V2X Hub

Plugins

Table of Contents

List of Tables

List of Figures

No table of figures entries found.

Executive Summary

This document outlines information about configuration of various V2X Hub plugins that could help transportation agencies to deploy and possibly customize V2X Hub.. V2X Hub contains several plugins that are listed in this document. The chapters below give descriptions of the plugins, along with information about the configuration values and any external configuration files that are needed to execute the plugins. This document is a part of the V2X Hub series of documents originally produced by the V2I Reference Implementation project. The rest of the documents are listed below. It is suggested to start with the V2X Hub Guidebook.

- V2X Hub Guidebook
- V2X Hub Plugin Programming Guide
- V2X Hub Administration Portal User Guide
- V2X Hub Deployment Guide
- V2X Hub Software Configuration Guide

V2X Hub*V2X Hub*. This document explains the basic core functionality and plugin architecture. The V2X Hub communicates with many external devices using several different protocols. These protocols are documented in the *V2X Hub Interface Control Document* (ICD) (Battelle, 2016). Please consult that document to better understand the communication protocols used by the different V2X Hub plugins.

Chapter 1. Dedicated Short Range Communication Message Manager

Description

The Dedicated Short-Range Communication (DSRC) Message Manager, known as the DSRC Message Manager, is a plugin which sends data to a DSRC 5.9 GHz Roadside Unit (RSU) compliant with the latest RSU 4.1 specification. The DSRC Message Manager is a plugin in the V2X Hub who monitors all messages in the hub, and packages those flagged for DSRC transmission. The message sent by the DSRC Message Manager to the RSU is transmitted using the immediate forward protocol described in Appendix B, which sends a message to the RSU over a UDP network connection. The data is then packaged and signed by the RSU and transmitted for other CV DSRC radios to use.

Configuration

The DSRC Message Manager can be configured to send messages to four different destinations. A destination is defined by a distinct IP address and port pair. This means that the DSRC Message Manager can support sending messages to four different RSUs if either the messages are the same or the RSU allows all traffic on one port. Some RSUs will only allow a single Provider Service Identifer (PSID) per immediate forwarded application. Examples of this include the Arada and Lear RSUs. These RSUs need an instance of their immediate forward application per PSID. RSUs that require multiple immediate forward applications to transmit (i.e. an application for PSID 0x8002 and a different application for PSID 0x003) will require configuration of multiple destinations to send all requested messages. The DSRC Message Manager can be configured to send messages to 4 different destinations. Table 1 below shows the configuration variables, description, default values for the DSRC Message Manager V2X Hub plugin.

Table 1. DSRC Message Manager Configuration Values

Key	Default Value	Description
Messages_Destination_1	{ "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 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	192.168.25.10:4589	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 of False value indicating whether the RSU should sign the messages being transmitted.

The message destination configuration values take a Java Script Object Notation (JSON) formatted string as their input. That JSON string contains the information needed to format and send the appropriate immediate forward message to the paired destination. Messages_Destination_1 contains the messages formatting information for Destionation_1 and so forth. The JSON string is described below:

- TmxType – internal message type used by the V2X Hub messages router
  - SPAT-P, MAP-P, TIM, BSM, RTCM are the standard J2735 message types used
    - SPAT-P and MAP-P specify USER encoding from the ASN.1. The transition to J2735 2015 allowed for UPER and DER encoded SPAT and MAP. All other messages were DER encoded.
- SendType – the message type string used to populate the immediate forward message
  - SPAT, MAP, TIM, BSM, RTCM are the standard J2735 message types used

- PSID – the PSID value used to populate the immediate forward message

Chapter 2. Map

Description

The Map plugin creates J2735 MAP messages in J2735 2016 format. Inputs into the plugin can be created using the U.S. DOTs ISD Message Creator. . . V2X Hub. The ISD Message Creator[1] is a MAP and SPAT creation tool for intersections in the USDOT Connected Vehicle Tools Library[2] . This online tool aids in the creation of J2735 2016 MAP files, which can be used as inputs for the V2X Hub MAP plugin. The input needed by the map plugin can be created by using the export tool, setting the Message Type dropdown to Map, and copying the 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.

Configuration

The Map plugin can be configured to send different Map files based on the action value received by the SPAT plugin. The frequency to which the Map can transmit its output can be configured by the Frequency configuration value, in milliseconds. The default is to send Map data at 1 second intervals. Table 2 shows the configuration values for the Map plugin.

Table 2. Map Configuration Values

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

The Map plugins uses a json value for the configuration key MAP_files which contains the map file to load associated with an action id. The action id is associated to an action plan in the traffic signal controller (TSC). The SPAT plugin will transmit the current action plan that it retrieves from the TSC to the V2X Hub, where the MAP plugin will receive it. The map plugin will load the appropriate map based on the action id. If an action id is supplied by the SPAT plugin doesn't have a corresponding item in the

[1] https://webapp.connectedvcs.com/isd/

[2] https://webapp.connectedvcs.com/

MAP_Files list, the map file associated with the first action 0 will be loaded. The description for the MAP_Files json is below.

- Action – the action id paired with the map file. The action id is associated to an action plan in the TSC. When this action id is active, the corresponding map file will be loaded.
- FilePath – Relative file name for the map file. This file will need to be located in the map plugin directory. (the default MAP plugin directory is /var/www/plugins/MAP/)

Chapter 3. Signal Phase and Timing

Description

The Signal Phase and Timing (SPAT) plugin communicates with the traffic signal controller (TSC) and creates J2735 SPAT messages conforming to the J2735 2016 format. The communication with the traffic signal controller is done using National Transportation Communications for ITS Protocol (NTCIP) communication over UDP receiving a bit encoded byte array. . The SPAT plugin only sends the necessary message to the TSC to enable and disable the sending of the UDP bit encoded stream. This plugin's main function is to receive the UDP bit encoded stream from the TSC and create the J2735 SPAT message.

Configuration

The SPAT plugin can be configured communicate and receive signal phase and timing information from a single traffic signal controller. Table 3 shows the configuration values for the SPAT plugin.

Table 3. SPAT Configuration Values

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 SignalGroups and phases.
Local_IP	192.168.25.20	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	192.168.25.50	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 1202 v2 communication.

The SignalGroupMapping contains a list of active phases from the signal controller and their corresponding signal groups in the J2735 SPAT message along with their phase type of either vehicle or pedestrian. Each intersection can be given a friendly name to include in the description of the SPAT message. The Intersection_Name configuration value contains the information that will be populated into that description. The Local_IP configuration value contains the local IP address of the V2X Hub on the same network as the traffic signal controller. This is the same IP address that you would use to configure the traffic signal controller as the endpoint to send the UDP SPAT stream. Paired with the Local_IP is the Local_UDP_Port. This port on the local V2X Hub to receive the UDP SPAT stream. For most of the traffic signal controllers, this is either a non-configurable value to set to 6053, or a configurable value defaulted to 6053. This configuration value should not need to be modified for the SPAT plugin to operate correctly. The last two configuration values, TSC_IP and TSC_Remote_SNMP_Port, are used by the V2X Hub to

send SNMP requests. The TSC_IP address contains the IP address of the TSC. The port for which the TSC listens for SNMP commands is configured by the TSC_Remote_SNMP_Port. Again, the default for the TSC_Remote_SNMP_Port shouldn't need to be modified.

Chapter 4. RTCM

Description

The Radio Technical Commission for Maritime Services (RTCM) is an international standards organization that has created the RTCM 10402.3 RTCM Recommended Standards for Differential GNSS Service v2.3 and RTCM 10403.3 Differential GNSS Service 3. The RTCM plugin uses the Networked Transport of RTCM via Internet Protocol (NTRIP) to receive the RTCM real time correction information, create the corresponding RTCM J2735 2016 message, and transmit it into the V2X Hub. Currently this plugin supports RTCM v 2.3.

Configuration

The RTCM plugin can be configured to create J2735 2016 RTCM messages from one NTRIP network. The network information is configured by the Endpoint IP and Endpoint Port. The network authentication information is supplied by the Username and Password configuration values. The NTRIP mountpoint to receive the RTCM stream is configured using the Mountpoint configuration values. Table 4 contains the configuration values for the RTCM plugin. The default values, not including the username and password, are used to retrieve RTCM 2.3 data from the state of Ohio's NTRIP network. However, the user is advised to pick a specific local IP address.

Table 4. RTCM Configuration Values

Key	Default Value	Description
Endpoint IP	156.63.133.118	NTRIP caster endpoint IP address
MAP_ Endpoint Port	2101	NTRIP caster endpoint port
Username	username	NTRIP caster authentication username
Password	password	NTRIP caster authentication password
Mountpoint	ODOT_RTCM23	NTRIP caster mountpoint

Chapter 5. Dynamic Message Sign

Description

The Dynamic Message Sign (DMS) plugin communicates with a DMS using NTCIP 1203 v3. The plugin receives requests for messages to be displayed from other plugins in the V2X Hub, and sends the appropriate message string from the configuration to the DMS for display. The plugin configures the DMS with 4 messages that can be called up using their index. This reduces the delay in displaying the message. The 4 messages can be configured using the configuration values. When the messages are changed, the updated message list is sent to the DMS. A limitation of the plugin is that if a clear display is needed, the plugin must use one of the 4 messages, reducing the number of messages to three.

Configuration

The DMS plugin can be configured to display four messages on a single DMS. The DMS must be connected to the same network as the V2X Hub. The DMS IP Address configuration value contains the IP address of the DMS on the network. The DMS Port is the port used by the DMS for receiving the NTCIP communications. The Message 01, Message 02, Message 03, and Message 04 configuration values contain the string used to configure the sign for message display. The default values showcase messages that worked when testing with the ADDCO DH250 sign. The “[nl]” key word in the message tells the messages to start on a new line. The configuration in Table 5 shows that the DMS is configured to display a blank page when message with id 01 is sent, 25 MPH when message id 02 is sent, Slow Down when message id 03 is sent, and Curve Ahead when message id 04 is sent. Configuration values Enable Sign Simulator, Enable DMS, Sign Sim IP Address, and Sign Sim Port are not shown or described because they are deprecated.

Table 5. DMS Configuration Values

Key	Default Value	Description
DMS IP Address	192.168.25.30	The IP address of the NTCIP Dynamic Message Sign.
DMS Port	9090	The port of the NTCIP Dynamic Message Sign.
Force Message ID	-1	Immediately activates the message ID specified, then resets back to -1.
Message 01		The text to display on the sign for ID 01 with any formatting (see NTCIP 1203).
Message 02	[jl3][pt15o0]25[nl]MPH	The text to display on the sign for ID 02 with any formatting (see NTCIP 1203).
Message 03	[jl3][pt15o0]SLOW[nl]DOWN	The text to display on the sign for ID 03 with any formatting (see NTCIP 1203).
Message 04	[jl3][pt15o0]CURVE[nl]AHEAD	The text to display on the sign for ID 04 with any formatting (see NTCIP 1203).

Chapter 6. Curve Speed Warning

Description

The Curve Speed Warning (CSW) plugin transmits information pertaining to an approaching sharp curve, monitors basic safety messages (BSM), and requests display of messages on a DMS when conditions are met. The CSW application monitors and transmits geometry leading up to the entrance of the curve using four zones. The zones are transmitted with the recommended speed for each zone using the J2735 Traveler Information Message (TIM) messages in J2735 2016 format. The zones for the CSW message lead up to the entrance of the curve and are numbered from 1 to 4, with 1 being the closest zone to the entrance of the curve. This means a vehicle driving towards the curve will enter zone 4 first, then 3 and so on until the vehicle enters the curve. The zones are directional, meaning that the vehicle traveling towards the curve will interact with the zones, but the vehicle traveling away from the curve will not. To give CSW alerts in both directions, zones will need to be created for each direction of travel. The four zones are used by an in-vehicle application which can give different alerts or warnings to the driver depending on zone and speed. When the CSW plugin detects that a BSM is going over the recommended speed in zones 3 through 1, the CSW plugin will sending a DMS request to display “Slow Down” on the DMS.

Configuration

Configuration for the CSW plugin is done using the following configuration values found in Table 6. The frequency of TIM transmission by the CSW plugin can be changed using the Frequency configuration values. This value is defaulted to 1000 milliseconds, or 1 second. The TIM geometry zones are configured using the MapFile configuration value. This value points to an xml file on the V2X Hub in the CSW plugin’s home directory that contains the configuration of the zones in xml format. This xml format is like that of the MAP plugin. The default location for the xml file is /var/www/plugings/CswPlugin. More information of the structure of the CSW input xml format can be found in Appendix C.

Table 6. CSW Configuration Values

Key	Default Value	Description
Frequency	1000	The frequency to send the TIM in milliseconds.
MapFile	IVP_TF_CSW.xml	The CSW xml input file.
Snap Interval	300	The interval in milliseconds to keep a vehicle within a zone before allowing it to transition out of all zones.
Vehicle Timeout	2000	Timeout in milliseconds when a vehicle is removed from all zones if a BSM has not been received.

Snap Interval and Vehicle Timeout config are used by the vehicle monitoring algorithm portion of the CSW plugin. The CSW will timeout a vehicle that it has not received an update from after the value in Vehicle Timeout, which is defaulted to 2 seconds. The snap interval is used to reduce jitter in the GPS position, and requires that 300 milliseconds elapse before transitioning a vehicle out of a zone.

Chapter 7. Basic Safety Message Receiver

Description

Most DSRC RSUs have an application that forwards certain messages from the radio to a configured device. The stored data is sent in J2735 2016 format, and the most common data type sent is the BSM. The BSM Receiver was written to take the forwarded BSM information in J2735 2016 format from an RSU and send it into the V2X Hub system for other applications, like the CSW plugin, to use in its processing. The BSM Receiver can work with any device that forwards BSMs in J2735 format over UDP including simulation software or future communication devices which support the same standard.

Configuration

The BSM Receiver can be configured to receive BSM message from one RSU. The configuration is simple and requires only the IP address for the RSU to be set for the IP configuration values, along with the Port to open on the V2X Hub. In most cases, you will need to supply the same port on the RSU as set for Port for the BSM Receiver. The RouteDSRC flag is used for testing in the lab and allows BSMs generated by other plugins to be sent out of the RSU. This configuration values should not be changed and kept at false for most tests and for all deployments.

Table 7. CSW Configuration Values

Key	Default Value	Description
IP	192.168.55.43	IP address for the RSU.
Port	26789	Port for the RSU network connection.
RouteDSRC	false	Set flag to true for the message to be sent out the DSRC radio by the DSRC Message Manager. This flag would only be used in testing or simulation. For example, having an RSU broadcast BSMs from simulation software.

Chapter 8. Operational Data Environment

Description

The Operational Data Environment (ODE) is a scalable data acquisition and distribution solution created by Intelligent Transportation Systems - Joint Program Office (ITS JPO). The ODE plugin allows transmission of BSMs received in the V2X Hub to the deployed ODE solution. In a deployed V2X Hub system, the BSM would be received by the RSU and forwarded to the V2X Hub. Once in the V2X Hub, the BSM would be received by the ODE plugin and then sent to the ODE using the set configuration. The ODE solution can also create TIM information based on the data it receives. The ODE plugin will query the ODE for TIM messages for the deployed V2X Hub's location, using the latitude and longitude obtained by GPS or set in configuration during deployment. Any TIM messages retrieved by the ODE plugin will be packaged and sent through the V2X Hub and ultimately out the RSU.

Configuration

The ODE Plugin Receiver can be configured to send BSM messages and receive TIM messages from one ODE instance using the configuration in Table 8. The ODEIP contains the configuration value for the IP address for the deployed ODE, and the ODEPort is the port used for transmission

Table 8. ODE Configuration Values

Key	Default Value	Description
ODEIP	192.168.55.43	IP address for the ODE network connection.
ODEPort	26789	Port for the ODE network connection.

Chapter 9. Pedestrian Safety Message

Description

Pedestrian Safety Message plugin provides a guideline to connect a pedestrian's status with the intelligent transportation system infrastructure. A pedestrian, on the sidewalk and about to cross the roadway, is able to send pedestrian calls to the roadway infrastructures such as traffic signals, RSU, controllers and TMC. Using this plugin, a pedestrian message is generated that is routed to the V2X Hub through open networked web service API. It is assumed that the pedestrian device, cell phones or handheld device is capable of interfacing with open webservice available on the compute node runnin a V2X Hub. Using the restful PUSH API the message is sent to the V2X Hub. This message is handled by V2X hub which re routes the message to be broadcasted through the nearby RSU. Before the routing is done, V2X Hub will encode the pedestrian call into J2735 Personal Safety Message (PSM) ASN.1 format. This message is broadcasted through the RSU and is used to alert nearby drivers on vehicles equipped with on board units. The plugin uses DSRCImmediateForward plugin for re broadcasting through the RSU. The pedestrian message originating from the near by pedestrian traffic can be used to identify if the pedestrians are inside a geo-fenced area surrounding the intersection (or RSU location), hence making it capable for RSU's to control traffic flow and allow safe crossing signal to the pedestrian.

Configuration

The pedestrian safety message plugin is dependent on successful integration with web service. The web service currently available with V2X Hub is Qt engine based and developed using openapi infrastructure. The plugin needs webservice to be up and running before any requests are delivered. The port number and IP address must be defined before the webservice is run so that the incoming messages are properly routed/serviced. The plugin needs some configuration parameters as shown below:

Key	Default Value	Description
Frequency	1000	The frequency to send the PSM in milliseconds.
Instance	0	The instance of the Pedestrian Plugin
WebServiceIP	127.0.0.1	The IP address associated with interface where the webservice is listening for requests. This should be set up such that anyone can send valid requests to the webservice, hence should be outside facing IP address on the V2X Hub machine
WebServicePort	9000	The port where webservice is bound to. The incoming requests must be associated with this port or else they wont be serviced.

Chapter 10. Preemption Plugin

Description

The preemption plugin provides support for V2X Hub to allow preempting the traffic signal timing plan for emergency vehicles. An emergency vehicle traveling with a strict response time needs an unblocked flow through any intersections it comes across. This means the traffic controller should be able to detect approaching emergency vehicle and be able to flush the signal phases such that by the time the vehicle arrives at the intersection, the path of flow of the vehicle is green. This is a very important feature in terms of emergency situations and responses. V2X Hub incorporates a preemption plugin that is designed to look for extended basic safety messages and in particular, look for any emergency vehicles with sirens and lights ON. The state of sirens and lights is embedded into spcial vehicle extension of J2735 message set [7]. Once the preemption plugin receives a BSM with sirens and lights enabled, it checks if the vehicle is inside a geo fence defined as an active boundary for triggering preemption. If the vehicle is not inside the geofence than no preemption request is issued to the signal controller. If the vehicle is inside the boundary and no preemption request has been issued successfully for the vehicle, then the preemption plugin would request preemption on the exact phase of the traffic signal set where geofencing was triggered. Once the vehicle leaves the geofence a preemption cancel request is issued as well and the traffic controller returns to the normal phase operation.

Configuration

Key	Default Value	Description
BasePreemptionOid	.1.3.6.1.4.1.1206.4.2.1.6.3.1.2.	The BasePreemptionOid of Preemption Plugin
Instance	0	The instance of the Preemption Plugin
ipwithport	192.168.55.49:6053	The IP address with port number where the signal controller is listening on for any request coming from Preemption Plugin
map_path	./geofence.json	Location of the map file that contains geo location for geo fence
snmp_community	public	Snmp_community for Preemption Plugin

Chapter 11. System Performance Measures

Description

The proliferation of connected vehicles provide a new opportunity for traffic engineers to collect performance data about transportation system elements, especially intersections. Historically, data collection at intersections has been focused on short intervals of a few hours, with a focus on a single measure, such as queue volume at regular intervals, or spot speed. Trajectory information from connected vehicles allows precision data about multiple measures (such as signal state and vehicle location) over extended periods of time. However, the market penetration rate for connected vehicles is still low and agencies may want to wait to adopt such performance measure calculation when proliferation supports reporting within acceptable error.

Configuration

The System Performance measures plugin pulls information offline from the BSM Logger Plugin. This plugin runs as a python script, and takes as input:

1. The Location of the BSM Logger Plugin outputs.
2. The timestamp of the moment in question.
3. Default values for shockwave speeds and stopping speeds where none can be calculated.

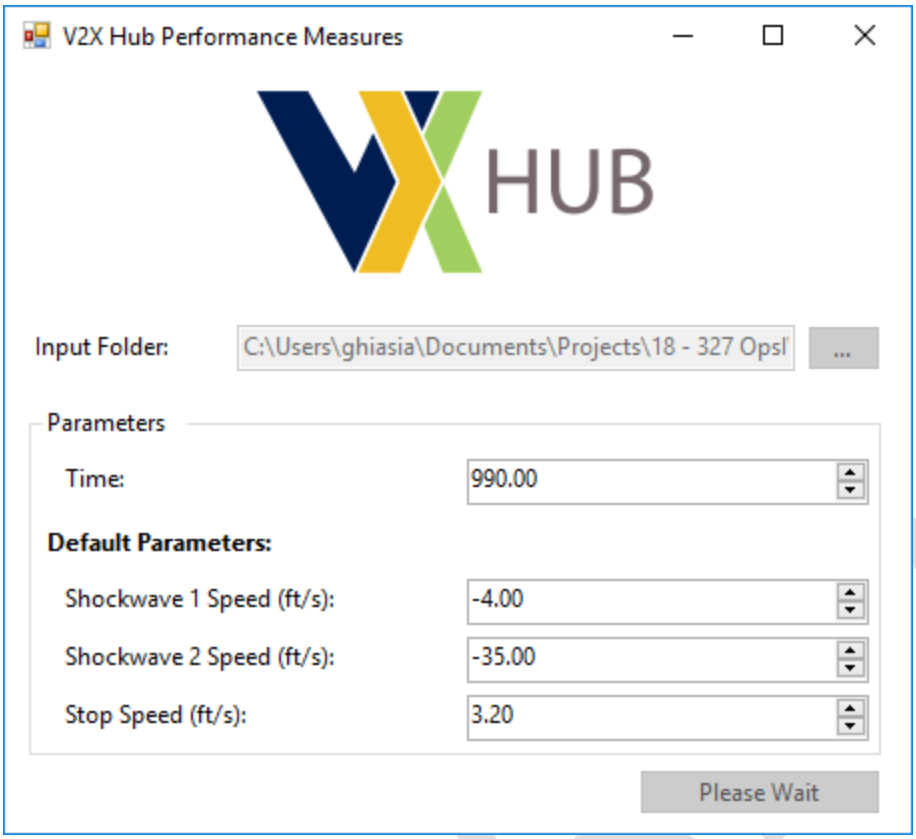
- 

Figure 1. Performance Measures Configuration Screen

Chapter 12. TIM Plugin

Description

Traveller Information Message (TIM) is primarily used for sending out roadway incident and information to surrounding vehicles. These messages include but not limited to lane closures, work zones, speed limit changes etc. One of the major roadways sections where TIM becomes more prominent are work zones. Workzones are mostly pre-mediated, so the road management is planned of upcoming work zone. This includes alerting the local area broadcast unit such as a RSU connected to a TMC via a dedicated connection and dispersing the work zone information to nearby vehicles. The work zone information can be identified by TIM. A TIM message contains the lane status information, speed limts etc. Hence these messages can be used for alerting a vehicle driving around an active work zone. For the purpose of demonstrating the TIM plugin using V2X Hub, the interface between TMC and the V2X Hub is also identified. The interface supports web api calls for posting an updated TIM message to the V2X Hub so that an active work zone with frequent changes in lane status can be updated as reliably as possible. The web end points for V2X Hub connects any web client running in TMC to send the updates as well as the timings for when to broadcast the TIM messages. Appendix F contains a sample of the xml TIM update POST message.

The TIM plugin has a manifest file setting that can also be used if there is no connection to a TMC. The initial setting on the manifest file includes start/stop date and time. However, these dates are no longer used. Instead, the startTime and duration peirod in the TIM XML are used to determine when the plugin will allow or block broadcast of a TIM messags preset as an xml document in a predefined location. The startTime and duration period can be specified to a resolution of a minute. The startTime expresse the number of elapsed minutes of the current year in the time system being used (typically UTC time). updates through web service overwrite the existing settings and therefore the broadcasting opration can be managed by remote TMC server.

Configuration

Key	Default Value	Description
Frequency	1000	The frequency to send the TIM in milliseconds.
MapFile	“filename.xml ”	Location of a pre-existing TIM message file (xml)
Start_Broadcast_Date	01-01-2019	The Start Broadcast Date for the TIM message in the (mm-dd-YYYY) format. **(No longer used)**
Stop_Broadcast_Date	12-31-2020	The Stop Broadcast Date for the TIM message in the (mm-dd-YYYY) format. **(No longer used)**
Start_Broadcast_Time	06:00:00	The Start Broadcast Time for the TIM message in the (HH:MM:SS) format. **(No longer used)**
Stop_Broadcast_Time	21:00:00	The Start Broadcast Time for the TIM message in the (HH:MM:SS) format. **(No longer used)**
WebServiceIP	127.0.0.1	IP address at which the web service exists
WebServicePort	9999	Port at which Web service exists

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
JSON	Java Script Object Notation
IVP	Integrated V2I Prototype
NTCIP	Networked Transport of RTCM via Internet Protocol
OBU	On-Board Unit
PSID	Provider Service Identifier
PSM	Pedestrian Safety Message
RSU	Roadside Unit
RTCM	Radio Technical Commission for Maritime Services
SPAT	Signal, Phase, and Timing
TIM	Traffic Information Message
TSC	Traffic Signal Controller
U.S. DOT	United States Department of Transportation
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle

XML	Extensible Markup Language

Appendix B. Active File Format

The active file format listed below is from the *DSRC Roadside Unit (RSU) Specifications Document v4.1* Appendix C. This has been included in its entirety as a reference.

```
# Message File Format

# Modified Date: 04/10/2014

# Version: 0.7

Version=0.7

# Message Dispatch Items

#

# All line beginning with # shall be removed in file sent to radio

#

# Message Type

# Values: SPAT, MAP, TIM, (other message types)

Type=<Type>

#

# Message PSID as a 2 Byte Hex value (e.g. 0x8003)

PSID=<PSID>

#

# Message Priority in the range of 0 (lowest) through 7

Priority=<priority>

#

# Transmission Channel Mode

# Allowed values: CONT, ALT

TxMode=<txmode>
```

Allowed values: 172, CCH, SCH(note: “CCH” refers to DSRC Channel 178 and SCH refers to the

#operator configured DSRC Service Channel)

TxChannel=<channel>

#

Transmission Broadcast Interval in Seconds

Allowed values: 0 for Immediate-Forwarding, 1 to 5 for Store-and-Repeat

TxInterval=<txinterval>

#

Message Delivery (broadcast) start time (UTC date and time) in the form:

#"mm/dd/yyyy, hh:mm”

Leave value blank if Immediate Forward mode

DeliveryStart=<mm/dd/yyyy, hh:mm>

#

Message Delivery (broadcast) stop time (UTC date and time) in the form:

#"mm/dd/yyyy, hh:mm”

Leave value blank if Immediate Forward mode

DeliveryStop=<mm/dd/yyyy, hh:mm>

#

Message Signature/Encryption

Signature=<True\False>

Encryption=<True\False>

#

Message Payload (encoded according to J2735 or other definition)

Payload=<DSRC message payload>

Appendix C. CSW XML File Format

The Curve Speed Warning (CSW) plugin uses an XML input file to create the geometry sent in the SAE J2735-2016 Traveler Information Message (TIM) for the curve. The input file contains the speed of the curve, the four (4) zones with their reference points, offsets to create the center line for the zones, and the zone width. Currently, the CSW plugin supports zones containing two (2) points, which create a straight line. The zones are referenced as Zone 1 through Zone 4, with Zone 1 being the closest to the entrance of the curve and Zone 4 the furthest. In the file, Zone 1 is the first region in the region list. As a vehicle approaches the curve, the vehicle will enter Zone 4, Zone 3, Zone 2, and finally, Zone 1. A sample XML file is included along with a description of each node.

Sample Input File

```
<?xml version="1.0" encoding="UTF-8"?>
<Curve xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
  <Version>1</Version>
  <CurveID>125</CurveID>
  <SpeedLimit>8.491</SpeedLimit>
  <Approaches>
    <Approach>
      <Regions>
        <Region>
          <ReferencePoint>
            <Latitude>38.9556187319798</Latitude>
            <Longitude>-77.1504409878302</Longitude>
            <Elevation>0</Elevation>
          </ReferencePoint>
          <Width>335</Width>
          <DirectionOfUse>Forward</DirectionOfUse>
          <Nodes>
            <Node>
              <Eastern>0</Eastern>
              <Northern>0</Northern>
              <Elevation>0</Elevation>
            </Node>
            <Node>
              <Eastern>141</Eastern>
              <Northern>-206</Northern>
              <Elevation>0</Elevation>
            </Node>
          </Nodes>
        </Region>
        <Region>
          <ReferencePoint>
            <Latitude>38.9554328674441</Latitude>
            <Longitude>-77.150277082612</Longitude>
            <Elevation>0</Elevation>
          </ReferencePoint>
          <Width>335</Width>
          <DirectionOfUse>Forward</DirectionOfUse>
          <Nodes>
```

```
            <Node>
              <Eastern>0</Eastern>
              <Northern>0</Northern>
              <Elevation>0</Elevation>
            </Node>
            <Node>
              <Eastern>173</Eastern>
              <Northern>-181</Northern>
              <Elevation>0</Elevation>
            </Node>
          </Nodes>
        </Region>
        <Region>
          <ReferencePoint>
            <Latitude>38.9552698871225</Latitude>
            <Longitude>-77.150076690332</Longitude>
            <Elevation>0</Elevation>
          </ReferencePoint>
          <Width>335</Width>
          <DirectionOfUse>Forward</DirectionOfUse>
          <Nodes>
            <Node>
              <Eastern>0</Eastern>
              <Northern>0</Northern>
              <Elevation>0</Elevation>
            </Node>
            <Node>
              <Eastern>256</Eastern>
              <Northern>-155</Northern>
              <Elevation>0</Elevation>
            </Node>
          </Nodes>
        </Region>
        <Region>
          <ReferencePoint>
            <Latitude>38.9551299660347</Latitude>
            <Longitude>-77.1497799675418</Longitude>
            <Elevation>0</Elevation>
          </ReferencePoint>
          <Width>335</Width>
          <DirectionOfUse>Forward</DirectionOfUse>
          <Nodes>
            <Node>
              <Eastern>0</Eastern>
              <Northern>0</Northern>
              <Elevation>0</Elevation>
            </Node>
            <Node>
              <Eastern>382</Eastern>
              <Northern>-138</Northern>
              <Elevation>0</Elevation>
            </Node>
          </Nodes>
        </Region>
      </Regions>
    </Approach>
  </Approaches>
</Curve>
```

Curve Node

```
<?xml version="1.0" encoding="utf-8"?>
<Curve xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <Version>1</Version> ← Version of the curve file
      <CurveID>125</CurveID> ← Curve ID
      <SpeedLimit>8.491</SpeedLimit> ← Speed Limit of Curve in m/s
      <Approaches> </Approaches> ← List of Approach Nodes
</Curve>
```

Approach Node

```
<Approach>
      <Regions> </Regions> ← List of Region Nodes
</Approach>
```

Region Node

Each region of the curve XML is information for one of the zones. Each Region should be formatted like the one below, and contains a reference point, width, and node list. Each node contains offset in decimeters from the reference point to create the center line for the region (or zone).

```
<Region>
      <ReferencePoint> ← Used in creating the center line for the region with the nodes
            <Latitude>38.9556187319798</Latitude> ← Latitude in decimal degrees
            <Longitude>-77.1504409878302</Longitude> ← Longitude in decimal
degrees
            <Elevation>0</Elevation> ← Elevation in meters (optional, set to 0 if not used)
      </ReferencePoint>
      <Width>335</Width> ← Width in decimeters (ex 335 = 3.35 meters)
      <DirectionOfUse>Forward</DirectionOfUse> ← Direction of use should always be
forward
      <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>
            <Node>
                  <Eastern>141</Eastern> ← Eastern offset in centimeters for node 2
                  <Northern>-206</Northern> ← Northern offset in centimeters for node 2
                  <Elevation>0</Elevation> ← Elevation offset in centimeters for node 2
            </Node>
      </Nodes>
```

```
</Region>
```

Appendix D. MAP XML File Format

The MAP plugin uses an XML input file as one option tocreate the SAE J2735-2016 MAP message for the intersection. The input file contains a reference point for the intersection, 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>
```

Appendix E. Pedestrian Safety Message XML Format

PSM XML format:

```
<PersonalSafetyMessage>
      <basicType>
            <aPEDESTRIAN/>
      </basicType>
      <secMark>0</secMark>
      <msgCnt>0</msgCnt>
      <id>87654321</id>
      <position>
            <lat>406680509</lat>
            <long>-738318466</long>
            <elevation>40</elevation>
      </position>
      <accuracy>
            <semiMajor>255</semiMajor>
            <semiMinor>255</semiMinor>
            <orientation>65535</orientation>
      </accuracy>
      <speed>75</speed>
      <heading>3672</heading>
      <crossState>
            <true/>
      </crossState>
      <clusterSize>
            <medium/>
      </clusterSize>
      <clusterRadius>6</clusterRadius>
</PersonalSafetyMessage>
```

Traveller Information Message XML POST sample

```
<TravelerInformation>
    <msgCnt>1</msgCnt>
    <timeStamp>115549</timeStamp>
    <packetID>000000000023667BAC</packetID>
    <dataFrames>
      <TravelerDataFrame>
        <sspTimRights>0</sspTimRights>
        <frameType>
          <advisory/>
        </frameType>
        <msgId>
          <roadSignID>
            <position>
              <lat>389549775</lat>
              <long>-771491835</long>
              <elevation>390</elevation>
            </position>
            <viewAngle>1111111111111111</viewAngle>
            <mutcdCode>
              <warning/>
            </mutcdCode>
          </roadSignID>
        </msgId>
        <startTime>139255</startTime>
        <duratonTime>1</duratonTime>
        <priority>7</priority>
        <sspLocationRights>0</sspLocationRights>
        <regions>
          <GeographicalPath>
            <anchor>
              <lat>389549775</lat>
              <long>-771491835</long>
              <elevation>390</elevation>
            </anchor>
            <directionality>
              <both/>
            </directionality>
            <closedPath>
              <true/>
            </closedPath>
            <description>
              <geometry>
                <direction>1111111111111111</direction>
                <circle>
                  <center>
                    <lat>389549775</lat>
                    <long>-771491835</long>
                    <elevation>390</elevation>
                  </center>
                  <radius>74</radius>
                  <units>
                    <meter/>
```

```
                    </units>
                  </circle>
                </geometry>
              </description>
            </GeographicalPath>
          </regions>
          <sspMsgRights1>0</sspMsgRights1>
          <sspMsgRights2>0</sspMsgRights2>
          <content>
            <advisory>
              <SEQUENCE>
                <item>
                  <text>27</text>
                </item>
              </SEQUENCE>
              <SEQUENCE>
                <item>
                  <itis>9476</itis>
                </item>
              </SEQUENCE>
              <SEQUENCE>
                <item>
                  <itis>13569</itis>
                </item>
              </SEQUENCE>
            </advisory>
          </content>
          <url>987654378</url>
        </TravelerDataFrame>
      </dataFrames>
    </TravelerInformation>
```

Appendix G. 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. (2009, September 25). 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