International Journal of Innovative Research in Advanced Engineering (IJIRAE) ISSN: 2349-2163
Issue 07, Volume 8 (July 2021) https://www.ijirae.com/archives

ECU EXTRACT ANALYZER AND MERGER TOOL

Nagaveni K S*
Department of ECE, JSS S&T University, Karnataka, India
nagaveniks8@gmail.com
Dr. S Gayathri
Department of ECE, JSS S&T University, Karnataka, India
sgmurthy_65@sjce.ac.in

Publication History
Manuscript Reference No: **IJIRAE/RS/Vol.08/Issue07/JLAE10084**
Research Article | Open Access
Peer-review: Double-blind Peer-reviewed
Article ID: IJIRAE/RS/Vol.08/Issue07/JLAE10084
Received: 19, July 2021
Accepted: 19, July 2021
Published Online: 30, July 2021
Volume 2021 | Article ID JLAE10084 | https://doi.org/10.26562/ijirae.2021.v0807.005
NAGAVENI & Gayathri (2021). ECU EXTRACT ANALYZER AND MERGER TOOL. IJIRAE::International Journal of Innovative Research in Advanced Engineering, Volume VIII, 167-175.
doi: https://doi.org/10.26562/ijirae.2021.v0807.005
Editor-Chief: Dr.A.Arul Lawrence Selvakumar, Chief Editor, IJIRAE, AM Publications, India

Abstract: AUTOSAR XML (.arxml) is a format introduced by the Automotive Open System Architecture consortium to contain the data used in and required by Electric Control Units which is based on Automotive Open System Architecture. Electric Control Unit development is done by 3 groups i.e. OEM like BMW, Tier1like Bosch, and Tier2 like a vector. Each one has its own data that is required for Software development. OEM shares the information with Tier1.Tier1 shares the same with Tier2 for implementing into Electric Control Unit. The DataBase Container (DBC) describes the properties of the Controller Area Network (CAN) / Controllable Area Network Flexible Data Rate (CAN FD). Data analysis in ECU Extract ARXML is very difficult due to the complex structure of data. Performing two-way and three-way comparisons/merges of XML files is difficult. To handle the mentioned challenges manually is a big task that requires huge manual efforts and time which may even result in a manual error. In this paper, we are presenting a newly developed Python-based tool that overcomes the above-mentioned drawbacks.

Keywords: analyzer ecu extract merger tool

I. INTRODUCTION

At present, there are few tools that perform the analysis of ARXML and DBC files, and even for merging and comparison of those files. Few tools are capable of performing only two-way comparisons and there are less few tools that perform three-way comparisons too. In almost all companies they use beyond compare tool for comparison of all types of files but it performs an only two-way comparison, it requires human intervention and performs line by line comparison of these files. In beyond compare it is a text-based comparison. In the dSPACE AUTOSAR Compare tool there are two types:

i) Manual comparison and merging
ii) Rule-based comparison and merging, applicable only for two-way comparisons.

Since XML files contain almost seven lakh lines which is a very tedious task to analyze these files using tools with human intervention and time-consuming. So it is necessary to have end-to-end automated tools without human intervention in between. The objective of the proposed work is to develop a Python-based tool that is carried out at four levels.

(i) Data Analysis of ARXML and DBC file using two different python scripts. The input given to the script for analysis of ECU Extraction is an ARXML file. The analysis of the DBC file is done by vector tools as a result eight CSV files are generated.
(ii) The generated CSV files are given as input to the DBC compare script. From the DBC Compare script the unwanted details which are present in CSV files are removed and generates twelve types of excel sheets in a single excel file namely New Rx Msg, Old Rx Msg, Removed Rx Msg, New Tx Msg, Old Tx Msg, Removed Tx Msg, New Rx Sgn, Old Rx Sgn, Removed Rx Sgn, New Tx Sgn, Old Tx Sgn, Removed Tx Sgn.
(iii) Two-way and three-way comparisons/merges of the XML files.
(iv) END to END automation.

Data analysis in ECU Extract ARXML is very difficult due to complex structure of data. Performing two way and three way comparisons/merges of XML files is difficult. To handle above challenges manually, it is a big task, requires huge manual efforts and time consuming which may even result in manual error. The proposed tool overcomes the above challenges and is completely automated. The basic GUI is developed to select the tasks to be performed. It takes the input from user end and to intimate the missing files. If some file is missing to perform the DBC compare analysis, the corresponding file names will be popped up. After analysis every time, the generated output will be stored in particular folder path.

Some messages and signals take the path of KCAN and some takes the path of both KCAN and BCM SUBCAN. While generating CSV files from DBC two set of CSV files one for KCAN and other for SUBCAN are generated. These two type of files are input to DBC Compare tool and generates two Excel file consisting of twelve excel sheets. The developed GUI application is simple GUI. It takes the input from user and performs the required operation like DBC, ECU Extract or merging. In DBC Compare if some files are not present in project folder which is needed for the operation, then that file name will be popped up as a message dialogue box. This can also be made even more user friendly by popping up the generated excel file instead of searching it in project folder as in some of the standard tools. The message box can be created at the beginning to instruct the user what to do and not to do to get the accurate output without error.

II. LITERATURE SURVEY

Many investigations are carrying out to generate Best end to end automated tool. The generated tool can read only a small subset of the objects which may be contained in an ARXML [1]. This tool is not aware of XML files (instead, it looks for hard-coded keywords in the XML), so if importing of AUTOSAR is not working with certain files, then it is needed to use ARXML-to-DBC file converter. The dSPACE AUTOSAR Compare tool which can compare and merge only two files is known as two way comparison [2]. It compares the elements properties and child Elements in a tree. Filters are used to detect the differences. Merging can be either rule based or comfortable merging. Beyond Compare tool, which allows quick and easy compare of files and folders is discussed as in [3]. Three way merging of files and synchronize of folders are possible but it includes human intervention and line by line comparison which is time consuming. Comparison of XML files with highlighted differences in side by side windows with merging in either direction is presented in [4]. Even three ways comparison is possible but not end to end automated and it is a line by line comparison. At present the tool discussing in this paper is better compare to others.

III. METHODOLOGY

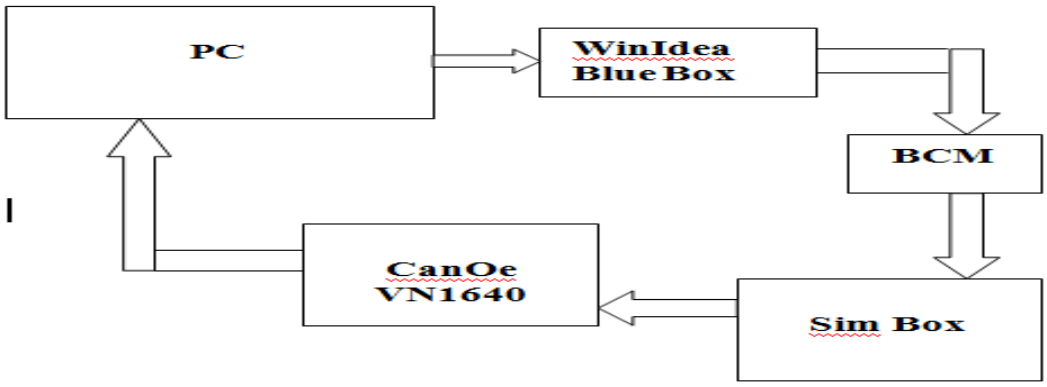


Fig 1. Hardware Connection Block Diagram

WinIdea: It is a debugger. It is supplied by Isystem. JTAG protocol is used for communication.

BCM: MCU used is ST MICRO Chorus 4M-SPC58EC80E7G0C0X. Total i/o pins is 192 which includes power supply too. In CAN Out of three channels two is used. In LIN out of 15 channels eight is used. System base chip includes power supply and few driver outputs from ST MICRO

CanOe: This tool is mainly used for simulation.VN1640 bus driver is used. This device has 2 CAN and 2 LIN provision.

Simbox: In-house manufactured for simulating the Physicals Switch and Outputs.

Fig 2 depicts the proposed Functional Block Diagram for testing. It contains components and the process involved in flashing of messages on the ECU Hardware. It mainly consists of 6 blocks,

a) ECU Extract provided by customer: This ECU Extract will be as per AUTOSAR norms and used as input to Generator tool. The Generator tool will be internal to the companies. The requirements from the customer will be in the ARXML format and DBC. It will be difficult to extract and analyze the messages and signals from them. So files are given as input to the proposed tool.

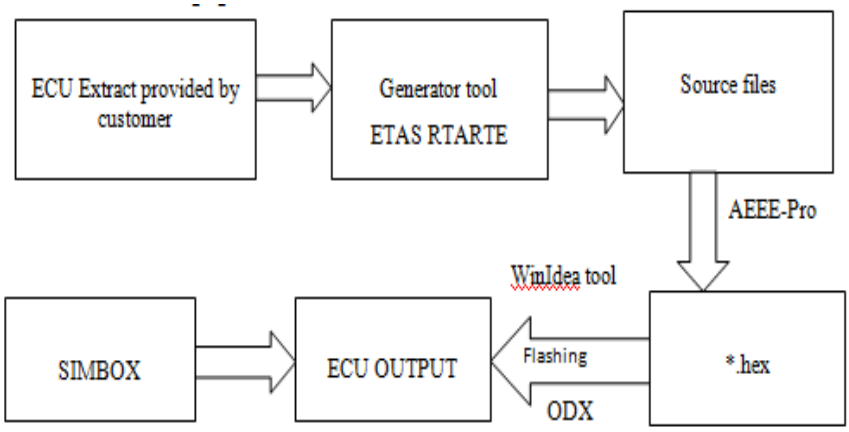


Fig 2. Proposed Functional Block Diagram

b) Generator Tool: ETAS RTARTE is used as generator tool. The messages and signals are given as input to generator tool. The output from the Generator tool known as AUTOSAR ECU values ARXML which will be sent to source file like CUBAS stack or Vector stack or EBtresos stack.
c) Source Files: The output from source code is sent to AEEE-pro tool which generate .C and .H files.
d) .*hex code: .C and .H files are flashed on ECU using ISYSTEM.
e) ECU OUTPUT: After flashing is completed validation message window will be popped.
f) SIMBOX: Once the message and signal flashed on ECU, the output can be noticed on SIMBOX as written in the script.

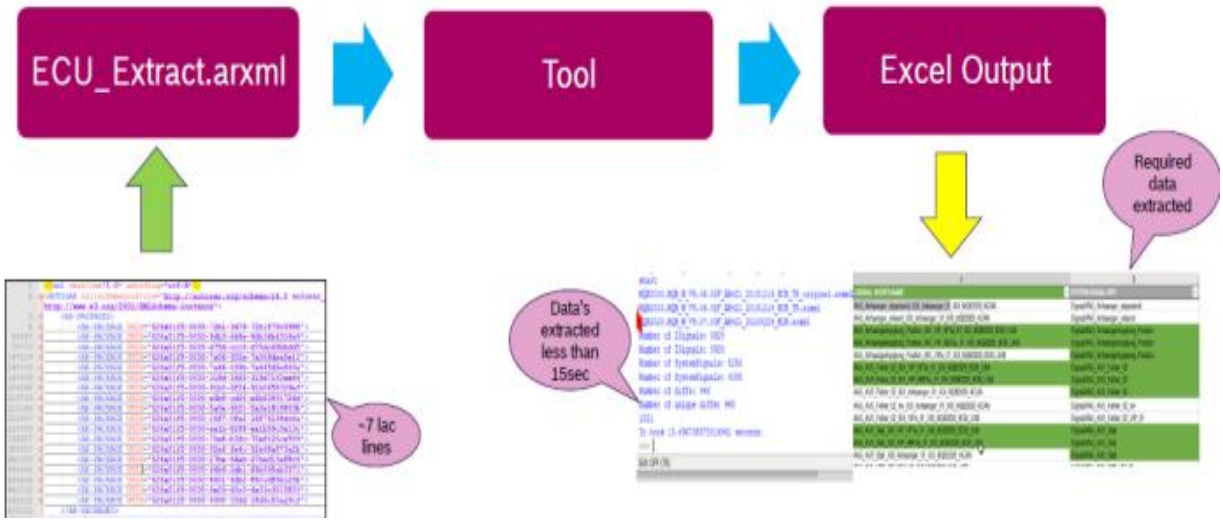


Fig 3. Block Diagram of the tool

Fig 3 shows the block diagram of proposed tool which developed using Python Script. It consists of three blocks i.e. input, processing and output. Input refers to ARXML and DBC file based on the operation to be carried out. The ARXML and DBC file will be very huge data file approximately it consists of 7lac lines. It's very difficult to analyzes and extract the required information. The processing is a tool developed by Python script. The tool can perform three operations namely ECU EXTRACT, DBC and MERGING. The tool consists of basic GUI for user input, selection of operation and to notify the missing files required for the operation. The output refers to excel sheet obtained from the tool. The tool takes only few seconds to generate the analyzed excel sheet with all details related to messages and signals.

IV. IMPLEMENTATION

Python is an interpreter, object-oriented, high-level programming language with dynamic semantics. Its built-in high-level data structure, combined with dynamic writing and dynamic linking, make it very attractive for rapid application development and use as a scripting language or glue for connecting existing components. Python's simple and easy-to-learn syntax emphasizes readability, thus reducing the cost of maintaining programs. Python supports modules and packages, which encourage program modularity and code reuse. The Python interpreter and extensive standard library are provided in source code or binary form, are available for all major platforms, and can be distributed for free.

The algorithm of the generated tool is as follow:

Step 1: Calling all the required modules.

Step 2: Creation of GUI main window with three input radio buttons.

Step 3: Selection of required operation.

Step 4: If DBC radio button is triggered, it looks for all the required files in project folder .While performing the task if any of the files needed at that point is missing, then it pop ups the alert message with the name of the missing file. If click on the OK button, it continues to seek the other files required for the next task and performs the operation with available files by leaving the task of input files missing.

Step 5: If ECU EXTRACT radio button is triggered, then GUI Top level window is generated.

Step 6: Asks the user to provide the three input ARXML files required to perform the task.

Step 7: In files one will be considered as modified, new and original ARXML files. Based on taglines it compares the original file short name and modified short name they must be equal and compares their system signal ref which must be different and it increases the modified count.

Step 8: Short name of both modified and new file will be checked and it must be equal. The system signal ref of New and modified must be equal. If not also for system signal ref of new file, the system signal ref of modified file is passed and increases the count of merging signal.

Step 9: If merging radio button is triggered, then from step 4-step 8 is repeated and from the generated output of DBC and ECU_EXTRACT the comparison of messages and signals takes place. It fetches all the information related to those messages and signals. It start analyzing if message and signal is present in DBC but not found in ECU_EXTRACT means it displays in the ECU_EXTRACT column of that message as not found in ECU_EXTRACT.

Step 10: If there is controversy in same message and signal name in ECU_EXTRACT and DBC such as in one it will be new and in other it will be old in that case it will highlighted.

V. RESULTS

Fig 4 shows the basic GUI asking users to trigger the required operation to perform. If user selects the DBC it performs the DBC analysis by fetching the related files to perform the task and produces the outputs in two ways.

Fig 4. Basic GUI

One is in python shell as shown in the Fig 5.It gives the count of messages and signals present in twelve excel files. In Fig5 it starting line It took 0.0 seconds indicates that once DBC button is triggered without delay it started it operation and It took 13.69 seconds shows the time taken to count the number of messages and signals and printing the same in python shell and to generate the excel sheet. The other output in Excel sheet as shown in the Fig 6.Generally the excel sheet will be created in project folder. Fig 6 shows the message name of first excels sheet i.e. New_Rx_Msg. Similarly there will be twelve Excel sheet

(i) New_Rx_Msg- Contains the new received Message name. This message name list is obtained by comparing New_Rx_Msg.csv and Old_Rx_Msg.csv .If message name is not in Old_Rx_Msg.csv than it is considered as New_Rx_Msg.

(ii) Removed_Rx_Msg- Contains the removed received message name. This message name list is obtained by comparing New_Rx_Msg.csv and Old_Rx_Msg.csv. If message name is not in New_Rx_Msg.csv than it is considered as Removed_Rx_Msg.

(iii) Changed_Rx_Msg- Contains the changed received message. This message name list is obtained by comparing New_Rx_Msg.csv and Old_Rx_Msg.csv. The message name will be present in both New_Rx_Msg.csv and Old_Rx_Msg.csv but some datas will be from Old_Rx_Msg.csv to New_Rx_Msg.csv.

```
It took 0.0 seconds.
----------------------------------------
Newly added Rx message : 30
----------------------------------------
Removed Rx message : 41
----------------------------------------
Changed Rx message : 161
----------------------------------------
----------------------------------------
Newly added Tx message : 8
----------------------------------------
Removed Tx message : 9
----------------------------------------
Changed Rx message : 173
----------------------------------------
----------------------------------------
Newly Added Rx Signal : 161
----------------------------------------
----------------------------------------
Removed Rx Signal : 116
----------------------------------------
----------------------------------------
Changed Rx Signal : 74
----------------------------------------
----------------------------------------
Newly Added Tx Signal : 119
----------------------------------------
----------------------------------------
Removed Tx Signal : 48
----------------------------------------
----------------------------------------
Changed Tx Signal : 41
----------------------------------------
END
It took 13.692799806594849 seconds.
```

Fig 5. DBC analysis output in python shell

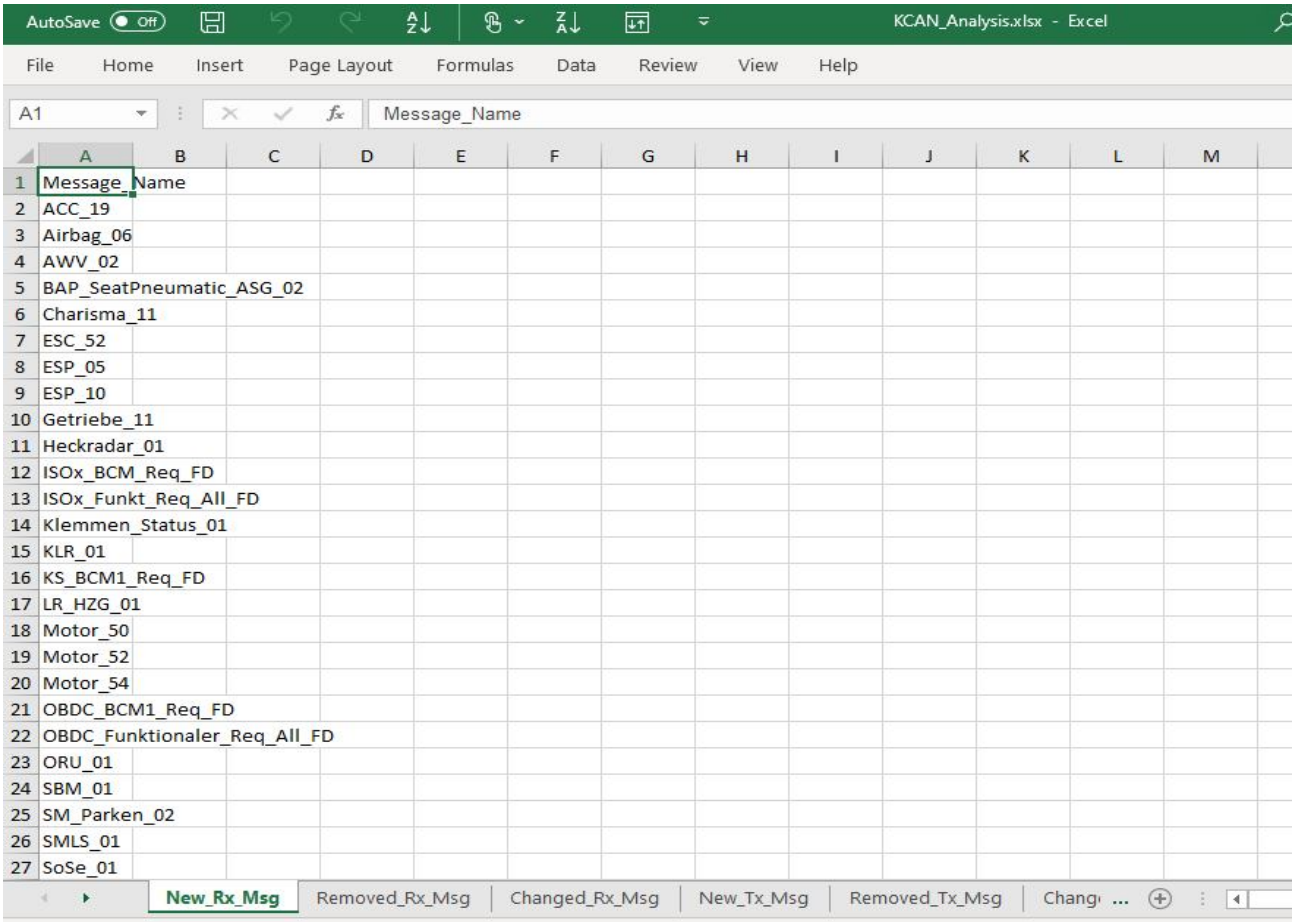


Fig 6. DBC analysis output in Excel sheet

(iv) New_Tx_Msg- Contains the new transmitted Message name. This message name list is obtained by comparing New_Tx_Msg.csv and Old_Tx_Msg.csv .If message name is not in Old_Tx_Msg.csv than it is considered as New_Tx_Msg.

(v) Removed_Tx_Msg- Contains the removed transmitted message name. This message name list is obtained by comparing New_Tx_Msg.csv and Old_Tx_Msg.csv. If message name is not in New_Tx_Msg.csv than it is considered as Removed_Tx_Msg.

(vi) Changed_Tx_Msg-Contains the changed transmitted message. This message name list is obtained by comparing New_Tx_Msg.csv and Old_Tx_Msg.csv. The message name will be present in both New_Tx_Msg.csv and Old_Tx_Msg.csv but some datas will be from Old_Tx_Msg.csv to New_Tx_Msg.csv.

(vii) New_Rx_Sgn

(viii) Removed_Rx_Sgn

(ix) Changed_Rx_Sgn

(x) New_Tx_Sgn

(xi) Removed_Tx_Sgn

(xii) Changed_Tx_Sgn

If user triggers ECU_EXTRACT it asks the user to select three files which are known as original, modified and new ARXML files using basic GUI as shown in Fig 7.

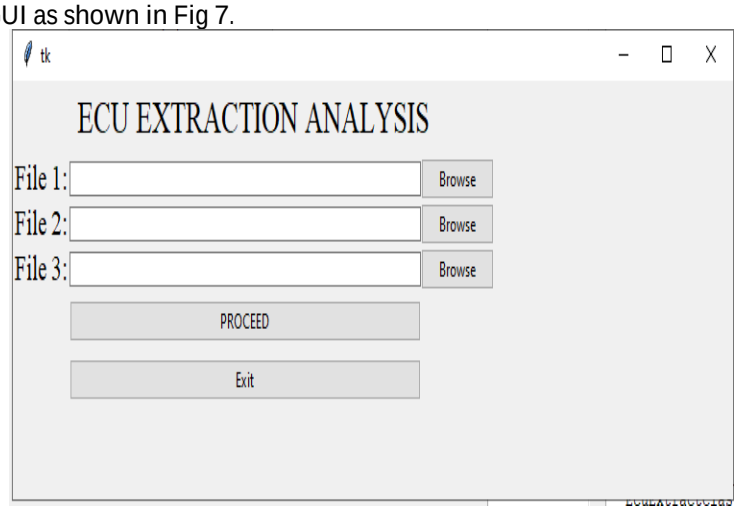


Fig 7. GUI to select files required for ECU EXTRACT

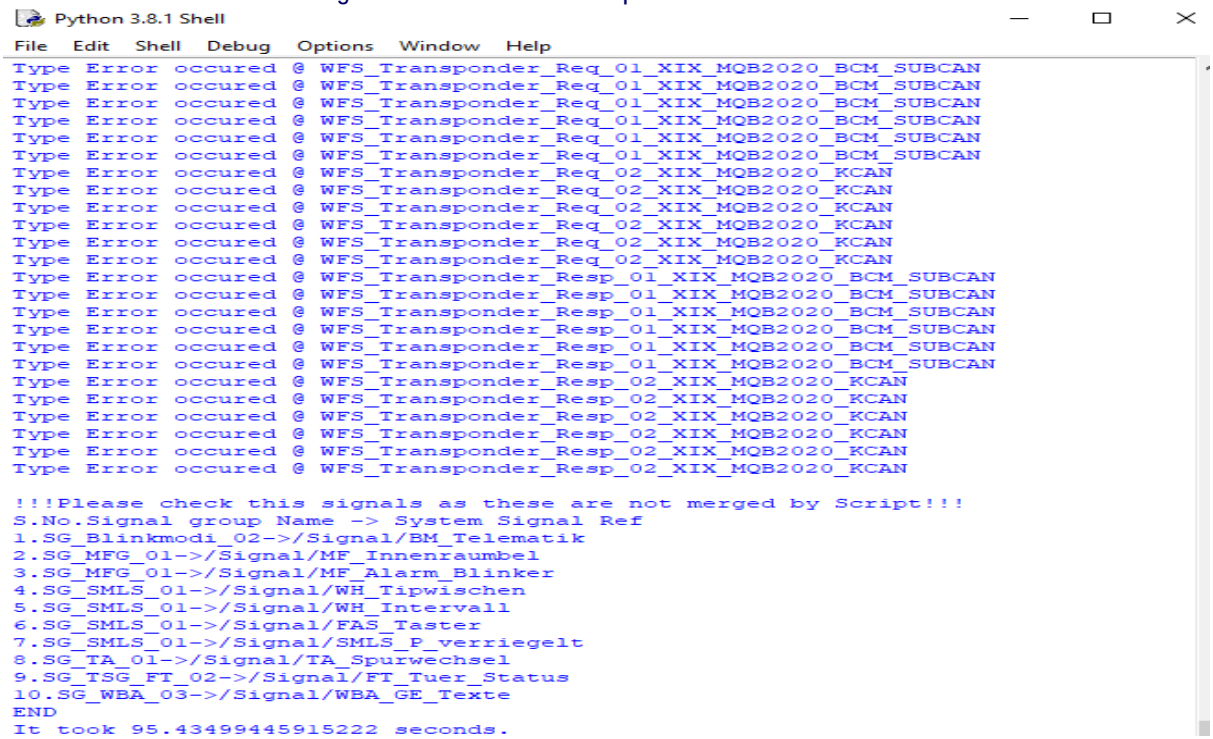
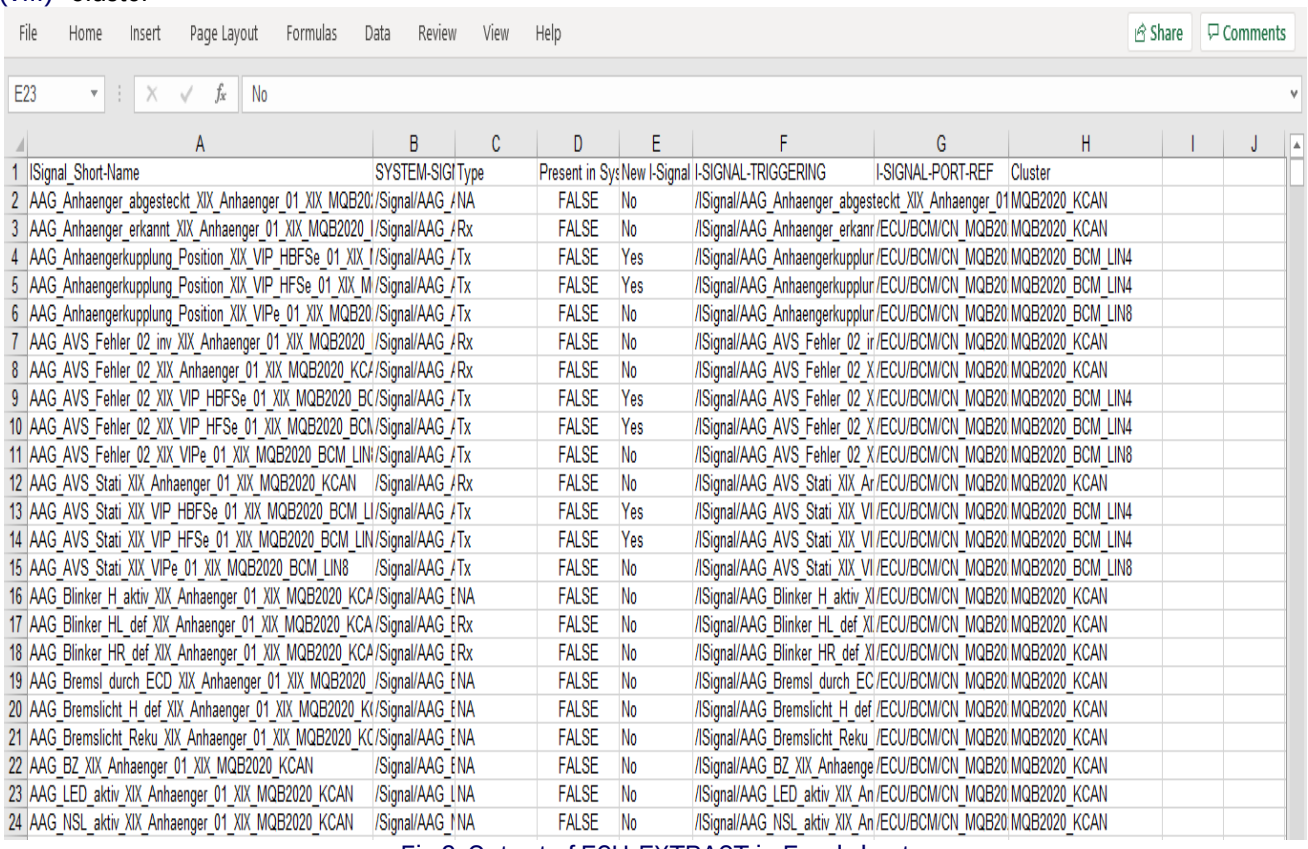


Fig 8. ECU EXTRACT Output in Python shell

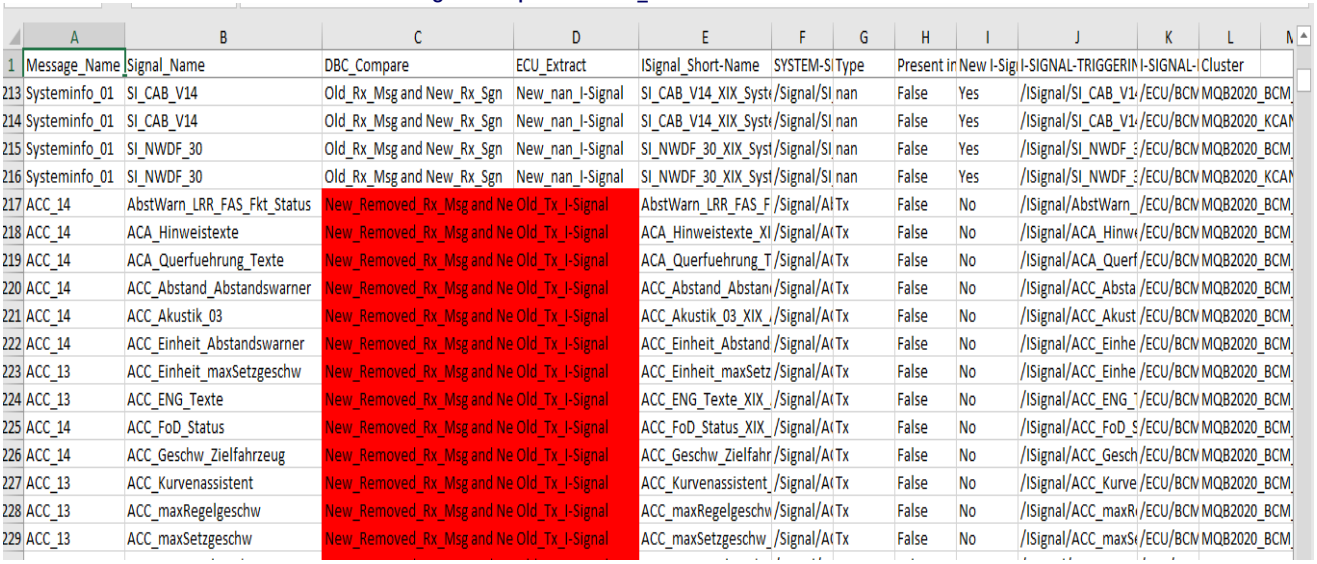
It performs the analysis of ARXML files and generates the output in two ways as mentioned for DBC analysis. It hardly takes 1-2 min to analyze 7lac lines which is depicted in Fig 8. Fig 9 shows the output of ECU_EXTRACT in Excel sheet. The excel sheet consists of eight columns namely

(i) ISignal_Short-Name-It includes message name ,signal name and type of bus path it choose. The path may be BCM_LIN,BCM_SUBCAN and KCAN.
(ii) System-Signal
(iii) Type
(iv) Present in System signal
(v) New I-Signal
(vi) I-Signal-TRIGGERING
(vii) I-SIGNAL-PORT-REF
(viii) Cluster



	A	B	C	D	E	F	G	H	I	J
1	ISignal_Short-Name	SYSTEM-SIGI	Type	Present in Sys	New I-Signal	I-SIGNAL-TRIGGERING	I-SIGNAL-PORT-REF	Cluster		
2	AAG_Anhaenger_abgesteckt_XIX_Anhaenger_01_XIX_MQB20	/Signal/AAG_A	NA	FALSE	No	/ISignal/AAG_Anhaenger_abgesteckt_XIX_Anhaenger_01	MQB2020_KCAN			
3	AAG_Anhaenger_erkannt_XIX_Anhaenger_01_XIX_MQB2020_I	/Signal/AAG_A	Rx	FALSE	No	/ISignal/AAG_Anhaenger_erkannt	/ECU/BCM/CN_MQB20	MQB2020_KCAN		
4	AAG_Anhaengerkupplung_Position_XIX_VIP_HBFSe_01_XIX_I	/Signal/AAG_A	Tx	FALSE	Yes	/ISignal/AAG_Anhaengerkupplur	/ECU/BCM/CN_MQB20	MQB2020_BCM_LIN4		
5	AAG_Anhaengerkupplung_Position_XIX_VIP_HFSe_01_XIX_M	/Signal/AAG_A	Tx	FALSE	Yes	/ISignal/AAG_Anhaengerkupplur	/ECU/BCM/CN_MQB20	MQB2020_BCM_LIN4		
6	AAG_Anhaengerkupplung_Position_XIX_VIPe_01_XIX_MQB20	/Signal/AAG_A	Tx	FALSE	No	/ISignal/AAG_Anhaengerkupplur	/ECU/BCM/CN_MQB20	MQB2020_BCM_LIN8		
7	AAG_AVS_Fehler_02_inv_XIX_Anhaenger_01_XIX_MQB2020_	/Signal/AAG_A	Rx	FALSE	No	/ISignal/AAG_AVS_Fehler_02_ir	/ECU/BCM/CN_MQB20	MQB2020_KCAN		
8	AAG_AVS_Fehler_02_XIX_Anhaenger_01_XIX_MQB2020_KCA	/Signal/AAG_A	Rx	FALSE	No	/ISignal/AAG_AVS_Fehler_02_X	/ECU/BCM/CN_MQB20	MQB2020_KCAN		
9	AAG_AVS_Fehler_02_XIX_VIP_HBFSe_01_XIX_MQB2020_BC	/Signal/AAG_A	Tx	FALSE	Yes	/ISignal/AAG_AVS_Fehler_02_X	/ECU/BCM/CN_MQB20	MQB2020_BCM_LIN4		
10	AAG_AVS_Fehler_02_XIX_VIP_HFSe_01_XIX_MQB2020_BCM	/Signal/AAG_A	Tx	FALSE	Yes	/ISignal/AAG_AVS_Fehler_02_X	/ECU/BCM/CN_MQB20	MQB2020_BCM_LIN4		
11	AAG_AVS_Fehler_02_XIX_VIPe_01_XIX_MQB2020_BCM_LIN	/Signal/AAG_A	Tx	FALSE	No	/ISignal/AAG_AVS_Fehler_02_X	/ECU/BCM/CN_MQB20	MQB2020_BCM_LIN8		
12	AAG_AVS_Stati_XIX_Anhaenger_01_XIX_MQB2020_KCAN	/Signal/AAG_A	Rx	FALSE	No	/ISignal/AAG_AVS_Stati_XIX_Ar	/ECU/BCM/CN_MQB20	MQB2020_KCAN		
13	AAG_AVS_Stati_XIX_VIP_HBFSe_01_XIX_MQB2020_BCM_LI	/Signal/AAG_A	Tx	FALSE	Yes	/ISignal/AAG_AVS_Stati_XIX_VI	/ECU/BCM/CN_MQB20	MQB2020_BCM_LIN4		
14	AAG_AVS_Stati_XIX_VIP_HFSe_01_XIX_MQB2020_BCM_LIN	/Signal/AAG_A	Tx	FALSE	Yes	/ISignal/AAG_AVS_Stati_XIX_VI	/ECU/BCM/CN_MQB20	MQB2020_BCM_LIN4		
15	AAG_AVS_Stati_XIX_VIPe_01_XIX_MQB2020_BCM_LIN8	/Signal/AAG_A	Tx	FALSE	No	/ISignal/AAG_AVS_Stati_XIX_VI	/ECU/BCM/CN_MQB20	MQB2020_BCM_LIN8		
16	AAG_Blinker_H_aktiv_XIX_Anhaenger_01_XIX_MQB2020_KCA	/Signal/AAG_E	NA	FALSE	No	/ISignal/AAG_Blinker_H_aktiv_XI	/ECU/BCM/CN_MQB20	MQB2020_KCAN		
17	AAG_Blinker_HL_def_XIX_Anhaenger_01_XIX_MQB2020_KCA	/Signal/AAG_E	Rx	FALSE	No	/ISignal/AAG_Blinker_HL_def_XI	/ECU/BCM/CN_MQB20	MQB2020_KCAN		
18	AAG_Blinker_HR_def_XIX_Anhaenger_01_XIX_MQB2020_KCA	/Signal/AAG_E	Rx	FALSE	No	/ISignal/AAG_Blinker_HR_def_XI	/ECU/BCM/CN_MQB20	MQB2020_KCAN		
19	AAG_Bremsl_durch_ECD_XIX_Anhaenger_01_XIX_MQB2020_	/Signal/AAG_E	NA	FALSE	No	/ISignal/AAG_Bremsl_durch_EC	/ECU/BCM/CN_MQB20	MQB2020_KCAN		
20	AAG_Bremslicht_H_def_XIX_Anhaenger_01_XIX_MQB2020_K	/Signal/AAG_E	NA	FALSE	No	/ISignal/AAG_Bremslicht_H_def	/ECU/BCM/CN_MQB20	MQB2020_KCAN		
21	AAG_Bremslicht_Reku_XIX_Anhaenger_01_XIX_MQB2020_KC	/Signal/AAG_E	NA	FALSE	No	/ISignal/AAG_Bremslicht_Reku_	/ECU/BCM/CN_MQB20	MQB2020_KCAN		
22	AAG_BZ_XIX_Anhaenger_01_XIX_MQB2020_KCAN	/Signal/AAG_E	NA	FALSE	No	/ISignal/AAG_BZ_XIX_Anhaenge	/ECU/BCM/CN_MQB20	MQB2020_KCAN		
23	AAG_LED_aktiv_XIX_Anhaenger_01_XIX_MQB2020_KCAN	/Signal/AAG_L	NA	FALSE	No	/ISignal/AAG_LED_aktiv_XIX_An	/ECU/BCM/CN_MQB20	MQB2020_KCAN		
24	AAG_NSL_aktiv_XIX_Anhaenger_01_XIX_MQB2020_KCAN	/Signal/AAG_N	NA	FALSE	No	/ISignal/AAG_NSL_aktiv_XIX_An	/ECU/BCM/CN_MQB20	MQB2020_KCAN		

Fig 9. Output of ECU_EXTRACT in Excel sheet



	A	B	C	D	E	F	G	H	I	J	K	L
1	Message_Name	Signal_Name	DBC_Compare	ECU_Extract	ISignal_Short-Name	SYSTEM-SI	Type	Present in	New I-Sig	I-SIGNAL-TRIGGERIN	I-SIGNAL-I	Cluster
213	Systeminfo_01	SI_CAB_V14	Old_Rx_Msg and New_Rx_Sgn	New_nan_I-Signal	SI_CAB_V14_XIX_Syst	/Signal/SI	nan	False	Yes	/ISignal/SI_CAB_V1	/ECU/BCM	MQB2020_BCM
214	Systeminfo_01	SI_CAB_V14	Old_Rx_Msg and New_Rx_Sgn	New_nan_I-Signal	SI_CAB_V14_XIX_Syst	/Signal/SI	nan	False	Yes	/ISignal/SI_CAB_V1	/ECU/BCM	MQB2020_KCAN
215	Systeminfo_01	SI_NWDF_30	Old_Rx_Msg and New_Rx_Sgn	New_nan_I-Signal	SI_NWDF_30_XIX_Syst	/Signal/SI	nan	False	Yes	/ISignal/SI_NWDF_3	/ECU/BCM	MQB2020_BCM
216	Systeminfo_01	SI_NWDF_30	Old_Rx_Msg and New_Rx_Sgn	New_nan_I-Signal	SI_NWDF_30_XIX_Syst	/Signal/SI	nan	False	Yes	/ISignal/SI_NWDF_3	/ECU/BCM	MQB2020_KCAN
217	ACC_14	AbstWarn_LRR_FAS_Fkt_Status	New_Removed_Rx_Msg and Ne	Old_Tx_I-Signal	AbstWarn_LRR_FAS_F	/Signal/A	Tx	False	No	/ISignal/AbstWarn_	/ECU/BCM	MQB2020_BCM
218	ACC_14	ACA_Hinweistexte	New_Removed_Rx_Msg and Ne	Old_Tx_I-Signal	ACA_Hinweistexte_XI	/Signal/A	Tx	False	No	/ISignal/ACA_Hinwe	/ECU/BCM	MQB2020_BCM
219	ACC_14	ACA_Querfuehrung_Texte	New_Removed_Rx_Msg and Ne	Old_Tx_I-Signal	ACA_Querfuehrung_T	/Signal/A	Tx	False	No	/ISignal/ACA_Querf	/ECU/BCM	MQB2020_BCM
220	ACC_14	ACC_Abstand_Abstandswarner	New_Removed_Rx_Msg and Ne	Old_Tx_I-Signal	ACC_Abstand_Abstan	/Signal/A	Tx	False	No	/ISignal/ACC_Absta	/ECU/BCM	MQB2020_BCM
221	ACC_14	ACC_Akustik_03	New_Removed_Rx_Msg and Ne	Old_Tx_I-Signal	ACC_Akustik_03_XIX_	/Signal/A	Tx	False	No	/ISignal/ACC_Akust	/ECU/BCM	MQB2020_BCM
222	ACC_14	ACC_Einheit_Abstandswarner	New_Removed_Rx_Msg and Ne	Old_Tx_I-Signal	ACC_Einheit_Abstand	/Signal/A	Tx	False	No	/ISignal/ACC_Einhe	/ECU/BCM	MQB2020_BCM
223	ACC_13	ACC_Einheit_maxSetzgeschw	New_Removed_Rx_Msg and Ne	Old_Tx_I-Signal	ACC_Einheit_maxSetz	/Signal/A	Tx	False	No	/ISignal/ACC_Einhe	/ECU/BCM	MQB2020_BCM
224	ACC_13	ACC_ENG_Texte	New_Removed_Rx_Msg and Ne	Old_Tx_I-Signal	ACC_ENG_Texte_XIX_	/Signal/A	Tx	False	No	/ISignal/ACC_ENG_	/ECU/BCM	MQB2020_BCM
225	ACC_14	ACC_FoD_Status	New_Removed_Rx_Msg and Ne	Old_Tx_I-Signal	ACC_FoD_Status_XIX_	/Signal/A	Tx	False	No	/ISignal/ACC_FoD_S	/ECU/BCM	MQB2020_BCM
226	ACC_14	ACC_Geschw_Zielfahrzeug	New_Removed_Rx_Msg and Ne	Old_Tx_I-Signal	ACC_Geschw_Zielfahr	/Signal/A	Tx	False	No	/ISignal/ACC_Gesch	/ECU/BCM	MQB2020_BCM
227	ACC_13	ACC_Kurvenassistent	New_Removed_Rx_Msg and Ne	Old_Tx_I-Signal	ACC_Kurvenassistent_	/Signal/A	Tx	False	No	/ISignal/ACC_Kurve	/ECU/BCM	MQB2020_BCM
228	ACC_13	ACC_maxRegelgeschw	New_Removed_Rx_Msg and Ne	Old_Tx_I-Signal	ACC_maxRegelgeschw	/Signal/A	Tx	False	No	/ISignal/ACC_maxR	/ECU/BCM	MQB2020_BCM
229	ACC_13	ACC_maxSetzgeschw	New_Removed_Rx_Msg and Ne	Old_Tx_I-Signal	ACC_maxSetzgeschw_	/Signal/A	Tx	False	No	/ISignal/ACC_maxS	/ECU/BCM	MQB2020_BCM

Fig 10. Output of MERGING in Excel sheet

If user triggers MERGING it performs both DBC and ECU EXTRACT operation and addition to that generates one more merged excel sheet by analyzing the excel sheet obtained from DBC and ECU EXTRACT. The merged excel sheet is as shown in Fig.10. If the message and signal received from DBC and ECU_EXTRACT have conflicts then indicating columns are highlighted by red as shown in Fig 10.Here conflict refers to same message and signal name in ECU_EXTRACT will be new signal and in DBC will be old signal and vice versa.

VI. CONCLUSION AND FUTURE WORK

In automotive industry message and signal routing is a important and major task. It is a very tedious and time consuming task to analyse the detailed information of the messages and signals from the ARXML file. Even though if there is a tool to analyse those files which is not end to end automated than also it is difficult to analyse those files without any human error and within the less time.

The most commonly used format to analyse the huge data is by tabulating the data in rows and columns one will choose excel sheets. In the present work a tool which analyses the ARXML files is discussed. It performs different kinds of comparison and merging of messages and signals and gives the ready data in Excel format. This is used for routing of those messages and signal on ECU hardware with the help of some other testing tools. The developed tool performs the analysis, comparison and merging tasks based on tagline which is very efficient and easy compared to other existing tool which use line by line and rule based approach. Since there is no human intervention there will be no human error and due to end to end automate there will be huge time saving.

For the future the tool can be made much more automated and user friendly by mentioning the path of created file in the beginning of basic GUI under the selection of task not only the path but also some other TO DO and NOT TO DO. Popping up of created file will also reduce the time of searching that file.

REFERENCES

1. www.mkt-sys.de
2. https://www.dspace.com/en/inc/home/products/sw/system_architecture_software/dspace_autosar_compare.cfm
3. https://www.scootersoftware.com/features.php?zz=features_focused
4. https://www.altova.com/xmlspy-xml-editor
5. Domenico Amalfitano. Information Extraction from Legacy Spreadsheet-based Information System, DATA2014-3rdInternational Conference on Data Management Technologies and Applications.
6. HuangBo, DongHui, WangDafang and ZhaoGuifan. Basic Concepts on AUTOSAR Development, 2010 International Conference on Intelligent Computation Technology and Automation.
7. Michael Roy Moore, Robert A. Bridges, Frank L. Combs, and Adam L. Anderson . Data-Driven Extraction of Vehicle States From CAN Bus Traffic for Cyberprotection and Safety, Digital Object Identifier 10.1109/MCE.2019.2928577 Date of current version 25 October 2019.
8. Clinton Young, Joseph Zambreno. Survey of Automotive Controller Area Network Intrusion Detection Systems 2168-2356 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications_standards/publications/rights/index.html for more information.
9. Jordan Svoboda,Joseph Zambreno. Towards Reverse Engineering Controller Area Network Messages Using Machine Learning, Authorized licensed use limited to: Middlesex University. Downloaded on November 02,2020 at 16:39:51 UTC from IEEE Xplore.
10. Heng-Fa Teng , Meng-Jil Wang and Cheng-Min Lin An Implementation of Android-Based Mobile Virtual Instrument for Telematics Applications, 2011 Second International Conference on Innovations in Bio-inspired Computing and Applications.
11. Hong Guo, Member, IEEE, Jacob A. Crossman, Member, IEEE, Yi Lu Murphey. Automotive Signal Diagnostics Using Wavelets and Machine Learning. IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 49, NO. 5, SEPTEMBER 2000.
12. Bryan J. Muscedere, Robert Hackman, Davood Anbarnam. Detecting Feature-Interaction Symptoms in Automotive Software using Lightweight Analysis, SANER 2019, Hangzhou, China Research Papers.
13. Mirco Marchetti and Dario Stabili. READ: Reverse engineering of automotive data frames, .2018.2870826, IEEE Transactions on Information Forensics and Security.

