MACH-ETH
Gateway Communication Protocol Specification

For firmware v1.2

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

Changes

Date	Change	Changed by
16.5.2022	CAN frame timestamp; CAN start and stops commands modified so that both CAN channels can be started by a single command	PK
11.3.2022	Fixes and clarifications	PH, MM
13.1.2022	Initial release	PK, KH, MM

Contents

1. Introduction 2
2. Communication Protocol 2
   2.1. Message Overview 3
   2.2. Error Codes 4
3. Message Specification 6
   3.1. General Response 6
   3.2. Error Response 6
   3.3. Device Messages 6
   3.4. Ethernet Messages 7
   3.5. LIN Messages 9
   3.6. CAN and CAN FD Messages 16
   3.7. IO Control Messages 33
   3.8. Miscellaneous Messages 34
4. Communication Examples 35
   4.1. LIN 35
   4.2. CAN 35
   4.3. CAN FD 35
5. Contact 37

1. Introduction

The MACH-ETH is a device that can realize an interface for accessing LIN and CAN(/FD) channels over Ethernet or USB. This document describes a binary protocol used over Ethernet TCP/IP and USB VCP.

2. Communication Protocol

The communication between the MACH-ETH and a user system is based upon a binary protocol over Ethernet / 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.

TCP default: IP address 192.168.1.100, subnet mask 255.255.255.0, port 8000
TCP server 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 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.

2.1. Message Overview

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

ID	Name	Request Data Length (bytes)	Response Data Length (bytes)	Description
0x01	BOOT_UP	- (no request needed)	0	A notification that the gateway was powered up. Sent only to 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	13	Restore the default communication configuration (see Communication Protocol)
0x15	ETH_READ_CONFIGURATION	0	13	Read configuration
0x16	ETH_WRITE_CONFIGURATION	7	0	Write configuration
0x17	ETH_READ_IP_ADDRESS	0	5	Read IP address and mask
0x18	ETH_WRITE_IP_ADDRESS	5	0	Write IP address and mask
0x19	ETH_READ_PORT	0	2	Read TCP port
0x1A	ETH_WRITE_PORT	2	0	Write TCP port
0x1B	ETH_READ_MAC_ADDRESS	0	6	Read MAC address
LIN configuration				
0x20	LIN_WRITE_CONFIGURATION	1	0	Configure LIN channel
0x21	LIN_READ_CONFIGURATION	0	1	Read LIN channel configuration
0x22	LIN_SAVE_CONFIGURATION	0	0	Save LIN configuration to EEPROM
0x23	LIN_LOAD_CONFIGURATION	0	0	Load LIN configuration from EEPROM
0x24	LIN_DEFAULT_CONFIGURATION	0	0	Load LIN default configuration
0x30	LIN_START	0	0	Start LIN channel
0x31	LIN_STOP	0	0	Stop LIN channel
0x32	LIN_ECHO_CONF	1	0	Configure LIN echo
0x40	LIN_MASTER_RESPONSE_TX	3 to 10	0	Transmit LIN Header + Response *Available when the device is configured as a Master*
0x41	LIN_MASTER_REQUEST_TX_RX	1	0 to 11	Transmit LIN Header + Receive Slave Response *Available when the device is configured as a Master*
0x50	LIN_SLAVE_RESPONSE_CONFIG	2 to 10	0	Configure Slave Response buffer
0x51	LIN_SLAVE_RESPONSE_TX_RX	- (no request needed)	2 to 11	A Slave response received or transmitted from/onto the bus.

CAN configuration				
0x60	CAN_WRITE_CONFIGURATION	6	0	Configure CAN channel
0x61	CAN_WRITE_CONFIG_TIM	9	0	Configure CAN channel set time quanta
0x62	CAN_READ_CONFIGURATION	1	12	Read CAN channel configuration
0x63	CAN_SAVE_CONFIGURATION	1	0	Save CAN configuration to EEPROM
0x64	CAN_LOAD_CONFIGURATION	1	0	Load CAN configuration from EEPROM
0x65	CAN_DEFAULT_CONFIGURATION	1	0	Load CAN default configuration
0x66	CAN_ECHO_CONF	2	0	Enable/Disable Tx echo
0x67	CAN_START_CHANNEL	1	0	Start CAN channel
0x68	CAN_STOP_CHANNEL	1	0	Stop CAN channel
0x69	CAN_GET_TIMESTAMP	1	9	Get time in microsecond from startup of channel
0x70	CAN_TRANSMIT_FRAME	5 to 71	1	Send CAN message
0x71	CAN_RECEIVED_FRAME	- (no request needed)	13 to 79	Received CAN message
I/O control				
0xE0	IO_WRITE	1	0	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 / TCP bootloader
0xFF	GENERAL_ERROR	-	1	An error occurred, see Error Codes for description.

2.2. Error Codes

The following tables describes error codes

Error Code	Comment
Communication protocol errors	
0xA0	Incorrect end byte on the Ethernet protocol
0xA1	Bad checksum on the protocol
0xA2	Unknown message ID
0xA3	Too large or incorrect data length
LIN bus errors	
0xB0	Checksum error
0xB1	Bus error
0xB2	Timeout overrun
0xB3	Buffer Full
CAN bus errors	
0xC0	CRC Error
0xC1	Bit Error
0xC2	Form Error
0xC3	Acknowledge Error
0xC4	Bit Stuff Error
0xC5	Buffer Full
General Bus errors	

0xF0	Configuration Error
0xF1	Channel running, channel should be stopped when is configure

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.

DATA 0
Error code

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

3.4. Ethernet Messages

3.4.1.Restore Default Configuration

MessageId = 0x14

Reset the default communication configuration – IP address, mask and port.
Request: Message without data.
Response: acknowledge.

3.4.2.Read and Write Configuration

MessageId = 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.

Request 0x15:
No data

Response:

DATA 0 – DATA 3	DATA 4	DATA 5 – DATA 6	DATA 7 – DATA 12
IP address	Mask	Port	MAC Address

Request for 0x16:

DATA 0 – DATA 3	DATA 4	DATA 5 – DATA 6
IP address	Mask	Port

Response: acknowledge.

3.4.3.Read and Write IP Address Configuration

MessageId = 0x17 for read, 0x18 for write

This command is for reading or writing device IP address and subnet mask. 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: acknowledge.
Response to ID 0x17 has same structure as request 0x18 above.

3.4.4.Read and Write TCP 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.
Request 0x19:

DATA 0 – DATA 1
Port

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

3.4.5.Read MAC Address

MessageId = 0x1B

This command is for reading device MAC address. Each device has unique MAC address which cannot be changed by the user.

Request 0x1B:

No data

Response:

DATA 0 – DATA 5
MAC address

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 two baud rates (9600 and 19200).

Request:

DATA 0
Configuration Register

Configuration Register:

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

Bit 7 **Reserved**

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
Error response:
Gateway cannot be reconfigured
Possible reasons:

- 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

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 disabled; 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 (default)

1 – Echo On

Bit 0 **RX Echo on/off**

0 – Echo Off

1 – Echo On (default)

Response:
No data

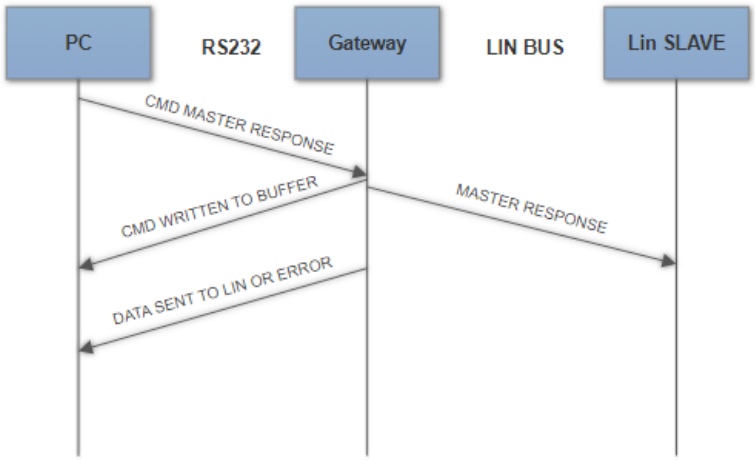
3.5.9. Transmit Master Response

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:
Master response sent
No data

Error response:

- Command cannot be processed
- Error occurred while sending frame with this

3.5.10. 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

Type of response:

1) **Data written to LIN buffer**
   No data

2) **Received data from slave when gateway is MASTER**

DATA 0 - status	DATA 1	DATA 2	DATA 3	DATA n
0x02	LIN ID	LIN DATALEN	LIN DATA 0	LIN DATA n

Error response:

Request cannot be sent, or slave doesn't respond.

3.5.11. Slave Response Configuration

MessageId=0x50

Request:

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.

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 3.5.12 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 3.5.12 for notification about this event.

Bit 5..0 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

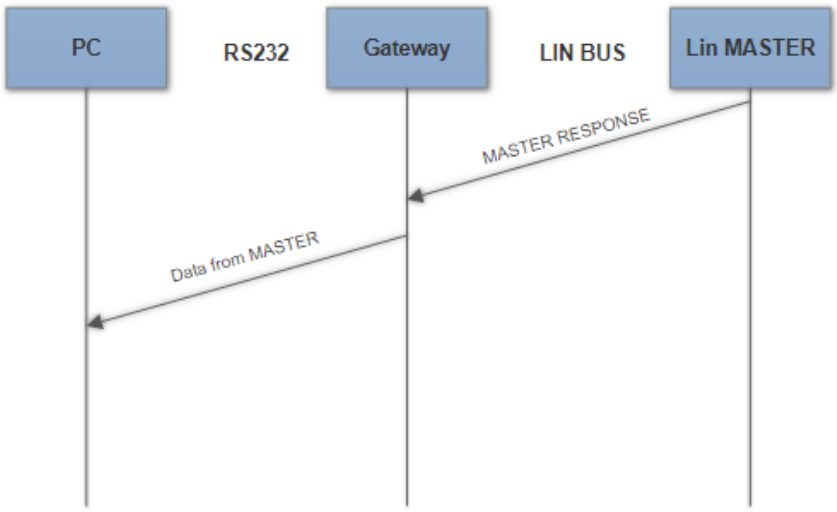
Error response:
Data cannot be written to the Slave Response buffer

Possible reasons:
- Buffer Direction is 1 (TX) but the DataLen is 0.

3.5.12. Slave Response

MessageId=0x51

When the gateway is configured as a Slave and it receives or transmits a Slave Response from/onto the bus, it sends a notification to the user.



Type of responses:

1) Received data from Slave Response from the bus

DATA 0 - status	DATA 1	DATA 2	DATA 3	DATA n
0x01	LIN ID	LIN DATALEN	LIN DATA 0	LIN DATA n

2) Transmitted Slave Response onto the bus
Slave Response containing data from Message Buffer has been sent onto the bus.

DATA 0 - status	DATA 1
0x02	LIN ID

Error response:
Bus error

3.6. CAN and CAN FD Messages

3.6.1.Channel Configuration

MessageId=0x60

This message configures a CAN(/FD) channel. The time quantas for CAN FD controller are chosen by given sample point and baud rate. Sample point cannot always be set exactly to the desired value. The closest value is used instead of it. The actual time quanta setting can be obtained by **Read Configuration** command. The CAN FD controller clock is 40 MHz. The Data Phase baud rate value of 8 MBd isn't stable due to a low number of time quantas (5) and may cause communication errors.

Request:
Data section of a packet:

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

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:

1) CAN channel reconfigured
   No data = success
2) CAN channel cannot be reconfigured
   General Error Message = error

   Possible reasons:
   - Wrong arbitration or data jump width.

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

Request:
Data section of a packet:

DATA 0	DATA 1	DATA 2	DATA 3	DATA 4
Channel	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)**	

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
DTSEG1_7	DTSEG1_6	DTSEG1_5	DTSEG1_4	DTSEG1_3	DTSEG1_2	DTSEG1_1	DTSEG1_0

Bit 0..7 **Data time segment 1**

0000 0000 – 1
0000 0001 – 2
0000 0010 – 2
0000 0011 – 3
…
1111 1111 – 256

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

Response:
1) CAN channel reconfigured
   No data = success
2) CAN channel cannot be reconfigured
   General Error Message = error

Possible reasons:
- 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

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**
Configuration Register 1	Configuration Register 2	Configuration Register 3	Configuration Register 6	Configuration Register 7	Configuration Register 8
DATA 6	**DATA 7**	**DATA 8**	**DATA 9**	**DATA 10**	**DATA 11**
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)**	Echo configuration Register 13

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 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 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 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

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

bit 7							bit 0
DTSEG1_7	DTSEG1_6	DTSEG1_5	DTSEG1_4	DTSEG1_3	DTSEG1_2	DTSEG1_1	DTSEG1_0

Bit 0..7 **Data time segment 1**

0000 0000 – 1
0000 0001 – 2
0000 0010 – 2
0000 0011 – 3
…
1111 1111 – 256

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 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 (default)

1 – Echo On

Bit 0 **RX Echo on/off**

0 – Echo Off

1 – Echo On (default)

3.6.4.Save Configuration

MessageId=0x63

Request 0x63:

DATA 0
Channel

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

No data = success
General Error Message = error

3.6.5.Load Configuration

MessageId=0x64

Request 0x64:

DATA 0
Channel

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

Type of response:
Default configuration was saved
No data = success

General Error Message = error

3.6.6.Default Configuration

MessageId=0x65

Request 0x65:

DATA 0
Channel

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

Type of response:
Default configuration was saved
No data = success

General Error Message = error

Possible reasons:

- CAN channel is already running.

3.6.7.Frame Echo Configuration

MessageId=0x66

Request 0x66:

DATA 0	DATA 1
Channel	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)

Type of response:

Echo configuration was changed

No data = success.

General Error Message = error

Possible reasons:

- CAN channel is already running.

3.6.8.Start Channel

MessageId=0x67

Request 0x67:

DATA 0
Channel

DATA 0

Bit 0..7 **Channel**

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

Response:

DATA 0	DATA 1
Channel	Error Code

IF Error Code == 0 CAN channel was successfully started

Else CAN Error Code == General Error Message

Possible reasons:

- CAN channel is already running.

If all channels option is set

Response:

DATA 0	DATA 1	DATA 2	DATA 3
Channel 0	Error Code	Channel 1	Error Code

3.6.9.Stop Channel

MessageId=0x68

Request:

DATA 0
Channel

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

Response:

DATA 0	DATA 1
Channel	Error Code

IF Error Code == 0 CAN channel was successfully started

Else CAN Error Code == General Error Message

If all channels option is set

Response:

DATA 0	DATA 1	DATA 2	DATA 3
Channel 0	Error Code	Channel 1	Error Code

3.6.10. Get Channel Timestamp

MessageId=0x69

Request:

DATA 0
Channel

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

Response:

DATA 0	DATA 1...8
Channel	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

Timestamp represents the time from startup of CAN(/FD) channel to transmit the frame in microseconds.

3.6.11. Transmit Frame

MessageId=0x70

This message transmits CAN frame. The structure of frame is different when Extended ID is set. Without extended ID frame is header 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	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	MESSAGE_INFO	ID0	ID1	ID2	ID3	DLC	DATA

Timestamp represents the time from startup of CAN(/FD) channel to transmit the frame in microseconds.

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 7..5 **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 represent the time from startup of CAN(/FD) channel in microseconds. The bit order in message is LSB.

Response:
CAN Frame was successfully sent:
No data = success
CAN channel error:
General Error Message = error

Possible reasons:
- 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 n
Channel	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 4	DATA 5
Channel	MESSAGE_INFO	Timestamp byte 0…7	ID0	ID1	ID2	ID3

DATA 12	DATA n
DLC	DATA

3.6.12. Receive Frame

MessageId=0x71

Message response has similar structure as Transmit Frame. It only differs with added timestamp bytes (bytes 2…9). Timestamp represents the time from startup of CAN(/FD) channel to receive the frame in microseconds. For this message no request is needed, it appears when frame from other CAN unit arrived. Error frames are interpreted like **CAN bus errors** (0xC0 – 0xC4).

Request:
IF EXTId==0

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

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, port or MAC address.

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. 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. Recommended web browser for firmware upload is Google Chrome. File must be in the binary format (.bin).

Request:

DATA 0
Bootloader selection

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

Response
No data

4. Communication Examples

4.1. LIN

Command	Bytes [hex]
Configure LIN channel 19200, Master, Enhanced Checksum, Auto-Message Length	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	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 01 00 01 33 03

4.2. CAN

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 00 00 60 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 00 00 61 03
Start CAN channel Channel 0	02 67 01 00 00 68 03 Gateway response: 02 67 02 00 00 00 69 03
Transmit CAN Frame Channel 0, format = CAN, ID = 0x01FF, DLC = 7, Data (hex) = 05 04 50 06 06 08 14	02 70 0C 00 00 00 FF 01 07 05 04 50 06 06 08 14 04 03 Gateway response: 02 70 00 00 70 03
Stop CAN channel Channel 0	02 68 01 00 00 69 03 Gateway response: 02 68 02 00 00 00 6A 03

4.3. CAN FD

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 00 00 60 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	02 61 09 00 00 60 0E 03 03 01 0E 13 00 00 03 Gateway response: 02 61 00 00 61 03

T_seg1 = 5, Data T_seg2 = 1, Arbitration SJW = 1, Data Prescaler = 1	
Start CAN FD channel Channel 0	02 67 01 00 00 68 03 Gateway response: 02 67 02 00 00 00 69 03
Transmit CAN FD Frame Channel 0, format = CAN FD, BRS, ID = 0x01FF, DLC = 7, Data (hex) = 05 04 50 06 06 08 14	02 70 0C 00 00 14 FF 01 07 05 04 50 06 06 08 14 18 03 Gateway response: 02 70 00 00 70 03
Stop CAN FD channel Channel 0	02 68 01 00 00 69 03 Gateway response: 02 68 02 00 00 00 6A 03

5. Contact

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

Company Registration: 29413893
VAT no.: CZ29413893