

**Multi Modal Intelligent Traffic Signal System**

**Build Docker Image – User Manual**

Revision 2.1

April 30, 2021

Contents

[Contents 2](#_Toc70638636)

[1. Purpose of Document 2](#_Toc70638637)

[2. Types of Docker Containers 2](#_Toc70638638)

[3. Prerequisites 3](#_Toc70638639)

[4. Building MMITSS Applications and Docker Containers 4](#_Toc70638640)

# Purpose of Document

The distribution and deployment of Multi-Modal Intelligent Traffic Signal System (MMITSS) is managed through docker containers, which provide containerized environment that includes pre-built libraries and 3rd party packages, along with binary executables of various components of MMITSS. This document describes the process of building different types of docker containers that are required to run MMITSS applications.

# Types of Docker Containers

Four types of docker container images that are built and used in the MMITSS are as follows:

1. *Base Containers:*

The base containers are derived from open source docker container ubuntu:20.04. In addition, the directory structure, necessary libraries, and 3rd party packages that are required for running MMITSS components are installed in the base container. On the docker-hub, the base containers are available as mmitssuarizona/mmitss-x86-base for x86-architecture processors, and mmitssuarizona/mmitss-arm-base for arm-architecture processors. All remaining types of containers that are described below are derived from the base containers.

1. *Roadside Processor (MRP) Containers:*

MRP containers are loaded with executables of all MMITSS applications that are required for roadside deployment of MMITSS, which are as follows:

1. Wireless Message Decoder
2. Message Encoder
3. Traffic Controller Interface
4. Map Spat Broadcaster
5. Snmp Engine
6. Priority Request Solver
7. Priority Request Server
8. Signal Coordination Request Generator
9. Trajectory Aware
10. System Interface
11. V2X Data Collector
12. V2X Data Transfer

On the docker-hub, the MRP containers are available as mmitssuarizona/mmitss-mrp-x86 for x86-architecture processors, and mmitssuarizona/mmitss-mrp-arm for arm-architecture processors. The MRP containers are used for both, field deployment and simulation of MMITSS.

1. *Vehicle-side Processor (VSP) Containers:*

VSP containers are loaded with executables of all MMITSS applications that are required for vehicle-side deployment of MMITSS, which are as follows:

1. Wireless Message Decoder
2. Message Encoder
3. Host Bsm Decoder
4. Priority Request Generator
5. Light Siren Status Manager
6. System Interface
7. V2X Data Collector
8. Data Compressor

On the docker-hub, the VSP containers are available as mmitssuarizona/mmitss-vsp-x86 for x86-architecture processors, and mmitssuarizona/mmitss-vsp-arm for arm-architecture processors. The VSP containers are only used for the field deployment of MMITSS.

1. *Simulation/Server Tools Containers:*

Simulation/Server Tools containers are loaded with executables of all MMITSS applications that are required for simulation of MMITSS, which are as follows:

1. Priority Request Generator Server
2. Message Distributor
3. Simulated Bsm Blob Processor
4. System Interface

On the docker-hub, the Simulation/Server Tools containers are available as mmitssuarizona/mmitss-simulation\_server-tools-x86 for x86-architecture processors, and mmitssuarizona/mmitss-simulation\_server-tools-arm for arm-architecture processors. These containers are only used for the simulation of MMITSS.

# Prerequisites

To build MMITSS applications and docker container images, following prerequisites must be satisfied:

1. Processor Architecture: x86 or arm
2. Operating System: Ubuntu 18.04LTS or 20.04LTS
3. Update Ubuntu repositories using:

sudo apt-get update

1. Install required system packages using the following command:

sudo apt-get install -y build-essential libssl-dev zlib1g-dev python3-pip python3-pil git

1. Install docker and docker-compose using:

sudo snap install docker

1. Install required python packages using:

pip3 install pyinstaller apscheduler sh psutil haversine bitstring numpy pysftp flask flask\_bootstrap flask\_wtf pytz

1. If building VSP applications on Raspberry Pi, install libraries that facilitate the use of automation hat, using the following command:

pip3 install automationhat ST7725

1. Install CyVerse iCommands by following the official instructions available on <https://learning.cyverse.org/projects/data_store_guide/en/latest/step2.html>

# Building MMITSS Applications and Docker Containers

After ensuring that the prerequisites are met, execute the following steps to build MMITSS applications and Docker containers:

1. Open a terminal and go to the directory <location> where MMITSS open-source repository can be cloned, by executing the following command:

cd <location>

1. Clone the MMITSS open-source repository:

git clone https://www.github.com/mmitss/mmitss-az

1. Now go to the “build/scripts” directory inside the mmitss-az repository:

cd mmitss-az/build/scripts

1. Setup the build environment by executing the following command, and by providing the required user inputs:

./setup-build-environment.sh

1. After setting up the build environment, execute the following command to run the self-guided script. Provide the required user inputs to build MMITSS applications and docker images:

./build-mmitss.sh