CANalyzer .A429

Product Information
Table of Contents

1 Introduction to CANalyzer ................................................................. 3
1.1 Features and Advantages ............................................................. 3
1.2 Application Areas ................................................................. 4
1.3 Bus Systems and Protocols ......................................................... 4
1.4 CANalyzer variants ................................................................. 4
1.5 System Requirements .............................................................. 4
1.6 Further Information ................................................................. 5
2 Functions ....................................................................................... 5
2.1 Database Support ....................................................................... 5
3 Analysis and Stimulation ............................................................... 5
3.1 Analysis Windows ...................................................................... 5
3.1.1 Measurement Setup ............................................................. 6
3.1.2 Trace Window ....................................................................... 7
3.1.3 Graphics Window ............................................................... 7
3.1.4 Data Window ....................................................................... 8
3.1.5 Bus Statistics Window .......................................................... 9
3.1.6 Write Window ...................................................................... 9
3.1.7 Interactive Generator ........................................................ 10
3.1.8 Triggers and Filters ............................................................ 10
3.1.9 Logging/Replay .................................................................. 11
3.2 Offline Evaluation ...................................................................... 11
3.3 Export Functions ....................................................................... 11
3.4 Integrated Desktops ................................................................. 11
4 Programming .................................................................................. 11
4.1 CAPL Interface ........................................................................ 11
4.1.1 C-Like Syntax ..................................................................... 11
4.1.2 Event-Oriented Control ...................................................... 12
4.1.3 Symbolic Access ............................................................... 12
4.1.4 Application-Specific Language Extensions ....................... 12
4.1.5 CAPL Browser ............................................................... 13
4.2 Visual Sequencer ....................................................................... 14
5 Panels .......................................................................................... 14
6 Software Interfaces ........................................................................ 15
6.1 Further Information .................................................................. 15
7 Hardware Interfaces ....................................................................... 16
8 Training ......................................................................................... 16

V2.0 05/2017
Valid for CANalyzer .A429 as of version 10.0

This document presents the CANalyzer .A429 application areas of analysis, stimulation and their individual functions. The document contains a brief overview of programming in CANalyzer .A429, supplemental programs and hardware interfaces.

Product information and technical data on CANalyzer in general are available in a separate document.
1 Introduction to CANalyzer

CANalyzer is a universal analysis tool for data buses in aircrafts. In addition to observing and analyzing, it is also quite easy to supplement the data traffic in the supported bus systems.

The tools powerful basic functions and user programmability cover all needs from simple network analysis to focused debugging of complex problems.

Figure 1: Standard CANalyzer configuration for analysis of a ARINC 429 system

1.1 Features and Advantages

User operation of CANalyzer is very intuitive and is based on a graphic block diagram that depicts the data flow from the bus, over the PC interface and to the various evaluation windows of the PC screen and to the logging file. The system is parameterized in this block diagram. Function blocks such as Filter, Generators or Replay Blocks can also be placed and configured there.
1.2 Application Areas

CANalyzer is a tool that optimally covers all application areas from a simple network analysis to a high-performance analysis and emulation system. For example, a Replay Block may be placed in the send branch to replay the previously logged data traffic of a functional bus system. Portions of the data traffic may be hidden by adding a downstream filter. This creates a perfect test environment for an ECU in the laboratory.

Other examples of CANalyzer applications are:

- Emulate bus modules
- Create simple test sequences
- Gateway between two buses with ability to manipulate messages
- Link different trigger conditions for logging and displaying the bus traffic
- User-specific online evaluation with clear text message

1.3 Bus Systems and Protocols

In CANalyzer, different CANalyzer options are available for the various bus systems and CAN-based protocols; these options may be used in any combination.

CANalyzer supports the following bus systems: CAN, CAN FD, LIN, MOST, FlexRay, J1708, Ethernet, WLAN, ARINC 429 and AFDX®.

Option CAN is the basis for these supported CAN-based protocols: J1939, CANopen, CANaerospace, ARINC 825.

You will find detailed information on the options in separate product information documents.

1.4 CANalyzer variants

- CANalyzer fun: The “fundamental” variant is ideal for simple analysis purposes, and it provides all standard interactive functions required for them. It does not offer programmability or user control panels.
- CANalyzer exp: The “expert” variant is ideal for nearly all standard applications, and it provides all functions and extensions without limitation. The only feature lacking in this variant is the ability to create and execute CAPL programs.
- CANalyzer pro: The “professional” variant provides all functions and extensions without limitation. It supports all use cases, ranging from simple observation of the bus traffic to complex analysis, stimulation and testing of heterogeneous systems.

1.5 System Requirements

<table>
<thead>
<tr>
<th>Component</th>
<th>Recommended</th>
<th>Minimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>Intel Core i5</td>
<td>3,0 GHz</td>
</tr>
<tr>
<td>Memory (RAM)</td>
<td>4 GB</td>
<td>1 GB</td>
</tr>
<tr>
<td>Hard drive capacity</td>
<td>≥ 2.0 GB (depending on options used and required operating system components)</td>
<td></td>
</tr>
<tr>
<td>Screen resolution</td>
<td>1280 x 1024</td>
<td>1024 x 768</td>
</tr>
<tr>
<td>Graphics card</td>
<td>DirectX 9.0c or higher and Shader Model 1.1 or higher</td>
<td></td>
</tr>
<tr>
<td>Operating system</td>
<td>Windows 10/8.1/8/7/Vista (Vista nur 32 Bit)</td>
<td></td>
</tr>
</tbody>
</table>

1 AFDX® is an Airbus registered trademark
1.6 Further Information

- **Vector Download Center**
  - Demo versions
    Various demo versions are available on the web for CANalyzer. They contain sample configurations for the various application areas as well as a detailed online help function, in which all CANalyzer functions are described.
  - Application notes
    In the following chapters, we refer to additional application notes that offer in-depth information on the individual application areas.
- **CANalyzer Feature Matrix**
  More information on variants, channels and bus system support is presented in the feature matrix.

2 Functions
The product's basic functions give you a broad range of use options; they include:

- Tracing of the bus data traffic
- Graphic and textual display of signal values
- Interactive sending of predefined messages
- Sending out logged messages
- Statistics on messages, on bus utilization and bus disturbances
- Logging of messages for replay or for offline evaluation
- Generation of bus disturbances on the message level
- Intuitive user interface with flexible docking concept and user-friendly menu structures
- Support of new Vector bus hardware VN0601 (4 TX & 4 RX ARINC 429 channels and I/O)
- User-programmability with CAPL (CANalyzer pro)

2.1 Database Support
CANalyzer supports system descriptions via the DBC format for CAN, ARINC 429 and AFDX®. Information of a database can be symbolically displayed and used in CANalyzer.

3 Analysis and Stimulation
The basis for analysis in CANalyzer is the data flow from the data source to its display or logging. The data may also be processed. For example, filters can be integrated that define which data should be explicitly considered in an analysis and which data should not. The data traffic can also be influenced by various stimulation options.

**Highlights**

- Easy to configure the analysis window by drag & drop. For example, this method can be used to copy or move messages or signals from one analysis window to another.
- For a multifunctional analysis, one type of window (e.g. Graphics Window) can be integrated multiple times in the data flow, which enables parallel analysis.
- Easy start and stop logging directly from the status bar.

3.1 Analysis Windows
CANalyzer supplies the user with windows and blocks such as those described below.
3.1.1 Measurement Setup

The data flow is graphically represented and configured in the Measurement Setup.

> Define data source (online/offline)
  The simulated bus or the real bus connected via the hardware (e.g. CANcardXL) serves as the online data source. A file with logged data serves as the offline data source.

> Insert analysis windows
  Data can be shown differently in the individual windows depending on analysis requirements, e.g. to graphically represent signal waveforms or to display signal values.

> Insert CAPL program nodes
  A CAPL program node can be used for such tasks as filtering data or implementing various arithmetic operations.

> Insert filters
  Filters can be used to obtain a more understandable representation of the data; they define which data should be passed and which should be explicitly blocked. Filters may be active during or after the measurement, and the objects filtered may range from individual signals to the channels of an entire bus system.

> Insert trigger conditions
  Like filters, trigger conditions can also be used to reduce data. Triggers are specifically configured as a reaction to bus events and can be combined with one another.

> Log data
  For an analysis after the measurement, data can be logged in a logging file that can later be reused as an offline data source and replayed.

![Measurement Setup with online data source](image)
3.1.2 Trace Window

Bus activities such as the sending or receiving of messages are listed in the Trace Window. Individual signal values may be displayed for each message. Functions such as those listed below are available for analyzing the data:

- **Insert filters**
  There are various types of filters in the Trace Window. They can be used to reduce the amount of data displayed, and data can even be deleted from the data stream.

- **Hide unchanged data**
  To improve ease of viewing, data that does not change is slowly faded or removed entirely from the screen.

- **Color events**
  Important events and messages can be highlighted in color.

- **Set markers**
  Markers can be set to identify and quickly find events. The marker is assigned to an event and therefore to its time stamp as well. The set markers can also be displayed in other analysis windows.

- **Show statistics**
  Various aspects of messages/signals, including their values, can be displayed in different views in detail. Differences between the time stamps or signal values may also be calculated.

- **Log data**
  It is possible to export some or all of the Trace Window contents. Files that have already been exported can be converted to a different format afterwards, e.g. to further process the same dataset in different programs.

![Figure 3: Trace Window with list of signals opened](image)

3.1.3 Graphics Window

The Graphics Window is used to graphically display the values of signals and environment data as curves. Listed below are some of the functions available for measurement and evaluation of these curves:

- **Show measurement markers/difference markers**
  Measurement or difference markers can be used to perform absolute or relative analyses of measurement values. The measurement marker can be synchronized to the Trace Window display.

- **Set markers**
  Markers can be set to identify and quickly find events. The marker is assigned to one event and therefore to its time stamp as well. The set markers can also be displayed in other analysis windows.

- **Show measurement columns**
  In the legend, global or local minima and maxima may be shown for each signal, or Y-differences between signals of the same type can be displayed.

- **Show statistics**
  Statistical data such as minimum, maximum, mean value and standard deviation can be compiled for selected signals or all signals of the Graphics Window.

- **Log data**
  Signals of the Graphics Windows can be logged automatically or manually during the measurement. This involves extracting the signals from the messages and saving them in binary form in signal-based MDF files.
In the Graphics Window, the entire signal waveform or just a visible section of the signal waveform can be saved to a file.

![Graphics Window](image)

**Figure 4:** Graphics window used to display the values transmitted in the messages over a time axis.

### 3.1.4 Data Window

The Data Window is used to display the values of signals and system variables in different types of representation.

- **Show values**
  The data may be displayed as raw or symbolic values. Other display variants are scientific notation and the display of global and local min/max values.

- **Log data**
  Signals can be logged during the measurement and saved to MDF binary format.

![Data Window](image)

**Figure 5:** Data Window with different representation types for incoming values.
3.1.5 Bus Statistics Window

The Statistics Window shows statistical information about bus activities (ARINC429) during a measurement. This includes such information as bus load, counters/rates for ARINC Words and errors.

![Bus Statistics for ARINC429](image)

Certain ARINC 429 statistics can be evaluated in analysis windows such as the Graphics Window, or in program nodes via automatically defined statistical system variables. These system variables are available for each configured network channel and are updated independently of the Bus Statistics Window.

3.1.6 Write Window

The Write Window displays system messages and user-specific outputs from CAPL programs.

- **Configure output**
  The Write Window offers different views for filtering system messages according to their source.

- **Log output**
  The Write Window output may be saved to a file or copied to the clipboard as text and be copied to other Windows applications from there.

- **Status Display**
  Informs about new unread warning and error messages in the Write Window.

![Write Window with system messages and CAPL outputs](image)
3.1.7 Interactive Generator

The Interactive Generator (IG) can be used to set signal values, define signal waveforms and send specific messages. This gives users an easy way to stimulate the bus.

> Send messages

The messages that are configured in the send list can be sent periodically when a specific screen button is pressed or by pressing a predefined keyboard key.

> Change signal values

In the Interactive Generator, individual signal values can be modified in the signal list, and signal waveforms (signal curves) may be defined with the integrated Signal Generator. These signal values can then easily be sent on the bus in the associated message, e.g. to check the reaction of an ECU.

![Interactive Generator with configured messages and their signals](image)

3.1.8 Triggers and Filters

Triggers and filters can react to specific bus events, and they serve to reduce the amount of displayed or logged data. Examples of trigger conditions are error states, messages, signals and signal changes (edges). Complex system states can be triggered by forming groups and linking them with logical operators.

> Filters in the Measurement Setup

Various filters are available in the Measurement Setup that can be used to define which data should be passed to the specific analysis windows and/or which data should be explicitly blocked. All filters can be used as Stop and Pass Filters.

> Triggers in the Measurement Setup

Different trigger conditions can be used in the Measurement Setup to influence the logging of data to a logging file.

> Filters in the Trace Window

In the Trace Window, data can be reduced for analysis both during and after the measurement using various filters. For example, you could set predefined filters to filter for individual signals and signal values or set different column filters.
3.1.9 Logging/Replay

Data can be logged in CANalyzer and replayed later in a post-measurement analysis.

> Replay
The Replay Block can be used to replay measurement sequences that have been logged in a logging file. The messages contained in the logging file are introduced into the data flow.

> Logging
The Logging block can log the bus traffic in the BLF and ASCII formats. The logged data can then be replayed in offline mode or with a Replay Block.

3.2 Offline Evaluation

The logged message traffic can be evaluated at a later point in time, in offline mode, using all CANalyzer functions. This makes it possible, for example, to conveniently analyze extensive recordings of a measurement afterwards in the workplace. Collections of logging files can also be specified, which are then chronologically replayed into the analysis based on their time stamps. In addition, logging files can be directly imported into the Trace Window for quick analyses.

3.3 Export Functions

Export functions for logging and the Trace and Graphics Windows enables conversion of logged files or window contents to other file formats, e.g. to *.csv. In export of the logging files, individual signal information is extracted from the logged, message-based data.

3.4 Integrated Desktops

CANalyzer users often have many analysis windows opened. That is why CANalyzer offers an integrated desktop concept, in which users can switch between any number of virtual desktops on tabbed pages. Opened windows can be distributed to desktops, and information can be sorted by work processes or subject matter.

4 Programming

4.1 CAPL Interface

The CAPL (Communication Access Programming Language) programming language extends the functional scope of CANalyzer tremendously. Special characteristics of CAPL include:

> Can be learned quickly since it is based on the C programming language
> Fully event-controlled in its operation. CANalyzer takes over control from the user.
> Supports symbolic access to all database information such as messages and signals. Signal values can be used directly in their physical form.
> The language has been extended with special functions for quick implementation of problem solutions in various use scenarios
> Flexible extension by external libraries

4.1.1 C-Like Syntax

The usual scalar data types and arrays are provided (1, 2, 4 and 8 byte long whole number types as well as an 8 byte long floating point type). Assignments, arithmetic operators and loop flow control conform to C-syntax.

```c
myFunction {
    int counter;
    for ( counter = 0; counter < 8; counter++ ) {
        doSomethingWithCounter ( counter );
    }
}
```
4.1.2 Event-Oriented Control

CAPL is an event-controlled programming language. In contrast to C, special predefined event handlers (event procedures) are available in CAPL, which are always executed whenever a specific event occurs (if time controlled then triggered by the hardware or internal to CANalyzer).

Here are just a few examples of these event handlers:

<table>
<thead>
<tr>
<th>Event Handler</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>On timer seconds cycle</td>
<td>Time controlled</td>
</tr>
<tr>
<td>On a429word FlapStatus</td>
<td>Message input or output</td>
</tr>
<tr>
<td>On signal update</td>
<td>Rewrites signal value</td>
</tr>
<tr>
<td>On sysvar</td>
<td>Modifies system variable</td>
</tr>
<tr>
<td>On a429worderror</td>
<td>Detects ARINC 429 errors</td>
</tr>
</tbody>
</table>

4.1.3 Symbolic Access

Signal values are generally accessed as physical values, regardless of the scaling of message transmission. This is set in the database and is taken from there.

- Physical access to signal values (only CANoe):
  // Definition of the representation in the database
  $EnergyMgmt::BatteryVoltage = 14.1;

- Access to raw value of a signal (only CANoe):
  // 8 to 18 Volt with 12bit resolution, without range check
  $EnergyMgmt::BatteryVoltage.raw = (14.1 - 8) / (18 - 8) * 4096;

- Access on a message base:
  a429word EnergyMgmt msg;
  // Most significant bytes Intel coded
  // with 12bit only the lower 4 bits are used
  // details are specified via database
  msg.BatteryVoltage = 14.1;
  output(msg);

4.1.4 Application-Specific Language Extensions

For all use cases of CANalyzer there are numerous functions that are specially tailored to everyday problems related to these topics.

- Analysis
  CAPL can be used in the analysis of measurement results – either online and offline. One simple task might be to count the occurrences of a specific event or perform computations with the contents of certain signals.
  On a429word Brake {
    long tempCounter = 0;
    $BRECounter++;
    // Take only the last 100000 values into account
    tmpCounter = $BRECounter;
    if (BRECounter > 100000)
      tempCounter = 100000;
    @AveragePressue = @AveragePressure * tempCounter + $Brake::Pressure;
    output ( this );
  }

- Stimulation
  You can also use CAPL to generate messages to stimulate ECUs. This relieves the developer of routine work tasks. Signals and messages of the bus are defined in the database (e.g. in DBC file).
4.1.5  CAPL Browser

The functionality of the CAPL Browser goes beyond that of an editor for CAPL programs. It offers functions of an advanced development environment, such as:

> Code auto-completion and syntax checking while writing code
> Configurable syntax highlighting
> Syntax-sensitive tabs
> Folding function blocks and functional references in a tree view for quicker navigation
> Find and replace in individual or multiple files
> Online help with references to functions
> Calling of the compiler with preselected source text lines in case of error
> Hierarchical function list with search function for direct copying into the source text

Objects of the CANalyzer database are available in the CAPL Browser as well, and they are also displayed in a tree view. The following database contents can be accessed in what is known as the Symbol Explorer:

> Network symbols such as nodes, messages and signals
> System variables that are used CANalyzer-wide

![Figure 9: CAPL Browser with opened CAPL program, contained event procedures and network symbols from the database](image)
4.2 Visual Sequencer

This is a quick way to graphically configure flow sequences without requiring programming. Variables and signals may be set within such sequences. Frames and diagnostic commands can also be sent. In addition, it is possible to wait for certain events, check values or define repetitions with control structures (repeat...until). These sequences are therefore ideal for simple tests of heterogeneous systems or for stimulating ECUs.

Figure 10: CANalyzer Visual Sequencer for creating test and stimulation sequences. Makes it easy to select commands and database objects with auto-complete support and to display detailed database information.

5 Panels

Panels are graphical elements that can be used to modify signal and variable values and display them with controls such as sliders or pointer instruments. Panels are used to display the analysis data from CAPL programs, for example, or to control CAPL programs and transfer values into the programs.

The Panel Designer can be used to conveniently create such panels. For example, it is easy to link a symbol to a control by drag & drop. The individual panels and controls are configured via the constantly open Properties Window, and a whole series of useful alignment functions ensure an optimal layout.

Figure 11: User-defined panels for displaying signal and variable values
6 Software Interfaces

The integrated COM Server (Component Object Model) enables control of the measurement sequence by external applications and convenient data exchange with standard software, e.g. for measurement data analysis or in-depth evaluation of the observed bus traffic. Frequently used programming/script languages here are Visual Basic or Visual Basic for Applications. C++/C# are also frequently used. The functionality that CANalyzer offers over the COM interface covers such aspects as:

> Control of the simulation, starting and stopping the measurement
> Loading existing configurations, generating new configurations, adding databases and blocks to the Tx branch
> Access to signals and system variables, access to CAPL functions, compiling of CAPL nodes

**Visual Basic Script example for starting the measurement:**

```vbscript
set app = createobject( "canalyzer.application")
set measurement = app.measurement
measurement.start
set app = nothing
```

**Visual Basic Script example for opening a configuration:**

```vbscript
set app = createobject( "canalyzer.application")
app.open "D:\PathToMyConfig\myconfig.cfg"
set app = nothing
```

6.1 Further Information

A general introduction to COM Server functionality of CANoe/CANalyzer is described in application note AN-AND-1-117_CANalyzer_CANoe_as_a_COM_server. Fundamental technical considerations and options are presented, and they are illustrated as Microsoft Visual Basic examples.
## 7 Hardware Interfaces

CANalyzer supports all bus system interfaces available from Vector. Optimal bus access is possible for every use case thanks to a large selection of different PC interfaces (PCMCIA, USB 2.0, PCI, PCI-Express, PXI) and bus transceivers. For ARINC 429, the VN0601 interface is used.

**Figure 12:** Overview of Vector hardware

<table>
<thead>
<tr>
<th>Hardware Interface</th>
<th>PC/Notebook</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCI/PCIe/PXI</td>
<td>PCMCIA</td>
</tr>
<tr>
<td>CANboardXL</td>
<td>USB</td>
</tr>
<tr>
<td>VN1600</td>
<td>ExpressCard</td>
</tr>
<tr>
<td>VN1630A VN1640A</td>
<td>PCMCIA</td>
</tr>
<tr>
<td>VN1611</td>
<td></td>
</tr>
<tr>
<td>VN1610</td>
<td></td>
</tr>
<tr>
<td>CANpiggy</td>
<td></td>
</tr>
<tr>
<td>TJA1051</td>
<td></td>
</tr>
<tr>
<td>TWINcab</td>
<td></td>
</tr>
<tr>
<td>CANcab</td>
<td></td>
</tr>
<tr>
<td>84440980</td>
<td></td>
</tr>
<tr>
<td>CANcardXL</td>
<td></td>
</tr>
<tr>
<td>CANcardXL</td>
<td></td>
</tr>
<tr>
<td>VN0601</td>
<td></td>
</tr>
<tr>
<td>VN5610</td>
<td></td>
</tr>
<tr>
<td>ARINC 429</td>
<td></td>
</tr>
<tr>
<td>AFDX®/Ethernet</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 13:** The compact VN0601 USB interface for the ARINC 429 bus system can operate up to four RX and four TX channels simultaneously.

## 8 Training

As part of our training program, we offer various classes and workshops on CANalyzer in our classrooms at Vector and on-site at our customers.

You will find more information on individual training courses and a schedule online at: [www.vector-academy.com](http://www.vector-academy.com)
Get More Information

Visit our website for:
> News
> Products
> Demo software
> Support
> Training classes
> Addresses

www.vector.com