Savari V2X SDK
Installation and User Guide

Release 5.9.0
Document Version 0.1
November 23, 17

Table of Contents

1 Introduction

MW-1000™ and SW-1000™ are Savari's 1609.X, SAE J2735, and 802.11p compliant OBU and RSU devices used for V2V and V2I applications. These devices are flexible and open platforms that can be customized for any application. They run Savari On-board Operating System (SOBOS), which is a custom Linux distribution. Savari V2X SDK provides feature rich libraries to facilitate development of V2V or V2I applications for MW-1000™ and SW-1000™.

This SDK Guide provides an overview of the overview of the Savari V2X software architecture and a detailed description of the APIs provided. It also provides instructions on compiling your application, installing it, and running it on the MW-1000™ and SW-1000™. Sample applications using the API described in this guide can be found in the SDK.

1.1 Intended Audience

This guide is written for developers interested in developing applications for MW-1000™ and SW-1000™ who are generally experienced C programmers with in-depth knowledge of the following:

- IEEE 802.11, 802.11p, 1609.X (2016), and SAE J2735 (2016) specifications
- General networking and network programming
- Knowledge of Linux kernel and driver development

See the Resources **Section 1.2** for specifications and other reference materials.

1.2 Resources

The following documents also provide additional information related to the MW-1000 SDK:

1. ***Savari MW-1000™ User Guide*.**
2. ***Savari V2X Library API (v5.9.0)***
3. IEEE P1609.3 Standard for Wireless Access in Vehicular Environments (WAVE) - Networking Services, Revision 2016.
4. IEEE P1609.4 Standard for Wireless Access in Vehicular Environments (WAVE)- Multichannel Operation, Revision 2016.
5. IEEE 802.11p Amendment to Standard for Information Technology – Telecommunications and information exchange between systems – Local and metropolitan networks – specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 3: Wireless Access in Vehicular Environments (WAVE), Revision D11, dated March 2010.
6. IEEE P1609.2 Standard for Wireless Access in Vehicular Environments – Security Services for Applications and Management Messages, Revision 2016.
7. SAE J2735 Dedicated Short Range Communications (DSRC) Message Set Dictionary Revision 2016-03.

Users of the SDK are strongly encouraged to read the respective User Guides of the devices before attempting to develop applications.

2 Accessing the MW-1000™ and SW-1000™

MW-1000™ and SW-1000™ can be accessed through ssh or serial console. Please refer MW-1000™ and SW-1000™ user guide for further information.

3 Using the Savari V2X SDK

MW-1000™ or SW-1000™ applications are cross-compiled (built on host PC for MW-1000™) using the Savari V2X SDK. They cannot be compiled directly on MW-1000™ or SW-1000™.

3.1 Build System Requirements

An x86 based 64-bit Linux system is required to build applications Using V2X SDK. Preferred Linux distro is Ubuntu 14.04 LTS but other versions should work. If Linux system is not available, then install VMware player on a Windows machine and use prebuilt Ubuntu VMware VMDK image. We have tested SDK with prebuilt VMWare VMDK images from OXBOXES. Use VMWare 64bit VMDK image from following link.

http://www.osboxes.org/ubuntu/#ubuntu-14_04-vmware/

Please refer to the following link on how to install VMware Player and its system requirements:

http://www.vmware.com/download/player/

http://www.vmware.com/in/products/player/faqs/install-requirements.html#system_requirements

We verified with Lenovo T440p with following configuration settings.
System type: 64-bit operating system, x64-based processor
BIOS Setting: Enable Virtualization Technology and VT-d Feature
VM Player display Setting: Disable Accelerate 3D graphics

3.2 Installing Savari V2X SDK

1. Login to Linux build system.

2. Download and extract **Savari_SDK_5_9_0.tar.bz2** file. This will create directory Savari_SDK_5_9_0. To extract the tarball run the following command

   # tar xvf Savari_SDK_5_9_0.tar.bz2

3. Change into SDK directory

   # cd $HOME/Savari_SDK _5_9_0

The SDK, sample applications and all other required software are installed in this directory.

Sample programs are in the following location:

$HOME/Savari_SDK_5_9_0/sample

The toolchain binaries, libraries and header files are in the following location:

$HOME/ Savari_SDK_5_9_0/toolchain

The V2X SDK libraries and header files are in the following location:

$HOME/ Savari_SDK_5_9_0/v2x_sdk

All samples are readily buildable MW-1000™ and StreetWAVE applications:

3.3 Building Applications

3.3.1 Building a Sample Application

The following example shows how to build a sample application.

```
# cd $HOME/Savari_SDK_5_9_0/sample
```

Under sample directory, will have applications for each layer as per the Savari V2X Library API document [ref-2]

```
# cd multi_layer
```

```
# make -f BSM_Makefile clean
```

```
# make -f BSM_Makefile
```

These commands generate the output binary named **bsmsample**. Then, copy the binary to the MW-1000™. MW-1000™ supports standard Linux secure copy command.

Default login prompt of StreetWAVE is CLI prompt. We need to shell-drop from CLI prompt of StreetWAVE to execute the sample application. Also file copy of StreetWAVE can only be initiated from the board. Shell-drop procedure will be shared separately by mail on request.

For example,

```
# scp bsmsample root@<MW-1000 IP>:/tmp
```

After you have copied the application, you can run the library on the MW-1000™ or SW-1000™ as shown below which sends BSM with PSID of 0x20 and delay of 100ms between packets. Make sure to stop BSM application before running the sample BSM application.

```
# /etc/init.d/bsmd stop
```

```
# cd /tmp
```

To transmit BSMs in secured mode at transmit interval of 100 milliseconds, run the following

```
# ./bsmsample -T -t 100 -s
```

To receive the BSMs in secured mode at the other box the same sample binary can be used. To run it in the receive mode, type in

```
# ./bsmsample -R -s
```

For more help on how to run this sample binary, type in

```
# ./bsmsample -h
```

3.3.2 Building Your Own Application

To write your own application apart from the sample application given above, do the following:

- Create a directory under **$HOME/Savari_SDK_5_9_0**/ (For example, Myapp)
- Write your own application for the use case needed (can refer to some of our sample applications)
- Write a Makefile for the corresponding application similar to one of the sample Makefiles. Link all the libraries required by your application by setting LIBS variable appropriately.
- Make sure you included all your C files in the SRC += option in the **Makefile**.
- Compile it, using **make -f <Makefile Name>**.

Last steps of copying and running the applications are same as the above example of sample application.

Please refer to Savari V2X Library API document for API related documentation.

4 Glossary

This glossary contains a list of technical terms used throughout the Savari SOBOS Programmer's Guide.

Terminology	Description
1609	Family of standards for Wireless Access in Vehicular Environments (WAVE).
1609.2	Trial Use Standard for Wireless Access in Vehicular Environments (WAVE) - Security Services for Applications and Management Messages defines secure message formats and processing. This standard also defines the circumstances for using secure message exchanges and how those messages should be processed based upon the purpose of the exchange.
1609.3	Trial Use Standard for Wireless Access in Vehicular Environments (WAVE) - Networking Services defines network and transport layer services, including addressing and routing, in support of secure WAVE data exchange. It also defines Wave Short Messages, providing an efficient WAVE-specific alternative to IPv6 (Internet Protocol version 6) that can be directly supported by applications. Further, this standard defines the Management Information Base (MIB) for the WAVE protocol stack.
1609.4	Trial Use Standard for Wireless Access in Vehicular Environments (WAVE) - Multi-Channel Operations provides enhancements to the IEEE 802.11 Media Access Control (MAC) to support WAVE operations.
2735	See SAE J2735.
802.11p	Draft amendment to the IEEE 802.11 standard to add wireless access in the vehicular environment (WAVE). It defines the enhancements to 802.11 required to support Intelligent Transportation Systems (ITS) applications. This includes data exchange between high-speed vehicles and between the vehicles and the roadside infrastructure in the licensed ITS band of 5.9 GHz (5.85-5.925 GHz).
ASN1	Abstract Syntax Notation 1
BSM	Basic Safety Message.
DSRC	Dedicated Short Range Radio Communications One-way or two-way short- to medium-range wireless communication channels specifically designed for automotive use and a corresponding set of protocols and message sets. It defines communication between the vehicle and roadside equipment and between vehicles. DSRC protocols ride over 802.11p enhancements. DSRC radio supports 802.11p in the hardware and transmits power capabilities of up to 26dbM.
GPS	Global Positioning System
MAP	Map Data
OBU/OBE	On-Board Equipment/On-Board Unit

PSID	Provider Service Identifier
RSE/RSU	Roadside Equipment/Roadside Unit
SAE J2735	Society of Automotive Engineers (SAE) standard for support of interoperability among DSRC applications through the use of standardized message sets, data frames, and data elements. This standard provides information that is useful in understanding how to apply the various DSRC standards, along with the message sets, data frames and data elements, to produce interoperable DSRC applications.
SOBOS	Savari on-board operating system (based on Linux).
SPAT	Signal Phase And Timing
TIM	Traveler Information Message
USDOT	United States Department of Transportation
V2I	Vehicle to Infrastructure
V2V	Vehicle to vehicle
WAVE	Wireless Access in Vehicular Environments
WSA	WAVE Service Advertisement
WME	WAVE Management Entity