

Goebelyzer Analyzer

User Manual

The Goebel Company

Copyright: The Goebel Company 2009



Purpose

This manual describes the software for the Goebelyzer System offered by The Goebel Company. This includes all operation of the analyzer normally performed by a user.

Notice

Information in this manual has been carefully reviewed and is believed to be accurate. The Goebel Company shall not be liable for errors contained herein. The Goebel Company reserves the right to make changes or additions to the software described herein.

Contact

For technical or other inquiries contact:

email: info@GoebelEtc.com

The Goebel Company 12486 Prowell Leavenworth WA 98826 USA Phone: 206-601-6010



Table of Content

1.Introduction	
2.Applicable Documents/Web Sites	
3.Quick GUI Layout	
4.Preparing Capture Options	
4.1General Options Notebook Page	
4.1.1Button Box	
4.1.2Capture Files Frame:	
4.1.3Display Options Frame	
4.1.4Stop Capture Frame	
4.1.5Capture Frame	10
4.2AFDX Options Page	
4.2.1Button Box	
4.2.2Network Selection Frame	
4.2.3Passthru Modify Filter Parameters Frame	16
4.2.4Speed Mode Frame	17
4.2.5Capture TX Packets Frame	
4.3P2P Options Page	18
4.3.1Channel Initialization (Enable/Speed) Frame	
4.3.2Button Box	
4.4A429 Options Page	
4.4.1RX Channel Init Frame	20
4.4.2Button Box	
4.5CAN Bus Options Page	
4.5.1SJA1000 Parameters Device Frame	23
4.5.2Button Box	
4.5.3Button Box	
4.6FSCC Options Page	20 26
4.6.1FSCC Parameters Channel Frame	20 26
4.6.2Button Box	
4.7Spacewire Options Page	
4.7.1Spacewire Parameters Channel Frame	20
4.7.2Button Box	
4.7.2 Button Box	
4.8AIDev Options Fage	
4.8.1Select Devices Frame	31
4.8.2Capture Fintels/ Higgers Frame	
4.9Real-Time Display Options Page	32
4.9.1 Select Realtime Display Data Frame.	32
4.9.2Selected Real-time Display Data Frame.	دد دد
4.9.3Button Box	
5. Active Data Capture	
5.1Capture Pkt Info Pages	
5.1.1Button Box	
5.2Capture Info Page	
5.3Packet Info Page	
5.4Data Display Page	
6.Digital Filters	
6.1Digital Filter Display Pane	
6.2Expression Syntax	
6.3Protocol Data Formats	
6.4Examples:	
7.Plotting	
8. Data Definition Files	
8.1AFDX XML Data File	
8.1.1Packet Data	
8.1.2Element Data	54



8.2AFDX VL XML File	
8.3A429 XML Data File	
8.3.1Label Data	
8.3.2Element Data	
8.4A429 Bus Data File	
8.5A429 Equipment ID Data File	
8.6P2P XML Data File	
8.6.1Label Data	
8.6.2Element Data	
8.7M1553 XML Data File	
8.7.1Label Data	
8.7.2Element Data	
9. Frequently Asked Questions	



1. Introduction

This manual discusses topics that are unique to the Goebelyzer System. Most other Goebelyzer functionality will be as described in the ethereal user's guide. The ethereal user's guide should be the first place to search for information. This document will be a supplement to the ethereal user's guide.

The AFDX,A429, P2P, CAN, and ALLDEV interfaces are unique to the Goebelyzer System. The steps necessary to capture data on these interfaces will include special options/conditions to perform a capture described in the guide.

This document will start with a quick description of the GUI layout including the terms and layout. Next an outline for performing a capture. Sections on capture options for each interface, and information pages displayed during a live capture. Finally, a chapter on various interface data file formats.

The Goebelyzer works with many unique avionics interfaces and provides the ability to defined the format of the payload data received on an interface. This functionality will allow the user to create meaningful names and display data in formats as described in an Interface Control Document.

2. Applicable Documents/Web Sites

The following lists contain information that is useful in learning to use the Goebelyzer System.

- 1. Ethereal User Guide V2.0.2 for Ethereal 0.10.12 (or latest) written by Richard Sharpe, Ed Warnicke, and Ulf Lamping. <u>Http://www.ethereal.com/docs</u>
- 2. Capture Filters. Defines the syntax of capture/trigger filters <u>http://wiki.ethereal.com/CaptureFilters</u>, .
- 3. Mike Horn Capture Filter Tutorial. Provides a good tutorial on use of capture filters. http://home.insight.rr.com/procana
- 4. Mark 33 Digital Information Transfer System (DITS) Part 1, ARINC Specification 429P1-15 Published September 1, 1995.
- 5. SJA1000 Data Sheet 2000 Jan 04 Philips Semiconductor.

3. Quick GUI Layout.

This section is a quick description of some terms and the layout of the GUI. The ethereal document contains more information on the GUI at <u>http://www.ethereal.com/docs</u>.

At the top, is a menu bar. It has pulldowns for File (read/write capture files), Edit (various packet find/mark operations), View (what is displayed in the GUI), Go (go to a specific packet), Capture (start/stop capture), and Analyze (enable/disable protocols, create display filters). The next layer down is the toolbar, it contains buttons to perform commonly executed functionality. The most used buttons are Options (display options menu), Capture (Execute a capture), Stop (Stop Execution of a capture). These toolbar buttons correspond to operations in the menu bar but are more convenient to the user.

The next hortizontal bar is used to create/execute display filters. Display Filters are used to eliminate packets not matching the result of the display filter boolean expression. For Example: only display packets with vl=1212 (eth.vl==1212) or udp destination port=2323 (udp.dstport==2323) or both conditions (eth.vl==1212 and udp.dstport==2323) are displayed with these filters. This will allow the user to only view packet meeting the defined condition. The expression button will display a dialog used to create expression conditions. Display filters are described in more detail in a later chapter and the ethereal user's guide at http://www.ethereal.com/docs.

The next window section down is the packet display list. It is the list of packets received on the device sorted in time with various columns to provide unique information related to the packet and interface. The columns may include data such as time, packet names, port numbers, or IP addresses.

GOEBEL

Goebelyzer

On the lower left is the packet tree window pane. When a packet is selected in the packet list, the packet tree will display a list of protocols contain in the packet. Each protocol can be expanded to view all the data items of the protocol. Using a data definition file, the user can define payload data that will be displayed and formatted in this pane.

On the lower right is the byte dump pane. When a packet is selected, the pane displays a raw byte dump of all data contained in the packet as shown for the highlighted packet.

<u>File</u> <u>E</u> dit	<u>V</u> iew <u>G</u> o	o <u>C</u> aptur	e <u>A</u> nal	yze <u>S</u> ta	atistics <u>H</u> e	lp														
Interfaces	Options	Capture	e St		Restart	Dpen (Save As	X Close	Refresh	Prin		Find		(Back	Forward	२००० Jump to	Top	Bottom	Colorizo	e Auto Scroll
Filter:							~	Expression	. ≽ <u>C</u> lear	🕜 Ap	ply									
Time .	Frame F	Protocol	Dir	Error	Pkt/VL Name	e	Src Port/0	Chan Car	d/DestChan	Dest	tPort/Lab	el C	RC	Pay	load/Info			VL Num	SrcIP	DestIP
4.160183	969 1	UDP	A RX				5			5				Sre	Port: r	je Dest 1	Port: r	ie 5	10.0.0.	5 10.1.0.5
4.160183			B RX				5			5						je Dest 1				5 10.1.0.5
4.160212	971 U	UDP	A RX				6			6						Dest Po:		6		6 10.1.0.6
4.160213	972 l	UDP	B RX				6			6				Sre	Port: 6	Dest Po:	rt: 6	6	10.0.0.0	5 10.1.0.6
4.212087	973 A	AFDX_1	A RX	NONE	vl_valfa	c_sim_dat	a 1			1		a	b069ba	5 Pay	load Len	gth: 80		1	10.0.0.1	L 10.1.0.1
4.212088	974 A	AFDX_1	BRX	NONE	vl_valfa	c_sim_dat	a 1			1		5	293fe4	Pav	load Len	gth: 80		1	10.0.0.1	l 10.1.0.1
4.212109	975 l	UDP	A RX				2			2				Sre	Port: 2	Dest Por	rt: 2	2	10.0.0.2	2 10.1.0.2
4.212110	976 l	UDP	B RX				2			2				Sre	Port: 2	Dest Por	rt: 2	2	10.0.0.2	2 10.1.0.2
4.212132			A RX				3			3						Dest Por		3		3 10.1.0.3
4.212133			B RX				3			3						Dest Por		3		3 10.1.0.3
4.212159			A RX				4			4						Dest Po:		4		10.1.0.4
4.212160	980 l	UDP	B RX				4			4				Sre	c Port: 4	Dest Po:	rt: 4	4	10.0.0.4	10.1.0.4
¢									111		_									>
▷ Frame	974 (127	bytes	on wir	e, 127	bytes c	aptured)				٨	0000	03 0	0 00 0	0 00	01 00 30	f7 02 41	0 40 08	00 45 00		0K@E
					40, Dst:	- 12 Q	:00:00:0	1			0010	00 6	ed e	f 00	00 01 11	b7 8f 0a	a 00 00	01 0a 01	.1	
									1 (10.1.0		100000				01 00 58			8d 00 8d		Х
					tcpmux						100000				00 00 00			00 00 00		
	rce port:		- St.	roru.	ucpillux	(1), Dat	rort. tej	hindr (1)			1000000					00 00 00				C
	Stan	50-00 - 00-000									1000 C C C C C C C C C C C C C C C C C C					00 00 4				
	tination	port: t	cpmux	(1)												ff 73 80				
	gth: 88																			
Chee	cksum: Ox	:0000 (r	none)																	
⊽ Avioni	ics Full-	Duplex	Switch	ed Eth	nernet: P	ort 1				4										
rese	erved_					=141,0	(8d				2									
dw_s	slowongnd	linvld1_				=141,02	(0000 a8d	0000 100	00 1101)											
slow	wongndinv	1d1				=FALSE	(
	wongnd1						(
	lautoslat	and							 	: 										
									••••	: 										
	dsyspress					=0.0000														
	dsyspress					=0.0000														
	dsyspress	elr				=0.0000														
mach	h					=0.0000	000													
true	eairspeed	L				=0.0000	000													
				100		0.000		1		¥ >										
il Ne centre		01 20061	0120010	/// 11 ent 11	96 KB 00:00:	0.2			P: 984 D: 984	Local Contract	rens. 0									

Perform a Data Capture.



File Edit View Go Canture Analyze Statistics Help

Interfaces	Options	Capture		el stop	Restart	Dpen (Save As	X Close	Refresh	Print		Find	ł	a ck	Forward			Top Botto	n Colo	rize A	auto Scroll
Filter:							∨ +	Expression	n ≽ <u>C</u> lear	🕜 <u>А</u> рр	ly										
Time 🗸	Frame	Protocol	Dir	Error	Pkt/VL Nam	е	Src Port/	Chan C	ard/DestChan	DestP	ort/Labe	el CF	C	Pay	load/Info			VL Nur	n SrcIP	De	stiP
4.160183	969	UDP	A RX		0		5	0		5		20		Src	Port:	rje D	est Port	: rje 5	10.0.0).5 10	.1.0.5
4.160183	970	UDP	B RX				5			5							est Port				.1.0.5
4.160212	971	UDP	A RX				6			6				Src	Port:	6 Des	t Port:	6 6	10.0.0	0.6 10	.1.0.6
4.160213		UDP	B RX				6			6							t Port:	6 6	10.0.0	0.6 10	.1.0.6
4.212087		AFDX_1		NONE		c_sim_dat				1					load Le			1			.1.0.1
4.212088		AFDX_1		NONE	vl_valfa	c_sim_dat				1		52	93fe4		load Le		11 A	1			.1.0.1
4.212109 4.212110		UDP UDP	A RX B RX				2			2 2							t Port: t Port:				.1.0.2
4.212110		UDP	A RX				2			2							t Port: t Port:				.1.0.2
4.212132		UDP	BRX				3			3							t Port:				.1.0.3
4.212159		UDP	A RX				4			4							t Port:				.1.0.4
4.212160		UDP	B RX				4			4				Src	Port:	4 Des	t Port:	4 4			.1.0.4
¢									111												>
Frame	074 (12	7 hutes	on wi	ro 12	27 bytes c	antured)				•	0000	03 00	00 0	0 00	01 00 3	0 f7	02 4b 40	08 00 45 0	0	0	.K@E.
					:40, Dst:	- 10 - IO -		1										00 01 0a 0			
									0.1 (10.1.0		020	00 01	00 0	1 00	01 00 5	8 00	00 00 00	00 8d 00 8	d	X .	
															00 00 0			00 00 00 00			
				c Port	: tcpmux	(1), Dst .	ort: to	pmux (1)						00 00 0			00 00 00 00			
	11 ····	: tcpmu		(4)											00 c3 8			80 00 00 0 40 00 00 0			.C .DO@
		n port: 1	tcpmux	(1)														29 3f e4			s)?.
	gth: 88																				
) 0000x(1																			
		-Duplex	Switc	hed Et	hernet: P					19											
	erved_					=141,0x				0											
	1000 00 00 00 00 00 00 00 00 00 00 00 00	dinvld1	-			=141,0x	8d (0000	0000 1	000 1101)												
slow	vongndir	vld1				=FALSE	(
slow	vongnd1					=FALSE	(•											
swf]	Lautosla	tcmd				=FALSE	(
chyd	lsyspres	sell				=0.0000	00														
lhyd	lsyspres	sell				=0.0000	00														
rhyd	lsyspres	selr				=0.0000	00			_											
mach	1					=0.0000	00														
true	eairspee	d				=0.0000	00														
	•	4850	_				••														
File: cantur	6318 00	001 20061	012081	811 cnt	196 KB 00:00:	03			P: 984 D: 984	> M: 0 Dr	nns:0										

Starting the Analyzer. The default preferences of the interface will be set by double clicking on the desktop icon corresponding to the interface. There should be icons for all hardware interface devices on the system desktop. By starting execution of the analyzer in this way, the capture interface will be set, the packet viewing columns pertaining to the interface are displayed, and default interface options set.

There is a common sequence of events to performing data captures. Once the interface and options are set, the user only needs to depress the *Capture* button to restart a capture.

- Select/Verify correct interface on the general options (select menu item *Capture->Options* or depressing toolbar item *Options*) page. This is described in <u>#5.1General Options Notebook Page.</u> outline
- Setup Capture Options for Interface. This is described in <u>#5.2AFDX Options Page.outline</u> for AFDX, <u>#5.3P2P Options Page.outline</u> for P2P, <u>#5.4A429 Options Page.outline</u> for A429, and <u>#5.5Alldev</u> <u>Options Page.outline</u> for ALLDEV.
- Start Capture by selecting menu item *Capture->Start* or depressing toolbar item *Capture*.
- Let capture execute. View any real-time packet/payload information Described in <u>#5.6Real-Time</u> <u>Display Option Pages.|outline</u> and <u>#6.Active Data Capture.|outline</u>.
- Stop Capture by selection menu item *Capture->Stop* or depressing toolbar item *Stop*, or when a trigger condition has been met.
- View packet trace data. Possibly using a display filter <u>#7.Digital Filters.outline</u> to display only packets of interest to the user.

©The Goebel Company



4. Preparing Capture Options.

The analyzer contains various interfaces which have special capture options. Theses options are contained in the notebook pages (located at the top of the dialog) of each interface. Options pages are only displayed for interface hardware contained in the system.

The following sections will describe the available options for each interface. The user must configure the interface options before the capture is started. Incorrect or no capture data is usually the result of incorrect capture options or setup. But for standard analyzer usage, the options should be set by default at startup.

4.1 General Options Notebook Page.

The general options page contains options that are not specific to an interface and general to the capture process.

4.1.1 Button Box.

- <u>Capture</u>. Depress the capture button to start execution of a capture.
- <u>Cancel</u>. Depress the cancel button to close the option dialog notebook and restore value to before opening options dialog.
- <u>Help.</u> Depress the help button will display the help dialog. The help dialog tries to provide useful information to the user.

4.1.2 Capture Files Frame:

- <u>File Entry.</u> This entry field will allow the user to specify the name of the capture file. The default is /*tmp/capture_<process_id>.cpt*. The process id is used to make each filename unique.
- <u>Use Multiple Files.</u> This option will allow the capture output to be saved in multiple files with limited size. When a file has exceeded defined storage, the system will move to the next file in a circular queue.

When a single file is used for storage, the file can grow to a size larger than the available memory space of the computer. When this occurs, the system performance will produce poor response displaying capture data. With the default of 2 ring buffer file of 15MB, the analyzer can run for a longer capture intervals with no performance hits.

- <u>Next File every X megabytes</u>. This option will specify the amount of data stored in each file with the multiple files option enabled. This is the default with a value of 15M.
- <u>Next File every X minutes</u>. This option will specify the amount of time packet data is stored in each file with the multiple files option enabled.
- <u>Ring buffer with X Files.</u> X number of files contained in a file ring buffer (when the last file is completed, the analyzer will return the first file) with the multiple files option enabled. This option is the default with a value 2 files.
- <u>Stop capture after X files</u>. Stop capturing of data when X files of data have been stored. Used with the multiple files option enabled.

4.1.3 Display Options Frame.

• <u>Update List of Packets in Real Time.</u> This option will update the packet list pane in real time. Otherwise the packet list pane will be updated after the capture has been stopped. Using this option with a display filter at capture start, will only display packet matching the display filter in semi real time fashion.



- <u>Automatic Scrolling in live capture</u>. This option will scroll the packet list pane in real time. It is only available when the Update List of Packets in Real Time option is selected. Otherwise the packet list pane will display the 1st bundle of packets received.
- <u>Display Word:Bit Locations On</u>. This option will display the location data in the packet tree view of each payload element. The format of the data is StartWord:StartBit..EndWord:EndBit.

🕒 🥥 🕑 Ethereal:	Capture Option	s		🛋 🗶
General AFDX1 AFDX R	FD P2P1 P2P RTC	A429 A429 RTC	ALLE	DEV
Capture				
Interface: All Device	es interface: alld	ev		¥
				uit. Al
Limit each packet t	o 68 🐴	bytes		
Capture Filter:				
Trigger:				▼
Capture After Trigge				
 Capture around Trig Capture Before Trig 		packe	L(S)	
Capture File(s)	1			Display Options
File: /tmp/capture_25	63.cpt	Brow	se	Update list of packets in real time
Use <u>multiple</u> files				
Next file every	15	megabyte(s)	•	Automatic scrolling in live capture
Next file every	1	minute(s)	•	Display Word:Bit Locations On
Ring buffer with		files	-	Display Bit Locations On
Stop capture after	1	file(s)		Hide capture info dialog
		4		
Stop Capture				Name Resolution
after 1		acket(s)		Enable MAC name resolution
🔲 after 🛛 15		megabyte(s)	7	Enable network name resolution
🗌 after 🛛 🔟	<u>A</u>	minute(s)	±	Enable <u>transport</u> name resolution
Help				🗙 Cancel 🔯 Capture

- <u>Display Bit Locations On</u>. This option will display the location data in the packet tree view of payload elements. The format of the data is StartBit..EndBit.
- <u>Hide Capture Info Dialog.</u> This option will allow the user to hide the capture info dialog.



4.1.4 Stop Capture Frame.

The Stop Capture Frame contains options for automatic capture completion.

- <u>After X packets.</u> The user can stop the capture after a specific number of packets.
- <u>After X Megabytes.</u> The user can stop the capture after a specified amount of disk space has been stored.
- <u>After X minutes.</u> The user can stop the capture after a specified amount of time has past.

If no options are enabled, the capture will need to be stopped by completion of a trigger event or depress the toolbar item *Stop* or menu item *Capture->Stop*.

4.1.5 Capture Frame.

- <u>Interface Pulldown</u>. The interface pulldown will allow the user to select an interface. At this time, the available interfaces are *afdx*, *p2p*, *a429*, *can* and *alldev*. The interface will be displayed in the pulldown when hardware for an interface is contained in the system. The analyzer will allow for multiple interfaces of the same type with a board number appended to the end of the name. The *alldev* is a pseudo interface that will receive data from all physical interfaces.
- <u>Limit each packet to X bytes.</u> This option will limit the size of a received packet to X bytes.
- <u>Capture Filter Button</u>. The Capture Filter button will open a dialog box as shown below. The dialog box will allow the user to create or select a saved capture filter. The capture filters are saved in the *cfilter* file contained in the analyzer configuration directory. When a capture filter is selected, it will be displayed in the capture filter pulldown. The capture filter pulldown will contain the active analyzer capture filter. If the capture filter entry is an empty string, this will allow all packets through the filter.

• •	ithereal: Capture Filter 📄 🛋 🗙							
Edit	Filter							
	VL_ADRF_C_OUTPUT 3							
	VL_ADRF_L_OUTPUT 1							
New	VL_ADRF_R_OUTPUT 2							
	VL_IRU_L_OUTPUT 4							
	VL_IRU_R_OUTPUT 5							
	MTF_TO_CMCF 5062							
	CMCF_TO_MTF 5066							
	FCM_CENTER_RX 5469							
(Dalata	FCM_LEFT_RX 5069							
Delete	FCM_RIGHT_RX 5269							
	VL_VALFAC_SIM_DATA 5650							
Properties								
Filter name:	FCM_CENTER_RX 5469							
Filter string:	ether[5] = 93 and ether[4] = 21							
🙆 <u>H</u> elp	Gave X ⊆lose ⊗ QK							



A new capture is created by depressing the *New* button and entering a capture name and actual capture Berkeley Packet Filter (BPF) string. The *Delete* button is used to remove the selected capture filter. The capture filter data is saved to file with the *Save* button. The *OK* button will only keep the data for the length of this capture session.

- <u>Capture Filter Entry.</u> Capture Filters are used to filter out unnecessary packets at capture time. This is done to reduce the size of the resulting capture output file and is especially useful on high traffic networks or for long term capturing. The analyzer uses the Berkeley Packet Filter language to describe capture filters. This language syntax is explained in the tcpdump man page (<u>http://www.tcpdump.org</u>) and <u>http://ome/insight.rr.com/procana.</u> Note: This capture filter language is different from the one used for the analyzer display filters.
- <u>AFDX Capture Filter Syntax</u>: The following is a short description of the capture filter language syntax. A capture filter takes the form of a series of primitive expressions, connected by conjunctions (and/or) and optionally preceded by not:

[not] primitive [and|or [not] primitive ...]

A primitive is simply one of the following:

[src|dst] host <host>

This primitive allows you to filter on a host IP address. You can optionally precede the primitive with the keyword src|dst to specify that you are only interested in source or destination addresses. If these are not present, packets where the specified address appears as either the source or the destination address will be selected.

ether [src|dst] host <ehost>

This primitive allows you to filter on Ethernet host addresses. You can optionally include the keyword src| dst between the keywords ether and host to specify that you are only interested in source or destination addresses. If these are not present, packets where the specified address appears in either the source or destination address will be selected.

[tcp|udp] [src|dst] port <port>

This primitive allows you to filter on TCP and UDP port numbers. You can optionally precede this primitive with the keywords src|dst and tcp|udp which allow you to specify that you are only interested in source or destination ports and TCP or UDP packets respectively. The keywords tcp|udp must appear before src|dst. If these are not specified, packets will be selected for both the TCP and UDP protocols and when the specified address appears in either the source or destination port field.

ether|ip|udp|afdx [offset : 1|2|4]

This primitive will point to the specific area of the buffer. With the offset used to index into the ether, ip, udp, afdx area with a size 1 (byte), 2 (short word), or 4 (long word). Default size is 1.

less|greater <length>

This primitive allows you to filter on packets whose length was less than or equal to the specified length, or greater than or equal to the specified length, respectively.

<expr> relop <expr>

This primitive allows you to create complex filter expressions that select bytes or ranges of bytes in packets. Please see the tcpdump man pages for more details.



AFDX Examples:

ether[4:2]==12345 – capture all traffic on vl 12345.

host 192.168.0.10 -- capture all traffic to and from the IP address 192.168.0.10.

host dst 192.168.0.10 -- only destination address.

host src 192.168.0.10 -- only source address.

udp port 5280 -- capture all traffic to and from the UCP port 5280.

udp src port 5280 -- only with source port of 5280.

udp dst port 5280 -- only with destination port of 5280.

ip[2] == 0x10 -- will capture if 2nd byte in IP header is 0x10.

ip[10] == 0x10 -- will capture if 10th byte in IP header is 0x10.

Verify the value of data in the UDP payload use the **afdx** primitive.

afdx[0] == 0x10 -- will capture if 1st byte of payload is 0x10.

afdx[0:2] == 0x2000 -- will capture if 1st short word of payload is 0x2000.

afdx[0:4] == 0x20002000 -- will capture if 1st long word of payload is 0x20002000.

ether[4:2]==12345 && udp port 5280 – capture all traffic with vl of 12345 and udp port of 5280.

udp port 5280 and ip[2] == 0x10 and (afdx[0] == 0x10 or afdx[4:4] < 12) -- and or used to make more complex expressions.

afdx[0] == 0x10 and afdx[2:2] != 0x2000 and afdx[4:4] < 0x10. capture data when the (payload byte 0 is 0x10) AND (short word at payload byte 2 is 0x2000) AND (long word at payload byte 4 is less than 0x10).

<u>P2P Capture Filter Syntax</u>: The following is a short description of the p2p capture filter language syntax. A capture filter takes the form of a series of primitive expressions, connected by conjunctions (and/or) and optionally preceded by not:

[not] primitive [and|or [not] primitive ...]

The only primitive for the P2P interface is P2P.

p2p [offset : 1|2|4]

This is the only primitive. It will point to the beginning of the buffer. With the offset used to index into the buffer with a size 1 (byte), 2 (short word), or 4 (long word). Size will default to 1 if not written.

p2p[8] == 0x10 and p2p[10:2] != 0x2000 and p2p[12:4] < 0x10

This example will capture when p2p byte 8 is 0x10 and short word at byte 10 is

0x2000 AND long work at byte 12 is less than 0x10.

P2P Examples:

p2p[0:2] == 5280 -- capture all traffic to and from the port 5280.

p2p[2:2] == 8 -- capture all traffic with a length of 8

GOEBEL

Goebelyzer

Verify the value of data in the P2P payload, payload data starts at byte 4. The message payload is located at P2P[4]. The P2P header contains 4 bytes. NOTE: a payload data item located at byte 4 would use a P2P[8] location in the filter.

p2p[8] == 0x10 -- will capture if 1st byte of payload is 0x10

p2p[8:2] == 0x2000 -- will capture if 1st short word of payload is 0x2000

p2p[8:4] == 0x20002000 -- will capture if 1st long word of payload is 0x2000

p2p[0:2] == 5280 and p2p[2] == 0x10 and (p2p[8] == 0x10 or p2p[12:4] < 12) -- and or used to make more complex expressions

A429 Capture Filter Syntax: The A429 capture filter syntax is same as the p2p with keyword p2p changed to a429.

A429 Examples:

a429[3:1] == **0123** -- capture all traffic with octal label 123.

a429[3]==0333 && ((a429[0:4]&0x3000000) == 0x1) – capture all traffic with label of octal 333 and ssm of 1.

a429[3]==0333 && ((a429[0:4]&0x300) == 0x1) – capture all traffic with label of octal 333 and sdi of 1.

a429[3] == 0333 && ((a429[0:4]&0x400) == 0x1) - capture all traffic with label of octal 333 and bit 11 set to 1.

<u>Capture Trigger Button.</u> The Capture Trigger button will open a dialog box as shown below. The dialog box will allow the user to create or select saved capture triggers. The capture triggers are saved in the *trigger* file contained in the analyzer configuration directory. When a capture trigger is selected it will be displayed in the capture trigger pulldown. The pulldown will contain the active analyzer capture trigger. The operations of the trigger dialog are similar to the capture filter.<u>#5.1.2Capture Files Frame:|outline</u>



Edit	Trigger
	PAUL_ADIRU
	PAUL_FC
Mew New	PAUL_FMS1
	PAUL_FMS2
	PAUL
Properties	
	PAUL_FMS2
	udp src port 8800
Trigger string:	udp src port 8800
Trigger string: Capture Af	La neuro contra de la contra de
Trigger string: Capture Af Capture ar	iter Trigger

- <u>Capture Trigger Entry.</u> Trigger on specific packet to start/stop the capture. Triggers are used to capture packets with respect to a trigger event (after trigger packet, around trigger packet, before trigger packet. The analyzer uses the BPF language for triggers. This language is explained in the tcpdump man page <u>http://www.tcpdump.org</u> or <u>http://home.insight.rr.com/procana</u>. Note: This trigger language is different from the one used for the analyzer display filters. But the same as capture filters described in previous sections.
- Trigger Event Type.

AROUND , AROUND NUMBER: selection of the AROUND trigger type, will capture packets waiting for the trigger event to occur. When the trigger event has occurred, the trigger packet is saved and the "AROUND NUMBER" of capture packets are saved. The capture will be stopped when the "AROUND NUMBER" packet is received.

AFTER: selection of the AFTER trigger type, will wait for the AFTER trigger event. When the event is triggered, all capture packets are saved including the trigger packet event. The capture will continue running and the operator must stop the capture.

BEFORE: selection of the BEFORE trigger type, will capture packets waiting for the BEFORE trigger event. When the BEFORE trigger event is decoded, this capture packet is saved and the capture is stopped.

• <u>Trigger Syntax</u>. The trigger language syntax is the same as the capture filter syntax described above except that a event type must also be selected.



4.2 AFDX Options Page.

This page contains options that are specific to the afdx interface. The notebook pages are named AFDX1..AFDXN where N is the number of afdx interface board in the system.

General AFDX1 A429 A429 RTD FDO RTD IDO RTD ALLDEV	
Network Selection	
V Network A	
☑ Network B	
Filter Redundant	
PassThru Mode	
Latency Til Enable Pass Thru of Data on AFDX Network (data is	
read and retransmitted on the network)	
Passthru Modify Filter Parameters	
Passthru:	
Wait Seconds: 0 🔺 Modify Frames: 0 🐥 Modify Seconds: 0 🖨	
Speed Mode	
○ 10 100	
Capture TX Packets	
Network A	
□ Network B	
Timestamp Resolution	
Microsecond O Nanosecond	
Save AFDX Config X Cancel	

4.2.1 Button Box.

- <u>Capture</u>. Depress the capture button to start execution of a capture.
- <u>Cancel.</u> Depress the cancel button to close the option dialog notebook and restore default values.

©The Goebel Company





• <u>Save AFDX Config.</u> The Save AFDX Conf button will save the state of channel and device configuration data for the duration of the program execution. After the AFDX configuration is saved, these will become the default AFDX settings when the dialog is opened in the future. Permanent options can be save to the preferences file using the *Edit->Preferences->AFDX HW* option. The file is located in the analyzer configuration directory.

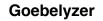
4.2.2 Network Selection Frame.

- <u>Network A.</u> This option will select Network A data to be received by the analyzer. This option is set by default.
- <u>Network B.</u> This option will select Network B data to be received by the analyzer. This option is set by default.
- <u>Filter Redundant</u>. This option will keep only the first physically received packet from the A and B networks. The later packet will be thrown away. When filter redundant is selected, the Network A and B buttons will be forced to the off state. Also, Filter Redundant operation is exclusive (only one of the 3 button can be set) of Passthru and Latency Timing.
- <u>Pass Thru Mode</u>. This option will enable the operation of the passthru mode. An analyzer will be connected inline with an LRU. Packets received on the A network port will be output on the B network port. The opposite will also true. This will allow for a sniffer type operation to verify all data going in and out of an LRU. The passthru filter is described below will allow for a packet data item to be changed during the passthru of data from the A to B networks ports.
- <u>Latency Timing.</u> TBD.
- <u>Time Manager.</u> TBD.

4.2.3 Passthru Modify Filter Parameters Frame.

• <u>Passthru Button</u>. The Passthru Button will open a dialog box shown as below. The dialog box will allow the user to create or select saved passthru filters. The passthru filters are saved in the *passthru* file contain in the analyzer configuration directory. When a passthru is selected it will be displayed in the passthru pulldown. The pulldown will contain the active analyzer passthru filter. The operations of the passthru dialog are similar to the capture filter dialog

	Ethereal: Capture F	≜ ×
Edit	Passthru	
	PAUL1	
	PAUL2	
<u>N</u> ew		
O Delet	e	
Properties Passthru r	ame	
Passthru s		



• <u>Pass thru Entry</u>. Passthru Filters are used to filter on specific packets and update data elements in the packet for retransmission. This could be useful to create errors in transmission of data on the physical interface. The syntax of the passthru expression is:

<bpf expr> modify <passthru assignment>

The passthru expression

bpf expr> uses the same primitives as described in capture filter/trigger section

<u>#4.1.5Capture Frame.|outline</u>. So if the

bpf expr> contains a TRUE value, the passthru assignment

cpassthru assignment> will be executed. If FALSE, no assignment is performed. This will allow the

user to be very specific when selecting a packet for passthru assignment.

The format of the passthru assignment is:

ether | ip | udp | afdx [pkt offset : 1|2|4] := expr | value

Passthru Syntax Examples:

ether[4:2]==1500 modify afdx[0:4] := afdx[0:4] | 0x7 – If vl is 1500, set first long word of payload to afdx[0:4] | 7.

udp port 5280 modify afdx[10:2] := 10 – If UDP port is 5280, set value of 10 at short word offset 10 byte from afdx payload.

udp src port 5280 modify udp[20:4] := -0x12345678 passthru udp[10:4] := 0x12345678 – If UDP source port is 5280, set value of 0x12345678 at long word offset 20 bytes from UDP header and set value of 0x12345678 at long word offset 10 bytes from UDP header.

udp port 5280 and udp[12] == 0x10 and udp[16:4] < 12 modify udp[44:2] := 10 – If UDP port is 5280 and data at UDP[12] is 0x10 and long word at UDP[16 is less than 12, set short word at UDP[44] to value of 10.

Note: use afdx[] to access payload data.

- <u>Wait Seconds</u>. This will delay the execution of the passthru modify by the number of seconds specified.
- <u>Modify Frames</u>. This spin button will specify the number of frames the modify passthru filter will set the requested data.
- <u>Modify Seconds</u>. This spin button will specify the number of seconds the modify passthru filter will set the requested data.

If both the modify frames and modify seconds parameters are specified, the filter will stop when the first condition is meet. Also, setting the modify values to zero, will have the modify filter execute for the length of the capture.

4.2.4 Speed Mode Frame.

A selection of radio buttons allows the user to pick the speed of the AFDX network. The possible values are 10MBits/sec, 1000MBits/sec, 1GBits/sec, or AUTO. The analyzer will negotiate with other end items to select a speed when AUTO is selected.

4.2.5 Capture TX Packets Frame.

- <u>Network A.</u> This option will allow any Network A data transmitted by the afdx card to be looped back through the analyzer.
- <u>Network B.</u> This option will allow any Network B data transmitted by the afdx card to be looped back through the analyzer.





4.3 P2P Options Page.

The p2p options page contains options that are specific to the p2p interface. The notebook pages are named P2P1..P2PN where N is the number of the p2p interface board in the system. Each p2p interface board has 32RX/TX channels.



4.3.1 Channel Initialization (Enable/Speed) Frame.

	🕑 Ethe	real: Ca	pture Op	tions)				
General	AFDX1 A	FDX RTD	P2P1 P2P	RTD A429 A4	29 RTD ALLE	DEV			
			le/Speed)	~		-	_	- ·	
Chan 1		🖌 ТХ	⊖ Low	High	Chan 17	N RX	🗆 тх	O Low	O High
Chan 2	RX 📝	🗆 ТХ	⊖ Low	High	Chan 18	RX	🔽 ТХ	Low	O High
Chan 3		TX 💟	⊖ Low	High	Chan 19	RX 📝	🗌 ТХ	Okonomic Low	O High
Chan 4	RX RX	🗌 ТХ	\bigcirc Low	High	Chan 20	RX	🟹 ТХ	Low	🔘 High
Chan 5	📝 RX	🗆 ТХ	Low	O High	Chan 21	RX	🗆 тх	O Low	🔘 High
Chan 6		📝 ТХ	O Low	⊖ High	Chan 22	RX 📝	📝 ТХ	O Low	🔘 High
Chan 7	📝 RX	🗆 тх	Low	🔿 High	Chan 23	RX RX	🟹 ТХ	O Low	🔿 High
Chan 8		📝 ТХ	O Low	🔿 High	Chan 24	RX RX	🟹 ТХ	Low	🔿 High
Chan 9		📝 ТХ	O Low	🔿 High	Chan 25	RX	🗆 тх	O Low	🔿 High
Chan 10	📝 RX	💟 ТХ	Low	🔘 High	Chan 26	RX	📝 ТХ	O Low	🔿 High
Chan 11		📝 ТХ	Low	🔘 High	Chan 27	📝 RX	🗆 тх	O Low	🔿 High
Chan 12	📝 RX	📝 ТХ	Low	🔿 High	Chan 28	RX	📝 ТХ	O Low	🔿 High
Chan 13		📝 ТХ	O Low	🔿 High	Chan 29	RX	🗆 тх	O Low	🔿 High
Chan 14	📝 RX	📝 ТХ	O Low	🔘 High	Chan 30	RX	📝 ТХ	O Low	🔿 High
Chan 15		📝 ТХ	O Low	🔿 High	Chan 31	📝 RX	🗌 тх	O Low	🔿 High
Chan 16	📝 RX	📝 ТХ	O Low	🔿 High	Chan 32	🗌 RX	📝 ТХ	O Low	🔿 High
	sa	ive P2P Co	onf	2 2	ancel		Ð	<u>Capture</u>	

This frame allows the user to configure the speed and which transmit type data is input by the analyzer. There is a group of buttons for each of the 32 channels on the P2P H/W.

- <u>RX.</u> This option will allow RX data received from the hardware on the specified channel to be input to the analyzer.
- <u>TX.</u> This option will allow TX data transmitted by the hardware to be looped back to the analyzer on the specified channel.
- Low/High Speed. This option will select the speed (LO or HI) of the each P2P hardware channel.

4.3.2 Button Box.

- <u>Capture</u>. Depress the capture button to start execution of a capture.
- <u>Cancel.</u> Depress the cancel button to close the option dialog notebook and restore default values.
- <u>Save P2P Conf.</u> The Save P2P Config button will save the state of channel and device configuration data for the duration of the program execution. After the P2P configuration is saved, these will become the default P2P settings when the dialog is opened in the future. Permanent options can be



save to the preferences file using the *Edit->Preferences->P2P HW* option. The file is located in the analyzer configuration directory.

4.4 A429 Options Page.

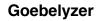
This page contains options that are specific to the a429 interface. There is only one A429 notebook page. Each A429 interface board has capabilities to transmit and receive on at the most 16RX/8TX channels.

🖲 🔇 🥝 Arinc 429 Ca	pture Options	Ĵ	-				≜ ×
General AFDX1 A429 A	429 RTD ALLDEV						
RX Channel Init (Enable/C	hannel/Speed/Bus/Equ	uipment	ID)				
Ch01 🗌 En 🔿 Lo 🧿 Hi	adc_1	•	EQID:	NONE	•	Proto:	A429 💌
Ch02 🗌 En 🔿 Lo 💿 Hi	rs_1	•	EQID:	NONE	•	Proto:	A429 💌
Ch03 🗹 En 💿 Lo 🔾 Hi	[fmc_1_general_1_rx	•	EQID:	NONE	•	Proto:	A429 💌
Ch04 🗌 En 💿 Lo 🔾 Hi	NONE	•	EQID:	NONE	•	Proto:	A429 💌
Ch05 🗌 En 🔿 Lo 🧿 Hi	NONE	•	EQID:	NONE	•	Proto:	A429 💌
Ch06 🗌 En 🔾 Lo 🖲 Hi	NONE	•	EQID:	NONE	•	Proto:	A429 💌
Ch07 🗌 En 🔿 Lo 🧿 Hi	NONE	•	EQID:	NONE	•	Proto:	A429 💌
Ch08 🗌 En 🔿 Lo 💿 Hi	NONE	•	EQID:	NONE	•	Proto:	A429 💌
Ch09 🗌 En 🔿 Lo 💿 Hi	NONE	•	EQID:	NONE	•	Proto:	A429 💌
Ch10 🗌 En 🔿 Lo 💿 Hi	NONE	•	EQID:	NONE	•	Proto:	A429 💌
Ch11 🗌 En 🔿 Lo 💿 Hi	NONE	•	EQID:	NONE	•	Proto:	A429 💌
Ch12 🗌 En 🔿 Lo 🧿 Hi	NONE	•	EQID:	NONE	•	Proto:	A429 💌
Ch13 🗌 En 🔿 Lo 🧿 Hi	NONE	•	EQID:	NONE	•	Proto:	A429 💌
Ch14 🗌 En 🔿 Lo 💿 Hi	NONE	•	EQID:	NONE	-	Proto:	A429 💌
Ch15 🗌 En 🔿 Lo 💿 Hi	NONE	•	EQID:	NONE	•	Proto:	A429 💌
Ch16 🗌 En 🔿 Lo 💿 Hi	NONE	•	EQID:	NONE	•	Proto:	A429 💌
Save A42	9 Conf	×c	ancel		Capt	ure	

4.4.1 RX Channel Init Frame.

This frame allows the user to configure each A429 RX channel. There is a group of buttons for each of the RX channels on the A429 H/W.

• <u>Enable</u>. This option will enable/disable the input of RX data from the corresponding channel.





- <u>RX Speed.</u> This option will set the speed of the RX channels on the device. The speed can be set to LO (10K) or HI (100K). The speed select must match the speed of the TX channel connected to this RX channel. Due to hardware limitions, the speed is selected in pairs (1,2),(3,4).
- <u>Bus Name Pulldown</u>. A429 hardware can be connected between any two end items that may transmit the same labels containing different data items. The analyzer needs a way to decide which data definition to attach the label with. So, when the user selects a bus from the pulldown list, the label will be decoded based on packets for the bus name. The packet name definitions in the *packet-a429.dat* file will contain the bus name with a _w<label> appended to the end of the string. All a429 bus names are defined and stored in the *packet-a429_busses.dat* file. The NONE value is used as a default to indicate that no value have been selected or when bus name definitions do not exist.

🗩 🤇 🕜 Ethereal: Capt	ure Options <@Goebelyzer	ITV:	2.goebel	Ltest.net>				≜ ×
General AFDX1 A429 A4	29 RTD CAN1 CAN2 ALL	DE	V					
RX Channel Init (Enable/C	nannel/Speed/Bus/Equipment	ID)						
Ch1 🗹 En 🔿 Lo 💿 Hi	adc_1	-	EQID:	NONE	-	Proto:	A429	-
Ch2 🗹 En 🔿 Lo 💿 Hi	NONE		EQID:	NONE	-	Proto:	A429	-
Ch3 🗹 En 🔿 Lo 💿 Hi	adc_1		EQID:	NONE	-	Proto:	A739	-
Ch4 🔽 En 🔿 Lo 💿 Hi	adc_2		EQID:	thrust_control_computer		Proto:	A429	
Ch5 🗌 En 🔿 Lo 💿 Hi	clock		EQID:	NONE	-	Proto:	A429	-
Ch6 🗌 En 🔿 Lo 💿 Hi	cmc		EQID:	NONE	-	Proto:	A429	T
Ch7 En O Lo O Hi	cockpit_mcp cockpit		EQID:	NONE	-	Proto:	A429	F
Ch8 En O Lo Hi	dcu_efiscp_1		EQID:	NONE	-	Proto:	A429	T
Ch9 En O Lo Hi	and the second se		EQID:	NONE	-	Proto:		
Ch10 En O Lo Hi			EQID:	1		Proto:		
Ch11 Ch11 Ch11 Ch11 Ch11 Ch11 Ch11 Ch11	dcu_eicas_2		EQID:			Proto:		
and the second	dme_1		Constantine .					
Ch12 En O Lo Hi	dme_2		EQID:		-	Proto:		-
Ch13 🗌 En 🔿 Lo 💿 Hi			EQID:		-	1	A429	-
Ch14 🗌 En 🔿 Lo 💿 Hi			EQID:		*		A429	-
Ch15 🗌 En 🔿 Lo 💿 Hi	and the second sec		EQID:	NONE	-	Proto:	1.1.2.5	-
Ch16 🗌 En 🔿 Lo 💿 Hi			EQID:	NONE	-	Proto:	A429	-
	eiu_1							
	eiu_2							
	eiu_3 eiul_1							
	eiul_2							
	eiul_3							
	fcc_1_mcp							
	fcc_2_mcp							
	fcc_3_mcp							
	fmc_1_acars_rx							
	fmc_1_at_data_rx							
	fmc_1_bus_10_rx							
	fmc_1_bus_1_rx							
	fmc_1_bus_2_rx							
	fmc_1_bus_4_rx							
	fmc_1_bus_5_rx							
	fmc_1_bus_6_rx							
	fmc_1_bus_7_rx		-		1 -		7	
Save A4	fmc_1_bus_8_rx		Cancel		⊆ap	ture		
	fmc_1_bus_9_rx							

• <u>Equipment ID Pulldown</u>. The document ARINC 429 SPEC 429P1-15 defines a set of equipment ids (various avionics end items) and labels transmitted by the end items. These definitions are contained in the data definition file *a429-eqid.dat*. So if the user knows that data transmitted on a channel is from an equipment id in the pulldown list, the channel can be decoded with the standard label definitions for the selected id. This will also be helpful if no a429 data/bus definitions exist. If both a bus name and equipment id are selected for an RX channel, the system will try to decode with the bus name first and the equipment id second.



🕤 🥥 🖉 Ethereal: Capture	e Options <@Goebelyzer	ITV2.go	ebel.test.net>	a x
General AFDX1 A429 A429	RTD CAN1 CAN2 AL	LDEV		
RX Channel Init (Enable/Chan		t ID)		
Ch1 🗹 En 🔿 Lo 💿 Hi 🔤	dc_1 [- EQI	D: NONE	Proto: A429
Ch2 🗹 En 🔿 Lo 💿 Hi 🖂	lock	👻 EQI	D: NONE	📤 Proto: 🗛29 💌
Ch3 🗹 En 🔿 Lo 💿 Hi N	ONE	- EQI		Proto: A739
Ch4 🗹 En 🔿 Lo 💿 Hi N	ONE	- EQI		Proto: A429 -
Ch5 🗌 En 🔿 Lo 💿 Hi N	ONE	- EQI	D: thrust_control_computer	Proto: A429 -
Ch6 🗌 En 🔿 Lo 💿 Hi N	ONE	- EQ	D: attitude_and_heading_ref_sy	Proto: A429 -
Ch7 🗌 En 🔿 Lo 💿 Hi N	ONE	- EQI		Proto: A429 -
Ch8 🗌 En 🔿 Lo 💿 Hi N	ONE	- EQI		Proto: A429 -
Ch9 🗌 En 🔿 Lo 💿 Hi N	222.2853.29	- EQI		Proto: A429 💌
Ch10 🗆 En 🔿 Lo 💿 Hi 🖪	220.2%322	EQ	ID: FAC_A310_	Proto: A429 💌
Ch11 🗆 En 🔿 Lo 💿 Hi 🖪	IONE	EQ	ID: Global_Positioning_System	Proto: A429 💌
Ch12 🗆 En 🔿 Lo 💿 Hi 🖪	IONE	EQ	ID: Airborne_ILS_Receiver	Proto: A429 -
Ch13 🗆 En 🔿 Lo 💿 Hi 🖪	IONE	EQ	Airborne_VOR_Receiver	Proto: A429 -
Ch14 🗆 En 🔿 Lo 💿 Hi 🖪	IONE	EQ	ID: Digital_Slat_Flap_Computer.	Proto: A429 -
Ch15 🗌 En 🔿 Lo 💿 Hi 🖪	IONE		ID: Engine_Parameter_Digitizer	Proto: A429 -
Ch16 🗆 En 🔿 Lo 💿 Hi 🖪	IONE	<u> </u>	ID: Performance_Data_Compute	Proto: A429 -
			DFS_System	
			Electronic_FltInst	
			ADDSC_EICAS	
			Perf_Nav_Computer_Syster	
			Digital_Fuel_Gauging_Syste EPR_IndicatorB737_	
			Full_Authority_EEC_A	
			Propulsion_Multiplexer	
			TCAS	
			Weight_and_Balance_Syste	
			ADIRS	
			Autopilot_Buffer_Unit	
			Tire_Pressure_Monitoringy	
			Center_of_Gravity_Control_(
			Full_Authority_ECC_B Fuel_Quan_indicating_Syste	
			Fuel_Quan_Indicating_Syste	
1			Fuel_Flow_IndicatorB747_	
	Conf	X Cano		ture
			FCC_Controller	

• <u>Protocol Pulldown</u>. The analyzer supports decoding A739, A429 (default), and EFIS protocols on the A429 h/w. The user can select a protocol using the pulldown menu and save the configuration.

4.4.2 Button Box.

- <u>Capture</u>. Depress the capture button to start execution of a capture.
- <u>Cancel.</u> Depress the cancel button to close the option dialog notebook and restore default values.
- <u>Save A429 Conf.</u> The Save A429 Config button will save the state of channel and device configuration data for the duration of the program execution. Permanent options can be save to the preferences file using the *Edit->Preferences->A429 HW* option. The file is located in the analyzer configuration directory. After the A429 configuration is saved, these will become the default A429 settings when the dialog is opened in the future.



Control Capture Options < @ Goebelyzer/TV2.goebel.test.net>	
General AFDX1 A429 CAN1 CAN2 CAN RTD ALLDEV	
SJA1000 Parameters Device	
Device Enable	
Baud Rate: O 50K 💿 100K O 125K O 250K O 500K O 1M O OTHER	
Other Baud Rate: 0	
Acceptance Filter: 0x0	
Acceptance Filter Mask: 0xffffffff	
Sync Jump Width: 3	
Sample Point Percent (%): 75	
TX Loopback	
Save CAN Conf 🔀 Cancel	

4.5 CAN Bus Options Page.

This page contains options that are specific to the CAN bus interface. The notebook page contains options for 2 Philips SJA1000 devices which are contain on the hardware device. The user may refer to the SJA1000 Data Sheet for more information on parameter option described on this page.

4.5.1 SJA1000 Parameters Device Frame

- <u>Baud Rate.</u> The user can change radio buttons to select one of the commonly used baud rates. The OTHER radio button is used to select a different baud rate with the entry widget below.
- <u>Other Baud Rate</u>. This entry widget is used to enter a numeric value from 0 to 1000000. Values greater than 1000000 will be changed to 1000000. Invalid values such as a string entry will result in a 0 baud rate value. This entry is only allowed when the *Other* radio button is selected.
- <u>Acceptance Filter</u>. This entry will select a hexadecimal value of the SJA1000 acceptance filter. Invalid entries will be ignored and a value of 0x0 used. The acceptance filter will define which identifiers are allowed to be received by the SJA1000. The format of the filter is bit32 ->ID.28...bit 3-©*The Goebel Company*



>ID.3, bit 2->RTR. When the ID and filter have matching values with a range defined by the acceptance mask. See SJA1000 Data Sheet Figure 10 for more info.

- <u>Acceptance Filter Mask.</u> This entry will select a hexadecimal value of the SJA1000 acceptance filter mask. Invalid entries will be ignored and a value of 0xfffffff used. The mask is used to define which bits of the received ID are matched with the acceptance filter. See SJA1000 Data Sheet Figure 10 for more info.
- <u>Sync Jump Width</u>. This spin button will select a value from 0-3 for the Sync Jump Width. To compensate for phase shifts between clock oscillators of different bus controllers, a bus controller must re-sync on any revelant signal edge. The sync jump width defines the maximum number of clock cycles a bit period may be shortened or lengthened by one re-sync.
- <u>Sample Point Percent</u>. This spin button will select a value from 0-100 for the sample point. The sample point defines the location of the sample point within a bit period. A value of 0 is at the beginning of the bit period, 50 in the middle, and 100 at the end. See SJA1000 Data Sheet Figure 13 for more info.
- <u>TX Loopback</u>. This button will enable the analyzer to receive TX transmissions send from this machine.

4.5.2 Button Box.

- <u>Capture</u>. Depress the capture button to start execution of a capture.
- <u>Cancel.</u> Depress the cancel button to close the option dialog notebook and restore default values.
- <u>Save CAN Conf.</u> The Save CAN Config button will save the state of device configuration data for the duration of the program execution. Permanent options can be saved to the preferences file using the *Edit->Preferences->CAN HW* option. The file is located in the analyzer configuration directory. After the CAN configuration is saved, these will become the default CAN settings when the dialog is opened in the future.

M1553 Options Page.



Co Control Con	
General AFDX1 M1553 M1553 RTD ALLDEV	
Enable Devices:	
CHAN1	
CHAN2	
CHAN3	
CHAN4	
Save M1553 Config 🔀 Cancel	

M1553 Parameters Frame.

• <u>Channel.</u> The user can change radio buttons to select any channel to have the analyzer listen on.

4.5.3 Button Box.

- <u>Capture</u>. Depress the capture button to start execution of a capture.
- <u>Cancel.</u> Depress the cancel button to close the option dialog notebook and restore default values.
- <u>Save M1553 Conf.</u> The Save M1553 Config button will save the state of device configuration data for the duration of the program execution. Permanent options can be saved to the preferences file using the *Edit->Preferences->M1553 HW* option. The file is located in the analyzer configuration directory. After the M1553 configuration is saved, these will become the default M1553 settings when the dialog is opened in the future.



Capture Options
General AFDX1 SPWIRE FSCC ALLDEV
FSCC CHANNEL 1
Interface:
Mode: SYNC O ASYNC
☑ Device Enable
Frequency: 15000000
Clock PPM: 3
Baud Rate Divisor: 1
Address Size: 2
UART Stop Bits: I BIT 1.5 BITS 2 BITS
UART Parity: O NONE O ODD O EVEN
FSCC CHANNEL 2
Interface:
Mode:
Device Enable
Frequency: 14000000
Clock PPM: 4
Baud Rate Divisor: 1
Address Size: 0
UART Stop Bits: I BIT 1.5 BITS 2 BITS
Save Conf

4.6 FSCC Options Page.

This page contains options that are specific to the FSCC bus interface. The notebook page contains options for the two serial channels.

4.6.1 FSCC Parameters Channel Frame

- Interface. The user can change set the electrical interface to RS-422 or RS-485.
- <u>Mode</u>. The user can set the mode to synchronous or asynchrous.
- Enable. This will enable the analyzer on the selected channel.
- <u>Frequency</u>. The Board Frequency in the range of 100K to 30M.

GOEBEL

- <u>Clock PPM.</u> Parts Per Million of the system crystal.
- <u>Baud Rate Divisor.</u> Divider of the Board frequency to create channel frequency.
- <u>Address Size</u>. The address size of received packets.
- <u>UART Stop Bits.</u> Number of stop bits in the asynchrous data transmission.
- <u>UART Parity</u>. Parity type in an asynchrous data transmission.

4.6.2 Button Box.

- <u>Capture</u>. Depress the capture button to start execution of a capture.
- <u>Cancel.</u> Depress the cancel button to close the option dialog notebook and restore default values.
- <u>Save Conf.</u> The Save Config button will save the state of device configuration data for the duration of the program execution. Permanent options can be saved to the preferences file using the *Edit-* >*Preferences-*>*FSCC HW* option. The file is located in the analyzer configuration directory. After the FSCC configuration is saved, these will become the default FSCC settings when the dialog is opened in the future.

Spacewire Capture Options
General AFDX1 SPWIRE FSCC ALLDEV
SPWIRE CHANNEL 1
☑ Device Enable
Frequency: 80
I/O Divisor: 1
Init Divisor: 8
Address Size: 2
SPWIRE CHANNEL 2
Frequency: 80
I/O Divisor: 1
Init Divisor: 8
Address Size: 0
SPWIRE CHANNEL 3
Device Enable
Frequency: 80
I/O Divisor: 1
Init Divisor: 8
Address Size: 0
SPWIRE CHANNEL 4
Device Enable
Frequency: 80
I/O Divisor: 1
Init Divisor: 8
Address Size: 0
Save Conf

4.7 Spacewire Options Page.

This page contains options that are specific to the Spacewire bus interface. The notebook page contains options for the two.

4.7.1 Spacewire Parameters Channel Frame

- <u>Enable</u>. This will enable the analyzer on the selected channel.
- <u>Frequency</u>. The Board Frequency in the range of 100K to 30M.
- <u>I/O Divisor.</u> Divider of the Board frequency to create channel frequency.
- Init Divisor. Divider of the Board frequency to creat initialization frequency of 10M.
- <u>Address Size</u>. The address size of received packets.

4.7.2 Button Box.

GOEBEL

- <u>Capture</u>. Depress the capture button to start execution of a capture.
- <u>Cancel.</u> Depress the cancel button to close the option dialog notebook and restore default values.
- <u>Save Conf.</u> The Save Config button will save the state of device configuration data for the duration of the program execution. Permanent options can be saved to the preferences file using the *Edit-* >*Preferences->Spacewire HW* option. The file is located in the analyzer configuration directory. After the Spacewire configuration is saved, these will become the default spacewire settings when the dialog is opened in the future.



C Arinc 429 Capture Options	×
General AFDX1 A429 A429 RTD ALLDEV	
Capture Filters/Triggers	
afdx1 Capture:	
afdx1 Trigger:	
A429 Capture:	
A429 Trigger:	
Capture After Trigger	
O Capture around Trigger	
O Capture Before Trigger	
	_
X Cancel	

4.8 AllDev Options Page.

This page contains options that Alldev Options Page specific to the alldev interface. The alldev is a pseudo interface which contains all hardware contained in system. So, if the system contains one afdx and one a429 board. When the alldev device is selected, time sorted packet data will be receive from both the afdx and a429 interfaces.

©The Goebel Company



4.8.1 Select Devices Frame.

This frame contains a list of all hardware devices in the system. The user may remove hardware device from the alldev device by selecting a device to be removed from this list. Multiple devices selections may be made by depressing <CTRL> and left mouse buttons. The alldev device by default will contains all hardware devices in the system.

4.8.2 Capture Filters/Triggers Frame

This frame contains a trigger button and pulldowns for each interface in the system. The user may enter a different trigger for each device. When the first trigger event is tripped, data will be capture on all devices and all other triggers will be disabled. Only one type of trigger event can be defined for the alldev device.

This frame contains a capture button and pulldown for each interface in the system. The user may enter a different capture filter for each device. When the capture has been started and an alldev trigger event has been tripped, each alldev capture filter will execute only for the device that it corresponds with.

Capture Filter/Trigger syntax is described in detail in section <u>#4.1.5Capture Frame.|outline</u>

4.8.3 Button Box.

- <u>Capture</u>. Depress the capture button to start execution of a capture.
- <u>Cancel.</u> Depress the cancel button to close the option dialog notebook and restore default values.



🕞 🦳 🥝 Ethereal: Capture Options			a X
General AFDX1 AFDX RTD P2P1 P2P RTD A429 A42	29 RTD ALLDEV		
Select AFDX RealTimeDisplay Data			
Name			^
▶ fcm_center_8500			
▶ fcm_left_8020			
▶ fcm_right_8260			
✓ vl_valfac_sim_data_1			
reserved_			
dw_slowongndinvld1_			1
slowongndinvld1			
slowongndl			
swflautoslatcmd			
chydsyspressell			
lhydsyspressell			
rhydsyspresselr			
mach			
trueairspeed			•
Selected AFDX Data Entries:			
Name	Туре	ByteOffset	Length
vl valfac sim data 1.reserved	FT UINT32	0	4
vl_valfac_sim_data_1.dw_slowongndinvld1_	FT_UINT16	4	2
vl_valfac_sim_data_1.slowongndinvld1	FT_BOOLEAN	4	1
vl_valfac_sim_data_1.slowongnd1	FT_BOOLEAN	4	1
vl_valfac_sim_data_1.swf1autoslatcmd	FT_BOOLEAN	4	1
vl_valfac_sim_data_1.chydsyspressell vl_valfac_sim_data_1.lhydsyspressell	FT_FLOAT FT_FLOAT	8 12	4
vl valfac sim data 1.rhydsyspressel	FT FLOAT	16	4
vi valfac sim data 1.mach	FT FLOAT	20	4
vl valfac sim data 1.trueairspeed	FT_FLOAT	24	4
vl_valfac_sim_data_1.impactpressure	FT_FLOAT	28	4
vl_valfac_sim_data_1.baroaltitudeers	FT_FLOAT	32	4
vl_valfac_sim_data_1.altituderateers	FT_FLOAT	36	4
		Que Class	All Selected
Read RTD File	RTD File	& clear	All Selected
AFDX RT DISPLAY FIL	ENAME: UNDEFIN	ED	
🔀 <u>C</u> ancel	0	(<u>C</u> apture	

4.9 Real-Time Display Options Page.

This page allow users to select various payload data definitions for real-time display after the capture has been started. Each interface that supports the real-time display will have a notebook tab displayed. The real-time display page is described in <u>#5.3Data Display Page.outline</u>

4.9.1 Select Realtime Display Data Frame.

This frame contains a list organized as a tree of packets defined for the interface. This data is imported from the *packet-<interface>.dat* configuration file. If the packet entry is expanded, all data elements of

©The Goebel Company



the packet are displayed. In the figure above, the vl_valfac_sim_data_1 packet is expanded. A list element is selected by double clicking on the entry. When an entry has been selected, a double click will clear the selection. If a packet is selected, all data elements of the packet will also be selected.

4.9.2 Selected Real-time Display Data Frame.

The frame contains a list of entries selected for realtime display on this interface. The list displays the data type, payload byte offset, byte length. The order of the list can be changed by selecting an entry and pulling the entry to a new location in the list.

4.9.3 Button Box.

- <u>Read RTD File.</u> This button will open a file selection dialog. The file selected will be imported in as selected items. This will be displayed in the selected realtime displa data frame. The read file must have be saved using the Save RTD File button.
- <u>Save RTD File</u>. This button will open a file selection dialog. The selected realtime display items will be saved to the file selected. This will allow the user to create display lists that may be restored (Read RTD File) with a minimal amount of work.
- <u>Clear All Selected</u>. This button will clear all selected data items contained in the selected realtime display data frame.
- <u>Capture</u>. Depress the capture button to start execution of a capture.
- <u>Cancel.</u> Depress the cancel button to close the option dialog notebook and restore default values.

5. Active Data Capture.

This section describes the unique operations that can be performed when the capture is active and the system is receiving data. There is a general data notebook, packet info page and data element display page for each interface in the system. These are described below:

5.1 Capture Pkt Info Pages.

After starting a capture, the capture packet info page will be displayed for the selected interface. This page will display information on all active packets on the interface. Each interface will have a different appearance. But the information will be similar and include total packets, rates, percent of total as labeled in the widget header. A packet that is invalid for more than 1 second will be marked with a red foreground color.



•	<u> </u>				n AFDX/Aring	: 664 netw	ork interia	æ)					
Ge	nera	al AFDX P	2P CAN											
A	FDX	Name	VL/Port	TotalPkts A	Period (MS) A	Pkt/Sec A	%ofTotal A	TotalPkts B	Period (MS) B	Pkt/Sec B	%ofTotal B			
⊳	VL	. 1	1	333	50	19.7	16.7%	0	0	0.0	0.0%			
Þ	VL	2	2	333	50	19.7	16.7%	0	0	0.0	0.0%			
⊳	VL	. 3	3	333	50	19.7	16.7%	0	0	0.0	0.0%			
	VL	4	4	333	50	19.7	16.7%	0	0	0.0	0.0%			
		PORT 4	4	333	50	19.7	100.0%	0	0	0.0	0.0%			
⊳	VL	. 5	5	333	50	19.7	16.7%	0	0	0.0	0.0%			
	VL	6	6	333	50	19.7	16.7%	0	0	0.0	0.0%			
		PORT 6	6	333	50	19.7	100.0%	0	0	0.0	0.0%			
Cap	oture	e Running										00:00:18		
												[😢 <u>S</u> top	b <u>C</u> lear
													C 3rop	⊘ clear

5.1.1 Button Box.

- <u>Stop.</u> Depress the button to stop execution of a capture.
- <u>Pause.</u> Depress the button to pause display. Depress again to restart display.
- <u>Clear</u>. Depress the clear button to the remove all packets from list and reset counters to 0.



ze Data		Size Distrib	ution			Timing		
escription Value	Size Dist	Value	% of Total		Description Value			
N PKT 135 AX PKT 299 FE PKT 217 JTAL KBYTES 3786.2K DTAL PKTS 17448	<128 129256 257512	<128 0 0.0% 129.256 8724 50.0% 257.512 8724 50.0% 512.1024 0 0.0%			START TIME Wed Oct 11 11:42:53 20 DURATION 00:01:24 PKTS/SEC 207.7 KBYTES/SEC 45.1		1:42:53 2006	
FDX P2P A429 CAN				Captured Pa	ckets	1		
FDX Errors Conditions Error Types					17448	% (% of total	
P CHECKSUM CRC FOO BIG	0 0 0 0	0 0		AFDX	17448	A	AFDX	
FOO SMALL NO BUFFER NPUT FIFO OVERFLOW MEMORY	0 0 0			P2P	0		P2P	0.0
NTEGRITY RSN MISMATCH NO PORT PRESENT	0 0 0			A429	0	A	429	0.0
IO VL PRESENT P LENGTH MISMATCH RAGMENT REASSEMBLY	0 0 0			CAN	0		CAN	0.0
VRONG NETWORK ACKET TRUNCATED TO FIT MPTY	0 0 0			IP-UDP	0	IP	-UDP	0.0
				IP-TCP	0	IF	P-TCP	

5.2 Capture Info Page.

The general capture info notebook will provide various general pieces of information to the user. The user will have to select the General notebook tab to display the page.

- <u>Size Data.</u> This frame contains general size (min,max,average) information about packets.
- <u>Size Distribution</u>. This frame contains general information about the number of packets contained in various size ranges.
- <u>Timing</u>. This frame contains time information about the capture. This includes start time, length of capture, and interface data rates.
- <u>Error Conditions</u>. This is a notebook containing pages for each interface. Each page contains a list of different types of receive errors related to the interface. This is the number of errors determined since the start of the capture.
- <u>Captured Packets</u>. This frame contains a set of progress bars for each interface. It lists the number of packets received on the interface and the percent of the total.

5.3 Packet Info Page.

This page displays higher level traffic information about the interface. The list above contains entries for each VL/Port (afdx) received by the system. A list entry will display various information like number of packets received, errors, data rates, percent of total. So, the user may view the page, to verify which VL (afdx) are being transmitted and at what rate and expanded to port contained on the VL. The a429/p2p interface have a tree structure with channels at the top and the label/ports for each channel displayed as a branch. The page information is from a live capture and updated at a 1 sec rate.

This page will contain different columns for each interface. Each interface page will look different but the concept of displaying information at a packet level is the same.



The Stop button will stop execution of the capture.

	AFDX RT Displa	y Filename	e: /tmp/afdx7354	l.rtd			
Name	Туре	VL	Dest Port	Dest IP Addr	Value	Min	Max
vl_valfac_sim_data_1.reserved_	FT_UINT32	1	1	10:1:0:1	602	1	602
/l_valfac_sim_data_1.dw_slowongndinvld1_	FT_UINT16	1	1	10:1:0:1	00090	00000	00255
/l_valfac_sim_data_1.slowongndinvld1	FT_BOOLEAN	1	1	10:1:0:1	FALSE	00000	00255
/l_valfac_sim_data_1.slowongnd1	FT_BOOLEAN	1	1	10:1:0:1	FALSE	00000	00255
'l_valfac_sim_data_1.swf1autoslatcmd	FT_BOOLEAN	1	1	10:1:0:1	FALSE	00000	00255
/l_valfac_sim_data_1.chydsyspressell	FT_FLOAT	1	1	10:1:0:1	0.000	0.000	0.000
/l_valfac_sim_data_1.lhydsyspressell	FT_FLOAT	1	1	10:1:0:1	0.000	0.000	0.000
l_valfac_sim_data_1.rhydsyspresselr	FT_FLOAT	1	1	10:1:0:1	0.000	0.000	0.000
/l_valfac_sim_data_1.mach	FT_FLOAT	1	1	10:1:0:1	0.000	0.000	0.000
/l_valfac_sim_data_1.trueairspeed	FT_FLOAT	1	1	10:1:0:1	0.000	0.000	0.000
/l_valfac_sim_data_1.impactpressure	FT_FLOAT	1	1	10:1:0:1	0.000	0.000	0.000
/ valfac_sim_data_1.baroaltitudeers	FT_FLOAT	1	1	10:1:0:1	0.000	0.000	0.000
/l valfac sim data 1.altituderateers	FT_FLOAT	1	1	10:1:0:1	0.000	0.000	0.000
t	00:00:44	111					

5.4 Data Display Page.

This page displays individual payload data items contained on the interface. The list contains entries for each data element selected from the real time option page <u>#5.6Real-Time Display Option Pages.|outline</u> A list entry will display various information like data type, VL/Port, engineering units value, min value, and max value. So, the user can view select payload data elements updated at an approximately 1 sec rate while the capture is executed.

This page will contain different columns for each interface. Each interface page will look different but the concept of displaying information at a payload data level is the same.

6. Digital Filters.

A digital filter provides a way to eliminate packets from the packet pane that do not meet the boolean condition of the digital filter. Each filter consists of one or more expressions combined with logical operators. For example: (expr1 and expr2) or expr3). Each expression consists of a protocol data field, comparison operator, and literal constant. For example: (frame.pkt_len > 10), (udp.dstport == 12312). After the user has constructed a few digital filters, the procedure will be easy to understand.

6.1 Digital Filter Display Pane.

The digital filter pane consists of 4 buttons (*Filter*, *Expression*, *Clear*, and *Apply*) and an entry box pulldown.

• <u>Filter Button</u>. The following dialog is displayed when the filter button is depressed. The dialog box will allow the user to create or select saved digital filters. The digital filters are saved in the *dfilters* file contain in the analyzer configuration directory. When a digital filter is selected with the apply button, it will be displayed in the digital filter entry box. The user can create new filters using the new



button or remove filter with the delete button. The save button will copy all filter data in the *dfilter* file. Each entry in the *dfilter* file consists of a filter name (something that can be remembered) and the actual filter text string.

Edit	Filter	
	VL_ADRF_C_OUTPUT 3	*
	VL_ADRF_L_OUTPUT 1	
New	VL_ADRF_R_OUTPUT 2	
<u> I</u>	VL_IRU_L_OUTPUT 4	
	VL_IRU_R_OUTPUT 5	
	MTF_TO_CMCF 5062	1
	CMCF_TO_MTF 5066	
	FCM_CENTER_RX 5469	
Pelete	FCM_LEFT_RX 5069	
	FCM_RIGHT_RX 5269	
	VL_VALFAC_SIM_DATA 5650	
	New filter	*
Properties		
Filter name:	VL_ADRF_L_OUTPUT 1	
Filter string:	oth viewal	Expression
riiter string:		Expression
🚱 <u>H</u> elp	Save Apply X Close	<u>о</u> к

- <u>Entry Widget</u>. The user can enter a digital filter text string in this field. The rules for creating digital filters is described in <u>#7.2Expression Syntax</u>. The user can use the pulldown to select previously used digital filters. A green background indicates that the expression is valid. A red background indicates that the expression does not have the correct syntax.
- <u>Expression Button</u>. When the expression button is depress, the expression dialog is displayed. The dialog provides a simple way to create filter expressions. The protocol field names are listed in a tree structure containing a row for each item. The protocol can be expanded to display all field elements contained in the protocol. The user can select the field name, relation operator, and literal constant value. When the OK button is depress, the dialog is removed and the expression is displayed in the digital filter entry widget. The created string is appended to end of text in entry box. So, it might be necessary to add a relation operator between two expressions if there is text in the entry box.



eld name	Relation	Value (floating point (single-precision
a429_mcp_b_rcv_w316	is present	1.234
a429_mcp_b_rcv_w320	==	Predefined values:
a429_mcp_b_rcv_w322	!=	2
a429_mcp_b_rcv_w365	>	
A661	<	
AFDX_1	>=	
AFDX_2	<=	
AFDX_58798		
AFDX_58908		
AFDX_59273		
afdx_59273.eecb_eng_t20_local_l - eecb_eng_t20_local_l		
afdx_59273.eecb_eng_t20_local_r - eecb_eng_t20_local_r		
AFDX_59276		Records Vallence (accepted)
AFDX_59277		Range (offset/length)
AFDY FOATA	* I I I I I I I I I I I I I I I I I I I	8 N

- <u>Clear Button</u>. This button is used to clear all text from the entry box widget.
- <u>Apply Button</u>. This button is used to apply the digital filter contained in the entry widget and update the packet list.

6.2 Expression Syntax.

A digital filter can be broken down into the following basic blocks. This section contains details of the basic building blocks of a digital filters.

<digital filter>=<expression> <relational operator> <expression>

<expression> = <protocol data field> <compare operator> <literal constant>

protocol data field>=this string is any protocol data item defined in the system. The user can depress the expression button to display a dialog with protocol data item listed in a tree format. This dialog is described in the *Expression* button section above.

<comparison operator>=There are 6 operators defined. They are used to compare a date item with a constant value. Each has an English term and C language term.

English	С	Description
	=\=	Equal (eth.vl==1212)
ne	!=	Not Equal (eth.vl != 1212)
gt	>	Greater Than (eth.vl > 1212)
lt	<	Less Than (eth.vl lt1212)
ge	>=	Greater Than or Equal (eth.vl >= 4545)
le	<=	Less Than or Equal (eth.vl le 4545)



literal constant>=constant literal has a type as listed below:

Types	Description
signed/unsigned integer	These values can be decimal, octal, or hexadecimal.
	eth.vl == 1122
	eth.vl == O366
	eth.vl == 0x345
Boolean	test for existance of data item in packet.
Ethernet Address	eth.addr == 12:34:56:78:9a:bc
IPv4 Address	ip.addr == 255.255.128.128
floating point	1.23

<relation operators>=these terms are used to combine expressions with other expressions to create more complex boolean expressions. There are 4 relational operators with English and C terms.

English	C	Description
and	&&	Logical AND
or		Logical OR
xor	^^	Exclusive OR
not	!	Logical NOT

6.3 Protocol Data Formats.

The main static protocols used in filters are:

• Frame (contains high level information about the packet).

frame.cap_len Capture Frame Length. Unsigned 32-bit integer.

frame.file_off File Offset Signed 32-bit integer.

frame.link_nr Link Number. Unsigned 16-bit integer.

frame.marked Frame marked. Boolean. Frame is marked in the GUI.

frame.number Frame Number. Unsigned 32-bit integer.

frame.board Adfx board number. Unsigned 32-bit integer.

frame.pkt_len Total Frame Length. Unsigned 32-bit integer.

frame.protocols Protocols in frame. String. Protocols carried by this frame.

frame.ref_time This is a Ref Time frame. No value. This frame is a Reference Time frame.

frame.time Arrival Time. Date/Timestamp. Absolute time when this frame was captured.

frame.time_delta Time delta from previous packet. Time duration. Time delta since previous diplayed frame.

frame.time_relative Time since reference or first frame. Time duration. Time relative reference or first frame.

• Ethernet (afdx ethernet packet parameters)

eth.addr Source or Destination Address. 6-byte Hardware (MAC) Address. Source or Destination Hardware Address

eth.dst Destination. 6-byte Hardware (MAC) Address. Destination Hardware Address

eth.equipment_id Equipment ID. Unsigned 8-bit integer. Equipment ID

eth.if Interface ID. Unsigned 8-bit integer. Interface ID

eth.len Length. Unsigned 16-bit integer eth.network_id Network ID. Unsigned 8-bit integer. eth.src Source. 6-byte Hardware (MAC) Address. Source Hardware Address eth.trailer Trailer Byte array. Ethernet Trailer or Checksum eth.trailer_crc CRC. Unsigned 32-bit integer. Ethernet CRC eth.trailer_rsn RSN. Unsigned 8-bit integer. Redundancy Sequence Number eth.type Type. Unsigned 16-bit integer eth.vl VL. Unsigned 16-bit integer. Virtual Link Number

- **EDE** (afdx EDE packet parameters).
 - ede.sn Unsigned 16 bit integer.
 - ede.timeu Unsigned 16 bit integer.
 - ede.timel Unsigned 32 bit integer.
 - ede.crcx Unsigned 16 bit integer.
 - ede.crcy Unsigned 16 bit integer.
- **IP**, (afdx IP packet parameters).
 - ip.addr Source or Destination Address. IPv4 address.
 - ip.checksum Header checksum. Unsigned 16-bit integer.
 - ip.checksum_bad Bad Header checksum. Boolean.
 - ip.dsfield Differentiated Services field. Unsigned 8-bit integer.
 - ip.dsfield.ce ECN-CE. Unsigned 8-bit integer.
 - ip.dsfield.dscp Differentiated Services Codepoint. Unsigned 8-bit integer.
 - ip.dsfield.ect ECN-Capable Transport (ECT). Unsigned 8-bit integer.
 - ip.dst Destination. IPv4 address.
 - ip.flags Flags. Unsigned 8-bit integer.
 - ip.flags.df Don't fragment. Boolean.
 - ip.flags.mf More fragments. Boolean.
 - ip.flags.rb Reserved bit. Boolean.
 - ip.frag_offset Fragment offset. Unsigned 16-bit integer.
 - ip.fragment Frame number. IP Fragment.
 - ip.fragment.error Defragmentation error Frame number. Defragmentation error due to illegal fragments.

ip.fragment.multipletails Multiple tail fragments found. Boolean. Several tails were found when defragmenting the packet.

ip.fragment.overlap Fragment overlap. Boolean. Fragment overlaps with other fragments.

ip.fragment.overlap.conflict Conflicting data in fragment overlap. Boolean. Overlapping fragments contained conflicting data.

ip.fragment.toolongfragment Fragment too long. Fragment contained data past end of packet.

ip.fragments No value.IP Fragments

- ip.hdr_len Header Length. Unsigned 8-bit integer.
- ip.id Identification. Unsigned 16-bit integer.

ip.len Total Length. Unsigned 16-bit integer. ip.proto Protocol. Unsigned 8-bit integer. ip.reassembled in Reassembled IP in frame. Frame number This IP packet is reassembled in this frame. ip.src Source. IPv4 address. ip.tos Type of Service. Unsigned 8-bit integer. ip.tos.cost Cost. Boolean. ip.tos.delay Delay. Boolean. ip.tos.precedence Precedence. Unsigned 8-bit integer. ip.tos.reliability Reliability. Boolean. ip.tos.throughput Throughput. Boolean. ip.ttl Time to live. Unsigned 8-bit integer. ip.version Version. Unsigned 8-bit integer. • UDP (afdx UDP packet parameters). udp.checksum Checksum --Unsigned 16-bit integer udp.checksum_bad Bad Checksum --Boolean udp.dstport Destination Port -- Unsigned 16-bit integer

udp.length Length -- Unsigned 16-bit integer

- udp.port Source or Destination Port -- Unsigned 16-bit integer
- udp.srcport Source Port --Unsigned 16-bit integer
- AFDX_VL<vl>_PORT<port>* (one for afdx payload packet defined). These protocols are defined based on data contained in data definition files.
- **P2PHDR** (P2P header info parameters, at the begininning of all p2p messages).

p2phdr.crc CRC. Unsigned 16-bit integer. P2P CRC.

p2phdr.dest DEST. Unsigned 16-bit integer. P2P Dest Location.

p2phdr.id ID. Unsigned 16-bit integer. P2P ID.

p2phdr.label LABEL. Unsigned 16-bit integer. P2P Msg Label.

p2phdr.length LENGTH. Unsigned 16-bit integer. P2P Msg Length.

p2phdr.src SRC. Unsigned 16-bit integer. P2P Src Location.

- **P2P_<port>*** (one for each P2P payload packet defined). These protocols are defined based on data contained in data definition files.
- A429 (A429 common info parameters, label, sdi, and ssm).

a429.label label. Unsigned 8-bit integer.

a429.sdi sdi. Unsigned 8-bit integer.

a429.ssm ssm. Unsigned 8-bit integer.

a429.word word. Unsigned 32-bit integer.

- <a429_busname>_<label>* (one for each A429 payload packets defined). These protocols are defined based on data contained in data definition files.
- CANHDR (CAN header info, at the beginning of all CAN messages).

canhdr.eff_id Unsigned 32-bit integer. CAN EFF Identifier.

canhdr.ff Unsigned 8-bit integer. CAN Frame Format. canhdr.len Unsigned 8-bit integer. CAN Data Length. canhdr.rtr Unsigned 8-bit integer. CAN Remote Transmission Request. canhdr.sff_id SFF ID --Unsigned 16-bit integer. CAN SFF Identifier

- CAN_<can id>* (CAN payload packets). These protocols are defined based on data contained in data definition files.
- A661 Header parameters in A661 headers.

a661_hdr.blk_seq Unsigned 32-bit integer Sequence Number a661_hdr.data_blk_size Unsigned 16-bit integer Data Block Size a661_hdr.dest Unsigned 8-bit integer Destination a661_hdr.extended_blk_size Unsigned 16-bit integer Extended Block Size a661_hdr.health Unsigned 8-bit integer Health Assumed a661_hdr.lowest_seq Unsigned 32-bit integer Lowest Sequence Number a661_hdr.num_of_grps Unsigned 8-bit integer Number of Groups a661_hdr.service Unsigned 8-bit integer Service Available a661_hdr.source Unsigned 8-bit integer Source a661_hdr.start_marker Unsigned 16-bit integer Start Marker

• A661 parameters in A661 blocks.

a661.bb context num Unsigned 16-bit integer BEGIN BLOCK: Context Num a661.bb keywork Unsigned 8-bit integer BEGIN BLOCK: Keyword a661.bb_layer_id Unsigned 8-bit integer BEGIN BLOCK: Layer Id a661.bb size Unsigned 32-bit integer BEGIN BLOCK: Size a661.cmd keyword Unsigned 16-bit integer Command Type a661.cmd size Unsigned 16-bit integer Command Structure Size a661.cmd widget id Unsigned 16-bit integer Command Widget ID a661.eb data grp size Unsigned 16-bit integer END BLOCK Data Group Size: a661.eb_keyword Unsigned 8-bit integer END BLOCK Keyword: a661.eb seq num Unsigned 32-bit integer END BLOCK Sequence Number: a661.eb start marker Unsigned 16-bit integer END BLOCK Start Marker: a661.enable_array_index Unsigned 8-bit integer Enable Array Entry Index a661.enable array value Unsigned 8-bit integer Enable Array Value a661.entry array enable Unsigned 8-bit integer Entry Array Enable a661.entry array index Unsigned 8-bit integer Entry Array Index a661.entry array num Unsigned 16-bit integer Entry Array Num Of Entries a661.entry array str Unsigned 16-bit integer Entry Array String Data a661.entry_array_strlen String Entry Array String Length a661.entry popup array picture Unsigned 16-bit integer Entry PopUp Array Picture a661.event keyword Unsigned 16-bit integer Event Type

a661.event origin Unsigned 16-bit integer Event Origin a661.event value Unsigned 32-bit integer Event Value a661.evt active panel id Unsigned 16-bit integer Event Active Tabbed Panel Id a661.evt button state Unsigned 8-bit integer Check Button State a661.evt cursor x Unsigned 32-bit integer Event Cursor X Position a661.evt cursor y Unsigned 32-bit integer Event Cursor Y Position a661.evt first visible Unsigned 16-bit integer Event First Visible Entry a661.evt_frame_x_Unsigned 32-bit integer Event Frame Position X a661.evt frame y Unsigned 32-bit integer Event Frame Position Y a661.evt item index Unsigned 16-bit integer Event Item Index a661.evt map x Unsigned 32-bit integer Event Map X Position a661.evt map y Unsigned 32-bit integer Event Map Y Position a661.evt num of inc Unsigned 32-bit integer Event Number of Increments a661.evt_selected_entry_Unsigned 16-bit integer Event Selected Entry a661.evt strlen Unsigned 16-bit integer Event String Length a661.evt sync data1 Unsigned 32-bit integer Sync Data Value 1 a661.evt sync data2 Unsigned 32-bit integer Sync Data Value 1 a661.evt_sync_datatype Unsigned 8-bit integer Sync Data Type a661.evt sync linkid Unsigned 16-bit integer Sync Link Id a661.evt value Unsigned 32-bit integer Event Value a661.except keyword Unsigned 16-bit integer Exception Type a661.mapitem axis ratio Unsigned 32-bit integer Map Item Axis Ratio a661.mapitem blk num Unsigned 8-bit integer MapItem Block Number a661.mapitem clear Unsigned 16-bit integer Map Item Clear Flag a661.mapitem color Unsigned 8-bit integer Map Item Color a661.mapitem crschg Signed 32-bit integer Map Item Course Change a661.mapitem_endflag_Unsigned 8-bit integer Map Item End Flag a661.mapitem fill style Unsigned 8-bit integer Map Item Fill Style Index a661.mapitem inbnd crs Signed 32-bit integer Map Item Inbound Course a661.mapitem index Unsigned 16-bit integer Map Item Index a661.mapitem last blk Unsigned 8-bit integer Last Block Boolean a661.mapitem legend str String Map Item Legend String a661.mapitem length Signed 32-bit integer Map Item Length a661.mapitem num Unsigned 16-bit integer Map Item Num of Elements a661.mapitem orientation Signed 32-bit integer Map Item Orientation a661.mapitem radius Signed 32-bit integer Map Item Radius a661.mapitem relative pos Unsigned 8-bit integer Map Item Relative Position a661.mapitem_styleset Unsigned 16-bit integer Map Item Style Set a661.mapitem symbol type Unsigned 16-bit integer Map Item Symbol Type



a661.mapitem sync data1 Unsigned 32-bit integer Map Item Sync Data1 a661.mapitem sync data2 Unsigned 32-bit integer Map Item Sync Data 2 a661.mapitem sync type Unsigned 8-bit integer Map Item Sync Type a661.mapitem_type Unsigned 8-bit integer Map Item Type a661.mapitem update num Unsigned 8-bit integer Sequence of Block a661.mapitem x lat range Signed 32-bit integer Map Item X Lat Range a661.mapitem x lat range2 Signed 32-bit integer Map Item X Lat Range 2 a661.mapitem_y_lat_bearing_Signed 32-bit integer Map Item Y Long Bearing a661.mapitem y lat bearing2 Signed 32-bit integer Map Item Y Long Bearing 2 a661.notify keyword Unsigned 16-bit integer Notitication Type a661.parm 1byte value Unsigned 8-bit integer SET PARAMETER: value 1 byte a661.parm 2byte value Unsigned 16-bit integer SET PARAMETER: value 2 byte a661.parm 4byte value Unsigned 32-bit integer SET PARAMETER: value 4 byte a661.parm_8byte_value1 Unsigned 32-bit integer SET PARAMETER: value1 8 byte a661.parm 8byte value2 Unsigned 32-bit integer SET PARAMETER: value2 8 byte a661.parm enable Unsigned 8-bit integer SET PARAMETER: enable a661.parm entry idx Unsigned 8-bit integer SET PARAMETER: entry index a661.parm_id Unsigned 16-bit integer CMD SET PARAMETER: id a661.parm num of str Unsigned 16-bit integer SET PARAMETER: number of strings a661.parm str String SET PARAMETER: string data a661.parm str idx Unsigned 16-bit integer SET PARAMETER: string index a661.parm str size Unsigned 16-bit integer SET PARAMETER: string size a661.req keyword Unsigned 16-bit integer CMD UA REQUEST: Request Type a661.req widget id Unsigned 16-bit integer CMD UA REQUEST: Widget ID a661.sym alignment Unsigned 8-bit integer Text Alignment a661.sym closed Unsigned 8-bit integer Symbol Number of Vertices a661.sym_color_idx Unsigned 8-bit integer Symbol Color Index a661.sym endangle Unsigned 32-bit integer Symbol End Angle a661.sym endx Unsigned 32-bit integer Symbol End X Position a661.sym endy Unsigned 32-bit integer Symbol End Y Position a661.sym filled Unsigned 8-bit integer Symbol Filled a661.sym font idx Unsigned 8-bit integer Symbol Font Index a661.sym halo Unsigned 8-bit integer Symbol Halo a661.sym inner radius Unsigned 32-bit integer Symbol Inner Radius a661.sym line style Unsigned 16-bit integer Symbol Line Style a661.sym linelength Unsigned 32-bit integer Symbol Line Length a661.sym num of vertices Unsigned 16-bit integer Symbol Number of Vertices a661.sym_outer_radius Unsigned 32-bit integer Symbol Outer Radius a661.sym radius Unsigned 32-bit integer Symbol Circle Radius

a661.sym_rotation_angle Unsigned 32-bit integer Symbol Rotation Angle a661.sym_size_x Unsigned 32-bit integer Symbol X Size a661.sym_size_y Unsigned 32-bit integer Symbol Y Size a661.sym_startangle Unsigned 32-bit integer Symbol Start Angle a661.sym_startx Unsigned 32-bit integer Symbol Start X Position a661.sym_starty Unsigned 32-bit integer Symbol Start Y Position a661.sym_x Unsigned 32-bit integer Symbol X Position a661.sym_x2 Unsigned 32-bit integer Symbol X2 Position a661.sym_x3 Unsigned 32-bit integer Symbol X3 Position a661.sym_y Unsigned 32-bit integer Symbol Y3 Position a661.sym_y2 Unsigned 32-bit integer Symbol Y2 Position a661.sym_y3 sym_y3 Unsigned 32-bit integer Symbol Y3 Position

• EFIS Buffers.

efis.baro hpa 747-400 Unsigned 32-bit integer Baro HPA (747-400) efis.baro in 747-400 Unsigned 32-bit integer Baro In (747-400) efis.baro metric 747-400 Unsigned 32-bit integer Baro Metric (747-400) efis.conic_angle Conic Definition Word (Subtended Angle) efis.conic init angle Conic Definition Word (Init Angle) efis.conic radius Conic Definition Word (Radius) efis.dyn state Unsigned 32-bit integer Start of Dynamic Data efis.eot Unsigned 32-bit integer EOT efis.fill in Unsigned 32-bit integer Fill-In Word efis.fpv 747-400 Unsigned 32-bit integer FPV (747-400) efis.label Unsigned 8-bit integer efis.lat Vector Latitude Word efis.lat fine Vector Latitude Word efis.long_fine Vector Latitude Word efis.metric alt 747-400 Unsigned 32-bit integer Metric Alt (747-400) efis.mode approach Unsigned 32-bit integer Mode Approach efis.mode map Unsigned 32-bit integer Mode Map efis.mode plan Unsigned 32-bit integer Mode Plan efis.mode_pos Unsigned 32-bit integer Mode Pos efis.mode sta Unsigned 32-bit integer Mode STA efis.mode vor Unsigned 32-bit integer Mode VOR efis.mode vor adf Unsigned 32-bit integer Mode VOR/ADF efis.mode_word Unsigned 32-bit integer Mode Word efis.mode wpt Unsigned 32-bit integer Mode WPT efis.range dist 747-400 Unsigned 32-bit integer Range Dist (747-400)

efis.range_dist_747-8 Unsigned 32-bit integer Range Dist (747-8)

efis.range word 747 400 Unsigned 32-bit integer Range Word efis.range word 747 8 Unsigned 32-bit integer Range Word efis.sdi Unsigned 8-bit integer efis.sot Unsigned 32-bit integer SOT efis.sot blk cnt Unsigned 32-bit integer SOT Block Count efis.sot word cnt Unsigned 32-bit integer SOT Word Count efis.ssm Unsigned 8-bit integer efis.sym_azimuth Symbol Azimuth Word (rotated symbols only) efis.sym dist 747-400 distance 747-400 Distance Word (747-400) efis.sym dist 747-8 Distance Word (747-8) efis.sym rwy leng Symbol Length Word (runway symbol only) efis.text id String Text ID String efis.word Unsigned 32-bit integer efis Data efis.wxr_data_747-400 Unsigned 32-bit integer Wxr Data (747-400)

• Arinc 739.

a739.ack Unsigned 32-bit integer ACK Word a739.bg Unsigned 32-bit integer Background Word a739.cmd Unsigned 32-bit integer Command Code a739.cntrl Unsigned 32-bit integer Cntrl Word a739.cntrl color Unsigned 32-bit integer Line Number a739.cntrl flash Unsigned 32-bit integer Flashing a739.cntrl init char Unsigned 32-bit integer Init Char Position a739.cntrl line num Unsigned 32-bit integer Line Number a739.cntrl underscore Unsigned 32-bit integer Underscore a739.cts Unsigned 32-bit integer CTS Word a739.data Unsigned 32-bit integer Data Word a739.data_str String Data Word String a739.disc Unsigned 32-bit integer Discrete Word a739.disc blank screen Unsigned 32-bit integer Blank Screen a739.disc clr acars Unsigned 32-bit integer Clear ACARS Request a739.disc clr disp Unsigned 32-bit integer Clear Display Buffer a739.disc_clr_fmc Unsigned 32-bit integer Clear FMC Request a739.disc clr sysstem Unsigned 32-bit integer Clear Subsystem Request a739.disc dspy ann Unsigned 32-bit integer DSPY Annunciator a739.disc exec ann Unsigned 32-bit integer EXEC Annunciator a739.disc msg ann Unsigned 32-bit integer MSG Annunciator a739.disc ofst ann Unsigned 32-bit integer OFST Annunciator a739.disc self test Unsigned 32-bit integer Self Test

a739.disc ssm Unsigned 32-bit integer SSM

a739.disc test req Unsigned 32-bit integer MCDU Test Request a739.eicas Unsigned 32-bit integer EICAS Discrete Word a739.eicas mcdu port Unsigned 32-bit integer EICAS MCDU Input Port a739.eng Unsigned 32-bit integer ENQ Word a739.enq mal Unsigned 32-bit integer ENQ MAL a739.eot Unsigned 32-bit integer EOT Word a739.etx Unsigned 32-bit integer ETX Word a739.func Unsigned 32-bit integer Function a739.label Unsigned 32-bit integer Label a739.mal label Unsigned 32-bit integer MAL Label a739.max rec cnt Unsigned 32-bit integer MAX Record Count a739.mcdu id Unsigned 32-bit integer MCDU ID Word a739.mcdu status Unsigned 32-bit integer MCDU Status a739.nak Unsigned 32-bit integer NAK Word a739.pb Unsigned 32-bit integer Push Button Word a739.pb code Unsigned 32-bit integer Push Button Code a739.pb seq num Unsigned 32-bit integer Push Button Sequence Number a739.port1_data Unsigned 32-bit integer Port 1 Data In a739.port1 rcv Unsigned 32-bit integer Port 1 Receiver a739.port2 data Unsigned 32-bit integer Port 2 Data In a739.port2 rcv Unsigned 32-bit integer Port 2 Receiver a739.port3 data Unsigned 32-bit integer Port 3 Data In a739.port3 rcv Unsigned 32-bit integer Port 3 Receiver a739.rec cnt Unsigned 32-bit integer Record Count a739.rts Unsigned 32-bit integer RTS Word a739.sal label Unsigned 32-bit integer SAL Label a739.scratchpad Unsigned 32-bit integer Scratch Pad Word a739.seq num Unsigned 32-bit integer Record Sequence Number a739.stx Unsigned 32-bit integer STX Word a739.stx seq num Unsigned 32-bit integer STX Sequence Number a739.stx wd cnt Unsigned 32-bit integer STX Word Count a739.subsystem id Unsigned 32-bit integer Subsystem ID Word a739.subsystem id sal Unsigned 8-bit integer Subsystem ID SAL a739.syn Unsigned 32-bit integer SYN Word a739.vector Unsigned 32-bit integer Vector Word a739.word Unsigned 32-bit integer Word a739.word cnt Unsigned 32-bit integer Word Count

* M1553.

m1553.broadcast_rcv Boolean



m1553.busy Boolean

m1553.cmd_word Unsigned 16-bit integer

m1553.data_cnt Unsigned 16-bit integer

m1553.dynam_bus_cntr_accept Boolean

m1553.instr Boolean

m1553.mode Unsigned 16-bit integer

m1553.mode_code Unsigned 16-bit integer

m1553.mode_word Unsigned 16-bit integer

m1553.msg_err Boolean

m1553.msg_type Unsigned 16-bit integer

m1553.rt_addr Unsigned 16-bit integer

m1553.serv_reg Boolean

m1553.stat_rt_addr Unsigned 16-bit integer

- m1553.stat_word Unsigned 16-bit integer
- m1553.sub_address Unsigned 16-bit integer

m1553.subsys_flag Boolean

m1553.t_r Boolean

• M1553_<subaddr id>_[tx|rx]* (M1553 payload packets). These protocols are defined based on data contained in data definition files.

* M1553 Generic.

m1553.word[1-32] Unsigned 16-bit integer

6.4 Examples:

- (eth.vl == 1212) and (udp.dstport > 2323) –filter packets such that only packets displayed will have a vl entry of 1212 and destination port greater than 2323.
- frame.pkt_len > 100 -- display only packets with a length greater than 100.
- frame.number > 10 and frame.number < 20 display only the 10-20 packet listed entries.
- ip.addr == 123.124.125.126 and udp.srcport == 65000. -- Display all packets with an ip address of 123.124.125.126 and udp source port of 65000.
- a429.label == 0376 or a429.label == 0377 display all a429 packets with a label of octal 376/377.
- p2phdr.label == 0x1234 display p2p packets with a port/label of 0x1234.

The user may right click in various columns (vl, src/dst port src/dst ip)of the packet window to create display filters. This will allow users to create display filters from data contained in the packet window.

There are 2 options: Apply as Filter, Prepare as Filter in the right click menu. The Apply option menu will execute the updates immediately (same depressing the Display Filter Apply Button). The Prepare option menu will add changes to the display filter string without execution. This will allow the user to add more expressions to the filter before execution.

A download menu from the Apply/Prepare entries will allow the addition of relational operators NOT, AND, OR, NOT AND, and NOT OR. As shown Below. In the first figure, the user move to a vl column, right clicks and selects Prepare a Filter->Selected.



			-	d.			11/10		19	State P		100	ce.cpt - Goel					-		2			4983			W. A		
•			.opi - Goe)																					ŀ	
	Edit View	Go	Capture	Analyz	ze <u>S</u> tat	istics <u>H</u> el	p																				_	
nterfa	aces Opt	ons	Capture	Stop	₽ F	€start	Open		a ve As	X Close	Refresh	Print	Find	a ck	Forward	වන Jump to	Тор	Bottom	Coloriz	e Auto Scroll	Zoom In	Zoom Ou	t Normal Siz	ze Resize (* Columns	CFilter	Filter	
70	ter: eth.vl	== 646	69										×	🖌 🔶 Expressi	on ≽ <u>C</u> le	ar 🕜 App	bly											
rame	. Time		Net	PKT N	Name			VL Num	Src	Port	Src IP		Dest Port	Dest IP		RSN	CRC	Error	Latenc	y Info								
	1 0.000				1g_h3			33695	58		10.42.1		58708		.131.159		c9fd6de9			Payload								
	2 0.000		B RX			age_at_1 age_at_r		34005 34071			10.42.1		58531 58514		.132.213	38 38	71dea8c3 e055ea5e			Payload Payload								
	4 0.000					ck_nsg_2		33266	58		10.42.1		58747		.129.242		adc5d93a			Payload								
	5 0.000			dca_	_misc_	lsp_out_	msg	33286			10.42.4		61971	224.224	.130.6	72	2938c8d9			Payload	Length:	136						
	6 0.000				_fc_i2			32825 64649	59- 60:		10.27.1		59404 60117		.128.57	1e	98cdb42c			Payload								
	7 0.000					a429_mes: _a429_me:			60. 63-		10.42.1		60117		.252.137		b42e4fad f9a5e90e			Payload Payload								
	9 0.000					nessage		64631			10.42.1		59112		.252.119		8ee75e8d	WRONG N	ΙEI	Payload								
	10 0.000							64645			10.42.1		59893		.252.133		68e25396						t: 59893					
	11 0.000		B RX					33545 64633	60 60		10.73.1		60078 60664	224.224		34	e96b7a7c			Payload			rt: 60664					
	12 0.000 13 0.000			mkn	a429	nessage		64655	60		10.42.1		60664 60644		.252.121		c6c614ba c3383383			Fayload			1. 00004					
	14 0.000					nessage		64657	59	756	10.42.1		59756		.252.145		8a0aa0a	WRONG M		Payload	Length:	24						
	15 0.000				s_11			33798	59		10.42.1		59937		.132.6	ba	bd572fc	WRONG N		Payload								
	16 0.000 17 0.000					nessage nessage		64649 64659	60		10.42.1		60133 63462		.252.137		370ba08d 1450506a			Payload Payload								
	18 0.000					nessage		64631	59:		10.42.1		59107		.252.119		6f6ba241			Payload								
	19 0.000	154	B RX	100				64 1	Mark Pac	ket (toggle)		69.0	60099	224.224	.252.157	3b	5afe8ab0	WRONG N	ΙEI	Src Port	: 60099	Dest Por	t: 60099					
	20 0.000				_fc_i1			32 (Dutput G	oebelyzer S		.8	59488 59501		.128.109		54432cbf			Payload								
	21 0.000 22 0.000			msg.	_fc_a2	1				Between P		.4 72.0	60655		.128.103		6f8cc6c7 30656709			Payload Pavload								
	23 0.000			msg.	_fc_i2	1		32	lime Ref	erence		.9	61368		.128.106		459a81ac			Payload								
	24 0.000				_adf_a				Apply as I			32.0	60635		.252.143		71419894			Payload								
	25 0.000				_dne_a			64 1	Prepare a	Filter	,	Selected Not Selec			.252.145		d4627e1a			Payload	Length:							
Fr	ame 19 (171 b	ytes on	wire,	171 b	ytes cap	tured					and Se								03 00 00 00								
Eti	hernet I	I, Sr	c: 02:0	0:00:2	a:a9:2	0, Dst:	03:00	. U	Print			<u>o</u> r Sele								00 98 7d e2 fc 9d ea c3						.}		
In	ternet P	rotoc	ol, Src	Addr:	10.42	2.169.0 (10.42	.1	show Pac	ket in New '	Window			252.157)						e0 5f 00 00								
				, Src	Port:	60099 (6	50099)	Dst	Port:	60099 (6	0099)	og not S	Selected						0040	00 00 00 00	0 00 00 0	00 00 0	$00 \ 00 \ 00$	00 00 0	0 00			
Da	ta (124	bytes)																	00 00 00 00 00 00 00 00 00 00 00 00 00								
																				00 00 00 00								
																				03 03 00 00								
																				00 00 00 00 00 00 c3 ad				00 00 0		Rt;Z		
																			0040	00 00 05 20	52 74 5	o sa re	04 00					
																			8									
																			1									
la t-	ent trace.cr	+1004	5 VB 00.00	10	_	_										_		_		854 D: 58854 M:	0			_				_
ie: Ke	m_uace.cp	1094	S NB 00:00:	10			_	_		_	_		_			_	_	_	P: 58	004 D: 00004 M:	U	_	_	_	_	_	_	

Then the user moves to a destination port column, right clicks and selects Prepare a Filter->... and Selected. At this point the filter can be executed by depressing the Apply button.



Ele E	i	D.	Analyze	Statistics He		Q	×	2	Print	En			8		⊻			0		6	\			V
nterface	-		udp.dstport	Restant 4	Open	Save	As Close	Refresh	Print		d Back	on ≽ <u>C</u> le	Jumj	p to Top	Bottom	Colori	ze Auto Scroll	Zoom Ir	Zoom (Dut Norma	al Size R	esize Columns	Oniter	Filter
-	1.			10000000			1							2	1-	1	1							
rame .	Time	Net B RX	PKT Nam			VL Num	Src Port	Src IP 10.42.1	11 0	Dest Por 58708		101 100	RSN f6		Error WRONG 1	Laten		T	100					
	L 0.000000 2 0.000073		r4_1g_ a429_n	n3 essage_at_1		33695 34005	58708 58531	10.42.1		58708		.131.159			WRONG 1			Length: Length:						
	3 0.000073			essage_at_r	- 1	34071	58514	10.42.1		58514		.133.23			WRONG 1		Payload	Length:	24					
	1 0.000092			block_msg_2		33266	58747	10.42.1		58747		.129.242			WRONG 1			Length:						
	5 0.000163 5 0.000109		dca_mi msg_fc	sc_lsp_out_		33286 32825	61971 59404	10.42.4		61971 59404	224.224	.130.6	72 1e		WRONG N			Length: Length:						
	0.000129			_121 ft_a429_mes		64649	60117	10.27.1		60117		.252.137			WRONG 1			Length:						
	8 0.000074		dcp_ri	ght_a429_me		64659	63465	10.42.1		63465		.252.147		f9a5e90e	WRONG 1	NET		Length:						
	0.000412		hdd_a4	29_nessage		64631	59112	10.42.1		59112		.252.119			WRONG N			Length:			202			
	0.000250					64645 33545	59893 60078	10.42.1 10.73.1		59893 60078		.252.133	41 34		WRONG N			t: 59893 Length:		ort: 598	593			
	2 0.000073					64633	60664	10.42.1		60664		.252.121			WRONG 1			: 60664		ort: 606	664			
	3 0.000073			29_nessage		64655	60644	10.42.1		60644		.252.143			WRONG 1			Length:						
	1 0.000074 5 0.000291		mkp_a4 r2_s_1	29_nessage		64657 33798	59756 59937	10.42.1 10.42.1		59756 59937	224.224	.252.145	67 ba	8a0aa0a	WRONG N			Length: Length:						
	5 0.000291 5 0.000120			1 29_message		53798 64649	60133	10.42.1		60133		.252.137			WRONG 1			Length:						
	0.000073			29_nessage		64659	63462	10.42.1		63462		.252.147			WRONG 1			Length:						
	3 0.000912		hpu_a4	29_nessage		64631	59107	10.42.1		59107	224.224	.252.119	a4		WRONG 1			Length:						
	0.000154		msg_fc	÷11		64669 32877	60099 59488	10.42.1		6009 5948	Mark Packet (WRONG N			t: 60099 Length:		ort: 600	099			
	L 0.000115		msg_fc			32871	59488	10.27.3		5946	Output Goebe				WRONG 1			Length:						
	2 0.000073					64633	60655	10.42.1		6065	Plot Time Bet Time Referen				WRONG 1			Length:						
	3 0.000109		msg_fc			32874	61368	10.27.3		6136			-		WRONG 1			Length:						
	1 0.000073 5 0.000073			f_a429_1 e_a429_1		64655 64657	60635 59769	10.42.1		6063 5976	Apply as Filter		>		RONG 1			Length: Length:						
	0.000073			e_4429_1		CACAD	0170	10.42.1			Prepare a Filt			Selected Not Selected	LONG 1			I						
Fram	ne 19 (171	bytes or	wire, 13	71 bytes cap	tured)							-	and Selected		0000	03 00 00 0							
				a9:20, Dst:							Print Show Packet i			gr Selected		0010 0020	00 98 7d el fc 9d ea cl						}	
				0.42.169.0					252.157	(224.22	Show Packet	n New Windo	w	and not Selecte	d	0030	e0 5f 00 0	0 00 00 0	3 03 0	3 03 00	00 00	00 00 00		
			, Src Por	rt: 60099 (0	50099)	, Dst Po:	rt: 60099	(60099)					3	or not Selected		0040	00 00 00 00							
Data	a (124 byt	es)														0050	00 00 00 00							
																0070	00 00 00 00							
																0080	03 03 00 0							
																0090 00a0	00 00 00 00 00 00 c3 a				00 00	00 00 00	Rt;Z	
																obao	00 00 05 4	1 52 74 5	io ja i	e da bo				
																1								

The user may also right click in the protocol tree to execute a display filter using payload protocol information.



	dit <u>V</u> iew <u>G</u>	o <u>C</u> apture	Analyze	Statistics Hel	p	_		~	_					_					-	-			-
nterface	es Options	Capture	Stop	Restart	Open	Save	As Close	Refresh	Print	Find	Back	Forward	SO Jump to	Тор	Bottom	Colorize	Auto Scroll	Zoom In	Zoom Out	Normal Size	Resize Column:	s CFilter	Filter
Eilter	r: afdx_59501	L.flt_dir_pitch	.pfd_fcm	_app_aff == 0						¥	💠 Expression	≽ <u>C</u> le	ar 🥜 App	у									
rame .		Net	PKT Nar			'L Num	Src Port	Src IP		Dest Port	Dest IP		RSN	CRC	Error	Latenc							
	1 0.000000 2 0.000073		r4_1g	_h3 nessage_at_1		3695 4005	58708 58531	10.42.16		58708 58531	224.224.		f6 38	c9fd6de9 71dea8c3			Payload 1 Payload 1						
	3 0.000073			nessage_at_r		4071	58514	10.42.17		58514	224.224.		38	e055ea5e			Payload 1						
	4 0.000092			_block_msg_2		3266	58747	10.42.14		58747	224.224.			adc5d93a			Payload 1						
	5 0.000163 6 0.000109		dca_m msg_f	isc_lsp_out_		3286 2825	61971 59404	10.42.41		61971 59404	224.224.		72 1e	2938c8d9 98cdb42c			Payload 1 Payload 1						
	7 0.000129			eft_a429_mes		4649	60117	10.42.16		60117	224.224.1			b42e4fad			Payload 1						
	8 0.000074		dcp_r	ight_a429_me	ssage 6		63465	10.42.16		63465	224.224.			f9a5e90e			Payload 1						
	9 0.000412 0 0.000250		hdd_a	429_nessage		4631	59112 59893	10.42.17		59112 59893	224.224.224.224.224.224.224.224.224.224			8ee75e8d 68e25396			Payload I Src Port		24 Dest Por	+· 50803			
	1 0.000133					3545	60078	10.42.17		60078	224.224.		34	e96b7a7c			Payload 1			33033			
12	2 0.000073	B RX		100		4633	60664	10.42.17		60664	224.224.3	252.121		c6c614ba	WRONG N	EI	Src Port	: 60664	Dest Por	t: 60664			
	3 0.000073 4 0.000074			429_nessage 429_nessage		4655	60644 59756	10.42.16		60644 59756	224.224.1			c3383383 8a0aa0a	WRONG N WRONG N		Payload 1 Payload 1						
	5 0.000291		r2_s_			3798	59937	10.42.16		59937	224.224.		ba	bd572fc	WRONG N		Payload 1						
	6 0.000120			429_nessage		4649	60133	10.42.16		60133	224.224.			370ba08d			Payload 1						
	7 0.000073 8 0.000912			429_nessage 429_nessage		4659	63462 59107	10.42.16		63462 59107	224.224.224.224.224.224.224.224.224.224			1450506a 6f6ba241			Payload 1 Payload 1						
	9 0.000154		npu_a	123_nessage		4669	60099	10.42.16		60099	224.224.1			5afe8ab0					Dest Por	t: 60099			
	0 0.000114		msg_f			2877	59488	10.27.3.		59488	224.224.			54432cbf			Payload 1						
	1 0.000115 2 0.000073		msg_f	c_a21		2871 4633	59501 60655	10.27.3.		59501 60655	224.224.			6f8cc6c7 30656709			Payload 1 Payload 1						
	3 0.000109		msg_f	r i21		2874	61368	10.27.3.		61368	224.224.1			459a81ac			Payload 1						
	4 0.000073		msg_	Expand Subtre	es		60635	10.42.16		60635	224.224.			71419894			Payload 1						
	5 0.000073		msg_	Expand All Collapse All			59769	10.42.16		59769	224.224.			d4627e1a			Payload 1						
Fram	me 21 (123	bytes on	wire,	Apply as Filter													03 00 00 00					g	
	ernet II, S			Prepare a Filte	r		elected										00 68 7d e4 80 67 e8 6d					.h} .g.m.m.T .	
	ernet Proto			Plot Variable			ot Selected	a graden	03 (224	.224.128.10	(3)					0030	e0 6d 00 00	00 00 0	3 03 03	03 c2 00 0	00 00 00 00	.n	
User EDE	r Datagram	Protocol,	Src	Plot Time Betw			and Selected	1)									00 00 00 00 00						
	onics Full-	Duplay Se	itcho				gr Selected and not Select	ad									03 03 00 00						
f	lare_retar	d rate for	ann	谢 Wiki Protocol P			or not Selecter	220	0							0070	00 00 b4 85	b3 a4 2	1 6f 8c	c6 c7		lo .	
	lev_spd_cm			Filter Field Ref		f	t/sec range																
	lt_dir_pit					d	egree range	-20-20															
	lt_dir_rol			Besolve Name		- Link	egree range									4							
	akeoff_gui									0)													
	lt_dir_pit						egree range egree range																
	lt_dir_rol: xcess_dev_J						egree range			0)													
	xcess_dev_)				=FALS																		
	resh_msg_f				=0,0x																		
	rc_msg_fc_				=0,0x																		

7. Plotting.

The user can plot the value of specific payload data for the all samples in the displayed trace. The user will right click on "Plot Variable(s)" option in the protocol tree on the payload data to be display As shown below.

The user can also create multiple variable plots using the "Add Variable to Multi-Plot" option.

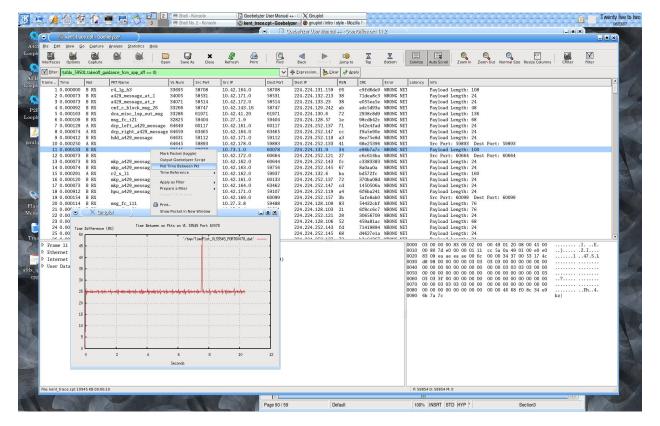
Variables are added using the "add variable to multi-plot" option with the last to plot selected using the "Plot Variable(s)" option.



D Cashal

User Manual

International basis Out of the line line of the li		ces Option	as Capture		1	Open	ave As	X Close	Refresh	Print	Find	Back		SS Jump to	Top	Bottom	Colori	ize Auto Scroll	Zoom In	Com Out	O Normal Size	F* Resize Column	s CFilter	Filter
Image: Description Image:	1p		0001 +-1#		11	~				_					-	Decom	color	at protocola	Loonin	Lonnout	That that size	nesice coornin	, and	
opport Flam.under_fact.grad_fa		D X C	Solquni								Dest Port	Dest IP		RSN		Error	Later	ncy Info						
33.1 24.2.4.12/.13 38 1/100000000000000000000000000000000000	-	den			flare_retar	d_rate_fcm_app_	iff			1			.131.159						ength: 1	08				
32.3 37.4 22.22.129.224 ab 383.6 72 22.82.129.127 ab 20.84.650 NORE NIT Projead length: 86 39.4 22.42.22.129.127 22.42.22.129.127 10.84.6503 NORE NIT Projead length: 86 39.4 22.42.22.129.127 10.84.6503 NORE NIT Projead length: 84 39.4 22.42.22.129.127 10.86.6523 NORE NIT Projead length: 84 39.4 22.42.22.129.127 10.86.6523 NORE NIT Projead length: 84 59.3 39.4 22.42.22.129.127 10.86.6523 NORE NIT Projead length: 84 59.3 22.42.22.21.23.23 41 66.253 NORE NIT Projead length: 24 59.3 22.42.22.21.23.23 41 66.253 NORE NIT Projead length: 24 59.3 22.42.22.21.23.23 41 66.253 NORE NIT Projead length: 24 59.3 22.42.22.23.23.31 41 66.253 NORE NIT Projead length: 24 60.75 NORE NIT Projead length: 24 10.955 NORE NIT Projead length: 24 59.3 22.42.22.23.21.23.12 <td< td=""><td></td><td></td><td>90: -32-0 T</td><td></td><td></td><td></td><td></td><td><u> </u></td><td></td><td>-</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></td<>			90: -32-0 T					<u> </u>		-														
-3.2						'/tmp/flare_n	stard_rate_i	fon_app_af	f,dat' ——															
92.7 0017 224.224.221.37 1 92/24.224.221.37 1 92/24.224.221.37 1 92/24.224.221.27 92/24.224.221.27 92/24.224.221.27 92/24.224.221.27 92/24.224.221.27 92/24.224.221.27 92/24.224.221.27 92/24.224.221.27 92/24.224.221.27 92/24.224.221.27 92/24.224.221.27 92/24.224.221.27 92/24.224.221.27 92/24.224.221.27 92/24.224.224.221.27 92/24.224.224.221.27 <td></td> <td>-32,3 • • • •</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>1</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>Payload L</td> <td>ength: 1</td> <td>36</td> <td></td> <td></td> <td></td> <td></td>		-32,3 • • • •								1								Payload L	ength: 1	36				
3.1 6365 224.224.327.13 3 86906 90000 NN NT Payload Length: 24 3.2 3.3 6845 224.224.327.13 3 869706 90000 NN NT Strepht 24 3.3 3.4 6845 224.224.327.13 4 6842530 80000 NT Strepht 24 3.3 3.4 90072 224.224.327.13 4 9007672 80000 NT Strepht 1600 Strepht 1600 3.3. 3.4 90072 224.224.327.13 4 90078 224.224.327.13 4 90078 80000 NT Strepht 16006 Strepht 1600 Strepht 16006 Strepht 1600 Strepht 1600 Strepht 16006 Strepht 1600 Strepht 16000 Strepht 1600																								
32.1 59893 224.224.232.131 41 68623366 6800.00 NT 57.207.20030 58930 58917.247.224.232.131 59893 58917.247.224.232.131 59893 58917.247.224.232.131 59893 58917.247.224.232.131 59893 58917.247.224.232.131 59893 58917.247.224.232.131 59893 58917.247.224.232.131 59893 58917.247.224.232.131 59893 58917.247.224.232.131 59893 58917.247.224.232.131 59893 58917.247.224.232.131 59893 58917.247.224.232.131 59893 58917.247.224.232.131 59893 58917.247.224.232.131 59893 58917.247.224.232.131 58917.247.224.232.131 58917.247.224.232.131 58917.247.224.232.131 58917.247.224.232.131 58917.247.224.232.131 58917.247.224.232.131 58917.247.224.232.131 58917.247.224.232.131 58917.247.224.232.131 58917.247.247.242.232.123 58917.247.247.242.232.123 58917.247.247.242.232.124 58917.247.247.242.232.124 58917.247.247.242.242.242.242.242.242.242.24		-52,2								1														
242 422 422 423 42 42 42 42 42 42 42 42 42 42 42 42 42		-32.1																			. 50902			
1 1																					: 28082			
91,0 9975 224,224,223,123,16 60 800,000,NET Payload Length: 24 9377 224,224,223,123,17 72 3700,000 Horth: 150 9775 224,224,223,123,17 72 3700,000 Horth: 150 9376 224,224,223,123,17 72 3700,000 Horth: 124 9775 970,000 Horth: 124 9376 224,224,223,123,10 4 66692,424 850,000 NOR NET Payload Length: 24 9376 224,224,223,123,10 4 66692,424 850,000 NOR NET Payload Length: 24 9301 224,224,223,123,10 8 306500 NOR NET Payload Length: 24 9301 224,224,223,123,10 8 3065600 NOR NET Payload Length: 24 9303 224,224,223,123,10 8 3065600 NOR NET Payload Length: 24 9301 224,224,224,234,108,10 8 656626 WOR NET Payload Length: 24 9304 224,224,234,108,10 8 656626 WOR NET Payload Length: 24 9301 224,224,234,108,10 8 656626 WOR NET Payload Length: 24 9304 234,224,234,108,10 8 656626 WOR NET Payload Length: 24 <		-32				-		-			60664	224.224	252.121	27	c6c614ba	WRONG I	VET	Src Port:	60664	Dest Port	: 60664			
3.1 937 22.42.22.12 4 6 9 10 12 6432 22.42.22.12 8 656241 WONK NET Payload Length: 24 -<																								
a.g. f G442 24, 224, 223, 118, 44 G442, 224, 224, 223, 118, 44 Ferral Algorithm of Ferral Algor		-31.9								1	59937							Payload L	ength: 1	60				
31,7		-31.8																						
3.1		0110																						
a1,6 9 2 4 6 8 10 12 60655 224/224/225/252/12 28 906570 90000 90000 90000 90000 90000 90000 90000 90000 90000 90000 90000 90000 90000 90000 90000 90000 90000 90000 900000 900000 900000 900000000 9000000000000000000000000000000000000		-31,7								-		224.224	252.157	Зb	5afe8ab0	WRONG I	VET .	Src Port:	60099	Dest Port	: 60099			
-3.6 -2 -4 6 8 10 12 -5.7																								
Second 0.938 224,224,122,143 68 242,224,222,143 68 242,214,222,143 68 64,20,000,00,00								10			60655	224.224	252.121	28	30656709	WRONG I	(EI	Payload L	ength: 2	4				
Lit Structure Party 2012 (22 (22 (22 (22 (22 (22 (22 (22 (23 (26 (20 (20 (20 (20 (20 (20 (20 (20 (20 (20		Ŷ	2		•	•	0	10		12														
Dit A DORDST R.W 10.14 24100 0010 10.14		23 0.00007	JDRA	insg_come			391	09	10.42.103															
Ethernest II, Src: 02:00:00:1b:03:20, Dist: 03:00:00:00:00:07 000 00:11 cd 3a 0a 10 30 4ed e0 h)				34	- 490 4		001	20	10 10 101	•	C0120	004 004	252 1.27	70	1-2-04207	MDOMC 1		02 00 00 00	80 67 03	4 00 00 1	b 02 20 08	00 45 00		F
Intermet Protocol, Src Addr: 10.27.3.4 (D1.27.3.4), Dt Addr: 224.224.128.103 (224.224.128.103) 0000 80 67 e8 66 e8 60 05 4 00 00 81 12 00 81 17 e7 e8 are reading and reading an							0:80:67										0010	00 68 7d e4	00 00 01	11 cd 3	a Oa 1b 03	3 04 e0 e0	.h}	
User Datagram Protocol, Sre Port: 59501 (59501) 0040 00 00 00 00 00 00 00 00 00 00 00 00								: 224.23	24.128.103	(224	.224.128.10	3)												
Aviencies Full-Duplex Satiched Ethernet: VI. 32871 Port 59501 0060 03 08 00 00 00 00 00 00 00 00 00 00 00 00			m Protoco	1, Src Por	t: 59501 ((59501), Dst	Port: 5	9501 (59	9501)								0040	00 00 00 00	00 00 00	00 00 0	0 03 03 03	3 03 00 00		
Fire_retrad_fate_fcc.app_aff Expand Subtrest gree/s range: 32-0 000 00 00 00 00 00 00 00 00 00 00 00 00																								
elev.spl.cnd_fca.spp.aff flt_dir_pitch_pfd_fca.spp.aff flt_dir_pitch_pfd_fca.spp.aff flt_dir_pitch_pfd_fca.spp.aff flt_dir_pitch_pfd_fca.spp.aff flt_dir_pitch_pfd_fca.spp.aff flt_dir_pitch_pfd_fca.spp.aff flt_dir_pitch_pfd_fca.spp.aff flt_dir_pitch_pfd_fca.spp.aff flt_dir_pitch_pfd_fca.spp.aff flt_dir_pitch_pfd_fca.spp.aff fresh.ssg_fc_a2l_fca.spp.aff fresh.ssg_fc_a2l_fca.spp.aff Beside Name Beside Name								o /c. ran	701 22 0													00 00 00		
f1c.dir.pitch.pdf.dfc.upp.aff couses AI se range: -20-20 f1c.dir.pitch.pdf.dfc.upp.aff Apply a filter se range: -20-20 f1c.dir.pitch.pdf.dfc.upp.aff e range: -20-20 f1c.dir.pitch.pdf.dfc.upp.aff Pequera Finer se range: -20-20 Rot melsenew Wanable e range: -20-20 Rot melsenew Wanable e range: -20-20 Mathematic for app.aff (mainable excess.dev.hul.eft.fc.upp.aff (mainable (rc.usg_fc_a21_fc_uspp.aff (mainable Bestoe Hame Bestoe Hame					expute																			
fit_dir_pit_fct_mapp.aff Apply a filter is e range: -20-20 rkseff_guidance_fcm_app.aff Preyma a filter is e range: -20-20 rit_dir_pit_fch_hud_fcm_app_aff Preyma a filter is e range: -20-20 rexess_dev_hud_right_fcm_app_aff VMN Protect Page is e range: -20-20 if resh_msg_fc_a2l_fcm_app_aff VMN Protect Page is e range: -20-20 if resh_msg_fc_a2l_fcm_app_aff Essive harme is e range: -20-20																								
takeoff_guidance_fcm_upp_aff Propries Pitter ,		flt_dir_ro	oll_pfd_fo	m_app_aff		as filter	, ee	range:	-20-20															
fit.dir.roll.hud.fct.app.aff Pettime Sensee Veribble se range: -20-20 excess_dev.hud.right.fcn.app.aff Verib Area Sape					Prepar	re a Filter					0)													
excess_dev_hud_left_fcm_app_af excess_dev_hud_right_fcm_app_af fresh.msgf_c_a2l_fcm_app_aff Besdev Forenes					1100.10																			
excess_dev_hud_right_fcn_spp_af fresh_msg_fc_a2L_fcn_app_aff ercc_msg_fc_a2L_fcn_app_aff Beside Name Beside Name Beside Name											0)													
fresh_msg_fc_a2L_fcm_app_aff																								
crc_msg_fc_a2l_fcm_app_aff Patces Preferences. Bestave Name		fresh_msg_	fc_a21_fc	m_app_aff	WIND PT																			
		crc_msg_fc	_a21_fcm_	_app_aff																				
					Rerolu	e Name																		
							ckét																	
		:/s range: -32-0	0 (afdx_5950	I.IIare_recaru_i	rate_tent_app	o_an), 4 bytes	_	_		-						_								



Goebelyzer

The user can plot the time between packet samples for all packet of the same type contained in the trace. The user will right click in the packet window on specific packet executing the "Time between packets" option. This is show above.

The data for the plot is contained in a file in the /tmp directory. The .plt file is a script to display the data contained in the .dat file. The filename is created using the data name or packet data. The user can use the command *"ls -lrt /tmp"* to view the latest file contained in the directory.

8. Data Definition Files.

These files define data to display packet elements in various formats. The analyzer will use these files to display packet elements in the format defined in the file. Data definition files are located in the following directories:

- · AFDX /home/user/.ethereal.a664
- · P2P /home/user/.ethereal.p2p.
- · A429 /home/user/.ethereal.a429.
- · CAN /home/user/.ethereal.can.
- · M1553 /home/user/.ethereal.m1553.
- · Spacewire /home/user/.ethereal.spwr
- · FSCC /home/user/.ethereal.fscc
- · ALLDEV /home/user/.ethereal.alldev.

8.1 AFDX XML Data File.

This section describes the some the details of format for the~user/.ethereal.a664/packet-afdx.xml file.

The packet-afdx.xml configuration file is used to provide afdx payload definition information to the user. The format is simple and compact. Each packet definition consists a <packet definition> and a list (1..N) of <elem definitions>. The <packet definition> defines the high level information about the packet include port, vl, protocol, rate, and length. The <elem definition> defines information about a specific data element of the packet including name, mask, type, offset, length, and info. The file consists of these types in the following repeated format: <packet definition, and N <elem definition>s. This will continue until the end of the file.

8.1.1 Packet Data.

- **name** = name description of packet...string of characters...256 max length
- **port** = The packet is received on this port...number value is decimal number with range between 0 and 65536
- vl = The packet is received on this VL number. Format is a decimal between values between 0-65536.
- **ede**= The value of the EDE identifier. A value of 0 will indicate no EDE data is contained in the packet. This field is optional and will default to 0 when no value is specified.
- **protocol** = NONE,A661,A429BLK. This field is optional and used to defined A661 or A429BLK is defined within an afdx packet. The A429BLK is used when the packet defines an A429 messages. The format of the A429BLK packet is:

struct a429blk_payload {
 uint32 reserved;
 uint8 fs1;
 uint8 fs2;



uint8 fs3;

uint8 fs4;

...A429 blocks containing a 2 byte length, 2byte pad, array of 4 byte A429 words of the requested length.

...possibly more A429 blocks.

} a429blk_payload_t;

- Other protocols may be added in the future.
- **proto_offset** = This field is optional used when a A661 protocol is defined. Start byte location of A661 protocol within packet.

8.1.2 Element Data.

- **name** = name description of element...string of characters...256 max length.
- mask = mask used to retrieve data not located on a byte boundary. 0 should be used if the data type uses all bits in the type. Value is hexadecimal number. Examples are 0xffff0000 (upper 16bits of 4 byte word) 0xfffc0, 0x7ffc000, Mask has no meaning for the following: FT_FLOAT,FT_DOUBLE.
- description types FT FLOAT 4 byte floating point number. FT DOUBLE 8-byte floating point number. FT UINT8 1 byte unsigned value. FT UINT16 2 byte unsigned value. FT INT16 2 byte signed (2's complement) value. FT UINT32 4 byte unsigned value. FT HEX 4 byte value display hexadecimal. FT INT32 4 by signed (2's complement) value. FT BOOLEAN 1 bit displayed within a 4 byte word. FT BNR 1,2 or 4 byte scaled integer. FT UBNR 1,2 or 4 byte unsigned scaled integer.
- **type=** the following types are supported for AFDX:

- **range**= float value...max range of scaling for FT_BNR,FT_UBNR. Not used for all other data types. So, fixed pt types will have one more field in the element definition.
- offset= offset in bytes into the payload data where the element is located. Value is decimal number.
- **length**= length in bytes of the data element sizes are listed in the type section for different types. Value is decimal number (1,2 or 4).
- info= string containing units of the engineering data (deg/sec, lbs, feet). Max length of 256 bytes.

AFDX XML Example File:

<afdxicd>

```
<packet name='vl1234' port='1' vl='1' protocol='none' rate='1000' length='20' >
```

<elem name='paul' mask='0xffffffff type='FT UINT32' offset='0' length='4' />



<elem name='paul1' mask='0xffffffff' type='FT_UINT32' offset='4' length='4' />

</packet>

<packet name='vl2345' port='1' vl='2345' protocol='none' rate='1000' length='20' >

```
<elem name='ted' mask='0xffffffff type='FT_UINT32' offset='0' length='4' info='deg/sec' />
```

```
<elem name='ted1' mask='0xffffffff type='FT_UINT32' offset='4' length='4' />
```

</packet>

....

</afdxicd>

8.2 AFDX VL XML File.

This section describes the file format for the~user/.ethereal.a664/packet-afdx_vl_db.xml file. This file is used to attach a name with a VL number.

The following describes the format of the configuration file used to provide afdx vl data definition to the user.

<afdxvlicd> <vl name='vl1' vl='2003' bag='4' /> <vl name='vl2' vl='2203' bag='4' /> <vl name='vl3' vl='2403' bag='4' /> </afdxvlicd>

- **name** = vl name description of packet...string of characters...256 max number of bytes.
- \mathbf{vl} = The number corresponding to the vl name value is decimal number with range between 0 and 65536.
- **bag** = bag value of vl between 0 and 256.

8.3 A429 XML Data File.

This section describes the some the details of format for the ~user/.ethereal.a429/packet-a429.xml file.

The packet-a429.xml configuration file is used to provide a429 label definition information to the user. The format is simple and compact. Each label definition consists a <label definition> and a list (1..N) of <elem definitions>. The <label definition> defines the high level information about the label include port, vl, protocol, rate, and length. The <elem definition> defines information about a specific data element of the packet including name, mask, type, offset, length, and info. The file consists of these types in the following repeated format: cpacket definition, and N <elem definition>s. This will continue until the end of the file.

8.3.1 Label Data.

- **name** = name description of packet...string of characters...256 max length.
- · **label** = The label number. Format is an octal value between 0-377.
- sdi = the sdi of the label. A value of 'XX' indicates no sdi. Otherwise in the range of 0 to 3.
- **rate** = rate of label in milliseconds.



8.3.2 Element Data.

- **name** = name description of element...string of characters...256 max length
- mask = mask used to retrieve data not located on a byte boundary. 0 should be used if the data type uses all bits in the type. Value is hexadecimal number. Examples are 0xffff0000 (upper 16bits of 4 byte word) 0xfffc0, 0x7ffc000.

types	mask used	description
FT_BNR	yes	4-byte scaled integer value.
FT_UBNR	yes	4-byte scaled unsigned value.
FT_UINT8	yes	1-byte unsigned value.
FT_CHAR	no	8-bit ascii value.
FT_ISO5	no	7-bit ascii value.
FT_BOOLEAN	yes	1-bit boolean value within a 4-byte word.
FT_UINT16	yes	2-byte unsigned value.
FT_UINT32	yes	4-byte unsigned value.
FT_BCD	yes	Binary coded decimal value

type= the following types are supported for A429:

- **range**= float value...max range of scaling for FT_BNR,FT_UBNR. Not used for all other data types. So, fixed pt types will have one more field in the element definition.
- offset = offset in bytes into the payload data where the element is located. Value is decimal number
- **length**= length in bytes of the data element sizes are listed in the type section for different types. Value is decimal number.
- **info**= string containing units of the engineering data (deg/sec, lbs, feet).

A429 XML Example File:

<a429icd >

```
<label name='paul_w203' label='203' sdi='XX' rate='50' >
```

<elem name='w203_label' mask='0xff' type='FT_UINT8' offset='3' length='1' />

<elem name='data' mask='0x1ffff800' type='FT_BNR' range='131072.000000' offset='0' length='4' info='range -131072..131072' />

</label>

```
<label name='paul_w204' label='204' sdi='XX' rate='50' >
```

<elem name='w204_label' mask='0xff' type='FT_UINT8' offset='3' length='1' />

<elem name='data1' mask='0x1ffff800' type='FT_BNR' range='13.0' offset='0' length='4' info='range -13.0..13.0' />

</a429icd>



8.4 A429 Bus Data File.

This section describes the file format for the packet-a429_buses.dat file.

The packet-a429_buses.dat configuration file is used to provide a429 bus definition data to the analyzer. These values are correlated with the packet name described in the packet-a429.dat file. The packet name will consist of the A429 bus name with a "_w<label>" appended to the end. Examples include paul_a429_bus_w132, bill_a429_bus_w376.

This will allow the user to select a specific bus to be received on a selected receive channel in the A429 option notebook page.

- ➤ <A429 bus name>
- > ...
- > <EOF>

8.5 A429 Equipment ID Data File.

This section describes the file format for the packet-eqid.dat file.

The a429-eqid.xml configuration file is used to provide a429 equipment ID definition data to the analyzer. The format of the a429-eqid.xml file is same as packet-a429.xml with the label defition containing the added field of "EQ_ID". This file will be provided by the Goebel Company as part of a standard installation.

Each equipment ID decimal value corresponds to avionics unit/black box. Standard labels are defined for each equipment ID. These equipment Ids data strings are correlated with the packet name described in the a429-eqid.dat file. The packet name will consist of the A429 equipment id name with a "_w<label>" appended to the end. Examples include flight_control_computer_w132, flight_management_computer_w376.

This will allow the user to select a specific equipment id to be received on a selected receive channel in the A429 option notebook page.

8.6 P2P XML Data File.

This section describes the file format for the ~user/.ethereal.p2p/packet-p2p.xml file.

The following describes the format of the configuration file used to provide p2p data definition to the user. Each label definition consists a <label definition> and a list (1..N) of <element definitions>. The <label definition> defines the high level information about the label. The <element definition> defines information about a specific data element of the packet. The file consists of these types in the following repeated format: label definition and N element definitions. This will continue until the end of the file.

8.6.1 Label Data.

- **name** = name description of packet...string of characters...256 max length.
- **label** = The label number. Format is an 16bit hex value.
- **length** = length of packet in bytes.
- **rate** = rate of label in milliseconds.

8.6.2 Element Data.

- **name** = name description of element...string of characters...256 max length
- mask = mask used to retrieve data not located on a byte boundary. 0 should be used if the data type uses all bits in the type. Value is hexadecimal number. Examples are 0xffff0000 (upper 16bits of 4 byte word) 0xfffc0, 0x7ffc000, Mask has no meaning for the following: FT_FLOAT,FT_DOUBLE ©*The Goebel Company*



 \approx **<type>** = element type.

The following types are supported for P2P:

types	mask used	description
FT_FLOAT	no	4 bytes floating point number.
FT_DOUBLE	no	8-byte floating point number.
FT_UINT16	yes	2-byte unsigned value.
FT_INT16	yes	2-byte signed (2's complement) value.
FT_UINT32	yes	4-byte unsigned value.
FT_HEX	yes	2-byte unsigned value displayed as hexadecimal.
FT_INT32	yes	4-byte signed (2's complement) value.
FT_BOOLEAN	yes	1 bit boolean contain in 2 byte value.
FT_BNR	yes	2-byte unsigned integer value.
FT_UBNR	yes	2 byte scaled integer value.
FT_UINT8	yes	1 byte unsigned value.

• **range**= float value...max range of scaling for FT_BNR,FT_UBNR. Not used for all other data types. So, fixed pt types will have one more field in the element definition.

- offset= offset in bytes into the payload data where the element is located. Value is decimal number
- **length**= length in bytes of the data element sizes are listed in the type section for different types. Value is decimal number.
- **info**= string containing units of the engineering data (deg/sec, lbs, feet).

P2P XML Example File:

<p2picd>

```
<packet name='packet1' label='0x1f10' rate='10' length='86' >
```

```
<elem name='data1' mask='0xff' type='FT_UINT16' offset='0' length='2' />
```

```
<elem name='data2' mask='0xff00' type='FT_UINT16' offset='0' length='2' />
```

```
<elem name='data3' mask='0xff00' type='FT_UINT16' offset='2' length='2' />
```

```
<elem name='data4' mask='0xffff' type='FT_UINT16' offset='4' length='2' />
```

```
<elem name='data5' mask='0xffff' type='FT_UINT16' offset='6' length='2' />
```

</packet>

</p2picd>

8.7 M1553 XML Data File.

This section describes the file format for the ~user/.ethereal.p2p/packet-m1553.xml file.

The following describes the format of the configuration file used to provide m1553 data definition to the user. Each packet definition consists a <packet definition> and a list (1..N) of <element definitions>. The <packet definition> defines the high level information about the packet. The <element definition> defines information about a specific data element of the packet. The file consists of these types in the following repeated format: packet definition and N pkt element definitions. This will continue until the end of the file.

8.7.1 Label Data.

GOEBEL

- **name** = name description of packet...string of characters...256 max length.
- **rt_addr** = remote terminal address value.
- **length** = subaddress value.
- **dir** = direction of the packet either 'rx' or 'tx'.
- **rate** = rate of label in milliseconds.
- **length** = length of the packet in bytes.

8.7.2 Element Data.

- **name** = name description of element...string of characters...256 max length
- mask = mask used to retrieve data not located on a byte boundary. 0 should be used if the data type uses all bits in the type. Value is hexadecimal number. Examples are 0xffff0000 (upper 16bits of 4 byte word) 0xfffc0, 0x7ffc000, Mask has no meaning for the following: FT_FLOAT,FT_DOUBLE
- \approx **<type>** = element type.

types	mask used	description
FT_FLOAT	no	4 bytes floating point number.
FT_DOUBLE	no	8-byte floating point number.
FT_UINT16	yes	2-byte unsigned value.
FT_INT16	yes	2-byte signed (2's complement) value.
FT_UINT32	yes	4-byte unsigned value.
FT_HEX	yes	2-byte unsigned value displayed as hexadecimal.
FT_INT32	yes	4-byte signed (2's complement) value.
FT_BOOLEAN	yes	1 bit boolean contain in 2 byte value.
FT_BNR	yes	2-byte unsigned integer value.
FT_UBNR	yes	2 byte scaled integer value.
FT_UINT8	yes	1 byte unsigned value.

The following types are supported for M1553:

- **range**= float value...max range of scaling for FT_BNR,FT_UBNR. Not used for all other data types. So, fixed pt types will have one more field in the element definition.
- offset= offset in bytes into the payload data where the element is located. Value is decimal number
- **length**= length in bytes of the data element sizes are listed in the type section for different types. Value is decimal number.
- **info**= string containing units of the engineering data (deg/sec, lbs, feet).

M1553 XML Example File:

<m1553icd>

<packet name='subaddr_15a_rx' rt_addr='16' subaddr='15' dir='rx' rate='1000' length='64' >

<elem name='paul1' mask='0x1' type='FT_BOOLEAN' offset='0' length='2' />



<elem name='paul2' mask='0x2' type='FT_BOOLEAN' offset='0' length='2' />

</packet>

</m1553icd>

9. Frequently Asked Questions

- → Can I talk with a live person (not in a foreign country) about questions I may have? YES, the Goebel Company (support@goebeletc.com, or 206.601.6010) will be happy to answer any question that the user may have.
- \rightarrow How do I start the execution of the analyzer?

On the Desktop, there will be icons for each interface and program. The title should be of the form Goebelyzer <interface> <program> where interface is afdx,a429,p2p,m1553,alldev...

 \rightarrow Where is the analyzer ICD configuration files located?

The ICDs are organized in directories under the */home/user/icd* directory. Each ICD corresponds to an aircraft and data creation date.

 \rightarrow How do I select an ICD?

The "ICD Selection" icon on the desktop can be used to select the data defines that the analyzer will load on startup.

 \rightarrow Where are the analyzer configuration information files located?

The files are located in the home directory of user at */home/user/.ethereal.*<*interface*> where the interface is a429,a664,p2p,can, or alldev.

→ How can I view the boolean state of bit=15 of long word=20 (80byte offset) in an AFDX packet with vl=1212 and destination port=4545 on a periodic basis?

1. The user must have an entry in the packet-afdx.xml configuration file for this data. This is described in section $\frac{\#8.1 \text{AFDX Data File.}|\text{outline}|$. It would be helpful if all data for a program was defined in the config file. But as an example, the following entry would need to be in the packet-afdx.xml file.

```
<packet name='pkt=-fms_10ms' port='4545' vl='1212' protocol='none' rate='1000' length='20' >
```

<elem name='boolean_value' mask='0x8000' type='FT_BOOLEAN' offset='80' length='4' />

</packet>

- 2. Start execution of the analyzer.
- 3. Depress the Options button to display the Options dialog. Select AFDX RTD Notebook. Double click the entry pkt_fmt_10ms->boolean_value. Start Capture. This is described in section <u>#5.7Real-Time Display Option Pages.</u>
- 4. Select the AFDX RTD Notebook page. The value will be updated on a 1 sec interval when the packet is received. This is described in section <u>#6.3Data Display Page.</u>

→ How can I capture data only from udp destination ports 2323 and 3434 on an AFDX device?

1. Start the analyzer with the AFDX device selected.

2. A capture filter of "udp dst port 2323 or udp dst port 3434" could be used. This is described in section $\frac{\#5.1.5Capture Frame}{1000}$. Then the capture button is depressed to start the capture.



3. OR perform a normal capture and enter a display filter (display filter entry is located below the menu) of udp.dstport == 2323 or udp.dstport == 3434. Display filters are to eliminate packets from the capture display that do not return a True result to the filter expression. In this case, an udp dst port of 2323 or 3434. The display filter can be with all items defined in the configuration file. So, a user may be able to view unique payload information. Packets with airspeed < 200, or altitude > 2500 or both using and/or/not boolean terms to combine boolean conditions into a filter with multiple conditions.

→ How can I capture all AFDX packets with a 0x12 at payload byte 4 after udp port 4545 is received with a 0x2323 at payload byte 10?

This is performed by entering a trigger entry of "udp port 4545 and udp[18:2] == 0x2323" and a capture filter of "udp[20:1] == 0x12". The afdx payload begins at 8 byte into the udp header. So, all afdx payload entries will be +8 bytes. Triggers and Capture Filters are described in section <u>#5.1.5Capture Frame</u>.