LAUTERBACH
DEVELOPMENT TOOLS

FlexRay Protocol Analyzer

MANUAL

FlexRay Protocol Analyzer

FlexRay Protocol Analyzer

Version 06-Jun-2024

General Function

FlexRay PROTOanalysis requires one of the following hardware configurations:

- PowerIntegrator with single-ended Probe
- PowerTrace-II / PowerTrace-III with single-ended Probe
- PowerProbe

Three target connections have to be done:

- FlexRay-Plus signal
- FlexRay-Minus signal
- GND

The PROTOanalyzer decodes the whole transmission, considering the FlexRay specifications.

The results of the Protocol Analysis are time-correlated to other TRACE32 tools like a program flow trace. This way there is a close linkage between FlexRay activity and program-execution and vice versa.

Features

The FlexRay PROTOanalyzer has a basic implementation of the Fibex XML file to reassign/rename the corresponding transmission which is defined by ID, offset and repetition. The extracted information is stored into a t32.flx file which allows editing and improves the speed of the PROTOanalyzer.

Both CRC (11- and 24-bit) are checked and marked if they are faulty. The PROTOanalyzer can be configured to confirm Channel-A or Channel-B CRC.

Chart display also uses the Fibex XML implementation. It displays the period of time in which a transmission was active.

STATistic display shows how often specific transmissions were active. Additionally it calculates the min/max and average transmission times.

How to use the PROTOanalyzer

First of all a recording of the FlexRay activity has to be done. Based on this recording various types of protocol displays can be done.

The PROTOanalyzer requires a FlexRay specific file (FlexRay.dll) which contains the algorithm to analyze and display the recorded FlexRay transmission.

The source code and the FlexRay.dll file are part of the installation DVD. The code matches the TRACE32 Protocol Analysis interface. It is open for user modifications to add the display of more detailed application specific information.

Documentation on installation DVD:

- **protocol_app.pdf**
- **protocol_user.pdf**

Format:	*<trace>*.**PROTOCOL.LIST** *<file> <channel+> <channel-> <frequency> <crc>*
<file>:	**FlexRay.dll**

<channel+>	signal name of positive FlexRay channel
<channel->	signal name of negative FlexRay channel
<frequency>	FlexRay clock (e.g. 10.MHz)
<crc>	“A” or ”B” (for Channel A/B CRC calculation)

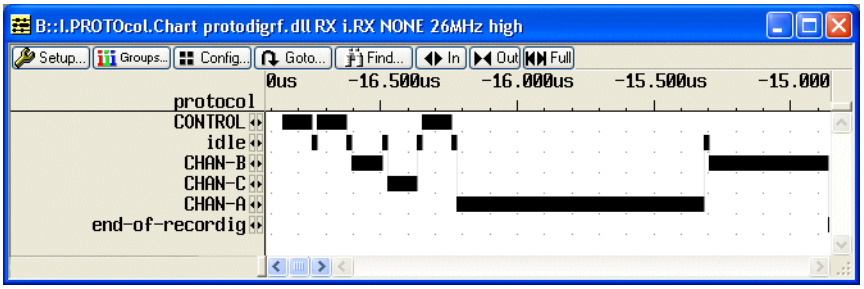
A wrong CRC selection will cause an error of the 24-bit CRC calculation.

record	spare	ti.back	ti.zero
-009640	ID: 0007 LEN: 016 CYC #44 00 00 00 00 00 00 00 00	2.211ms	740.370s
-009464	ID: 0008 LEN: 016 CYC #44 1d f7 ff fe ff fe ff fe	35.860us	740.370s
-009275	ID: 0025 LEN: 016 CYC #44 00 00 00 00 00 00 00 00	576.196us	740.371s
-009085	ID: 0026 LEN: 016 CYC #44 00 00 00 00 00 00 00 00	35.844us	740.371s
-008900	ID: 0040 LEN: 016 CYC #44 40 09 00 00 00 00 00 ff	474.200us	740.371s
-008692	ID: 0041 LEN: 016 CYC #44 c3 09 00 00 00 00 00 00	34.000us	740.371s
-008493	ID: 0042 LEN: 016 CYC #44 03 09 00 00 00 00 00 00	34.004us	740.371s
-008291	ID: 0043 LEN: 016 CYC #44 00 00 00 00 ff ff ff ff	34.004us	740.371s
-008068	ID: 0044 LEN: 016 CYC #44 00 00 00 00 00 00 00 00	34.004us	740.371s
-007892	ID: 0046 LEN: 016 CYC #44 92 f2 d0 07 7d 00 0e ff	69.828us	740.371s
-007661	ID: 0047 LEN: 016 CYC #44 00 00 00 00 00 00 00 00	34.004us	740.371s
-007488	ID: 0051 LEN: 016 CYC #44 00 00 00 00 00 00 00 00	134.188us	740.371s
-007288	ID: 0052 LEN: 016 CYC #44 00 00 00 00 00 00 00 00	34.004us	740.372s
-007129	ID: 0053 LEN: 016 CYC #44 00 00 00 00 00 00 00 00	34.000us	740.372s
-006948	ID: 0060 LEN: 016 CYC #44 f4 08 00 00 00 00 00 00	238.024us	740.372s
-006758	ID: 0061 LEN: 016 CYC #44 6b 09 00 00 00 00 00 ff	34.000us	740.372s
-006562	ID: 0063 LEN: 016 CYC #44 00 00 00 00 00 00 00 00	68.004us	740.372s
-006363	ID: 0064 LEN: 016 CYC #44 00 00 00 00 00 00 00 00	34.008us	740.372s
-006193	ID: 0066 LEN: 016 CYC #44 ff ff ff ff ff ff ff ff	69.812us	740.372s
-005978	ID: 0067 LEN: 016 CYC #44 d8 f2 6e 82 64 21 ff ff	34.004us	740.372s
-005708	ID: 0068 LEN: 016 CYC #44 ff ff ff ff ff ff ff ff	34.000us	740.372s

Format: *<trace>*.**PROTOCOL.CHART** *<file> <channel+> <channel-> <frequency> <crc>*

The command format has the same structure as that one above. Only the BaseCommand is different.

This visualization shows the duration of every transmission and when the bus is in idle state. It's very useful for checking the traced transmissions for any errors or rare message IDs.



Format	*<trace>***.PROTOCOL.STATistic** *<file> <channel+> <channel-> <frequency> <crc>*

The command format has the same structure as that one above. Only the BaseCommand is different.

This visualisation shows the share of every transmission type. It's very useful for checking the transmission timing. This way critical timings can be found easily.

samples: 952. total: 91.226ms

protocol	time	min	max	avr	count	ratio%	1% 2%
(other)	0.000	0.000	0.000	0.000	0.	0.000%	
ID_0040	461.372us	24.184us	24.352us	24.283us	19.	0.505%	←
Idle	79.524ms	9.648us	2.186ms	167.068us	476.(-1)	87.172%	▬▬▬▬▬▬▬▬
ID_0041	460.876us	24.184us	24.352us	24.257us	19.	0.505%	←
ID_0042	242.688us	24.184us	24.352us	24.269us	10.	0.266%	←
ID_0043	461.212us	24.188us	24.352us	24.274us	19.	0.505%	←
ID_0044	460.732us	24.188us	24.352us	24.249us	19.	0.505%	←
ID_0046	459.620us	24.180us	24.204us	24.190us	19.	0.503%	←
ID_0047	459.640us	24.180us	24.200us	24.192us	19.	0.503%	←
ID_0051	460.872us	24.184us	24.356us	24.256us	19.	0.505%	←
ID_0052	460.896us	24.184us	24.352us	24.258us	19.	0.505%	←
ID_0053	461.196us	24.184us	24.352us	24.273us	19.	0.505%	←
ID_0060	339.128us	24.184us	24.356us	24.223us	14.	0.371%	←
ID_0061	461.184us	24.184us	24.352us	24.273us	19.	0.505%	←
ID_0063	461.376us	24.184us	24.356us	24.283us	19.	0.505%	←
ID_0064	461.192us	24.184us	24.352us	24.273us	19.	0.505%	←
ID_0066	459.696us	24.180us	24.204us	24.194us	19.	0.503%	←
ID_0067	459.624us	24.180us	24.204us	24.191us	19.	0.503%	←
ID_0068	459.608us	24.180us	24.204us	24.190us	19.	0.503%	←
ID_0069	435.436us	24.180us	24.204us	24.191us	18.	0.477%	←
ID_0074	436.996us	24.188us	24.352us	24.278us	18.	0.479%	←
ID_0088	436.704us	24.184us	24.356us	24.261us	18.	0.478%	←
ID_0089	435.432us	24.180us	24.204us	24.191us	18.	0.477%	←
ID_0007	436.864us	24.184us	24.364us	24.270us	18.	0.478%	←

Fibex Implementation

FIBEX is a XML standard, which is used to describe the FlexRay communication structure. Almost every information , like payload length or even the meaning of the payload, is described there.

The PROTOanalyzer includes a low-level implementation to extract the FlexRay message names out of the FIBEX XML file.

Attention:

- The FIBEX file name has to be **T32.XML**
- The extracted information is save to file **T32.FLX**.

The extractor algorithm automatically searchs for the string: "<fx:FRAME ID". In this part of the XML file the IDs and its corresponding strings can be found in a quite simple structure.

The ID, Offset and Repitition are found by the PROTOanalyzer by searching for a "**.**". After every "**.**" in this line, it reads first the ID, then the Offset and last the Repitition. If the Fibex structure is changed the PROTOanalyzer code has to be modified.

The message name itself is not in the same line, it comes along in the line starting with "<ho:LONG-NAME xml:lang=".

The whole procedure is aborted when it reaches the "</fx:FRAMES>" sequence which marks the end of the this part of the code.

T32.flx The interims file

By default the PROTOanalyzer uses the **T32.FLX** file to assign message names to certain message IDs. If this file is not present the PROTOanalyzer searches for a **T32.XML** file to generate a new **T32.FLX** file. If none of the two files can be found, no name assignment can be done, the message IDs are displayed as numeric values.

The T32.FLX file is a textfile with a very simple format:

```
-ID- -OFF -REP ---String---
0223 0005 0016 DummyMessage
0222 0000 0016 FlexRayMessage
```

The T32.flx is used to improve the flexibility. The user is allowed to add new messages or to do minor modifications.

FAQ

Q.: I copied a new t32.xml to the directory, but the labels are still the old ones.

A.: Please check if the directory contains a t32.flx file and delete it because the program prefers the flx-file.

Q.: The traced raw material is full of spikes.

A.: Try to adjust the threshold of the Probe, this should improve the result.

Q.: The PROTOanalyzer shows a lot of protocoll errors.

A.: Check if the wires are correctly connected and you entered first the BP- and second the BM-Channel. If this isn't successful check for spikes and adjust the threshold.