MMITSS-CA version 2.0

# Content for Application Page

## Tab 1: Summary

Multi-Modal Intelligent Traffic Signal System (MMITSS) is the next generation of traffic signal systems that seeks to provide a comprehensive traffic information framework to service all modes of transportation, including general vehicles, transit, emergency vehicles, freight fleets, and pedestrians and bicyclists in a connected vehicle (CV) environment.

The MMITSS applications bundles includes the Arizona version of MMITSS (MMITSS-AZ) and the California version of MMITSS (MMITSS-CA). The MMITSS-AZ aims to provide adaptive traffic and signal priority control using connected vehicle data. The MMITSS-CA aims to improve the existing actuated-coordinated traffic signal control system, with performance enhancements and signal coordination and priority control techniques enabled by the connected vehicle data. Traffic control agencies are provided with options for selective use of MMITSS-AZ or MMITSS-CA based on their needs. This package includes software set for the MMITSS-CA. The MMITSS-AZ software set can be download from OSADP (<https://www.itsforge.net/index.php/community/explore-applications/for-search-results#/30/63>).

The MMITSS-CA software set includes software to be hosted at three (3) physical devices:

* An Ubuntu Linux-based MMITSS Roadside Processor (MRP) that generates SAE J2735 MAP, SPaT (Signal Phase and Timing) and SSM (Signal Status Message) messages, processes SAE J2735 BSM (Basic Safety Message) and SRM (Signal Request Message) messages, calculates performance measures, determines MMITSS traffic and signal priority control strategies, and communicates control commands to the traffic signal controller;
* A messages transceiver software module running on an RSU (RoadSide Unit) that routes outbound SAE J2735 MAP, SPaT and SSM messages to be broadcasted over-the-air, and routes inbound over-the-air SAE J2735 BSM and SRM messages to the MRP; and
* Software modules running on an OBU (On-Board Unit) that generate and broadcast SAE J2735 BSM and SRM messages, and receives and process SAE J2735 MAP, SPaT and SSM messages.

## Tab 2: Description

The MMITSS-CA includes three (3) arterial traffic signal applications:

* Intelligent Signal Control (I-SIG)

I-SIG provides CV-based signal actuation and dilemma zone protection. The CV-based coordination control will be included in the future release of MMITSS-CA. I-SIG takes vehicle trajectory data from BSMs, places a service call to the controller on the phase that controls the vehicle's movement, and when needed, extends the green phase for dilemma zone protection. The phase call and extension control commands are communicated with the traffic controller through AB3418 over RS-232 communications. Interfacing with NTCIP controllers for MMITSS-CA traffic and priority control will be included in the future release of MMITSS-CA.

* Signal Priority (SP)

Signal priority provides priority control to different modes of vehicles including transit vehicles and trucks. Priority eligible vehicles (e.g., OBUs) receive SAE J2837 MAP, SPaT and SSM from RSU when they enter the DSRC communication range and generate and broadcast SRMs. The MRP receives and processes the SRMs, considers all the active priority requests from different vehicles to decide the appropriate priority treatment, and communicates the control command to the traffic signal controller. The adaptive signal priority feature that solves for optimal priority green splits based on active priority requests and prevailing traffic condition in terms of minimization of weighted traffic and priority vehicle delay will be included in the future release of MMITSS-CA.

* Mobile Accessible Pedestrian Signal System (PED-SIG)

PED-SIG works together with Savari SmartCross Application (<http://savari.net/solutions/smart-phone/>), which includes a pedestrian cloud sever that receives SAE J2735 MAP and SPaT from connected RSUs, and pedestrian phase request from SmartCross phone app. The MRP receives and processes the pedestrian SRM, and places a pedestrian service call to the traffic signal controller.

The MMITSS-CA has the following functionalities:

* MRP\_TCI (Traffic Controller Interface, hosted by the MRP computer)
  + Communicate with the traffic controller
    - Poll traffic controller's configuration data
    - Receive traffic controller’s status data, and populate SPaT data elements
    - Receive traffic controller’s loop count and occupancy data
    - Transmit MMITSS traffic and priority control commands
  + Communicate with MRP\_DataMgr (Data Manager)
    - Share controller’s configuration data through timing-card file
    - Transmit signal status, loop count, and occupancy data
    - Receive MMITSS traffic and priority control commands
* MRP\_DataMgr (Data Manager, hosted by the MRP computer)
  + Read intersection MAP description file, and encode MAP
  + Communicate with MRP\_TCI
    - Receive signal status message, and encode SPaT
    - Receive loop count and occupancy messages
    - Transmit MMITSS traffic and priority control commands
  + Communicate with RSU\_msgTransceiver (Message Transceiver)
    - Transmit encoded MAP, SPaT and SSM
    - Receive over-the-air BSM and SRM
  + Communicate with Savari’s pedestrian cloud server
    - Transmit encoded MAP and SPaT
    - Receive pedestrian SRM
  + Communicate with MRP\_Aware
    - Transmit BSM, SRM, and pedestrian SRM
    - Transmit signal status message
    - Receive encoded SSM
    - Receive vehicle BSM trajectory data message, and calculate intersection performance measures
    - Receive MMITSS traffic and priority control commands
* MRP\_Aware (hosted by the MRP computer)
  + Process BSMs, locate and track BSMs on MAP, determine vehicle's travel lane, distance- and time-to-arrival at the stop-bar. When needed, transmit vehicular phase call or green extension to MRP\_TCI via MRP\_DataMgr
  + Process pedestrian SRMs. When needed, transmit pedestrian phase call to MRP\_TCI via MRP\_DataMgr
  + Process SRMs and associate SRMs with BSMs, determine the priority strategy, and transmit priority control command to MRP\_TCI via MRP\_DataMgr
  + Transmit encoded SSM to RSU\_msgTransceiver via MRP\_DataMgr
  + Process and transit vehicle trajectory data to MRP\_DataMgr
* RSU\_msgTransceiver (Message Transceiver, hosted by RSU)
  + Register to RSU WME stack for receiving and transmitting over-the-air DSRC messages
  + Communicate with MRP\_DataMgr
    - Receive encoded MAP, SPaT and SRM, and manage to broadcast over-the-air
    - Transmit over-the-air BSM and SSM
* OBU\_msgTransceiver (Message Transceiver, hosted by OBU)

OBU\_msgTransceiver serves the identical role as RSU\_msgTransceiver. It

* + Register to OBU WME stack for receiving and transmitting over-the-air DSRC messages
  + Communicate with OBU\_Aware
    - Receive encoded BSM and SRM, and manage to broadcast over-the-air
    - Transmit over-the-air MAP, SPaT and SSM
* OBU\_Aware (hosted by OBU)
  + Read GPS data, and encode BSM
  + Process MAP, SPaT and SSM
  + Determine the upcoming MMITSS intersection, locate and track vehicle’s location on MAP, determine vehicle's travel lane, distance- and time-to-arrival at the stop-bar
  + Generate and encode SRM
  + Communicate with OBU\_msgTransceiver
    - Receive MAP, SPaT and SSM
    - Transmit BSM and SRM

The MMITSS-CA software components run on three (3) physical devices (platforms):

* The MRP\_xxx software components (version 2.0) are written in C++11 and run on an Ubuntu Linux computer located inside the traffic control cabinet;
* The RSU\_xxx software component (version 2.0) is written in C++, compiled in Savari On-Board Operating System (SOBOS), and runs on Savari StreetWAVE (RSU) version 3.1.2; and
* The OBU\_xxx software components (version 2.0) are written in C++11, compiled in Savari SOBOS, and run on Savari MobiWAVE (OBU) version 3.1.2.
* The executable files under directory of ‘rsu/bin’ and ‘obu/bin’ can be directly used in version 3.1.2 Savari StreetWAVE and MobiWAVE devices, respectively. For newer version StreetWAVE or MobiWAVE devices, the source code for RSU\_ and OBU\_ components need to be compiled with the corresponding version of Savari SOBOS VMware image.
* The SmartCross phone app is maintained and supported by Savari.

In addition, three (3) shared libraries are used to provide common functions for the MRP and OBU components. These common functions include 1) encoding and decoding of SAE J2735 DSRC messages, and 2) locating vehicle location on MAP:

* libasn is created with a set of C source code (under ‘mrp/asn1’ directory) which are outputted from the open source ASN.1 compiler asn1c (<https://github.com/vlm/asn1c>), using version 2016-03 SAE J2735 message set definition (<https://www.sae.org/standardsdev/dsrc/>);
* libdsrc is compiled with C++11 source code (under ‘mrp/asn1j2735’ directory) which provide encoding and decoding functions for SAE J2735 (version 2016-03) MAP, SPaT, BSM, SRM, and SSM messages; and
* liblocAware is created with C++11 source code (under ‘mrp/ locationAware’ directory) to provide function of locating a vehicle on MAP.

## Tab 2: Release Notes:

### License

The MMITSS-CA software set is licensed under the Educational Community License (ECL) version 2.0. This software set is intended for use in Connected Vehicle research or deployment projects supported by the U.S. Department of Transportation and shall not be used for, or with, commercial products. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

This software set makes use of open source ASN.1 compiler ‘asn1c’, which is licensed under BSD 2-Clause (‘simplified’ BSD License). These license information is shown below.

1. Educational Community License version 2.0

The Educational Community License version 2.0 ("ECL") consists of the Apache 2.0 license, modified to change the scope of the patent grant in section 3 to be specific to the needs of the education communities using this license.

Licensed under the Educational Community License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.osedu.org/licenses/ECL-2.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1. APACHE version 2.0

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1. BSD 2-Clause (“Simplified BSD License”)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1). Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2). Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

### Installation and Removal Instructions

The installation instructions for MRP components on the Ubuntu Linux computer, RSU components on Savari StreetWAVE devices, and OBUS components on Savari MobiWAVE devices are described in README under directory ‘mrp’, ‘rsu’, and ‘obu’, respectively.

* MRP software installation (on Ubuntu Linux computer)
  + copy ‘mrp’ directory to Ubuntu /home/MMITSS-CA;
  + add ‘export MRP\_MK\_DEFS=/home/MMITSS-CA/mrp/build/mrp\_linux.mk’ to system wide configure file /etc/bash.bashrc or to user’s configure file ~/.bashrc;
  + verify socket configurations for files under ‘mrp/conf’ directory are consistent with your Ubuntu settings;
  + cd /home/MMITSS-CA/mrp; make startup; - enable MRP executables to as Systemd service
  + make install; make install; and
  + reboot Ubuntu, or start MRP executables manually with ‘sudo systemctl start mmitss.mrp.service’
* MRP software removal (on Ubuntu Linux computer)
  + sudo systemctl stop mmitss.mrp.service;
  + sudo systemctl disable mmitss.mrp.service;
  + sudo systemctl daemon-reload; and
  + delete directory of /home/MMITSS-CA
* RSU software installation (on Savari StreetWAVE device)
  + copy ‘rsu/bin’ and ‘rsu/conf’ to RSU /home/MMITSS-CA;
  + chmod +x /home/MMITSS-CA/bin/\*;
  + on RSU, mkdir /nojournal/wmelogs;
  + copy /home/MMITSS/bin/msgTransMon to /etc/init.d/;
  + cd /etc/rc.d/; ln -s ../msgTransMon S99msgTransMon; and
  + reboot RSU, or start RSU executables manually with /etc/init.d/msgTransMon start.
* RSU software removal (on Savari StreetWAVE device)
  + /etc/init.d/msgTransMon stop;
  + delete /etc/rc.d/S99msgTransMon;
  + delete /etc/init.d/msgTransMon; and
  + delete directory of /home/MMITSS-CA
* OBU software installation (on Savari MobiWAVE device)
  + copy ‘obu/bin’ and ‘obu /conf’ to OBU /home/MMITSS-CA;
  + chmod +x /home/MMITSS-CA/bin/\*;
  + on OBU, mkdir /nojournal/ mmitss\_logs/awr, and /nojournal/mmitss\_logs/wme;
  + copy /home/MMITSS/bin/msgTransMon and obuMon to /etc/init.d/;
  + cd /etc/rc.d/; ln -s ../msgTransMon S96msgTransMon; ln -s ../obuMon S99obuMon; and
  + reboot OBU, or start OBU executables manually with
    - /etc/init.d/msgTransMon start; and
    - /etc/init.d/obuMon start
* OBU software removal (on Savari MobiWAVE device)
  + /etc/init.d/msgTransMon stop;
  + /etc/init.d/obuMon stop;
  + delete /etc/rc.d/S99msgTransMon S99obuMon;
  + delete /etc/init.d/msgTransMon obuMon; and
  + delete directory of /home/MMITSS-CA

### Operating requirements

* MRP Ubuntu Linux computer
  + Processing power: intel Core i5 or equivalent
  + Minimum memory: 4 GB
  + Hard drive space: 500 GB or above
  + Connectivity: Ethernet, 2 RS232
  + Operating system: Ubuntu 16.02

### Related web sites

* The software is distributed through the USDOT's JPO Open Source Application Development Portal (OSADP), <http://itsforge.net/>
* MMITSS project background and detailed documentations is available through the Connected Vehicle Pooled Fund Study website, <http://www.cts.virginia.edu/cvpfs_research/> .

## Tab 3: Documentation

* [MMITSS Concept of Operations](http://www.cts.virginia.edu/wp-content/uploads/2014/05/Task2.3._CONOPS_6_Final_Revised.pdf)
* [MMITSS-CA Detailed System and Software Design](http://www.cts.virginia.edu/wp-content/uploads/2014/04/32-MMITSS-Phase-2-Detailed-Design-CA-final.pdf)

## Tab 4: Discussion

* Main discussion (link)
* Issue discussion (link)

## Tab 5: Related Applications

* List of all applications with same categorization