R_Visualizer Manual

Christian Neuberger

November 23, 2016

Contents

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 priviledges and start the application via the *R_Visualizer.exe* file. Do not delete any file or folder from the unzipped folder.

3 Getting Started

3.1 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.

3.1.1 Connection to the System

Make sure that the system is not powered[1]. Connect the CAN-Analyser USB interface to one of the free connectors on the system's CAN bus using a supplied CAN compatible cable.

3.1.2 Connection to the Host PC

Simply plug the CAN-Analyser USB interface into one of your host PC's USB ports. 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-Anylser USB interface is moderately sized and only held by the USB connector, the USB connector can easily

[1]Sidenote: 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.

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 forced.

Additionally it is advised to plug the CAN-Analyser USB interface into a self-powering USB hub in order not to damager 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 worthing noting.

3.2 Overview

The general user interface is designed with simplicity in mind. Nevertheless, 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 side, 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.

3.2.1 Menu

The *Menu* is comprised of the items: *File*, *Trace* and *User*.

The *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.

The *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.[2]

3.2.2 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. Per default messages are displayed with their respective timestamp, id and code plus data.

[2]Keep 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.

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

3.2.3 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.

3.2.4 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.

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 hexademical format[3]. 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 hexademical format[4]. 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. If Hex is specified you can enter hex values in portions of bytes, else if Dec Data is specified you can enter decimal values in portions of bytes[5], else if Dec Value is specified you can enter a decimal value which will automatically be translated to bytes by the application[6], else if Bin is selected you can enter the data byte-wise in binary notation. When switching the selection of the combo box the value is automatically translated from the last selection to the new selection[7]

[3]for hexadecimal prepend 0x to the hexadecimal number
[4]for hexadecimal prepend 0x to the hexadecimal number
[5]Keep in mind that one byte must not exceed the number 255
[6]Keep in mind not to exceed the 56bit limitation, which is 72057594037927935
[7]Keep in mind that translating from a value-based notation to a byte-wise notation or vice versa might mess up the byte order.