T-BUS Protocol

revision 4.2.20

Protocol description

T-BUS – protocol (heheinafter reffered just as "protocol"), developed by "TEC electronics", allows 2 way exchange between additional equipment and CAN bus of the vehicle. Protocol allows to receive system state from the CAN bus, and send control commands into vehicles CAN bus.

It is an open protocol and it is supported by "TEC electronics".

Terms

Unit – device made by "TEC electronics", which supports the protocol. Connects to CAN bus of the vehicle and additional equipment.

Receiver – device which connects to the unit through UART via protocol.

CANDRIVER – software block, which processes data from the CAN bus of the vehicle. Send commands to CAN bus. Part of the units firmware.

Physical level

Protocol is used in "point-to-point" topology. Protocol is duplex and asynchonous. Protocol parameters: UART, 8N1, 9600 baud, 2 wires.

Protocol is SPI (Serial peripheral interface). Every bit is a one of two possible levels – "0" и "1". Every byte is a uniterrupted sequence of bits, least significant bit (LSB) first:

- Start-bit. always "0"
- Bits 0-7 of data byte
- Stop-bit. Always "1"

Frequency: f=9600 bits/s.

Time to transmit 1 bit: Tbit=1/f±5%=104,2±5% mks.

Time to transmit 1 byte: Tbyte=Tbit*10±5%=1042±5% mks.

Data is transmitted in packets. There are no time dividers between packets.

Physcal layer and electrical characteristic depend on a Unit and described in its documentation.

Transport layer

Data between two devices is transferred in packets. At the beginning and the end of each packet a special byte 0xAA (packet divider) should be transmitted. Packet divider cannot be present in the middle of the packet. For it there is a special escape-byte 0xA8, to encode 0xAA.

Every 0xAA and 0xA8 byte in original packet should be replaced with a sequence of 2 bytes: escape-byte 0xA8 and source byte with inverted 0 bit. So, 0xAA byte is encoded with sequence 0xA8AB, and byte 0xA8 – 0xA8A9.

Packet (Amount of bytes between dividers) should not be more than 32 bytes.

Unique byte 0xAA allows to drop pauses as dividers, and from pause control. Framing packet with dividers allow to not count bytes in a packet. Example of using escape-byte with division to packets can be seen on figure 1. Example algorithm of packet reception can be seen in figure 2. In this example bytes 0xAA and 0xA8 decoded upon reception, without transferring escape-byto to incoming buffer. This allows to save space for by not using buffer

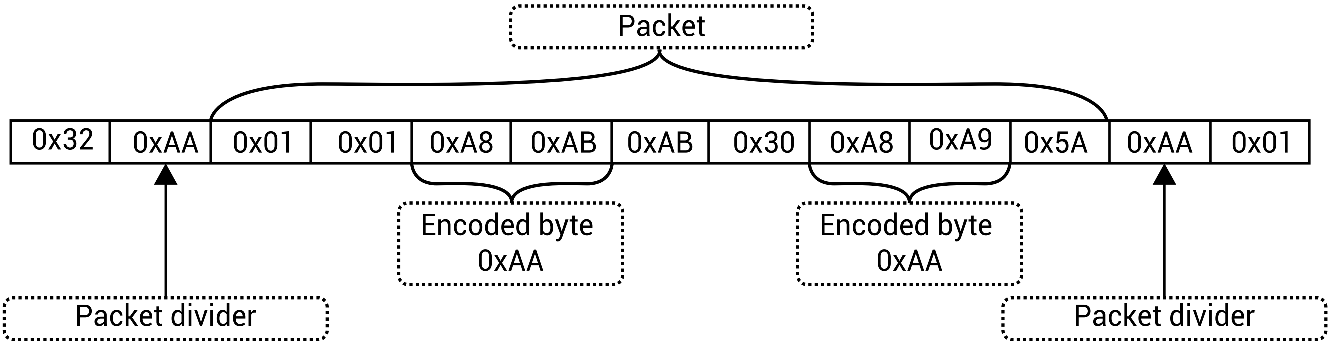

Figure 1. Dividing datastream to packets

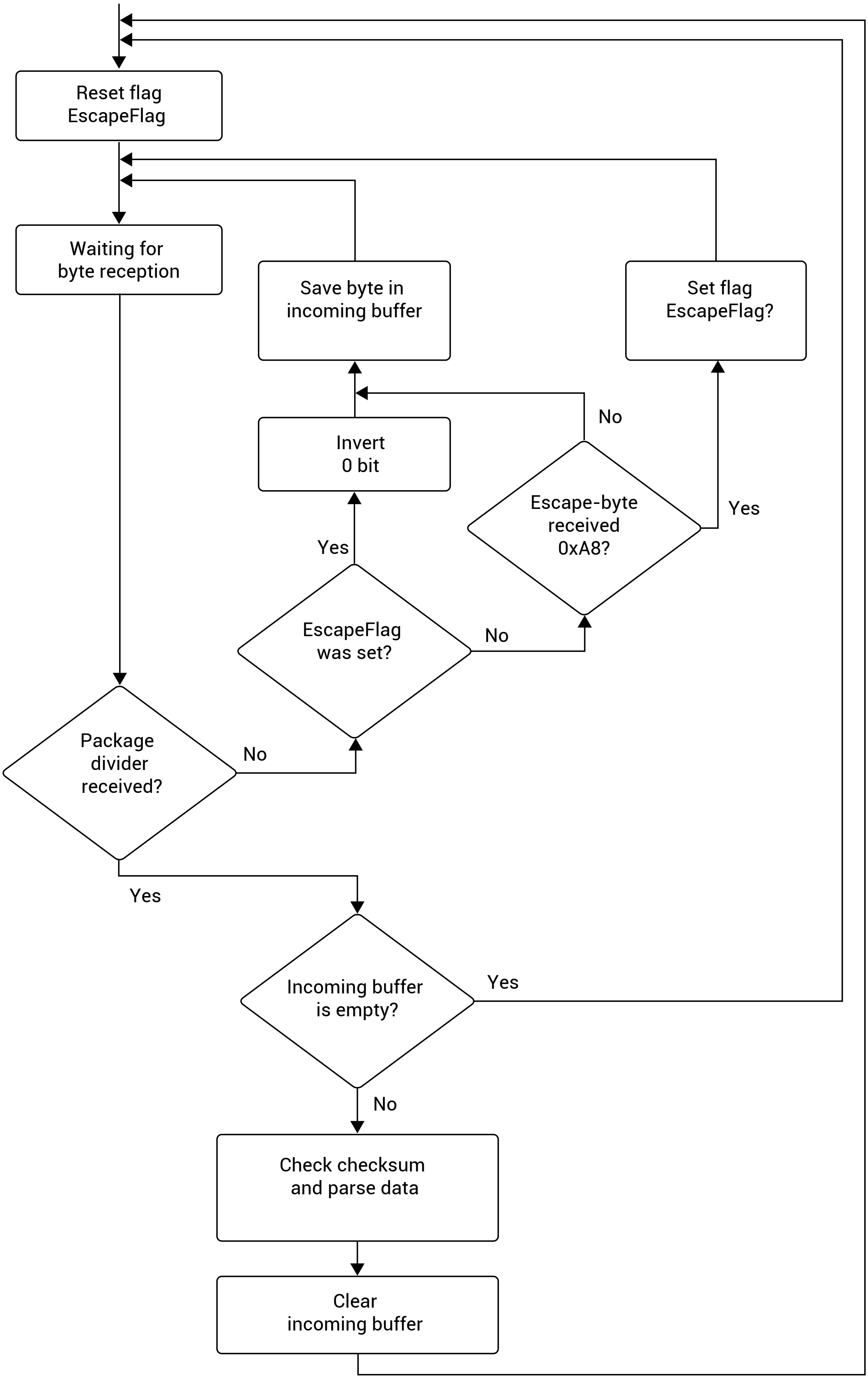

Figure 2. Scheme of possible reception algorithm

Interaction algorithm

1. Packet transfer

Packets are divided to state, command and confirmation packets. Command packets are sent asynchronously, at any given time. Confirmation packets are send once as a response to command packets. State packets are send in line from time to time. Packets are described in table 2.

2. Low power consumption mode

Unit can has 2 operation states: active and low power consumption mode(or Sleep mode).

Unit goes into Sleep mode after T_{sleep} after CAN bus of the vehicle stops communication, stopping transferring of all packets. T_{sleep} = 60 seconds. 3 before going into Sleep mode corresponding bit will be set in "State of main systems in the vehicle". Unit will leave Sleep mode when CAN bus becomes active again. Leaving Sleep mode can take some time (up to 2 seconds). This delay happens because of the interaction algorithms between CANDRIVER and CAN bus of the vehicle.

Unit can be forced to leave Sleep mode by the Receiver. To do so Receiver has to transfer through T-BUS any packet, but the packet won't be parsed by the Unit.

Recommend seqquence to awake Unit:

1. Send packet "Commands to control unit" with zeroed Data field(see «Packet structure»).
2. Wait for the first correct packet.
3. Transmit packet with required command.

Packet structure

Structure of packets used by the protocol can be found in table 1.

Table 1. Packet structure

Packet field			
Protocol identificator Protocol_ID (1 byte)	Packet indetificator (Packet name) Packet_ID (1 byte)	Data Data (0..13 bytes)	Checksum CRC-8 (1 byte)

Protocol_ID – Protocol identificator. Defines sets of packets with this protocol supports (see table 1). Field – 1 byte.

Packet_ID – Packet identificator. Defines size and datafield structure (Data) of the packet (see table 1). Field size – 1 byte.

Data – main payload. Data structure depends on Protocol identificator and Packet identificator.

Data field contains numerical parameters or state parameters. Numerical parameter can be longer than 2 bits, and can take following values:

- All bits are set to "1" – parameter was not identified
- All bits are set to "1", least significant bit set to "0" – paramenter analysis denied
- All bits are set to "0" – parameter disabled
- Numerical parameter value.

Parameter that are longer than 1 byte, are send from least significant byte to most significant byte (little-endian).

State parameters are defined by 2 bits and can take following values:

- 11 – parameter state are not identified
- 10 – parameter analysis is forbidden
- 00 – parameter is in off state (OFF)
- 01 – parameter is in on state (ON).

At first start all parameters are not identified. Parameter will receive a state after reading data from CAN bus. For example, drivers door will be identified only after it was opened and command was transferred via CAN bus. After performing "reset to factory settings" all parameters will be reset to unidentified state.

All unused data fields in packets will be filled with "parameter was not identified".

Checksum (CRC-8)

To calculate checksum cyclic code CRC-8 with polynomial $x^8+x^5+x^4+1$ and normal representation 0x5A. During calcurlation of checksum all escape-sequences should be decoded. Size – 1 byte.

CRC-8 calculation in C

```
/*
  Checksum calculation function.
  Algorythm: CRC-8
  Polynominal: 0x31    x^8 + x^5 + x^4 + 1
  Representation: 0x5A
  Confirmation value: 0x94 ("123456789")
*/
unsigned char Crc8(unsigned char *pcBlock, unsigned int len)
{
  unsigned char crc = 0x5A;
  unsigned int i;
  while (len--)
  {
    crc ^= *pcBlock++;
    for (i = 0; i < 8; i++)
       crc = crc & 0x80 ? (crc << 1) ^ 0x31 : crc << 1;
  }
  return crc;
}
```

Check packet delivery

For confirmation of succesfull command packet delivery a special confirmation packet is used. Commands are send in a following sequence:

1. Sender (Unit) send a packet with command to control Receiver. This packet contains command number (TxCmdCount).
2. Receiver receives command packet and sends packet with confirmation, which contains same command number(TxCmdCount).
3. Unit awaits for packet with confirmation. If packet with confirmation was received, and it contains command number (TxCmdCount), them packet delivery counts as succesfull. if within Trxtimeout there was no confirmation, then packet delivery will be marked as unsuccesfull and can be send again. Значение Trxtimeout = 100 ms.

Packet description

Table 2. Packet description (Protocol_ID = 0x02)

Packet description	Data field description
Commands to control unit Packet_ID = 0x01	D0.7-D0.0 – Command number *Number of the command which will be send in Packet_ID = 0x02. Used by the Receiver to confirm command*
	D2.7-D1.0 – Command code 0x0001: close central lock *Same as closing central lock from the interior* 0x0002: open central lock *Same as open central lock from the interior* 0x0004: close central lock and arm factory alarm 0x0044: close central lock, arm factory alarm, turn on "Comfort" 0x0005: close central lock without arming factory alarm 0x0045: close central lock without arming factory alarm, turn on "Comfort" 0x0008: open central lock and disarm factory alarm 0x000A: close central lock without disarming factory alarm 0x000C: open drivers door and disarm factory alarm 0x0046: open drivers door without disarming factory alarm 0x0010: open trunk 0x0020: flash with hazard lights 0x0040: start "Comfort" system 0x0080: stop "Comfort" system" 0x1000: imitate open/close drivers door to turn off ACC *While this command is performing all door data from the vehicle will be ignored* 0x0130: reset vehicle model and all settings *After receiving this command unit there will be a software reset* 0x0200: set group and subgroup of the vehicle *After setting there will be a software reset. Group and subgroup have to be set in D3 and D4* 0x0400: software reset of the unit *Reset should take about 5 seconds.* *For this time all fuctions of the system will be stopped* 0x0800: remote start\stop engine via CAN *Command is the same as command from the factory remote «remote start|remote stop of the engine»* 0x8000: start heater 0x9000: stop heater
	D3.7-D3.0 – group number *Should be send only with* "set group and subgroup of the vehicle"
	D4.7-D4.0 – subgroup number *Should be send only with* "set group and subgroup of the vehicle"
Control command confirmation Packet_ID =0x02 *Packet sends 1 time after reception of commands Packet_ID = 0x01.*	D0.7-D0.0 – command number *Command number that should be confirmed*

Packet description	Data field description
State of main systems in the vehicle Packet_ID = 0x03 *Packet is sent from the Unit to Receiver with period Tstat = 60 ms*	D0.1-D0.0 – CAN state (active/deactivated)
	D0.3-D0.2 – left turn signal
	D0.5-D0.4 – right turn signal
	D0.7-D0.6 – Security state
	D1.1-D1.0 – engine start, starter ON
	D1.3-D1.2 – "READY" state of hybrid
	D1.7-D1.4 – not used
	D2.1-D2.0 – not used
	D2.3-D2.2 – drivers door
	D2.5-D2.4 – passengers door
	D2.7-D2.6 – door rear left
	D3.1-D3.0 – door rear right
	D3.3-D3.2 – trunk
	D3.5-D3.4 – hood
	D3.7-D3.6 – stop-signal *with turned off ignition this state is not controlled –* *Value in disabled state - 0x2*
	D4.1-D4.0 – hazard lights
	D4.3-D4.2 – key in the ignition switch
	D4.5-D4.4 – ACC
	D4.7-D4.6 – Ignition (IGN)
	D5.1-D5.0 – state of hazard lights
	D5.3-D5.2 – factory security state alert
	D5.5-D5.4 – command to close central lock from factory remote
	D5.7-D5.6 – command to open central lock from factory remote
	D6.1-D6.0 – command to open trunk from factory remote
	D6.3-D6.2 – central lock state (open/close) 0x0: close 0x1: open 0x2: state is disabled for analysis 0x3: state was not indentified
	D6.5-D6.4 – engine is running *there is a delay up to 0,7 second after launch*
	D6.7-D6.6 – parking brake *with turned off ignition this state is not controlled –* *Value in disabled state - 0x2*
	D7.2-D7.0 – automatic gearbox selection lever 0x1: Park, P 0x2: Reverse, R 0x3: Neutral, N 0x4, 0x5: Drive, D 0x0: state is disabled for analysis *Sets while lever is being switched* *or when ingition turned off* 0x7: state is not identified
	D7.5-D7.4 – sleep *To indicate that mode will be changed to Sleep .* *Will be changed to ON within 3 seconds*
	D7.7-D7.6 – vehicle is moving (spped > 0 km/h)
	D11.7-D8.0 – state of factory buttons (1 button 1 bit)

Packet description	Data field description
Commands to control alternate channels Packet_ID = 0x04 *Packet sends from Unit to Receiver, in random amount of time. After transmitting unit wait for confirmation Tretry = 50 ms. Then packet will be sent again. Confirmation have Packet_ID = 0x05, which will be sent by Receiver.* *This can be repeated up to 3 times*	D0.7-D0.0 – command number *Command number which will be sent in Packet_ID = 0x05. Used by Unit ot confirm execution of command*
	D1.1-D1.0 – signal to turn on/turn off digital output in Receiver to turn on/turn off hazard lights (out connects to hazard lights switch in the vehicle) 0x0: turn on 0x1: turn off
	D1.3-D1.2 – signal to turn on/turn off digital output in Receiver to open/close central lock via alternate control (out connects to central lock switch) 0x0: turn on 0x1: turn off
	D1.7-D1.4 – not used
Confirmation of commands to control alternate channels Packet_ID = 0x05 *Packet sends 1 time after successful reception of Packet_ID = 0x04 by Receiver*	D0.7-D0.0 – command number *Command number which will be sent in Packet_ID = 0x05. Used by Unit ot confirm execution of command*
Operation Data Packet_ID = 0x06 *Packet sends from Unit to Receiver with period Tstat = 500 ms*	D1-D0 – vehicle speed (1 bit = 1 km/h)
	D3-D2 – engine RPM (1 bit = 1 RPM)
	D7-D4 – milleage (1 бит = 5 м)
	D8 – engine temperature. *temperature will be send with 50°C displacement,* *e.g. engine temperature -50°C = 0, -49°C = 1…, 0°C = 50, 1°C = 51*
	D10.5-D9.0 – fuel level in tank ([1 bit = 1 liter]/[1 bit = 0,1%])
	D10.7-D10.6 – dimension of "fuel level in tank" 0x0: percent (%) 0x1: liters (liter)
Request identification parameters Packet_ID = 0x07 *Packet transmitted from Receiver to Unit.* *Responce to this request is a packet with indentification parameters*	D0.7-D0.0 – number of requested identification parameters
Identification parameters №1 Packet_ID = 0x08 *Packet transmitted from Unit to Receiver after request from Receiver.*	D7-D0 – serial number of the device (ASCII)
Identification parameters №2 Packet_ID = 0x09 *Packet transmitted from Unit to Receiver after request from Receiver.*	D3-D0 – Software version
	D1-D0 – low digit of version
	D2 – middle digit of version
	D3 – high digit of version
	D4 – group of the vehicle
	D5 – subgroup of the vehicle

Packet description	Data field description
Identification parameters №3 Packet_ID = 0x0A *Packet transmitted from Unit to Receiver after request from Receiver.*	D3-D0 – software version (bootloader)
	D7-D4 – software version (OS)
Identification parameters №4 Packet_ID = 0x0B *Packet transmitted from Unit to Receiver after request from Receiver.*	D3-D0 – software version (Firmware)
	D7-D4 – software version (CANDRIVER)
Identification parameters №5 Packet_ID = 0x0C *Packet transmitted from Unit to Receiver after request from Receiver.*	D1.6-D0.0 –revision of bootloader
	D1.7 – not used
	D3.6-D2.0 – revision of firmware
	D3.7 – not used
	D5.6-D4.0 – revision of CANDRIVER
	D5.7 – not used
Identification parameters №6 Packet_ID = 0x0D *Packet transmitted from Unit to Receiver after request from Receiver.*	D0 – high digit of protocol version *Same as type of protocol, equal 4*
	D1 – middle digit of protocol version *Same as identificator of protocol, equal 2*
	D3-D2 – low digit of protocol version, equal to 20
State of additional systems Packet_ID = 0x0E *Packet sends from Unit to Receiver with period Tstat = 200 мс*	D0.1-D0.0 – state of the heater 0x0: on 0x1: off 0x3: not avaliable
	D0.7-D0.2 – not used
	D1.7-D1.0 – not used

All identification parameters will be send once after Unit was turned on. Parameters will be sent with period 100 ms in sequence starting with parameter №1.