DiagBox User Manual

VESTEL

HISTORY OF THE DOCUMENT

Edition	Date	Nature of the modification	Author	Reviewer
0.1	14-July-2023	Draft	Alpay Toprak	
0.2	24-July-2023	Updated Features	Alpay Toprak	
0.3	14-Sep-2023	Updated Features	Ozan Kandas	

CONTENTS

1   OVERVIEW

This document's objective is to instruct and guide all DiagBox Tool users. The document includes instructions on how to use, set up, and configure this tool as well as several examples.

2   ABBREVIATION

Abbreviation	Definition
CAN	Controller Area Network
UDS	Unified Diagnostic Services

3   Introduction to the Tool

The DiagBox is a tool designed to perform the following diagnostic tasks:

- Scan and find hardware device (e.g. Intrepid, Vector, or Kvaser) connected to ECU automatically.
- Connection with the ECU thanks to the already-founded hardware device.
- Transmit, receive CAN & UDS messages.
- Log the data for both connections and transmit and receive processes.

4   Installation of the Tool

Below steps should be followed and details should be taken into account.

- The tool can be installed by ".exe" which is shared with the user.
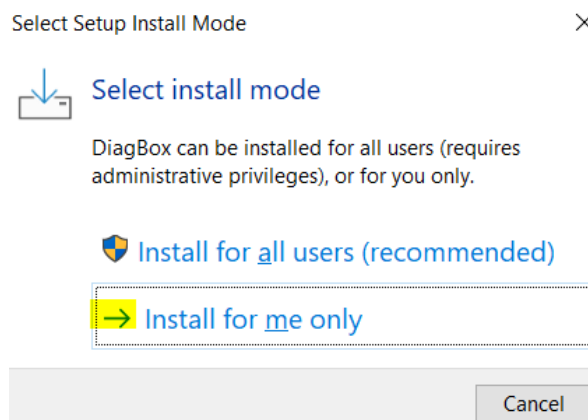- On the installation, install mode should be selected as "Install for me only".

Figure 1. Selection of the Install Mode

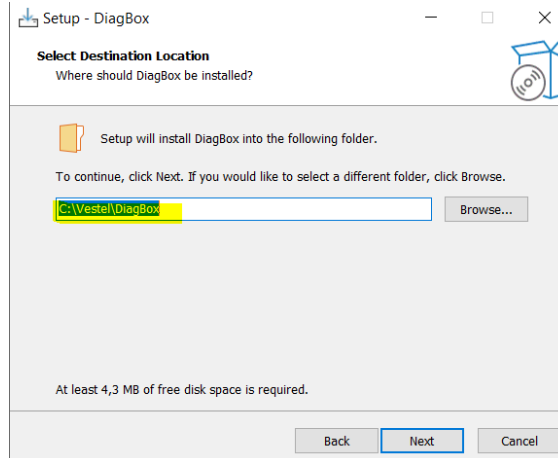- The installation path should be remain as it is.

Figure 2. Installation Path

- After installation is completed, the tool will be asking a key to the user. For the license activation of the tool steps that need to be followed can be checked from the User Guide of that window.

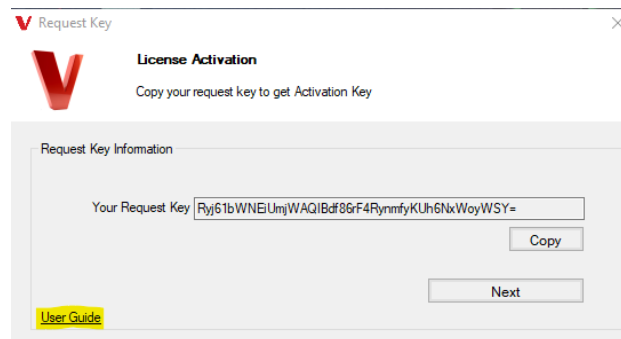Figure 3. Guide for the License Activation

5 Main Window and Features

Each pop-up of the tool can be closed with the "ESC" key.

5.1 Connection Creation

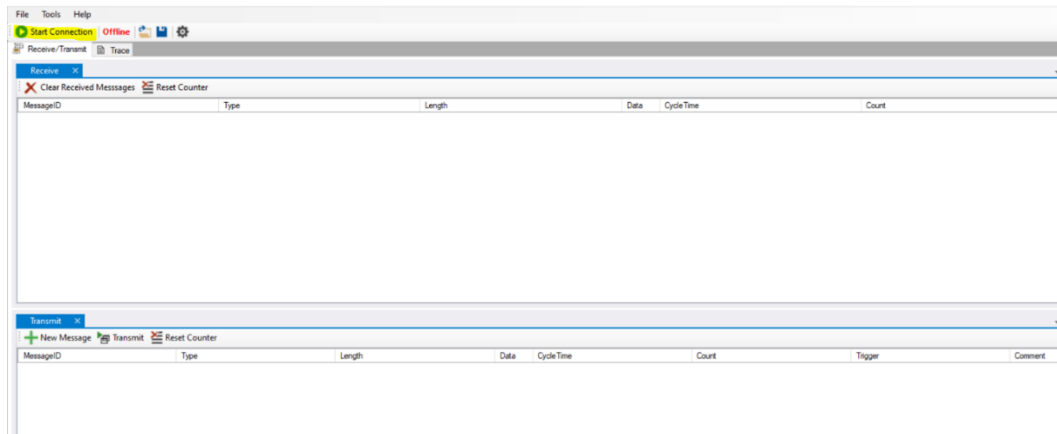The connection can be created with the ECU by clicking on the Start Connection button.

Figure 4. Start Connection

If there is any device connected to the PC, it will be shown in the pop-up window(Figure5), if there is no device found, then pop-up will be opened with the Warning message(Figure 6).
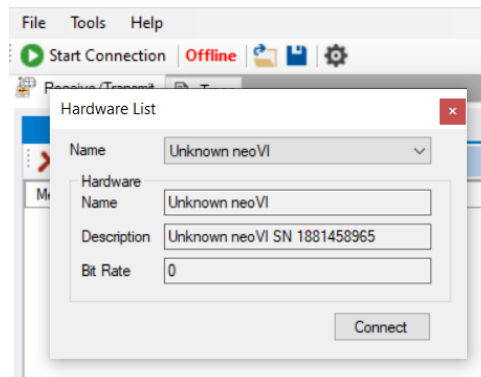

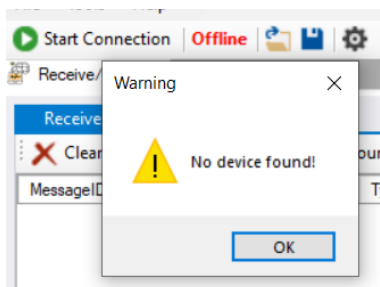Figure 5. Hardware Connected to the PC

Figure 6. No Device Connected

Device can be selected and "Connect" button should be clicked, If the connection is established with the ECU via hardware device, "Start Connection" will be changed to "Stop Connection" and "Offline" will be changed to "Online {device name}".

If there is any issue with connection creation, the pop-up will be opened with the issue message, issue details can be found in the "dev.log" file in the installation path of the tool. Some common and possible issue reasons are listed below:

- Hardware drivers may not be installed.
- Connection with the ECU and hardware may not be wired correctly.

5.2   Save and Open Messages

Messages which were added into the grid of the Transmit panel can be saved into the local and opened for future usage.
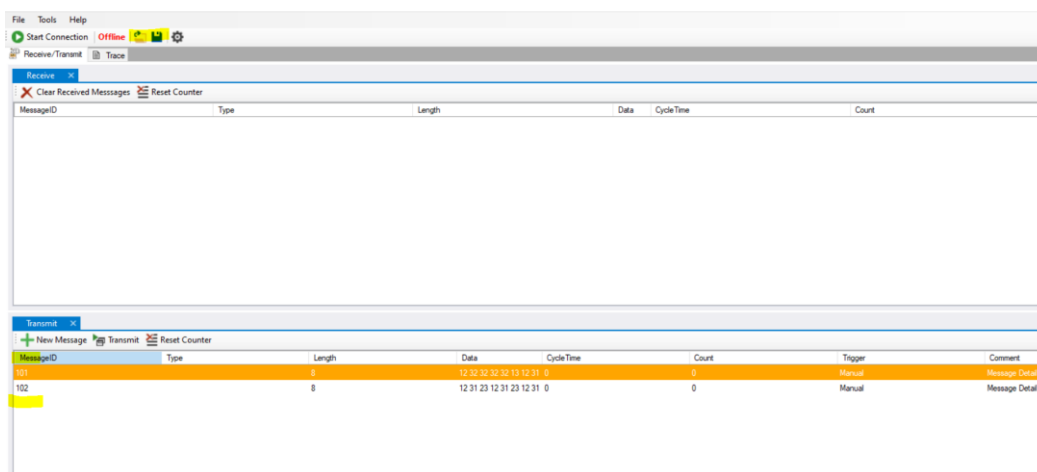

Figure 7. Save and Open Buttons

If the user tries to save messages for the first time, pop-up will be opened and new file can be created to the desired path with given name. If any file is saved before, recent files will be used and updates will be saved to there. Recent files can be seen in the "File", "Save As" feature is also included in there.

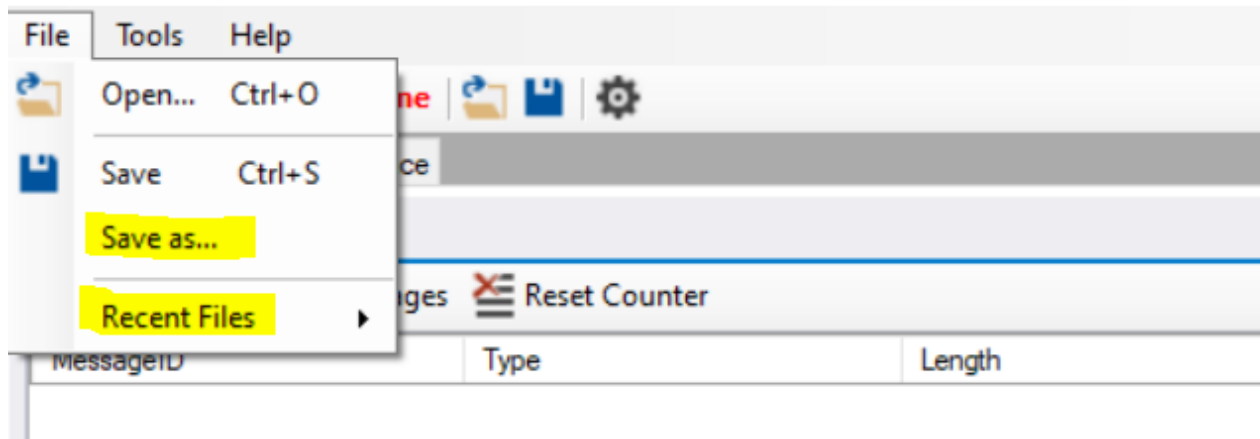Figure 8. Save as and Recent Files

With the open button, from the selected path, messages that were saved before will be populated to Transmit panel messages. The opened file will be shown in the Transmit panel tooltip with the gray color.

Figure 9. Opened File

5.3   Options

To access the settings of the DiagBox tool, please click on the settings icon.
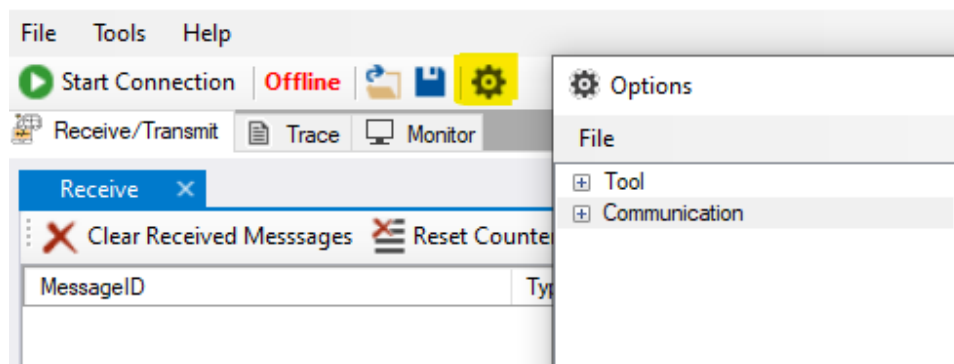

Figure 10. Options Page Content

5.3.1 General

Flush to UI: Controls how many log messages you see on the trace.

Flush to file: Sets how many lines of logs are written before saving to a file.

Rolling after: Decides when to start a new log file based on its size in megabytes.

File Path: Tells the tool where to save the log files on your computer.

5.3.2 Serial Port Configuration

In addition to CAN devices, DiagBox now supports connecting to devices through Serial Ports. Various parameters are available to establish a SerialPort connection. Here is a guide on how to set each one:

Port: Enter the appropriate port number through which the device connects in the textbox.

Baud Rate: Input the correct data transmission rate in the Baud Rate field.

Data Bits: Define the number of data bits for the communication in the Data Bits field.

Parity: Select a parity setting from the dropdown menu to ensure error checking during data transmission.

Stop Bits: Choose an option from the dropdown menu to dictate how the communication ends.

Read Timeout and Write Timeout: Determine the timeout durations for read and write operations by setting values in the Read Timeout and Write Timeout fields, respectively.

Remember to save the settings after configuring them to apply the new configurations.

5.3.3 CAN Hardware Bit Rate Configuration

Setting the bit rate is an essential process to facilitate the correct pace for data transmission over the CAN bus. Here is how you can set the bit rate and the various options available:

- Click on the settings icon to open the settings menu and select Can Network from the Communication heading.
- Select Device: Under the 'Options' section, you will find a dropdown menu to select your device. The available options are:
  - Intrepid
  - Kvaser (TBD)
  - Vector (TBD)

- Bit Rate Setting: Depending on your selected device, you will now select the desired bit rate from the following options:
    - 0
    - 2000
    - 33333
    - 50000
    - 62500
    - 83333
    - 100000
    - 125000
    - 250000
    - 500000
    - 800000
    - 1000000

Bit Rate Options: Explanation

- 0: Disables the bit rate, halting any communication over the CAN bus.
- 2000 to 1000000: Various bit rate settings to suit different data transmission requirements, ranging from very basic setups to environments necessitating very high-speed data transmission.

Notes:

- Ensure the selected bit rate is compatible with other devices on your network to maintain seamless communication.
- Higher bit rates allow faster data transmission but might increase the risk of errors; hence, a balance is necessary.

For further guidance, you may refer to the referenced documentations or contact support.

- CAN network selection is used to direct communication between different networks in a vehicle. The purpose of the selection is to regulate data transmission that needs to take place over a specific network, which facilitates targeted diagnostics and system checks. You can choose your preferred CAN network based on the unique needs and configurations of setup.

5.4   Help

Help of the tool includes "About" for now, in the about, version of the current release, integrated products are defined.
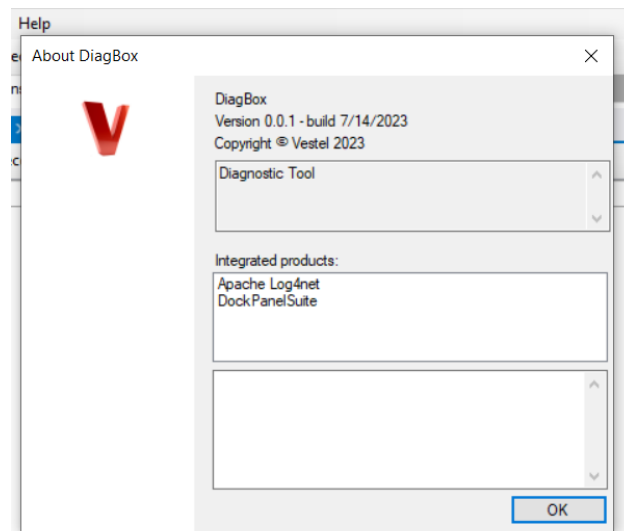

Figure 11. About from the Help

5.5   Transmitted and Received Rates

Top Right Corner: It shows the current success rate of sent and received messages. The percentage indicates the overall success rate, while the two numbers indicate the number of successful sends (up arrow) and receives (down arrow).

Title: The title bar shows a simplified version of the message success rate. The first number is the total number of messages sent, then the number of successful sends (up arrow) and the last number is the number of successful receptions (down arrow).
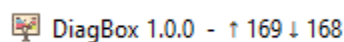
5.6   Importing CAN Messages Using a CSV File

Users can import CAN messages seamlessly using a CSV file. This facilitates a smoother workflow by allowing batch imports of CAN messages, saving time and reducing the chance of error compared to manual entry.

- Download the Predefined Template: To assist you in this process, we provide a predefined template that can be easily downloaded via a dedicated button available in the application.

	A	B	C	D	E	F	G	H	I	J
1	MessageID	Data	Cycle Time	Cycle Count	Delay Time	Count	Trigger	Comment	Multi Message	
2	13	00 55 00 11 00 55 44 33	55	1	0	1	1	1	FALSE	
3	-	Multi Messages	x	x	x	x	x	Text		
4	13	00 55 00 11 00 55 44 33	55	1	0	1	1	1	TRUE	
5										

- Import the CAN Messages: Once you have prepared your CSV file according to the predefined template, proceed to the CAN messages section in the application. Here, you will find an option to import the CAN messages. Select your CSV file and initiate the import process.

MessageID	Type	Length	Data		CycleTime	Cycle Count	Count	Trigger	Comment
13	Single-Message	0	00 55 00 11 00 55 44 33	55	1	1	1	1	
-	Multi-Messages	0	Multi-Messages	0	0	0	Manual	Text	

Multi-Messages

➕ New Message   💾 Save   Comment : Text

	MessageID	Type	Length	Data		Cycle Time	Cycle Count	Delay Time	Count
	13	Single-Message	0	00 55 00 11 00 55 44 33	55	1	0	1	

- Importing UDS Service Messages: In addition to the standard fields, two new columns have been introduced for UDS services – "Service Name" and "Service Function". These columns should be populated with the appropriate data for each UDS service to be imported.

	A	B	C	D	E	F	G	H	I	J	K	L
1	MessageID	Data	Cycle Time	Cycle Count	Delay Time	Count	Trigger	Comment	Multi Message	Service Name	Service Function	
2	13	00 55 00 11 00 55 44 33	55	1	0	1	1		1 FALSE	ReadDataByIdentifier	0xD100 - Active Diagnostic Session	
3	-	Multi Messages	x	x	x	x	x	Text				
4	13	00 55 00 11 00 55 44 33	55	1	0	1	1		1 TRUE	ReadDataByIdentifier	0xD100 - Active Diagnostic Session	
5												

6   Receive/Transmit

Receive/Transmit tab can be used to transmit and receive CAN and UDS messages with general CAN message format.

6.1   Receive

Receive panel grid includes received response messages to the transmitted messages. If the response message includes "7F" which means that it is negative feedback, message row will be written with red color. If it is positive feedback, it will be displayed in green color.
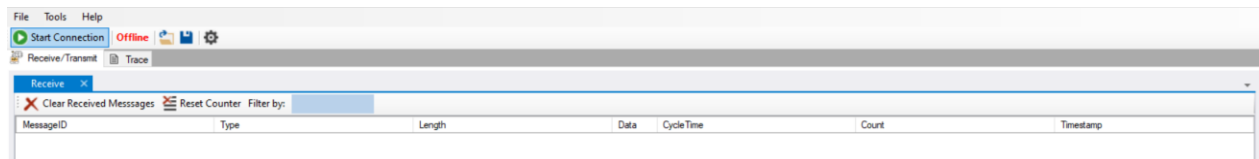

Figure 12. Toolstrip Buttons of the Receive Panel

Clear Received Message: Received messages included in the grid can be cleared.

Reset Counter: Total count of the received messages can be resetted for testing purposes or etc.

Filter By: Light blue back colored textbox can be used for filtering of the grid of the received messages.

Message ID: ID of the received message.

Type: Received message type.

Length: Length of the received message data.

Data: Received message data including several parameters.

Cycle Time: Indicates the intervals at which a message is received, measured in milliseconds.

Count: Total count of how many times the same message with same data is received.

Timestamp: Column to show when the message is received.

In addition to the above buttons, received messages can be unselected to remove back color of it and deleted one by one with the right click on the desired item by opening the context menu.
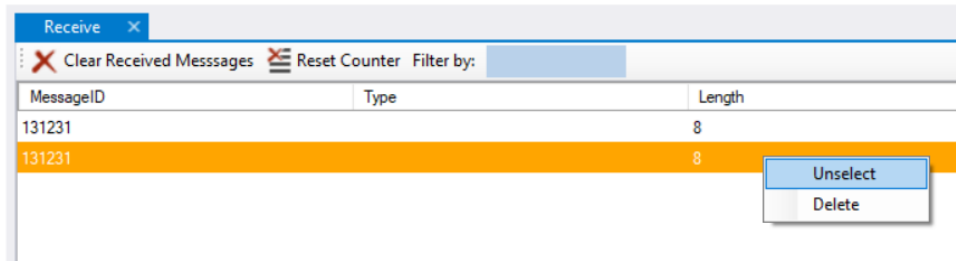
Figure 13. Context menu for the unselection and deletion

6.2 Transmit

Transmit panel grid includes transmitted messages to the ECU with the acknowledgment if it is really transmitted.
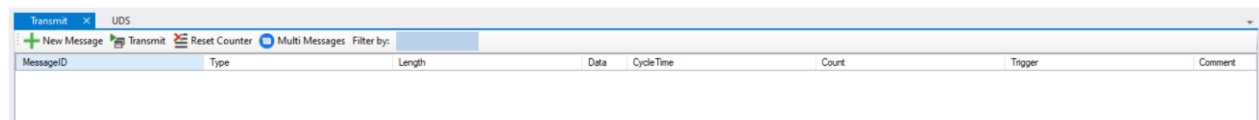

Figure 14. Toolstrip Buttons of the Transmit Panel

New Message: Users can add new messages to be able to transmit to the ECU.

Transmit: Messages can be transmitted to the ECU with this button, instead of this button "Space" key cut can be also used to transmit message.

Reset Counter: Total count of the transmitted messages can be resetted for testing purposes or etc.

Multi Messages: With this button, there is a possibility to send messages in desired order with the sort option on the opened window.
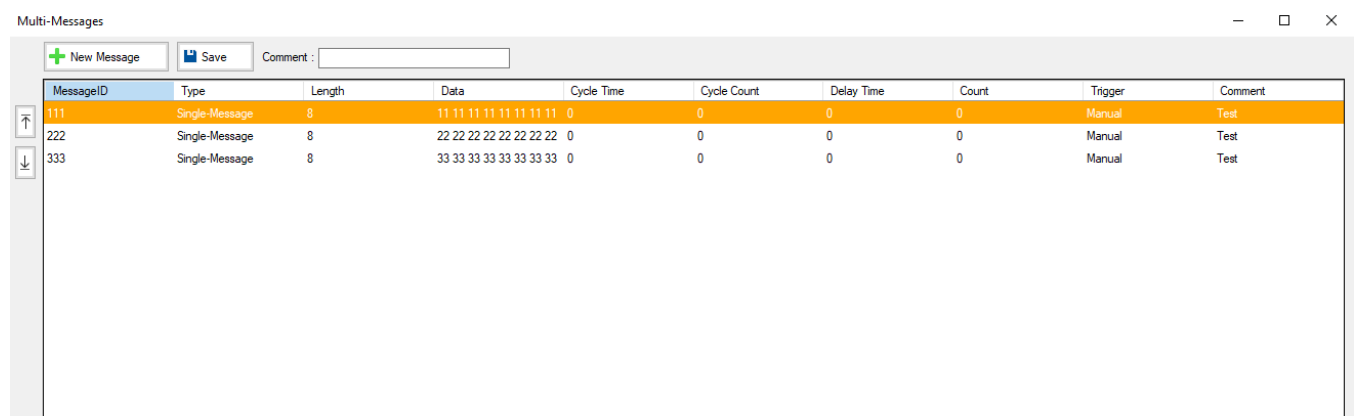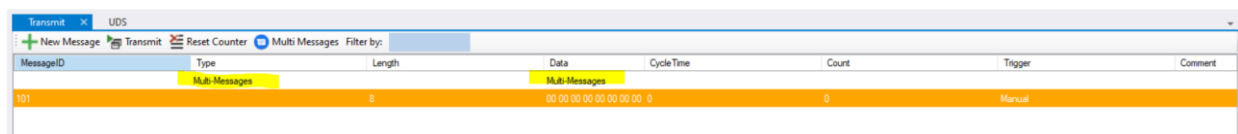

Figure 15. Multi messages pop-up window

Multi messages pop-up window is similar to new message addition, additionally it includes a sorting option for each message to transmit them in a desired order. Any message can be added and sorted, and before clicking the "Save" button, users have the opportunity to add comments to provide additional context or notes for each message.

After adding the desired messages and comments, click the "Save" button to finalize the setup. Then, "Multi Messages" can be shown in the datagridview of the transmit messages (Figure 12.). All of those messages would be transmitted with the "Transmit" or "Space" shortcut on the selected Multi Messages items row.

Figure 16. Multi messages in the datagridview

Filter By: Light-blue back colored textbox can be used for filtering of the datagridview of the transmitted messages.

MessageID: ID of the transmitted message.

Type: Transmitted message type.

Length: Length of the transmitted message data

Data: Transmitted message data including several parameters.

Cycle Time: Users can set a specific cycle time to transmission of messages at regular intervals.

Count: Total count of how many times message is transmitted.

Delay Time: During execution, each message will wait for the given delay time before being transmitted.

Trigger: If the message will be triggered manually or periodically.

Comment: Details of the transmitted message.

Messages included in the transmit grid can be copied and deleted with the right click option, whenever "Copy" is clicked row will be copied and pasted automatically at the end (Figure 11.).

Figure 17. Copy, Delete and Edit for transmit message

NOTE: The line containing the message in the delivery panel can also be updated by double-clicking on the updated line, the same pop-up window as Adding a New Message will open, updates can be saved when processed and the message will be updated in datagridview.
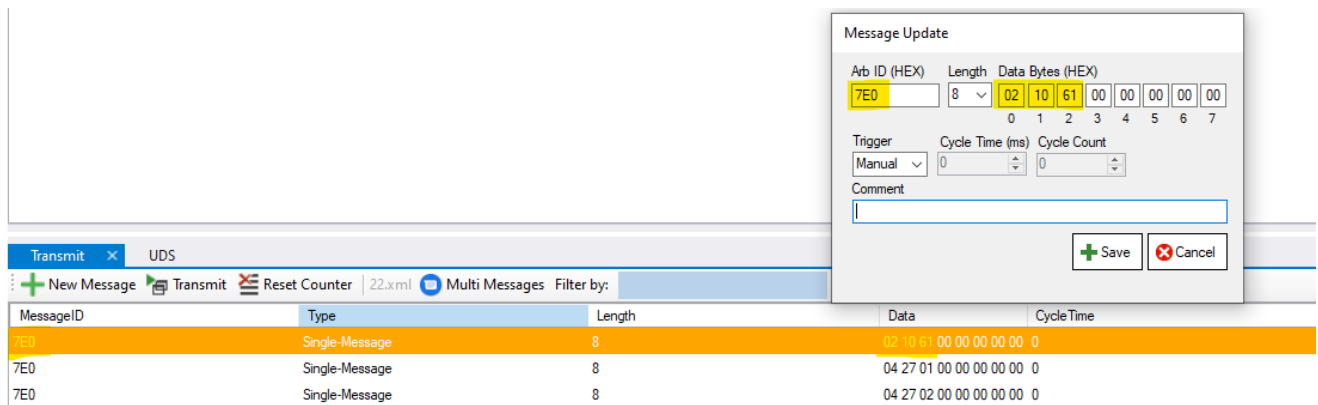

Figure 18. Editing Existing Message

NOTE: Messages in the transmission grid can be moved up or down in the grid with the up and down buttons on the left panel.
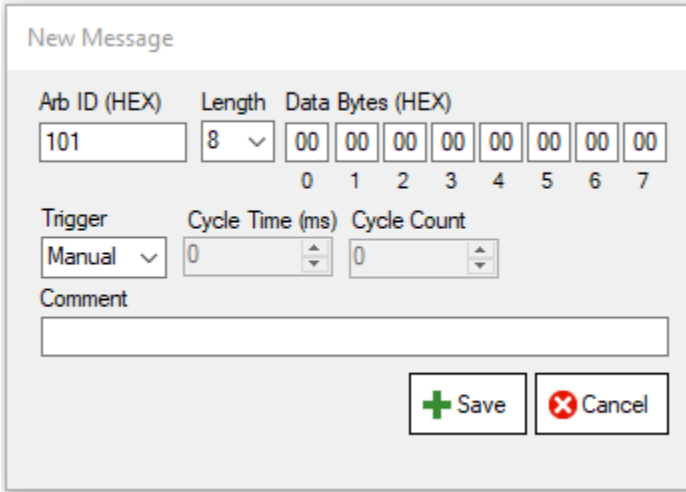

Figure 19. Message Movement Buttons

6.2.1   New Message Addition

Whenever "New Message" button is clicked on the Transmit panel. Below pop-up window will be opened.

Figure 20. New Message Addition Form

Notes:

- Message ID (Arb ID) and Data bytes of the messages should be entered with hex values, upper or lower case don't matter, every word will be written in uppercase.
- If user wants to add a new message in the multi-messages screen, an additional field named "delay time" will be added. In this field, users can specify, in milliseconds, how long they want to delay the transmission of the message. This allows users to create a pause between the transmission of individual messages within the multi-messages set.
- Periodic Trigger Type: When you select the "Periodic" trigger type from the combo in the popup, it enables the "Cycle Time" and "Cycle Count" fields, allowing you to set up periodic message transmissions based on the defined cycle time and cycle count parameters.

Columns included in the Transmit panel will be populated with the elements entered this pop-up. Each item included in this pop-up already explained in the Transmit section.

6.3 UDS Panel

The UDS section lists a collection of messages that will be transmitted using the UDS protocol, offering the following capabilities:

- New Message: Defines a new UDS message of a particular service type and appends it to the UDS message list.
- Transmit: Directly transmits the selected message to the connected device.
- Reset Counter: Restores the transmitted messages' counter to its initial state.
- Multi-Messages: Defines a set of messages to be transmitted to the ECU in a specific order.
- Filter By: Searches for a message based on any of its attributes.

6.3.1 New UDS message

Provides the user with the capability to define a new UDS message along with its core properties (such as ID, service name, trigger type, comment, etc.).

Upon selecting a service from the list, the corresponding user interface controls will become visible.

Note: The data of the message to be transmitted will be presented at the bottom of the dialog.

Note: For detailed guidance on utilizing the Periodic Trigger function and setting up the Delay Time for each message, refer to the respective sections in our user manual.

The services that are supported include:

6.3.1.1 DiagnosticSessionControl:

The DiagnosticSessionControl service is employed to manage various diagnostic sessions in a vehicle's electronic control unit (ECU), as outlined in the Unified Diagnostic Services (UDS) standard.

Users can select from a list of predefined sessions available in the Sub-Function ComboBox. The list includes but is not limited to:

- Default Session (0x01)
- Programming Session (0x02)
- Extended Diagnostic Session (0x03)
- EOL (End of Line) Extended Diagnostic Session (0x40)
- Application Programming Session (0x41)

If a distinct subfunction needs to be provided, the user can select the UserDefined check box and enter the value manually.

ISSUE 0.1
DATE 14-July-2023
PAGE 19/24

6.3.1.2 ECUReset:

The ECU reset service is used to trigger a reset or restart of one or multiple electronic control units (ECUs) within a vehicle's network. This is typically done when there's a need to restore the ECU to its default or initial state.

A predefined set of supported sub-functions will be displayed in the Sub-Function ComboBox.

- hard reset (0x01)
- soft reset (0x03)

If a distinct subfunction needs to be provided, the user can select the UserDefined check box and enter the value manually.

6.3.1.3 SecurityAccess:

The SecurityAccess service is a vital part of the Unified Diagnostic Services (UDS) and is instrumental in ensuring secure communication with the vehicle's Electronic Control Unit (ECU). This service essentially involves two primary functions: Request Seed and Send Key, which facilitate secure access to protected resources within the ECU. Below, we outline the primary components and guidelines for operating the SecurityAccess user control.

A predefined set of supported sub-functions will be displayed in the Sub-Function ComboBox.

- Request Seed (0x01): This option initiates a security access request, prompting the ECU to generate and send a security seed. This seed is used in the subsequent step where the key is generated and sent back to the ECU.
- Send Key (0x02): Following the request seed, use this option to send the generated key to the ECU, thus granting access to protected services and resources.

If a distinct subfunction needs to be provided, the user can select the UserDefined check box and enter the value manually.

6.3.1.4 TesterPresent:

The TesterPresent service is a vital part of the Unified Diagnostic Services (UDS) protocol, designed to maintain the active diagnostic session and keep it alive. This service is simple yet essential, especially in long diagnostics procedures where keeping the session alive is crucial to prevent timeouts and disruptions. Here we provide a detailed understanding of how to use the TesterPresent user control.

A predefined set of supported sub-functions will be displayed in the Sub-Function ComboBox.

- Zero Sub-function (0x00): This option sets the service to its standard operating mode, essentially notifying the server (usually an ECU) that the tester (diagnostic tool) is still present.

If a distinct subfunction needs to be provided, the user can select the UserDefined check box and enter the value manually.

6.3.1.5 ReadDataByIdentifier:

The ReadDataByIdentifier service, as a part of Unified Diagnostic Services (UDS), allows a diagnostic tester to retrieve specific data from a vehicle's ECU by providing the identifier of the requested data. The identifiers for data are selected from predefined sub-functions or manually entered as user-defined identifiers.

A predefined set of supported sub-functions will be displayed in the Sub-Function ComboBox.

- Active Diagnostic Session (0xD100): Retrieves data regarding the current diagnostic session.
- ECU Core Assembly Number (0xF111): Obtains the assembly number of the ECU core.
- ECU Main Calibration Data Number (0xF124): Retrieves the calibration data number for the main ECU.
- Diagnostic Specification Version (0xF163): Gets the version of the current diagnostic specification.
- Vehicle Manufacturer ECU Software Number (0xF188): Retrieves the software number assigned by the vehicle manufacturer to the ECU.
- Vehicle Identification Number (0xF190): Fetches the VIN of the vehicle.

A predefined set of supported sub-functions will be displayed in the Sub-Function ComboBox.

6.3.1.6 WriteDataByIdentifier:

The WriteDataByIdentifier service, part of the Unified Diagnostic Services (UDS), grants diagnostic testers the ability to write specific data into a vehicle's ECU by indicating the identifier associated with the data they intend to alter. Users can choose identifiers from predefined sub-functions or input them manually as user-defined identifiers.

A predefined set of supported sub-functions will be displayed in the Sub-Function ComboBox.

- Vehicle Identification Number (0xF190): This function is designed to write the VIN into the vehicle's system.

A predefined set of supported sub-functions will be displayed in the Sub-Function ComboBox.

6.3.1.7 InputOutputControlByIdentifier:

The InputOutputControlByIdentifier service is a critical aspect of the Unified Diagnostic Services (UDS). It is instrumental in facilitating the control of different I/O parameters in a vehicle's Electronic Control Unit (ECU). The service leverages predefined and user-defined identifiers to control the respective outputs.

A predefined set of supported sub-functions will be displayed in the Sub-Function ComboBox.

- returnControlToECU (0x00): Returns the control to the ECU.
- FreezeCurrentState (0x00): Maintains the current state without any change.
- ShortTermAdjustment (0x03): Allows for temporary adjustments to the settings.

Data Identifier (DID): Enter the Data Identifier (DID in text area, representing the data you wish to control.

A predefined set of supported sub-functions will be displayed in the Sub-Function ComboBox.

7 Trace

Trace tab includes basic logs of the tool such as connection details, received and transmitted messages.

Figure 21. Trace Tab Content

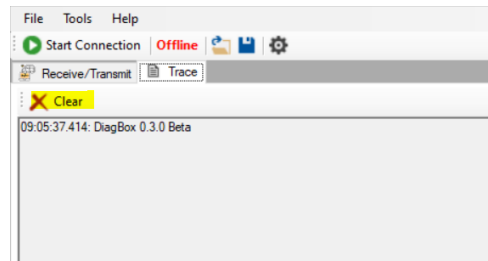Trace window can be cleared with the "Clear" button in its view.

Figure 22. Clear of the trace

Trace window content can be saved to the local as LogFile (.txt) format.
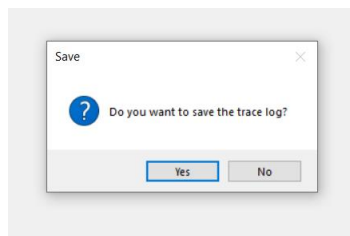

Figure 23. Saving of the trace window

7.1   Trace Dialog

The Trace Dialogue Box allows simultaneous viewing of all information from the main trace section on an additional window.
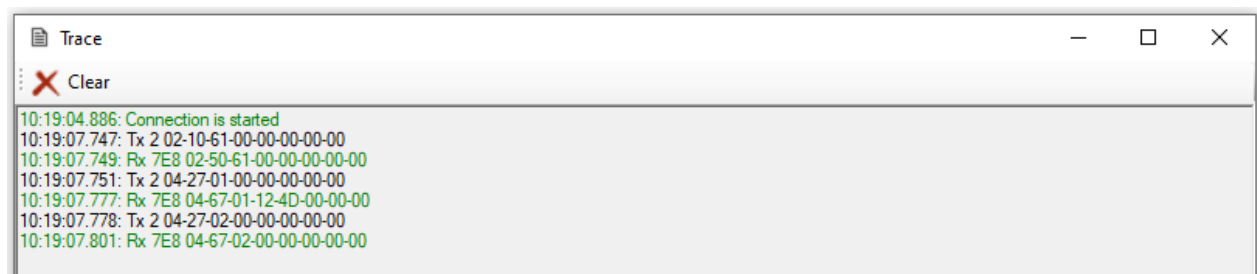
Trace dialog can be cleared with the "Clear" button in its view.

Figure 24. Trace dialog window