MTX Analyzer

This piece of software will analyze the signal structure of TCMS SIL2 safe-applications.

How to use this program

1. Place all the safe-application .mtx files into the **mtx_files** folder.

> Example:
>
> > LOT_CCUS_SA4_Door.mtx
> >
> > - This file contains all the safe-signals and corresponding safe-connections for the SA4 application.

2. Place the 'types.mtx' file into the **mtx_files** folder, next to the SA mtx files.

> types.mtx represents the external interface of a given Safe Application

Description

- When executed, this tool will look for all Safe-application files

  These files are categorised with the corresponding project and SA number.

Functionalities

In this section it will be described the software work-flow as well as possible console output messages.

- Ability to detect any 'not connected' safe-signal within all the safe applications;

> Output:
>
> > - "[!] [NC] Signal_name "
> >
> > in case of a non connected signal.

- In case of a core application, the program will detect different safe-signal pool sizes, e.g., a signal is present for LOT and not for EA.

> Output: (These are raised as an info and do not represent an error)
>
> > - "Different signal pool size detected!"
> > - Safe-signal pool size (number of safe-signals) for all the projects for a given safe application.

- The application is able to detect whether the same safe-signal across different projects has a different safe-connection pool.

> Output:

> - INFO "Different safe-connection signal pool size" (If the number of safe-connections are different)
> - INFO "Different safe-connection signals for safe-signal" (If the number of safe-connections are the same, but different safe-connection names)
> - A list of safe-signals that do not share the same connection interface across all projects.

- Ability to verify 32-bit alignment for the internal interface of each safe application.

> Output:
>
> - ERROR [!] "Invalid alignment detected for **SAFE_APPLICATION** at **SIGNAL_NAME**"

- During the previous process, the software will also detect whether a given safe application internal interface has the correct reserved ending format

> Output:
>
> - ERROR [!] "Ending of the member structure in **SAFE_APPLICATION** does not meet the standard."

- Ability to detect whether a signal on the internal interface has any safe-connection

`For each valid signal on the safe application internal interface, the software will verify if any safe application has a corresponding safe-connection member`

> Output:
>
> - ERROR [!] "No connection found for signal: **SIGNAL_NAME**"

Reporting

- At the moment, all the console output will be exported into a TXT file.

Future Work

- Ability to export the results to an excel spreadsheet
- Ability to view the results in a html

Developed at Critical Software, Coimbra, Portugal