IVP Deployment Guide

Deployment of IVP at a Signalized Intersection to Send SPAT and MAP

Created by Battelle

Version 4.0

May , 2016

1 Contents

2 Summary

This document describes the installation and configuration of a typical Integrated Vehicle to Infrastructure Prototype (IVP) unit at an intersection sending SAE J2735: 2015 (a SAE standard for Dedicated Short Range Communications) Signal Phase and Timing (SPaT) and MAP messages. The target audience for this document is the IT personnel for the deploying agency. The end user must have knowledge using telnet, secure shell (ssh), secure copy (scp), and basic command line tasks in Linux.

The IVP system is comprised of the IVP Communication Hub (IVP Hub) hardware running the IVP software. Any external components used by IVP like the Traffic Signal Controller (TSC), Road Side Unit (RSU), etc. will need to be obtained and deployed separately. This document includes sections on how to configure an Arada Locomate Commando RSU and an Econolite TSC for use with IVP, but these are not the only RSUs and TSCs that can be used with the IVP system.

This document assumes that the end user already has an IVP Hub with the current software on it. This document describes the physical connections needed for the IVP software to function correctly. While there are many possible uses for the IVP system, including placement at a curve to send Curve Speed Warning (CSW) alerts and placement at a rail crossing to send information for a Rail Crossing Violation Warning application, this document only describes the installation of the IVP system at an intersection. A sample diagram of an intersection deployment is shown in Figure 1.

The IVP software was created and tested on Ubuntu 14.04 LTS, but can run on most Linux operating systems. The IVP software uses a plugin architecture so that each installation can be configured to run a suite of different software applications and plugins. The main software application, ivpcore, 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 IVP software is created to do a single function, such as communicate with a signal controller and produce J2735 SPaT messages. This document provides instructions to setup an IVP system that contains four plugins; Traffic Signal Controller Plugin, MAP Plugin, GPS Plugin, and Real Time Correction Management (RTCM) Plugin. These plugins were developed under the IVP project for U.S. DOT.

3 Required IVP Hardware and Software

The hardware and software listed below are the minimum requirements for installing and deploying IVP at an intersection. Photos of these required items are included on the following pages.

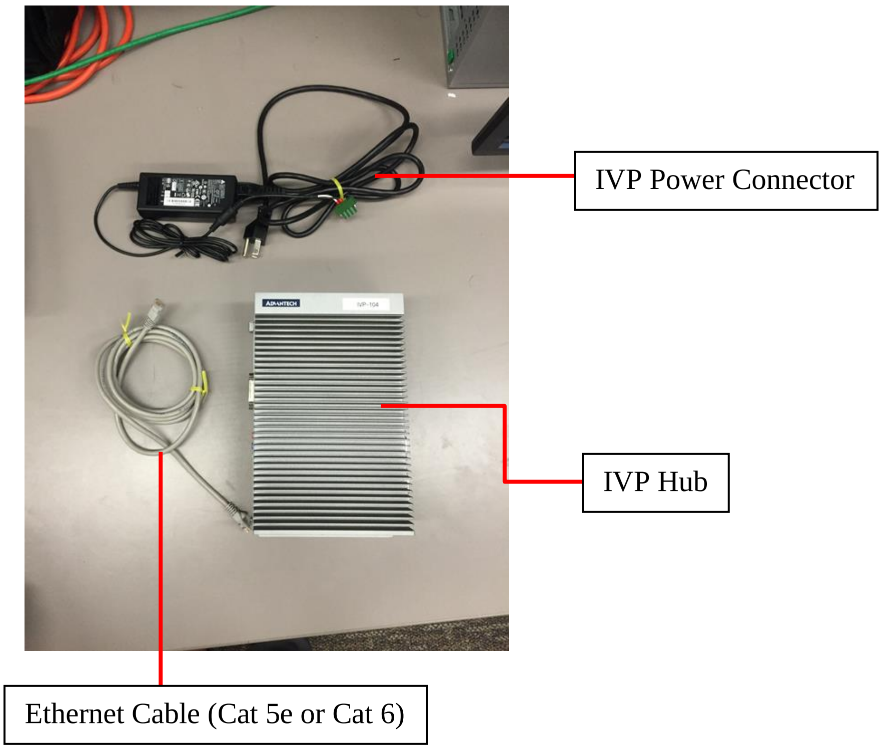
Hardware

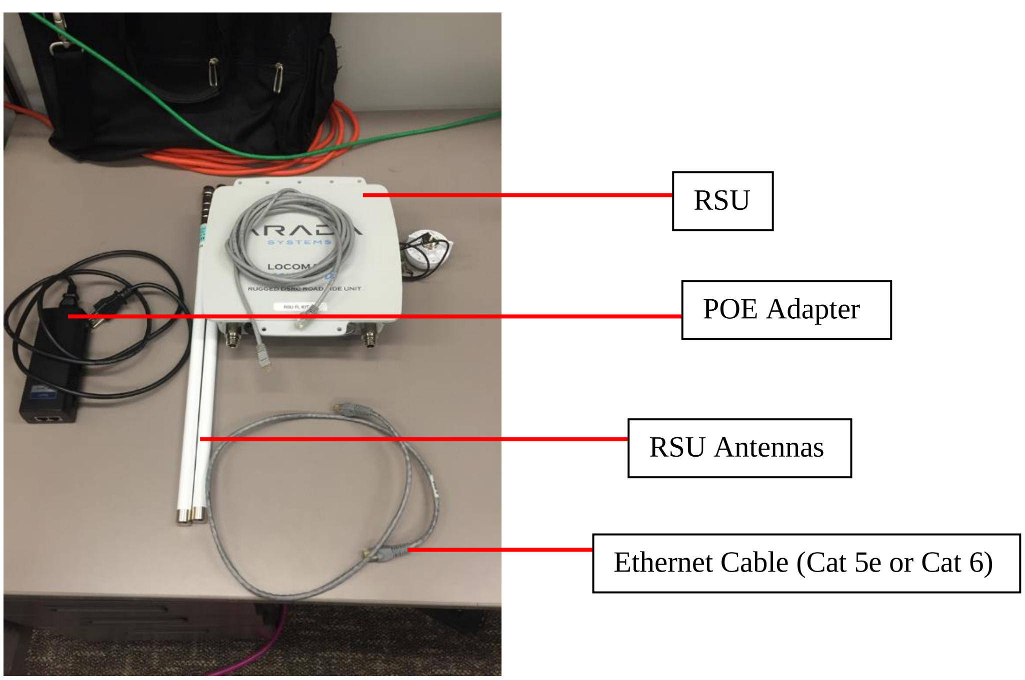
1. IVP Hub
2. IVP Power Connector
3. Traffic Signal Controller (TSC) with an available 110 volt power outlet (example: Econolite ACS3 w/ latest firmware update)
4. Road Side Unit (RSU) installed external to the traffic signal controller cabinet (example: Arada Locomate Commando)
5. GPS RSU attachment
6. On-board Unit (OBU) (example: Arada Locomate Mini 2, Arada Locomate Me)
7. Tablet for vehicle interface
8. Power over Ethernet (PoE) adapter
9. Maintenance computer/Laptop
10. Four (4) Ethernet Cables (Cat 5e or Cat 6)
11. IPv4 or IPv6 switch

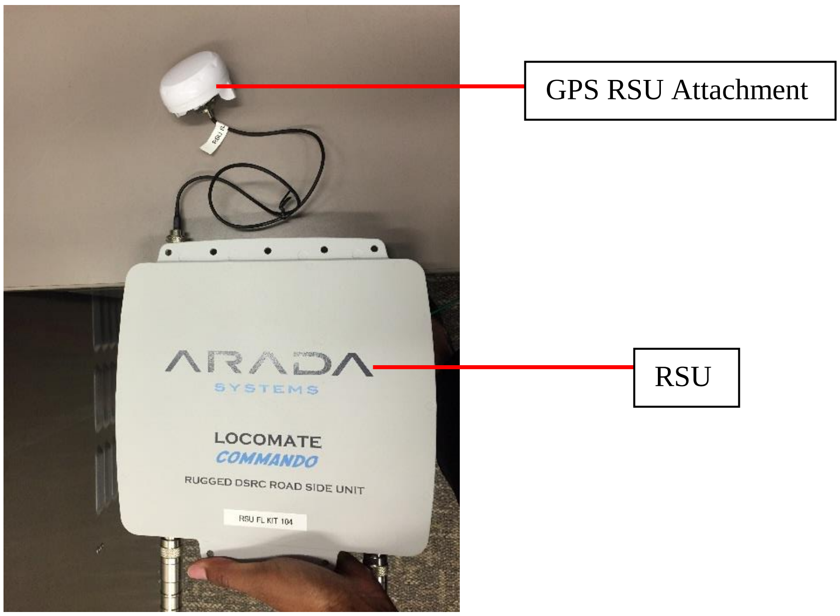
Software

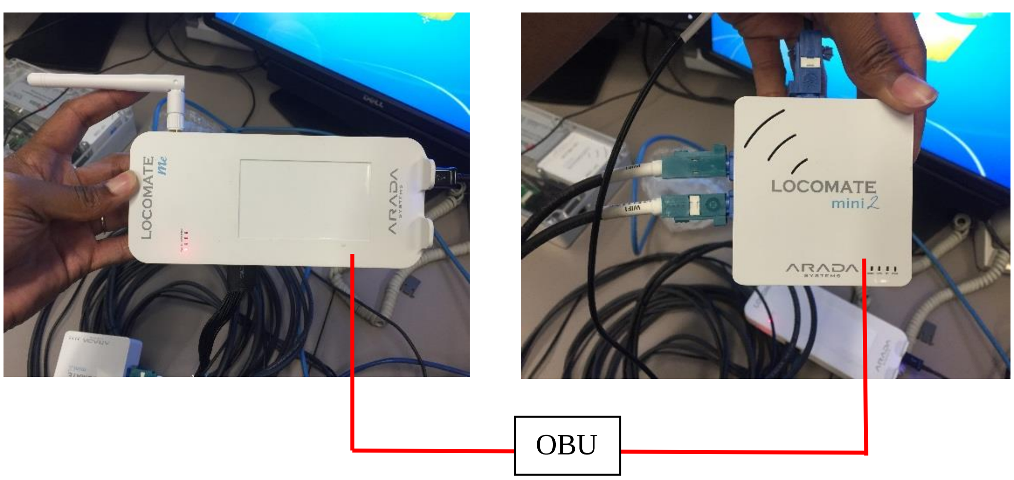
1. The IVP core application, pre-installed on the IVP Hub
2. Maintenance computer with Windows or Linux operating systems
3. Maintenance computer with PuTTY.exe or other telnet/ssh client software installed
4. Maintenance computer with WinSCP.exe software installed
5. Maintenance computer with web browser (i.e. Firefox, Internet Explorer, etc.)

4 Photos of the IVP Hardware Components



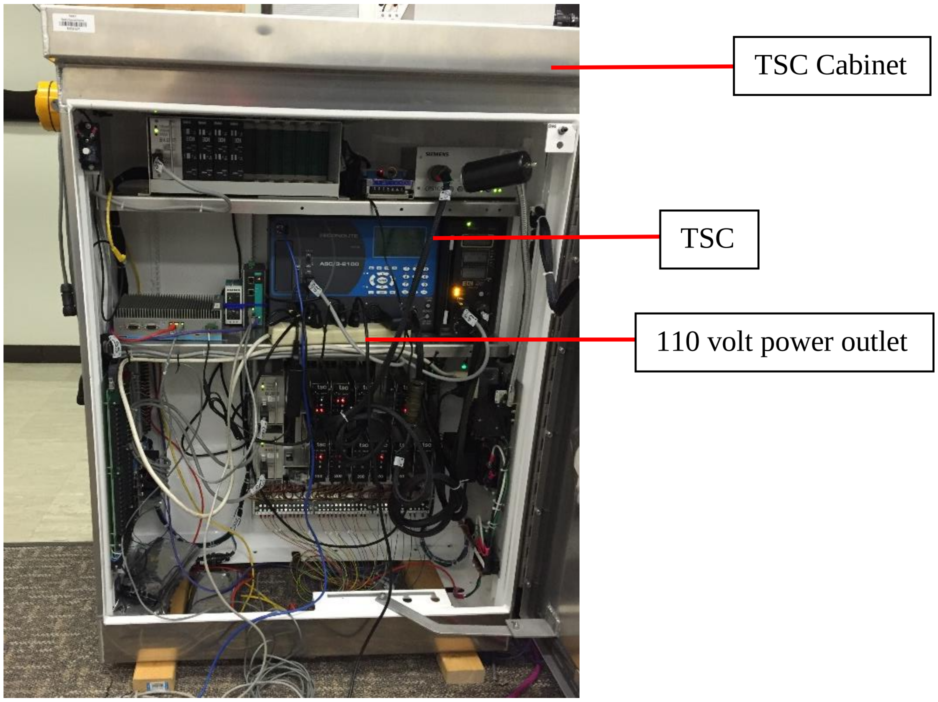



GPS RSU Attachment
RSU
ARADA
SYSTEMS
LOCOMATE
COMMANDO
RUGGED DSRC ROAD SIDE UNIT
RSU FL KIT 104


LOCOMATE
ARADA
LOCOMATE
mini2
ARADA
OBU


GRIFFIN
Tablet for Vehicle Interface


MOXA
IPv4 or IPv6 switch


TSC Cabinet
TSC
110 volt power outlet

5 Diagram of Sample Intersection Deployment

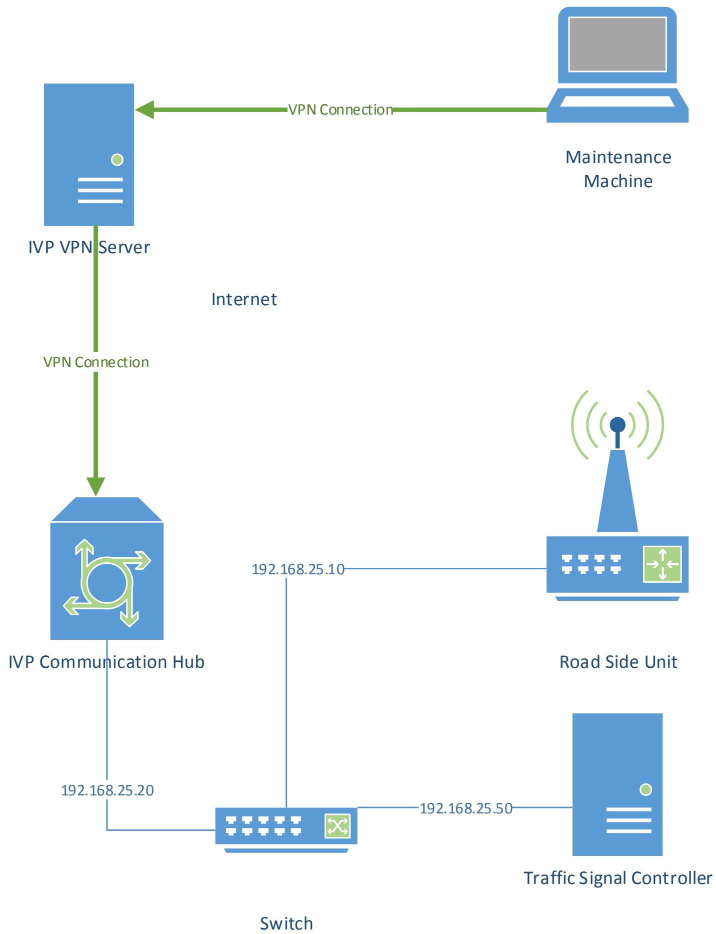


FIGURE 1 SAMPLE INTERSECTION DEPLOYMENT

6 IVP System Prerequisites

Certain prerequisites must be satisfied before the IVP system is installed within the signal controller cabinet. First, the cabinet must have the following:

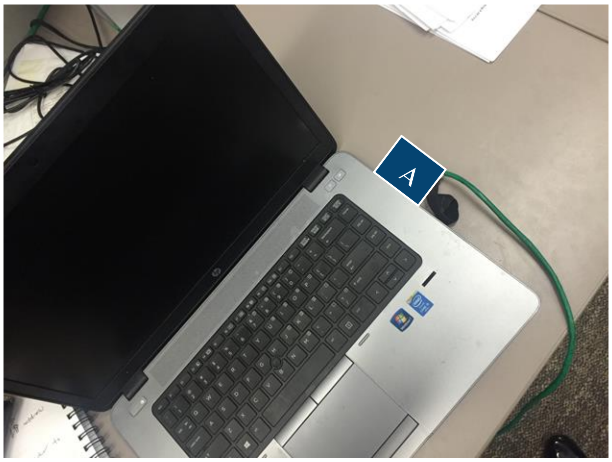
- Open 110-volt power outlet for the IVP box.
- Traffic signal controller sending the Traffic Signal Controller Broadcast Message as defined in the *Interface Control Document for the Signal Phase and Timing and Related Messages for V-I Applications, 2013.* The Traffic Signal Controller Broadcast Message table can be referenced in Appendix B: Traffic Signal Controller Broadcast Message
-
- Road side unit installed external to the cabinet, but powered from a Power over Ethernet (PoE) adapter from inside the cabinet.
- Maintenance computer (e.g. laptop) to use during system configuration. This computer is only used during configuration and does not stay with the system. It must have the following software:
    - PuTTY or other telnet/ssh client.
    - Web Browser

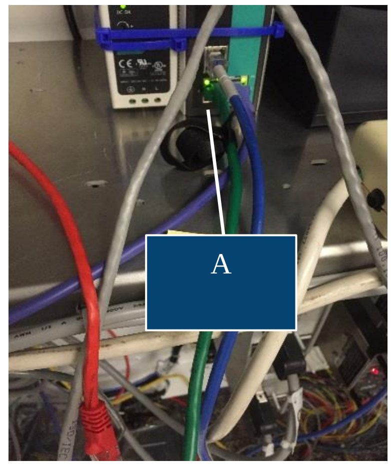
  IPv4 switch with four or more ports for connectivity to the RSU, TSC, IVP Hub, and Maintenance computer.
    - Optionally a switch capable of IPv4 and IPv6 can be utilized. The RSU can communicate using IPv4 or IPv6.
    - All devices connected to this switch must have static IP addresses on the same class C network (i.e. netmask of 255.255.255.0).
    - An example configuration is:
        - IVP Hub: 192.168.25.20
        - TSC: 192.168.25.50
        - RSU: 192.168.25.10
        - Maintenance computer: 192.168.25.200
- Optional Internet connection in the cabinet for remote connectivity to the IVP Hub using a second Network Interface Controller (NIC). For this connection, the IVP Hub obtains this IP address via DHCP from a Virtual Private Network (VPN) server.

7 IVP Hardware Photos

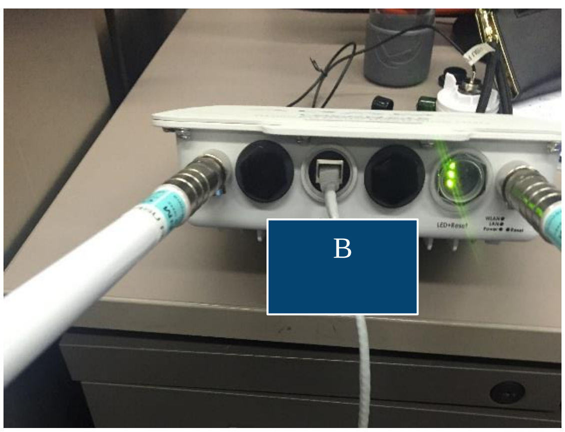
The following is a sequential set of steps that describes how to connect the hardware components needed to deploy IVP.

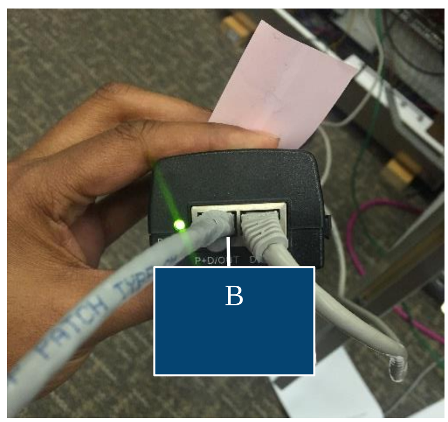
1. Connect the maintenance computer to any Ethernet port on the IPv4 or IPv6 Switch via an Ethernet cable



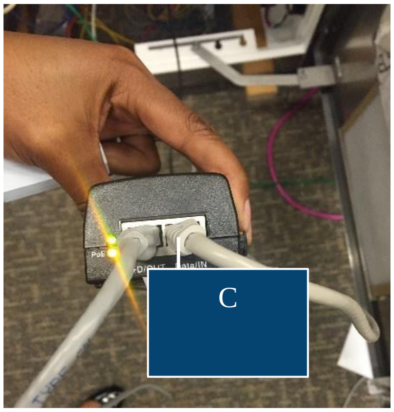


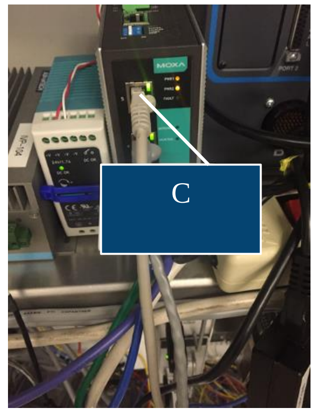
2. Connect the RSU to the Power and Data port on the POE adapter with an Ethernet cable



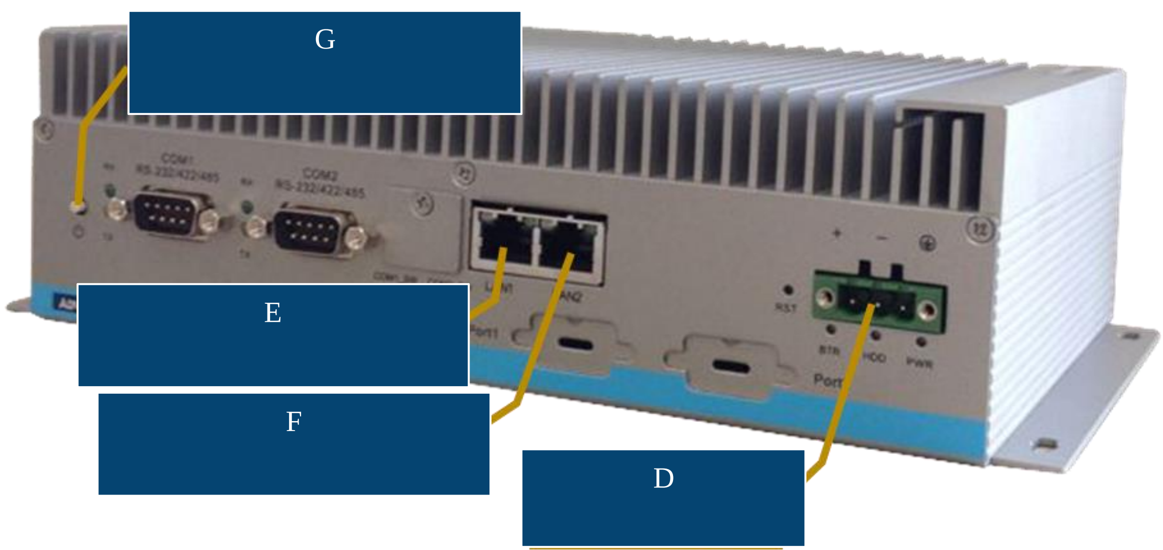


3. Connect an Ethernet cable to the Data port on the POE adapter and connect the other end to any port on the IPv4/IPv6 switch. Connect the POE adapter to the power supply using the power adapter that came with the POE adapter.





4. Plug the IVP Power Connector into the IVP Hub (D) and connect the IVP Power Connector to an electrical outlet. Connect an Ethernet cable to the LAN port on the IVP Hub (E) and connect the opposite end to any port on the IPv4/IPv6 switch. Connect another Ethernet cable to the WAN port of the IVP Hub (F) and connect the opposite end of the Ethernet cable to an active internet/VPN server. Push the power button on the IVP Hub (G).



8 Supporting Software, TSC Verification, IVP Hardware Connection, Plugins & Communication Web Portal

This section of the document will detail the steps required to:

1. Install the software applications used to log-in, support and transfer files to the IVP hardware elements.
2. Verify the outputs of a properly configured traffic signal controller
3. Connect the IVP Hub hardware
4. Verify the outputs the IVP Hub plug-ins. The plug-ins will be pre-loaded to the IVP Hub and assumed to be correctly configured. However, if the IVP plug-ins need to be modified/updated, this section will also describe the steps required to change common parameters.
5. View/Verify Plugin Message Activity

8.1 Installing the Supporting Software Applications

A temporary maintenance computer is recommended for configuration of the components in the traffic signal controller cabinet. This computer must have an IP address in the same subnet as the other devices and be attached to the local IPv4/Ipv6 switch as described in the IVP System Prerequisites and IVP Hardware Set-Up Photos. Alternatively, components can be configured remotely through a VPN connection if the IVP Hub is connected to the Internet and configured for VPN access. VPN setup is beyond the scope of this document.

The maintenance computer can run either the Windows or Linux operating system as described in the following sections.

For a **Windows maintenance computer**, the recommended applications are PuTTY and WinSCP.

- PuTTY is a telnet and SSH (Secure Shell) client for Windows. Telnet is used to connect to the RSU for configuration and SSH is used to connect to the IVP Hub. PuTTY is a free software package available at: http://www.putty.org. All that is required is the *putty.exe* executable.

- WinSCP is a SCP (Secure Copy Protocol) client for Windows. It is used to transfer files to a Linux computer (e.g. IVP Hub, RSU). WinSCP is a free software package available at: https://winscp.net. **Note: the WinSCP application may only be necessary to download if updates/modifications to the IVP plug-ins are required**.

For a **Linux maintenance computer**, no special applications are required when Linux is the operating system on the maintenance computer. The standard Bash Unix shell can be used for all telnet, SSH, and SCP operations. For example, the following are valid commands that can be executed in a Bash shell:

- *telnet 192.168.25.10*
- *ssh 192.168.25.20*
- *scp map.xml ivp@192.168.25.20:~/*

8.2 TSC Verification

The following steps detail how to ensure that the TSC is configured to communicate with the IVP Hub. Select the following sequence of menus using the TSC controls:

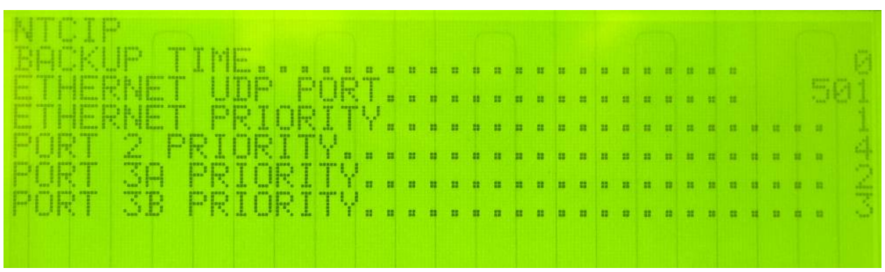
MAIN MENU → 1. CONFIGURATION → 5. COMMUNICATIONS → 1. ETHERNET

The following example shows the TSC with a local IP address of 192.168.25.50. In this case, the SERVER IP field refers to the IP of the IVP Hub. It is configured to send the Traffic Signal Controller Broadcast Message to the IVP Hub at 192.168.25.20. The UDP port used by the IVP Hub to receive communications from the TSC cannot be modified in this screen on the Econolite TSC, and is hard coded to 6053. If the IP Address of the TSC is not on the same network as the IVP Hub, please refer to your TSC's user guide to change the IP address.

Select the following sequence of menus using the TSC controls:

MAIN MENU → 1. CONFIGURATION → 5. COMMUNICATIONS → 5. NTCIP

The following example shows that the TSC is configured for SNMP NTCIP communications using port 501.



8.3 IVP Hub Configuration

The IVP Hub comes configured with a static ip address of 192.168.25.20 for its LAN port. Follow the steps below to change the ip address of the IVP Hub for your network.

To Log-in to the IVP Hub:

- The maintenance machine and IP 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)
- Open the Putty *putty.exe* application
- Enter the **IP Address (Default: 192.168.25.20)**
- Select **SSH** for connection type
- Click **Open**, as shown in Figure 2.
- **Note: the configuration is the same as above when using PuTTY as an SSH client, except the SSH Connection type should be specified.**
- Once the command prompt appears, change directory to /etc/network (cd /etc/network)
- Open the interfaces file in nano (sudo nano interfaces) or a text editor of your choice as super user by prefixing the command with “sudo”
    - Enter **B@ttelle** when prompted for password
- Modify 192.168.25.20 to your ip address
    - Save the file (nano ctrl + X, enter Y for save file)
- Reboot the IVP hub by typing sudo reboot

8.4 Road Side Unit (RSU) Log-in and Confirmation

This section describes how to log-in and configure the Arada Locomate Commando RSU to send J2735 SPaT, MAP, and RTCM messages via the radio when they are received from the IVP system. The instructions were created with an Arada Locomate Commando RSU v3.1 running firmware WAVE_LOCOMATE-200_1.90.0.16_firmware. The instructions assume that the PuTTY application is used as the telnet client.

To Log-in to the RSU:

- Open the Putty *putty.exe* application
- Enter the **Host Name (IP Address)**
- Select **Telnet** for connection type
- Click **Open**, as shown in Figure 2.
- **Note: the configuration is the same as above when using PuTTY as an SSH client, except the SSH Connection type should be specified.**
- Once the black log-in screen appears, type **ivp** for login
- Type **B@ttelle** for the password. **Note: the cursor may not move and the password may not display while entering in the corresponding password**

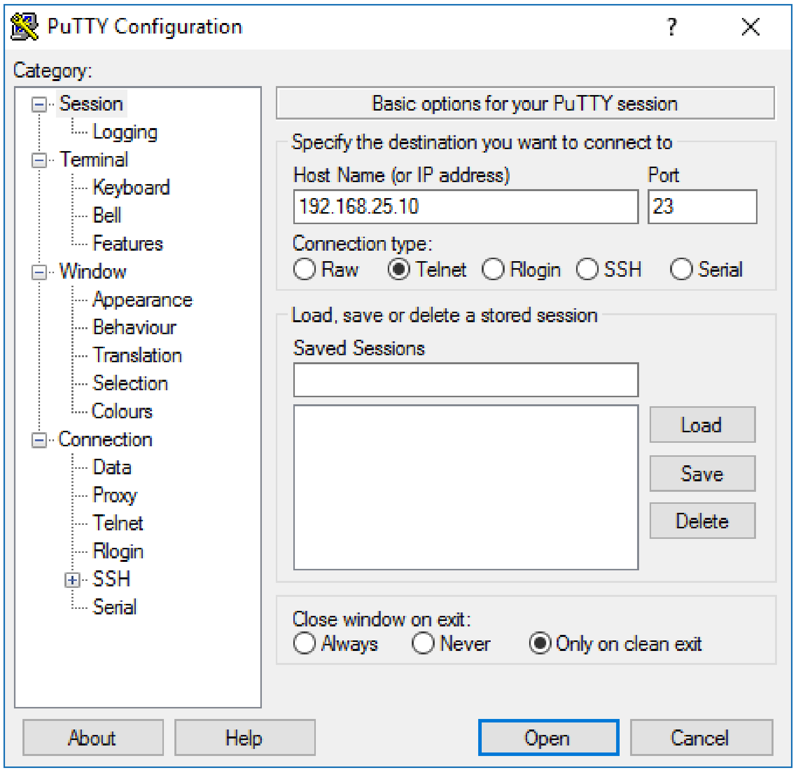
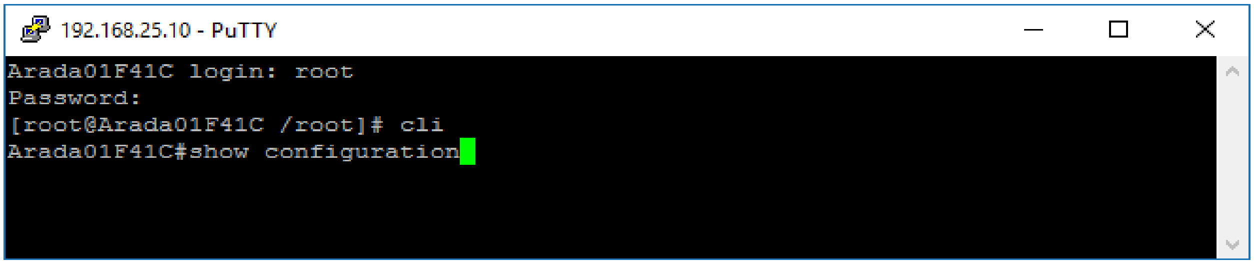
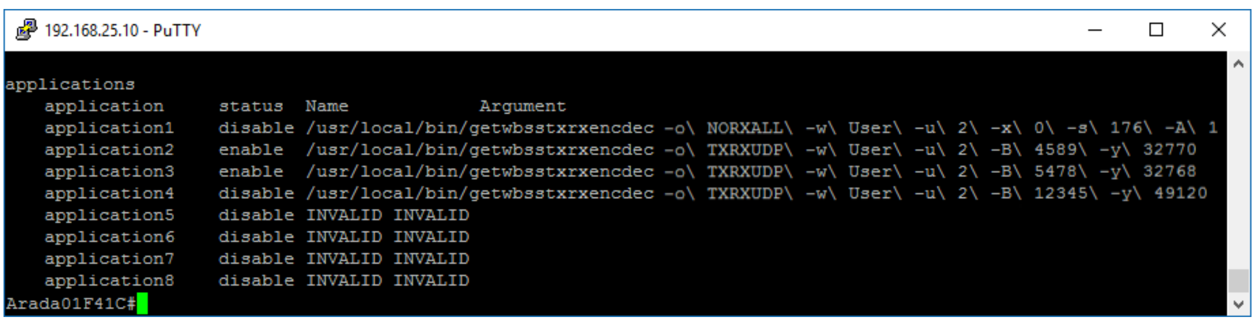


FIGURE 2 PUTTY CONFIGURATION

- Once the black log-in screen appears, type **root** for login
- Type **B@ttelle** for the password. **Note: the cursor may not move and the password may not display while entering in the corresponding password**
- Type **cli** to enter the Arada command line interface.
- Type **show configuration** to list the current configuration as shown in the figure below.



- Press enter
- After Enter is pressed, at the end of the output, the list of configured applications is displayed as shown in the figure below. This figure shows the final desired configuration, where the *getwbsstxrxencdec* application is configured twice to forward the appropriate messages.
- Examine the output. **All applications should be disabled except for application2 and application3**.
- Ensure that the command line appear exactly as displayed below. If the output matches the figure below exactly (note that any application with a status of disable can be ignored), then you can continue to the next steps. If the output does not match the figure below, see Appendix C.



- To close the command line interface, type **exit** and press enter.
- Type **exit** and press enter again to close the putty application.
- **Note: the Arada command line interface is entered by typing *cli*. Multiple commands can be given while in the command line interface. *cli* should not be typed while already in the command line interface.**

8.5 IVP Hub Plug-ins and Communication Web Portal

The IVP Hub is preconfigured to operate at an intersection sending J2735 SPAT and J2735 MAP messages. This includes the following four plugins:

1. DSRC Message Manager

    - Listens for messages and forwards them to the DSRC Radio (i.e. the RSU).
    - The Arada RSU is currently the only supported radio.
2. MAP r41
    - Reads intersection geometry from a configuration file and publishes a J2735 r41 MAP message.
3. SPaT r41
    - Reads Phase to Lane Mapping (PTLM) data from a configuration file, receives live data from the signal controller, and publishes a J2735 r41 SPaT message.
4. RTCM
    - Connects and listens to an NTRIP caster and publishes a J2735 RTCM message.

8.5.1 IVP Communication Hub Web Portal

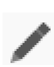
Plugin installation and configuration is performed in a web portal running on the IVP Communication Hub. To access and login to the portal:

- Ensure that the Maintenance computer is configured and attached to the IPv4/IPv6 local switch
- Open a web browser on the maintenance computer (i.e. Firefox, Internet Explorer)
- Navigate to the following address: **http://ipaddress/ivp**
    - Note: the *ipaddress* is the IP address of the IVP Communication Hub
    - Example: http://192.168.25.20/ivp
- Login with system administrator credentials
    - **User: _battelle**
    - **Password: B@ttelle**
- Under the System Admin Menu, click Installed Plug-ins to view the pre-loaded plug-ins on the IVP hub

Figure 3 shows the *Installed Plugins* page with the four preinstalled plugins. This page shows all installed plugins, common properties of each plugin, and controls to configure and update each plugin. The available controls are:

 The *gears* icon shows all plugin specific configuration parameters and allows them to be edited.

 The *upload* icon allows a new version of the plugin to be uploaded.

 The *pencil* icon edits the record of the plugin (i.e. Enabled/Disabled, Max Message Interval).

 The *trashcan* icon deletes the plugin.

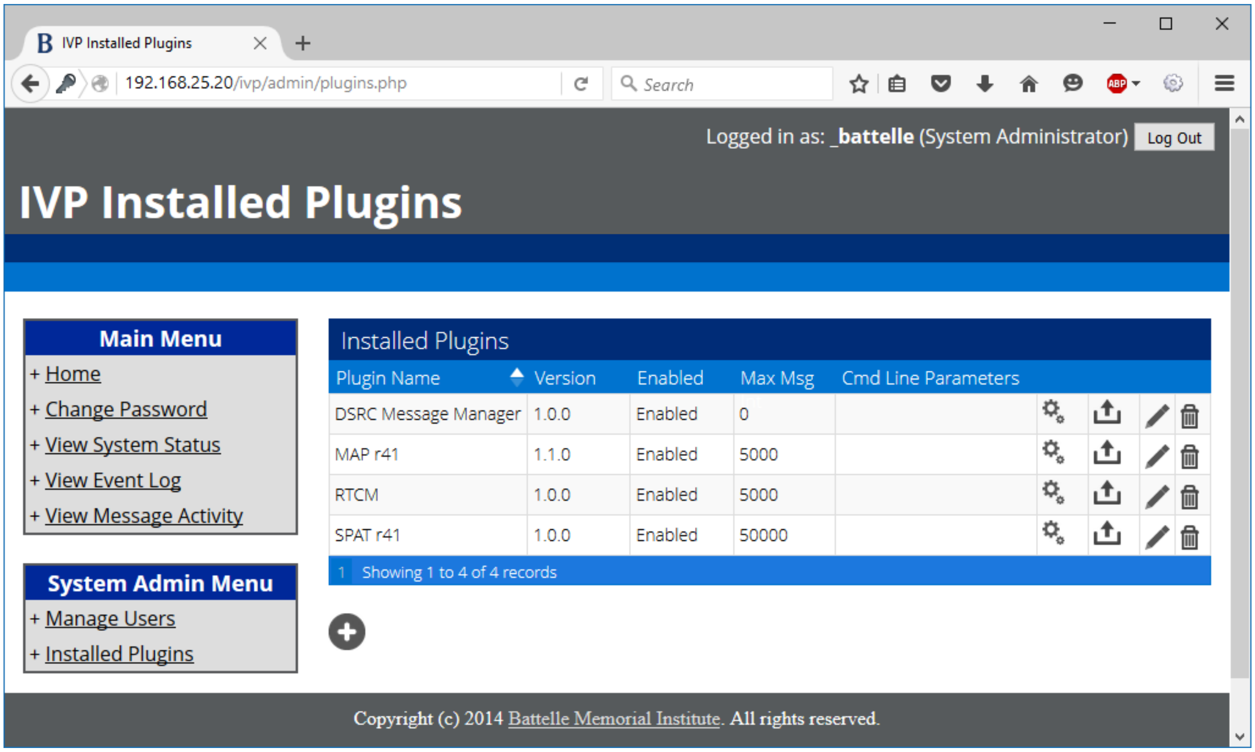


FIGURE 3 WEB PORTAL INSTALLED PLUGINS

8.5.2 DSRC Message Manager Plugin

To confirm the parameters for the DSRC Message Manager Plugin:

- Navigate to the *Installed Plugins* page of the web portal
- Click the *gears* icon for DSRC Message Manager to display all configuration parameters of the plugin as shown in Figure 4.
- Use the *pencil* icon next to each configuration parameter to change a value
    - Most parameters are preconfigured with the correct values.
- Verify that the IP address under the DSCR_Radio_IP is correct**.** If not, edit the following parameters as needed.
    - **DSRC_Radio_IP** – Specify the correct IP address for the DSRC radio (i.e. the RSU).

Plugin Name	Version	Enabled	Max Msg Int	Cmd Line Parameters
DSRC Message Manager	1.0.0	Enabled	0	

Configuration Parameters

Key	Value	Default Value	Description
DSRC_Radio_IP	192.168.25.10	192.168.25.10	The IPv4 or IPv6 address of the destination DSRC radio (i.e. the RSU).
Messages_UDP_Port_1	{ "Messages": [{ "TmxType": "SPAT-P", "SendType": "SPAT", "PSID": "0x8002" }, { "TmxType": "MAP-P", "SendType": "MAP", "PSID": "0x8002" }] }	{ "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 on UDP Port 1.
Messages_UDP_Port_2	{ "Messages": [ ] }	{ "Messages": [ ] }	JSON data defining the message types and PSIDs for messages forwarded to the DSRC radio on UDP Port 2.
Messages_UDP_Port_3	{ "Messages": [ ] }	{ "Messages": [ ] }	JSON data defining the message types and PSIDs for messages forwarded to the DSRC radio on UDP Port 3.
Messages_UDP_Port_4	{ "Messages": [ ] }	{ "Messages": [ ] }	JSON data defining the message types and PSIDs for messages forwarded to the DSRC radio on UDP Port 4.
Signature	False	False	True or False value indicating whether to sign the messages.
UDP_Port_1	4589	4589	The destination UDP port number on the DSRC radio for all messages specified by Messages_UDP_Port_1.
UDP_Port_2	0	0	The destination UDP port number on the DSRC radio for all messages specified by Messages_UDP_Port_2.
UDP_Port_3	0	0	The destination UDP port number on the DSRC radio for all messages specified by Messages_UDP_Port_3.
UDP_Port_4	0	0	The destination UDP port number on the DSRC radio for all messages specified by Messages_UDP_Port_4.

1 Showing 1 to 10 of 10 records

FIGURE 4 DSRC MESSAGE MANAGER CONFIGURATION PARAMETERS

8.5.3 MAP r41 Plugin

The MAP r41 plugin requires one or more MAP XML files to be uploaded to the IVP Hub. Instructions on creating a MAP XML file can be found Appendix D: RSU Application Configuration. To confirm the parameters for the MAP r41 Plugin:

- Navigate to the *Installed Plugins* page of the web portal
- Click the *gears* icon for MAP r41 to display all configuration parameters of the plugin as shown in Figure 5
- Use the *pencil* icon next to each configuration parameter to change a value
    - Most parameters are preconfigured with the correct values.

- Verify that the value in **MAP_ Files** is correct**.** If not, edit the following parameters as needed.
    - **MAP_Files** – This parameter specifies the correct MAP XML file to load for the environment (e.g. intersection) for each action set specified by the TSC. Many intersections will only have one MAP file specified, as in the example shown in Figure 5. Each map file specified by the *FilePath* JSON key must be present in the plugin folder as described earlier.
        - Use WinSCP or ssh/scp to transfer the XML files containing geometry for the intersection to the IVP Hub. The XML file must be placed in the folder for the MAP r41 plugin: */var/www/plugins/MAPr41*.

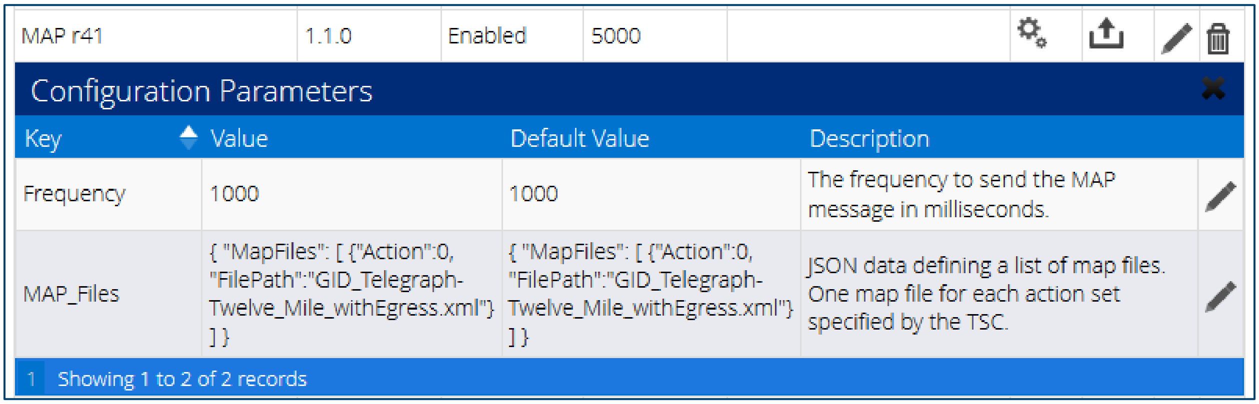
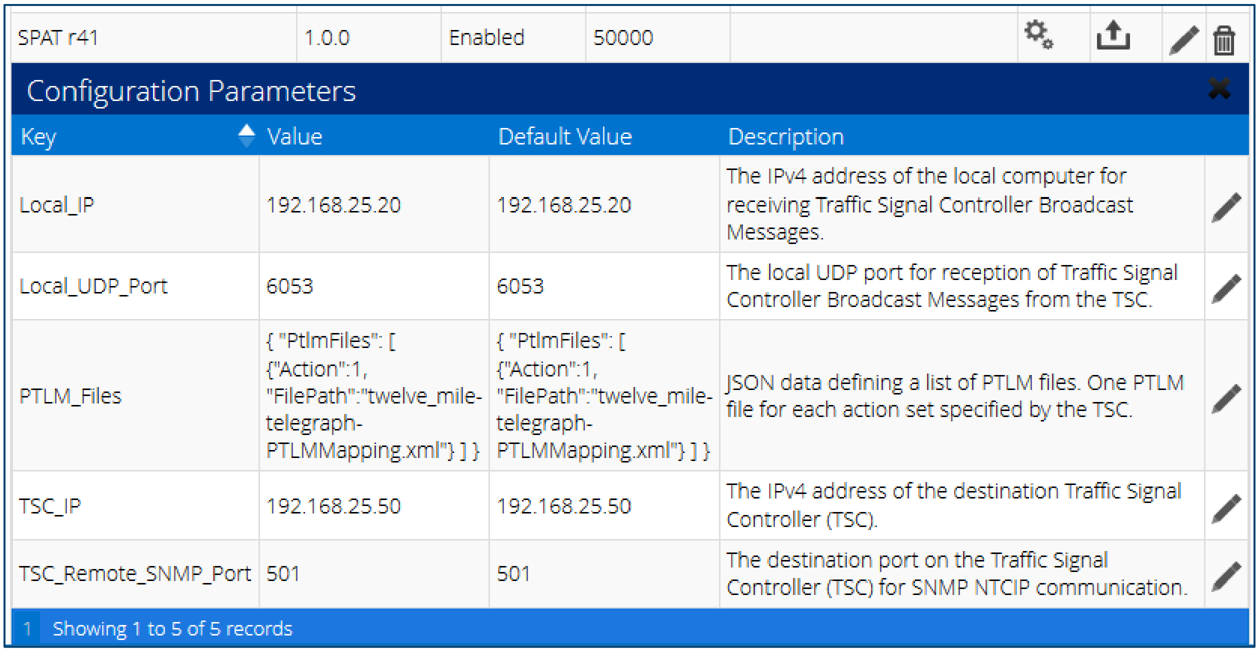


FIGURE 5 MAP R41 PLUGIN CONFIGURATION PARAMETERS

8.5.4 SPAT r41 Plugin

The SPAT r41 plugin requires one or more PTLM XML files to be uploaded to the IVP Hub. To confirm the parameters the SPAT r41 Plugin:

- Navigate to the *Installed Plugins* page of the web portal
- Click the *gears* icon for SPAT r41 to display all configuration parameters of the plugin as shown in Figure 6.
- Use the *pencil* icon next to each configuration parameter to change a value
    - Most parameters are preconfigured with the correct values.
- Verify that the value in **Local_IP** and **PTLM_Files** are correct**.** If not, edit the following parameters as needed.
    - **Local_IP** – Specify the correct IP address of the local IVP Hub computer (i.e. 192.168.25.20 is the default).
    - **PTLM_Files** – This parameter specifies the correct PTLM XML file to load for the environment (e.g. intersection) for each action set specified by the TSC. Many intersections will only have one PTLM file specified, as in the example shown in Figure 6. Each map file specified by the *FilePath* JSON key must be present in the plugin folder as described earlier.
        - Use WinSCP or ssh/scp (see section 8) to transfer the XML files containing PTLM data for the intersection to the IVP Hub. The XML file must be placed in the folder for the SPAT r41 plugin: */var/www/plugins/SPATr41*.



SPAT r41	1.0.0	Enabled	50000

Configuration Parameters

Key	Value	Default Value	Description
Local_IP	192.168.25.20	192.168.25.20	The IPv4 address of the local computer for receiving Traffic Signal Controller Broadcast Messages.
Local_UDP_Port	6053	6053	The local UDP port for reception of Traffic Signal Controller Broadcast Messages from the TSC.
PTLM_Files	{ "PtlmFiles": [{"Action":1, "FilePath":"twelve_mile-telegraph-PTLMMapping.xml"}] }	{ "PtlmFiles": [{"Action":1, "FilePath":"twelve_mile-telegraph-PTLMMapping.xml"}] }	JSON data defining a list of PTLM files. One PTLM file for each action set specified by the TSC.
TSC_IP	192.168.25.50	192.168.25.50	The IPv4 address of the destination Traffic Signal Controller (TSC).
TSC_Remote_SNMP_Port	501	501	The destination port on the Traffic Signal Controller (TSC) for SNMP NTCIP communication.

1 Showing 1 to 5 of 5 records

FIGURE 6 SPAT R41 PLUGIN CONFIGURATION PARAMETERS

8.5.5 RTCM Plugin

To confirm the parameters the RTCM plugin:

- Navigate to the *Installed Plugins* page of the web portal
- Click the *gears* icon for RTCM to display all configuration parameters of the plugin as shown in Figure 7.
- Use the *pencil* icon next to each configuration parameter to change a value.
  - Most parameters are preconfigured with the correct values.
- Verify that the values are correct. If not, edit the following parameters as needed.
  - **Endpoint IP** – Specify the correct IP address of the NTRIP caster endpoint.
  - **Endpoint Port** – Specify the correct NTRIP caster endpoint port
  - **Mountpoint** – Specify the NTRIP caster mountpoint
  - **Username** – The username for the account to access the NTRIP information.
  - **Password** – The password for the account to access the NTRIP information.

RTCM	1.0.0	Enabled	5000

Configuration Parameters

Key	Value	Default Value	Description
Endpoint IP	136.181.134.24	156.63.133.118	NTRIP caster endpoint IP address
Endpoint Port	10001	2101	NTRIP caster endpoint port
Mountpoint	NEAREST_RTCM-GG	ODOT_RTCM23	NTRIP caster mountpoint
Password	B@ttelle	password	NTRIP caster authentication password
Username	ivp	username	NTRIP caster authentication username

1 Showing 1 to 5 of 5 records

FIGURE 7 RTCM PLUGIN CONFIGURATION PARAMETERS

8.5.6 Verify Message Activity

To confirm that all of the plug-ins are responding appropriately:

- Navigate to the *Installed Plugins* page of the web portal
- Under Main Menu, click *View Message Activity*
- The counts for each message from the SPAT and MAP plugins should increment every 2 seconds.

Message Activity

Plugin Name	Message Type	Message Subtype	Count	Last Timestamp	Average Interval
MAP	Battelle-DSRC	GID	86540	2016-05-05 13:30:44	1000
SPaT Generator R41	SIGCONT	ACT	41644	2016-05-05 13:30:43	2000
SPaT Generator R41	J2735	SPAT-P	832888	2016-05-05 13:30:44	100.503

1 Showing 1 to 6 of 6 records

Appendix A: Acronyms, Definitions and Descriptions

Abbreviation	Definition	Description
DSRC	Dedicated Short Range Communications	DSRC is a two-way short- to- medium-range wireless communications protocol that supports vehicle-to-vehicle, vehicle-to-roadside, and roadside-to-vehicle communication.
DHCP	Dynamic Host Configuration Protocol	The Dynamic Host Configuration Protocol (DHCP) is a standardized network protocol used on Internet Protocol (IP) networks for dynamically distributing network configuration parameters, such as IP addresses for interfaces and services.
IPv4	Internet Protocol version 4	Internet Protocol version 4 (IPv4) is the fourth version of the Internet Protocol (IP).
IPv6	Internet Protocol version 6	Internet Protocol version 6 (IPv6) is the most recent version of the Internet Protocol (IP), the communications protocol that provides an identification and location system for computers on networks and routes traffic across the Internet.
IVP	Integrated Vehicle to Infrastructure Prototype	A USDOT project to develop an Integrated Vehicle-to-Infrastructure (V2I) Prototype System that brings results from mapping, positioning, communications research, and Signal Phase and Timing (SPaT) and related message set development into a single operating environment that supports V2I communications-based connected vehicle applications.
J2735	Society of Automotive Engineers (SAE) Standard J2735	the SAE standard J2735 Dedicated Short Range Communications (DSRC) Message Set
MAP	Map Data	Describes the static physical geometry of one or more intersections; i.e., lane geometries and the allowable vehicle movements for each lane, and introduces the idea of "intersection data frame" which describes barriers, pedestrian walkways, shared roadways and rail lines that may affect vehicle movements.
NIC	Network Interface Controller	A network interface controller (also known as a network interface card, network adapter, LAN adapter or physical network interface, and by similar terms) is a computer hardware component that connects a computer to a computer network.
NTCIP	National Transportation Communications for Intelligent Transportation Systems Protocol	A family of standard protocols for allowing traffic management systems to talk to intelligent transportations systems field devices such as: dynamic message signs, CCTV cameras, vehicle detection sensors, traffic signals, Road weather information stations (RWIS), along with many other types for roadway devices
NTRIP	Network Transport of Real Time Correction Management data over Internet Protocol	A protocol for streaming differential GPS data over the Internet in accordance with specification published by RTCM
PoE	Power over Ethernet	describes any of several standardized or ad-hoc systems which pass electrical power along with data on Ethernet cabling
PuTTY	PuTTY	A free and open-source terminal emulator, serial console and network file transfer application that supports several network protocols, including Secure Copy (SCP), Secure Shell (SSH), Telnet, rlogin, and raw socket connection.

Abbreviation	Definition	Description
RSU	Road-Side Unit	DSRC communication unit that is located aside a road that provides connectivity support to passing vehicles
RTCM Plugin	Radio Technical Commission for Maritime Plugin	???
SNMP	Simple Network Management Protocol	Is an Internet-standard protocol for collecting and organizing information about managed devices on IP networks and for modifying that information to change device behavior.
SPaT	signal phase and timing data	Is the real-time provision of traffic signal phase and timing information to vehicles approaching signalized intersections. Describes the signal state of the intersection and how long this state will persist for each approach and lane that is active. The SPaT message sends the current state of each phase, with all-red intervals not transmitted. Movements are given to specific lanes and approaches by use of the lane numbers present in the message.
Telnet/ssh	Telnet/Secure Shell	Telnet is a network protocol that allows a user to communicate with a remote device. It is a virtual terminal protocol used mostly by network administrators to remotely access and manage devices. SSH uses encryption, which means that all data transmitted over a network is secure.
TSC	Traffic Signal Controller	A set of electrically operated signal lights used to direct or control traffic at intersections
VPN	Virtual Private Network	a method employing encryption to provide secure access to a remote computer over the Internet
WinSCP	Windows Secure Copy	An open source Secure file transfer protocol client for Windows. It allows secure file transfers between the client's local computer and the remote server.

Appendix B: Traffic Signal Controller Broadcast Message

Bytes	Description
0	0xcd
1	number of phase/overlap blocks below (16)
2-14	0x01 (phase number) (1 byte) spatVehMinTimeToChange.1 (2 bytes) spatVehMaxTimeToChange.1 (2 bytes) spatPedMinTimeToChange.1 (2 bytes) spatPedMaxTimeToChange.1 (2 bytes) spatOvlpMinTimeToChange.1 (2 bytes) spatOvlpMaxTimeToChange.1 (2 bytes)
15-196	< repeat above for each phase and overlap >
194-209	0x10 (phase#) (1 byte) spatVehMinTimeToChange.16 (2 bytes) spatVehMaxTimeToChange.16 (2 bytes) spatPedMinTimeToChange.16 (2 bytes) spatPedMaxTimeToChange.16 (2 bytes) spatOvlpMinTimeToChange.16 (2 bytes) spatOvlpMaxTimeToChange.16 (2 bytes)
210-215	phaseStatusGroupReds (2 bytes bit-mapped for phases 1-16) phaseStatusGroupYellows (2 bytes bit-mapped for phases 1-16) phaseStatusGroupGreens (2 bytes bit-mapped for phases 1-16)
216-221	phaseStatusGroupDontWalks (2 bytes bit-mapped for phases 1-16) phaseStatusGroupPedClears (2 bytes bit-mapped for phases 1-16) phaseStatusGroupWalks (2 bytes bit-mapped for phases 1-16)
222-227	overlapStatusGroupReds (2 bytes bit-mapped for overlaps 1-16) overlapStatusGroupYellows (2 bytes bit-mapped for overlaps 1-16) overlapStatusGroupGreens (2 bytes bit-mapped for overlaps 1-16)
228-229	flashingOutputPhaseStatus (2 bytes bit-mapped for phases 1-16)
230-231	flashingOutputOverlapStatus (2 bytes bit-mapped for overlaps 1-16)
232	spatIntersectionStatus (1 byte, see **Error! Reference source not found.**)
233	spatDiscontinuousChangeFlag (1 byte, upper 5 bits are message version)
234	spatMessageSeqCounter (1 byte, lower byte of controller up-time)
235-239	spatTimestamp (5 bytes, hours-minute-second-millisecond)
240-241	spatPedestrianCall (2 bytes, bit-mapped for phases 1-16)
242-243	spatPedestrianDetect (2 bytes, bit-mapped for phases 1-16)

Appendix C: RSU Application Configuration

The following steps assume that a new RSU is being configured and that only the applications configured here need to be running. Adjust the application slots and enable/disable states as required for advanced setups.

This section describes how to log-in and configure the Arada Locomate Commando RSU to send J2735 SpaT, MAP, and RTCM messages out the radio when they are received from the IVP system. The instructions were created with an Arada Locomate Commando RSU v3.1 running firmware WAVE_LOCOMATE-200_1.90.0.16_firmware. The instructions assume that the PuTTY application is used as the telnet client.

To Log-in to the RSU:

- Open the Putty *putty.exe* application
- Enter the **Host name (IP Address)**
- Select **Telnet** for connection type
- Click **Open**, as shown in Figure 2.
- **Note: the configuration is the same as above when using PuTTY as an SSH client, except the SSH Connection type should be specified.**

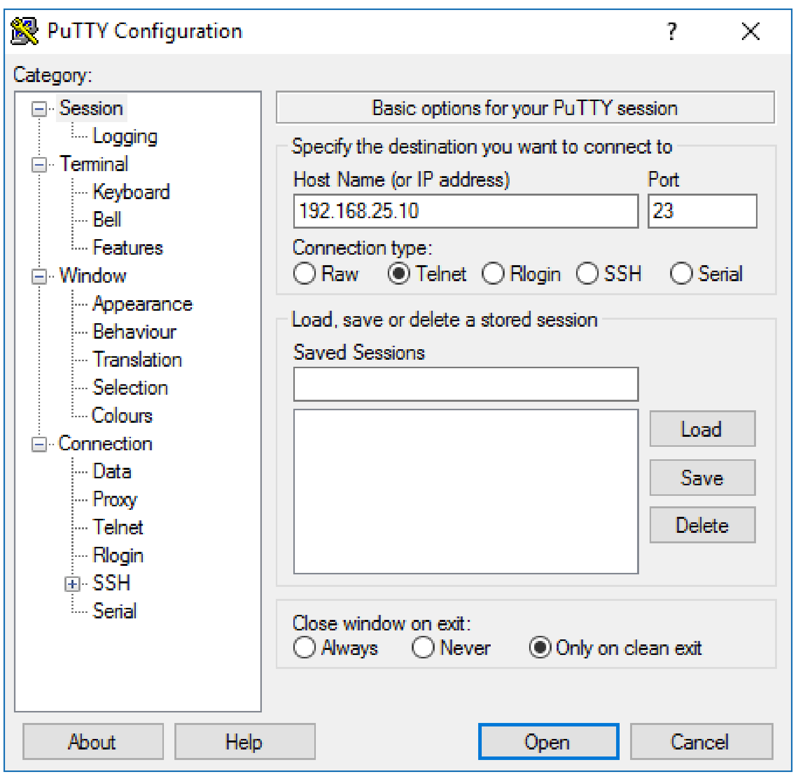
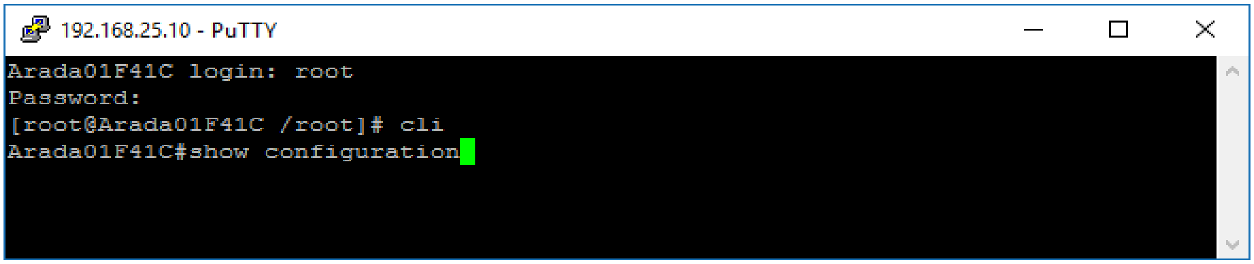
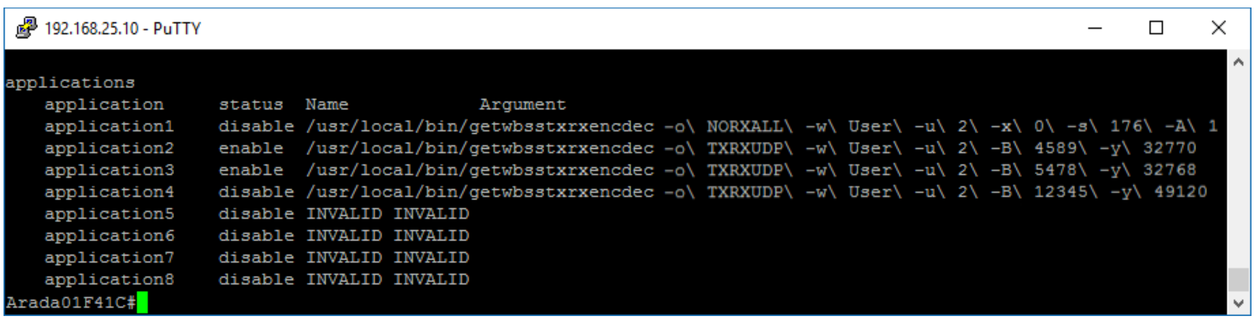


FIGURE 8 PUTTY CONFIGURATION

- Once the black log-in screen appears, type **root** for login
- Type **B@ttelle** for the password. **Note: the cursor may not move and the password may not display while entering in the corresponding password**
- Type **cli** to enter the Arada command line interface.
- Type **show configuration** to list the current configuration as shown in the figure below.



- Press enter
- After Enter is pressed, at the end of the output, the list of configured applications is displayed as shown in the figure below. This figure shows the final desired configuration, where the *getwbsstxrxencdec* application is configured twice to forward the appropriate messages.
- Examine the output. **All applications should be disabled except for application2 and application3.**
- Ensure that the command line appear exactly as displayed below.
- If the output does not match the figure below, follow the next steps



1) Disable all applications except application2 and application 3 as required.
>> cli
Arada# config application app1Status disable
Arada# config application app4Status disable
Arada# config application app5Status disable
Arada# config application app6Status disable
Arada# config application app7Status disable
Arada# config application app8Status disable
Arada# exit

2) Setup the RSU to send the SPaT and MAP J2735 r41 message with PSID 0x8002
>> cli
Arada# config application app2Name /usr/local/bin/getwbsstxrxencdec
Arada# config application app2arg -o TXRXUDP -w User -u 2 -B 4589 -y 32770
Arada# config application app2Status enable
Arada# exit

3) Setup the RSU to send the RTCM messages with PSID 0x8000
>> cli
Arada# config application app3Name /usr/local/bin/getwbsstxrxencdec
Arada# config application app3arg – o TXRXUDP -w User -u 2 -B 5478 -y 32768
Arada# config application app3Status enable
Arada# config exit

4) (Optional) Setup the RSU IP Address. In the instructions below, <ip address> is replaced with the actual IP address (e.g. 192.168.25.10).
>>cli
Arada# config ip address <ip address> 255.255.255.0
Arada# exit
Arada# reboot

- To close the command line interface, type **exit** and press enter
- Type **exit** and press enter again to close the putty application.
- **Note: the Arada command line interface is entered by typing *cli*. Multiple commands can be given while in the command line interface. *cli* should not be typed while already in the command line interface.**

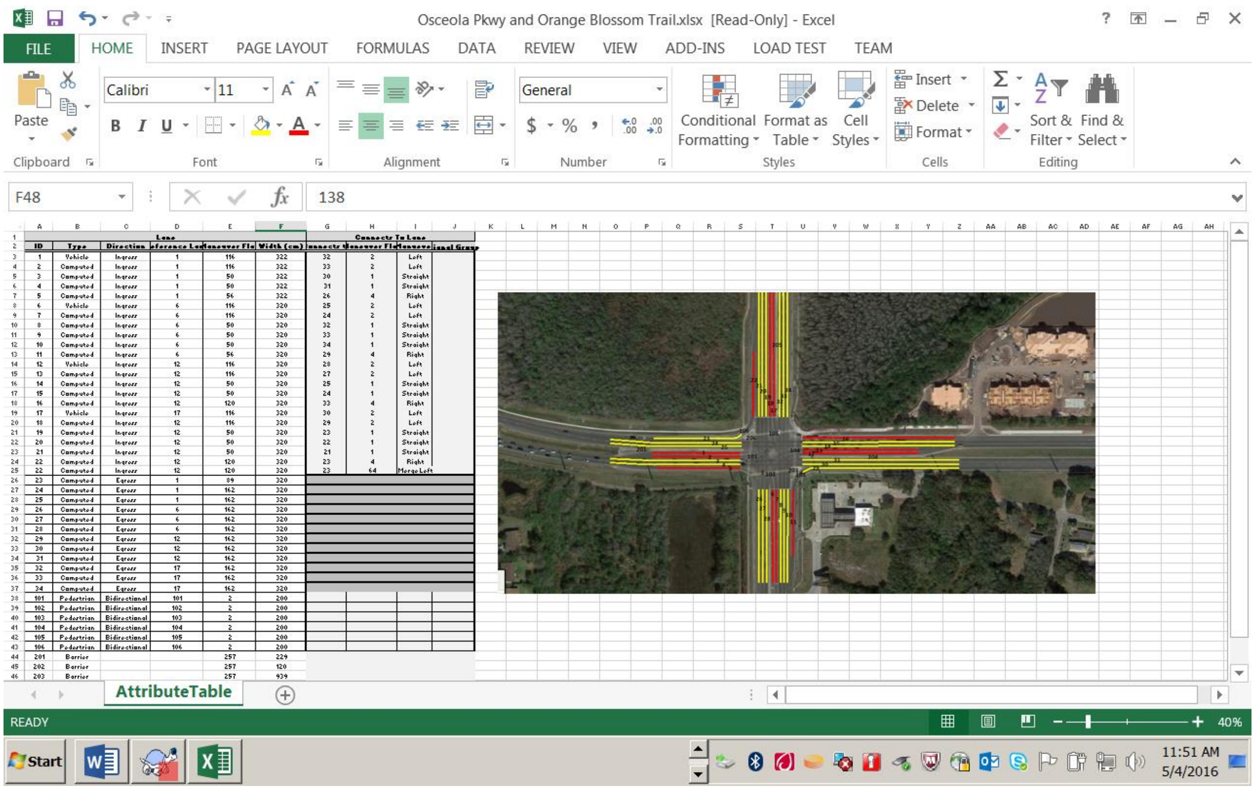
Appendix D: RSU Application Configuration

This section contains the process used to create a MAP file for an intersection.

1) The first thing needed is an install of Google Earth if not already installed
2) In Google Earth, find the intersection to be mapped and determine the reference point for the intersection.
3) Put a pin in the location and set it as the reference point for that intersection.
4) Write down the reference point's latitude and longitude for later use.
5) Create a screen shot of the intersection for lane markup and future reference.
6) In Paint or other image manipulation program, paste the image of the intersection.
7) Determine the west most set of lanes, find the center line of that set of lanes.
8) From the stopping line draw a yellow line for each lane going away from the stopping line. Note each additional yellow line should be going away from the center line, in this case south.
9) Once all ingress lanes have been marked, move counter clockwise to the south most lanes.
10) Find the center line of that set of lanes.
11) From the stopping line draw a yellow line for ingress each lane going away from the stopping line. Note each additional yellow line, draw in each additional lane, should be going away from the center line, in this case east.
12) Repeat from step 9 until all lane sets are exhausted. Note going away from the center line direction will change according to which lane group you are currently mapping. West lane group-> from center line going south, South lane group-> from center line going east, East lane group-> from center line going north, and North lane group-> from center line going west.
13) Now that all ingress lanes are mapped, from the stopping line draw a red line for each egress lane going away from the stopping line. Note each additional red line, draw in each additional lane, should be going away from the center line, in this case south.
14) Once all of the egress lanes are mapped start with the lane mapped in step 11 and label it with the number 1 and label each additional lane in that lane group will receive a number one higher than the previous number, e.g. n+1 where n=0 for lane one.
15) Move along the intersection counter clock wise labeling all of the ingress lanes.
16) Once all of the ingress lanes are labeled, start numbering the egress lanes starting with the next integer greater than the last ingress lane. Note start with the egress lane closest to the center line of that lane group going away from the center line of that lane group.
17) Once all egress lanes are mapped, determine if there are any pedestrian crosswalks present in the image and label them starting with the pedestrian crosswalk closest to the west most lane group going counter clockwise. Start the numbering at the integer101 and monotonically increase until all pedestrian crosswalks have been exhausted.
18) Once all pedestrian lanes are mapped, determine if there are any barriers present in the image and label them starting with the barrier closest to the west most lane group going counter clockwise.

Start the numbering at the integer 201 and monotonically increase until all pedestrian crosswalks have been exhausted.

19) Once the intersection mapping is complete on the image, fill in the LaneAttributes.XLSX spreadsheet with the information from the marked up image.



20) Using the ruler function in Google Earth, determine the widths of each lane, ingress and egress, pedestrian crosswalk and barrier if applicable.

21) For each ingress lane, there should be a matching egress lane and on the ingress lanes row, the connecting lane information will represent the egress lane for that ingress lane.

22) The vehicle (computed and connecting), pedestrian and barrier lanes attributes are determined from the following.

9 Lane Attributes

Motorized Vehicle or Computed Lanes

The attributes for the motorize vehicle and computed lane types relate the allowed navigational maneuvers and other restrictions as follows.

0 (LSB)Lane provides a two-way travel

1 Straight maneuver permitted

2 Left turn maneuver permitted

3 Right turn maneuver permitted

4 Yield

5 No U-turn

6 No turn on red

7 No stopping

8 HOV lane

9 Bus only lane

10 Bus and taxi only lane

11 Shared two-way left turn lane

12 Bike lane

13-15 Reserved

Pedestrian Lanes

The attributes for a pedestrian lane relate the type of crosswalk, bicycle-crossing, or other non-motorized lane as follows.

0 (LSB)Lane provides a two-way travel

1 Pedestrian crosswalk

2 Bicycle crossing

3 Railroad track is present

4-15 Reserved

Barrier Lanes

The attributes for barriers relate the type of barriers being described. A barrier is any object which normal vehicle traffic cannot transverse as follows.

0 (LSB)Median

1 White line

2 Stripped lines

3 Double-stripped lines

4 Traffic cones

5 Construction barrier

6 Traffic channels

7 No curbs

8 Low curbs

9 High curbs

10 HOV, do not cross

11 HOV, entry allowed

12 HOV, exit allowed

13-15 Reserved

Connected Lanes

The connected lane must be of the same type as the described lane. The maneuver code is a bit-mapped value as follows.

0 (LSB)	Straight Ahead	4	Soft Left Turn
1	Left Turn	5	Soft Right Turn
2	Right Turn	6	Merge Left
3	U-Turn	7	Merge Right

10 The logic used to determine the attributes

The lane attributes is a bit-encoded value made up of an unsigned 16-bit integer. Basically that means that there are 16 bits, each bit (if set to a one) means a particular attribute is present, while if set to zero means that the attribute is not present. The bit map is described below:

0 (LSB) Lane provides a two-way travel

1 Straight maneuver permitted

2 Left turn maneuver permitted

3 Right turn maneuver permitted

4 Yield

5 No U-turn

6 No turn on red

7 No stopping

8 HOV lane

9 Bus only lane

10 Bus and taxi only lane

11 Shared two-way left turn lane

12 Bike lane

13-15 Reserved

Bit 0 is the Least Significant Bit, so for example:

If a lane only allows vehicles to proceed straight, or turn right then its attribute code would be:

0000 0000 0000 1010

Or

000A (hex)

Or

10 (decimal)

The attributes are specified in decimal, so for this lane, its attributes would be a value of 10.

If the same lane has a No-Turn-on-Red restriction, the is code would be a value of 74:

0000 0000 0100 1010

Or

004A (hex)

Or

74 (decimal)

23) Once the LaneAttributes.XLSX is completed the actual XML file is created using the following structure.

```
<?xml version="1.0"?>
<J2735.GID.blob>
  <Version>1</Version>
  <IntersectionID></IntersectionID>
  <Elevation>false</Elevation>
  <Resolution>decimeter</Resolution>
  <Geometry>
    <ReferencePoint>
      <Latitude></Latitude>
      <Longitude></Longitude>
    </ReferencePoint>
    <Approach>
    <Lane Number="1">
```

```
  <Type>Vehicle</Type>
  <Attributes></Attributes>
  <Width></Width>
  <Nodes>
    <Node Number="1">
      <Eastern></Eastern>
       <Nothern></Nothern>
    </Node>
    <Node Number="2">
      <Eastern></Eastern>
      <Nothern></Nothern>
    </Node>
  </Nodes>
  <Connections>
    <Connection>
      <LaneNumber></LaneNumber>
      <Maneuver></Maneuver>
      <SignalGroup></SignalGroup>
    </Connection>
  </Connections>
</Lane>
<Lane Number="2">
  <Type>Computed</Type>
  <Attributes></Attributes>
  <Width>359</Width>
  <ReferenceLane>
    <LaneNumber>11</LaneNumber>
    <XOffset>-36</XOffset>
    <YOffset>0</YOffset>
  </ReferenceLane>
  <Connections>
```

```
        <Connection>
          <LaneNumber></LaneNumber>
          <Maneuver>1</Maneuver>
          <SignalGroup>1</SignalGroup>
        </Connection>
        <Connection>
          <LaneNumber></LaneNumber>
          <Maneuver>4</Maneuver>
          <SignalGroup>1</SignalGroup>
        </Connection>
      </Connections>
    </Lane>
    <Lane Number="3">
      <Type>Pedestrian</Type>
      <Attributes></Attributes>
      <Width></Width>
      <Nodes>
        <Node Number="1">
          <Eastern></Eastern>
          <Nothern></Nothern>
        </Node>
        <Node Number="2">
          <Eastern>9</Eastern>
          <Nothern></Nothern>
        </Node>
      </Nodes>
    </Lane>
</Approach>
<Egress>
  <Lane Number="4">
  <Type>Computed</Type>
```

<Width>452</Width>
<ReferenceLane>
<LaneNumber>11</LaneNumber>
<XOffset>45</XOffset>
<YOffset>0</YOffset>
</ReferenceLane>
</Lane>
</Egress>
</Geometry>
</J2735.GID.blob>

11 Summary of all processes used to complete the XML structure above

Intersection ID

The Intersection ID object is used to globally and uniquely define an intersection within a country or region in a 32 bit field. Assignment rules for this value are established elsewhere and may use regional assignment schemas that vary.

Object Format:

Field	**Note**
Object Identifier	0x02
Size	0x04
Intersection ID	Unsigned 32-bit Integer

Example:

```
02 04 00 00 00 01  -- Intersection ID #0001
```

Reference Point

The Reference Point object provides a definitive and precise location of the intersection in the world geodetic system (WGS-84) coordinate system from which short offsets may then be used to define lane node positions using the flat earth projection.

Object Format:

Field	**Note**
Object Identifier	0x03
Size	Number of bytes in the payload

Latitude	Signed 32-bit Integer
Longitude	Signed 32-bit Integer
Elevation	Signed 16-bit Integer, optional

This data object provides the latitude and longitude of the center of the intersection in decimal degrees to one-tenth of a micro-degree where the LSB represents 10^{-7} degrees. When included, the elevation data is provided in decimeters above or below the reference ellipsoid. The elevation value is an asymmetric range of positive and negative values.

The elevation data is only included if indicated in the Message Attributes data object.

Elevation	**Encoding**	**Decimal Values**
Positive Range	0x0000 to 0xEFFF	0 to 61439 decimeters
Negative Range	0xF001 to 0xFFFF	-1 to -4095 decimeters

Examples:

```
03 08 00 00 00 00 00 00 00 00          -- Lat and Long coordinates
03 0A 00 00 00 00 00 00 00 00 00 00    -- Lat, Long, and Elevation data
```

Approach/Egress/Barrier

The Approach/Egress/Barrier data object indicates the beginning of a structure used to relate one or more motor vehicles lanes for an intersection approach or egress description with any associated pedestrian and special purpose lanes or to describe one or more barriers.

Object Format:

Field	**Note**
Object Identifier	0x04
Size	0x01
Direction	Unsigned 8-bit Integer

The enumerated values of Direction data are as follows.

1 Approach

2 Egress

3 Barrier

The lanes described in an approach may appear in any order; however computed lanes must appear after the description of their reference lane. All the lane descriptions must be provided before the next Approach/Egress/Barrier flag is encountered in the GID message.

Examples:

```
04 01 01   -- Beginning of the intersection approach lanes
04 01 02   -- Beginning of the intersection egress lanes
```

Lane

The Lane data object indicates the beginning of a lane description and provides the lane number and lane type information.

Object Format:

Field	Note
Object Identifier	0x05
Size	0x02
Lane Number	Unsigned 8-bit Integer
Lane Type	Unsigned 8-bit Integer

The data objects that describe each lane must be provided before the next Lane data object is encountered in the GID message. The lane description may be comprised of the following set of data objects. The data objects that are included in the lane descriptions are determined by the geometric and navigational flags in the Message Attributes data object. The data objects can be in any order.

Geometric Data	Navigational Data
• Width • Node List • Reference Lane	• Connections

The enumerated values of Lane Type data object are as follows.

1 Motorized Vehicle Lane

2 Computed Lane

3 Pedestrian Lane

4 Special Purpose Lane

Examples:

```
05 02 01 01    -- Lane #1, Motorized Vehicles
05 02 04 03    -- Lane #4, Pedestrian
```

Lane Attributes

The Lane Attributes data object is a bit-mapped value that describes a combination set of characteristics specific to a lane type.

Object Format:

Field	Note
Object Identifier	0x06
Size	0x02
Lane Attributes	Unsigned 16-bit Integer

The Attributes value is obtained by combining (logical OR) the various bit-masks together to produce a single value. A bit set to a logic one indicates that the described attribute is present while a bit set to zero indicates that it is not present.

Motorized Vehicle or Computed Lanes

The attributes for the motorize vehicle and computed lane types relate the allowed navigational maneuvers and other restrictions as follows.

0 (LSB)Lane provides a two-way travel

1 Straight maneuver permitted

2 Left turn maneuver permitted

3 Right turn maneuver permitted

4 Yield

5 No U-turn

6 No turn on red

7 No stopping

8 HOV lane

9 Bus only lane

10 Bus and taxi only lane

11 Shared two-way left turn lane

12 Bike lane

13-15 Reserved

Special Purpose Lanes

The attributes for special purpose lanes relate the allowed navigational maneuvers and other restrictions. These attributes deal with lanes describing trains and transit vehicles that are part of an intersection as follows.

0 (LSB)Lane provides a two-way travel

1 Lane is a railroad track

2 Transit vehicle only lane

3 HOV lane

4 Bus only lane

5 Vehicles entering

6 Vehicles leaving

7-15 Reserved

Pedestrian Lanes

The attributes for a pedestrian lane relate the type of crosswalk, bicycle-crossing, or other non-motorized lane as follows.

0 (LSB)Lane provides a two-way travel

1 Pedestrian crosswalk

2 Bicycle crossing

3 Railroad track is present

4-15 Reserved

Examples:

```
06 02 00 6A   -- Vehicle: Straight and right turns, no turn on red
06 02 00 13   -- Pedestrian: Crosswalk and bicycle crossing, two way travel
```

Barrier Attributes

The Barrier Attributes data object is a bit-mapped value that describes a combination set of characteristics specific to a barrier.

Object Format:

Field	Note
Object Identifier	0x07
Size	0x02
Barrier Attributes	Unsigned 16-bit Integer

The Attributes value is obtained by combining (logical OR) the various bit-masks together to produce a single value. A bit set to a logic one indicates that the described attribute is present while a bit set to zero indicates that it is not present.

The attributes for barriers relate the type of barriers being described. A barrier is any object which normal vehicle traffic cannot transverse as follows.

0 (LSB)Median

1 White line

2 Stripped lines

3 Double-stripped lines

4	Traffic cones
5	Construction barrier
6	Traffic channels
7	No curbs
8	Low curbs
9	High curbs
10	HOV, do not cross
11	HOV, entry allowed
12	HOV, exit allowed
13-15	Reserved

Example:

```
07 02 08 00    -- Barrier: HOV lane, entry allowed
```

Width

The Width data object conveys the width of a lane or barrier in centimeters. The width can be expressed at various levels of granularity; such as an approach, lane, or individual node. Lower levels supersede higher levels.

Object Format:

Field	Note
Object Identifier	0x08
Size	0x02
Lane Attributes	Unsigned 16-bit Integer

Example:

```
08 02 01 90    -- Width is 4 meters (400 centimeters) wide
```

Node List

The Node List data object structure provides the sequence of offset values relative to the intersections reference point to determine the coordinates of individual nodes that together describe the lane centerline. The node offset values are given in absolute distance and can have either a one centimeter or one decimeter resolution as indicated in the Message Attributes data object, bit 3.

Object Format:

Field	Note

Object Identifier	0x09
Size	Number of bytes in the payload
Attributes	Unsigned 8-bit Integer
Nodes	Varies in length

The bits of the node list attributes are as follows. A bit set to a logic one indicates that the described condition is present while a bit set to zero indicates that it is not present.

0 (LSB) Width data is included in this node list.

1 Node data is packed in 12-bit format instead of standard 16-bit.

2-7 Reserved

A node is a spot on the ground. When describing a path, the first node is the one closest to the intersection for the lane or the beginning point in a roadway segment and its offsets. This is typically located on the stop-line for approach lanes and where the outbound lane begins for egress lanes. Subsequent nodes provide points further away along the lanes center line. A lane must always have at least two nodes; however as many nodes as necessary to characterize the lane curvature should be included.

The first node offset values are relative to the intersections reference point. The offset values for subsequent nodes are relative to the previous node position. The Node object has the following structure.

Object Format:

Field	**Note**
Eastern Offset	Signed 16-bit Integer
Northern Offset	Signed 16-bit Integer
Elevation Offset	Signed 16-bit Integer, optional
Width	Unsigned 16-bit Integer, optional

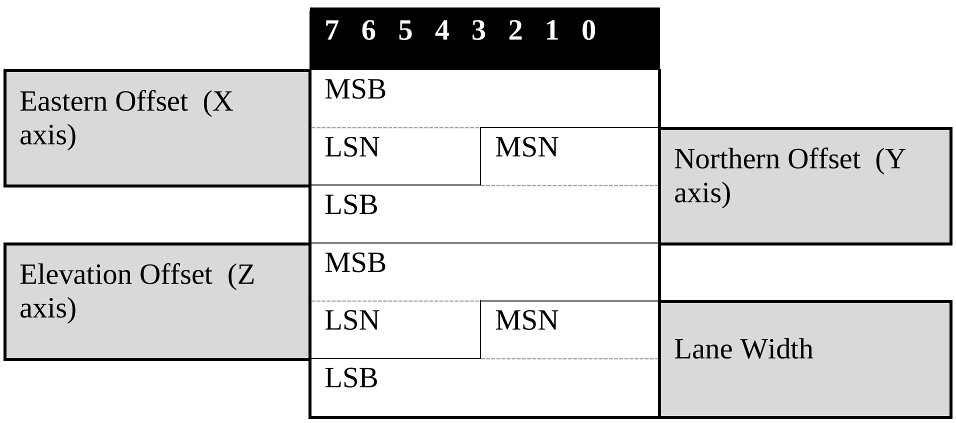
Lateral offset values are referenced using the Cartesian coordinate system on an intersection with a northern orientation; thus eastern offsets are indicated by a positive value while western offsets are negative. Likewise, northern offsets are positive while southern offsets are negative.

Node elevation offsets may also be present in the node list as indicated in the Message Attributes data object, bit 0. Positive elevation values indicate a rise in elevation while negative values indicate a drop in elevation relative to the intersections reference point.

Variable widths along the path may be specified in centimeters when width information is present in the node list as indicated in the node list attributes, bit 0.

The node list can be compacted when byte efficiency is critical. The use of the compacted data format is indicated in the node list attributes, bit 1. The compacted node list encodes all node data as 12-bit integers rather than 16-bit integers. This is useful when there are no node offsets that require the 4 most-significant-bits to be used, thus reducing the overall byte count of the node list by approximately 25% as illustrated in the figure below. The maximum offset distance that can be specified is a function of the maximum value of the node offset and the node offset resolution as indicated in the Message Attributes data object.

	7 6 5 4 3 2 1 0
Eastern Offset (X axis)	MSB
	LSB
Northern Offset (Y axis)	MSB
	LSB
Elevation Offset (Z axis)	MSB
	LSB
Lane Width	MSB
	LSB



Examples:

```
09 05 00 XX XX YY YY          -- X, Y offsets
09 04 02 XX XY YY             -- X, Y offsets, packed
09 07 00 XX XX YY YY ZZ ZZ    -- X, Y, Z offsets
```

Connection

The Connection data object identifies a previously defined lane to which this lane connects through the use of a specified type of maneuver.

Object Format:

Field	Note
Object Identifier	0x0A
Size	0x02
Lane Number	Unsigned 8-bit Integer

Maneuver Code	Unsigned 8-bit Integer
Signal Group	Unsigned 8-bit Integer

The connected lane must be of the same type as the described lane. The maneuver code is a bit-mapped value as follows.

0 (LSB)Straight Ahead 4 Soft Left Turn

1 Left Turn 5 Soft Right Turn

2 Right Turn 6 Merge Left

3 U-Turn 7 Merge Right

Example:

```
0A 02 04 02    -- This lane connects to lane #4, left turn
0A 02 09 01    -- This lane connects to lane #9, straight ahead
```

Reference Lane

The Reference Lane data object identifies the reference lane and lateral offset of the center line that a computed lane parallels. This data object is only present the lane type is a computed lane.

Object Format:

Field	Note
Object Identifier	0x0B
Size	0x03
Reference Lane	Unsigned 8-bit Integer
XOffset	Signed 16-bit Integer
YOffset	Signed 16-bit Integer

The referenced lane number must be previously described prior to being referenced by a computed lane. The orientation is from the perspective of the driver where a negative value is a leftward offset and a positive value is rightward. The offset is independent of any width values and uses the node offset resolution. Offset is measured in centimeters.