

R_Visualizer Manual

Christian Neuberger

November 25, 2016

Contents

| | | |
|----------|--|-----------|
| 1 | About R_Visualizer | 3 |
| 2 | Installing | 3 |
| 3 | Getting Started | 3 |
| 3.1 | Overview | 3 |
| 3.1.1 | Menu | 4 |
| 3.1.2 | Message Stream | 6 |
| 3.1.3 | System Overview | 7 |
| 3.1.4 | Send Messages | 7 |
| 3.1.5 | Message Configuration | 8 |
| 3.2 | Connecting the Host PC to the System | 13 |
| 3.2.1 | Connection to the System | 13 |
| 3.2.2 | Connection to the Host PC | 13 |
| 3.3 | Quickstart Guide | 13 |
| 4 | Observing a System | 15 |
| 5 | Inspecting the Aggregated Data | 16 |
| 6 | Controlling a System | 16 |

1 About R_Visualizer

R_Visualizer is a graphical tool used to observe and control the state of a compatible system. Said system is connected to the host pc running R_Visualizer via the CAN-Analyser USB interface. This interface will continually receive CAN messages from the system and propagate these messages to R_Visualizer and vice versa. Received messages will subsequently be visualized by R_Visualizer. The application itself is capable of sending messages to the system via the CAN-Analyser USB interface to control the system state or behavior.

2 Installing

There is no installation needed. Simply unzip the provided zip file to any location you desire that does not require administrative privileges and start the application via the *R_Visualizer.exe* file. Do not delete any file or folder from the unzipped folder.

3 Getting Started

This section will guide you through R_Visualizer and show you how to use the basic functionalities of the Application.

- For a quick Overview refer to subsection 3.1 Overview
- For a quick Connecting to the system guide refer to subsection 3.2 Connecting the Host PC to the System
- For a quick Quickstart guide refer to subsection 3.3 Quickstart Guide

3.1 Overview

The general user interface is designed with simplicity in mind. The main goal was to expose as much information as possible without distracting the user with cluttered configuration dialogs. Therefore, the interface is divided into two main parts: the message stream on the left-hand side and the overview/configuration dialogs on the right-hand side.

Most of the application is tailored to the current user. A user can either be a regular user or an *Admin*. The regular user is only exposed to basic functionality that is needed to monitor and control the observed system. On the other hand, the *Admin* user has advanced functionality like storing or editing configuration entries. The user roles can be switched with the respective symbol in the shortcut panel or from the *User* menu at the top of the application.

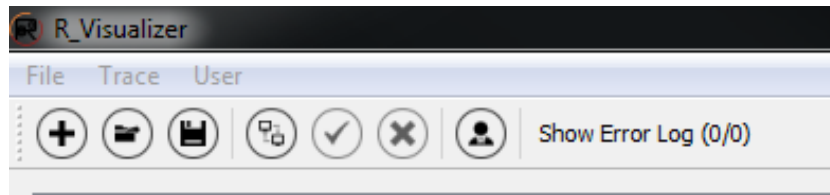


Figure 1:

3.1.1 Menu

The *Menu* is comprised of the items:

- File (see Figure 2 File Menu)
can be used to load a recent *Message Stream* or store the current *Message Stream*. Additionally, the *Error Log* is accessible from the *File* menu.¹
 - New - TESTING ONLY
 - Open - Open a previously saved *Message Stream*
 - Save - Save the current *Message Stream*
 - Show Error Log - Displays the current *Error Log*²
- Trace (see Figure 3 Trace Menu)
can be used to establish a connection to the CAN-Analyser USB interface and to disconnect R_Visualizer from the CAN-Anylser USB interface, as well as starting and stopping the recording of messages.³
 - (Dis)Connect - Connect or disconnect to the CAN-Analyser USB interface⁴
 - Start - Start recording incoming messages to the *Message Stream*(Connect first!)
 - Stop - Stop recording incoming messages to the *Message Stream*
- User (see Figure 4 User Menu)
can be used to switch between *Admin* and *User* mode.
 - Admin/User - Switch to the currently displayed user role

All items from all sub-menus can also be accessed through the respective shortcut buttons in the menu panel.

¹Currently the *New* option is used for testing. It does not flush the *Message Stream* to create a new stream but rather adds example messages to the stream...

²ONLY CAN errors are displayed here.

³Keep in mind that stopping the message recording will stop the *Message Stream* from storing incoming messages. Messages that are not stored by the *Message Stream* are lost and cannot be retrieved by any means.

⁴The functionality of the menu entry depends on the current connection state.

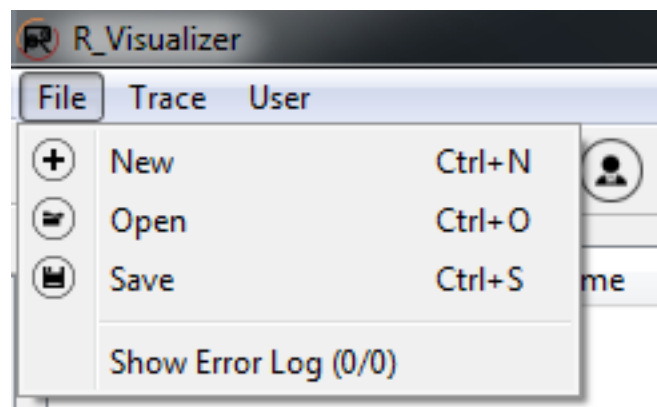


Figure 2:

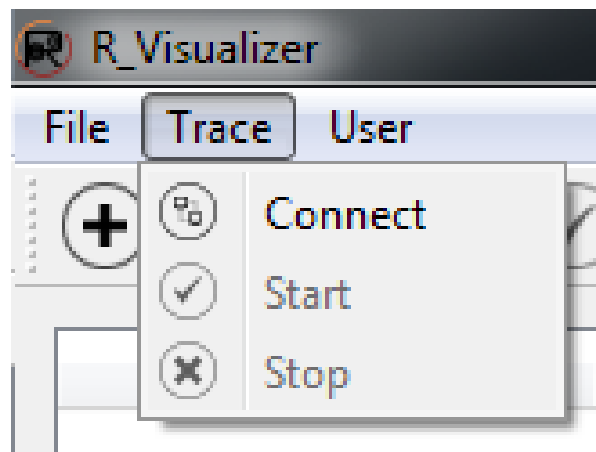


Figure 3:

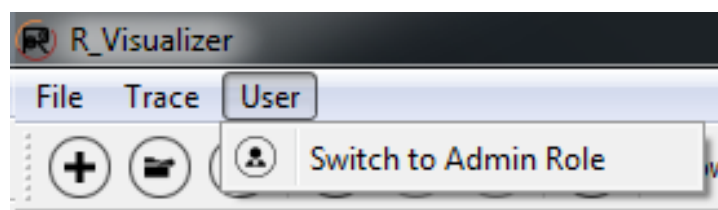


Figure 4:

3.1.2 Message Stream

To achieve maximum exposure of the system behavior, the *Message Stream* is visible in all configuration dialogs and during the system overview dialog. The incoming messages are continuously updated and displayed.

The *Message Stream* displays messages in the order they are picked up by the CAN-Anylser USB interface: in a chronological order. The messages visualization is customizable via the configuration dialogs in the *Message Config* tab (see Figure 5 Message Stream). Per default messages are displayed with their respective timestamp, id and code plus data.

The user can use scrolling to view any messages that were picked up by the *Messages Stream*, there is no limitation on the history.

| Timestamp | Name | Message |
|---------------------------|--------|--|
| 16.11.2016 - 20:03:21.404 | Master | Code: SYS_TICK_code Tick nr. 6 |
| 16.11.2016 - 20:03:21.557 | Pump | Code: PUMP_STAT_OFF PUMP is OFF |
| 16.11.2016 - 20:03:21.719 | Pump | Code: PUMP_STAT_ON_code PUMP is ON |
| 16.11.2016 - 20:03:21.881 | Pump | Code: PUMP_STAT_OFF PUMP is OFF |
| 16.11.2016 - 20:03:22.042 | Pump | Code: PUMP_STAT_ON_code PUMP is ON |
| 16.11.2016 - 20:03:22.064 | Temp | Code: TEMP_code TEMP: 0°C |
| 16.11.2016 - 20:03:22.066 | Temp | Code: TEMP_code TEMP: 1°C |
| 16.11.2016 - 20:03:22.068 | Temp | Code: TEMP_code TEMP: 2°C |
| 16.11.2016 - 20:03:22.069 | Temp | Code: TEMP_code TEMP: 3°C |
| 16.11.2016 - 20:03:22.071 | Temp | Code: TEMP_code TEMP: 4°C |
| 16.11.2016 - 20:03:22.073 | Temp | Code: TEMP_code TEMP: 5°C |
| 16.11.2016 - 20:03:22.074 | Temp | Code: TEMP_code TEMP: 6°C |
| 16.11.2016 - 20:03:22.076 | Temp | Code: TEMP_code TEMP: 7°C |
| 16.11.2016 - 20:03:27.380 | Master | Code: SYS_TICK_code Tick nr. 7 |
| 16.11.2016 - 20:03:27.534 | Pump | Code: PUMP_STAT_ON_code PUMP is ON |
| 16.11.2016 - 20:03:27.695 | Pump | Code: PUMP_STAT_OFF PUMP is OFF |
| 16.11.2016 - 20:03:27.857 | Pump | Code: PUMP_STAT_ON_code PUMP is ON |
| 16.11.2016 - 20:03:28.019 | Pump | Code: PUMP_STAT_OFF PUMP is OFF |
| 16.11.2016 - 20:03:33.358 | Master | Code: SYS_TICK_code Tick nr. 8 |
| 16.11.2016 - 20:03:33.512 | Pump | Code: PUMP_STAT_OFF PUMP is OFF |
| 16.11.2016 - 20:03:39.334 | Master | Code: SYS_TICK_code Tick nr. 9 |
| 16.11.2016 - 20:03:39.487 | Pump | Code: PUMP_STAT_OFF PUMP is OFF |
| 16.11.2016 - 20:03:45.311 | Master | Code: SYS_TICK_code Tick nr. 10 |
| 16.11.2016 - 20:03:51.287 | Master | Code: SYS_TICK_code Tick nr. 11 |
| 16.11.2016 - 20:03:57.264 | Master | Code: SYS_TICK_code Tick nr. 12 |
| 16.11.2016 - 20:03:57.417 | Pump | Code: PUMP_STAT_ON_code PUMP is ON |
| 16.11.2016 - 20:03:57.579 | Pump | Code: PUMP_STAT_ON_code PUMP is ON |
| 16.11.2016 - 20:04:03.241 | Master | Code: SYNC_code Message: 0x00 0x00 0x70 0x3e 0x00 0x00 0x00 |
| 16.11.2016 - 20:04:03.242 | Master | Code: SYNC_code Message: 0x00 0x00 0x70 0x3e 0x00 0x00 0x00 |
| 16.11.2016 - 20:04:03.243 | Master | Code: SYS_TICK_code Tick nr. 1 |
| 16.11.2016 - 20:04:03.395 | Pump | Code: PUMP_STAT_OFF PUMP is OFF |
| 16.11.2016 - 20:04:09.217 | Master | Code: SYS_TICK_code Tick nr. 2 |

Figure 5: The *Message Stream* displaying incoming messages

3.1.3 System Overview

CURRENTLY NOT IN A PRODUCTION STATE. USE AT OWN RISK.

The System Overview tab can be used to visually recreate the connected system and observe the system's current state.

3.1.4 Send Messages

In order to communicate and control the system, the *Send Messages* tab provides the means to send single messages or whole packages of messages to the system (see Figure 6 Send Messages Dialog)

Sending single messages can be achieved with the Send Single Message sub-dialog within the *Send Messages* tab. Type in an ID that shall be exposed to the system as the sender of the message, the desired code to send and additionally the data you want to send with the message.

The ID/Name field can be filled with a numerical value in decimal or hexadecimal format⁵. If the ID to name mapping has already been loaded or created in the *Message Config* tab, it is also possible to type in the specified name or choose one from the auto-completion list.

The Code field can be filled with a numerical value in decimal or hexadecimal format⁶. If the Code to name mapping has already been loaded or created in the *Message Config* tab, it is also possible to type in the specified name or choose one from the auto-completion list.

The Data field can be filled with numerical values according to your selection in the the combo box right next to the data field. The application will translate the entered data to an internal byte-wise representation automatically. The specific data format can be one of the following:

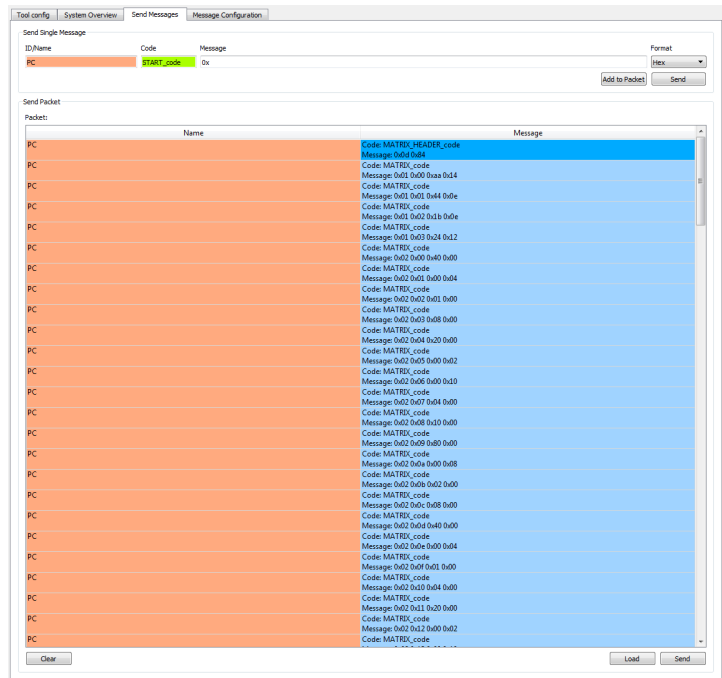


Figure 6: The *Send Messages* dialog that can be used to send single messages or whole packages of messages in the specified order

⁵for hexadecimal prepend 0x to the hexadecimal number

⁶for hexadecimal prepend 0x to the hexadecimal number

- Hex
hex values in portions of bytes
- Dec Data
decimal values in portions of bytes⁷
- Dec Value
one decimal value which will automatically be translated to bytes by the application⁸
- Bin
binary values (byte-wise)

When switching the selection of the combo box the value is automatically translated from the last selection to the new selection⁹

Sending packages of messages can be achieved with the *Send Packet* sub-dialog in the *Send Messages* tab. In order to load a matrix of messages, simply click load and select the desired matrix to load in either .csv or .json format¹⁰. To Send the whole package of messages in a chronological order from top to bottom simply click on Send. To clear the current Packet click on the Clear button. It is not necessary to clear the view before loading another Packet. The display style of the loaded message packet is similar to the *Message Stream's* display style, because both entities rely on the same *ID* and *Message Type* configuration.

3.1.5 Message Configuration

The *Message Configuration* tab directly interferes with the *Message Stream*. It consists of three separate sub-dialogs: Filter, IDs and Message Types (see Figure 7 Message Configuration Dialog)

⁷Keep in mind that one byte must not exceed the number 255

⁸Keep in mind not to exceed the 56bit limitation, which is 72057594037927935

⁹Keep in mind that translating from a value-based notation to a byte-wise notation or vice versa might mess up the byte order.

¹⁰The .json format is kept to comply with the rest of the application's data. Only load .json files that were previously saved with the Send Packet Store function.

Tool config
System Overview
Send Messages
Message Configuration

Filter:

ID/Name

Master
PC

Disable
Add
Remove

Message Code

SYS_TICK_code
MATRIX_code
OK_code

Enable
Add
Remove

Timestamp

From 16.11.2016 12:06:00.000
Disable

To 16.11.2016 12:07:00.000
Disable

IDs

| ID | Name | Color |
|--------|--------|---------|
| 0x00ff | PC | #ffaa7f |
| 0x00f0 | Master | #abd0ff |
| 0x0100 | Pump | #d7b4ff |
| 0x0110 | Temp | #ff9c9c |
| 0x0120 | Valve0 | #7a5d7f |
| 0x0121 | Valve1 | #7f5b71 |

Load

Message Types

| Code | Name | Message Format | Color |
|------|----------------|------------------|---------|
| 0x0b | MATRIX_code | | #a0d3ff |
| 0x10 | SYS_TICK_code | Tick nr. #Data0# | #00ff95 |
| 0x20 | PROGRAM_code | | #ffff7f |
| 0x40 | SYNC_code | | #86c5ff |
| 0x4f | OK_code | | #00ce00 |
| 0x50 | PUMP_SENT_code | | #d778ff |
| 0x51 | PUMP_ON_code | | #bfa6ff |
| 0x52 | PUMP_OFF_code | | #ffa2ff |

Load

Figure 7: The *Message Configuration* dialog that can be used to enhance the visual representation of messages in the *Message Stream* and to filter for specific messages in the *Message Stream*

The Filter section can be used to filter the *Message Stream* for certain properties (see Figure 8 Message Configuration Filter). The filters are chained. To be more specific, the filters interact with each other to display only messages that suffice all enabled filter criteria at once. Use multiple filters to get a fine-grained filtering of the *Message Stream*. Since the *Messages Stream* is updated live, it is advised to first setup all of the desired filter criteria and then enable all filters, starting from the most general one.

The ID/Name subsection can be used to filter for certain IDs or -provided the ID to name mapping has been setup appropriately- names. To create an ID filter, simply click the *Add* button and type in the desired ID in either decimal or hexadecimal representation¹¹. If and only if an ID to name mapping has been set up, the respective names can be entered directly or chosen from the auto-completion dialog. To remove a filter, select the according entry from the list and click on the *Remove* button. Per default the ID/Name filter is disabled. To enable the filter click on the *Enable* button.¹².

The Code subsection can be used to filter for certain code or -provided the code to name mapping has been setup appropriately- names. To create a code filter, simply click the *Add* button and type in the desired Code in either decimal or hexadecimal representation¹³. If and only if a code to name mapping has been set up, the respective names can be entered directly or chosen from the auto-completion dialog. To remove a filter, select the according entry from the list and click on the *Remove* button. Per default the Code/Name filter is disabled. To enable the filter click on the *Enable* button.¹⁴.

¹¹for hexadecimal prepend 0x to the hexadecimal number

¹²It is advised to first add all desired filter criteria before enabling the filter, because the *Message Stream* will be updated with each new entry which leads to increased workload

¹³for hexadecimal prepend 0x to the hexadecimal number

¹⁴It is advised to first add all desired filter criteria before enabling the filter, because the *Message Stream* will be updated with each new entry which leads to increased workload

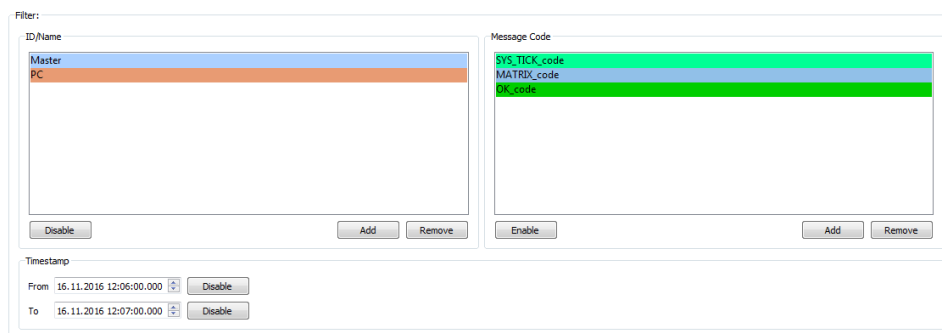
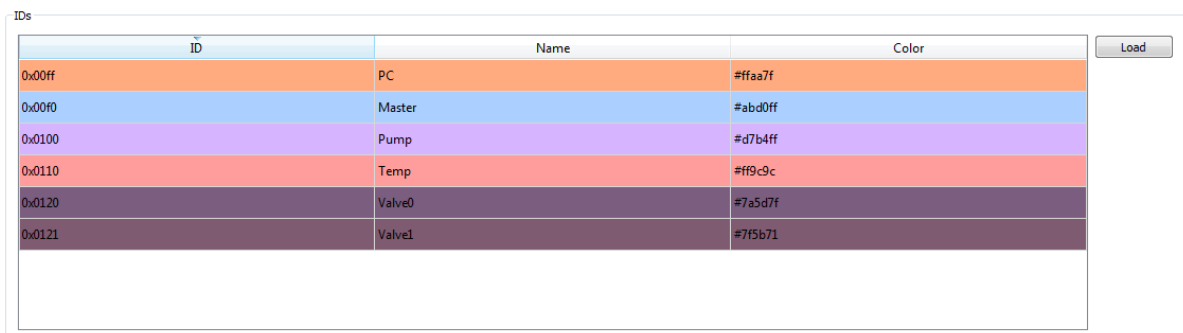


Figure 8: The *Message Configuration* Filters that can be used to filter the *Message Stream* for specific messages

The Timestamp subsection can be used to filter for a certain period of time. Both, the *from* as well as the *to*, can be enabled or disabled independently from another. Setting the *from* parameter will automatically adjust the *to* parameter and vice versa to form a meaningful time span -even if disabled. Again, it is advised to adjust the parameters with the filters being disabled in order to gain performance.

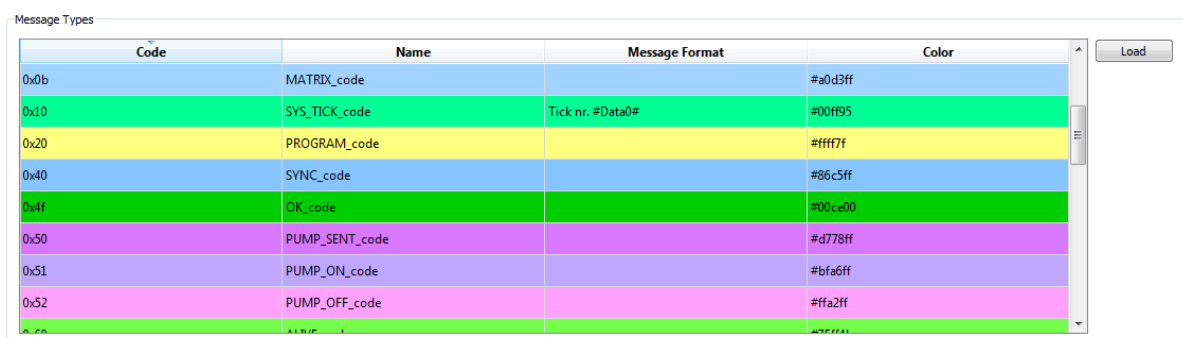
The IDs section can be used to alter the visualization of the *Message Stream's Name* column (see Figure 9 Message Configuration IDs). Each ID can be assigned to an alias and a color. If a message in the *Message Stream* matches against a configured ID, its name will be displayed instead of the numerical representation and the respective field will be colored in the configured color. This is especially useful to rapidly distinguish different message senders in the *Message Stream*. In addition, all fields in all other menus that require IDs are updated to auto-complete if a name to a configured ID is entered. Most of the fields are then also colored respectively.



| ID | Name | Color |
|--------|--------|---------|
| 0x00ff | PC | #ffa7f |
| 0x00f0 | Master | #abd0ff |
| 0x0100 | Pump | #d7b4ff |
| 0x0110 | Temp | #ff9c9c |
| 0x0120 | Valve0 | #7a5d7f |
| 0x0121 | Valve1 | #7f5b71 |

Figure 9: The *Message Configuration* IDs that can be used to set alias name and coloring for the *Message Stream's Name* column

The Message Types section can be used to alter the visualization of the *Message Stream's* *Message* column (see Figure 10 Message Configuration Message Types). Each Code can be assigned to an alias and a color. In addition, a message format can be set with the message formatter dialog that will be used to parse the data portion and code of an incoming message to a visually appealing format. If a message in the *Message Stream* matches against a configured Code, its name will be displayed instead of the numerical representation and the respective field will be colored in the configured color. In addition, the configured message format will be used to enhance the representation of the message. This is especially useful to rapidly distinguish different messages by code in the *Message Stream*. In addition, all fields in all other menus that require Codes are updated to auto-complete if a name to a configured Code is entered. Most of the fields are then also colored respectively.



| Code | Name | Message Format | Color |
|------|----------------|------------------|---------|
| 0x0b | MATRIX_code | | #a0d3ff |
| 0x10 | SYS_TICK_code | Tick nr. #Data0# | #00ff95 |
| 0x20 | PROGRAM_code | | #ffff7f |
| 0x40 | SYNC_code | | #86c5ff |
| 0x4f | OK_code | | #00ce00 |
| 0x50 | PUMP_SENT_code | | #d778ff |
| 0x51 | PUMP_ON_code | | #bfa6ff |
| 0x52 | PUMP_OFF_code | | #ffa2ff |

Figure 10: The *Message Configuration* Message Types that can be used to set alias name and coloring for the *Message Stream's* *Message* column and format the messages' data

3.2 Connecting the Host PC to the System

The CAN-Analyser USB interface has exactly two outputs: a standard USB connector and a four pin CAN connector. The CAN connector shall be connected to the observed System's CAN bus, whereas the USB connector shall be connected to the host PC.

3.2.1 Connection to the System

Make sure that the system is not powered¹⁵. Connect the CAN-Analyser USB interface to one of the free connectors on the system's CAN bus using a supplied CAN compatible cable.

3.2.2 Connection to the Host PC

Simply plug the CAN-Analyser USB interface into one of your host PC's USB ports¹⁶. Windows should be able to deduct the needed driver and install the CAN-Analyser USB interface accordingly.

The drivers needed are the HID driver and a USB driver. When the host PC is equipped with either a keyboard, a mouse or any other kind of standard input device, the HID driver should be pre-installed. When the host PC is equipped with USB ports, the USB drivers should be pre-installed.

It is advised to use a USB extension cable. Since the CAN-Analyser USB interface is moderately sized and only held by the USB connector, the USB connector can easily be destroyed by force in either direction when applied to the plugged in CAN-Analyser USB interface. Moreover, the whole CAN-Analyser USB interface might take damage when exposed to force.

Additionally it is advised to plug the CAN-Analyser USB interface into a self-powering USB hub in order not to damage the host PC's USB ports in case of unexpected system failure. This scenario is quite unrealistic, since the CAN-Analyser USB interface has been tested thoroughly, but worth noting.

3.3 Quickstart Guide

In order to test the application's functionality follow these steps to set up R_Visualizer quickly with some example data. This guide is intended to display the capabilities and mechanisms. Since this guide will only focus on loading, displaying and configuring already existent data, the steps needed to connect to a real system are left out. For further details on interaction with a real system please refer to section 4 Observing a System.

Quickstart:

¹⁵Sidenote: It is possible to plug in the CAN-Analyser USB interface while the system is powered on (or even running), but it is not advised. System failure or damage could be the result.

¹⁶Either USB 2 or USB 3; both are supported.

- Navigate to the application directory using either the explorer or a command line
- Start the Application by starting *R_Visualizer.exe*
→ R_Visualizer starts and displays its main screen
- Open the already prepared test *Message Stream* by either using the *File* menu, the respective shortcut button in the menu panel or by using the shortcut *ctrl+o*
 - In the opening dialog navigate to the *TestData* directory and open *TestRun.json*
 - R_Visualizer will now load an example *Message Stream* that was recorded by R_Visualizer earlier
- To customize the appearance of the *Message Stream*, navigate to the *Message Configuration* to on the right-hand side of the main window
 - To enhance the visual appearance of the *Message Stream's* Name column, load an *IDs* File by clicking on *Load* in the *IDs* subsection
 - * In the *TestData* directory you will be able to locate an *IDs.json* file.
 - * Open said file and R_Visualizer will automatically populate the IDs subsection with the data contained in the file
 - * If successful the *Message Stream* will update immediately to display the enhanced Name column
 - To enhance the visual appearance of the *Message Stream's* Message column, load an *Message Types* File by clicking on *Load* in the *Message Types* subsection
 - * In the *TestData* directory you will be able to locate an *MessageTypes.json* file.
 - * Open said file and R_Visualizer will automatically populate the Message Types subsection with the data contained in the file
 - * If successful the *Message Stream* will update immediately to display the enhanced Message column
- Feel free to browse through the *Message Stream*
- To filter the *Message Stream*, navigate to the *Message Configuration* tab on the right-hand side of the main screen
 - To filter for certain IDs/Names use the *ID/Name* subsection of the Filter section
 - * To add an ID/Name filter criteria click on the Add button
 - * You will now be prompted to enter the ID/Name to filter
 - * Since you have already loaded an *IDs* file, you can now enter the Name to filter for or the numerical representation in either decimal or hexadecimal

- * Finally enable filtering for your ID/Name filter criteria by clicking on Enable
- To filter for certain Message Codes use the *Message Code* subsection of the Filter section
 - * To add a Message Code filter criteria click on the Add button
 - * You will now be prompted to enter the Code/Name to filter
 - * Since you have already loaded an *Message Types* file, you can now enter the Name to filter for or the numerical representation in either decimal or hexadecimal
 - * Finally enable filtering for your Message Code filter criteria by clicking on Enable
- To filter for a certain timespan use the *Timestamp* subsection of the Filter section
 - * To only display messages that were recorded after a certain point in time edit the From field of the Timestamp subsection
 - * To only display messages that were recorded before a certain point in time edit the To field of the Timestamp subsection
 - * Enable the respective Timestamp filter by clicking on the respective Enable button
 - * Both Timestamp filters can be combined by enabling both Timestamp filters in order to filter for a certain timespan
- To close R_Visualizer simply click on the X in the top right corner, use the shortcut *alt+F4* or any other preferred method of closing windows applications.

4 Observing a System

In order to observe a system, first connect the system via the CAN-Analyser USB interface to the host PC as described in subsection 3.2 Connecting the Host PC to the System. After connecting the interface to the host PC establish a connection between R_Visualizer and the interface by clicking on the connect symbol in the menu panel or selecting the respective entry from the *Trace* menu. If the connection is established successfully the *Connect* symbol changes to the *Disconnect* symbol. If the connection cannot be established you are prompted to connect the CAN-Anylser USB interface to the host PC correctly. Given a connection has been established start recording incoming messages by clicking on the Start symbol in the menu panel or selecting Start from the trace menu. If the recording has started successfully the Start symbol will be greyed out and the Stop symbol will be activated.

As soon as the recording is started the *Message Stream* is updated with each new incoming message. All incoming messages are stored in the application and can be viewed using the *Message Stream*.

The Message Stream visualization can be configured using the Message Configuration tab as described in subsection 3.1.5 Message Configuration.

5 Inspecting the Aggregated Data

A currently recorded *Message Stream* can be saved to disk by clicking on the Save shortcut in the menu panel or selecting Save from the File menu. At any time a *Message Stream* can be loaded by clicking on the Open shortcut in the menu panel or selecting Open from the File menu. It is advised to stop and save a currently active recording before opening a new *Message Stream*, because the current stream will be flushed and the data is lost if not saved. New incoming messages will be appended to the loaded *Message Stream*.

All configurations made in the *Message Configuration* tab are applied to the current *Message Stream*. Use filters to narrow the displayed data to a desired result.

6 Controlling a System

If and only if a connection to the system has been established correctly (see section 4 Observing a System), it is possible to directly interact with the system by sending single messages or whole packages of messages to the system.

To send single messages use the *Send Single Message* sub-dialog in the *Send Messages* tab. The ID/Name that will be used to indicate the sender of the message can be entered into the ID/Name field. It is possible for the R_Visualizer application to be seen as any node in the system from the system's point of view. This is especially useful when testing new functionalities or provoking abnormal system behavior¹⁷. Usually the ID/Name to be entered here is the one of the host PC, in order to appear as the PC node to the system.

To send a control matrix, load the corresponding matrix into the *Send Packet* sub-dialog by clicking Load underneath the Packet visualization. A loaded matrix will be displayed in the Packet visualization upon loading. To send the loaded matrix click on *Send*, which will then instruct R_Visualizer to send the matrix with the given messages to the system chronologically starting from the first entry.

It is possible to construct a new package by entering the desired message into the Send Single Message sub-dialog and clicking on *Add to Packet*. This will append a new message with the given fields to the current message package. In order to clear a previously loaded message package (e.g. to construct a new one, click on *Clear*).

¹⁷USE WITH CAUTION!

List of Figures

| | | |
|----|---|----|
| 1 | Menu | 4 |
| 2 | File Menu | 5 |
| 3 | Trace Menu | 5 |
| 4 | User Menu | 5 |
| 5 | Message Stream | 6 |
| 6 | Send Messages Dialog | 7 |
| 7 | Message Configuration Dialog | 9 |
| 8 | Message Configuration Filter | 10 |
| 9 | Message Configuration IDs | 11 |
| 10 | Message Configuration Message Types | 12 |