ISSN: 0974-5823 Vol. 6 No. 3 October-December, 2021

International Journal of Mechanical Engineering

Android App for Reading Electric vehicle Parameters

Arunkumar A
Department of Electrical and Electronics Engineering, PSG College of Technology, Coimbatore, Tamil Nadu, India.
aarun9297@gmail.com

Bogaraj T
Department of Electrical and Electronics Engineering, PSG College of Technology, Coimbatore, Tamil Nadu, India.
tbr.eee@psgtech.ac.in[2]

Abstract —

This Paper describes about developing an android app that can communicate with vehicle On-Board diagnostics and display the vehicle parameters. The main objective is to develop an android app that can read vehicle parameters such as speed, throttle etc. Moreover apps already available in the market are capable of receiving only the internal combustion engine parameters and not electric vehicle parameter. So an app can be developed to read electric vehicle parameters and display it in a device such as smart phone or tablet. The various vehicle parameters are communicated in the car through CAN bus. Normally vehicle parameters are monitored in the car dashboard but the parameters are always limited. So an app is developed that can receive usual vehicle parameter as well as other parameters required. The vehicle parameters are identified using the Parameter-ID (PID). Electric vehicle PID is not yet standardized and it is to be formulated. The app is able to read parameters such as rpm, calculated engine load, absolute throttle, after malfunction indicator lamp (MIL) distance, battery voltage and coolant temperature etc. Finally the instrument cluster can be replaced with the tablet/Smartphone or it can be used as an additional display for reading parameters.

I. INTRODUCTION

The concept of electric vehicle was introduced in the mid-19th century but more intensive development was carried out only in the 21st century after the development of renewable energy sources. Electrifying the road transportation is the one of the prime challenge the world has at present because of issues due to the IC engine based transportation system, such as i) depletion of oil and gas, ii) environmental pollution, and iii) global warming etc. Oil will exist up to 2060 if the present consumption trend is continued. Hence it is necessary to find alternate fuel or energy sources for transportation. One of the alternate is battery vehicles which are called as Electric Vehicles (EVs). In 2016, Road transportation demanded 44.8 % of total oil consumption of the world and contributed 24% of global CO_2 emissions. India spent Rs 4.7 Lakh Crores in 2016-17 in importing crude oil. Introduction of Electric Vehicles and Renewable energy based power generation will help lower this cost. The NDA Government set the goal that India to be 100% electric vehicle nation by the year 2030. The National Electric Mobility Mission Plan (NEMMP) of India targets seven million electric and hybrid vehicles by 2020 itself [1].

On-board Diagnostics (OBD) was made compulsory in all vehicles. The OBD port is usually within the reach of the driver. It can only read the data of the car and cannot write the data so it cannot cause damage to the vehicle. The connector used in the project is ELM327. It comes with interface such as Bluetooth, USB and Wi-Fi. The interface chosen is Bluetooth. The protocols used is ISO 15765-4 CAN (11 bit ID, 500 Kbit/s). ELM327 also handles the header and format. One can just enter the mode and the Parameter-ID (PID) and the device automatically adds the header. ELM327 can read vehicle parameter such as rpm, calculated engine load, absolute throttle etc. and display it in a smartphone.

II. HARDWARE AND SOFTWARE FOR OBD DEVELOPMENT

A. OBD Port

OBD port consists of a standard 16 pin J1962 connector and every car has it. It supports five protocols such as J1850 VPW, ISO-9141-2, KWP2000, J1850 Pulse Width Modulation (PWM) and CAN (Controller Area Network). CAN protocol is mostly used [2]. The pin 4 and 5 are always tied to ground and the pin 16 is connected to the auxiliary battery (12V). Other pins are connected according to the protocol to be used. For CAN protocol, 6 and 14 pins are given to CAN high and CAN low. The main reason for using CAN is the higher bit transfer rate. On-board diagnostics port is capable of communicating with CAN bus and ELM327 stands as a bridge between android app and On-board diagnostics port. The ELM327 is a microcontroller produced by ELM electronics for translating the On-board diagnostics found in cars and it is widely used to communicate with the car.

B. OBD – II PIDS

PID are the codes that are used to give request and receive data from the vehicle. A PID is send to the CAN bus and the device responsible for the PID gives the value for the PID and it can be

visualized [3]. In the latest OBD – II standard there are ten modes specified. Each manufacturer need not support all modes in their vehicle. They can also add custom modes according to their need.

C. ELM327

ELM327 is a microcontroller developed by ELM electronics for reading the vehicle parameters through OBD. The first ELM327 was developed with the help of PIC18F2480 microcontroller [4]. It translates the low level protocol interface into a simple interface and present it to the user. It comes with USB, Bluetooth, RS-232 and Wi-Fi. The device supports many protocols [5]. In the project Bluetooth interface is used. Supply voltage is 4.2 - 5.5V and it draws very less varying current of 0.2 - 0.4 A depending on whether the device is in discovery mode or in transmission and reception mode.

D. CAN

A Controller Area Network (CAN) is a protocol designed with the intention to communicate with controllers without the need for host. It is based on messages. The primary reason for the development of CAN is for the reduction of electrical wiring inside the vehicle but it is used for other purposes too. It is multi-master based configuration. In cars there are many Electronic Control Unit (ECU) and they are called as nodes. For the communication there must be at least 2 nodes present. Each and every node is connected to a bus which has two wires called CAN high and CAN low. These wires have a 120 Ω resistance between them for termination.

E. ECU

An Electronic Control Unit (ECU) is used for controlling the electrical system in the vehicle. A vehicle can have up to 80 ECU. Managing the ECU is the major difficulty faced by the Original Equipment Manufacturer (OEM). The main elements of ECU are microcontroller, memory, inputs, outputs and communication links. The ECU function can be modified by tuning them for various needs like a vehicle in normal road should have its ECU tuned when it's working with off road. However now most of the modern vehicles comes with ECU locks which does not allow tuning it and the tuning of ECU is considered as illegal. The ECU gets its power from 12V battery and the communication is through CAN.

F. ANDROID OPERATING SYSTEM

Android is an operating system developed by Google. It was developed for camera device and its now in the mobiles. It has the highest number of users in the world. Android is hugely popular because of its open source and it is based on Linux Kernel. It comes under the project name of Android Open Source Project (AOSP). Other companies get the source and they change it according to their need. The nexus and pixel line up has the pure OS whereas others uses the customization over the top. The updates are easily available for nexus and pixel line up because of no customization whereas the others have to customize and provide the updates. Moreover now Google has developed project treble which allows other companies to provide quicker updates. In addition to mobile now some manufacturers manufacture their laptop with android OS. Android has the flexibility of modifying the manufacturer's software through rooting. Rooting is the process of gaining full access of the OS. After rooting one can modify the default frequency of the processor set by the manufacturer by either overclocking it or under clocking it providing the facility of increased performance or increased battery saving and higher level in modification leads to development of custom OS.

G. APP INVENTOR

App Inventor is a server based application development platform for android developed by Google. User without any knowledge of android programming can develop an android app with the help of it. It is drag and drag development. It is based on events that is after completion of an event an operation will take place. There are two mains screens in it. They are viewer and blocks editor. Viewer is for adding the visual components in the app while the blocks editor is for adding the program blocks. For Eg. Button should be placed on the viewer and the functionality of the button should be in the in the blocks editor. After finishing both, the developer must choose to generate apk and the data will be uploaded and converted into relevant programming in the server side and provide the final apk to the developer. It provides the functionality of backing up the project. It also has backpack function which provides switching of blocks between projects. It also supports extension which allows for further features developed by other users that can be used in app inventor.

H. VECTOR DEVICE

It is a device used for high performance application and to be used in conjunction with CANoe / CANalyzer. It can be used as a data monitoring device. This device has the flexibility of supporting CAN, LIN, J1708 and Flex Ray. After connecting the device with the PC the device must be selected in CANoe through hardware window and assigning the appropriate channels to the connected hardware. It has the advanced features of time synchronization in terms of both hardware and software.

I. CANOE

CANoe is a tool for the development, testing and analysis of the ECU's. OEM uses this tool for predicting the performance of the ECU. It provides an environment in which the real model can be simulated and tested. This provides the option for rectifying the errors at an early stage of development. It executes the test cases automatically and generates the report automatically for these test cases. This contains three tabs they are trace, configuration and analysis. In trace there are four tabs in which the overview tab is used for viewing the CAN messages and its source. In the simulation setup tab nodes are created based on the requirement. It also has the functionality to enable and disable the specific node. A database must be created in which the environmental variables with its identifier, minimum, maximum value and datatype should be specified. It provides the functionality of simulating the bus with virtual CAN and also with real CAN hardware.

J. CAPL SCRIPT

Communication Access Programming Language (CAPL) is used to program the CANoe which work using CAN protocol. The features of CAPL script are node emulation, network emulation, node testing and gateway.Most programs are developed using the CAPL Browser, which provides an easy path in development process. Even though provided as an integrated component of CANalyzer or CANoe, the CAPL Browser application program can also be used separately.

III. DESIGN OF THE APP AND CANOE

The connection of various blocks and circuit connection between ELM327 and Vector Device are discussed below. The communication between android app and ELM327 is through Bluetooth. ELM327 communicates with Vector device through CAN and the Vector device is connected with PC.

A. ANDROID APP DEVELOPMENT

The android app is able to communicate with ELM327 through Bluetooth. The android app sends the PID to the ELM327. The ELM327 converts the PID into CAN Tx and CAN Rx and it has inbuilt CAN transceiver which converts it to CAN high and CAN low. The CAN high and CAN low are connected to the vector device which is connected to the PC using Universal Serial Bus (USB). The CAPL script from the PC sends the response like the response send from ECU. The input PID sends the request using the address 7DFh and it will be displayed in the trace window of the CANoe. The CAPL script should send the acknowledgment and the data requested by the PID. The response address can vary from 7E8h to 7EFh.The diagram showing the connections from android app to CAPL script is shown in Fig.1.

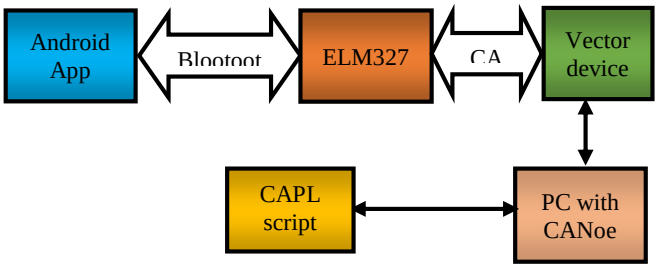

Fig. 1Connection between android app and CAPL script

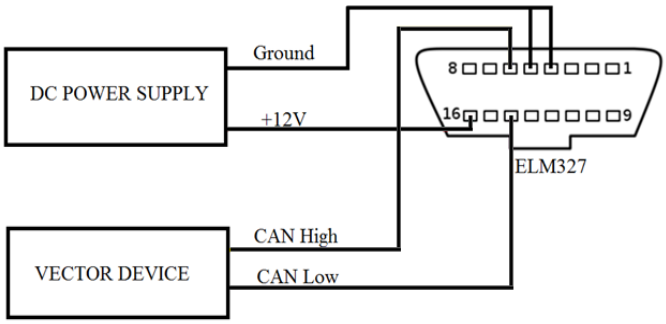

Fig. 2 Circuit Connection between ELM327 and Vector Device

The pin 16 in ELM327 is connected to the 12V in the power supply. The pin 4, 5 are shorted and connected to the ground of the power supply. The pin 6 is CAN high and is connected to the Vector Device. Similarly pin 14 is CAN low and is connected to the Vector Device. The circuit connection between ELM327 and Vector Device is shown in Fig.2.

IV. IMPLEMENTATION OF THE APP AND CANOE

The various blocks in developing the app and designing of the panel in CANoe are discussed below. After finishing the app inventor blocks the app is loaded into the smartphone and the panel is added into the CANoe configuration.

A. APP INVENTOR BLOCKS

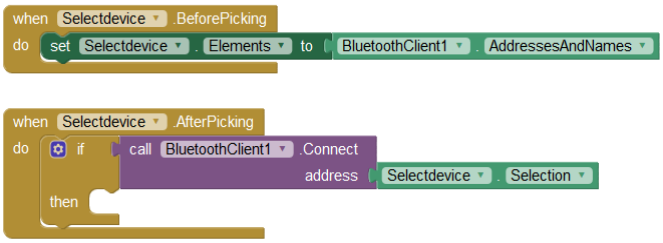

Fig.3 Bluetooth Selection Block

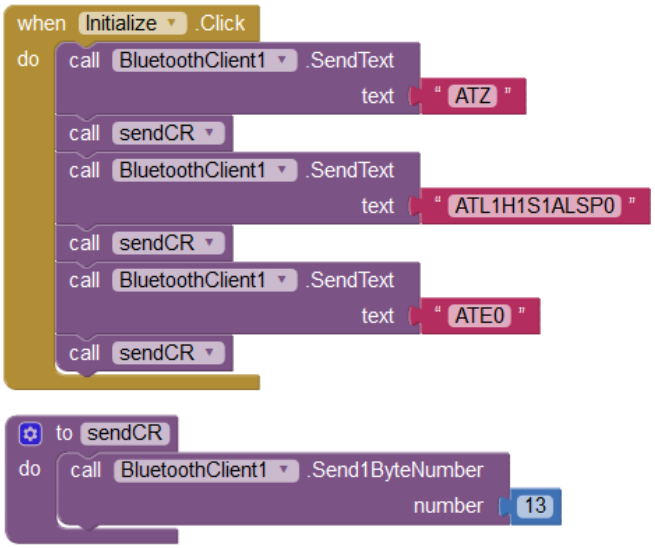

Fig. 4 ELM327 Initialization Block

The app inventor blocks for Bluetooth selection and ELM327 initialization blocks are shown in Fig. 3 and Fig.4 respectively.

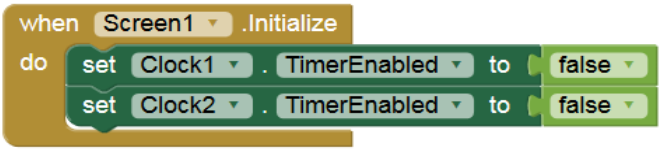

Fig. 5 Screen Initialization Block

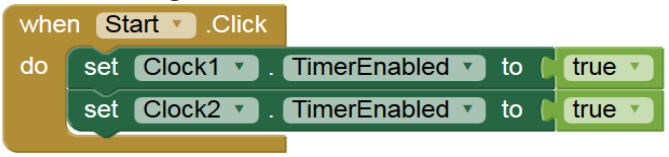

Fig. 6 Start Block

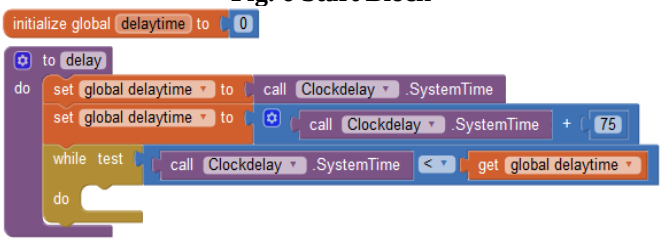

Fig. 7 Delay Block

The app inventor blocks for screen initialization, start and delay blocks are shown in Fig. 5, Fig. 6, and Fig. 7 respectively. The app inventor block for clock1 is shown in Fig. 8. The app inventor block for clock2 is shown in Fig. 9.

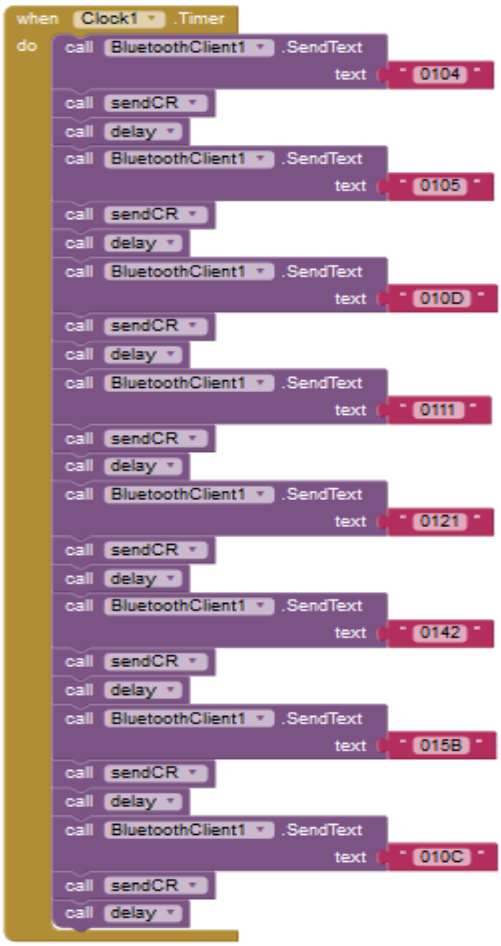

Fig. 8 Clock1 Block

B. CANOE

In the CANoe simulation setup should be opened and a node is to be created. The node is named as ECU. Simulation setup in CANoe is shown in Fig. 10.

After the node is created, database should be created. In the database the environmental variables should be declared. The environmental variables are shown in Fig. 11.

After the environmental variables are created the panel is created and the environmental variables are attached to their corresponding slider. The CAPL script is then written in the panel which is shown in Fig. 12.

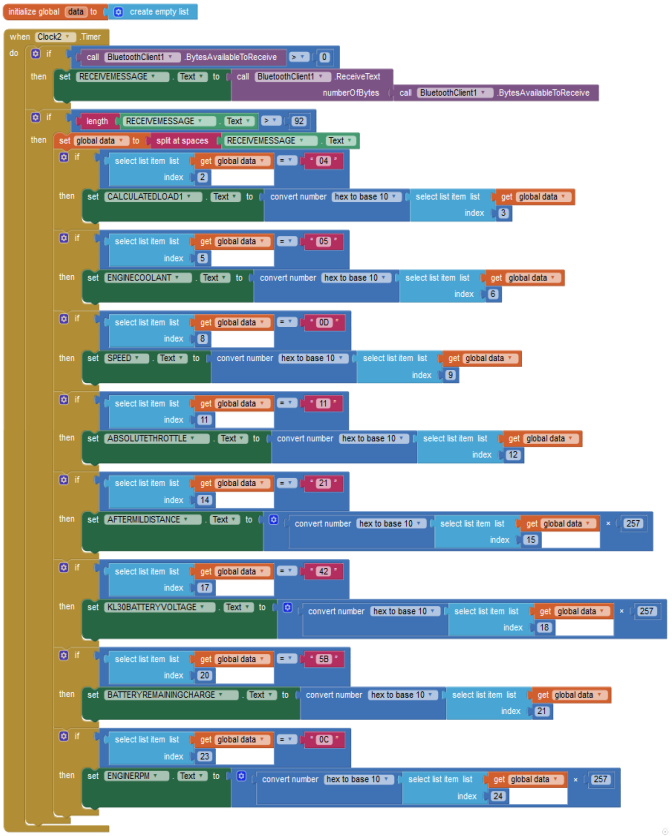

Fig. 9 Clock2 Block

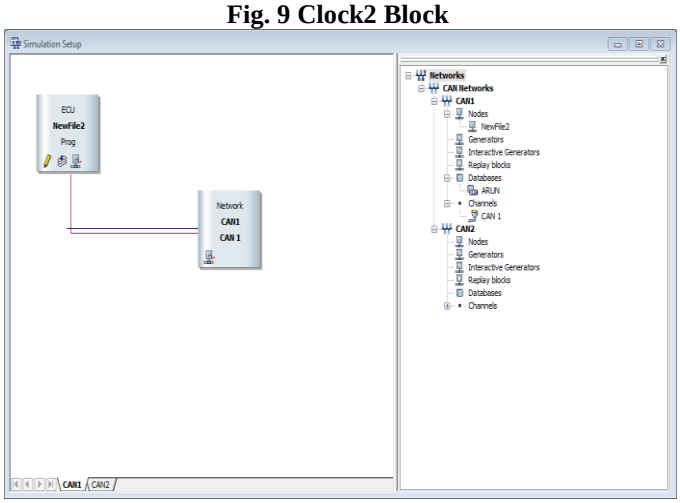

Fig. 10 Simulation Setup in CANoe

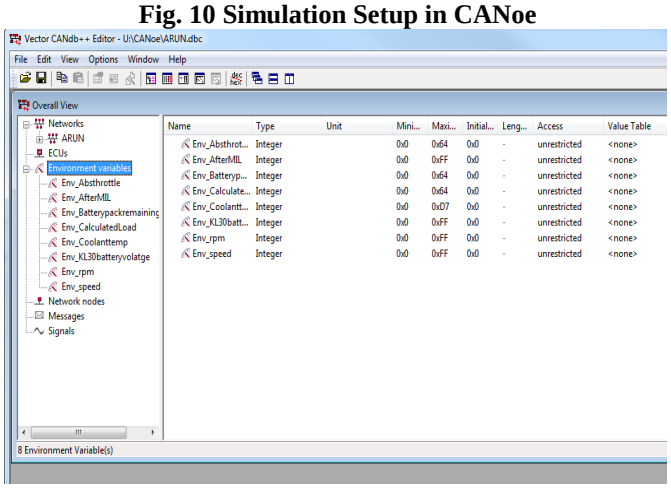

Fig. 11 Environmental Variables

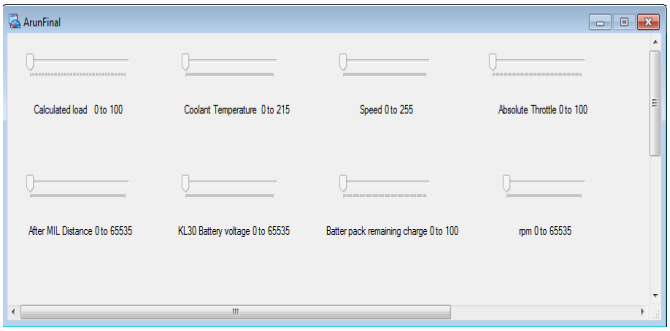

Fig. 12 Panel

V. RESULTS AND DISCUSSION

After the start button in the smartphone's app is clicked the PID are requested and the data is available in the trace window of the CANoe. The trace window is shown in Fig. 13.

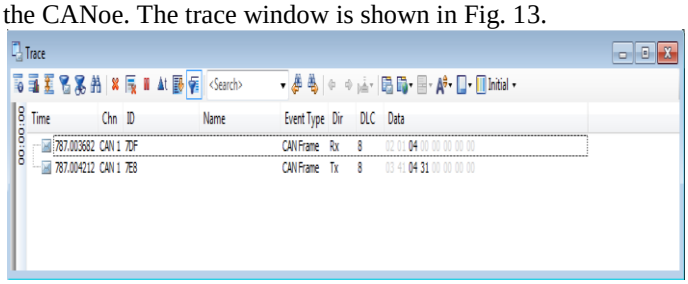

Fig. 13 Trace window - I

In the Fig 13, it can be seen that the PID requested is 04 with the address 7DF. The response is from the CAPL script with the data 31 from the address 7E8. Fig. 14 shows another PID being requested and the CAPL script's response.

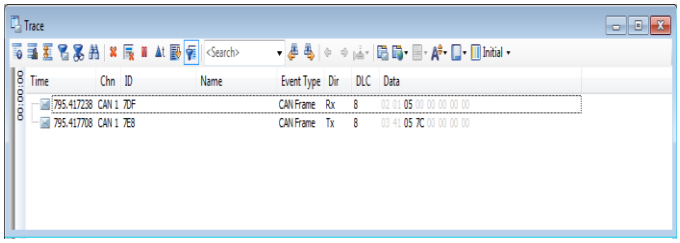

Fig. 14 Trace window – II

The data that is being send is set via the panel. Initially everything is set to be 0. Fig. 15 shows the panel with the PID data being set to 0.

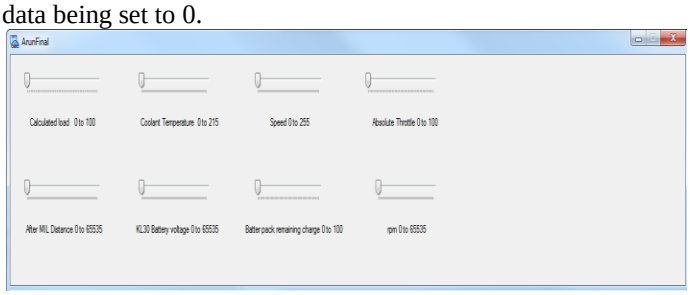

Fig. 15 Panel – I

The Fig. 16 shows the data that are being visualized from the app when it is set by panel in Fig. 17. The Fig. 17 shows the data in the panel is being changed using the slider. The Fig. 18 shows the data being changed in the app according to the change in the panel.

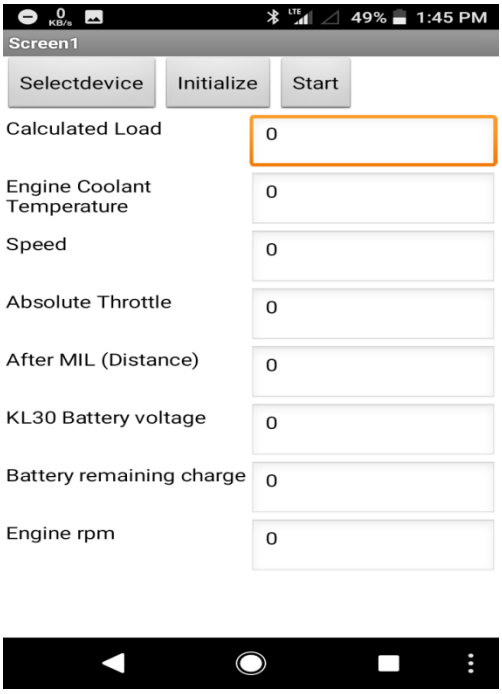

Fig. 16 App – I

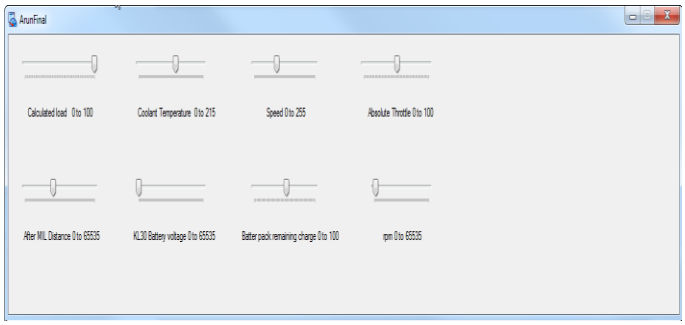

Fig. 17 Panel - II

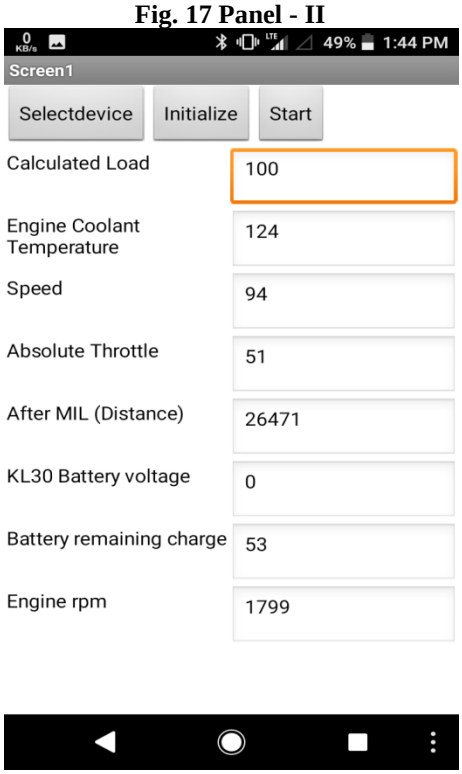

Fig. 18 App – II

CONCLUSION

The android app that is developed is able to receive data from CAN using ELM327 and decode it and display the information in the smart phone. The CAPL script is generated and it sends the response to the ELM327 device like in real vehicle. The app can read parameters such as engine load, coolant temperature, speed, absolute throttle, after MIL distance, battery voltage, battery remaining charge and rpm. Smart phone with the app loaded can be used as an additional display for viewing parameters that are not visualized in the dashboard. The app can be further developed with the full parameters and it can be replace the dashboard in the vehicle with a smart display. Moreover the app can be improved to control the vehicle with the smart phone.

REFERENCES

1. http://indiacsr.in/overview-indias-2030-vision-electric-vehicle/
2. A. Xandra, A. Sim and B. Sitohang, "OBD-II standard car engine diagnostic software development", 2014 International Conference on Data and Software Engineering (ICODSE), pp. 1-5, 2014.
3. A. Tahat, A. Said, Zf. Jaouni and W. Qadamani, "Android-based universal vehicle diagnostic and tracking system", 16th International Symposium on Consumer Electronics (ISCE), pp. 137-143, 2012.
4. Z. Szalay, Z. Kanya, L. Lengyell, P. Ekler, T. Ujj, T. Balogh and H. Charaf, "ICT in road vehicles – reliable vehicle sensor information from OBD versus CAN", 2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), pp. 469-476, 2015.
5. Elm Electronics, "ELM327 OBD to RS232 Interpreter", https://www.elmelectronics.com/wp-contents/uploads/2016/07/ELM327DS.pdf