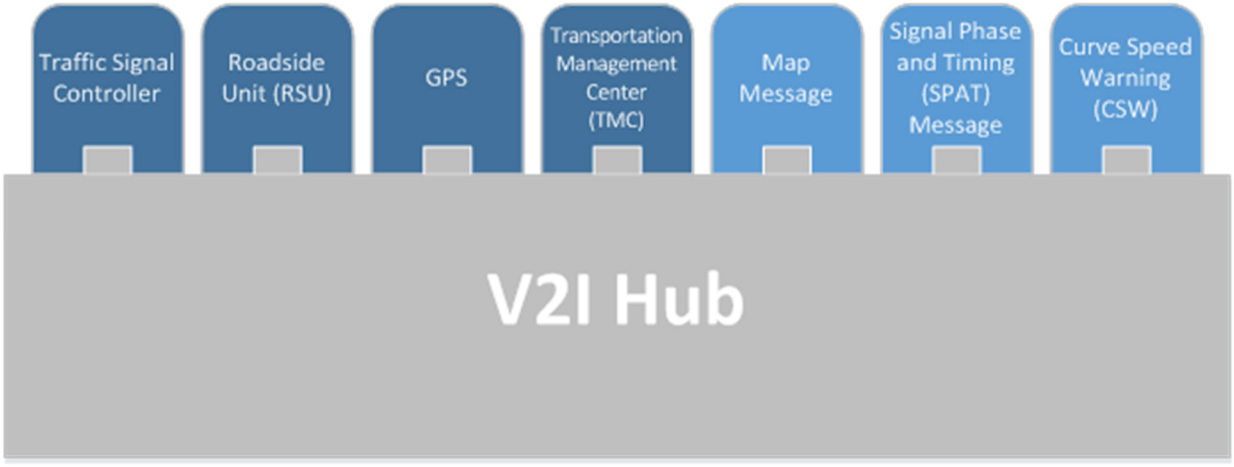
# V2I Hub

## Software Configuration Guide

www.its.dot.gov/index.htm

**Final Report – July 3, 2018**
**FHWA-JPO-18-645**




U.S. Department of Transportation

Produced by Battelle Memorial Institute under DTFH61-12-D-00040
U.S. Department of Transportation
Office of the Assistant Secretary for Research and Technology
Joint Program Office

# Notice

**Technical Report Documentation Page**

| 1. Report No. | 2. Government Accession No. | 3. Recipient's Catalog No. |
|---|---|---|
| **FHWA-JPO-645** | | |

| 4. Title and Subtitle | 5. Report Date |
|---|---|
| V2I Hub Software Configuration Guide | July 3, 2018 |
| | **6. Performing Organization Code** |
| | Battelle |

| 7. Author(s) | 8. Performing Organization Report No. |
|---|---|
| Gregory Zink, Greg Baumgardner | |

| 9. Performing Organization Name and Address | 10. Work Unit No. (TRAIS) |
|---|---|
| Battelle | |
| 505 King Ave | **11. Contract or Grant No.** |
| Columbus, OH 43201-2693 | DTFH61-12-D-00040 |

| 12. Sponsoring Agency Name and Address | 13. Type of Report and Period Covered |
|---|---|
| Federal Highway Administration | Final |
| 1200 New Jersey Avenue, S.E. | **14. Sponsoring Agency Code** |
| Washington, DC 20590 | FHWA |

**15. Supplementary Notes**

**16. Abstract**

Connected Vehicle technologies help reduce the number of driving related injuries and fatalities by allowing road users to be aware of potentially dangerous situations on the road. There are two main types of Connected Vehicles communications, vehicle-to-infrastructure and vehicle-to-vehicle. Vehicle-to-infrastructure communication takes place between vehicles and deployed roadside communication devices, which capture vehicle generated data while providing information pertaining to safety, mobility, and environmental conditions

This software configuration guide provides instructions and examples on how to configure the plugins for infrastructure V2I Hub deployment with MAP and SPaT.

| 17. Keywords | 18. Distribution Statement |
|---|---|
| Connected Vehicle (CV), V2V, V2I, safety, deployment, V2I Hub, IVP, SPaT, V2I Reference Implementation, Infrastructure Connectivity Equipment | Unlimited |

| 19. Security Classif. (of this report) | 20. Security Classif. (of this page) | 21. No. of Pages | 22. Price |
|---|---|---|---|
| Unclassified | Unclassified | 44 | |

**Form DOT F 1700.7 (8-72)** **Reproduction of completed page authorized**

# Table of Contents

## List of Tables



## List of Figures

# Executive Summary

The V2I Hub Software Configuration Guide was developed during the V2I Reference Implementation project for IT staff deploying and configuring the V2I Hub in supporting Connected Vehicle (CV) technology. This guide gives detailed instructions and examples needed by IT staff when configuring the V2I Hub for their infrastructure deployment. The V2I Hub is a CV platform developed to be deployed at signalized intersections, and other infrastructure locations, with the intention of making roadways safer and smarter by reducing accidents and providing informational alerts to drivers. This document is a part of the V2I Hub series of documents produced by the V2I Reference Implementation project. The rest of the documents are listed below. It is suggested to start with a review of the V2I Hub Guidebook.

- V2I Hub Guidebook
- V2I Hub Plugin Programming Guide
- V2I Hub Plugins
- V2I Hub Deployment Guide
- V2I Hub Administration Portal User Guide.

# Chapter 1. Introduction

## V2I Hub

The Integrated Vehicle-to-Infrastructure Prototype (IVP), called V2I Hub, is part of U.S. DOT's Vehicle-to-Infrastructure (V2I) Program and was developed to support jurisdictions in deploying V2I technology by reducing infrastructure connectivity equipment integration issues and enabling use of their existing transportation management hardware and systems. V2I Hub is a software platform that utilizes plugins to translate and route messages between different devices in support of running connected vehicle safety applications on roadside equipment.

The V2I Hub software was created and tested on Ubuntu 16.04 LTS but can run on most Linux operating systems. The V2I Hub software uses a modular plugin architecture so that each installation can be custom configured to run a selected set of software applications. The V2I Hub software contains the communication routing for the plugins, the configuration for the plugins, and the processes to start, stop, and monitor the plugins. Each plugin in the V2I Hub software is created to perform one or more related functions, such as communicate with a signal controller and produce Signal Phase and Timing (SPaT) messages.

## Deployment Assumptions

This document assumes that the end user already has completed the installation instructions for the hardware outlined in the *V2I Hub Deployment Guide (Battelle, March 2018)*. The Software Configuration Guide walks the user through the installation of V2I Hub plugins and configuration for the system to operate at a signalized intersection or other infrastructure location. Similar processes can be used to tailor the V2I Hub deployment per requirements set forth from the deploying agency. The Software Configuration Guide also assumes that the deploying agency has access to test equipment, including a DSRC packet sniffer like the 3M DSRC Packet Sniffer, an OBU with RSU radio, or similar equipment to view the messages transmitted by the RSU.

# Chapter 2. V2I Hub Configuration

## V2I Hub Setup

The V2I Hub software should be installed onto your device either by purchasing a device with the V2I Hub already installed, or by downloading the V2I Hub from OSADP and following the directions to compile and install for your platform. Please visit https://www.itsforge.net/ to obtain the source code for V2I Hub. If installing V2I Hub on your own hardware see Table 1 for the recommended hardware requirements. The V2I Hub was developed to run on a Linux computer running Ubuntu 16.04 LTS. This operating system is the supported operating system for the V2I Hub, but other Linux operating systems could be used by compiling the V2I Hub software package for your own distribution.

**Table 1. V2I Hub Hardware Requirements**

| Hardware Component | ARM Hardware Specifications | Intel Hardware Specifications |
|---|---|---|
| Processor | Freescale iMX6 ARM Cortex A9 Quad Core 1 GHz | Intel Core i3 processor Dual Core 1.7 GHz |
| Memory | 1 GB DDR3 | 4 GB RAM |
| Hard Drive Space | 16 GB SSD | 16 GB SSD |
| Operating System | Ubuntu 16.04 LTS | Ubuntu 16.04 LTS |

For our deployment we will be installing the V2I Hub at a signalized intersection to send SAE J2735-2016 SPaT and MAP messages. The V2I Hub must communicate with a TS2 traffic signal controller providing the Traffic Signal Controller Broadcast Message over Ethernet to send the SPaT message. A DSRC RSU meeting the RSU 4.1 specification must be installed for DSRC transmissions from the V2I Hub.

## V2I Hub Network Configuration for Ubuntu

If the V2I Hub has not been configured for your network, you will need to use a monitor and keyboard to access the device to change the network settings or access the device via SSH. Once you have a terminal follow these instructions to change the settings for your V2I Hub.

- Once the command prompt appears, change directory to /etc/network (cd /etc/network).
- Open the interfaces file in a text editor like nano (sudo nano interfaces) or a text editor of your choice as super user by prefixing the command with “sudo”.
- Modify the IP address, netmask, gateway and dns-nameservers of your V2I Hub to the desired settings for your network.
- Save the file (nano ctrl + X, enter Y for save file).

- Reboot the V2I Hub (sudo reboot). This will terminate the connection.

One thing to note is that some hardware might have a read only file system, which prohibits changing the IP address in the above manner. Please consult your hardware manufacturer's user guide for specific instructions on how to change the IP address for purchased hardware.

This section explains the minimal networking setup to get the V2I Hub operational. The V2I Hub does not have built in security and relies on the platform it is installed on for security due to the diversity of installation options. Consult your network administrator for network settings that meet the requirements of your network.

# V2I Hub Application Software Installation

Once you have the V2I Hub installed on your device, plugins will be needed to be installed to add functionality. For a list of plugins created for the V2I Hub during the IVP and V2I Reference Implementation projects, please consult the *V2I Hub Plugins* document. This document contains details including configuration information for the V2I Hub plugins. The plugins that will be installed depend on the type of deployment. If the deployment is at a signalized intersection, the recommended plugins are: DSRC Message Manager, MAP, and SPaT. These plugins together will send MAP and SPaT for a signalized intersection out of an RSU.

Once the deployed plugins have been decided, installation packages for those plugins will need to be obtained. This can be done by contacting the solution provider for any equipment which came pre-installed with the V2I Hub, or by using the obtained source code and compiling and creating the installation packages for the plugins that have been selected.

Installation of the selected V2I Hub plugins will be done using the V2I Hub Admin Portal. Full documentation for the V2I Hub Admin Portal can be found in the *V2I Hub Administration Portal User Guide.* Below we will outline how to install the DSRC Message Manager, MAP, and SPaT plugins for our sample deployment.

## Install Plugins

To install the plugins, you will need to obtain the plugin installation packages for the DSRC Message Manager, MAP plugin and SPaT plugin either from your hardware provider or by compiling the source code for the V2I Hub from OSADP. They will either be a Debian installation package ending in .deb or an archive file ending in .zip or .tar.gz. Once the installation packages have been obtained, install the plugins with the admin portal using the steps below.

1. Log into the V2I Hub Administration Portal.
2. Navigate to the Plugins tab.
3. Click the "Upload File" button and follow the prompts.
4. Click the "Upload Plugin" button on the dialog that appears.
5. Click the "Browse.." button.
6. Select the installation package when requested. The plugin installation can be of any of the following compressed types - .deb, .zip, .tar.gz, and .tgz.
7. Once file is selected, click the "Submit" button.

8. The download and install status will appear at the bottom of the portal.
9. Upon successful completion, the plugin will appear in the plugin list as disabled.

# V2I Hub Software Configuration

## DSRC Message Manager Configuration

The DSRC Message Manager listens for V2I Hub messages that need to be routed out the DSRC radio, packages the messages, and transmits them to the DSRC radio using the Immediate Forward protocol found in the RSU 4.1 specification. The DSRC Message Manager plugin can be configured to send the messages to four different destinations. A destination is dependent on the type of RSU used. For example, the Lear RSU must have an immediate forward application running for each Provider Service Identifier (PSID) which means to send four different PSIDs to one RSU four different destinations would need to be configured to communicate with the four different instances of the immediate forward application running on the Lear RSU. Another example uses only one destination for a Cohda RSU since the Cohda RSU immediate forward application does not limit its inputs based on PSID. Examples for configuration are shown in the following sections.

### *DSRC Message Manager Sending SPaT and MAP to One RSU*

Sending SPaT and MAP out of one RSU that has either an immediate forward application per PSID or one immediate forward application for all PSIDs, uses the same configuration settings. This is because the P-encoded PSID value for both MAP and SPaT are 0x8002 (or 0x82 hex). Use the configuration in Table 2 for sending SPaT and MAP out of one RSU. Note that for the Lear RSU the PSID values are the two-character hexadecimal code, not the P-encoded four-character code. For example, MAP and SPaT are 0x82 and not 0x8002.

**Table 2. DSRC Message Manager Configuration Values Example 1**

| Key | Value | Description |
|---|---|---|
| Messages_Destination_1 | { "Messages": [ { "TmxType": "SPAT-P", "SendType": "SPAT", "PSID": "0x82" }, { "TmxType": "MAP-P", "SendType": "MAP", "PSID": "0x82" } ] } | JSON data defining the message types and PSIDs for messages forwarded to the DSRC radio at destination 1. |
| Messages_Destination_2 | { "Messages": [ ] } | JSON data defining the message types and PSIDs for messages forwarded to the DSRC radio at destination 2. |
| Messages_Destination_3 | { "Messages": [ ] } | JSON data defining the message types and PSIDs for messages forwarded to the DSRC radio at destination 3. |
| Messages_Destination_4 | { "Messages": [ ] } | JSON data defining the message types and PSIDs for messages forwarded to the DSRC radio at destination 4. |
| Destination_1 | 192.168.55.43:1516 | The destination UDP server(s) and port number(s) on the DSRC radio for all messages specified by Messages_Destination_1. |
| Destination_2 | 0 | The destination UDP server(s) and port number(s) on the DSRC radio for all messages specified by Messages_Destination_2. |
| Destination_3 | 0 | The destination UDP server(s) and port number(s) on the DSRC radio for all messages specified by Messages_Destination_3. |
| Destination_4 | 0 | The destination UDP server(s) and port number(s) on the DSRC radio for all messages specified by Messages_Destination_4. |
| Signature | False | True or False value indicating whether the RSU should sign the messages being transmitted. |

### *DSRC Message Manager Sending SPaT and MAP to TWO RSUs*

To send both SPaT and MAP to two different RSUs on the same network, an additional destination is needed. The configuration shown below in Table 3 shows sending SPaT and MAP to two different destinations, where destination 1 at IP 192.168.55.43 uses the hex value for PSID, and the destination 2 at 192.168.55.53 uses the P-encoded PSID value.

**Table 3. DSRC Message Manager Configuration Values Example 2**

| Key | Value | Description |
|---|---|---|
| Messages_Destination_1 | { "Messages": [ { "TmxType": "SPAT-P", "SendType": "SPAT", "PSID": "0x82" }, { "TmxType": "MAP-P", "SendType": "MAP", "PSID": "0x82" } ] } | JSON data defining the message types and PSIDs for messages forwarded to the DSRC radio at destination 1. |
| Messages_Destination_2 | { "Messages": [ { "TmxType": "SPAT-P", "SendType": "SPAT", "PSID": "0x8002" }, { "TmxType": "MAP-P", "SendType": "MAP", "PSID": "0x8002" } ] } | JSON data defining the message types and PSIDs for messages forwarded to the DSRC radio at destination 2. |
| Messages_Destination_3 | { "Messages": [ ] } | JSON data defining the message types and PSIDs for messages forwarded to the DSRC radio at destination 3. |
| Messages_Destination_4 | { "Messages": [ ] } | JSON data defining the message types and PSIDs for messages forwarded to the DSRC radio at destination 4. |
| Destination_1 | 192.168.55.43:1516 | The destination UDP server(s) and port number(s) on the DSRC radio for all messages specified by Messages_Destination_1. |
| Destination_2 | 192.168.55.53:1516 | The destination UDP server(s) and port number(s) on the DSRC radio for all messages specified by Messages_Destination_2. |
| Destination_3 | 0 | The destination UDP server(s) and port number(s) on the DSRC radio for all messages specified by Messages_Destination_3. |
| Destination_4 | 0 | The destination UDP server(s) and port number(s) on the DSRC radio for all messages specified by Messages_Destination_4. |
| Signature | False | True or False value indicating whether the RSU should sign the messages being transmitted. |

### *DSRC Message Manager Sending SPaT, MAP, TIM to One RSU (Lear)*

Table 4 shows the configuration of an RSU, like the Lear, which uses the hex two-character encoding sending SPaT, MAP and TIM. Since this hardware uses an immediate forward application per PSID, two destinations are used. One destination is setup to send the SPaT and MAP with PSDI 0x82, and the other destination to send 0x83 to a second instance of the immediate forward application on the same hardware.

**Table 4. DSRC Message Manager Configuration Values Example 3**

| Key | Value | Description |
|---|---|---|
| Messages_Destination_1 | { "Messages": [ { "TmxType": "SPAT-P", "SendType": "SPAT", "PSID": "0x82" }, { "TmxType": "MAP-P", "SendType": "MAP", "PSID": "0x82" } ] } | JSON data defining the message types and PSIDs for messages forwarded to the DSRC radio at destination 1. |
| Messages_Destination_2 | { "Messages": [ { "TmxType": "TIM", "SendType": "TIM", "PSID": "0x83" } ] } | JSON data defining the message types and PSIDs for messages forwarded to the DSRC radio at destination 2. |
| Messages_Destination_3 | { "Messages": [ ] } | JSON data defining the message types and PSIDs for messages forwarded to the DSRC radio at destination 3. |
| Messages_Destination_4 | { "Messages": [ ] } | JSON data defining the message types and PSIDs for messages forwarded to the DSRC radio at destination 4. |
| Destination_1 | 192.168.55.43:1516 | The destination UDP server(s) and port number(s) on the DSRC radio for all messages specified by Messages_Destination_1. |
| Destination_2 | 192.168.55.43:1517 | The destination UDP server(s) and port number(s) on the DSRC radio for all messages specified by Messages_Destination_2. |
| Destination_3 | 0 | The destination UDP server(s) and port number(s) on the DSRC radio for all messages specified by Messages_Destination_3. |
| Destination_4 | 0 | The destination UDP server(s) and port number(s) on the DSRC radio for all messages specified by Messages_Destination_4. |
| Signature | False | True or False value indicating whether the RSU should sign the messages being transmitted. |

### *DSRC Message Manager Sending SPaT, MAP, TIM to One RSU (Cohda)*

The final example shows how to send SPaT, MAP, and TIM to one destination who forwards all PSIDs. This case we change the JSON value and add an addition element to send the TIM message to the same destination. This is shown in Table 5.

**Table 5. DSRC Message Manager Configuration Values Example 4**

| Key | Value | Description |
|---|---|---|
| Messages_Destination_1 | { "Messages": [ { "TmxType": "SPAT-P", "SendType": "SPAT", "PSID": "0x8002" }, { "TmxType": "MAP-P", "SendType": "MAP", "PSID": "0x8002" }, { "TmxType": "TIM", "SendType": "TIM", "PSID": "0x8003" } ] } | JSON data defining the message types and PSIDs for messages forwarded to the DSRC radio at destination 1. |
| Messages_Destination_2 | { "Messages": [ ] } | JSON data defining the message types and PSIDs for messages forwarded to the DSRC radio at destination 2. |
| Messages_Destination_3 | { "Messages": [ ] } | JSON data defining the message types and PSIDs for messages forwarded to the DSRC radio at destination 3. |
| Messages_Destination_4 | { "Messages": [ ] } | JSON data defining the message types and PSIDs for messages forwarded to the DSRC radio at destination 4. |
| Destination_1 | 192.168.55.43:1516 | The destination UDP server(s) and port number(s) on the DSRC radio for all messages specified by Messages_Destination_1. |
| Destination_2 | 0 | The destination UDP server(s) and port number(s) on the DSRC radio for all messages specified by Messages_Destination_2. |
| Destination_3 | 0 | The destination UDP server(s) and port number(s) on the DSRC radio for all messages specified by Messages_Destination_3. |
| Destination_4 | 0 | The destination UDP server(s) and port number(s) on the DSRC radio for all messages specified by Messages_Destination_4. |
| Signature | False | True or False value indicating whether the RSU should sign the messages being transmitted. |

## SPaT Plugin Configuration

The SPaT plugin receives signal phase and timing information from a traffic signal controller (TSC) and sends a SAE J2735-2016 SPaT message to the V2I Hub flagged for DSRC transmission. The SPaT plugin needs to be configured to communicate with the TSC by setting the TSC_IP and TSC_Remote_SNMP_Port parameters. Those values are obtained by the configuration on the TSC, where the IP address and SNMP ports are configured. In our example, the TSC is at IP address

192.168.55.50, and listening on port 501 for SNMP communications. The Local_IP and Local_UDP_Port are used to open the UDP connection where the TSC will stream the signal phase and timing data. In this case our local IP address is 192.168.55.30 and we are using 6053 as the port the V2I Hub is using for the UDP connection. The Intersection Id is a field used to uniquely identify an intersection in the SPaT message and can be configured by the Intersection_Id configuration value. The Intersection_Name value is like the Intersetion_Id and can be set to a human readable value to easily identify the intersection.

The last configuration value is the SignalGroupMapping, which contains the configuration active signal groups for the intersection. In our example we have 8 phases, that are mapped to 8 signal groups, all with type vehicle. Configure this to include all active phases for your intersection. The phases can map to a distinct signal group, or share a signal group with another phase (i.e. phase 4 is the east-bound through, and phase 8 is the west-bound through. Both can be grouped into a single signal group 4.) In this configuration value you can also specify pedestrian signal groups. In our example we have one pedestrian signal group, which maps to pedestrian phase 2. This will send a separate signal group in the SPaT message for the phase 2 pedestrian phase. The configuration described above can be found in Table 6.

There is a third type of phase that is used at many intersections where the signal controller is programmed to combine the status of multiple phases. This type is “overlap” and the phase number corresponds to the overlap number in the signal controller.

**Table 6. SPaT Configuration**

| Key | Default Value | Description |
|---|---|---|
| Intersection_Id | 1 | The intersection id for SPaT generated by this plugin. |
| Intersection_Name | Intersection | The intersection name for SPaT generated by this plugin. |
| SignalGroupMapping | {"SignalGroups":[ {"SignalGroupId":1, "Phase":1, "Type":"vehicle"}, {"SignalGroupId":2, "Phase":2, "Type":"vehicle"}, {"SignalGroupId":3, "Phase":3, "Type":"vehicle"}, {"SignalGroupId":4, "Phase":4, "Type":"vehicle"}, {"SignalGroupId":5, "Phase":5, "Type":"vehicle"}, {"SignalGroupId":6, "Phase":6, "Type":"vehicle"}, {"SignalGroupId":7, "Phase":7, "Type":"vehicle"}, {"SignalGroupId":8, "Phase":8, "Type":"vehicle"}, {"SignalGroupId":22, "Phase":2, "Type":"pedestrian"}]} | JSON data defining a list of active signal groups and phases. |
| Local_IP | 192.168.55.30 | The IPv4 address of the local computer for receiving Traffic Signal Controller Broadcast Messages. |
| Local_UDP_Port | 6053 | The local UDP port for reception of Traffic Signal Controller Broadcast Messages from the TSC. |
| TSC_IP | 192.168.55.50 | The IPv4 address of the destination Traffic Signal Controller (TSC). |
| TSC_Remote_SNMP_Port | 501 | The destination port on the Traffic Signal Controller (TSC) for SNMP NTCIP communication. |

## MAP Plugin Configuration

The MAP plugin reads a configuration file either in XML or JSON format and creates the SAE J2735-2016 MAP messages for the V2I Hub. This message is flagged for DSRC transmission. The MAP input file can be created by using the USDOT ISD Tool. The MAP plugin accepts inputs resulting in an Extensible Markup Language (XML) format that describes the intersection with approach and egress lanes. The ISD Message Creator (https://webapp.connectedvcs.com/isd/) is a MAP and SPaT creation tool for intersections in the U.S. DOT CV Tools Library (https://webapp.connectedvcs.com/). This online tool aids

in the creation of J2735 2016 MAP files, which can be used as inputs for the V2I Hub MAP plugin. In addition to accepting XML directly, the input needed by the MAP plugin can be created by using the export tool by setting the Message Type dropdown to MAP, and copying the JavaScript Object Notation (JSON) from the MAP Data text block into a file with a .json extension. The JSON format is a lightweight data exchange format that is human readable and easy for machines to serialize and de-serialize. Alternately, the input needed by the MAP plugin can be created by using the tool to export binary (UPER Hex) data. If this option is chosen, a .txt extension should be chosen and the tight encoding option is recommended so that space conserving mode X-Y offset values are utilized rather than lengthy latititudes/longitudes. The next section explains the map input files in more detail. The input file(s) to be used must be placed in the /var/www/plugings/MAP directory.

The MAP plugin can handle multiple maps by the action information received from the SPaT plugin. To configure the MAP plugin to do this, each action must be paired with a file in the json input. In the example shown in Table 7, we have different MAP files sent for action 0 and action 1. The MAP plugin needs only action 0 to function, which is shown in Table 8.

**Table 7. MAP Configuration Multiple Actions**

| Key | Default Value | Description |
|---|---|---|
| Frequency | 1000 | The frequency to send the MAP message in milliseconds. |
| MAP_Files | {"MapFiles":[{"Action":0, "FilePath":"GID_TF.xml"}, {"Action":1, "FilePath":"GID_TF1.xml"}]} | JSON data defining a list of MAP files. One MAP file for each action set specified by the TSC. |

**Table 8. MAP Configuration Single Action**

| Key | Default Value | Description |
|---|---|---|
| Frequency | 1000 | The frequency to send the MAP message in milliseconds. |
| MAP_Files | {"MapFiles":[{"Action":0, "FilePath":"GID_TF.xml"}]} | JSON data defining a list of MAP files. One MAP file for each action set specified by the TSC. |

# V2I Hub Plugin Inputs

## MAP XML Input

The MAP plugin uses an XML input file as one option to create the SAE J2735-2016 MAP message for the intersection. The input file contains a reference point for the intersection, and approach and egress lanes. Each lane is defined by a set of nodes, which have offsets. The offset for the first node is from the reference lane, and subsequent nodes are from the previous node. A sample input file is shown below, and a description of the file follows. Each approach lane needs a connection to an egress lane with the signal group from the related SPaT message controlling the maneuver that connects the lanes.

### *Sample Input File*

```
<J2735.GID.blob>
  <Version>1</Version>
  <IntersectionID>1001</IntersectionID>
  <Elevation>false</Elevation>
  <Resolution>centimeter</Resolution>
  <Geometry>
    <ReferencePoint>
      <Latitude>38.954997</Latitude>
      <Longitude>-77.149386</Longitude>
    </ReferencePoint>
    <Approach>
      <Lane Number="2">
        <Type>Vehicle</Type>
        <Attributes>14</Attributes>
        <Width>305</Width>
        <Nodes>
          <Node Number="1">
            <Eastern>-1540</Eastern>
            <Northern>320</Northern>
            <Elevation>0</Elevation>
          </Node>
          <Node Number="2">
            <Eastern>-1020</Eastern>
            <Northern>500</Northern>
            <Elevation>0</Elevation>
          </Node>
        </Nodes>
        <Connections>
          <Connection>
            <LaneNumber>106</LaneNumber>
            <Maneuver>1</Maneuver>
            <SignalGroup>2</SignalGroup>
          </Connection>
      </Lane>
      <Lane Number="6">
        <Type>Vehicle</Type>
        <Attributes>2</Attributes>
        <Width>305</Width>
        <Nodes>
          <Node Number="1">
            <Eastern>1450</Eastern>
            <Northern>-300</Northern>
            <Elevation>0</Elevation>
          </Node>
          <Node Number="2">
            <Eastern>1550</Eastern>
            <Northern>-270</Northern>
            <Elevation>0</Elevation>
          </Node>
          <Node Number="3">
            <Eastern>1520</Eastern>
            <Northern>-170</Northern>
            <Elevation>0</Elevation>
          </Node>
        </Nodes>
        <Connections>
          <Connection>
```

```
            <LaneNumber>107</LaneNumber>
            <Maneuver>1</Maneuver>
            <SignalGroup>6</SignalGroup>
          </Connection>
        </Connections>
      </Lane>
    </Approach>
    <Egress>
      <Lane Number="106">
        <Type>Vehicle</Type>
        <Attributes>2</Attributes>
        <Width>305</Width>
        <Nodes>
          <Node Number="1">
            <Eastern>1450</Eastern>
            <Northern>-650</Northern>
            <Elevation>0</Elevation>
          </Node>
          <Node Number="2">
            <Eastern>1550</Eastern>
            <Northern>-270</Northern>
            <Elevation>0</Elevation>
          </Node>
        </Nodes>
      </Lane>
      <Lane Number="107">
        <Type>Vehicle</Type>
        <Attributes>2</Attributes>
        <Width>305</Width>
        <Nodes>
          <Node Number="1">
            <Eastern>-1420</Eastern>
            <Northern>690</Northern>
            <Elevation>0</Elevation>
          </Node>
          <Node Number="2">
            <Eastern>-1030</Eastern>
            <Northern>490</Northern>
            <Elevation>0</Elevation>
          </Node>
          <Node Number="3">
            <Eastern>-850</Eastern>
            <Northern>380</Northern>
            <Elevation>0</Elevation>
          </Node>
          <Node Number="4">
            <Eastern>-1100</Eastern>
            <Northern>530</Northern>
            <Elevation>0</Elevation>
          </Node>
        </Nodes>
      </Lane>
    </Egress>
  </Geometry>
</J2735.GID.blob>
```

### *J2735.GID.blob Node*

```
<J2735.GID.blob>
      <Version>1</Version>   ← Version of the curve file
      <IntersectionID>1001</IntersectionID>   ← Intersection ID (needs to match SPaT)
      <Elevation>false</Elevation>   ← Boolean indicating if elevation is included
      <Resolution>centimeter</Resolution>   ← Node and elevation offset resolution
      <Geometry>   ← Intersection geometry
      </Geometry>
</J2735.GID.blob>
```

### *Geometry Node*

```
<Geometry>
      <ReferencePoint>   ← Reference point location
            <Latitude>38.954997</Latitude>   ← Latitude in decimal degrees
            <Longitude>-77.149386</Longitude>   ← Longitude in decimal degrees
      </ReferencePoint>
      <Approach />   ← List of approach lanes
      <Egress />   ← List of egress lanes
</Geometry>
```

### *Approach and Egress Node*

Approach and Egress have the same format. Lanes in the egress node will not have the connections node.

```
<Approach>
      <Lane Number="2">
            <Type>Vehicle</Type>
            <Attributes>14</Attributes>
            <Width>305</Width>
            <Nodes />
</Approach>
```

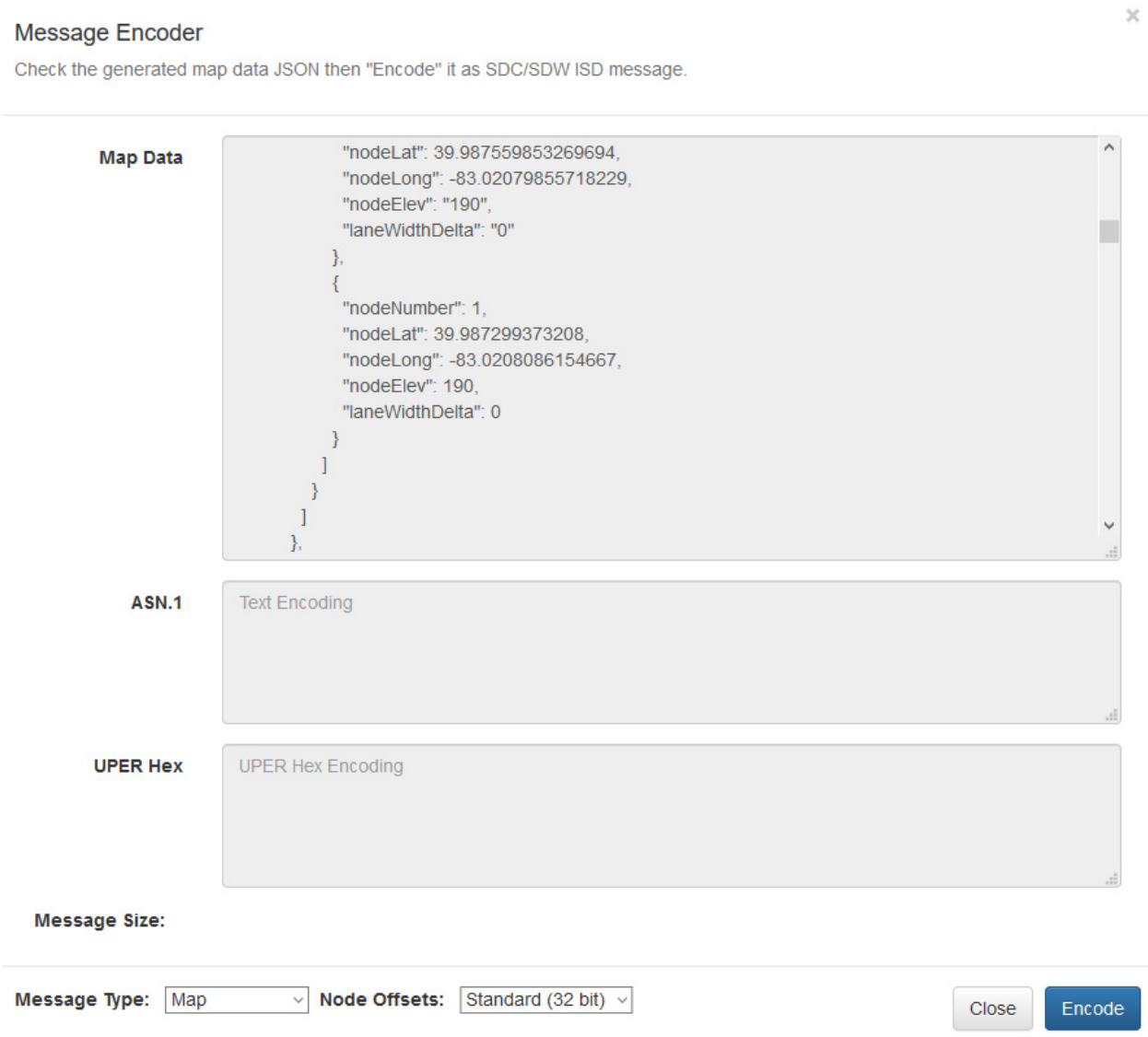
### *Nodes Node*

```
<Nodes>
      <Node>
            <Eastern>0</Eastern> ← Eastern offset in centimeters for node 1
            <Northern>0</Northern> ← Northern offset in centimeters for node 1
            <Elevation>0</Elevation> ← Elevation offset in centimeters for node 1
      </Node>
      <Connections>   ← List of connections (Approach Only)
```

```
            <Connection>
                  <LaneNumber>107</LaneNumber>   ← Connecting lane number
                  <Maneuver>1</Maneuver>   ← Connecting maneuver bitmapped int
                  <SignalGroup>6</SignalGroup>   ← Connecting signal group
            </Connection>
      </Connections>
<Nodes>
```

## MAP JSON Input

The MAP plugin can also use JSON input created by the USDOT ISD Message Creator. Use the tool to create a map following their instructions. Once the MAP has been created, select Tools -> Encoder to bring up the encoder dialog seeing in Figure 1. The JSON data is in the top text area titled Map Data. Copy everything from the Map data text box. Open your favorite text editor and paste the data into a new file and save it with the extension ".json".




*Source: Battelle, 2018*

**Figure 1. ISD Tool Message Encoder**

# Chapter 3. Testing

Once the system has been installed and configured, the system needs to be tested for basic functionality. It is important that each deploying agency define a test plan that meets their specific needs. Below we outline a sample testing plan that will verify that the deployed V2I Hub system is sending both SAE J2735-2016 SPaT and MAP. This testing plan utilizes the V2I Hub admin portal and a Cohda OBU using their packet sniffing software package.

## Application Functionality Testing

### SPaT TSC Communications

- Log into the Administration Portal.
- View the Message Activity for the SPaT plugin by expanding the plugin Messages section.
- The SPaT message count should be increasing.
- The SPaT message average interval should be 100 ms (1/10 second).

### MAP Generation

- Log into the Administration Portal.
- View the Message Activity for the MAP plugin by expanding the plugin Messages section.
- The MAP message count should be increasing.
- The MAP message average interval should be 1000 ms (1 second) unless changed by the configuration.

### DSRC Message Manager Transmission

- Log into the Administration Portal.
- View the Message Activity for the DSRC plugin by expanding the plugin Messages section.
- Verify that no errors exist by checking the Event Log tab and looking for DSRC Message Manager Error messages.

# System Functionality Testing

## DSRC Message Transmission

**Using Cohda radio as a packet sniffer.**

The message transmission can be checked using a Cohda radio and their version of Wireshark. A radio and access to their support site is needed for instructions. The article about using the radio as a sniffer is named “Channel Sniffer/Monitor” and is located under the MK5 Tools section.

Channel Sniffer/monitor

- View the Message Activity for the DSRC plugin.
- Verify that no errors exist.

Alternative sniffer can be used to check the transmission of the messages.

# Chapter 4. Maintenance and Support

## Scheduled Preventative Maintenance

It is recommended that each deploying agency develop and refine their own plan for monitoring and maintaining the hardware and software equipment comprising its V2I Hub deployments. Like a Wi-Fi router, this equipment occasionally experiences issues where the radio stops transmitting or communicating with the V2I Hub and needs to be reset. However, there is no way to predict when equipment may be reset or when it might fail. One approach to maintenance is to periodically monitor the V2I Hub equipment by using the V2I Hub Admin Portal. The V2I Hub Admin Portal can be remotely monitored for errors in the message log and anticipated message creation by the deployed plugins. Alternately, or in conjunction with use of the V2I Hub Admin Portal, agencies can use a packet sniffer or similar tool locally to view the transmitted DSRC messages at each deployed V2I Hub. The table below shows a sample maintenance plan that a deploying agency can follow to confirm operation and discover issues.

**Table 9. Sample Monitoring Plan for V2I Hub Equipment**

| Frequency | Checks |
|---|---|
| Weekly (remote) | • Using the V2I Hub Admin Portal, check the deployed plugins for the creation and communication of anticipated messages<br>  ○ i.e. MAP plugin should be creating MAP messages every second and the SPaT plugin should be creating SPaT messages every 100 ms<br>• Check the event logs for errors<br>• Check the DSRC message manager status for dropped messages |
| Monthly (locally where the equipment is installed) | • At the deployed location use a DSRC packet sniffer (or similar equipment) to capture 1-minute worth of DSRC data.<br>• Analyze the data looking for the transmission and rate of the messages and data expected<br>  ○ i.e. MAP data being transmitted every second and SPaT data being transmitted every 100 ms |

The deployment agency should refine the monitoring and maintenance plan based on the frequency and method(s) with which issues and errors are detected.

## System Update

In Chapter 2, two ways of obtaining the V2I Hub software for the deployed equipment were mentioned. It is important to know the installation method for the V2I Hub software to complete a system update. If the

V2I Hub was installed on your device when purchased from a vendor, the vendor will need contacted to acquire vendor processes and procedures for updating the V2I Hub. If the V2I Hub was obtained from OSADP and compiled and installed by following the directions in OSADP, the same process will need to be completed to update the main V2I Hub core software. If only a plugin needs updated, obtain the source code from OSADP, follow the OSDAP documentation to compile the plugin, and create the installation zip file. Additional instructions for updating a plugin are presented in the *V2I Hub Administration Portal User Guide* file upload section.

# Troubleshooting Guide

## SPaT Not Transmitting

### *SPaT Message Not Sending from SPaT Plugin*

Check the message activity for the SPaT plugin. The SPaT message should be increasing and have an average interval of 100ms.

*Troubleshooting*

- Check admin portal event logs for connectivity errors with the Traffic Signal Controller from SPaT plugin
    - Check the State variable “TSC Connection” on the Administration Portal.
    - The state should be “Connected”.
- Check network cables.
- Check IP Addresses.
- Check any firewall rules for the network that would prohibit connectivity.
- Check configuration for the SPaT plugin.

### *SPaT Message Not Received by OBU or Sniffer*

Check the plugin state for the DSRC Message Manager. The last transmission time for the SPaT message should be 100 ms ago.

*Troubleshooting*

- Check admin portal event logs for connectivity errors with the Traffic Signal Controller from DSRC Immediate Forward plugin
    - Check the State variables on the Administration Portal.
- Check network connectivity between the RSU and V2I Hub.
- Check the RSU to see if it is in an operational state. Refer to the RSU manufacturer documentation.
- Check the RSU for GPS.  Most RSUs won’t broadcast DSRC messages without GPS.
- Check the OBU / Sniffer for GPS. Most OBUs / Sniffers won’t receive DSRC messages without GPS.
- Check to see if the OBU / Sniffer is receiving other CV messages.

- Check both RSU and OBU hardware certificates and make sure they are valid.
- If all fails, reboot the RSU.

## MAP Not Transmitting

#### *MAP Message Not Sending from MAP Plugin*

Check the message activity for the MAP plugin. The MAP message should be increasing and have an average interval of 1000ms.

*Troubleshooting*

- Verify that the MAP input XML or JSON file in located in /var/www/plugins/MAP directory.
- Verify that the configuration for the MAP plugin is correct.
- Verify that the MAP input XML or JSON file has the correct permissions, group and owner
    - Group: www-data
    - Owner: www-data
    - Permissions: 644 (rw-r—r--).

#### *MAP Message Not Received by OBU or Sniffer*

Check the plugin state for the DSRC Message Manager. The last transmission time for the MAP message should be 1000 ms ago.

*Troubleshooting*

- Check admin portal event logs for connectivity errors with the Traffic Signal Controller from DSRC Immediate Forward plugin
    - Check the State variables on the Administration Portal.
- Check network connectivity between the RSU and V2I Hub.
- Check the RSU to see if it is in an operational state. Refer to the RSU manufacturer documentation.
- Check the RSU for GPS. Most RSUs won't broadcast DSRC messages without GPS.
- Check the OBU / Sniffer for GPS. Most OBUs / Sniffers won't receive DSRC messages without GPS.
- Check to see if the OBU / Sniffer is receiving other CV messages.
- Check both RSU and OBU hardware certificates and make sure they are valid.
- If all fails, reboot the RSU.

# Common Issues

When problems are detected through periodic monitoring, more detailed investigation and action may be needed to restore the V2I Hub equipment to normal operation. Table 10 below presents the most commonly experienced V2I Hub equipment issues experienced and steps to take for resolution.

**Table 10. Common Issues**

| Issue | Resolution |
|---|---|
| Roadside Unit (RSU) Not Broadcasting | • Determine if the RSU is transmitting by checking the size of the log files for growth.<br>• Check to see if the operational lights are functioning based on manufacturers specifications. If not, then reboot the RSU by unplugging the Power over Ethernet unit for a minute then plugging it back in.<br>• Check that the RSU has the appropriate immediate forward applications configured and running. |
| Can't connect to V2I Hub unit with Maintenance Computer | • Confirm the maintenance computer and V2I Hub are connected to the same network.<br>• Double check that the accessing computer has an IP address on the same local network.<br>• If it's a managed switch, confirm the ports used are on the same local network without communication restrictions.<br>• Some organizations use connection security software, such as Cisco AnyConnect, that may be blocking access to the V2I Hub unit. Disabling or using a maintenance computer without the software may be required. Alternatively, it may be possible to access the V2I Hub's Admin Web Portal by entering the IP address of the V2I Hub unit into the web browser of a computer on the same network. |
| V2 Hub unit can't communicate with the RSU | • Check RSU lights for functionality or error based on manufacturers user guide.<br>• Check RSU's GPS signal. Most RSUs won't operate without a GPS signal.<br>• Check that RSU is in operate mode, and not standby. Consult manufacturers user guide for details on how to change mode.<br>• Check network cabling. The Ethernet cable should be plugged into the same switch as the V2I Hub unit.<br>• Check the IP address of the RSU and confirm that it matches the RSU IP address in the DSRC Message Manager Plugin of the V2I Hub Admin Web Portal.<br>• If it's a managed switch, confirm the ports used are on the same local network without communication restrictions.<br>• Use an Ethernet cable tester to verify the Ethernet cable is working (e.g., not damaged during installation) and that the RJ45 connectors are installed correctly. |

| Issue | Resolution |
|---|---|
| Data not being sent out the RSU for specific plugins | • Check that the plugin in question is enabled via the V2I Hub Admin Web Portal.<br>• Verify that the plugin output count is incrementing in the V2I Hub Admin Web Portal.<br>• Check that the plugin is configured correctly via the V2I Hub Admin Web Portal.<br>• Check that the RSU is configured correctly.<br>• Check that the DSRC Message Manager plugin is enabled and configured correctly. |
| SPaT Plugin not generating SPaT message | • Check configuration of the SPaT plugin for the correct IP address and port information for the traffic signal controller.<br>• Check that traffic signal controller network connection is plugged into the same network switch as the V2I Hub unit's local network connection (LAN).<br>• Check that the traffic signal controller is configured to send the SPaT data to the V2I Hub. |
| RTCM Plugin not generating RTCM messages | • Check correct configuration parameters for IP address, port, username and password via the V2I Hub Administration Web Portal.<br>• Check internet connectivity on V2I Hub unit and that the Internet network cable is plugged into the WAN port. Checking internet connectivity can be done by pinning a known website, like espn.com.<br>• Check network configuration for firewall rules preventing the V2I Hub unit from connecting to the NTRIP network. This may require coordination with IT personnel managing the network. |
| Plugin fails to upload and install | • Check for the existence and the permissions of the upload directory under the Command Plugin.<br>• Verify the Command Plugin is running as the *plugin* user.<br>• This may be due to a malformed manifest file. Run manifest file through a JSON validator to verify it. |
| MAP Plugin not generating MAP message | • Check that the MAP plugin has the correct MAP files in its configuration.<br>• Check that that the MAP files in the configuration exist in the MAP plugin directory on the V2I Hub unit. |
| Unable to receive or interpret incoming messages | • Make sure that broadcasting and receiving devices are configured to send and receive messages according to the same standardized formats (i.e., SAE J2735-2016).<br>• Verify that the broadcasting device is sending messages signed with valid security certificates and that the receiving device is enabled to validate signed messages according to the SCMS.<br>• Verify that the broadcasting device has not been 'blacklisted' by the SCMS or that the security certificates haven't expired. |

# Appendix A. Acronyms

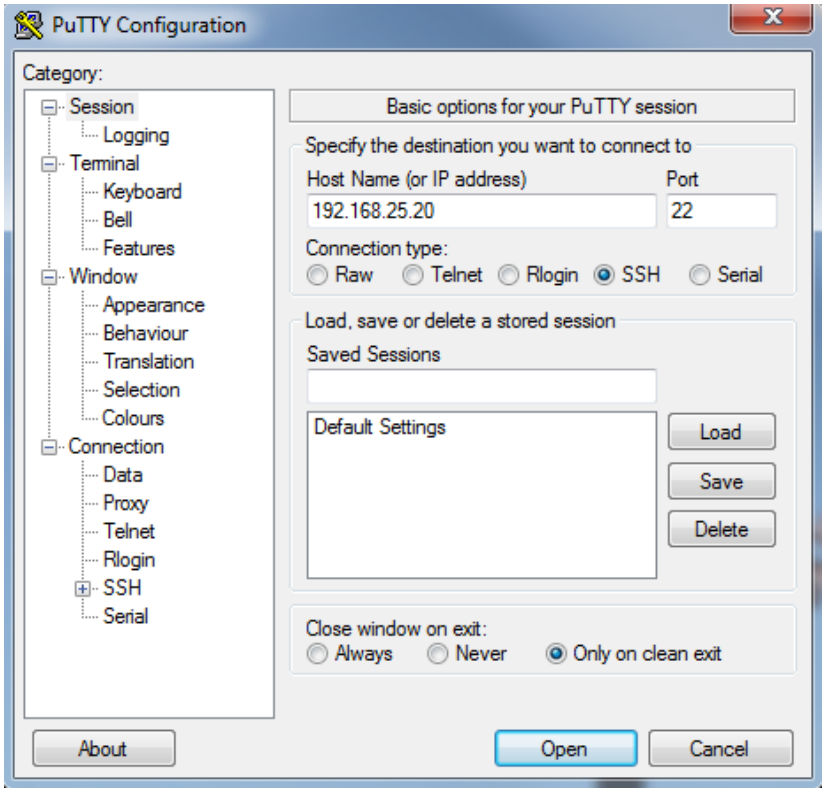
| | |
|---|---|
| **ASC** | Actuated Signal Controller |
| **CV** | Connected Vehicle |
| **BSM** | Basic Safety Message |
| **DSRC** | Dedicated Short-Range Communications |
| **FCC** | Federal Communications Commission |
| **GPS** | Global Positioning System |
| **ISD** | Intersection Situation Data |
| **IVP** | Integrated V2I Prototype |
| **JSON** | JavaScript Object Notation |
| **OBU** | On-Board Unit |
| **OSADP** | Open Source Application Development Portal |
| **PSID** | Provider Service Identifier |
| **RSU** | Roadside Unit |
| **RTCM** | Radio Technical Commission for Maritime Services |
| **SNMP** | Simple Network Management Protocol |
| **SPaT** | Signal, Phase, and Timing |
| **TIM** | Traffic Incident Message |
| **TSC** | Traffic Signal Controller |
| **UDP** | User Datagram Protocol |
| **U.S. DOT** | United States Department of Transportation |

| | |
|---|---|
| **V2I** | Vehicle-to-Infrastructure |
| **V2V** | Vehicle-to-Vehicle |

# Appendix B. Manual MAP Copy Instructions

Instructions below show how to use putty to ssh into the V2I Hub and copy over the MAP input XML or JSON file to the correct directory.

- The maintenance machine and V2I Hub will need to be on the same network. Change the IP address of your maintenance machine to 192.168.25.* (i.e., 192.168.25.82).
- Use WinSCP or some other secure file transfer (like scp) to copy the desired XML or JSON MAP input file to the V2I Hub. It is best to copy to the user's home directory (/home/ivp).
- Use an application like PuTTY, shown in Figure 2, to ssh into the V2I Hub.
    - Enter the IP Address (for our example, 192.168.25.20).
    - Select SSH for connection type and click open.




*Source Battelle, 2018*

**Figure 2. PuTTY Configuration for V2I Hub**

- Log in to the V2I Hub using the username and password provided by the owner of the V2I Hub.
- Once logged in change directory to /var/www/plugins/MAP (cd /var/www/plugins/MAP).
- Copy the files you copied to the user's root directory (sudo cp ~/MyNewMap.xml .).
- Change owner (sudo chown www-data MyNewMap.xml).
- Change group (sudo chgrp www-data MyNewMap.xml).
- Change permissions (sudo chmod 644 MynewMap.xml).
- Exit.

# Appendix C. V2I Hub Command Line Interface

While it is strongly recommended that the software configuration of V2I Hub be done using the V2I Hub Administration Portal. However, it should also be noted in this guide that V2I Hub also comes with an equivalent command line interface tool that can be used to perform all of the same tasks in case the portal is not available or the V2I Hub Command Plugin is not being used due to other concerns. The command is **tmxctl**, and the normal installation would place it under /usr/local/bin. This tool can be used to start, stop, enable and disable any Plugin just like on the web portal. Additionally, configurations can be set or modified using the tool, logging levels changed, or even new plugins installed. Below shows the command line options of **tmxctl**:

```
$ /usr/local/bin/tmxctl --help
Usage: bin/tmxctl:

  -h [ --help ]                         This help screen
  -l [ --level ] arg                    Log level, i.e. ERROR, WARNING, INFO,
                                        DEBUG, DEBUGn where n=1-4
  -o [ --output ] arg (=-)              Log output file.  Use - for standard
                                        output
  --plugin arg                          The plugin to control
  --list                                List the plugin information
  -e [ --enable ]                       Enable the plugin for automatic startup
  -d [ --disable ]                      Disable the plugin for automatic
                                        startup
  --start                               Start the plugin immediately
  --stop                                Stop a running plugin.  If enabled, it
                                        will restart automatically
  --status                              Return the current running status of
                                        the plugin
  --config                              Return the current configuration
                                        parameters
  --state                               Return the current state, i.e. the
                                        status values, of the plugin
  --set                                 Set a configuration value.  Must also
                                        set --key and --value.
  --reset                               Reset a configuration value to its
                                        default.  Must also set --key.
  --remove                              Remove a plugin from the database.
  --messages                            Show plugin message activity.
  --events                              Show event log.
  --clear-event-log                     Clear out event log in database.
  --system-config                       Return the current system configuration
                                        parameters.
  --set-system                          Set a system configuration value.  Must
                                        also set --key and --value.
  --user-info                           Display user information for a user.
                                        Must set --username.
  --all-users-info                      Display user information for all users.
```

```
  --user-add                            Add a TMX user. Must set --username,
                                        --password, and --access-level.
  --user-update                         Update a TMX users info. Must set
                                        --username, --password, and
                                        --access-level.
  --user-delete                         Delete a TMX user.
  -M [ --max-message-interval ] arg     Set the max message interval for the
                                        plugin
  -L [ --plugin-log-level ] arg         Set the log level for a running plugin
  -O [ --plugin-log-output ] arg        Redirect the logging of a running
                                        plugin to the specified file
  -a [ --args ] arg                     Set the command line arguments for the
                                        plugin
  -m [ --load-manifest ] arg            (Re-)load the plugin manifest to the
                                        database
  --plugin-install arg                  Decompress and install the specified
                                        plugin install file on this system.
  --plugin-remove arg                   Delete the specified plugin on this
                                        system.  No wildcards accepted.
  -j [ --json ]                         Format any output in JSON.  Default is
                                        false
  -x [ --xml ]                          Format any output in XML.  Default is
                                        false
  --no-pretty-print                     Do not pretty print the output.
                                        Default is false
  --key arg                             The parameter key name
  --value arg                           The parameter value
  --defaultValue arg                    The parameter default value for insert.
                                          Defaults to ''
  --description arg (=Added by tmxctl)  The parameter description for insert.
                                        Defaults to 'Added by tmxctl'
  -h [ --host ] arg (=127.0.0.1)        The MySQL DB host, if different than
                                        the default localhost
  -p [ --port ] arg (=3306)             The MySQL DB port, if different than
                                        the default 3306
  -d [ --plugin-directory ] arg (=/var/www/plugins)
                                        Directory to find the plugins
  --eventTime arg                       Event log entries greater than this
                                        time returned
  --username arg                        A TMX system user
  --password arg                        A TMX system users password
  --access-level arg                    A TMX system users access level. 1 =
                                        ReadOnly, 2 = ApplicationAdministrator,
                                        3 = SystemAdministrator
```

For example, a plugin archive can be installed with:

```
$ /usr/local/bin/tmxctl --plugin-install <plugin archive>.zip
```

Or, the MAP plugin can be restarted simply by stopping the current execution and letting the V2I Hub server start a fresh instance:

```
$ /usr/local/bin/tmxctl --stop MAP
```

Or, the destination address and port for the DSRC Message Manager could be changed with:

```
$ /usr/local/bin/tmxctl --set --key Destination_1 –value 192.168.55.43:1516 \
DSRCMessageManager
```

# Appendix D. References


1. U.S. DOT. (2009, September 25). IEEE 1609 - Family of Standards for Wireless Access in Vehicular Environments (WAVE). In *United States Department of Transportation*. Retrieved from https://www.standards.its.dot.gov/factsheets/factsheet/80

2. National Marine Electronics Association. (2008, November). NMEA 0183 Standard. In *National Marine Electronics Association*. Retrieved from https://www.nmea.org/content/nmea_standards/nmea_0183_v_410.asp

3. U.S. DOT. (2009, September 18). NTCIP 1202 - Object Definitions for Actuated Traffic Signal Controller Units. In *United States Department of Transportation*. Retrieved from https://www.standards.its.dot.gov/Factsheets/Factsheet/22

4. U.S. DOT. (2009, September 18). NTCIP 1203 - Object Definitions for Dynamic Message Signs (DMS). In *United States Department of Transportation*. Retrieved from https://www.standards.its.dot.gov/Factsheets/Factsheet/23

5. U.S. DOT. (2016, October 31). DSRC Roadside Unit (RSU) Specifications Document v4.1. In *United States Department of Transportation*. Retrieved from http://www.fdot.gov/traffic/Doc_Library/PDF/USDOT%20RSU%20Specification%204%201_Final_R1.pdf

6. Radio Technical Commission for Maritime Services. (n.d.). RTCM 10402.3 RTCM Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 2.3. In *Radio Technical Commission for Maritime Services*. Retrieved from http://www.rtcm.org/about.html

7. U.S. DOT. (2016, March). SAE J2735 - Dedicated Short-Range Communications (DSRC) Message Set Dictionary. In *United States Department of Transportation*. Retrieved from https://www.standards.its.dot.gov/Factsheets/Factsheet/71

8. U.S. DOT. (2016, March). SAE J2945/1 On-board Minimum Performance Requirements for V2V Safety Communications. In *United States Department of Transportation*. Retrieved from https://www.standards.its.dot.gov/Standard/531

9. Battelle. (2018, July). V2I Hub Administration Portal User Guide

10. Battelle. (2018, July). V2I Hub Deployment Guide

U.S. Department of Transportation
ITS Joint Program Office – HOIT
1200 New Jersey Avenue, SE
Washington, DC 20590

Toll-Free “Help Line” 866-367-7487

www.its.dot.gov

FHWA-JPO-18-645

U.S. Department of Transportation