

**Multi Modal Intelligent Traffic Signal System**

**Build Docker Image – User Manual**

Revision 0.3

February 08, 2020

**Table of Contents**

1. Purpose of Document 3

2. Systems Requirements 3

3. Build Docker Image 3

**List of Figures**

Figure 1: Snapshot of the Dockerfile for base image (arm) 4

Figure 2: Snapshot of the Dockerfile for base image (x86) 4

Figure 3: Snapshot of the Dockerfile for the VSP image 5

Figure 4: Snapshot of the Dockerfile for the MRP image 6

**Revision History**

|  |  |
| --- | --- |
| **Revision No. (Date)** | **Description** |
| 0.2  (09/13/2020) | (1) Updated system requirements  (2) Updated the name of build-scripts |
| 0.3  (09/13/2021) | (1) Updated snapshot of the Dockerfile for the VSP and MRP |

# Purpose of Document

This document is a guide for building docker images to deploy Multi-Modal Intelligent Traffic Signal System (MMITSS) software components in docker containers. The base image contains the source code, libraries, dependencies, tools, and other files which are required to run the applications. The vehicle side processor (VSP) image or mmitss roadside processor (MRP) image can be built by extending the base image. The VSP image contains only those applications which are required from the connected vehicle perspective and or MRP image contains only those applications which are required from the intersection perspective.

# Systems Requirements

To build docker image for deploying MMITSS in the docker container, following requirements are required to meet:

1. MMITSS roadside software components can be run in the Connected Vehicle Co-Processor (CVCP). To build the docker image and run the MMITSS roadside software components in the CVCP, install Ubuntu Bionic 18.04.3 operating systems. The operating systems can be installed by following the instructions found in [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 are run on a Raspberry Pi. Building the docker image and running the MMITSS vehicle software components on the Raspberry Pi, Ubuntu 18.04 Server operating systems must be installed.
3. Both roadside and vehicle side applications can be run on the x86 platform. Building the docker image and running the MMITSS software components on the x86 platform require that the Ubuntu 18.04.4 operating systems is installed.
4. Update the ubuntu repositories by entering following command:

sudo apt-get update

1. Once updating of repositories is complete, run the following command to install required packages

sudo apt-get install -y chrony build-essential tcpdump libssl-dev zlib1g-dev python3-pip

1. Install the required python packages

sudo pip3 install pyinstaller

sudo pip3 install apscheduler

sudo pip3 install sh psutil

sudo pip3 install haversine

1. Install CyVerse iCommands by following the official instructions available on <https://learning.cyverse.org/projects/data_store_guide/en/latest/step2.html>
2. Install docker and supervisor and clone the mmitss repository.
3. If MMITSS path is not set already, set the MMITSS path in the .bashrc file by executing the following command:

export /MMITSS\_ROOT=<mmitss directory>

For example, if mmitss is cloned on home/user directory then the command will be:

export /MMITSS\_ROOT=/home/user

1. Run the script lmmitss-initialize available at mmitss/scripts to initialize the libraries required for building mmitss application.

# Build Docker Image

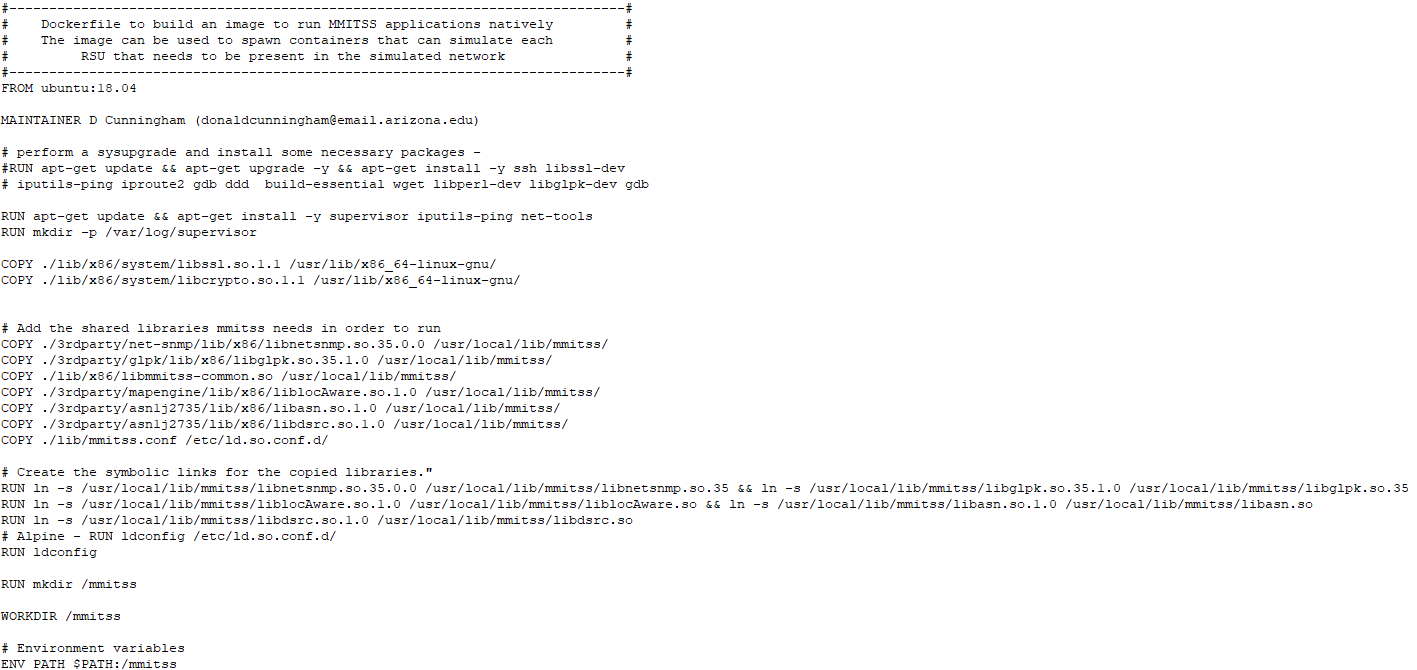
Docker images for VSP and MRP are available on Docker Hub. However, if you wish to build your own images then the following applies. To build the docker image for deploying the MMITSS software components requires the following:

**Step 1:** Define a Dockerfile for the base image

The base image can be built for an arm or a x86 machine. The dockerfile for the base image is located in the root level of the mmitss repository. Examples of the base image of the Dockerfiles for the arm and x86 machine are as follows. The only difference between the architectures would be the naming of the base image and libraries i.e. arm or x86.



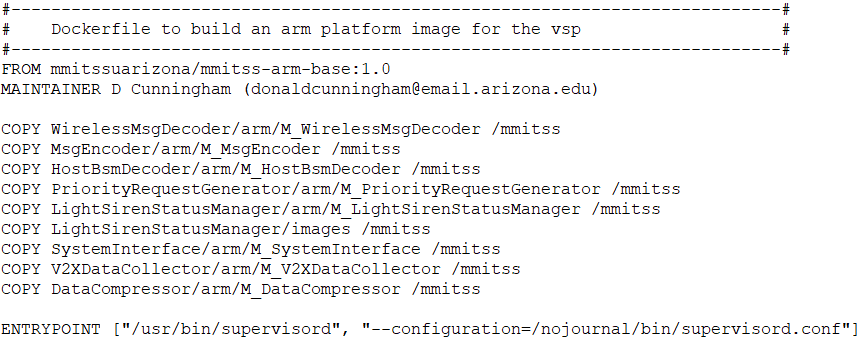
**Figure 1:** Snapshot of the Dockerfile for base image (arm)



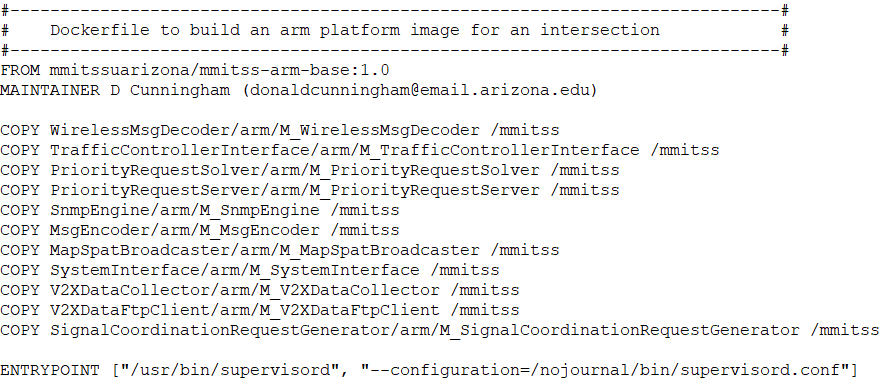
**Figure 2:** Snapshot of the Dockerfile for base image (x86)

**Step 2:** Define a Dockerfile for the VSP/MRP

The binaries included in the VSP and MRP are different. Thus, it is necessary to create a unique VSP and MRP docker image. To create a docker image for either VSP or MRP, a unique Dockerfile for each platform is necessary. In this Dockerfile, the name of the base image has to be defined correctly for the platform of interest. Examples of the Dockerfiles for the VSP and MRP are as follows:



**Figure 3:** Snapshot of the Dockerfile for the VSP image



**Figure 4:** Snapshot of the Dockerfile for the MRP image

**Step 3:** Build the docker image

To build the docker image for VSP and MRP following commands can be executed sequentially:

1. Go to the directory where base image dockerfile is located. For example:

cd /home/user/mmitss

1. Build the base docker image.

docker build –t <name of the image>:<tag> -f <path to dockerfile> .

For example, to build the base image for the arm box following command can be executed:

docker build –t mmitssuarizona/mmitss-arm-base:1.0 -f /home/user/mmitss/Dockerfile.arm\_base .

1. Go to mmitss/scripts directory and build all applications For example to build all the applications for the arm box, execute the following:

cd /home/user/mmitss/scripts

./build-arm.sh

To build all the applications for the x86 box, run the following script:

./build-x86-simulaton.sh

1. Build VSP or MRP image by executing the following command

docker build –t <name of the image>:<tag> -f <path to dockerfile> .

For example, to build the VSP docker image for the arm box, following command can be executed:

docker build –t mmitssuarizona/mmitss-vsp-arm:1.1 -f /home/user/mmitss/bin/dockerfiles/arm/Dockerfile-vsp.arm .