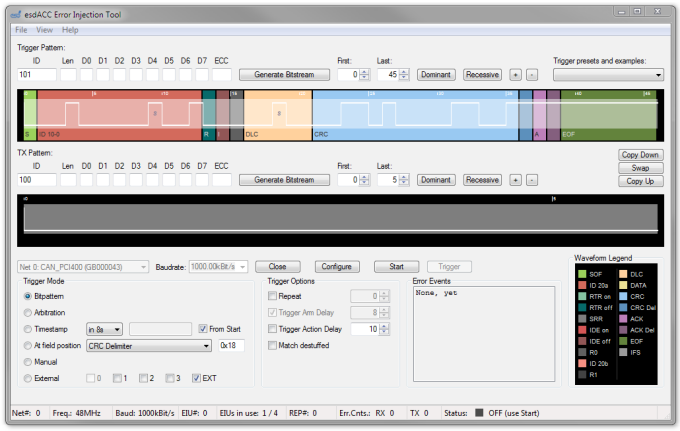
Error Injection Tool

GUI Tool for esdACC Error Injection



Software Manual

to Product C.1113.01

esd electronic system design gmbh
Vahrenwalder Str. 207 • 30165 Hannover • Germany
http://www.esd.eu
Phone: +49 (0) 511 3 72 98-0 • Fax: +49 (0) 511 3 72 98-68

N O T E

The information in this document has been carefully checked and is believed to be entirely reliable. **esd** makes no warranty of any kind with regard to the material in this document, and assumes no responsibility for any errors that may appear in this document. In particular descriptions and technical data specified in this document may not be constituted to be guaranteed product features in any legal sense.

esd reserves the right to make changes without notice to this, or any of its products, to improve reliability, performance or design.

All rights to this documentation are reserved by **esd**. Distribution to third parties, and reproduction of this document in any form, whole or in part, are subject to **esd**'s written approval**.**

© 2015 esd electronic system design gmbh, Hannover

esd electronic system design gmbh
Vahrenwalder Str. 207
30165 Hannover
Germany

Phone:	+49-511-372 98-0
Fax:	+49-511-372 98-68
E-Mail:	info@esd.eu
Internet:	www.esd.eu

Trademark Notices

CANopen® is a registered community trademark of CAN in Automation e.V.
Windows® is a registered trademark of Microsoft Corporation in the United States and other countries.

All other trademarks, product names, company names or company logos used in this manual are reserved by their respective owners.

Document file:	I:\Texte\Doku\MANUALS\PROGRAM\CAN\C.1113.01_esdACC_Error_Injection_GUI\esdACC_Error_Injection_Tool_Software_en_10.odt
Date of print:	2015-11-25
Document type number:	DOC0800

Document History

The changes in the document listed below affect changes in the hardware as well as changes in the description of the facts, only.

Rev.	Chapter	Changes versus previous version	Date
1.0	-	First English manual	2015-11-25

Technical details are subject to change without further notice.

Table of contents

1. Overview

The esdACC Error Injection Tool is a menu-controlled program which is used for controlling the esdACC Error Injection. This manual explains in detail the menus and functions of esdACC Error Injection Tool. The description is followed by application examples.

The Error Injection is an extension to an FPGA base d CAN controller board. Via several trigger modes it is possible to generate all kinds of errors on CAN networks. This implies the possibility to test existing systems with reproducible CAN errors. Depending on the configuration, a CAN controller can have several units, so it is possible to create complex error scenarios by combining these units.

1.1 System Requirements

- Linux or Windows® XP (or later)
- CAN Interface with Error Injection Support
  (see compatible list: https://esd.eu/en/products/can-error-injection)
- CAN driver from esd

1.2 Program Call

The program is automatically installed with the CAN SDK. Before calling the program the driver has to be started.

The Error Injection Tool can run parallel to other CAN applications. It is possible to run multiple instances simultaneously. This allows you to define more complex trigger scenarios.

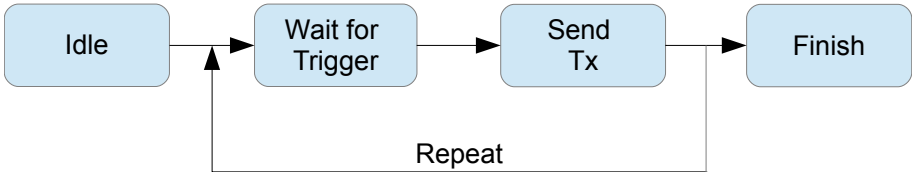
For example you may:
- cascade trigger units (use Trigger externally)
- have different trigger conditions on the same CAN bus

1.3 Quick Start

General Procedure:
1. Use combobox to select a CAN net
2. Click ***Open*** to get a control of a free Error Injection Unit
3. Select your desired trigger scenario
4. Click ***Configure*** to configure the Error Injection Unit with your scenario
5. Click ***Start*** to activate the Error Injection Unit

INFORMATION
Before committing a new configuration, the Error Injection Unit needs to be stopped!

The following figure shows the Error Injection Unit states:



1.4 Error Injection Introduction

The Error Injection is divided into several Error Injection units. These units can be assigned to the different CAN controllers in the FPGA. Figure 1 shows a simplified overview of the FPGA structure. The Error Injection is an additional module to a CAN IP core. It works parallel to the normal function of the CAN controller.

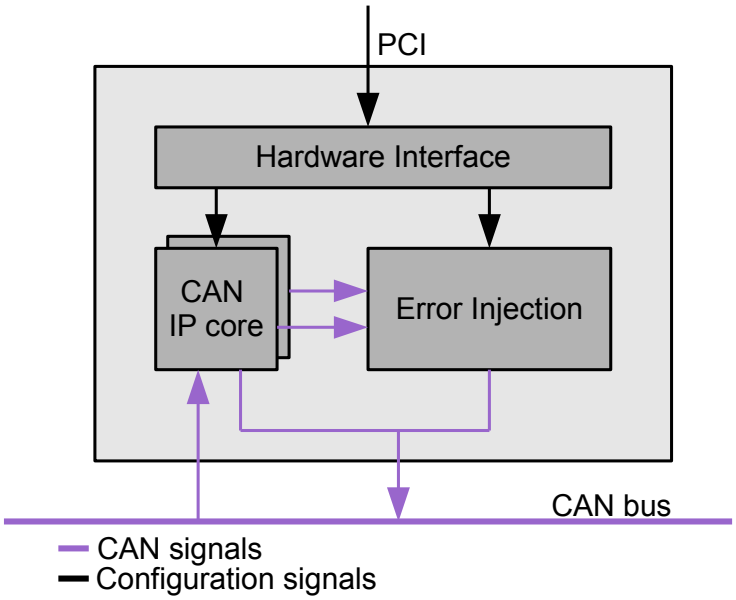


Figure 1: Overview CAN controller board with Error Injection

The CAN protocol works on the wired-AND principle. A recessive level only occurs on the CAN bus, if no transceiver is actively driving the bus. The bus is dominant when at least one transceiver is sending. A dominant level always overrides a recessive level.

The error injection uses standard CAN hardware, thus only dominant bits can be injected, as dominant bits cannot be overridden by recessive bits. Through this characteristic all types of errors can be generated except for the acknowledge error.

More information you will find at: https://esd.eu/en/products/can-error-injection and in the referenced paper https://esd.eu/sites/default/files/CAN_Error_Injection.pdf.

If you want to write your own Error Injection application, you will find the complete API documentation in the NTCAN Part 1 [1].

2. Functions of the User Interface Elements

2.1 Display of the Error Injection Tool Window

The Error Injection window is structured as shown below:

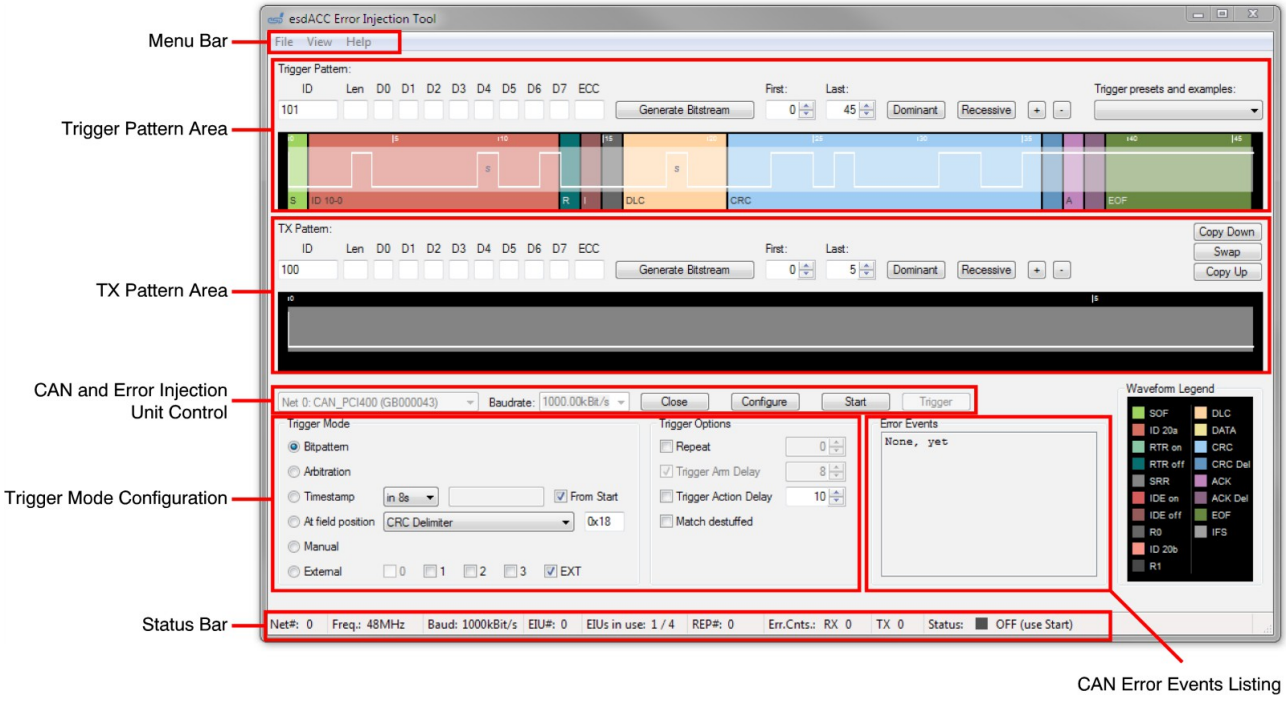


Figure 2: Structure of the Error Injection Tool Window

2.2 Menu Bar

The main menu in the menu bar contains the menu items: **File**, **View** and **Help.**

In menu item **File** the configuration of the Error Injection Tool can be opened or saved. Additionally you can configure a Default configuration or restore the defaults or factory defaults.

With ***Export TX-Pattern As...*** the TX Pattern is exported binary in an new text file. In the requester that opens, you can select how much time quanta is used per bit.

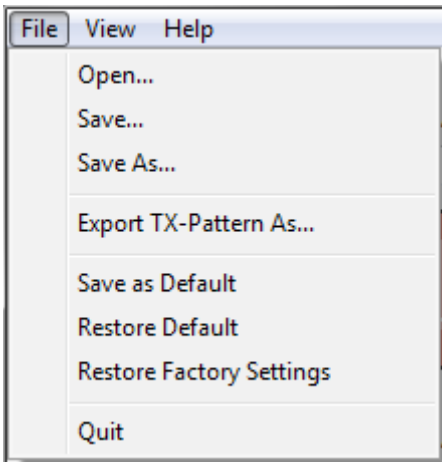


Figure 3: Menu item File

The view of the waveforms can be configured in menu item **View.** A click to ***Copy Waveform Trigger to TX*** copies the configuration of Trigger Pattern to TX Pattern. ***Copy Waveform TX to Trigger*** does the reverse. With ***Swap Waveforms*** swaps the configuration of Trigger Pattern and TX Pattern. You will find the same functions on the right side between the waveforms on extra buttons (***Copy Down, Swap, Copy Up***).

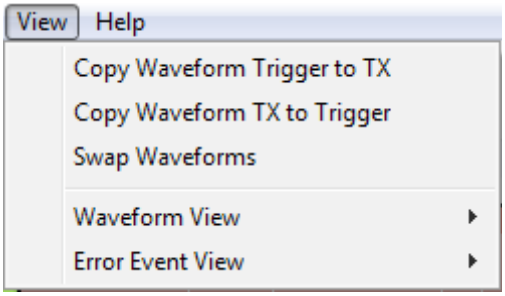


Figure 4: Menu item View

You will find a quick help in the menu item **Help** and an information box at ***About.***

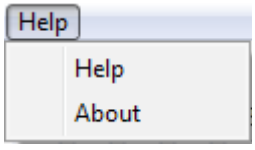


Figure 5: Menu item Help

2.3 Waveform Area

In the Waveform Area is done the configuration of Trigger Pattern and TX Pattern. The waveform Trigger Pattern is only used in Trigger Mode Bitpattern.

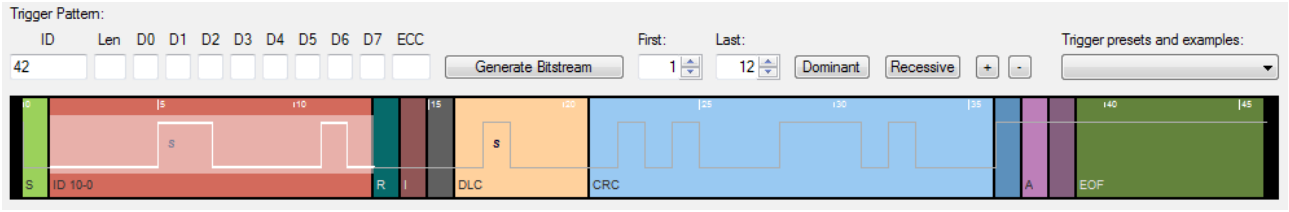


Figure 6: Trigger Pattern Area

In the following table describes the configuration of the waveforms.

ID	CAN frame ID with up to eight hexadecimal digits (bit 29 for CAN 20b frame)
Len	CAN frame DLC with up to two hexadecimal digits (bit 4 for RTR frame)
D0 - D7	CAN frame data byte 0 to 7 with up to two hexadecimal digits
ECC	When generating a new bitstream, you may use the ECC field to automatically generate a bitstream with an erroneous field. In order to find a code, simply use the ***At field position*** combobox and copy the generated code. Enter two hexadecimal digits.
Generate Bitstream	Start generating the bitstream from the given values.
First / Last	Can be used either to select only a part of the shown bitstream or to determine the length of for constant stream generation
Dominant	Generate a bitstream of n dominant bits (***First*** / ***Last*** determine n)
Recessive	Generate a bitstream of n recessive bits (***First*** / ***Last*** determine n)
+ / -	Add / delete a bit to the end of the bitstream

To edit the waveform use the left double click to toggle the bit. In ***Trigger Pattern*** you can toggle the mask bit (just like a don't care bit) with right double click. Right double click on ***TX Pattern*** sets the force recessive, for this is needed a special external hardware component.

A “Point of transmission” is required for the correct beginning. This is created by a small and simple transmit automat, which sends an user defined bit stream to the CAN bus via a shift register. The bit stream is sent without CRC calculation or bit stuffing. There is no CAN bus feedback, so the transmission will not be terminated, if CAN error frames are encountered. This unit is called “CAN TX”.

The configuration is done with the TX Pattern and in the same usage as the Trigger Pattern.

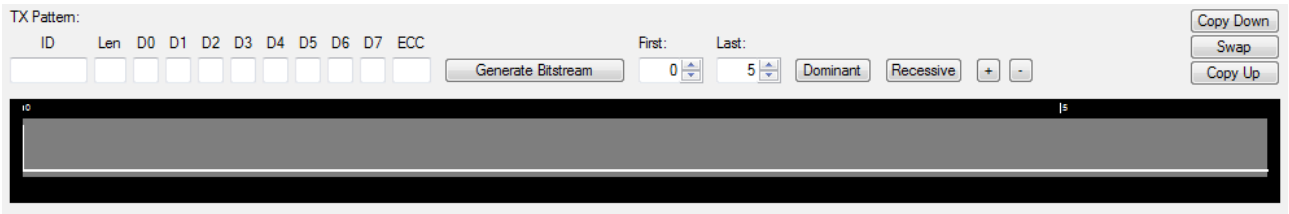


Figure 7: TX Pattern Area

The background colour of the waveform is described in the legend shown in Figure 8. The Inter Frame Space (IFS) is not part of the CAN frame, but the display of the IFS can be activated in ***Waveform View*** in menu item **View**.

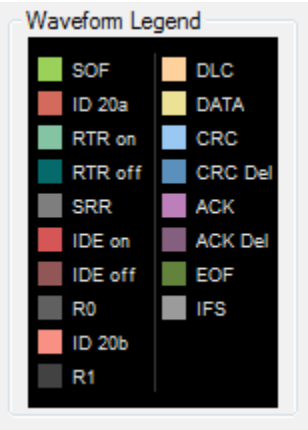


Figure 8: Waveform Legend for Trigger Pattern and TX Pattern

2.4 CAN and Error Injection Unit Control

With the left selector choose the CAN net you want to use. The Error Injection is only available on esdACC supported boards (see compatible list: https://esd.eu/en/products/can-error-injection).

The ***Baudrate*** selector can be used as text box or as selector. Choose ***Don't touch*** if the baudrate is already set.
The following baudrate formats are available (*X* = hexadecimal digits and *D* = decimal value):

- *nD* - Numerical baudrate in bit/s
- *bXXXXXXXX* - BTR value
- *XXXXXXXX* - Used directly as parameter to canBaudrateSet()

Open the CAN net with ***Open*** or close is with ***Close.***

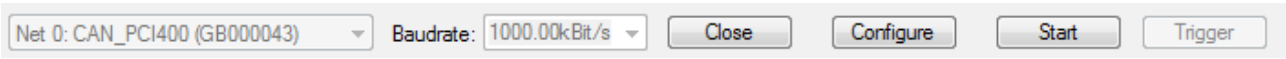


Figure 9: CAN and Error Injection Unit control

If you finish the configuration of an Error Injection Unit use the ***Configure*** button to write the configuration to the CAN interface. If you change the configuration which is not saved to the hardware the button changes to ****Configure****.

To start the Error Injection Unit, use the ***Start*** button. If you want to disarm the Error Injection Unit use the ***Stop*** button. In status “Waiting for Trigger” you can use the Trigger button as well, to start the TX pattern immediately.

2.5 Trigger Mode Configuration

To activate the TX Pattern there are some trigger modules defined:

- Bitpattern
- Arbitration
- Timestamp
- Field Position
- Manual
- External Input

Only one trigger unit can be activated at a time. You can choose one of the Trigger modes in the ***Trigger Mode*** area.

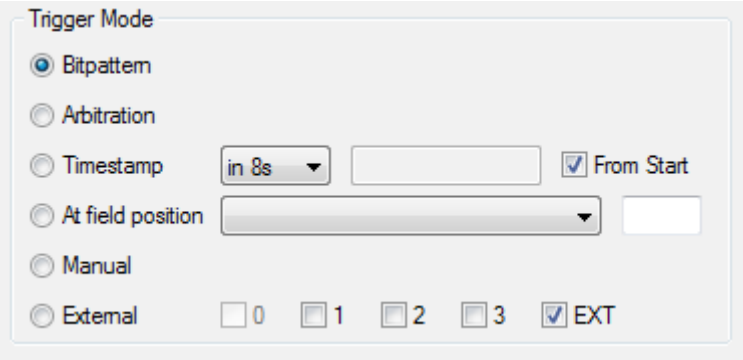


Figure 10: Trigger Mode configuration

The trigger ***Bitpattern*** can search for a user defined bit stream. If the bit stream matches the sampled CAN bit stream, the CAN TX Module is triggered. Additionally, a bit mask defines the region which is to be compared. The configuration is done in ***Trigger Pattern*** area.

Every trigger mode has some ***Trigger Options***, the first three shown in Figure 11 are available in any trigger mode. With ***Repeat*** the Error Injection Unit is restarted after finish and is looking again for a trigger point. With the value you can set the number of times to repeat (0 = infinite).

The ***Trigger Arm Delay*** sets the number of bittimes to delay the rearming of an Error Injection Unit in repeat mode.

The ***Trigger Action Delay*** sets the number of bittimes to delay the injection of TX pattern.

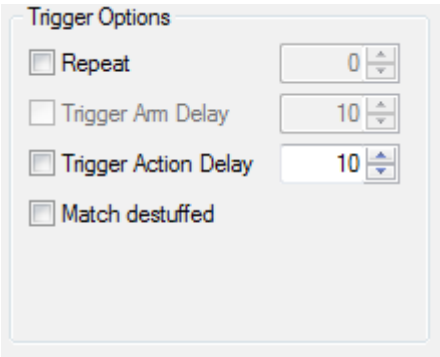


Figure 11: Trigger Options for Mode Bitpattern

In trigger ***Bitpattern*** it can be chosen between the direct CAN bit stream and the destuffed bit stream.

The trigger ***Arbitration*** sends a bit stream via the CAN TX module under the rules of arbitration. It is possible to send correct CAN frames or CAN frames with errors. This provides for example the means to send CAN frames with CRC errors otherwise impossible with standard CAN controllers.

For this trigger mode an ***Abort on error*** is available at trigger options. So the transmission of the TX pattern is aborted at any CAN error which is detected by the CAN controller.

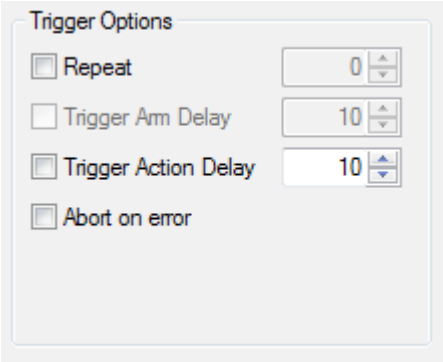


Figure 12: Trigger Options for Mode Arbitration

With trigger ***Timestamp*** the CAN TX module is triggered by an expired timer. The esdACC CAN boards have an internal 64 bit timestamp and this is compared with the user defined timestamp. When the time elapses, the CAN TX module is triggered.

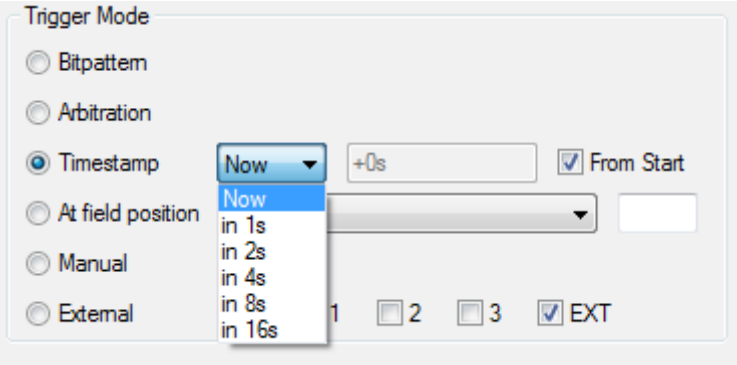


Figure 13: Trigger Mode configuration for Mode Timestamp

In the Error Injection Tool the timestamp can be set as offset from start or as an absolute value. Basically you can either enter absolute timestamps or offsets relative to the starting point:

- *0xX* - Absolute timestamp in hexadecimal notation
- *+0xX* - Timestamp offset in hexadecimal notation
- *+Dus* - Timestamp offset in decimal microseconds
- *+Dms* - Timestamp offset in decimal milliseconds
- *+Ds* - Timestamp offset in decimal seconds

A “Bus free” option is available for this trigger mode. The Error Injection Unit will wait for bus free or bus idle if the timestamp has elapsed.

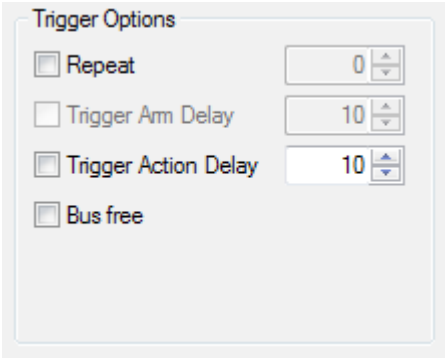


Figure 14: Trigger Options for Mode Timestamp

The trigger ***Field Position*** triggers at a specific position in a CAN frame. For example, it can be set to trigger within data length code or on CRC delimiter. At the esdACC boards an error code of a detected error can be used to configure this trigger module to repeat or simulate a CAN error. The available positions are shown in Figure 15.

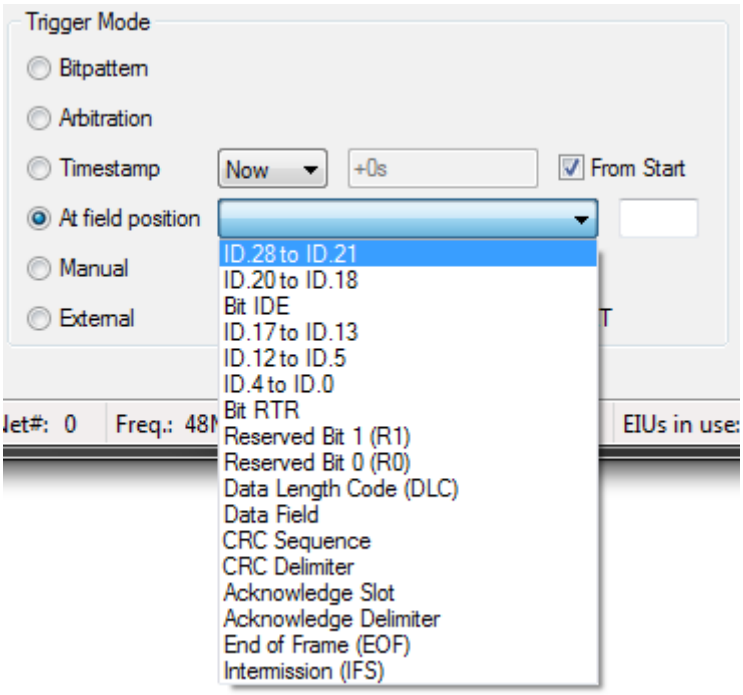


Figure 15: Trigger Mode configuration for Mode "At field position"

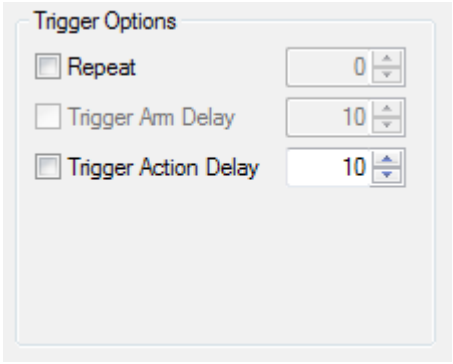


Figure 16: Trigger Option for other modes

For the last three trigger modes there are no special options available.

The trigger ***Manual*** triggers by clicking the ***Trigger*** button.

The trigger ***External Input*** triggers on an external signal. This may also be used to combine trigger sources or to trigger on an event provided by another Error Injection Unit on another CAN bus. Each Error Injection Unit has a trigger out signal which can be an external trigger source for another trigger module. Choose which trigger unit is the trigger source via the checkboxes. Additionally, it is possible to trigger on an external IO pin.

2.6 CAN Error Events Listing

In the CAN error events listing the CAN events from the selected CAN net are shown. The information can be configured in ***Error Event View*** in the **View** menu item.

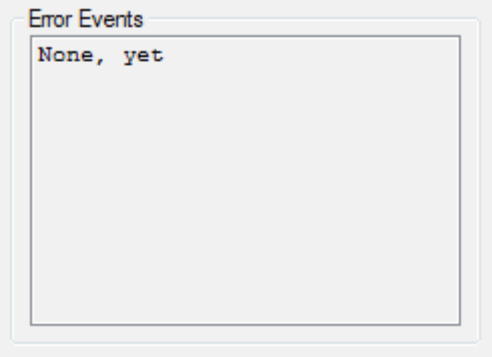


Figure 17: CAN Error Events listing

2.7 Status Bar

In the status bar you will find information about the CAN net and the Error Injection Unit.

Figure 18: Status Bar

Net#	The chosen CAN net number
Freq.	The timestamp frequency of the CAN interface
Baud	The chosen CAN baudrate of the CAN net
EIU#	The number of the **E**rror **I**njection **U**nit
EIUs in use	Shows how much Error Injection units are in use
Rep#	Shows the repeat counter of the unit
Err. Cnts.	Shows the RX and TX CAN error counters of the CAN controller
Status	Shows the Status of the Error Injection Unit: Status: No handle opened Status: OFF (use Start) Status: WAIT_TRIGGER Status: SENDING Status: FINISHED (use Stop)

3. Examples

To get a quick start up with the Error Injection you can use our example configurations.

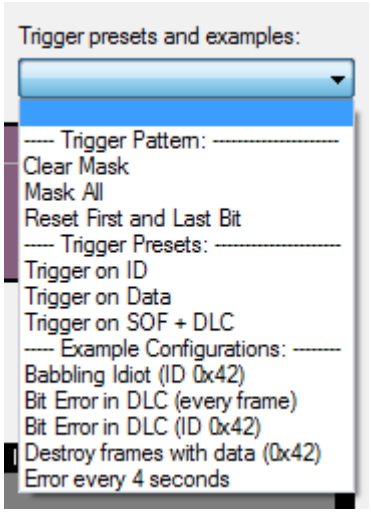


Figure 19: Trigger presets and examples

A first scenario is the so called “Babbling Idiot” or “ID Pollution” and simulates a defective sensor which sends continuously with a fixed CAN ID. An Error Injection Unit is configured in ***Trigger Arbitration Mode*** and the repeat flag is set but without any delay time. The ***TX pattern*** is set to a complete CAN Frame with the CAN ID 0x42 for example.

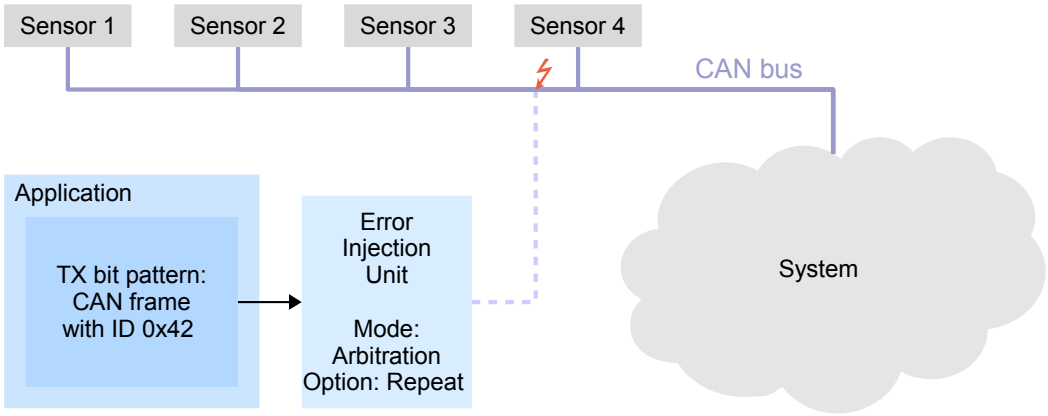


Figure 20: Use case scenario 1: "Babbling Idiot", "ID Pollution"

Figure 20 shows the complete scenario. The activated Error Injection Unit sends continuously CAN frames with ID 0x42 back to back, so any CAN frame with an ID greater 0x42 will loose the arbitration. Thus the user can check whether vital CAN frames (e.g. emergency shut down commands like CANopen® NMT messages) can be sent correctly and whether the system remains reactive.

A second scenario (Bit Error in DLC (ID 0x42)) is an error frame injection on the sensor 2. An Error Injection Unit is configured in ***Pattern Matching Mode*** and the ***Trigger Pattern*** is set to the CAN ID of data from sensor 2. The ***TX Pattern*** (the bit stream which is sent by CAN TX module when triggered) is set to an Error Frame of six dominant bits.

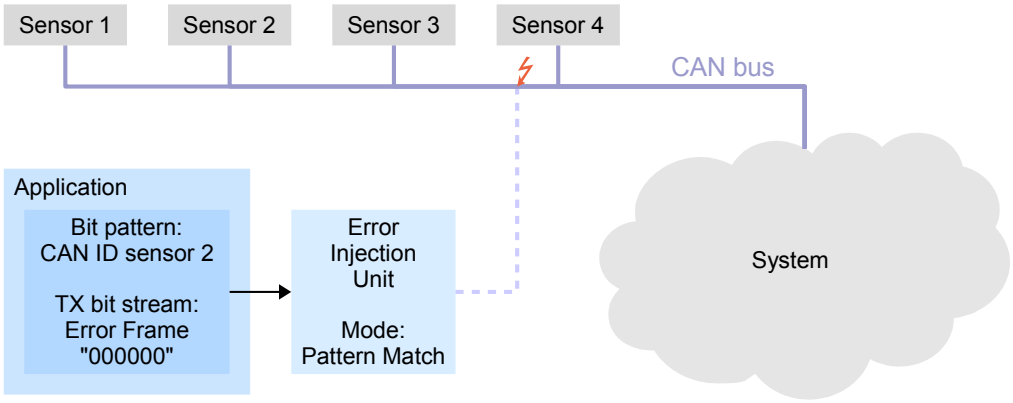


Figure 21: Use case scenario 2: defective sensor

Figure 21 shows the complete scenario. Sensor 2 may be a temperature or an acceleration sensor which sends his value at any given rate. The activated Error Injection Unit compares the current bit stream with the defined sensor CAN ID. As soon as the ***Trigger Pattern*** match module detects the CAN ID from sensor 2 in the current CAN bit stream, the stored error frame is sent and the sent CAN frame from sensor 2 is destroyed.

4. References

[1] NTCAN Part 1: Application Developers Manual Rev.4.7, Order No.: C.2001.21, 2015-10-12, esd electronic system design gmbh

5. Order Information

The Error Injection Tool is contained in the scope of delivery of the CAN-SDK an can be downloaded from our website www.esd.eu.

PDF Manuals
Manuals are available in English and usually in German as well. For availability of English manuals see table below.
Please download the manuals as PDF documents from our esd website www.esd.eu for free.

Manuals		Order No.
Error Injection Tool-ME	Hardware manual in English	C.1113.21
CAN-API-ME	NTCAN-API: Application Developers Manual NTCAN-API: Driver Installation Guide	C.2001.21

Table 1: Available manuals

Printed Manuals
If you need a printout of the manual additionally, please contact our sales team: sales@esd.eu for a quotation. Printed manuals may be ordered for a fee.