

**Multi Modal Intelligent Traffic Signal System**

**Field Deployment – User Manual**

Revision 0.0

(Initial Release)

July 12, 2020

**Table of Contents**

1. Purpose of Document 3

2. Systems Requirements: 3

3. Deployment – Docker Containers 3

# Purpose of Document

This document is an instruction guide for deploying Multi-Modal Intelligent Traffic Signal System (MMITSS) applications in the real world. The document contains the detailed configuration and usage instructions for deploying the MMITSS software components in the docker container.

# Systems Requirements:

To deploy MMITSS in the real world following requirements are required to meet

1. MMITSS roadside software components can be run in the Connected Vehicle Co-Processor (CVCP). To run MMITSS roadside software components in CVCP, install Ubuntu Bionic 18.04.3 LTS operating systems.The operating systems can be installed by following the instruction from [https://boundarydevices.com/ubuntu-bionic-18-04-3-lts-for-i-mx6-7-boards-august-2019-kernel-4-14-x/#](https://boundarydevices.com/ubuntu-bionic-18-04-3-lts-for-i-mx6-7-boards-august-2019-kernel-4-14-x/)
2. MMITSS vehicle side software components can be run in the Raspberry pi. To run MMITSS vehicle software components in the Raspberry pi, Ubuntu 18.04 Server operating systems can be installed.
3. Install docker in the mmitss roadside processor (mrp) and vehicleside processor (vsp) arm box.
4. If the arm box won’t have the internet access in the field, download and load docker image (mmitssuarizona/mmitss-mrp-v2.0 docker image for roadside applications or download mmitssuarizona/mmitss-vsp-arm-v1.0)

docker pull <image name>

docker image load –i <directory of the image>

# Deployment – Docker Containers

To deploy MMITSS software components in the field, following steps can be followed:

**Step1:** Create configuration file

It is required to create mmitss-phase3-master-config.json, mmitss-data-external-clients.json configuration files for mrp container and mmitss-phase3-master-config.json for vsp container. The configuration files contain the IP addresses, UDP ports etc. which are required to establish communication between the MMITSS software components.

{

"HostIp": "xxx.xxx.xxx.xxx",

"SourceDsrcDeviceIp": "xxx.xxx.xxx.yyy",

"IntersectionName": "xxx",

"MapPayload":001283fe38083020315abe2149d0eecf1800a0000271c4fcbd028280",

"IntersectionID" : XXXX,

"RegionalID" : 0,

"DataCollectorIP": "xxx.xxx.xxx.xyx",

"HMIControllerIP": "xxx.xxx.xxx.yxx",

"MessageDistributorIP": " xxx.xxx.xxx.zzz ",

"PriorityRequestGeneratorServerIP": "xxx.xxx.xxx.zzz",

"VehicleType" : 6,

"Logging" : "True",

"SRMTimedOutTime" : 10.0,

"PortNumber":

{

"MessageTransceiver":

{

"MessageSender": 10003,

"MessageReceiver": 10002,

"MessageEncoder": 10003,

"MessageDecoder": 10002

},

"MessageDistributor": 5000,

"RsmDecoder": 10006,

"OBUBSMReceiver": 10005,

"HostBsmDecoder": 10005,

"TrajectoryAware": 20001,

"PriorityRequestServer": 20002,

"PrioritySolver": 20003,

"PriorityRequestGenerator": 20004,

"TrafficControllerInterface": 20005,

"TrafficControllerCurrPhaseListener": 20006,

"TrafficControllerTimingPlanSender": 20007,

"PerformanceObserver": 20008,

"HMIController": 20009,

"PrioritySolverToTCIInterface": 20010,

"SignalCoordination": 20011,

"MapSPaTBroadcaster": 6053,

"DsrcImmediateForwarder": 1516,

"PriorityRequestServer\_SendSSM": 50003,

"DataCollector": 30006,

"SnmpEngine": 20020,

"SnmpEngineInterface": 20021,

"PriorityRequestGeneratorServer": 20022

},

"psid":

{

"map": "E0000017",

"spat": "8002",

"rsm": "8003",

"srm": "E0000019",

"ssm": "E0000020",

"bsm": "20"

},

"msgId":

{

"map": "0012",

"spat": "0013",

"rsm": "0021",

"srm\_lower": "001d",

"srm\_upper": "001D",

"ssm\_lower": "001e",

"ssm\_upper": "001E",

"bsm": "0014"

},

"SignalController":

{

"IpAddress": " xxx.xxx.xxx.yyy",

"NtcipPort": 502,

"TimingPlanUpdateInterval\_sec": 60,

"NtcipBackupTime\_sec": 300,

"Vendor": "Econolite",

"TimingPlanMib": "/nojournal/bin/EconoliteMib.py",

"InactiveVehPhases":[],

"InactivePedPhases":[],

"SplitPhases":

{

"1": 6,

"3": 8,

"5": 2,

"7": 4

},

"PermissiveEnabled":

{

"1": true,

"3": true,

"5": true,

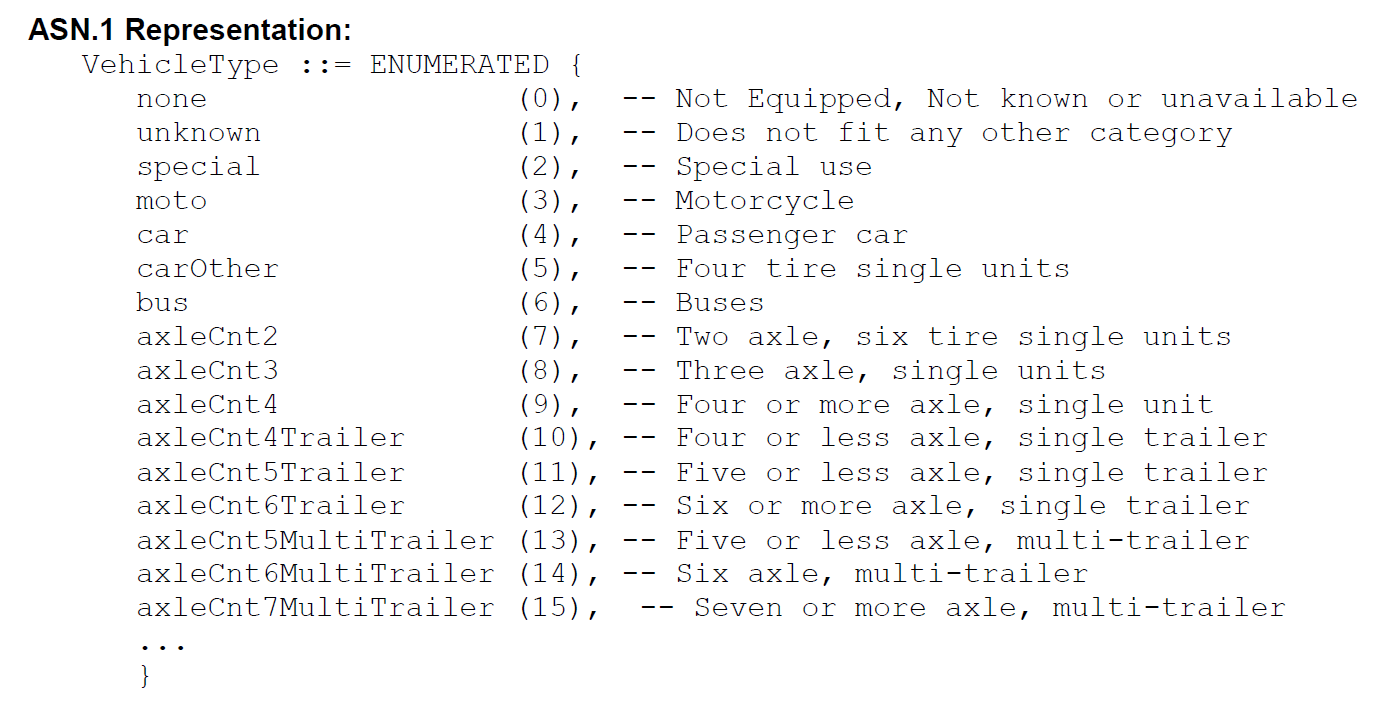
"7": true

}

}

}

1. For mrp container, “*HostIp”,* “*SourceDsrcDeviceIp”, “IntersectionName”, “MapPayload”, “IntersectionID”*, *signal controller* “*IpAddress”, “NtcipPort”,* and “*NtcipBackupTime\_sec”, “Vendor”, “TimingPlanMib”* are required to be specified. The *HostIP* must match the ip address of the connected vehicle co-processor (CVCP). The “*SourceDsrcDeviceIp”*. The map payload can be obtained by creating an intersection map using USDOT map tool (<https://webapp.connectedvcs.com/isd/>).
2. For vsp container, *“HostIp”,* “*SourceDsrcDeviceIp”,* “*VehicleType”* are required to be specified. The vehicle type has to be specified based on *J2735 2016* standard.



1. Create a log folder in the nojournal/bin directory for each intersection and simulation tools. To log the data, specify *“Logging”: “True”* in the mmitss-phase3-master-config.json file.

**Step 2:** Launch scripts

To run the mrp/vsp container mmitss\_launch\_docker\_arm.sh is required. The script can be placed in the /home directory.

read -p "Full absolute path of MMITSS configuration directory: " config\_path

read -p "Name of container image on the Dockerhub: " container\_image

read -p "Name of container: " container\_name

docker run -v $config\_path:/nojournal --network host --name container\_name $container\_image > /dev/null 2>&1 &

**Step 3:** Define the source of configuration files

1. Go to cd /home directory
2. Create a folder which can be named as intersection name (for mrp container) or vehicle type (for vsp container). Then create bin folder. In the bin folder place the mmitss configuration files and create log folder. For example-

cd /home

mkdir emergency-vehicle

cd emergency-vehicle

mkdir /bin

cd bin

cp /home/ mmitss-phase3-master-config.json mmitss-phase3-master-config.json

mkdir log

**Step 4:** Run docker container

1. Run following script to run the docker container:

mmitss\_\_launch\_docker\_arm.sh

Full absolute path of MMITSS configuration directory: /home/emergency-vehicle

Name of container image on the Dockerhub: mmitssuarizona/mmitss-vsp-arm-v1.0

Name of container: vsp\_container

1. To monitor the containers execute the following command:

Docker container exec –it <container name> /bin/bash

1. To stop all the containers, execute the following command:

Docker container stop <container name>

1. To start container the execute the following command:

Docker container start <container name>