Journal of Advanced Research in Computer Graphics and Multimedia Technology
Volume 1, Issue 1 - 2019, Pg. No. 00-00
Peer Reviewed Journal

Article

Implementation of Automatic Functional Testing Tool using CanSniff for Automotive Applications

KSSM Reddy[1] Abhishek Shukla[2]

I N F O

Corresponding Author:KSSM Reddy

E-mail Id:kssmreddy@gmail.com

How to cite this article:
Shukla A, Reddy KSSM. Implementation of Automatic Functional Testing Tool using CanSniff for Automotive Applications *J Adv Res Comp Graph Multim Tech* 2019; 1(1): 19-22.

Date of Submission: 2019-04-02
Date of Acceptance: 2019-05-08

A B S T R A C T

The goal of this work is to create a Test Tool that automates all of the manual work. By using this tool, one can actually create and run test cases automatically. This tool automatically reads the commands for read/write Shadow Variables, Application Variables, Application Cans, physical signals, etc. from the specified file and executes the required commands without any user intervention, and hence helps to automate the entire process of manually testing every step. Using the CanSniff tool, one can actually read/write the contents of a memory (EEPROM) location within an ECU. This tool also allows the user to include delays before executing any command thereby providing time to the user to observe things of how they are working out.

This tool also integrates one more tool called "Harness Configuration Utility", which has the functionality to map the controller pins to the hardware and provide alias names to the mapped pins. It also has a scaling function which provides a conversion of physical units to engineering units.

Keywords: CanSniff, Automatic Functional Testing for Automotive Industry, Console Application, ECU, LabVIEW, Virtual instrumentation

Introduction

The tool automatically reads the commands for read/write Shadow Variables, Application Variables, Application Cans, physical signals etc. from the specified file and executes the required commands without any user intervention, and hence helps to automate the entire process of manually testing every step. Using the CanSniff tool, one can actually read/write the contents of a memory (EEPROM) location within an ECU. The output data pertaining to a particular command is basically compared with the expected output and if it matches, then the result is said to have passed for that particular command. Similarly, this kind of comparison is performed on all of the commands that are there in that Test Case, and if each command result is pass then we can say that the entire test case has passed, otherwise it has failed. This tool allows the user to create new or import/ export the existing work configuration file and also Test Cases within the work. This tool allows the user to select his/ her own Custom Control Style based on his/ her choice from three different custom control styles that are available, for controlling the Digital/ Analog I/O of a Controller. This tool also allows the user to include delays before executing any command thereby providing time to the user to observe things of how they are working out.

This tool also integrates one more tool called "Harness Configuration Utility", which basically provides a user friendly GUI for importing ECU pin configurations and mapping them to DAQ hardware depending upon the measurement type, i.e. one to one mapping of controller pins depending upon whether its an analog I/O or digital I/O etc.

The Harness Configuration Utility will have the functionality

to map the controller pins to the hardware and provide alias names to the mapped pins. It will also have a scaling function which will provide a conversion of physical units to engineering units. The configuration module will generate a "Harness configuration table" of mapped controller pins which will be input for HIL configuration utility.

Introduction of CANSNIFF

CanSniff allows you to monitor and interact with a CAN network in many possible ways, collecting data and metrics, as well as sending messages to the network for a variety of reasons.

CanSniff is a 32 bit "console" application. In the old days, this would be called a DOS application (and it would be a 16 bit program). CanSniff is configured primarily with its command line options. It is possible to run it from a windows shortcut, but simply clicking on the CanSniff.exe icon does not let you set the many parameters that make it the powerful tool that it is.

Architecture Description

As depicted in fig. 1, all inbound messages are passed into the decode module first, which examines the type of message. Based on the message type, it can be routed to the filters or the Console interface, or it could be decoded as a trigger causing either closure of the log file or initiation of a playback file. Also in the decode block, network stats are tallied and made available via the user interface manager.

- The filters are then serialized, with first the Notch filters, followed by the Pass filters. These are then passed to the user interface manager as well as the log block.
- Logged activity can be either streamed to disk, or buffered locally and then only the buffer is written to disk.
- The keyboard/ display can access the network messages, or it can be used to access internal information-statistics and so forth.
- Playback files can be initiated from the command line or triggered from the configuration of a trigger message in the decode block.
- Any outbound messages are internally looped back

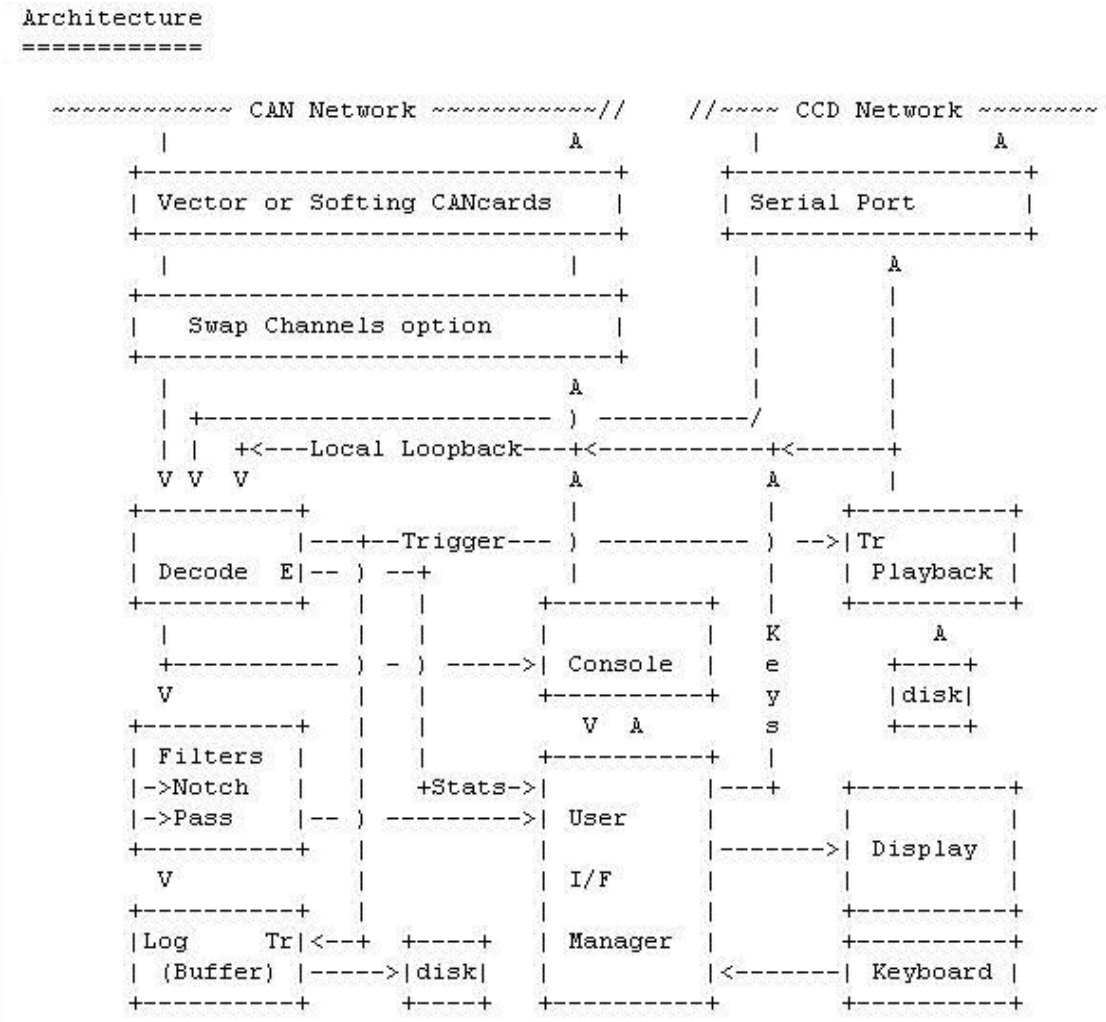

Figure 1.Architecture Description of CANSniff

so those can be logged as well as accumulated as network statistics.

Introduction to Console Application

A console application is a computer program designed to be used via a text- only computer interface, such as a text terminal, the command line interface of some operating systems or the text- based interface included with some Graphical User Interface operating systems, such as the Win32 console in Microsoft Windows. A user typically interacts with a console application using only a keyboard and display screen, as opposed to GUI applications, which normally require the use of a mouse or other pointing device. Many console applications such as command line interpreters are command line tools, but numerous Text User Interface programs also exist.

As the speed and ease- of- use of GUI applications have improved over time, the use of console applications has greatly diminished, but not disappeared. Some users simply prefer console based applications, while some organizations still rely on existing console applications to handle key data processing tasks.

Testing Tool

The step by step application of testing tool is given below:

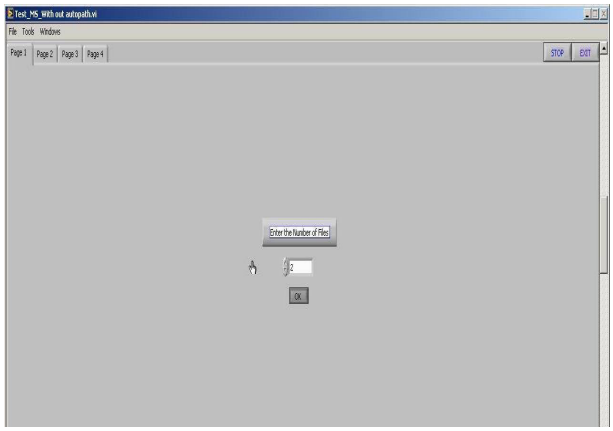

Figure 2.Starting Window

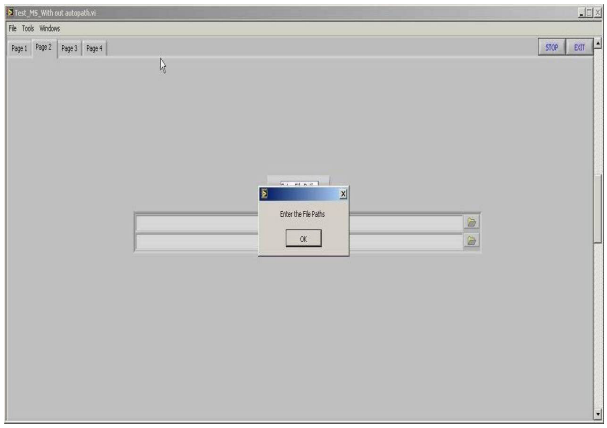

Figure 3.Enter File Paths

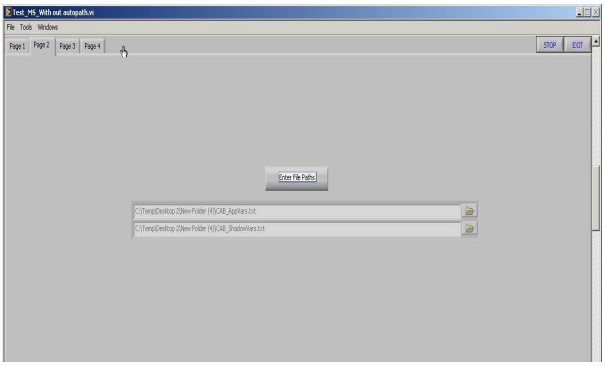

Figure 4.After Providing File Paths

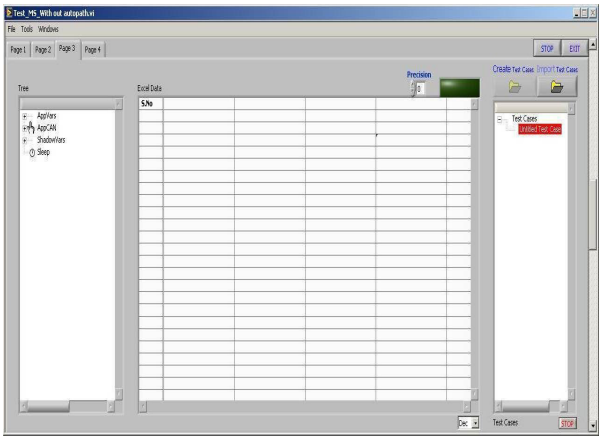

Figure 5.Main Interface

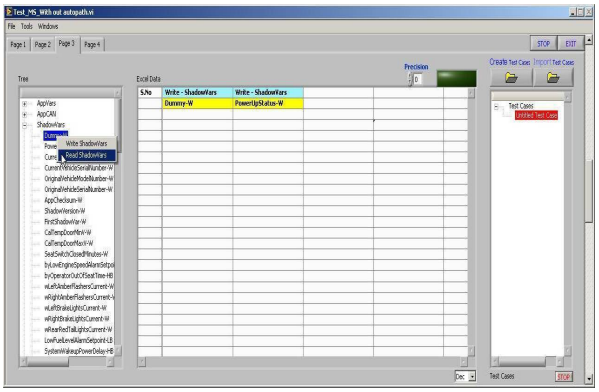

Figure 6.Main Interface after writing some Parameters

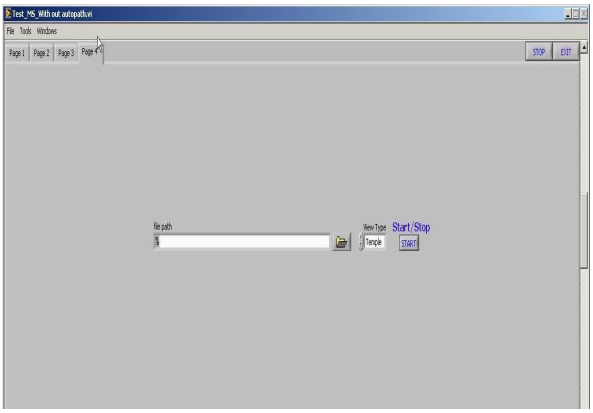

Figure 7.File Path for Harness Configuration File

The purpose of this testing tool is to save a lot of time and efforts put by an individual in making and running test cases. By using this tool, one can create and run test cases in minimum time with less effort.

Before the development of this tool, the creation of test cases and running them was taking a lot of time because of more manual work, and because of this manual work, there was more chance of noting a wrong value into the Test Case which ultimately fails the entire Test Case, resulting in a waste of time and efforts as well. By using this tool, one can eliminate all such problems to the maximum level.

References

1. Sumathi S, Surekha P. LabVIEW Based Advanced Instrumentation Systems. Springers, 1/e, 2007.
2. Johnson GW, Jennings R. LabVIEW Graphical Programming. McGraw-Hill Professional, 2006.
3. Conway J, Watts S. A software engineering approach to LabVIEW. Prentice Hall PTR, May 05, 2003.
4. CanSniff help documents.