Integrated V2I Prototype Application Description

# Description

## Project Objectives

The objective of the Integrated V2I Prototype (IVP) project is to identify, develop, implement, test, document and deploy a roadside prototype system that supports an integrated, interoperable deployment of multiple V2I safety, mobility, and environmental applications.

IVP is an interface system supporting the collection, integration and dissemination of data between infrastructure and vehicles for a wide variety of applications:

* Signal Phase and Timing
* Mapping (Intersections and Road Segments)
* Other Roadside Equipment (i.e. signage, detectors)
* Positioning / Corrections
* Communications (DSRC, cellular)
* Security
* Road Condition and Weather Data

It is not the intention of this document to describe the details of each V2I application itself. Rather, this document is intended to describe the IVP platform architecture capabilities to support these and other applications through an extensible message architecture. This document describes the core features of the IVP architecture, as well as the implementation interfaces to a number of useful external subsystem components to the IVP platform. This document also describes in some detail the interfaces between the listed V2I applications and the core of the IVP platform, to the extent that those applications have been implemented, in the case of several simpler, utility features of the platform, or ported, as is the case with the INFLO and SPAT applications. The details of the interfaces to external subsystem components are described in the IVP Interface Control Document (ICD), while the details of particular V2I applications can be found elsewhere.

## Supported Applications

Figure 2‑1 lists the applications and application groups that the IVP platform is designed to support. Note that while the initial deployment of the IVP platform will only implement a subset of DMA applications, nonetheless the IVP platform has been designed to be readily extended to accommodate any of the listed applications using the techniques outlined in this document.

|  |  |
| --- | --- |
| **Dynamic Mobility Applications**   * + INFLO Speed Harmonization (SPD-HARM)   + INFLO Queue Warning (Q-WARN)   + RESCUME Incident Zone (INC-ZONE) - Low latency comm for V2V; High latency comm for V2I   + FRATIS – High latency communications   **Multi-Modal Intelligent Traffic Signal System**   * Intelligent Traffic Signal System * Transit Signal Priority * Pedestrian Mobility * Freight Signal Priority * Emergency Vehicle Priority   **AERIS Applications**   * Eco-Signal Operations * Eco-Traffic Signal Timing * Eco-Approach and Departure at Signalized Intersections * Eco-Traffic Signal Priority * Connected Eco-Driving * Dynamic Low Emissions Zones * Dynamic Eco-Lanes | **Transit Applications**   * Pedestrian Crossing Warning (PCW)   **V2I Safety Applications**   * Red-Light Violation Warning (RLVW) * Stop Sign Gap Assist (SSGA) * Curve Speed Warning (CSW) * Stop Sign Violation Warning (SSVW) * Railroad Crossing Violation Warning (RCVW) * Spot Weather Information Warning (SWIW) * Oversize Vehicle Warning (OVW) * Reduced Speed Zone Warning (RSZW) – Speed Reduction and Lane Closure Advisories * Reduced Speed Zone Warning (RSZW) – Lane Closure Alerts & Warnings   **Road Weather Connected Vehicle Applications**   * Enhanced Maintenance Decision Support System (MDSS). * Information for Maintenance and Fleet Management Systems. * Weather-Responsive Traffic Management. * Motorist Advisories and Warnings. * Information for Freight Carriers. * Information and Routing Support for Emergency Responders. |
| Source: Battelle | |

Figure 2‑1. Supported V2I Applications

## User Needs

Multiple applications require a local communication and computational/processing platform for:

* Message Handling across Multiple Interfaces
  + Integrating data from multiple sources and compiling messages for delivery to vehicles and drivers and nomadic devices via multiple communication methods
  + Obtaining and aggregating data from multiple vehicles and nomadic devices and delivery to Traffic Management Entity
  + Distribution of TME messages to local vehicles
* Local Infrastructure Based Computation and Processing:
  + e.g. Local computation of safe speeds and safe stopping distances using real time weather and road condition data for crash imminent V2I safety scenario such as Reduced Speed (Work Zone) Warning and Spot Weather Information Warning
  + e.g. Aggregation of vehicle weather data for efficient communication to Traffic Management Entity for Weather Responsive Traffic Management
  + e.g. MMITSS “Intersection level” functions including MAP and SPAT broadcast manager, equipped vehicle tracker, priority request server, and interface to traffic signal controller

# Development Guidance

## Development Prerequisites

* Operating System: Ubuntu 14.04 LTS
* Pre-Installed Packages (to be installed by running ‘sudo apt-get install [package name]’)
  + cmake
  + gcc-4.8
  + g++-4.8
  + libboost1.55-dev
  + libboost-thread1.55-dev
  + libboost-regex1.55-dev
  + libboost-log1.55-dev
  + libboost-program-options1.55-dev
  + libboost1.55-all-dev
  + libxerces-c-dev
  + libcurl4-openssl-dev
  + libsnmp-dev
  + libmysqlclient-dev
  + libjsoncpp-dev
  + uuid-dev
  + libusb-dev libusb-1.0.0-dev
  + libftdi-dev
  + swig
  + liboctave-dev
  + gpsd libgps-dev
  + portaudio19-dev libsndfile-dev
  + libglib2.0-dev libglibmm-2.4-dev
  + libpcre3-dev
  + libsigc++-2.0-dev
  + libxml++2.6-dev
  + libxml2-dev
  + liblzma-dev
* Pre-Installed Packages (to be installed from the included TMX\_OAM/Externals directory
  + Mysql-connector-c\_\_-1.1.3
    - Run the following from your TMX\_OAM/Externals directory
      * tar xzf mysql-connector-c++-1.1.3.tar.gz
      * cd mysql-connector-c++-1.1.3
      * cmake .
      * make
      * sudo make install
      * cd ..
      * rm -rf mysql-connector-c++-1.1.3
  + J2735r41
    - Run the following from your TMX-OAM/Externals directory
      * ./setupAsnJ2735\_r41.sh

## Compilation Instructions

Each project contains its own cmake file. You will need to compile the projects in TMX/Core first. Start with the TmxUtils project, then the TmxApi, followed by the TmxCore. After those are built, you can compile any of the projects from the Plugins directory.

To compile a project

* Change directory to the Debug or Release subfolder for the project (cd Debug)
* Run cmake (cmake ..)
* Run make (make)
  + This will compile the project
* To create a deployment package type make package

## Deployment Instructions

Deployment Instructions can be found in the *IVP Deployment Guide* document.

## Additional Documents

Documents included in the Documents folder:

* *IVP Requirements*
  + This document defines the requirements for the IVP platform and message handlers recommended for development
* *IVP Interface Control Document*
  + This document captures the external interfaces necessary to support the Integrated Vehicle-to-Infrastructure (V2I) Prototype (IVP) Platform and related messages between the platform and a variety of infrastructure devices, including traffic signal controllers, back office systems, mobile devices, etc., organized around the V2I applications supported by the platform.
* *IVP Design*
  + This report documents the System Design Document (SDD) for an Integrated V2I Prototype (IVP) platform that is a key research activity within the DMA program.