Complex Electronic Systems Developer Tool dBricks

Overview

dBr.0003.07

Table of Contents

1 List of Abbreviations, Terms and Definitions

Abbreviation/Term	**Full Name/Definition**
AC	Aircraft
AE	Avionic Equipment / Avionics
AE Component	System element forming part of AE systems
API	Application Programming Interface
CAD	Computer Aided Design (system)
CDU	Create, Delete, Update – three basic functions for working with objects
DB	Database
DIM	Device Interface Module
EI	Export/Import – one of the main mechanisms for data group add/edit
FHA	Fault Hazard Analysis
FIM	Function Interface Module
HIRF	High-Intensity Radiated Fields
ICD	Interface Control Document
Integration	Process of system integration (combining) of subsystem devices into one system and ensuring functioning of subsystems together as a single system
PL	Part List
SaaS	Software as a Service
SW	Software
WDD	Working design documentation
WS	Wiring System

2 Problem Statement

Over the last decades, innovations in avionics have enabled unprecedented increase in flight safety, pilot convenience and timely departures. However, the downside of smart high-tech on-board systems is design complexity.

Airborne system engineers are facing the daunting task of integrating a huge number of components (which already are complex per se) into a synergy that would provide seamless operating. The task is further complicated by the need to meet the highest safety standards and comply with multiple industry standards.

Design of complex on-board systems cannot do without drafting numerous documents such as:

a) Interface control documents,
b) Structural diagrams,
c) Connection diagrams,
d) Wiring lists and schematics,
e) Interconnection diagrams,
f) Software specifications and requirements,
g) Network switch configuration tables,
h) Software design specifications.

As a rule, these are long documents containing project implementation details – from wires to digital bus parameters refresh rates. For the resulting equipment system to operate fault-free and to be certified by aviation authorities, all documents should be consistent and comply with industry standards, regulatory documents, aircraft, and avionics requirements.

The challenge of documentation consistency is a consequence of modern avionics complexity in general and the fact that a dozen of departments within the aircraft manufacturer company and dozens of third-party OEMs are involved in such documentation development. Organizing efficient interaction between all parties involved is a non-trivial task both from the technical and organizational point of view.

Aircraft (AC) development is governed by AC technical and functional requirements, AC certification requirements (certification basis), applicable industry and AC manufacturer internal standards. All these requirements must be consistent, their traceability must be controlled (if possible, by automatic means).

The initial design data alongside with requirements and regulations include large amounts of information shared between the AC manufacturer and OEMs. Such initial data can change at different design, testing and certification stages, and even during the aircraft operation. Controlling coordinated changes in all systems and equipment affected by the initial data changes is one of the most time-consuming tasks of aviation equipment development.

Unfortunately, progress in the field of avionics as such has not touched on the avionic equipment (AE) development tools, and engineers are compelled to conduct their projects in an old-fashioned way using spreadsheet and text editors such as Microsoft Excel and Microsoft Word. These editors, albeit convenient and powerful document development tools, are not intended for airborne equipment design, since they do not allow to control integrity, introduce changes simultaneously throughout project implementation or provide multi-user access to the project developed, etc.

The lack of specialized tools results in the need to perform a wide range of tasks involving thorough check and manual configuration management for documents being developed and reviewed, which can lead to delays and/or errors.

The existing problem in avionic equipment development processes is recognized by most industry experts, however the market has not offered automation tools for the above processes so far.

3 Purpose

The purpose of dBricks tool is to:

1. Reduce development effort for complex electronic systems in general and avionics equipment in particular,
2. Minimize testing and commissioning costs for systems developed,
3. Improve quality of documents developed.

The tool's purpose is achieved through automation of the following processes:

1. Configuration management of on-board equipment,
2. Ensuring prompt cooperation between all parties within the development process
3. Development of AE structural diagrams,
4. Development of wiring lists,
5. Development of interconnections diagrams,
6. Development of connection diagrams,
7. Development of interface control documents,
8. Development of flight software (SW) design specifications,
9. Development of data networks configuration,
10. Development of design documentation (DD) for test bench and flight simulator wiring,
11. Development of configuration files for on-board measuring and maintenance systems,
12. Development of configuration files for simulation IOs as part of a test bench or a flight simulator,
13. Ensuring compliance of input data with the requirements of national and industry standards, regulatory documents, including developer's internal documents,
14. Analysis of equipment compliance with the relevant requirements, including but not limited to the following checks:
    a) Functions source data coverage analysis,
    b) Fault hazard analysis for equipment and wiring failures,
    c) Analysis of bus load and delays in digital data channels,
    d) Analysis of computational resource loading.

4 Development History

dBricks tool is designed based on vast experience gained from the development of all modern civil airliner projects in the Russian Federation.

The concept of using AE design automation tools emerged during the development of digital flight and navigation equipment for IL-96, Tu-204 and IL-114 aircrafts. Around the same time, the first basic design tools were developed for interface control documents, tables and connection diagrams.

Later on, the tools were further developed and expanded for automated generation of aircraft circuit diagrams, connection diagrams, interface control documents and other documents used within manufacturing, installation and testing of SuperJet-100 aircraft (RRJ-95) wiring and AE. With the help of these tools, data were also obtained for automated assessment of AE resistance against external exposure factors including HIRF, and Fault hazard analysis.

Upon completion of SuperJet-100 development, a further step was made to connect disparate tools into a unified system with enhanced functionality. Today, the results of these works are widely used in the development and testing of avionic equipment for MS-21 aircraft.

dBricks is the ideological successor to all the aforementioned attempts to design an automation tool, which has incorporated all gained experience and knowledge.

5 Tool Description

5.1 General Description

dBricks system is comprised of normalized database, as well as data input, output, and modification tools, including graphical user interface.

The tool is based on client-server technology. The server side includes network data storage and a server module. The server module interacts with the client module for processing input data and prepares output data for visualization in the client module.

The client module is a graphical shell intended for displaying data received from server and providing convenient access to data. dBricks client part is implemented as a web page accessed through a web browser (e.g. Google Chrome).

Normalized database assumes that all data (objects) related to systems and links between them are stored as a single object – from measuring units up to links to connection of communicating systems. Other objects can only refer to this unique object, not duplicate it in any part. This approach enables traceability of changes to objects resulting from changes to other objects to ensure that output is up to date.

5.2 Data Storage Subsystem Architecture

Following the concept of data standardization, dBricks tool considers data describing avionic equipment not as a whole, but as a set of objects consisting of other objects and links between them. The objects that make up a project, in turn also consist of smaller objects and links between them. A standardized approach completely eliminates the need for repeated input and storage of same-type data – among other things, when using the same device versions in different projects.

Here are three global object families:

1. Common objects: This group contains objects that describe basic concepts such as bus types, connector types, bus types, data types, units, etc. Description of these objects does not contain references to other objects, however complex structure objects can refer to them.
2. Device templates: Device templates are composite objects that describe structure and properties of device types. Usually, they describe all devices with the same part number (P/N). Device Template description can contain the following components:
    a) Device description, including name, P/N, description, weight, size, manufacturer, etc.,
    b) List of device connectors, including connector names, block (normally receptical) and cable (normally plug) parts of the connectors , connector pin layout, materials, manufacturer, etc.,
    c) List of device ports: Device port allows to link a given device to other devices, e.g. it can be ARINC 429 output port. Port properties include name, type, description, etc.,
    d) List of functions that determine device purpose, indicating functional parameters required for device operation (received and/or transmitted by the device) and their definition principle,

e) List of individual software elements or partitions (acc. to ARINC 653 standard) implemented in the device with a list of input/output parameters,
f) Device port contents contain a detailed description of the transport layer and configurable port characteristics, e.g. such as ARINC 429 bus transmitted data (word list, refresh rates, parameter mapping to data bits, transmitted data characteristics, etc.). Port contents vary considerably depending on port type. Port contents can be mapped to functional parameters and/or input/output variables of software elements, allowing linking the data flow requirements at the logical level to the transmission method description.

3. Projects are top level objects providing general description of AE design. In the first approximation, any project is made up of devices instantiated from device templates and links between those devices. Device connection description includes:
    a) links between device ports which form data buses,
    b) links between devices' functional parameters that determine data flows at the logical level,
    c) Description of configuration of complex digital buses such as ARINC 664, in particular the requirements for addresses used, distribution of messages over virtual links (Virtual Link), etc.

5.3 Tool Multiuser Features

dBricks tool is based on client-server architecture, which allows the entire project team to use the tool at the same time. This is one of the main advantages of this tool. After tool deployment, latest project status information will always be available to each project stakeholder. Any changes made by one user automatically become available to everyone. Based on the Customer's need, the tool can be made available not only to company employees, but also to specialists of contractor companies. Information security and unauthorized access control are ensured by flexible access right control system. The system also enables logging of all user actions, which allows to find every change creator.

5.4 User Interface Approach

The tool is developed in close coordination with end-users (avionics developers). During development, dozens of user suggestions were accounted in the user interface. Main technical solutions have been tested on actual project data. Tools and system performance allow to work efficiently both with modern airliner projects comprised of hundreds of devices, tens of thousands of wires and hundreds of thousands of parameters, and with any project that involves electronic equipment.

For entering and editing large volumes of data, the tool offers bulk data input/editing functionality that allows to substantially reduce complexity of tasks.

5.5 Configuration Management

Within development of complex AE projects, it’s sometimes necessary to manage configuration of multiple project versions resulting from both variability of the project (e.g. implementation of options) and the need to track AE generations featuring different sets of requirements (test bench version, first flight version, certification version, etc.). dBricks uses AE configuration management mechanism that enables the following functions:

1. Create a baseline version of the device or project configuration template. Once created, the baseline version cannot be changed, which makes reports generated from such baseline version reliable for further work.
2. Create new versions of objects based on the baseline versions. In the tool, this process is called "unfreezing". During such unfreezing, the tool creates a new exact and editable

copy of the baseline version. At the same time, several variants of device or project templates can be created from each baseline version, which allows simultaneous development of several modifications.

3. Compare versions, generate lists of differences.

5.6 Data Validity Control

The tool allows to eliminate significant number of errors upon data input. The system allows to filter errors that violate the requirements of regulatory documents, industry standards and configurable project constraints. Errors that violate the requirements of regulations and standards include – not limited to – attempts to link different buses type or attempts to include data in the data exchange description that cannot be physically implemented. Errors that violate configurable project constraints may include violation of the naming convention or restrictions on random parameter combination with different data types.

5.7 Tool Export Features

dBricks with a populated database allows you to automate the following processes.

Note: Generally speaking, once the tool is deployed, the implementation of these processes is reduced to populating the database. Once the database is populated, the resulting documents are generated automatically.

5.7.1 Export of Structural Diagrams , Connection Diagrams , Wiring Lists, and Interconnections diagrams

The aforementioned diagrams are the main documents describing connection of AE devices to each other. dBricks makes diagram generation automatic. By pressing “Export”, you receive an end diagram file in Microsoft Visio or Microsoft Excel format (in the case of lists and tables).

Note: The format of tables and diagrams can be customized according to specific requirements.

5.7.2 ICD Export

ICD development is one of the fundamental tasks of the entire AE development process and one of the key purposes of dBricks tool. The tool allows you to export ICD implemented through the following interface types:

1. Discrete IOs,
2. Analog IOs incl. proximity sensor signals,
3. Linear Variable Differential Transformer (LVDT) signals,
4. ARINC 429,
5. ARINC 664 (AFDX),
6. ARINC 825,
7. MIL-1553,
8. General serial protocols.

In addition to effort reduction, automated ICD export facilitates elimination of human errors emerging when drafting documents manually.

Note: ICD export format can be customized based on customer requirements.

5.7.3 Export of I/O and Software Specifications

In modern projects, the development of flight software responsible for data input and output is usually carried out with specialized OEM CAD systems based on special format spreadsheets. Availability of all information related to the equipment interaction in dBricks makes it possible to prepare relevant spreadsheets by simply converting the information into a requested format. It helps to avoid potential sources of transfer errors and reduce table preparation time.

Flight software specifications are documents that fully describe functional requirements of on-board equipment. dBricks allows generating such documents automatically based on data stored in the database.

5.7.4 Export of Data Networks Configuration Files

In modern data networks such as ARINC 664, configuration of information flows is described in configuration files uploaded to data exchange nodes. Such files can also be automatically generated in dBricks based on data entered into the database.
Note: For detailed description of the configuration requirements see ARINC 664 part 7.

5.7.5 Development and Export of WDD for Wiring of Test Benches and Flight Simulators

Traditionally, development of wiring WDD for test benches and flight simulators (hereinafter referred to as test benches) has been a very time-consuming task coming from the large number of errors occurring when converting aircraft data into documentation adapted to test bench development. dBricks allows to:

1. Completely eliminate errors that occur during data conversion,
2. Reduce test bench Wiring WDD development effort by 4-5 times,
3. Automatically generate documentation for test bench wiring update when AC design changes are introduced.

5.7.6 Export of Configuration Files for On-Board Measuring and Maintenance Systems

On-board measuring system used for aircraft testing and maintenance system are very similar in terms of data exchange. dBricks offers complete information about pick-up points and structure of information transmitted by the systems allowing to automatically set up

1. registration system input device requirements according to the nomenclature and the number of receivers,
2. wiring lists
3. information formats for its processing.

5.7.7 Export of Configuration Files for Simulation IOs as Part of Test Bench or Flight Simulators

As part of HIL test bench and flight simulator (hereinafter referred to as test benches) development, you need to describe data exchange of equipment simulators not presented on the test benches. Usually, test bench simulation systems can be configured through specialized configuration files. Automated creation of such files based on data entered during ICD development allows to eliminate errors in the development of the files, as well as to reduce process effort by 95-99%.

5.7.8 Export of Analytical Reports

Availability of integrated information on all aircraft equipment, including support systems (electrical system, hydraulic system, etc.), its location in the aircraft and interconnection allows implementing several important analytical functions:

1. Analysis of power bus loads, analysis of equipment failures when one or more power buses are down,
2. Analysis of data bus loads, search for common points in case of data transmit/receive failures,
3. Tracing a signal through all wiring technological elements and data converters, which significantly accelerates troubleshooting during aircraft systems development,
4. Analysis of the list of equipment involved in the aircraft's critical functions,

5. Evaluation of potential loss of aircraft functions (including critical ones) due to potential failures (FHA).

5.8 REST API Functionality

dBricks has a built-in function of interfacing directly with the backend of the tool via REST API, bypassing user interface. API access allows to:

1. CDU database entries,
2. Generate any reports and machine-readable files,
3. Integrate third-party design automation tools.

The detailed REST API description and required support are also provided to our customers if needed.

6 Tool Deployment Options

Depending on the Customers' need, the tool can be used in one of the ways listed below.

6.1 Cloud-Based Software As A Service

The Customer gets access to a software copy deployed in the Vendor's cloud service via the internet. Access is granted for a period paid by the Customer and is automatically renewed given that the next period is paid on time. At Customer's request, after switching to installing the tool on the Customer's local server, work results obtained with the in-cloud tool instance can be transferred to the tool installed on the Customer's local server.

6.2 Customer-Hosted Server

This method involves installing the tool on a server hosted by the Customer. The license term is unlimited subject to the terms of the license agreement. Software maintenance and updates are carried out during maintenance periods specified in the license agreement. Maintenance period can be extended on the terms specified in the license agreement.

7 Tool Modules

Depending on the Customer's needs, tool functionality can be customized by adding/removing the following modules and extension packages.

7.1 Basic Module

Basic module is an integral part of dBricks that makes all other modules tick. This module offers the following functionality:

a) Control access rights,
b) Create, Delete, Update (CDU) basic elements,
c) Export/import (EI) of such basic elements,
d) Manage input data constraints,
e) Generate basic reports.

Basic elements fitting CDU functionality include:

a) Bus types,
b) Cable types,
c) Connector types,
d) Data types,
e) Device templates,
f) Device templates connectors,
g) Device templates physical ports,
h) Device templates functions, including function parameters,
i) Projects,
j) Devices as part of projects,

k) Links between physical device ports within the project,
l) Links between function parameters of devices in the project.

EI functionality for basic elements includes EI of:

a) Parameters of device template functions,
b) Logic implemented by device template function,
c) Device template physical ports,
d) Project devices,
e) Links between device function parameters within the project.

Functionality of data input constraints control allows to:

a) Set constraints for valid names and object indexes,
b) Specify method for auto-generation of bus names,
c) Set constraints on links between variables by data type and measuring units.

Basic reports implemented in the module include:

a) List of project devices,
b) Device Links in the project,
c) Connections of same-type device buses within the project,
d) Parameter path shows full path of data from producer to consumer,
e) Diagrams.

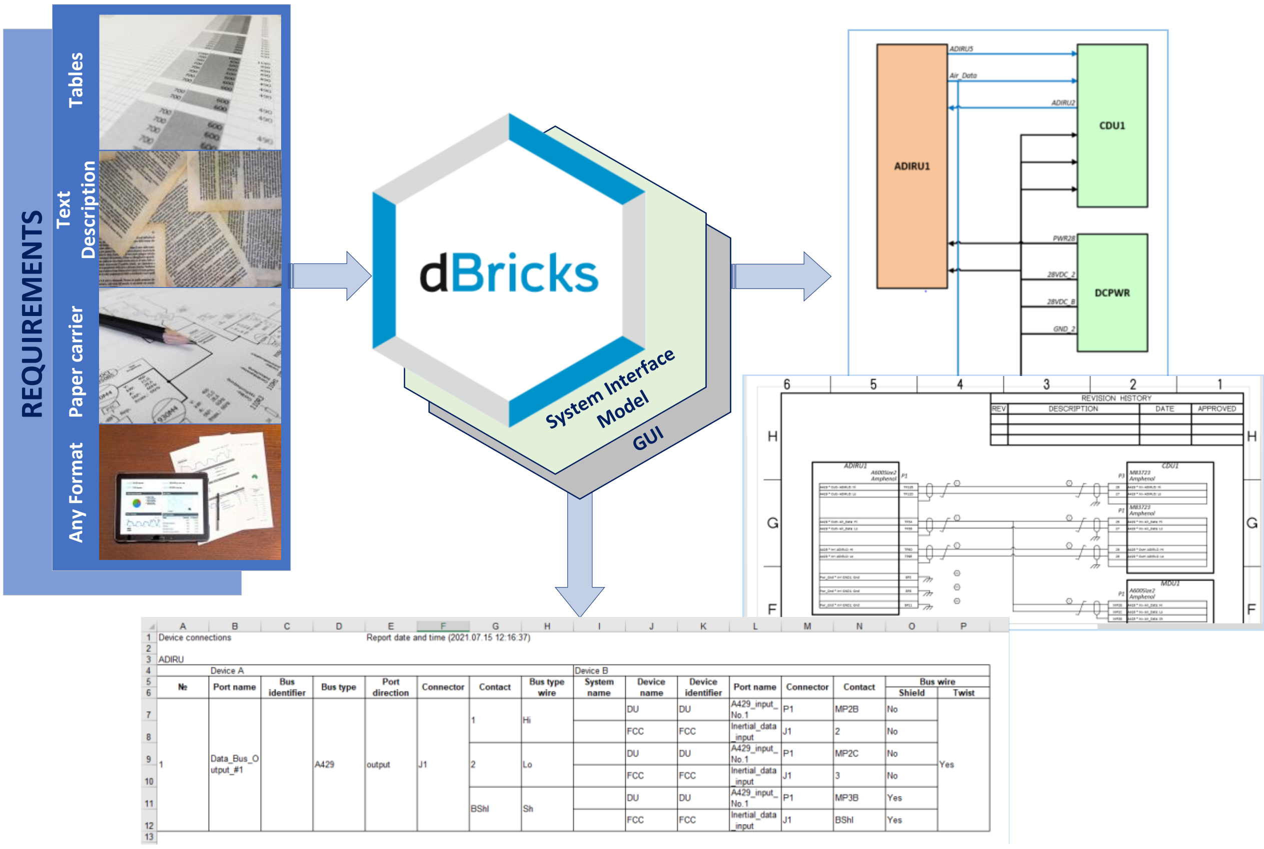


Figure 1. Basic Module Features

7.2 Module for Discrete and Analog Buses

The module allows to CDU information for the following bus types:

a) "Open/Ground" discrete IOs,
b) "Open/Voltage" discrete signals,
c) "Variable amplitude voltage" analog buses,

d) "Variable resistance" analog buses,
e) "LVDT" analog buses,
f) "Inductive proximity sensor" analog buses.

The module also enables export of ICDs by bus types listed above.

7.3 Module for ARINC 429 Buses

This module allows to describe ARINC 429 bus data transfer:

a) In manual mode via intuitive graphic user interface,
b) Automatically by importing from Microsoft Excel spreadsheets.

This module offers the following functionality:

a) CDU of ARINC 429 bus data elements,
b) Export/Import of ARINC 429 bus data elements list into Microsoft Excel spreadsheets,
c) Specialized reports describing ARINC 429 bus data exchange between devices and analyzing bus load,
d) Automated description generation for ARINC 429 data transmission section in ICD (Interface Control Document).

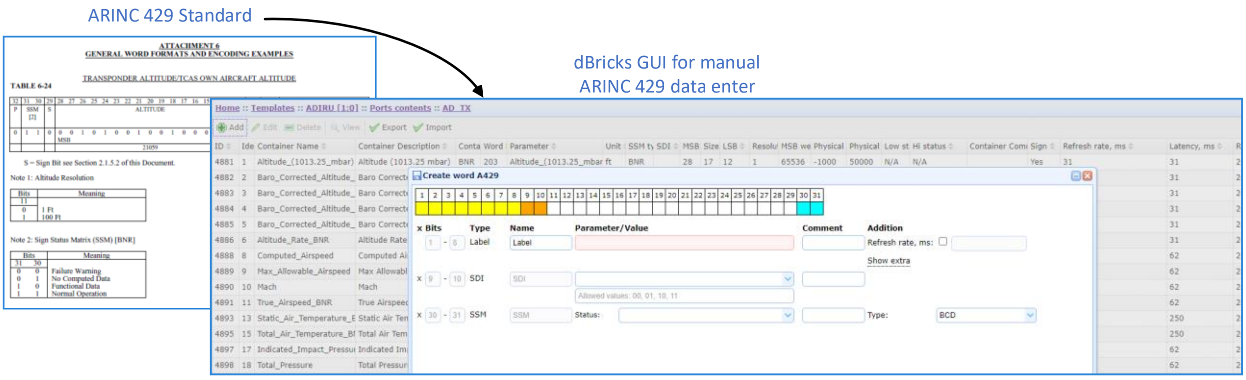


Figure 2. ARINC 429 Data Input

7.4 Module for ARINC 825 (CAN) Buses

This module allows to describe ARINC 825 bus data transfer:

a) In manual mode via intuitive graphic user interface,
b) Automatically by importing from Microsoft Excel spreadsheets.

This module offers the following functionality:

a) CDU of ARINC 825 bus data elements,
b) Export/Import of ARINC 825 bus data elements list into Microsoft Excel spreadsheets,
c) Specialized reports describing ARINC 825 bus data exchange between devices,
d) Automated description generation for ARINC 825 data transmission section in ICD.

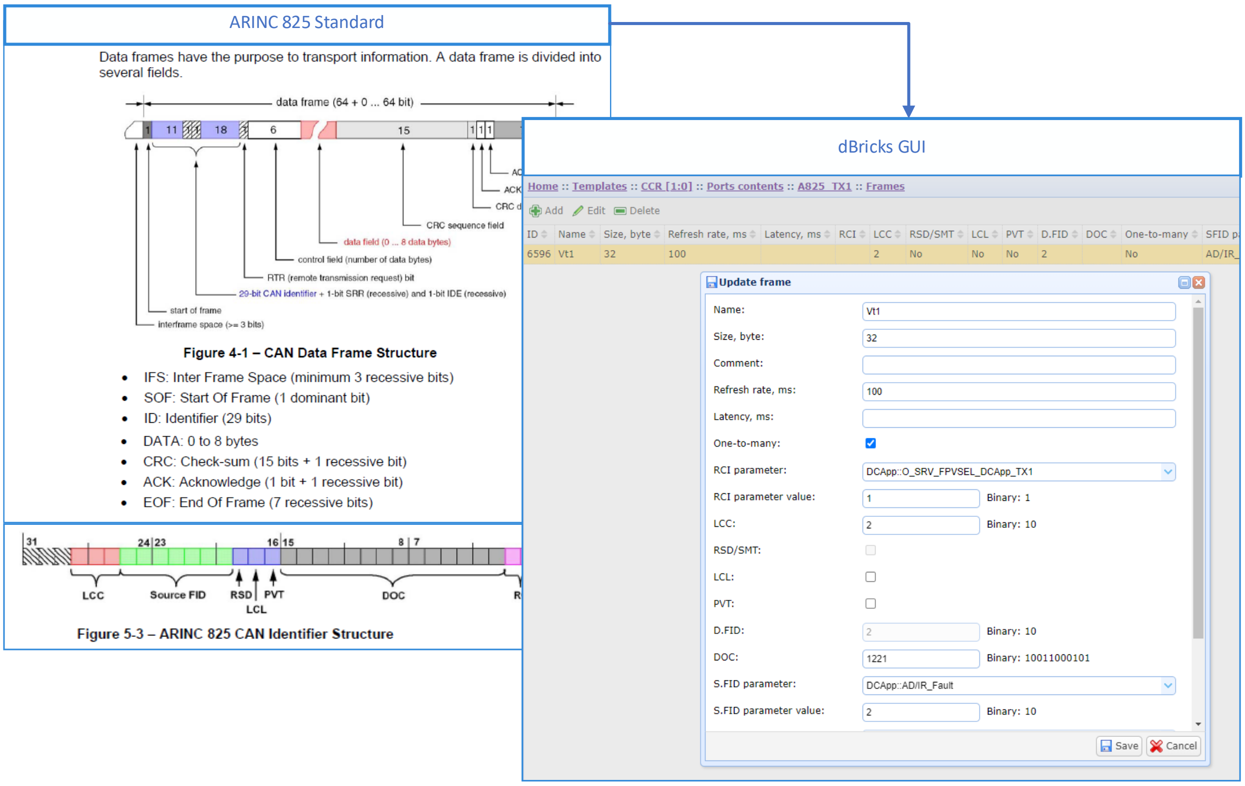


Figure 3. Data Frame Identifier Structure for One-to-Many Communication

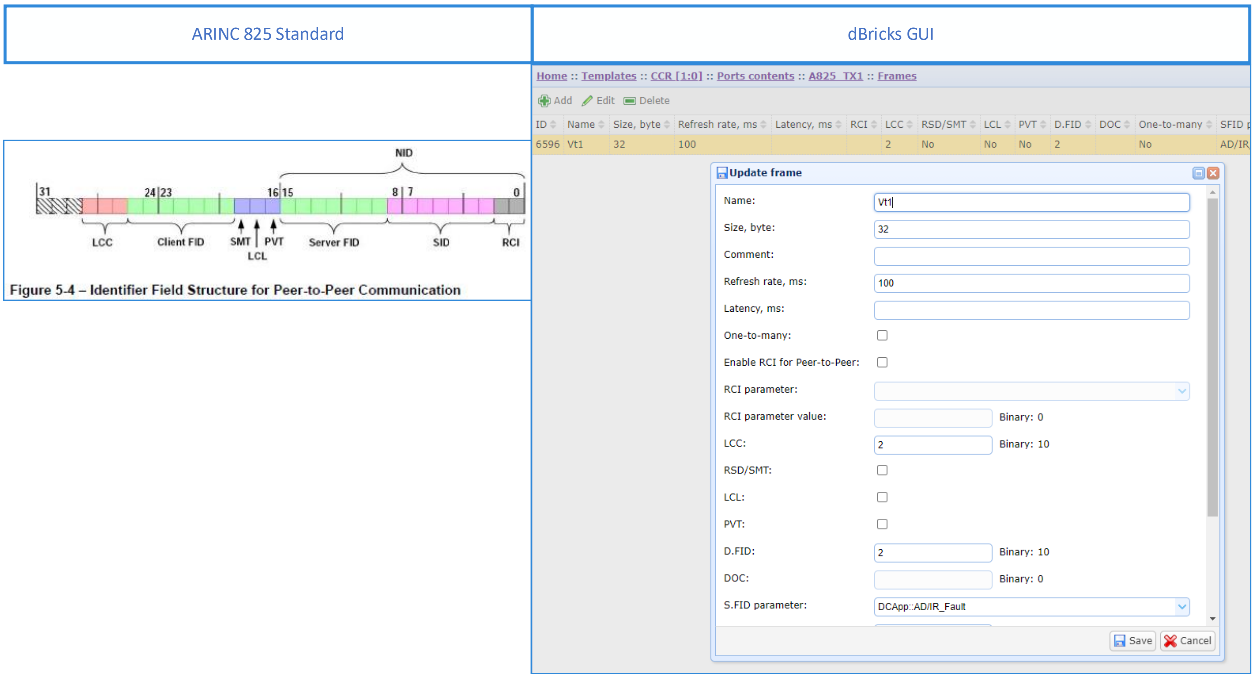


Figure 4. Data Frame Identifier Structure for Peer-to-Peer Communication

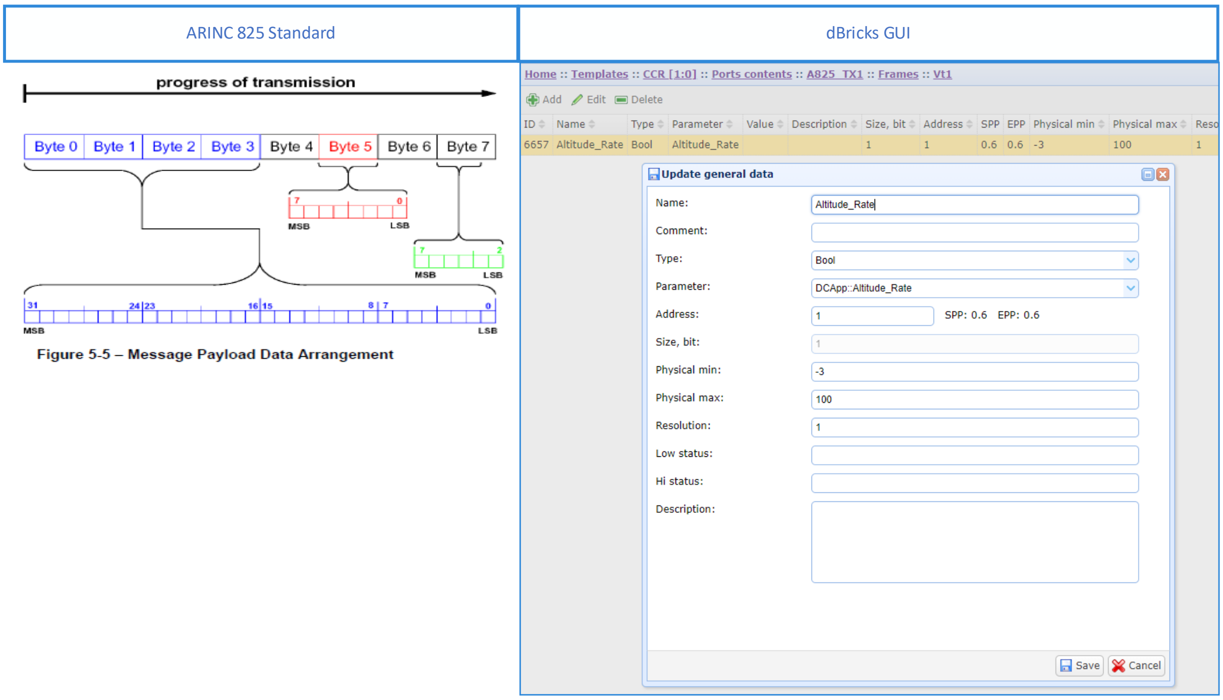


Figure 5. Payload Content (Data Structure)

7.5 Module for ARINC 664 (AFDX) Buses

This module allows to describe ARINC 664 (AFDX) bus data transfer including descriptions of flight software data exchange port structures (application/partition ports acc. to ARINC 653 standard):

a) In manual mode via intuitive graphic user interface,
b) Automatically by importing from Microsoft Excel spreadsheets.

This module offers the following functionality:

a) CDU of ARINC 664 data exchange elements, such as virtual channels, data messages, ES information ports, etc.,
b) Export/Import of ARINC 664 (AFDX) bus data elements list into Microsoft Excel spreadsheets,
c) Specialized reports describing ARINC 664 (AFDX) bus data exchange between devices,
d) Automated description generation for ARINC 644 (AFDX) data transmission section in ICD (Interface Control Document).

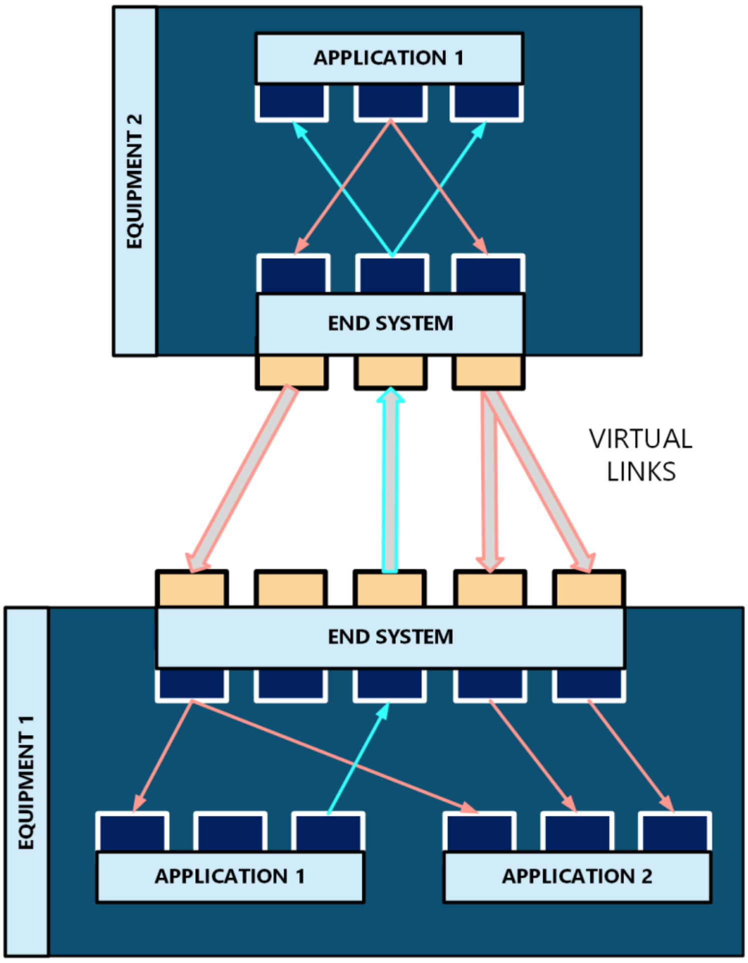
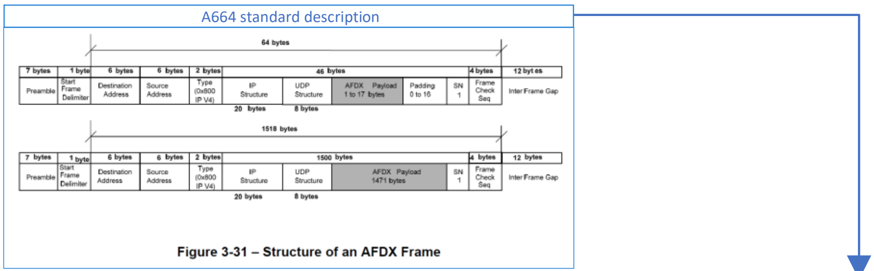


Figure 6. Application Ports Interaction Diagram

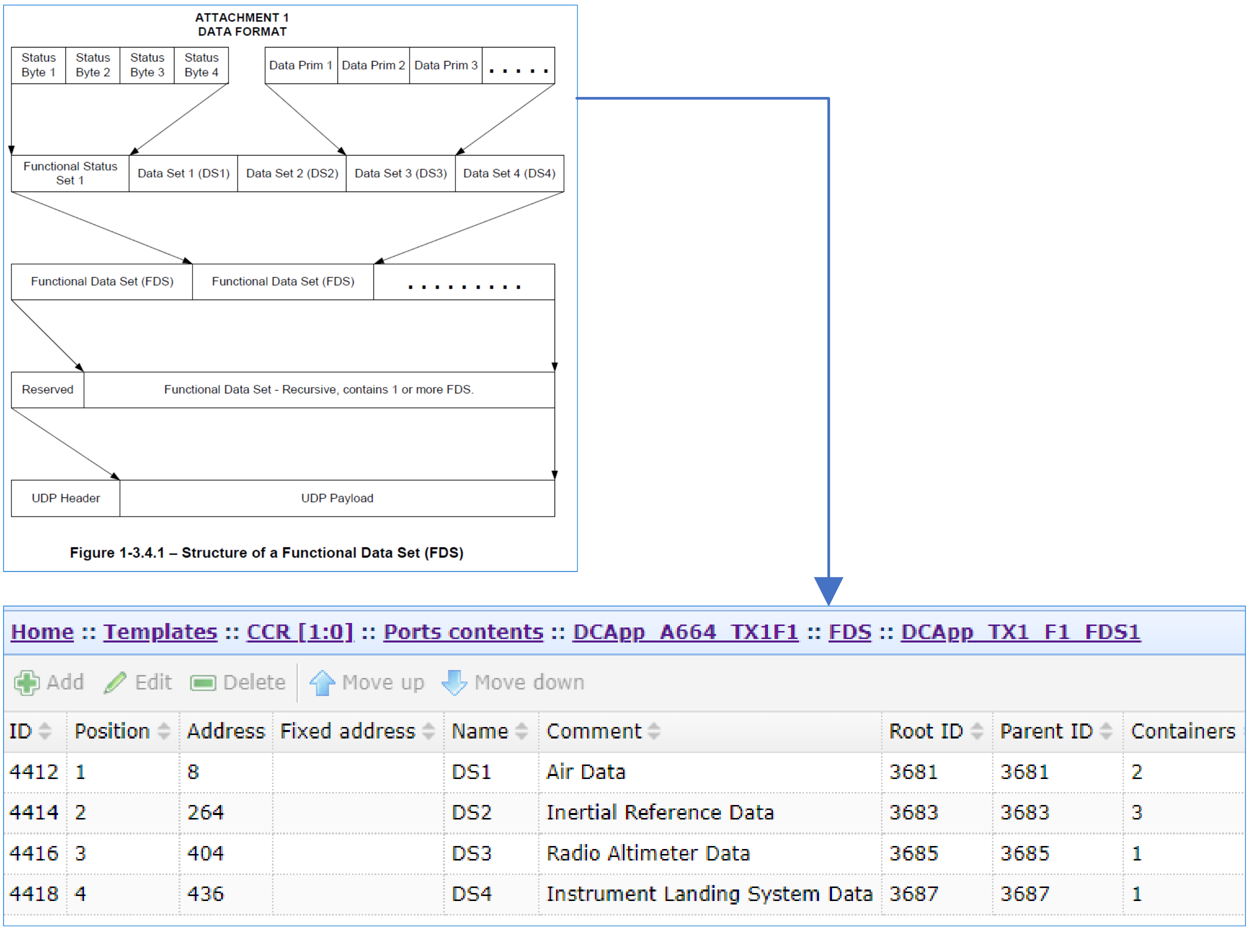


Home :: Templates :: CCR [1:0] :: Ports contents :: DCApp_A664_TX1F1 :: Full view

Export Import

ID	Bit number	Size, bytes	Name	Container type	Parameter name	Data type	Embedded port content	Address	FDS name	DS name	DS position	Relative address	Position	FDS address	DS address
4413		128	DCApp_AD	A664_Comm_OA429B			DCApp_A429_TX1_AD	8	DCApp_TX1_F1_FDS1	DS1	1	0	1	4	8
4415		128	DCApp_IR	A664_Comm_OA429B			DCApp_A429_TX2_IR	264	DCApp_TX1_F1_FDS1	DS2	2	0	1	4	264
4417		32	DCApp_RA	A664_Comm_OA429B			DCApp_A429_TX3_RA	404	DCApp_TX1_F1_FDS1	DS3	3	0	1	4	404
4419		128	DCApp_ILS	A664_Comm_OA429B			DCApp_A429_TX4_ILS	436	DCApp_TX1_F1_FDS1	DS4	4	0	1	4	436
5389		8	DCA_Pitch_Cons	A664_Comm_General_Data	Pitch_Angle	DOUBLE		392	DCApp_TX1_F1_FDS1	DS2	2	128	2	4	264
6844	31	4	test_498	A664_Comm_General_Data	ADS_Computer_Status	Bool		400	DCApp_TX1_F1_FDS1	DS2	2	136	3	4	264
6897		128	qwe	A664_Comm_OWA				136	DCApp_TX1_F1_FDS1	DS1	1	128	2	4	8

Figure 7. dBricks GUI for A664 description. AFDX Frame Example



ID	Position	Address	Fixed address	Name	Comment	Root ID	Parent ID	Containers
4412	1	8		DS1	Air Data	3681	3681	2
4414	2	264		DS2	Inertial Reference Data	3683	3683	3
4416	3	404		DS3	Radio Altimeter Data	3685	3685	1
4418	4	436		DS4	Instrument Landing System Data	3687	3687	1

Figure 8. dBricks GUI for A664 Description. FDS Example

7.6 Module for General Serial Protocols

This module allows you to describe any data exchange protocol by yourself without specific dBricks modifications. E.g. common serial data transfer protocols with defined physical layer constraints: RS-232, RS-485, Raw Ethernet, or any other proprietary serial protocols, could be described.

This module offers the following functionality:

a) CDU of serial data transfer protocols,
b) Export/Import of the list of serial transfer protocol info content elements,
c) Automated description generation for serial protocol data transfer in ICD (Interface Control Document).

Based on this functionality, dBricks already has MIL-STD-1553B data transfer protocol description.

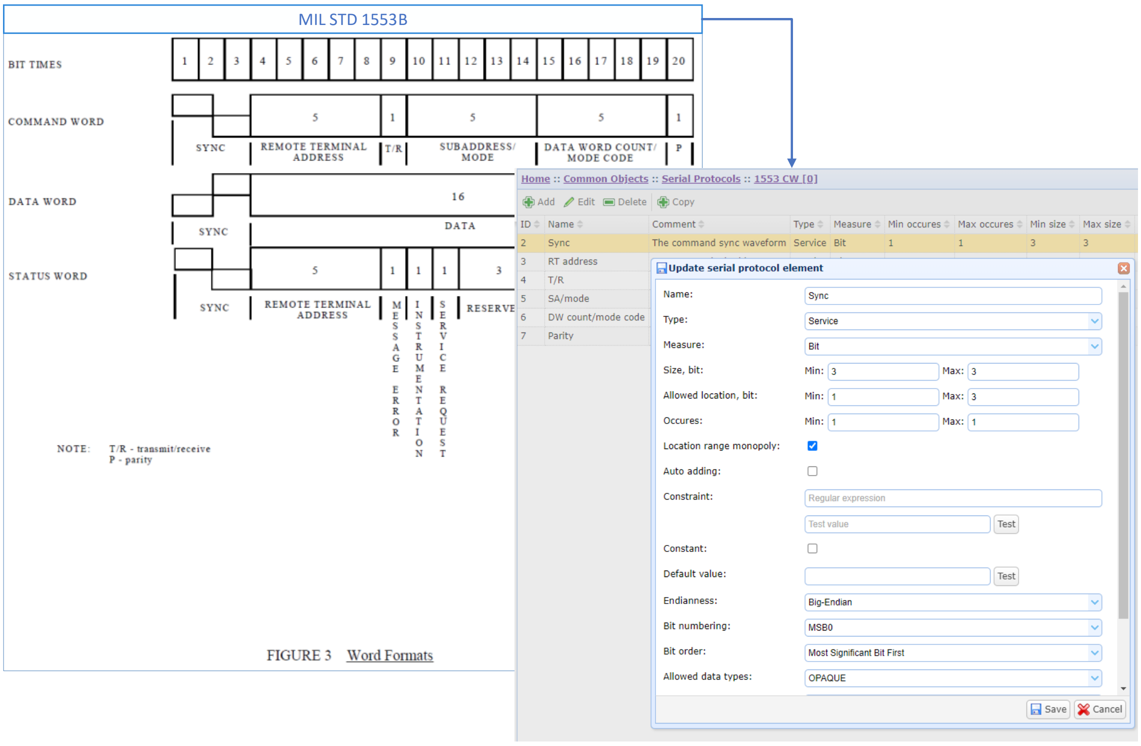


Figure 9. dBricks GUI for Serial Protocol Description. MIL-STD-1553B Example

7.7 Microsoft Visio Diagrams Export Module

The module implements generation of structural diagrams, connection diagrams and etc. in Microsoft Visio format. Examples of automatically generated MS Visio diagrams:

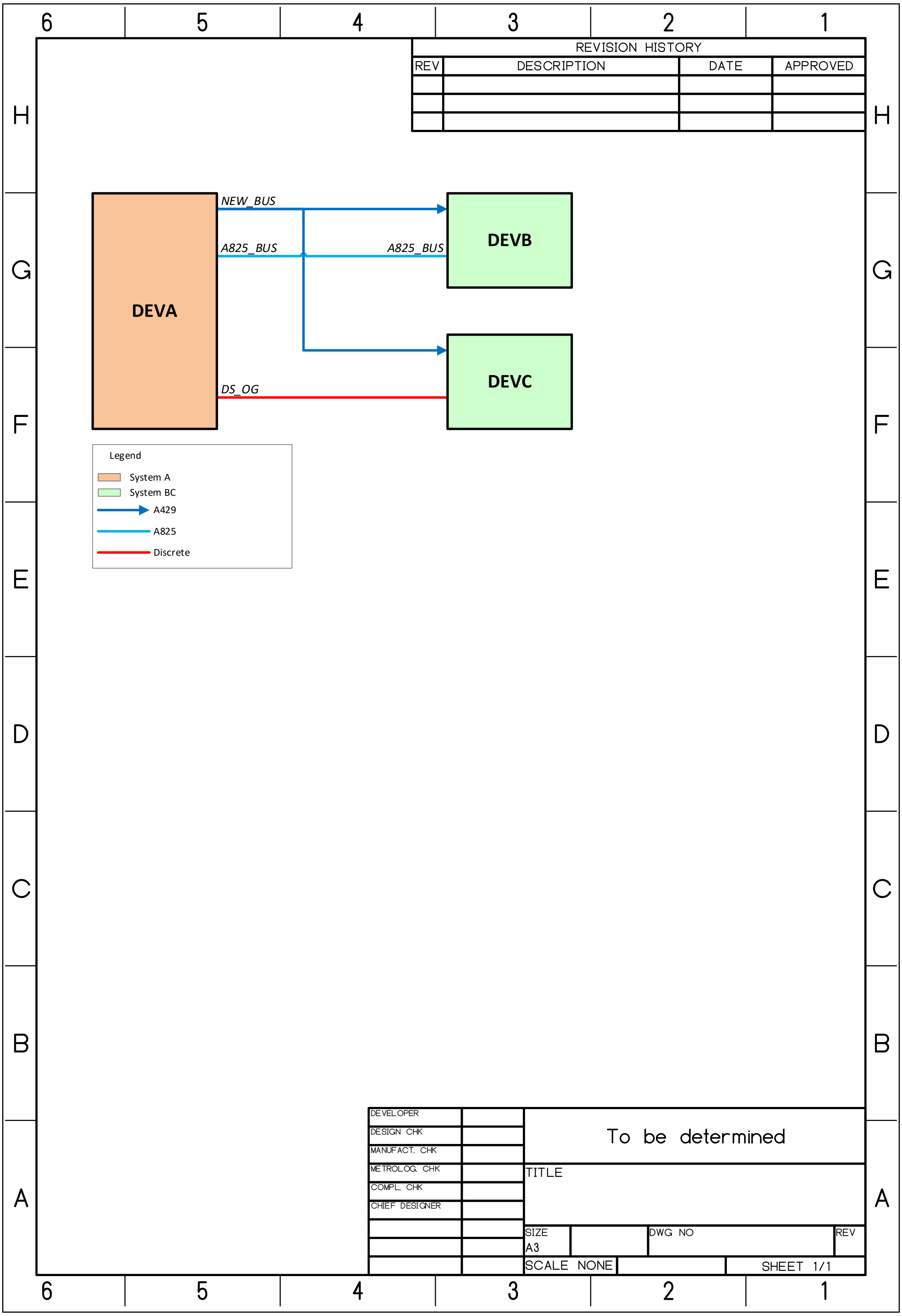


Figure 10. Structural Diagram (ASME Template)

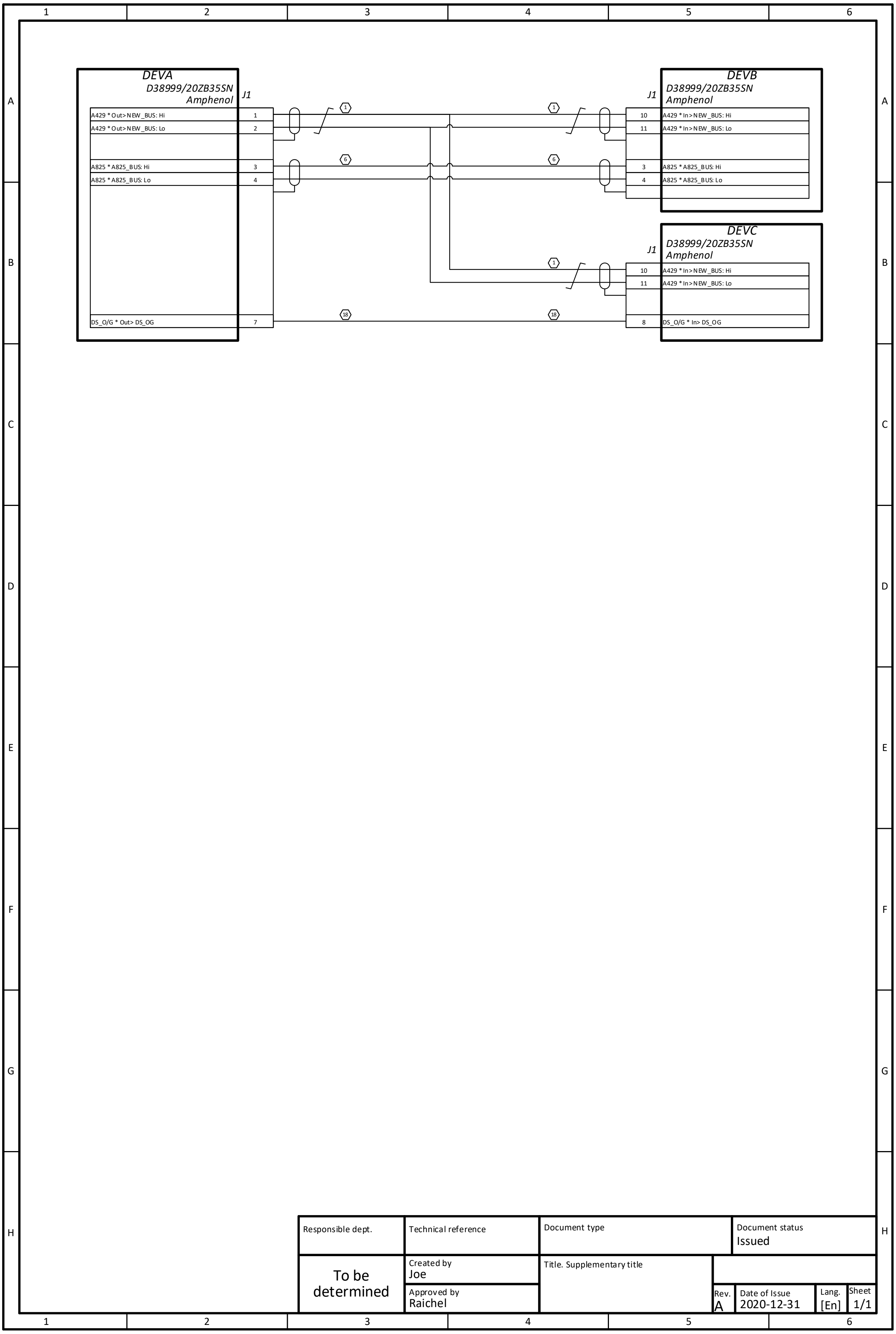


Figure 11. Connection Diagram (ISO Template)

7.8 Wiring Topology Development Module

The module is intended for planning and designing EWIS. This module allows you to go from functional diagram connections to physical wiring layout connections. This module offers the following functionality:

a) Bus topology design with consideration for technological nodes via built-in graphic editor,
b) Completeness and sufficiency analysis for physical wiring connections,

This module provides a set of initial data for harness topology development:

a) Buses divided into wire edges,
b) Crimp splices or connection modules are added for wire multiplication,
c) The following properties are defined for each wire edge:
- Cable and wire types,
- Wire label type,
- Label text,
- Wire length (if necessary),
- Contacts' P/N for bus edges (if necessary, in case of onboard bonding).

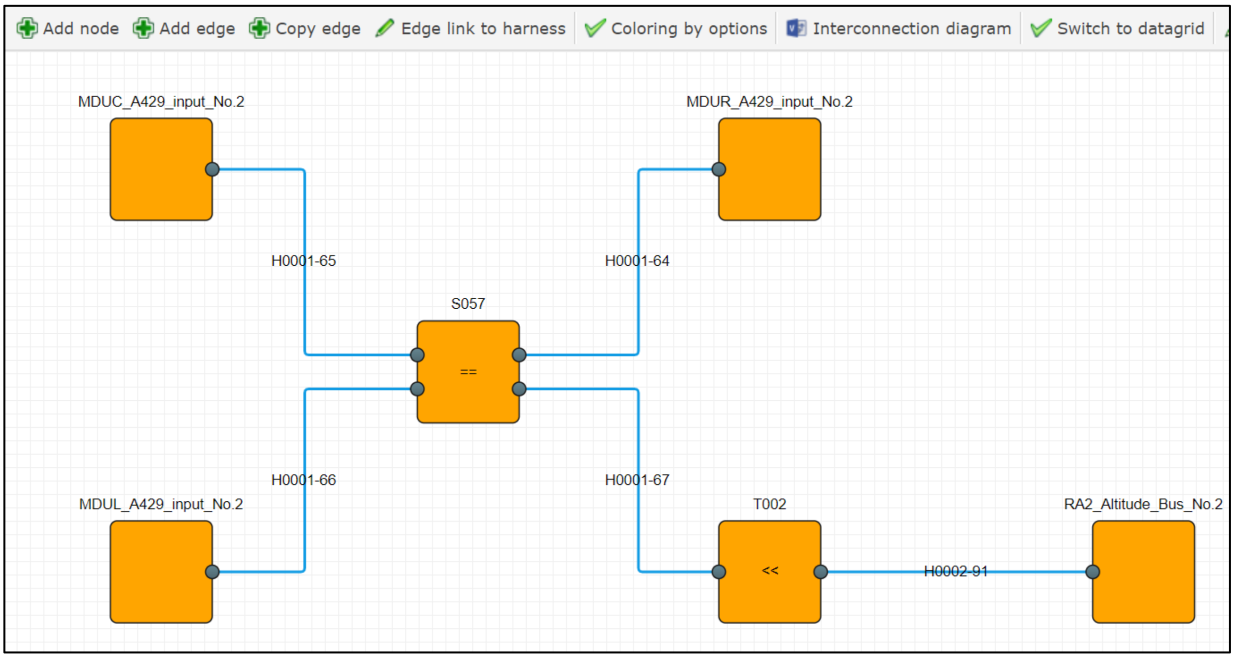


Figure 12. Example of Splitting a Bus into Wire Edges. GUI Screenshot.

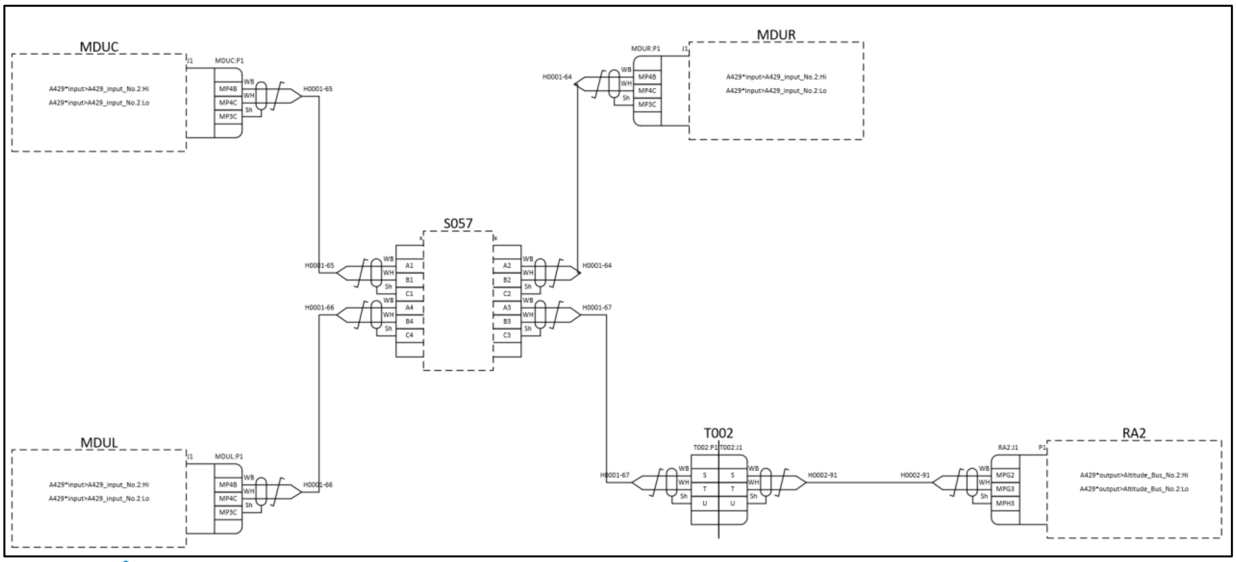


Figure 13. Example of Interconnection Diagram for One Bus

7.9 Harness Development Module

The module is intended for EWIS harness development and harness WDD package generation. Initial data for harness development is the outcome of wiring topology development module. The following functionality is offered:

a) CDU of harness configuration specifying:
   - Length of each tap-off,
   - Length of harness bundle,
   - Crimp splices forming a harness,
   - Harness wire edges,
   - Materials and components required for harness assembly (connectors, clamps, contacts, terminal lugs, braids, lacing tape, labels, etc.).

b) Automatic link of wire edges to respective harnesses,

c) Create and Update harness topology in built-in graphic editor,

d) Automatic generation of WDD package for harness production in MS Visio and MS Excel formats in accordance with the ISO and ASME standards.

Harness document package generated via harness development Module includes:

e) Harness Part List,

a) Harness Wire List,

b) Harness Data List

c) Harness assembly drawing,

Documentation format and file types can be changed and modified according to customer requirements.

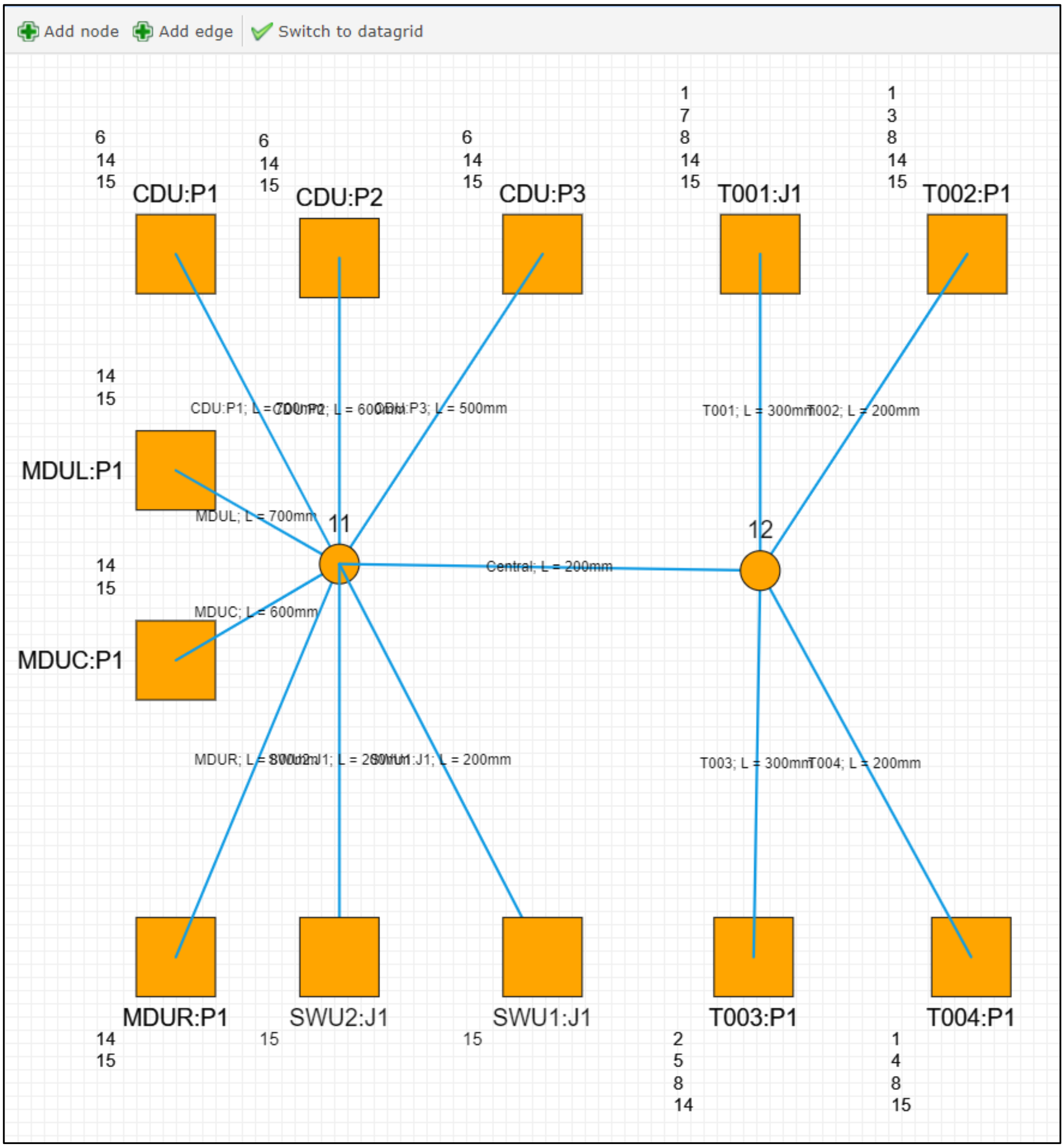


Figure 14. GUI Diagram editor mode for harness development. Screenshot.

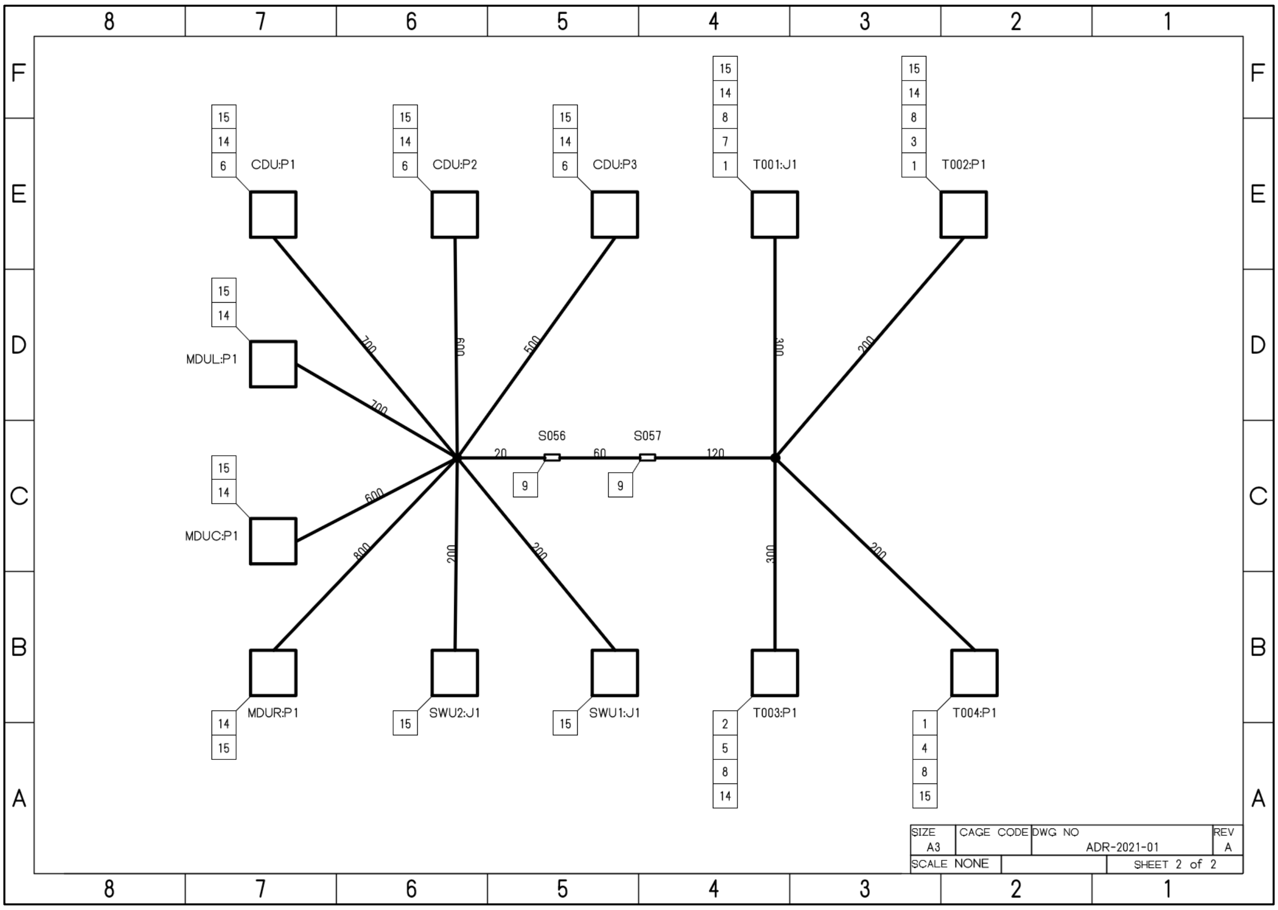


Figure 15. Example of ASME Harness Assembly Drawing

PARTS LIST	DESIGN ACTIVITY NETWORK SUPPLIER	CAGE CODE	LIST NO PL-2021-01	REVISION A
LIST TITLE HARNESS H0001 (COCKPIT)	END ITEM	APPROVED	REV AUTORIZATION NO	SHEET 1 OF 2

FIND NO.	QTY REQD	UNITS	CAGE CODE	PART OR IDENTIFING NO	NOMENCLATURE OR DESCRIPTION	NOTE OR REMARK
1	3	ITEM		M85049/38-19W	BACKSHELL	AMPHENOL
2	1	ITEM		M85049/38-21W	BACKSHELL	AMPHENOL
3	1	ITEM		D38999/26ZF32PA	PLUG	AMPHENOL
4	1	ITEM		D38999/26ZF32PB	PLUG	AMPHENOL
5	1	ITEM		D38999/26ZG20PN	PLUG	AMPHENOL
6	3	ITEM		M83723	PLUG	AMPHENOL
7	1	ITEM		D38999/20ZF32SN	RECEPTACLE	AMPHENOL
8	4	ITEM		SEB-E	SIDE BUSHING	TYCO
9	2	ITEM		D-150-0175	SOLDERSHIELD SPLICE	TYCO
10	37,2	M.		55PC0214-24-9	DEFAULT	TYCO
11	3,3	M.		55PC0221-22-9/96	DEFAULT	TYCO
12	14,9	M.		55PC2124-24-9/96-9	DEFAULT	TYCO
13	5,4	M.		ABS 1503 KD 24	DEFAULT	NEXANS
14	32	ITEM		TMS-SCE-1/8-2,0	HEAT-SHRINKABLE LABEL	TYCO
15	84	ITEM		TMS-SCE-3/32-2,0	HEAT-SHRINKABLE LABEL	TYCO
16	1	ITEM		CM-SCE-1/2-4H-4	LABEL	TYCO
17	20	ITEM		CM-SCE-1/4-4H-4	LABEL	TYCO
18	5,3	M		A-A 52081 TYPE II	LACING TAPE	TBD
19	4,4	M.		ROUNDIT 2000NX HT 13-5	SLEEVE	TBD

REV	DESCRIPTION	DATE	APRVD	REV	DESCRIPTION	DATE	APRVD
A							

Figure 16. Example of Harness Part List

WIRE LIST	DESIGN ACTIVITY NETWORK SUPPLIER	CAGE CODE	LIST NO WL-2021-01	REVISION A
LIST TITLE HARNESS H0001 (COCKPIT)	END ITEM	APPROVED	REV AUTORIZATION NO	SHEET 1 OF 6

WIRE NO	COLOR	SIZE AWG	WIRE RUNS				NOTE OR VIEW	FUNCTION ROUTING REMARKS	APPROX LENGTH, m	FIND NO
			FROM		TO					
			DEVICE	TERMINAL	DEVICE	TERMINAL				
H0001-1	SH		MDUL:P1	MP6B	T001:J1	R		ADIRU1 DB2:SH	1,2	12
H0001-1	WB	24	MDUL:P1	MP5B	T001:J1	N		ADIRU1 DB2:HI	1,2	12
H0001-1	WH	24	MDUL:P1	MP5C	T001:J1	P		ADIRU1 DB2:LO	1,2	12
H0001-10	WH	24	CDU:P1	9	T001:J1	X		CDU AD ON/OFF DISCRETE 1:HI	1,2	10
H0001-11	WH	24	CDU:P2	9	T002:P1	j		CDU AD ON/OFF DISCRETE 2:HI	1	10
H0001-12	WH	24	CDU:P3	9	T004:P1	F		CDU AD ON/OFF DISCRETE 3:HI	0,9	10
H0001-13	WH	24	CDU:P3	14	T002:P1	Z		CDU ALIGN/NOT READY:HI	0,9	10
H0001-14	WH	24	CDU:P1	14	T001:J1	U		CDU ALIGN/NOT READY 1:HI	1,2	10
H0001-15	WH	24	CDU:P2	14	T002:P1	c		CDU ALIGN/NOT READY 2:HI	1	10
H0001-16	WH	24	CDU:P1	5	T001:J1	d		CDU CDU ON/OFF 1:HI	1,2	10
H0001-17	WH	24	CDU:P2	5	T002:P1	g		CDU CDU ON/OFF 2:HI	1	10
H0001-18	WH	24	CDU:P3	5	T002:P1	h		CDU CDU ON/OFF 3:HI	0,9	10
H0001-19	WB	24	CDU:P1	24	T001:J1	A		CDU CONTROL BUS 1:HI	1,2	12
H0001-19	WH	24	CDU:P1	25	T001:J1	B		CDU CONTROL BUS 1:LO	1,2	12
H0001-19	SH		CDU:P1	ExtG	T001:J1	C		CDU CONTROL BUS 1:SH	1,2	12
H0001-2	WH	24	CDU:P2	12	T004:P1	A		CDU-M1 DISCRETE 2:HI	1	10
H0001-20	WB	24	CDU:P2	24	T002:P1	A		CDU CONTROL BUS 2:HI	1	12
H0001-20	WH	24	CDU:P2	25	T002:P1	B		CDU CONTROL BUS 2:LO	1	12
H0001-20	SH		CDU:P2	ExtG	T002:P1	C		CDU CONTROL BUS 2:SH	1	12

REV	DESCRIPTION	DATE	APRVD	REV	DESCRIPTION	DATE	APRVD
A							

Figure 17. Example of Harness Wire List

DATA LIST	DESIGN ACTIVITY NETWORK SUPPLIER	CAGE CODE	LIST NO	REVISION
LIST TITLE HARNESS H0001 (COCKPIT)	END ITEM	APPROVED	REV AUTORIZATION NO	SHEET 1 OF 1

NO	DWG SIZE	DOCUMENT NO	NO OF SHEETS	REV STATUS	NOMENCLATURE OR DESCRIPTION
1	A3	ADR-2021-01	2	A	ASSEMBLY DRAWING
2	A4	PL-2021-01	2	A	PARTS LIST
3	A3	WL-2021-01	6	A	WIRE LIST

REV	DESCRIPTION	DATE	APRVD	REV	DESCRIPTION	DATE	APRVD

Figure 18. Example of Harness Data List

7.10 Module for ICD Export to MS Word

The module is intended for generating Interface Control Documents (ICD) between project devices arbitrarily selected by the user in the form of a ready-made MS Word text document. ICD includes data transfer description of the following interface types:

a) Discrete IOs,
b) Analog IOs incl. proximity sensor signals,
c) Linear Variable Differential Transformer (LVDT) signals,
d) ARINC 429,
e) ARINC 825,
f) Ethernet,
g) ARINC 664 (AFDX),
h) MIL-1553,
i) RS-485,
j) RS-232,
k) Other general serial protocols.

Document format and structure meet the requirements of most of the world's leading AE manufacturers. If necessary, document format can be changed according to customer requirements.

7.11 Mathworks Simulink Integration Toolbox

dBricks Toolbox module is an application for Mathworks Simulink interfaced with dBricks via REST API protocol. It is used to create links in Mathworks Simulink and control developed:

a) Function Interface-Model (FIM),
b) Device Interface-Model (DIM),
c) AE data exchange models, where information about interface, connections, links and properties are stored in dBricks project.

Correlation between interface-models, their properties, and data links stored in a dBricks project ensures convergence of the main Model-Based Development (MBD) phases: Hardware-in-the-Loop (HIL), Model-in-the-Loop (MIL), Software-in-the-Loop (SIL), Processor-in-the-Loop (PIL).

dBricks Toolbox is designed for the following specialists:

- Simulink models developers,
- System engineers,
- Model-based design specialists,
- Test systems specialists.

7.11.1 Function Interface-Model (FIM) Development

dBricks Toolbox allows you to develop function interface-model (FIM) for dBricks device template function. E.g. ILS function of MMR device template with a set of input/output parameters is already described in dBricks:

Inputs:

ID	Name	Direction	Unit	Data type
1123	Selected_Runway_Heading	input	°	DOUBLE
1124	Landing_System_Mode/Frequency_IN	input	MHz	DOUBLE
1125	Paired_DME_Frequency	input	MHz	DOUBLE
1126	Latitude	input	°	DOUBLE
1127	Longitude	input	°	DOUBLE
1128	Ground_Speed	input	kn	DOUBLE
1129	Track_Angle_(True)	input	°	DOUBLE

Outputs:

ID	Name	Direction	Unit	Data type
1060	Selected_Runway_Heading_BCD	output	°	DOUBLE
1062	Landing_System_Mode/Frequency	output	N/A	OPAQUE
1064	MLS_Channel_Selection	output	N/A	OPAQUE
1066	Selected_Runway_Heading_BNR	output	°	DOUBLE
1068	Localizer_Deviation	output	°	DOUBLE
1070	Glideslope_Deviation	output	°	DOUBLE
1072	Approach_ID_#1_1st_Char	output	N/A	CHAR
1073	Approach_ID_#1_2nd_Char	output	N/A	CHAR
1075	Approach_ID_#2_3rd_Char	output	N/A	CHAR
1076	Approach_ID_#2_4th_Char	output	N/A	CHAR
1077	Aircraft_Altitude	output	ft	DOUBLE
1078	Anchor_Point_Latitude	output	°	DOUBLE
1079	Anchor_Point_Longitude	output	°	DOUBLE
1080	Anchor_Point_Altitude	output	ft	DOUBLE
1082	Local_Magnetic_Deviation	output	°	DOUBLE
1083	Runway_Threshold_Latitude	output	°	DOUBLE
1084	Runway_Threshold_Longitude	output	°	DOUBLE

dBricks Toolbox creates FIM in Simulink with these IO parameters according to dBricks data:

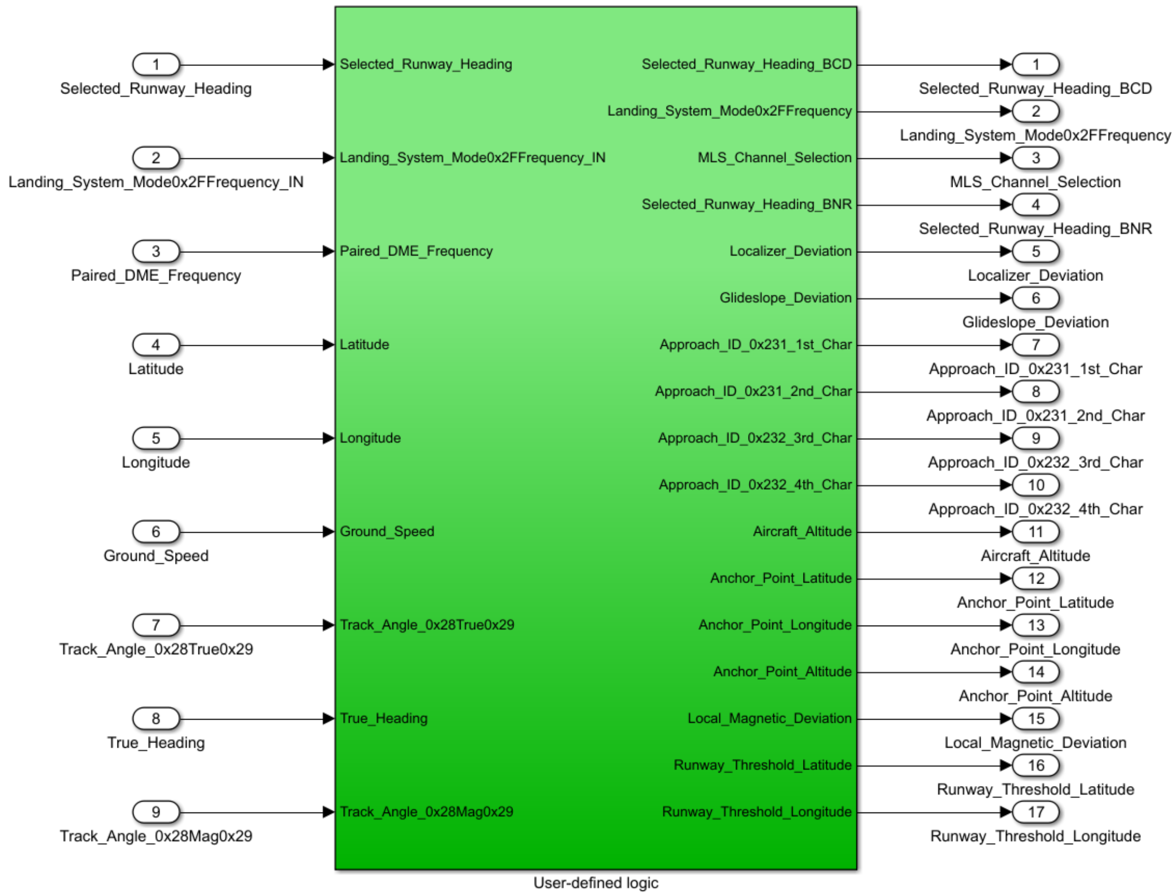


Figure 19. Example of Automatically Generated FIM inputs/outputs in Simulink

Automatically generated input/output parameters used by models to interface with each other are in white; section for user defined logic description for automatically generated model is in green. FIM contains Simulink ports that correspond to the list of input/output parameters of ILS function of MMR template designed in dBricks. Simulink ports names, sequence, and properties were predetermined in dBricks, in particular:

- Parameter data types,
- Measuring units,
- Limit values.

7.11.2 Device Interface-Model (DIM) Development

Device Interface-Model (DIM) consists of logic and transport layers.
Logic layer is a set of FIM.
Transport layer corresponds to:

- Device ports described in dBricks,
- Port contents described in dBricks,
- Links of port content parameters with FIM (red arrows, Figure 20),
- Links between device port contents.

Schematic interpretation of logic and transport DIM layers look as follows:

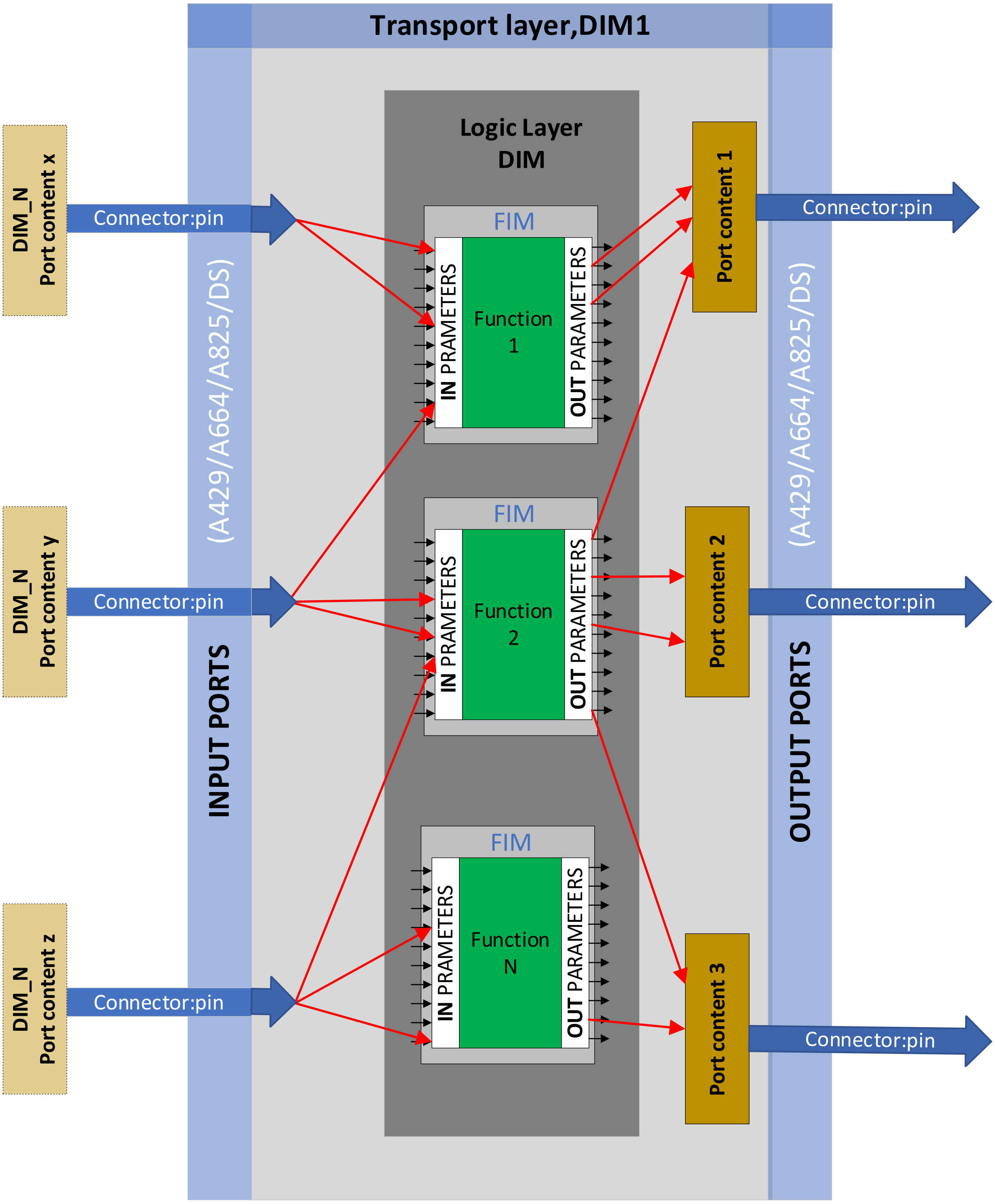


Figure 20. FIM Schematic

Example of A429 port content:

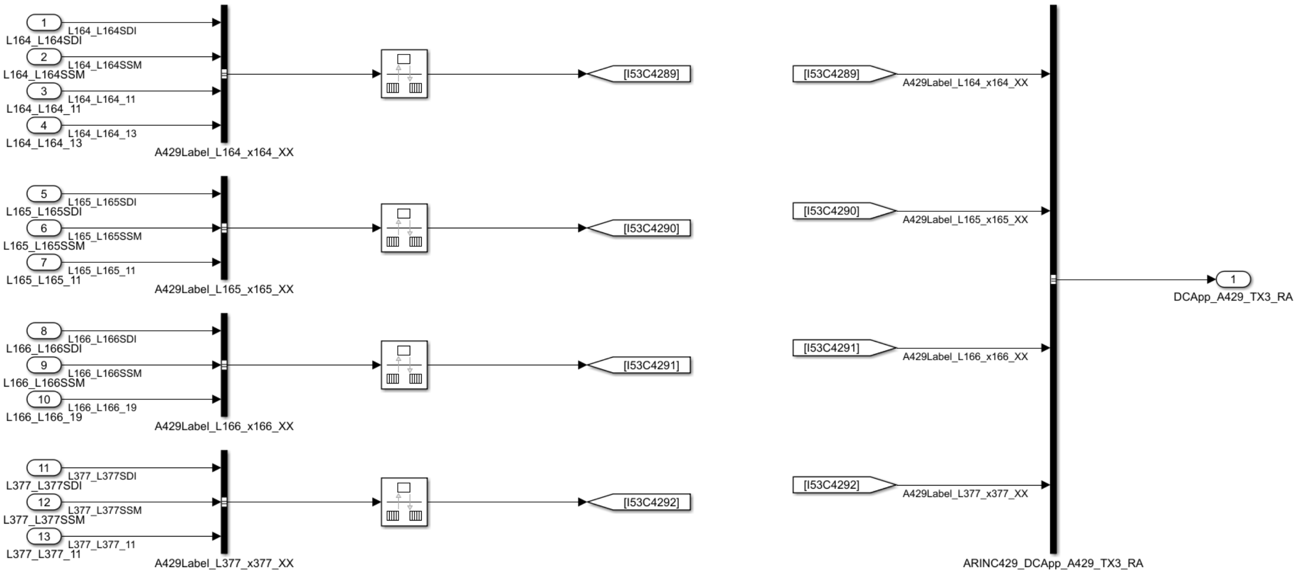


Figure 21. Example of ARINC 429 DIM Port Content in Simulink

dBricks Toolbox allows to develop AE data exchange models based on:

- Data stored in dBricks,
- Models created in dBricks,
- Logic defined by user in models created in dBricks.

E.g. let's consider aircraft scale project: there are hundreds of thousands of parameter links between devices. To link FIM/DIM parameters manually and control conformity of data exchange between models in dBricks is an extremely hard task for such project. Since there is no direct integration between dBricks and MATLAB, FIM interfaces, data exchange models and information stored in dBricks will inevitably diverge over time. dBricks and dBricks Toolbox for MATLAB/Simulink resolves this issue by providing the user an opportunity to check the set of FIMs interfaces according to the updated data in dBricks.

7.12 Module for Export of Data Exchange Configuration Files for HIL Test Benches

The module is designed to generate configuration files for data exchange model of a device, systems, or entire AE in dBricks for HIL test benches. A configuration file is any machine-readable file with predefined structure. E.g. data exchange configuration in VHTNG (Virtual Hybrid Testing Next Generation project), which is a machine-readable xml file. VHTNG files are used to configure HIL test benches. Find out more about VHTNG https://www.sae.org/publications/technical-papers/content/2018-01-1949/ or see examples of VHTNG module application by TechSAT https://techsat.com/systems-and-solutions/development-virtual-integration/ed-247-vistas/

7.13 Toolbox for Development of Test Bench and Flight Simulator Wiring

Toolbox implements Wiring WDD development functionality for test benches and flight simulators based on AC AE device data entered into dBricks. User selects AE project to be implemented at the test bench and enters information about AC AE location at the test bench, as well as information about location of bench equipment and technological components. Based on input data, the toolbox automatically generates harnesses and the following set of documents:

a) Part lists required for production of harnesses/harness tap-offs,
b) Harness wire lists,

c) Connections tables for harnesses and technological components,
d) Wire identification tables,
e) Wiring production effort tables (typical operations necessary for wiring production).

Test bench harnesses have simplified topology, whereby harness topology and tap-off lengths are generated automatically in dBricks. End nodes in the test bench harness is always the cable part of the device connector on one side; on the other, the harness is split into one or more tap-offs with terminal devices located in BOB (Brake Out Box). The harness automatically includes all edges associated with the end node of the device. Test Bench Harness Schematic:

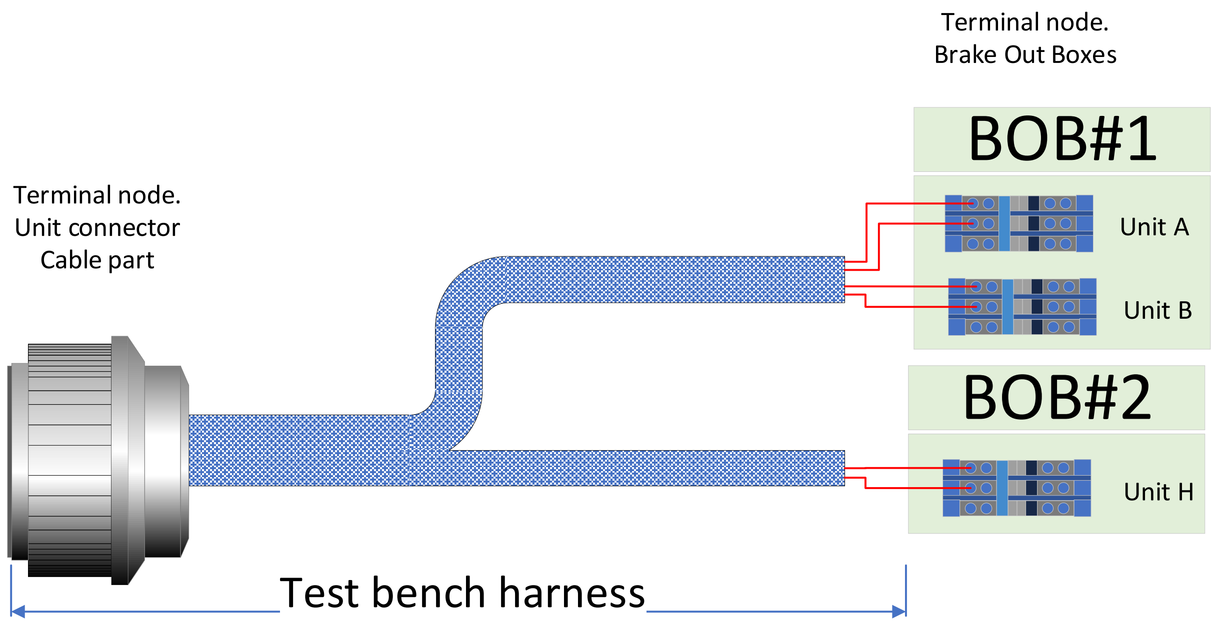


Figure 22. Test Bench Harness Schematic

The toolbox also enables automatic generation of configuration files used to configure test bench and flight simulation systems. List and format of files varies depending on the equipment used by the Customer, but usually includes the following main groups of files:

a) Lists of simulated devices,
b) Lists of simulated device ports,
c) Lists of input/output parameters of simulated devices functions,
d) Lists of links between the parameters of simulated devices functions,
e) Description of the format of simulated ports transmitted/received data with mapping to parameters. Description depends on the simulated port type.

8 Technical Support and Tool Update Policy

Technical support of the tool comprises Vendor services listed below. For license agreements with SaaS tool access, technical support is provided throughout the access period. For license agreements with tool installation on the Customer's local server, technical support is provided during the time specified in the license agreement.

8.1 Phone Support

Such support includes consulting for Customer's representatives by the phone indicated in the license agreement.

Unless otherwise specified in the license agreement, support is provided from 10:00 to 18:00 (Moscow time) on weekdays in accordance with the laws of the Russian Federation.

Unless otherwise specified in the license agreement, the total scope of consulting should not exceed 10 hours per month.

8.2 Email Support

Such support includes response to Customer inquiries via e-mail detailed in the license agreement.

Unless otherwise specified in the license agreement, the deadline for such response is two working days in accordance with the laws of the Russian Federation.

8.3 Response to Comments Submitted in Bug&Improvement Tracking System

Such support includes registration, processing, resolving and/or provision of a reasoned correction refusal, on the basis of Customer comments in the bug tracking system (located at URL specified in the license agreement).

Response time for comments depends on the error severity. Error severity levels are as follows:

a) Blocking: the error results in total inoperability of the tool. Response period: within one calendar day.
b) High: the error leads to a substantial decrease in tool’s intended availability. Response period: one business day in accordance with the laws of the Russian Federation.
c) Low: the error results in minor tool usability issues. Response period: five business days in accordance with the laws of the Russian Federation.
d) Improvement Suggestion means Customer's suggestions for tool functionality improvement as compared to the current version. Response period: fifteen business days in accordance with the laws of the Russian Federation.

Note: Please, note that response time does not equal time to fix an issue or correct an error.

8.4 Upgrade Policy

Within the technical support period, the Vendor provides the Customer with information on available new versions of the tool. At the Customer's request, the Vendor can upgrade the tool to the latest available version.

9 Conclusion

dBricks system allows to:

a) Significantly reduce development effort for integrated systems and AE,
b) Ensure automatic management of project data changes and their reflection in project documentation,
c) Reduce design time by increasing accessibility of existing data and latest changes for design process stakeholders,
d) Minimize scope of equipment and optimize AE architecture through FHA at early design stages,
e) Reduce design and certification risks by automating application of existing regulations and certification requirements.