Infotainment Business Unit
The NMEA Over CAN Application

1 Introduction

Scope of this document is to provide an overview of the NoC (*NMEA Over Can*) application, which is given in the first chapter of this document.

Then the specification of the protocol used to map the NMEA information inside the standard CAN frames is provided.

In the end, the description of the NoC Viewer GUI (a tool developed to test and validate the NoC application), is given.

2 Contents

2.1 Index

2.2 List of Tables

2.3 List of Figures

3 Document Management

3.1 Revision History

Rev	Date	Author	Notes
1.0	22/2/2016	A. Occhipinti	First Version
1.1	04/04/2016	A. Occhipinti	Figures aligned to NoC Viewer 0.6 Introduced “CAN USB group box” section Introduced “NoC Viewer Terminal” section
1.2	21/06/2016	S. Chambrillon	Add note in 5.1 for FreeRTOS API usage.

Table 1: Revision history

3.2 Acronyms

Keyword	Definition
API	Application Programming Interface
CRC	Cyclic Redundancy Check
DLC	Data Length Code
GUI	Graphical User Interface
LLD	Low Level Driver
NMEA	National Marine Electronics Association
NoC	Nmea Over Can

Table 2. Acronyms

4 Bibliography

1. ***Di Girolamo, A. et al.*** ST GNSS NMEA specification and commands. *Naples : STMicroelectronics, 2011. Release 3.1.*

2. ***Lawicel AB***, CAN USB Manual, *November 2011, Version 1.0D*

3. ***Di Girolamo, A. et al.*** GNSS API specification. *STMicroelectronics, 2015. Release 4.2.*

4. **CAESAR DataSystem,** QIC GPS Datasheet, Message Format on the CAN bus

5 NoC Application Overview

The NoC (NMEA Over CAN) application is conceived to transmit NMEA information computed by the STA8088/STA8090 GNSS receiver over the CAN bus; moreover, the NoC application is also conceived to asynchronously receive CAN frames to setup the list of CAN frames that shall be transmitted.

Concerning the unsolicited and synchronous transmission of messages on the CAN bus (from the STA8088/STA8090 device to the host), the open specification of the QIC GPS compact receiver (http://www.caemax.de) have been chosen and implemented.

In the following sections, firstly the SW architecture to transmit & receive CAN messages is described; then the format of the CAN messages used in the NoC application is detailed.

5.1 SW Architecture

When the NoC application is started, two brand new tasks are created.

The first one (`NoC_tx_process`) has been conceived to transmit the QIC messages over the CAN bus at 1Hz rate; the second one (`NoC_rx_process`) has been conceived to receive asynchronous messages from the host.

The TX process is triggered by the FIX event (working at 1Hz in this application) and every second a new set of QIC messages are prepared and transmitted over the CAN bus.

First step is to select the QIC messages to be sent using a message list; in the beginning this message list is initialized to 0xF (all-pass filter) and it can be modified when the 0xA1 message (coming from the host – NoC Viewer) is received by the STA8088/STA8090 device (see [5.4] for a more detailed description).

In a second step the proper information is retrieved from the GNSS library (by using the GNSS lib API, see [3]); hence the fields of each of the QIC messages in the message list are fulfilled.

In the end, the selected QIC messages are sent over the CAN bus.

Note: gpOS API are used by default in this demo application but user can choose to use instead FreeRTOS API by setting `DEMO_USE_FREERTOS_API` compilation switch in `build.mk` file.

CAN Messages Specification

The NoC application makes use of the CAN 2.0A protocol running at 125kbps, continuously sending data in output to the CAN bus (unsolicited mode) with a rate of 1Hz.

In Figure 1 the format of the CAN frame is reported.

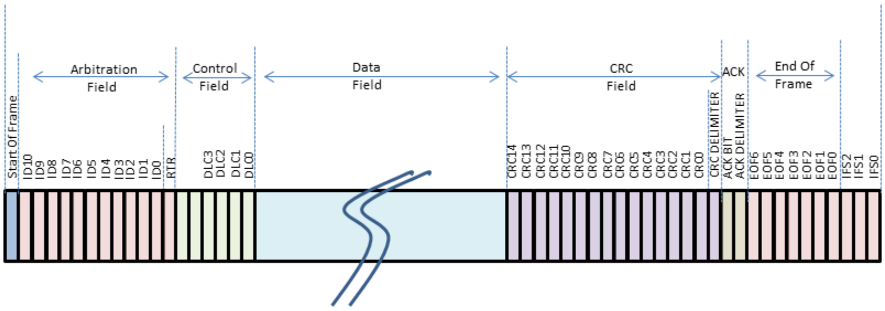


Figure 1: CAN Standard Frame Format

In TX, the NoC application purpose is to fill the proper NMEA information into the *Data Field* of the CAN frame as specified by the QIC specification (refer to [4]) and hence transmit over the CAN bus the standard CAN frames with the proper Message IDs.

In RX, the NoC application simply consists in receiving only the message 0xA1, by which the QIC message list can be configured.

Clearly the mechanism by which the CAN frames are transmitted and received over the CAN bus is provided by the Os20 CAN service and by the STA8088/STA8090 CAN LLD.

Field Name		Length (bit)	Description
Start Of Frame (SOF)		1	Denotes the Start Of a Frame transmission
Arbitration Field	Identifier	11	The Identifier of the CAN Frame; it represents the message priority in the CAN bus
	RTR (Remote Transmission Request)	1	Must be dominant (0) for data frames and recessive (1) for remote request frames[1]
Control Field	IDE (IDentifier Extension bit)	1	Must be dominant (0) for standard frame format with 11-bit identifiers[2]
	Reserved Bit	1	
	DLC (Data Length Code)	4	Number of bytes of data (from 0 to 8; having 4 bits the max number of bytes could be 2^4 = 16, but CAN protocol 2.0 allow 8 bytes as max number of bytes in the payload)
Data Field		0-63	Data to be transmitted; the length in bytes dictated by the DLC field
CRC Field	CRC	15	Cyclic Redundancy Check
	CRC Delimiter	1	Must be Recessive (1)
ACK Field	Ack Slot	1	Transmitter sends recessive (1) and any receiver can assert a dominant (0)
	Ack Delimiter	1	Must be Recessive (1)
End Of Frame (EOF)		7	Must be Recessive (1)

Table 3: CAN Standard Frame Format Description

[1] it shall be dominant (0) as no Remote Request are used by the NoC application

[2] it shall be dominant (0) as NoC application only uses Standard CAN frames

5.2 TX Messages

In the following sections, the specification of the QIC messages used by the NoC application is reported.

For each message, firstly the associated **Messaged ID** and **DLC** are reported.

Hence the parameters allocated inside the **Data Field** of each CAN frame are specified.

5.2.1 QIC Message 1

Message ID: `1800` (`0x708`)

DLC: `8`

Parameter	Position (Format)	Range of values	Units
Time Day	Byte 0 (unsigned char)	1 .. 31	
Time Month	Byte 1 (unsigned char)	1 .. 12	
Time Year	Byte 2 (unsigned char)	0 .. 99	
Time Hour	Byte 3 (unsigned char)	0 .. 23	
Time Minute	Byte 4 (unsigned char)	0 .. 59	
Time Second	Byte 5 (unsigned char)	0 .. 59	
Altitude	Byte 6, 7 (LSB, MSB)	0 .. 17999	“m” (1m)

Table 4: Format of the QIC Message1

5.3 QIC Message 2

Message ID: `1801` (`0x709`)

DLC: `8`

Parameter	Position (Format)	Range of values	Units
Latitude Degrees	Byte 0 (Bit 0 .. 7)	-90 … +90	“Deg”
Latitude Minutes	Byte 1 (Bit 8 .. 13)	0 .. 59	“Min”
Latitude Seconds	Byte 2, 3 (Bit 16 .. 28)	0 .. 5999	“Sec”
Longitude Degrees	Byte 4 (Bit 32 .. 40)	-180 … +180	“Deg”
Longitude Minutes	Byte 5 (Bit 41 .. 46)	0 .. 59	“Min”
Longitude Seconds	Byte 6, 7 (Bit 48 .. 60)	0 .. 5999	“Sec”

Table 5: Format of the QIC Message 2

5.3.1 QIC Message 3

Message ID: `1802` (`0x70A`)

DLC: `4`

Parameter	Position (Format)	Range of values	Units
Speed	Byte 0, 1 (LSB, MSB)	0 .. 9999	“Km/h”
Heading	Byte 2, 3 (LSB, MSB)	0 .. 3599	“Deg”

Table 6: Format of the QIC Message 3

5.3.2 QIC Message 4

Message ID: `1803` (`0x70B`)

DLC: `8`

Parameter	Position (Format)	Range of values	Units
Number Of Active Satellites	Byte 0 (Bit 0 .. 3)	0 … 12	“”
Type	Byte 0 (Bit 4 .. 7)	0	“”
Number Of Visible Satellites	Byte 1 (unsigned char)	0 .. 16	“”
PDOP (positional accuracy)	Byte 2, 3 (LSB, MSB)	0 .. 999	“m”
HDOP (horizontal accuracy)	Byte 4, 5 (LSB, MSB)	0 .. 999	“m”
VDOP (vertical accuracy)	Byte 6, 7 (LSB, MSB)	0 .. 999	“m”

Table 7: Format of the QIC Message 4

5.4 RX Messages

The only Message ID enabled to be received by the NoC application running on the STA8088/STA8090 device is the 0xA1.

This is obtained by setting the Object 1 (the higher priority one) of the STA8088/STA8090 CAN peripheral with 0x7FF filtering mask (*FULL CAN* mode); in this way every time the host (the NoC Viewer in our test) transmits a CAN frame with 0xA1 message ID, the first byte of the payload is decoded; hence the NoC application retrieves the QIC message list and applies it to the next set of QIC messages to be transmitted.

Here the format of the 0xA1 CAN frame.

Parameter	Position (Format)	Range of values	Description
Byte-0	Byte 0 (Bit 0 .. 1)	0 … 3	The QIC message list is here transmitted
Byte-1	reserved for future use		
Byte-2	reserved for future use		
Byte-3	reserved for future use		
Byte-4	reserved for future use		
Byte-5	reserved for future use		
Byte-6	reserved for future use		
Byte-7	reserved for future use		

Table 8: payload specification for the 0xA1 CAN frame

6 Testing the NoC Application (NoC Viewer)

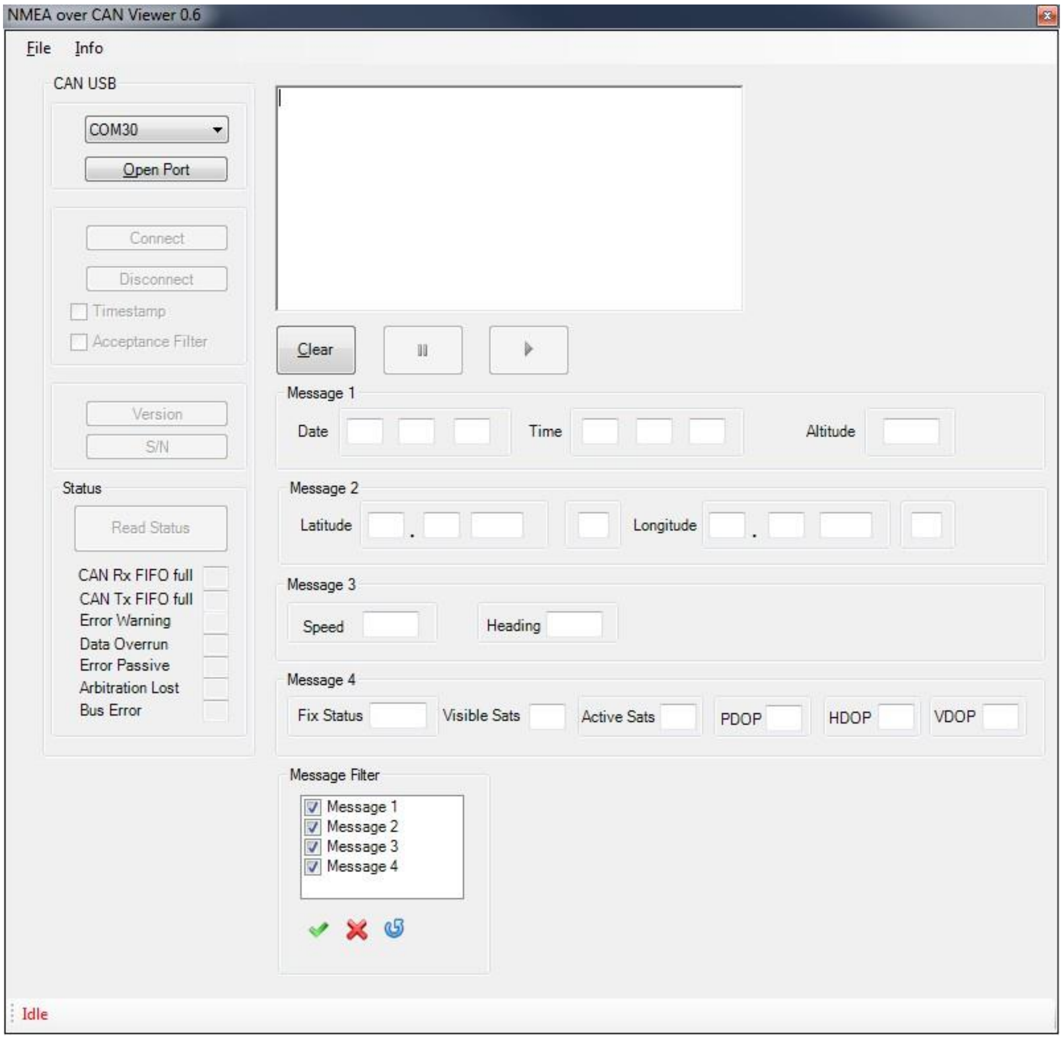
In order to test the NoC application, a GUI running in the host and named **NoC Viewer** has been developed. The NoC Viewer reports the information sent by the NoC application running on the STA8088/STA8090 device over the CAN bus and received by the host (PC with Windows 7).

The interface of the NoC Viewer with the CAN bus is provided by means of the CANUSB connector from Lawicel (http://www.can232.com/canusb/).

In the following sections, the procedure describing how to setup the CAN USB connection and the main features of the NoC Viewer are reported.

6.1 CAN USB Connection Procedure

When the NoC Viewer is opened, it is in the “*Idle”* state as reported in Figure 2



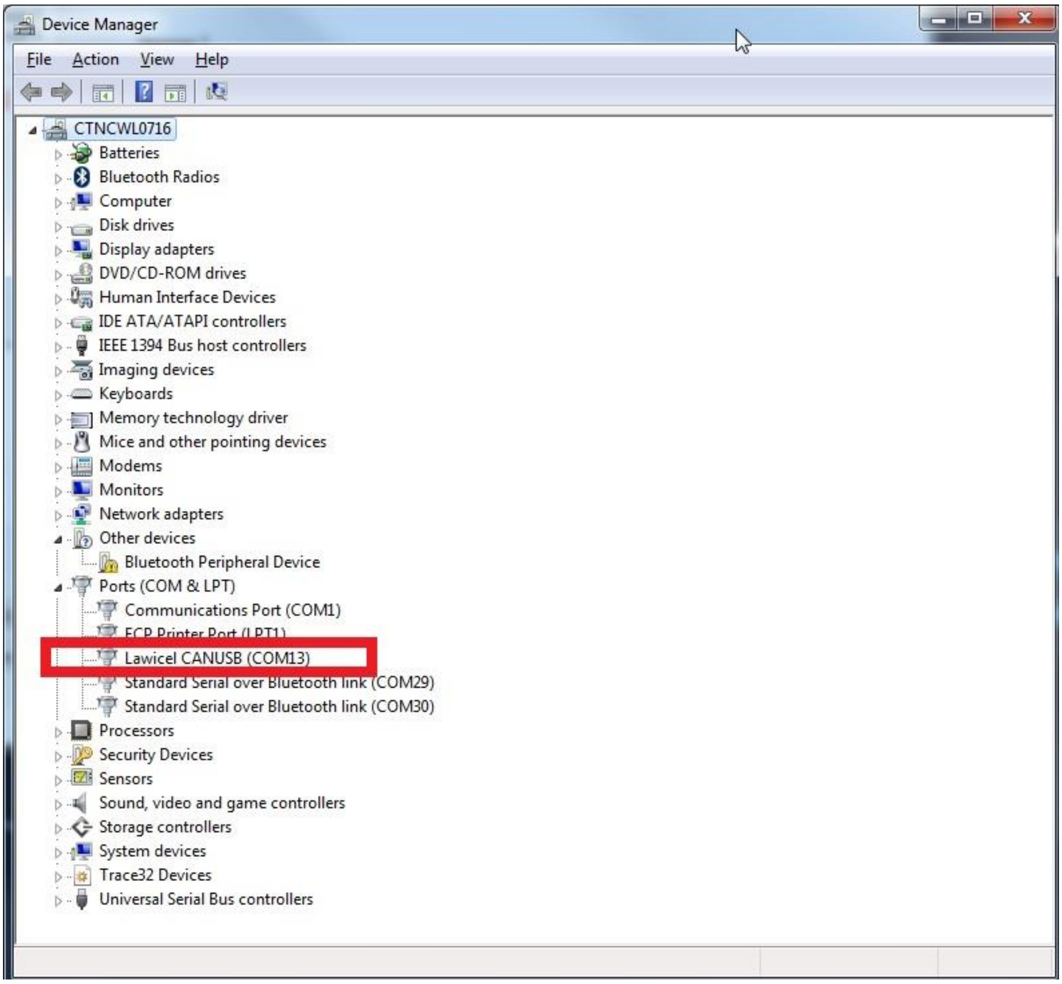
COM30

☑ Message 1
☑ Message 2
☑ Message 3
☑ Message 4

Idle

Figure 2: NoC Viewer: CANUSB is in Idle state

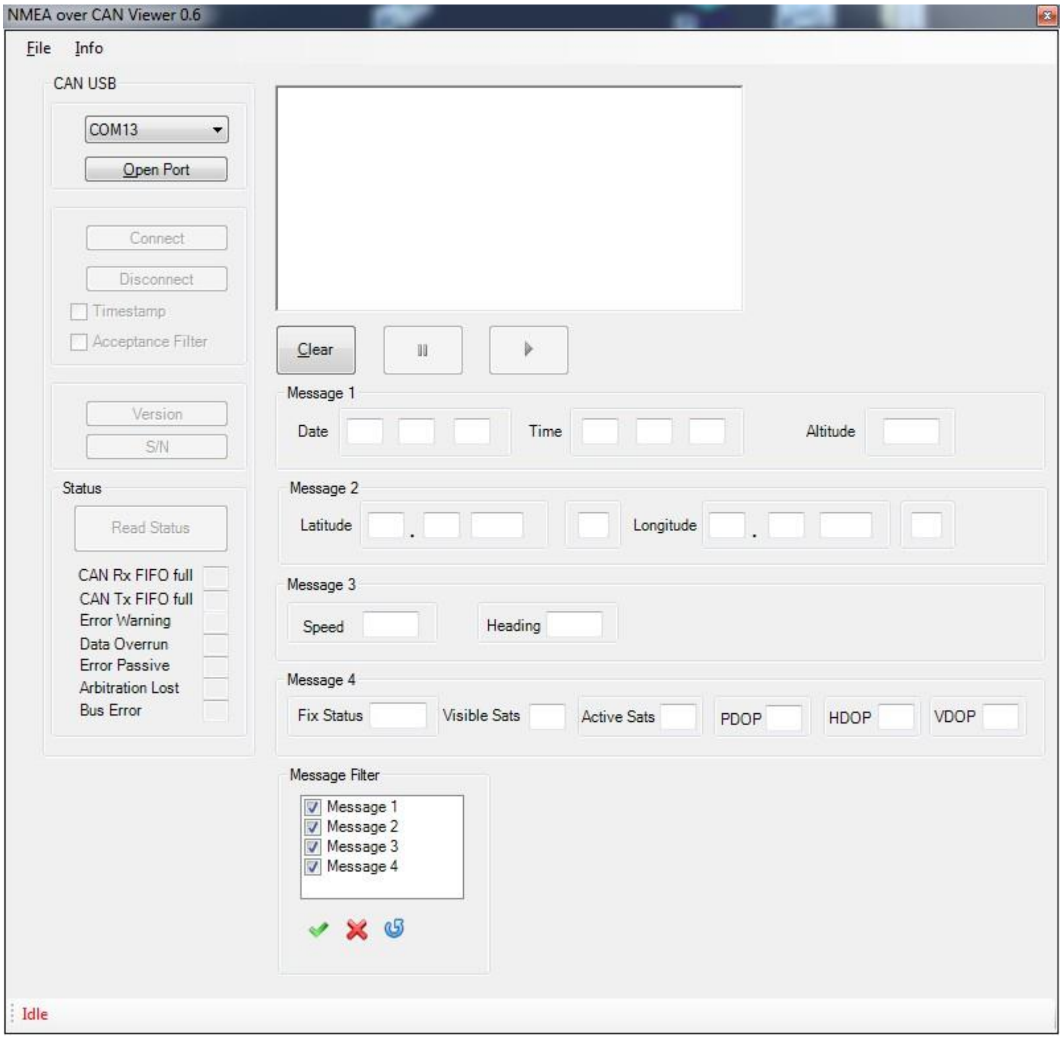
By opening the Windows Device Manager, the Virtual COM associated to the Lawicel CAN USB can be discovered (COM13 in the example, see Figure 3)



CTNCWL0716
- Batteries
- Bluetooth Radios
- Computer
- Disk drives
- Display adapters
- DVD/CD-ROM drives
- Human Interface Devices
- IDE ATA/ATAPI controllers
- IEEE 1394 Bus host controllers
- Imaging devices
- Keyboards
- Memory technology driver
- Mice and other pointing devices
- Modems
- Monitors
- Network adapters
- Other devices
  - Bluetooth Peripheral Device
- Ports (COM & LPT)
  - Communications Port (COM1)
  - ECP Printer Port (LPT1)
  - Lawicel CANUSB (COM13)
  - Standard Serial over Bluetooth link (COM29)
  - Standard Serial over Bluetooth link (COM30)
- Processors
- Security Devices
- Sensors
- Sound, video and game controllers
- Storage controllers
- System devices
- Trace32 Devices
- Universal Serial Bus controllers

Figure 3: Device Manager: Lawicel CANUSB VCOM

Inside the CANUSB message box select the virtual COM associated to the Lawicel CANUSB (COM13 in the example).



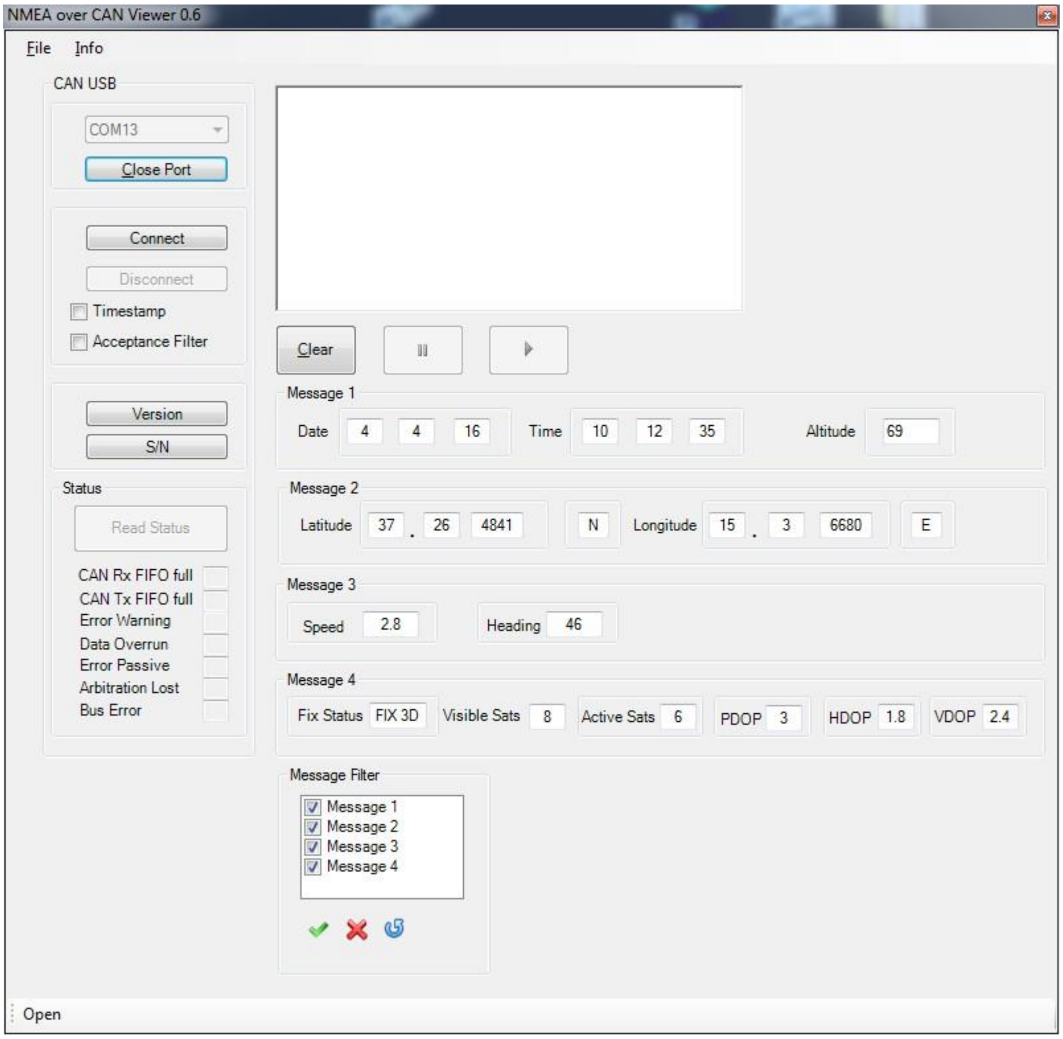
COM13

☑ Message 1
☑ Message 2
☑ Message 3
☑ Message 4

Idle

Figure 4: NoC Viewer: Lawicel CANUSB VCOM selected, Closed state

and press the “*Open Port*” button.



COM13

☐ Timestamp

☐ Acceptance Filter

CAN Rx FIFO full □
CAN Tx FIFO full □
Error Warning □
Data Overrun □
Error Passive □
Arbitration Lost □
Bus Error □

Message 1
Date 4 4 16 Time 10 12 35 Altitude 69

Message 2
Latitude 37 . 26 4841 N Longitude 15 . 3 6680 E

Message 3
Speed 2.8 Heading 46

Message 4
Fix Status FIX 3D Visible Sats 8 Active Sats 6 PDOP 3 HDOP 1.8 VDOP 2.4

☑ Message 1
☑ Message 2
☑ Message 3
☑ Message 4

Open

Figure 5: NoC Viewer: CANUSB in Open state

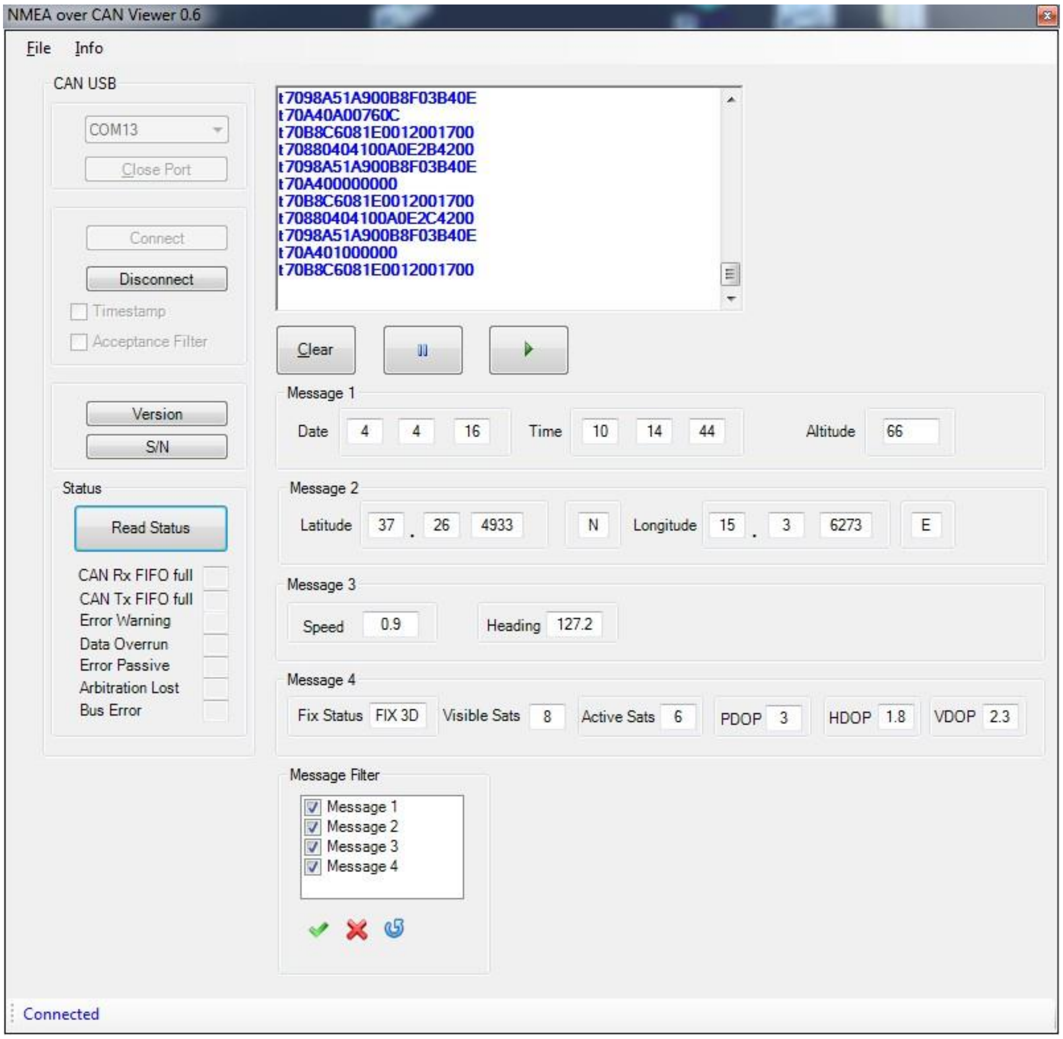
Once the COM associated to the CAN USB has been opened, the NoC Viewer is in the "*Open"* state; in this state, the **Acceptance Filter** and **Timestamp**[3] check boxes can be checked.

When selecting the **Acceptance Filter** checkbox, the NoC Viewer configures the ACn and AMn registers of the Philips SJA1000 CAN controller inside the CAN USB, in order to filter only the CAN frames with Message IDs in the range [0x700, 0x70F].

When the **Acceptance Filter** checkbox is not selected, any other CAN frame in the bus is received by the NoC Viewer (i.e. in this way the terminal of the NoC Viewer can be used to listen to all the CAN frames in the bus).

In the "*Open"* state, press the "*Connect*" button to connect CAN USB to the CAN bus.

[3] NoC Viewer version 0.6 still not implement this feature



COM13

□ Timestamp
□ Acceptance Filter

```
t7098A51A900B8F03B40E
t70A40A00760C
t70B8C6081E0012001700
t70880404100A0E2B4200
t7098A51A900B8F03B40E
t70A400000000
t70B8C6081E0012001700
t70880404100A0E2C4200
t7098A51A900B8F03B40E
t70A401000000
t70B8C6081E0012001700
```

Message 1: Date 4 4 16 Time 10 14 44 Altitude 66

Message 2: Latitude 37 . 26 4933 N Longitude 15 . 3 6273 E

Message 3: Speed 0.9 Heading 127.2

Message 4: Fix Status FIX 3D Visible Sats 8 Active Sats 6 PDOP 3 HDOP 1.8 VDOP 2.3

CAN Rx FIFO full □
CAN Tx FIFO full □
Error Warning □
Data Overrun □
Error Passive □
Arbitration Lost □
Bus Error □

☑ Message 1
☑ Message 2
☑ Message 3
☑ Message 4

Figure 6: NoC Viewer: Connected state

In the “*Connected”* state the **Status** button allows to retrieve on the terminal the information on the status of the CAN USB (refer to [2]).

6.2 CAN USB group box

The CAN USB group box allows the user to control the CAN USB emulator. Apart from the Open/Close and the Connect/Disconnect buttons already described in [6.1], the CAN USB group box embeds:

- the **Version** button to retrieve on the terminal the Version of the CAN USB;
- the **S/N** button to retrieve on the terminal the Serial number of the CAN USB
- the **Read Status** button to retrieve the status of the CAN USB[4]

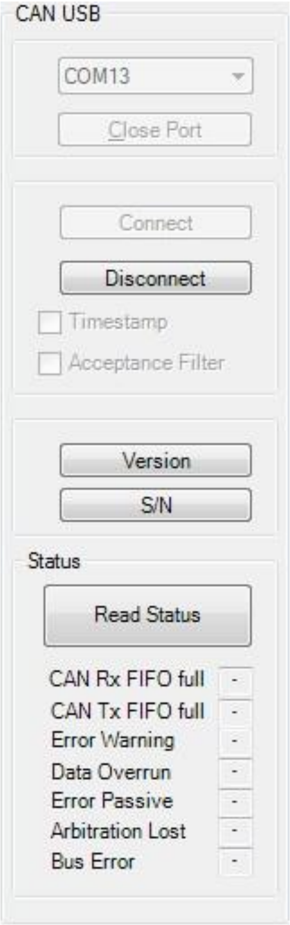


Figure 7: NoC Viewer: CAN USB group box

[4] refer to [2] for a detailed description of the CAN USB status information

6.3 NoC Viewer Terminal

The NoC Terminal shows the CAN messages sent by the STA8090 device to the NoC Viewer and some information of the CAN USB emulator (version, S/N).

Three buttons are provided in order to control the message flow on the NoC Viewer terminal:

1. the **Clear** button to clear all the content of the NoC Viewer terminal,
2. the **Pause** (⏸) button to stop the messages flow on the NoC Viewer,
3. the **Play** (▶) button to restart the message flow if previously stopped.

6.4 NoC Messages

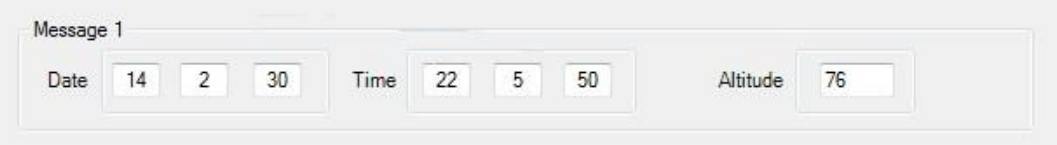
All the information carried by each of the 4 QIC messages are displayed by the NoC Viewer and updated by the received CAN frames at 1Hz rate.

In this way a real time comparison can be done with the information in output on the NMEA port of the STA8088/STA8090 device.

In the following each of the 4 displayed messages are described.

6.4.1 Message 1

The information carried by the message 1 (refer to 5.2.1) are displayed in the **Message 1** box as reported in Figure 8



Message 1

Date 14 2 30 Time 22 5 50 Altitude 76

Figure 8: NoC Viewer: Message 1

The information in the **Date** box reports:

- day (2 digits),
- month (2 digits),
- Year (2 digits).

The information in the **Time** box reports:

- hour (2 digits),
- minutes (2 digits),
- Seconds (2 digits).

The information in the **Altitude** box reports:

- Altitude (4 digits) in meters.

6.4.2 Message 2

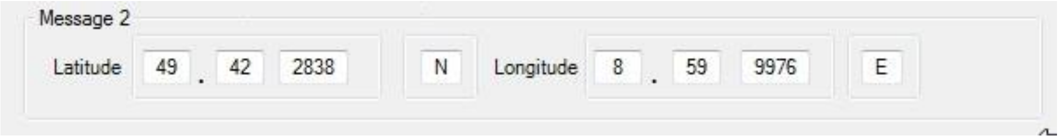
The information carried by the message 2 (refer to 5.3) are displayed in the **Message 2** box as reported in Figure 9

The information in the **Latitude** box reports:

- degrees (2 digits),
- minutes (2 digits),
- parts minutes (4 digits)
- lat direction (N or S)

The information in the **Longitude** box reports:

- degrees (2 digits),
- minutes (2 digits),
- fractional minutes (4 digits)
- lon direction (E or W)



Message 2

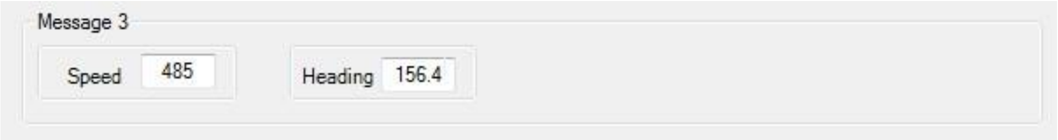
Latitude 49 . 42 2838 N

Longitude 8 . 59 9976 E

Figure 9: NoC Viewer: Message 2

6.4.3 Message 3

The information carried by the message 3 (refer to 5.3.1) are displayed in the **Message 3** box as reported in Figure 10



Message 3

Speed 485

Heading 156.4

Figure 10: NoC Viewer: Message 3

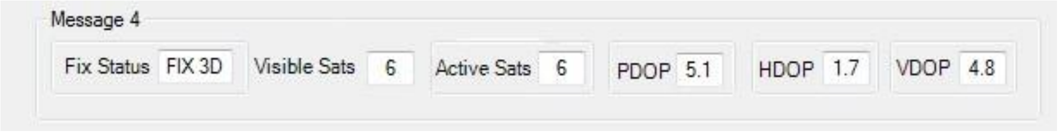
6.4.4 Message 4

The information carried by the QIC message 4 (refer to 5.3.2) are displayed in the **Message 4** Box as reported in Figure 11

The **Fix Status** field can be:

- NO_FIX
- FIX_2D
- FIX_3D

The **Visible Sats** reports the number of satellites. The **Active Sats** reports the number of satellites. **PDOP**, **HDOP** and **DOP** reports related information



Message 4

Fix Status FIX 3D
Visible Sats 6
Active Sats 6
PDOP 5.1
HDOP 1.7
VDOP 4.8

Figure 11: NoC Viewer: Message 4

6.5 Message Filtering

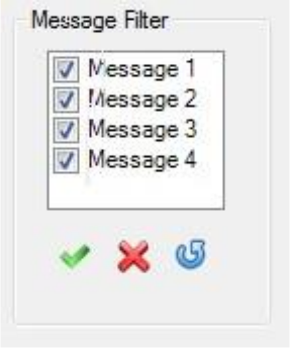
The Message Filter list box allows the user to select which messages shall be displayed by the NoC Viewer. This is achieved by transmitting the CAN frame with message ID 0xA1 to the STA8088/STA8090 device, as described in 5.4. The 0xA1 CAN frame carries (in its first byte of the payload) the message list that shall be applied by the “Can Tx process” before sending the QIC messages to the host (NoC Viewer).

By default all the Messages are selected.

The **Select All** (✓) button allows the user to select all the messages.

The **Unselect All** (✗) button allows the user to unselect all the messages in the list box.

Once the chosen messages are selected, the **Refresh** button (↻) shall be pressed in order to send the proper message list to the STA8088/STA8090 device.



Message Filter

☑ Message 1
☑ Message 2
☑ Message 3
☑ Message 4

Figure 12: NoC Viewer: Message Filtering List box

7 Disclaimer

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.