100BASE-T1 Media Gateway
CAN(/FD) Gateway Application Note

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

Changes

Date	Changes	Changed by
05.08.2024	Initial version	PK

Contents

List of Figures

1 Introduction

The **100BASE-T1 Media Gateway** offers Ethernet – CAN(/FD) Gateway function. The following outlines the communication protocol used by the Gateway function.
The Media Gateway features three 100BASE-T1 Single-Pair Ethernet ports via the D-SUB connector (pins 1+2, 4+5, and 6+7), one 10/100/1000BASE-T port through an RJ-45 connector, two CAN channels with CAN FD support, and one USB port with virtual COM port (VCP) profile.
The CAN gateway function allows to forward received CAN(/FD) frames onto either Ethernet or USB.

When the CAN-Ethernet gateway is configured, CAN frames received by the device are converted into Ethernet frames with the structure defined in 3.3 and then transmitted via the internal Ethernet switch to either the 100BASE-T1 or 1000BASE-T port. CAN frames can be transmitted onto the CAN bus by sending Ethernet frames with the structure defined in 3.1 and 3.2 to the gateway.

Master/Slave settings for the T1 ports can be set over the on-board DIP switches, over the device's web site, or remotely via the communication protocol. CAN channels configuration can be set over the device's web site, or remotely via the communication protocol. Both the 100BASE-T1 and CAN configurations can be stored into the internal non-volatile memory so that it is loaded on power-up.

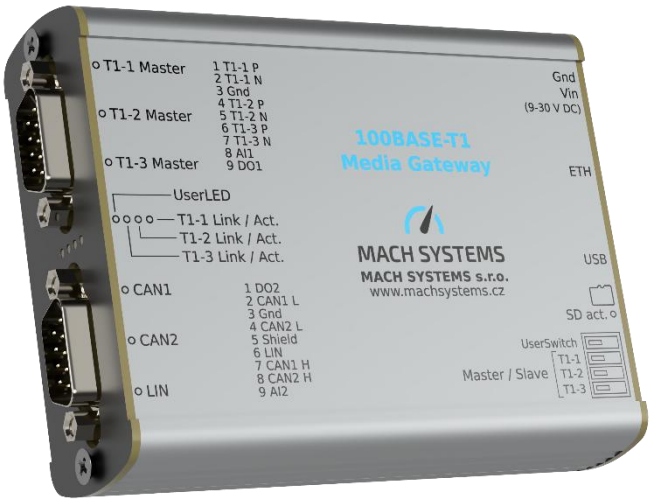

Figure 1 100BASE-T1 Media Gateway

2 CAN Gateway Function

CAN and CAN FD messages, upon conversion to Ethernet packets, are represented by the format outlined in the following sections.
The device's gateway port (8001 by default) is always by one higher than the Communication protocol port (8000 by default).
To transmit a CAN/CAN FD message, an Ethernet packet must be transmitted onto the gateway port. For TCP, the device always acts as a server. For UDP, both the destination IP address and port need to be configured by the user.

The gateway function for each CAN(/FD) channel can be configured over the embedded web server (see Figure 2).

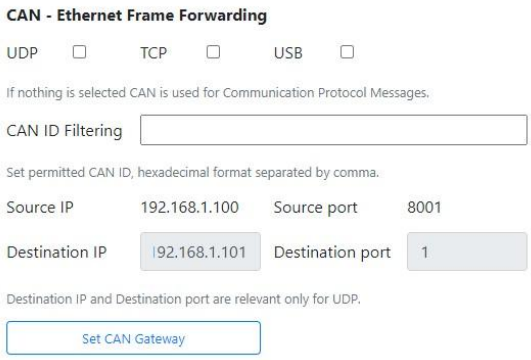

Figure 2 Ethernet - CAN Gateway configuration over web

2.1 Ethernet

Ethernet default configuration:

IP address	**192.168.1.100**
Subnet mask	255.255.255.0
Default gateway	192.168.1.1
Port for Ethernet - CAN Gateway	**8001**
Port for communication protocol	8000
Port for embedded web server	80
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 over the communication protocol or over the embedded web server (Google Chrome browser is recommended).

TCP packet structure:

Ethernet Header (14B)	**IP Header (20 B)**	**TCP Header (20-60B)**	**DATA (X B)**
Destination address + Source address + Length	Destination IP address (4B)	TCP Destination Port (2B)	CAN Message Rx (xB), CAN Message Tx or CAN Error Packet(xB)

UDP packet structure:

Ethernet Header (14B)	**IP Header (20 B)**	**UDP Header (8B)**	**DATA (X B)**
Destination address + Source address + Length	Destination IP address (4B)	UDP Destination Port (2B)	CAN Message Rx (xB), CAN Message Tx or CAN Error Packet(xB)

The rest of the documentation refers to DATA part only. The Destination IP address and Destination Port can be changed by the Communication protocol or the web page.

2.2 Virtual COM port

The virtual COM port is available through the USB connection. The VCP is primarily used for communication messages that are described in *100BASE-T1 MG Communication Protocol Specification*. However, the VCP can be also used for the USB-CAN Gateway function. The **data section is the same** as the Ethernet packet data section (see 2.1).
VCP configuration: 115200 Baud, 8 data bits, no parity, 1 stop bit.

VCP (USB) packet structure:

DATA (X B)
CAN Message Rx (xB), CAN Message Tx (xB) or CAN Error Packet(11B)

2.3 CAN bus

All CAN bus settings (such as baud rate, sample point et.c) can be changed over the web or the Communication protocol. Also, the CAN to Ethernet frame forwarding can be configured the web or the Communication protocol. You can set the destination channel which the CAN frames are transmitted onto (USB, TCP and UDP). If nothing is selected, the gateway function is disabled. The UDP and TCP cannot be selected simultaneously. The CAN frame forwarding can be restricted to permitted CAN IDs set up in CAN ID Filtering.

3 Message Specification

3.1 Transmit CAN Frame

This message transmits a CAN(/FD) 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.
Once the frame has been transmitted on the CAN bus, an echo “Receive CAN frame ”message is transmitted onto Ethernet.

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

DATA 0	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5	DATA 6 - n
Message Type	Channel	Message Flags	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	DATA 8 - n
Message Type	Channel	Message Flags	ID0	ID1	ID2	ID3	DLC	DATA

Message type: 0x20

Channel: must be set to 0

Message Flags:

bit 7							bit 0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	RTR	EXT ID

- 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.

ID: CAN ID Little endian format

DLC: Number of data bytes 0-8.

3.2 Transmit CAN FD Frame

This message transmits a CAN FD 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.
Once the frame has been transmitted on the CAN bus, an echo "Receive CAN frame "message is transmitted onto Ethernet.

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

DATA 0	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5	DATA 6 - n
Message Type	Channel	Message Flags	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	DATA 8 - n
Message Type	Channel	Message Flags	ID0	ID1	ID2	ID3	DLC	DATA

Message type: 0x21

Channel: must be set to 0

Message Flags:

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

- 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 0: **EXT ID**
    - 0 – Standard ID. Request without data is 5 bytes.
    - 1 – Extended ID. Request without data is 7 bytes.

ID: CAN ID Little endian format

DLC: Number of data bytes 0-64.

3.3 Receive CAN(/FD) Frame

Message response has similar structure as Transmit Frame. The only difference is the timestamp bytes (data bytes 2 to 9) are added. The message Timestamp represents the time from startup of CAN(/FD) channel to reception of the frame in microseconds. This message is automatically forwarded onto the gateway channel once a CAN(/FD) frame is received.

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 13	DATA 14-n
Message Type	Channel Number	Message Flags	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	DATA 3...10	DATA 11
Message Type	Channel Number	Message Flags	Timestamp Byte 0 - 7	ID0
DATA 12	**DATA 13**	**DATA 14**	**DATA 15**	**DATA 16-n**
ID1	ID2	ID3	DLC	DATA

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

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

3.4 Receive CAN Error Frame

This message is sent asynchronously when there is some error related to CAN (typically the CAN error frame).
Response:

DATA 0	DATA 1	DATA 2	DATA 3...10
Message Type	Channel number	Error type	Timestamp byte 0 – 7

Message type: 0x30

Channel number: Always 0

Error type:

- 0: Bit Stuff Error
- 1: Form Error
- 2: Acknowledge Error
- 3: Bit Error
- 4: CRC Error
- 5: TX Buffer full

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

4 Message Examples

The bytes below represent the DATA section of packet **only**.

Command	Bytes [hex]
Transmit CAN Frame Channel 0, format = CAN, ID = 0x01FF, DLC = 7, Data (hex) = 05 04 50 06 06 08 14	0x20 0x00 0x00 0xFF 0x01 0x07 0x05 0x04 0x50 0x06 0x06 0x08 0x14 0xFE
Transmit CAN FD Frame Channel 0, format = CAN FD, BRS, ID = 0x01FF, DLC = 7, Data (hex) = 05 04 50 06 06 08 14	0x21 0x00 0x14 0xFF 0x01 0x07 0x05 0x04 0x50 0x06 0x06 0x08 0x14 0x12
Receive CAN Frame Channel 0, format = CAN, ID = 0x01FF, DLC = 7, Data (hex) = 05 04 50 06 06 08 14, Timestamp = 91 601 865µs (1.526698 min)	0x20 0x00 0x00 0x09 0xbb 0x75 0x05 0x00 0x00 0x00 0x00 0xff 0x01 0x07 0x05 0x04 0x50 0x06 0x06 0x08 0x14
Receive CAN Error Frame Channel 0, Error type = 0 (Bit Stuff Error), Timestamp = 652 185 335µs (10.869756 min)	0x30 0x00 0x00 0xf7 0x8e 0xdf 0x26 0x00 0x00 0x00 0x00

5 Contact

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

Company Registration: 29413893
VAT no.: CZ29413893