User Manual

CDL – Proof of Concept

Version : 1.1

Date : 11 October 2016

Author : Seung-Hyun Yun

History

Rev.	Date	Author	Change
1.0	07 Sep 2016	Seung-Hyun Yun	Document created
1.1	11 Oct 2016	Seung-Hyun Yun	- added git repository information - added library search path

Table of Contents

1 Introduction

This document describes how to install and use the Proof of Concept for CDL project.

2 Background

The component Car Data Logger is responsible for collecting, storing car related data, and providing the data to other GENIVI compliance components and off-board servers.

CDL is started from an idea that if we could collect, store, and provide car related data, it would make a very valuable service to auto-makers, users, and other GENIVI compliance components.

So, the goal of CDL is defining what kind of data should be collected, how to collect, and store data properly. It should also provide data to on-board and off-board (devices.)

Additionally, guaranteeing the data integrity is required and security of data should be considered as well.

This PoC realizes concept of Data Collect, Store and Provide requirements of the CDL Vehicle Level Requirements, especially focused on the Vehicle Data.

2 Folder Structure

- project – directory for source codes
- env – directory for CommonAPI configuration file (commonapi.ini)
- json – directory for json files for application using CommonAPI SOME/IP Runtime
- script – directory for run scripts

3 Installation

3.1 Tested Environment

This PoC is implemented and tested on:

- VMWare Workstation 12 Player (12.1.1 build-3770994)
- Ubuntu 14.04 64bit
- Qt 5.6.1

3.2 Precondition

To build the PoC, following packages are required

- CommonAPI 3.1.5 (including DBus and SOME/IP Runtime)
- vSomeIP 2.0.1
- automotive-dlt 2.15 (or later)
- Qt5 (5.6.1 is recommended)

3.3 Clone Source Codes

Clone source codes from GENIVI GitHub using following command in the terminal window:

```
$ git clone https://github.com/GENIVI/car-data-logger.git


$ git checkout proof-of-concept
```

3.4 Build & Install

All applications are implemented based on Qt5.

Before build the project, please make sure Qt is properly installed using following command in the terminal window.

```
$ qmake -version
```

In order to build the project the pkgconfig files of the patched libdbus library must be added to the PKG_CONFIG_PATH.

For example, if the patched libdbus library is available in /usr/local, set the PKG_CONFIG_PATH variable as follows:

```
$ export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig:$PKG_CONFIG_PATH
```

Now use qmake to build the project. Default CONFIG value is release

```
$ qmake -r [-spec linux-g++] [CONFIG+=<debug|release>]

$ make

$ make install
```

After the installation, you can find binaries, libraries and other files in deploy/debug or deploy/release directory depending on build option.

4 Usage

4.1 Run

Navigate to the installed directory.

```
$ cd deploy/debug or cd deploy/release
```

Before run applications, add library search path for the patched libdbus to the LD_LIBRARY_PATH.

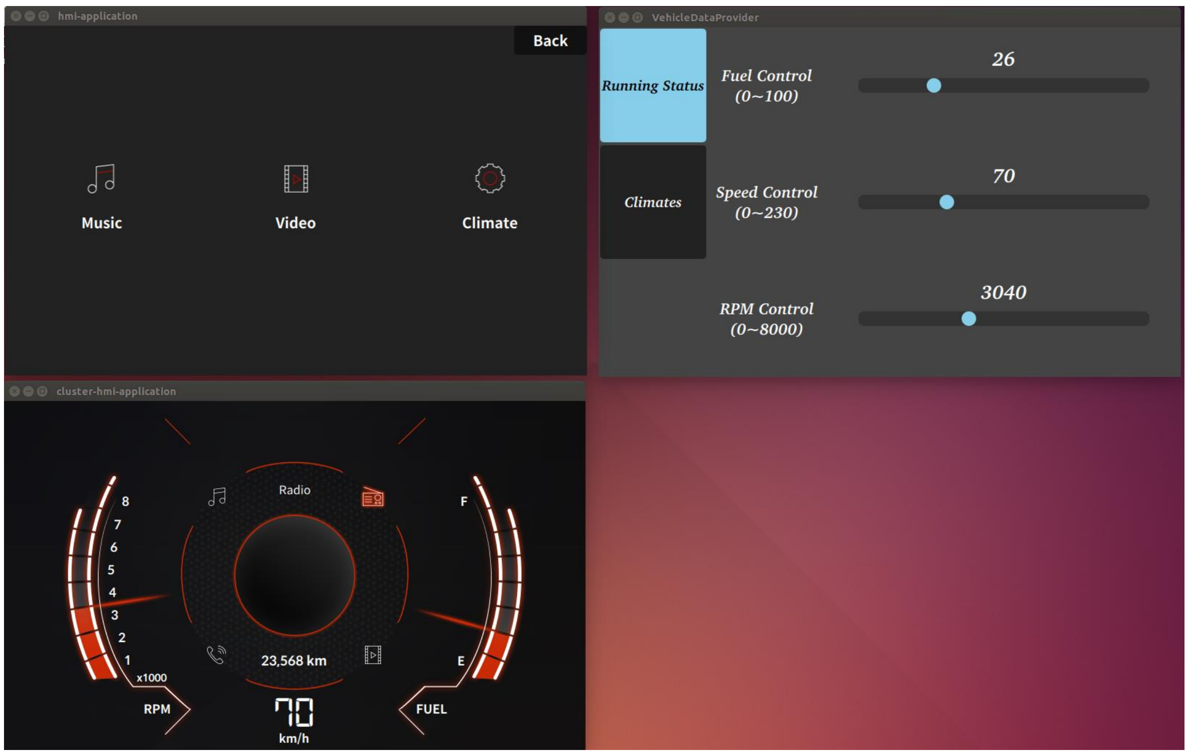
For example, if the patched libdbus library is available in /opt/lib, set the LD_LIBRARY_PATH variable as follows:

```
$ export LD_LIBRARY_PATH=/opt/lib:$LD_LIBRARY_PATH
```

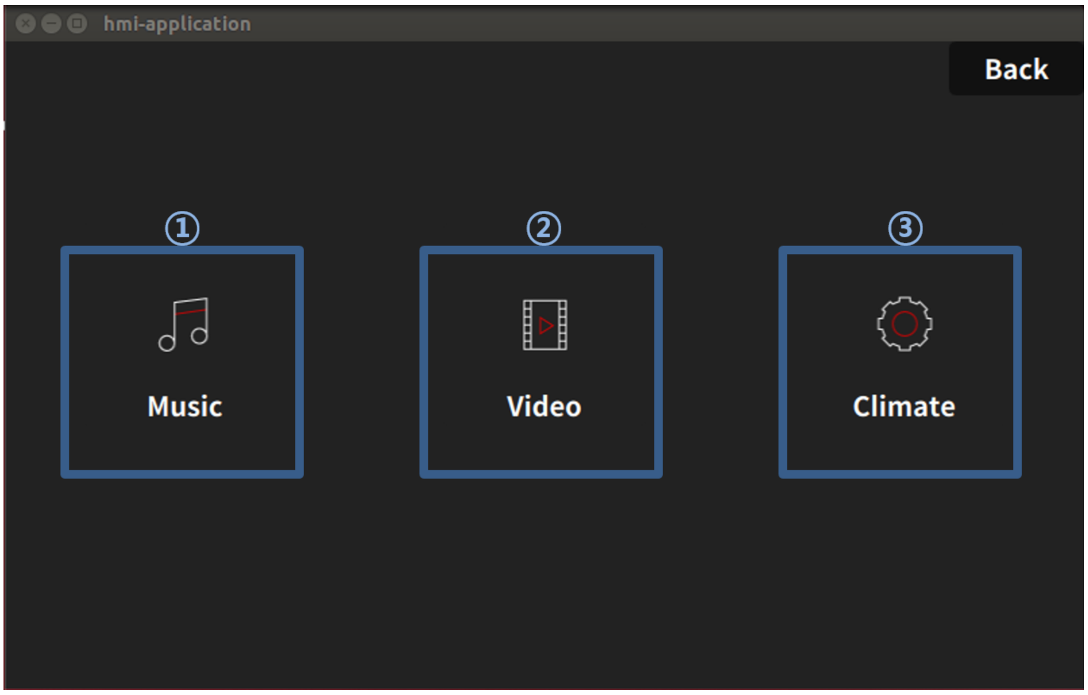
Enter the following command in the terminal window.

```
$ ./run_all.sh
```

The following 3 GUI applications should appear. The other applications are run as daemon.



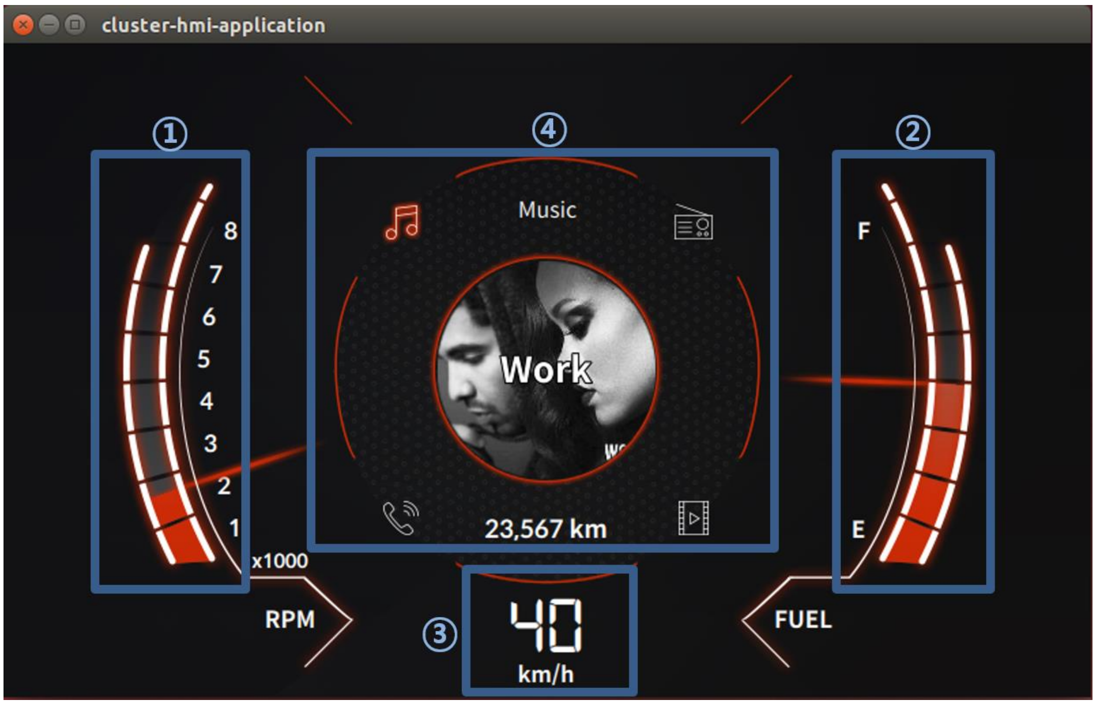
4.2 AV Application



Following features are implemented.

- Audio / Video screens with emulated media data - ①, ②
- Low Fuel Pop-up on any screen referring to fuel level
- Driving restriction on video screen referring to vehicle speed - ②
- Climates information screen - ③

4.3 Cluster Application



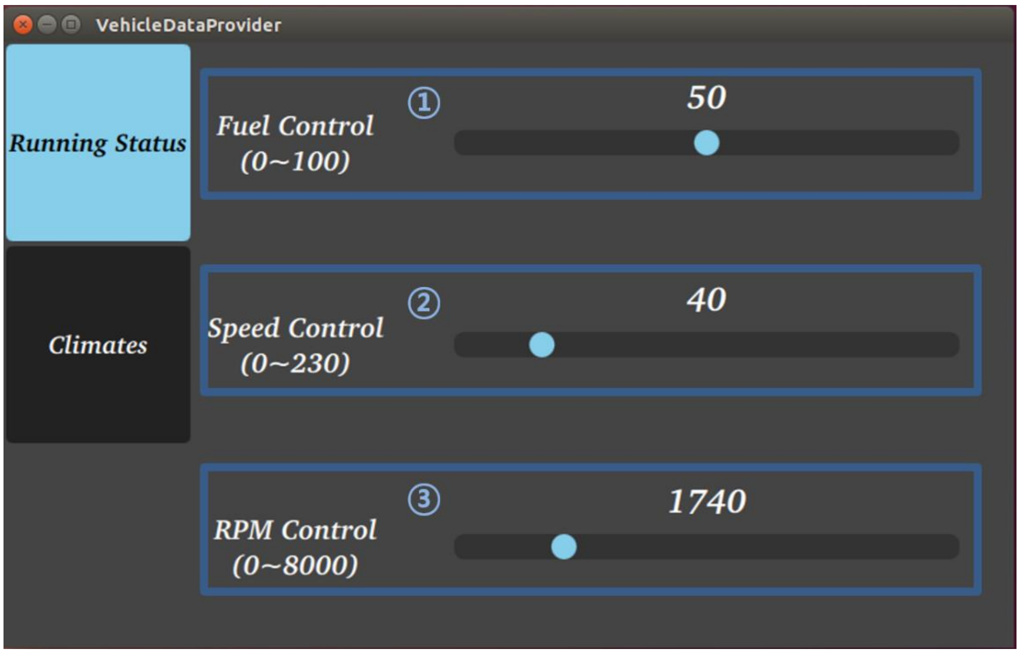
Implemented to shows running status data as follows:

- Engine Speed - ①
- Vehicle Speed - ③
- Fuel Level - ②
- Media Metadata - ④

4.4 Vehicle Data Provider

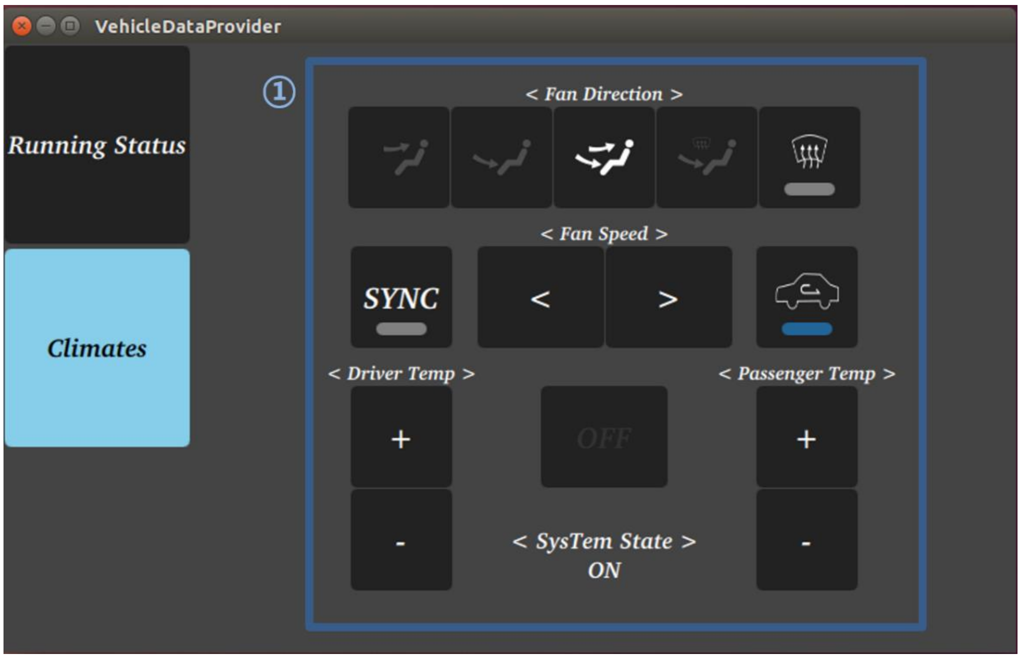
Using Vehicle Data Provider, you can generate vehicle data.

Running Status Data



- Engine Speed - ③
- Vehicle Speed - ②
- Fuel Level - ①

Climates Data - ①



5 Architecture

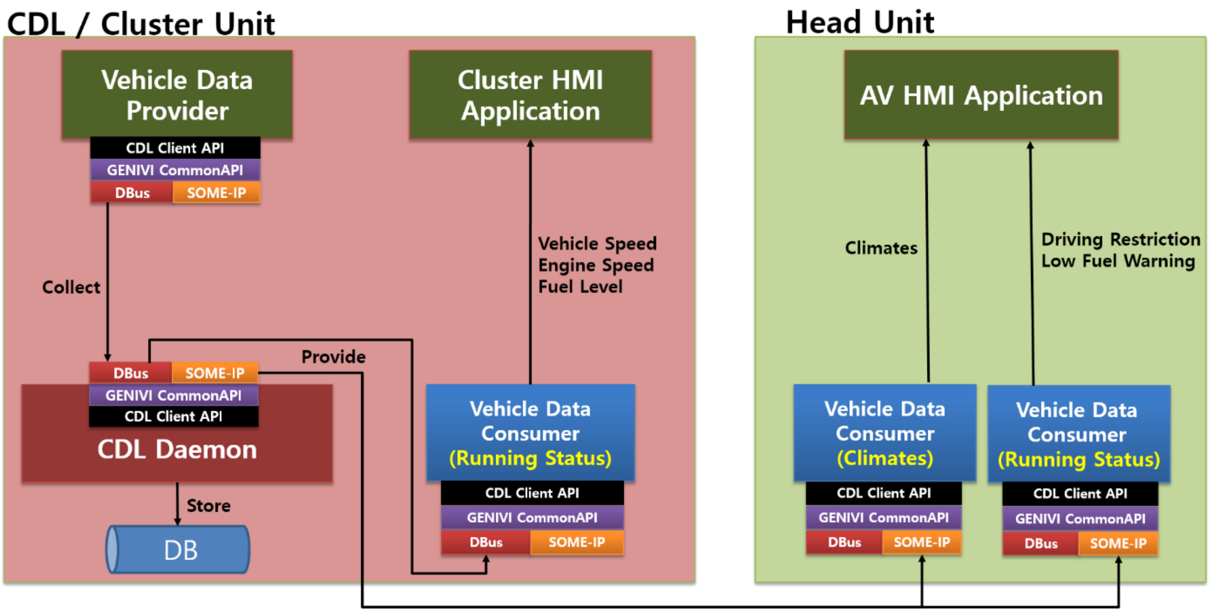
5.1 CDL Features

For this Poc, CDL Daemon is implemented that collects vehicle data provided from Vehicle Data Provider.

Once vehicle data is collected, CDL Daemon stores vehicle data and provides to Vehicle Data Consumer via DBus and SOME/IP both (Local and Remote).

Each Vehicle Data Consumer handles and transforms vehicle data properly as needed.

Vehicle Data Provider, CDL Daemon, and Vehicle Data Consumers are using specified Vehicle Data Specification that defined referring to GENIVI Vehicle Web API.

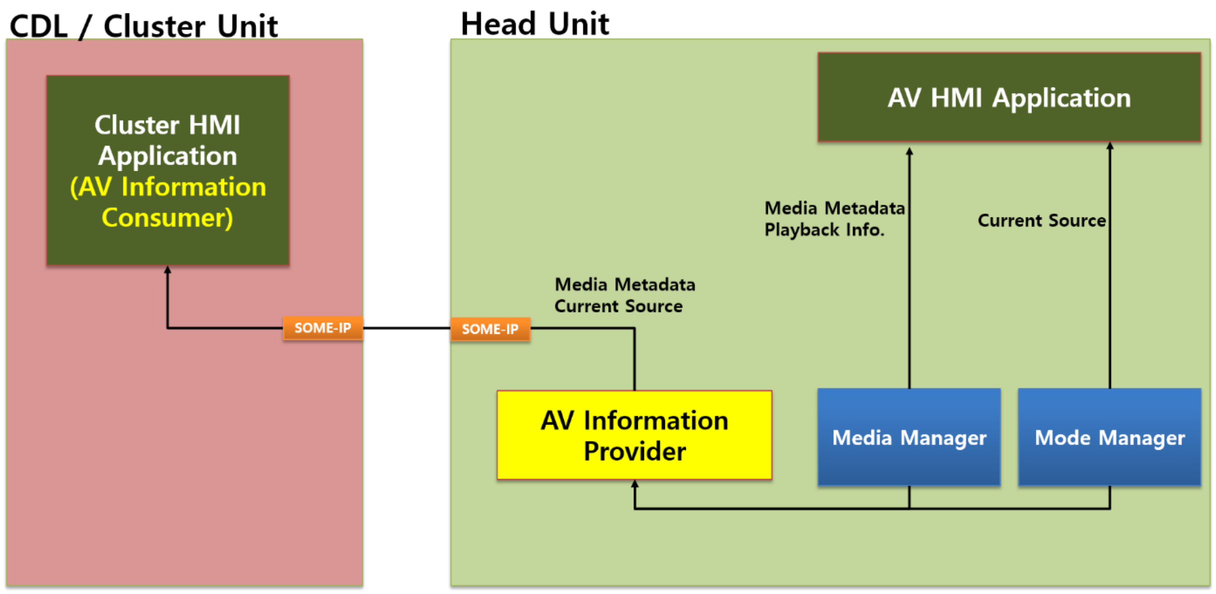


5.2 AV Information Feature

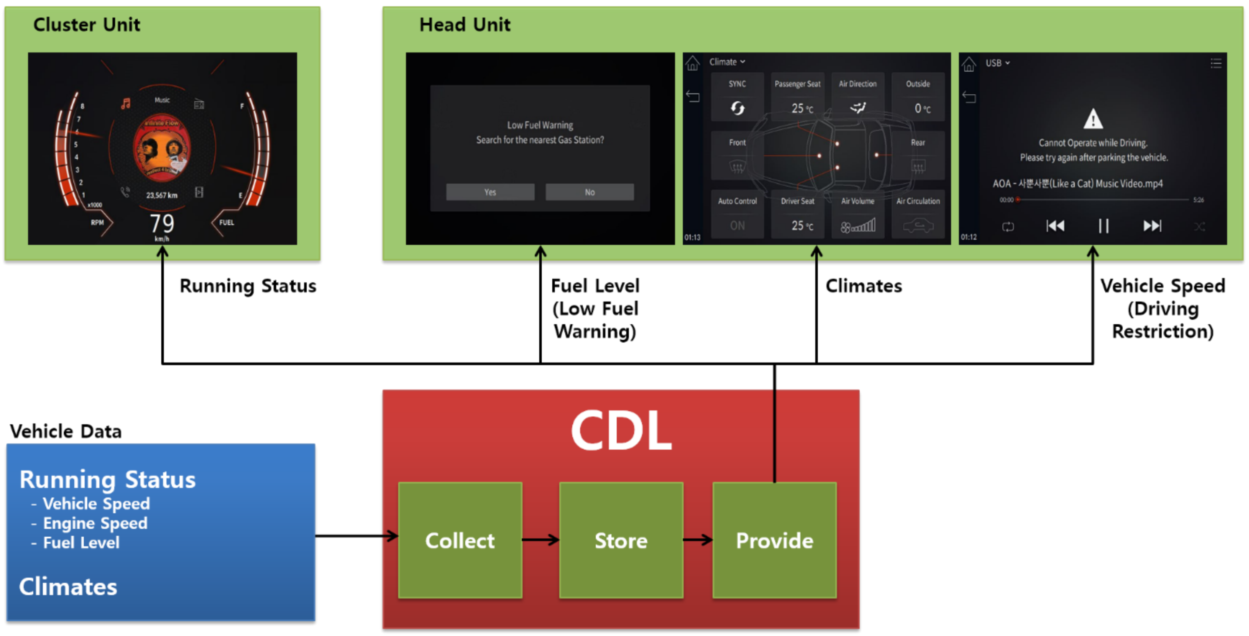
This is additional feature of PoC, and implemented to display current AV source and metadata information on Cluster.

The current AV source and metadata information is provided from AV Information Provider of Head Unit, information is transferred via SOME/IP.

In case of audio application, when playing track is changed, AV Information Provider provides title and cover art image data to AV Information Consumer (Cluster Application).



6 Test Cases


Cluster Unit
Head Unit
Low Fuel Warning
Search for the nearest Gas Station?
Yes
No
Cannot Operate while Driving.
Please try again after parking the vehicle.
Running Status
Fuel Level (Low Fuel Warning)
Climates
Vehicle Speed (Driving Restriction)
Vehicle Data
Running Status
- Vehicle Speed
- Engine Speed
- Fuel Level
Climates
CDL
Collect
Store
Provide

3.1 Cluster Data (Vehicle Speed, Engine Speed, Fuel Level)

When running status data is received such as vehicle speed, engine speed, and fuel level, cluster application displays running status data.

3.2 Climates Data

When climates data is received, AV application displays climates data.

3.3 Driving Restriction

If vehicle speed is higher than specified level (>20 km/h), driving restriction is enabled, and on video screen of AV application, driving restriction function is activated.

3.4 Low Fuel Warning

If fuel level is lower than specified level (>10%), low fuel warning is enabled, and on any screen of AV application, low fuel warning pop-up is appeared.