response: 02 6A 01 00 00 6B 03
Asynchronous response when frame transmitted Channel index: 0 (0x00) FDF = BRS = 1, ESI = RTR = EXT ID = 0 (0x14) ID = 0x333 DLC = 16 bytes Data: same as in the preceding request	No request Response: 02 6A 1D 00 00 14 6E 9B 65 0A 00 00 00 00 33 03 10 01 02 03 04 05 06 07 08 09 0A 0B 00 00 00 00 00 9B 03

5.5. T1 diagnostic

Command	Bytes [hex]
Read T1 status port 1, link-up, mode master, polarity no inversion, normal operation mode	02 70 01 00 01 72 03 Response: 02 70 02 00 01 03 76 03
Read SQI port 1, SQI = 8	02 71 01 00 01 73 03 Response: 02 71 02 00 01 08 7C 03
Cable Test port 0, open circuit	02 72 01 00 00 73 03 Response: 02 72 02 00 00 01 75 03

6. Contact

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

Company Registration: 29413893
VAT no.: CZ29413893