

CAN REAL-TIME FAULT INSERTION UNIT

BENEFITS

- Deterministic operation with manual or programmatic interface, suitable for integration into hardware-in-the-loop rigs.
- Digital trigger for deterministic fault insertion and duration control.
- Comprehensive real-time fault insertion:
 - Bit inversion.
 - Incorrect IDE or DLC.
 - Missing frames.
 - Duplicated frames.
 - Physical layer faults (switch to battery/ground).
 - Randomised bit errors.
- Configure over CAN or Ethernet.
- Up to 1Mb baud rate.
- Extend the capability of your existing hardware-in-the-loop rig.
- Save time and money.



SYSTEM OVERVIEW

A Controller Area Network (CAN bus) is a robust vehicle bus standard designed to allow microcontrollers and devices to communicate with each other via a distributed control network. Today automotive vehicle design features many different CAN nodes controlling multiple applications including those that are safety-critical, such as the braking system. A failure in a computer system that controls such an application can not only lead to significant economic losses but even the loss of life. Therefore the need to test for, identify and mitigate the cause of any potential failure is critical.

The Real-Time CAN Fault Insertion Unit (CAN RT FIU) not only gives non-intrusive, deterministic, frame by frame control over fault insertion, but also allows the user to trigger faults in response to outside stimulus (via the digital trigger lines) and control the duration for which these faults are "engaged". Having this level of control is essential for reproducing time critical edge-case conditions which form the majority of uncaught CAN errors on the bus. Good design tools have improved fault tolerance and edge-case detection, but sub-millisecond timing can introduce a whole other level of error detection requirements.

The CAN RT FIU can also implement random fault injection, for testing strategies where the fault causes are unknown. On-chip PSNGs (Pseudorandom number generators) allow randomised injection of faults in the bus messages, but also allow the user to work back to the source when errors are found.

CAN REAL-TIME FAULT INSERTION UNIT

TEST SYSTEM FEATURES

The CAN RT FIU features 3 operating modes:

- Mode 1: Pass-through. The CAN bus is passed directly through the unit and is unmanipulated.
- Mode 2 - Bit Manipulation. An error is inserted into the frame and passed to the receiving node.
- Mode 3 - Bit Manipulation and Correction (CRC). An error is inserted into the frame, however, the cyclic redundancy check (CRC), the checksum that CAN data frames and remote frames contain as a safeguard, is then also corrected prior to sending to the receiving node.

STANDARD FAULT INSERTION

In addition, to direct CAN message manipulation, the unit also offers more standard fault insertion features (disconnection, switch to battery, switch to ground etc.). Being able to physically fault the bus, combined with the ability to test the lowest abstraction layer means the CAN RT FIU is able to offer hardware-in-the-loop test across physical, software and application domains.

HARDWARE SPECIFICATION

The CAN RT FIU is housed in a 19inch enclosure making it simple to accommodate within most test rigs.

It features:

- 2x CAN Ports for connection into CAN network under test.
- 2x Trigger channels.
- 2x Trace channels.
- LCD screen for configuration information.
- Ethernet RESTful API and/or separate CAN Port for configuration and HIL integration.
- Standard and extended frame CAN (CAN FD future upgrade).
- Up to 1M baud rate.

SOFTWARE SPECIFICATIONS

System requirements

No software is required (other than for initial config), but if the user wishes to use the Windows UI, Windows 7 or 10 is recommended, with 4GB RAM and standard laptop or desktop processor.

Configuration options

- Two ways of using the system; desktop and API mode:
 - Desktop mode: Configure the system from the UI and select the type of fault and other fault configuration options.
 - API mode: Programmatically use a RESTful API Ethernet call or use the onboard CAN configuration channel to setup the fault and set trigger options.
- Example code is provided for making configuration calls from LabVIEW (RESTful API and CAN examples) and Python (RESTful API calls example).
- Faults (via the above configuration options) can be configured to be introduced on a digital trigger event, or a specific time after that event, and can be held "faulted" for a specified duration.

CUSTOMISATION

This extensible unit is modifiable to accommodate specific test and integration requirements. Please, contact our team for further information.