

PhysiMUX app

Sital Technology

v 2.2

User Guide

TABLE OF CONTENTS

Table of contents	2
Introduction	3
Minimum System Requirements	3
Topology Files	4
UI Overview	6
Device connection and the Info Panel	7
Faults detection	8
Short Circuit Detection	9

INTRODUCTION

Sital's PhysiMUX Wiring Faults tester and software app are a tool for real-time MIL-STD-1553 bus physical layer faults diagnostics and monitoring. The software app uses Sital's PhysiMUX (Sital's SnS device) to detect intermittent or continuous open and short circuit faults and indicates its findings with visualizations on the monitored bus's topology diagram. This tool reduces the 1553 systems maintenance, diagnostics and repair time and costs unprecedentedly by eliminating the need for time-consuming manual troubleshooting and enabling preventive maintenance. This is a user manual for the early-stage PhysiMUX Wiring Faults app, describing its features for detecting and locating open and short circuit faults.

MINIMUM SYSTEM REQUIREMENTS

In order to work with the PhysiMUX Wiring Faults app, 2 assets are required:

1. Sital's SnS (Safe and Secure) device "PhysiMUX" - a physical 1553 LRU that connects to the bus and monitors activity.
2. A Windows PC computer with a USB 2.0 port.

In addition to the PhysiMUX and the USB host computer, it's also necessary for the user to develop and attached a Topology file to the software app. The topology file is a text file with a dedicated format that must use a ".topo" suffix that describes key system topology attributes for a specific bus. Note that topology files are interpreted by the PhysiMUX app, and therefore should be carefully checked for syntax and logic as if it was a piece of code.

PhysiMUX app will include a dedicated wizard to create *.Topo files.

TOPOLOGY FILES

Topology files are Sital's own "topo" format that describe 1553 system topologies by enumerating some of its high-level attributes. These files are interpreted by the app as providing the topology for the system to be monitored. At this early stage, the topology interpreter cannot handle exceptions in topo files. This will be added in future releases. Therefore, writing topo files should be done carefully and double-checked for syntax and logic errors. However, they are very easily read and written.

Topology files handling

- Writing topology files can be done using any text editor. The syntax used to write it is explained in the following 'structure' paragraph.
 - When the file is ready, change its extension to .topo in order for the application to recognize it as a topology file.
 - Drag and drop the file into the application's window. The topology file is now available for use.
- Topology files need to be stored in the folder ~/Sital PhysiMUX/Topologies. The /PhysiMUX directory can be found in the user's home directory upon first launch of the application. In Windows, the home directory is found at C:\Users\- In the app, available topo files can be found in the lower red bar, next to the  icon followed by a white triangle.



Available topo files can be found in the lower red bar, next to the  icon followed by a white triangle.

Structure:

- Each line describes an attribute.
- No comments or any content other than the required attribute lines are allowed.

- The following block is a topology file template that includes an explanation for each line. The explanations are indented and start with a '\$' symbol. The strings inside <> brackets are values inserted by the author of the file. Any other string outside the <> brackets are constant strings that are included in the specific line for every topo file, and should not be altered by the author.

\$Constant title for every topology file

Topology File Format Version 1553.0, Sital Technology format,

\$Name of the bus

Name of this bus: <string: BUS_NAME>

\$bus A annotation: tells the interpreter that the following LRUs list declares all the LRUs that are connected to bus A

bus: A

\$list of LRUs sorted by their real position on bus A, relative to one of the terminations.

\$Note that the SnS's and BC's (bus controller's) positions are mentioned as well by dedicating a row in the list that states "SnS," or "BC," and nothing else.

\$each line has 2 comma-separated columns. the first states the LRU's name, the second grounds its address.

<string: LRU_1_NAME>, <int: LRU_1_ADDRESS>

<string: LRU_2_NAME>, <int: LRU_2_ADDRESS>

:

SnS,

:

BC,

:

<string: LRU_n_NAME>, <int: LRU_n_ADDRESS>

The same applies for Bus B.

bus B:

<string: LRU_1_NAME>, <int: LRU_1_ADDRESS>

::

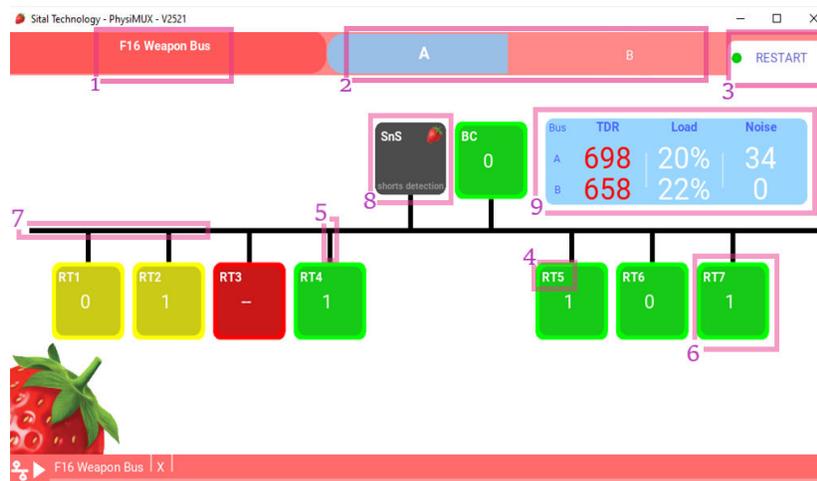
- **A topology file template can be found in the Topologies directory "topo_example.txt"**

UI OVERVIEW

Upper bar [indexes 1-3 in the picture]:

The upper bar shows

The **1**name of the topology on the left, and frames some basic control over the SnS device and the topology diagram’s visualization. The **2**bus switch button is for selecting whether Bus A or Bus B should be visualized. In this screenshot, bus A is selected because the button that represents it is blue. Right of the bus switch on the upper bar, there is a **3**white tab. The SnS device connection state is deduced from the white tab. If it displays a red LED and a label "SnS not found", no device is connected to the app. If the LED appears green and the label turns to a 'RESTART' button, the device is connected successfully. Pressing the "RESTART" button will restart any learned attributes about the bus and clean any previous fault indications from the topology diagram. If no topology is selected, (or topologies are explicitly deselected by choosing 'X' in the topologies bar) the upper bar will indicate NO TOPO MODE. In this mode, only the **9**info panel will be shown (the info panel is described below).



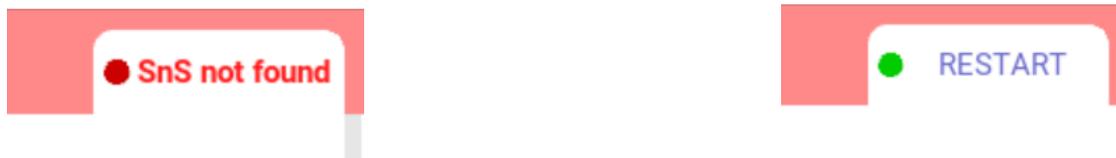
Topology diagram [indexes 4-9 in the picture]:

The PhysIMUX app draws a topology diagram based on a topo file for Bus A or Bus B. This diagram represents the system that is currently being monitored. It consists of: **7**main bus-line - a horizontal black line that represents the main bus. **5**Stub - a vertical black line that connects the main bus to an LRU. It represents a physical stub. **6**LRU - a colored square, connected by a stub to the main bus-line, and positioned below the main bus-line. It represents a physical LRU. The LRU is **4**labeled by its name. An LRU’s color represents its responsiveness: *green- responsive; red- unresponsive; yellow- intermittent unresponsiveness; grey - no message sent to by the BC.* **8**SnS - a dark grey square, connected by a stub

to the main bus-line, positioned above the main bus-line. It represents the physical SnS device connected to the bus and to its host computer. The SnS has an "SnS" label on it and a little strawberry icon. The BC is shown above the main bus line as well, in order to distinguish it from other LRUs. On each LRU and the BC, there's an integer that expresses the deviation of its current measured TDR rate from its respective average. See next section for more information about the [Info panel](#).

DEVICE CONNECTION AND THE INFO PANEL

In order to install a PhysiMUX, simply connect it to its host computer with a USB cable. In the app, a connection LED indicator is located on a white tab in the right corner of the upper bar. If the app doesn't recognize any PhysiMUX device, the LED will turn red, and a flashing red label "SnS not found" will appear next to it. Following a successful connection, the LED indicator will turn green, and a RESTART button will show up next to it. This RESTART button will clean any previous open and short circuit fault indications from the topology diagram.



Another important element that appears following a successful connection with a PhysiMUX device is the Info Panel. The Info Panel is a large light blue widget located on the upper right corner of the topology diagram. It shows some important metrics, measured by PhysiMUX in real-time for both A and B busses. The following is a list of the metrics displayed in Info Panel:

1. **TDR average** -

TDR is a measure of the deviation of a signal's voltage transition time (also known as the "tail") relative to its nominal time in accordance with MIL-STD-1553. The TDR average is the average TDR value of all messages from all RTs. A significantly high TDR average means that most messages are transmitted poorly and are often linked with circuit faults.

2. **Bus Load** -

Percentage of bus time utilization.

3. **Noise** -

Periodic count of pulses that were too short to be interpreted as valid signals. Noise is often linked with intermittent short circuits.

These 3 metrics are key elements for the diagnosis of a faulted CAN-Bus.

Bus	TDR	Load	Noise
A	691	12%	302
B	658	20%	0

FAULTS DETECTION

Fault detection using the PhysiMUX app is done by looking at the topology diagram and the info panel. When the bus contains faults, the 5 metrics on the info panel are used to classify their type. For instance, unusual TDR average and CAN-low noise may suggest CAN-low wire disconnection on the main bus. The following table enumerates the different types of faults that can be deduced out of the info panel, and characterizes their metrics values.

Fault type	TDR average	Visual indicator on diagram
Main bus wire disconnection	At least 40 ns more than normal	a single or a sequence of LRUs are red
Stub wire disconnection	Slightly higher	an LRU is red
Main bus short between the wires	No update. Bus is dead.	All LRUs are red.
Stub short between the wires	lower than normal	an LRU is red

Understanding the topology diagram:

When a topology is successfully loaded and the PhysiMUX is successfully connected to the host computer, the 1553 Bus and the various LRUs change their color. LRUs may be green, yellow, or red. Green indicates that the LRU transmits messages on every *fault isolation interval** without exceptions. A red LRU means that it did not transmit any message at the current fault isolation interval. If an LRU is yellow, it means that it transmitted a message on the current fault isolation interval, but did not transmit any message on some previous intervals. Once the metrics on the info panel suggests some kind of fault, these color changes on the topology diagram can help locate them. For example, if the metrics suggest a stub 1553 disconnection, and there is an LRU that is constantly red, it might mean that this LRU is disconnected from the bus. If the metrics suggest a 1553 disconnection on the main bus and the diagram shows a sequence of red LRUs, the fault may be between the first red LRU in the sequence and last green [or yellow] LRU relative to the position of the SnS.

**Faults isolation interval - a period in which all messages are stored in the SnS device for a learning algorithm to isolate received messages from the knowingly unreceived messages. The interval rate is configured in the topology file.*

SHORT CIRCUIT DETECTION

Short circuit detection by the app is an internally automated process. The shorts that can be detected are:

1. Main Bus short – These shorts take down the bus. As a result of such a short, only short pulses exist on bus. Their length is displayed in the Noise column. The farther away the short is, the larger the number. There is no direct translation from the ‘Noise’ number and the location on the bus. Users may choose to perform a few shorts on the bus with varying distances from the BC, and record the Noise value.
2. Stub Short, or Coupler Short – these shorts do not take the bus down, but actually reduces the average TDR quite significantly. The combination of reduced values of average TDR and no response from LRUs, provides a direction of where the short is.

Bus	TDR	Load	Noise
A	691	12%	302
B	658	20%	0