GCDC16 Local Message Set

Albin Severinson

September 10, 2015

Contents

Version:	0.1	First draft
	0.2	Revised CAM and added DENM

1 Introduction

This document presents the local message set (LMS) to be used for GCDC16. LMS will be used by the sensor fusion system to generate CAM messages, and by the scenario control models to generate DENM/iGAME messages.

The communication stack includes a vehicle adapter that will receive these messages and use them to create proper CAM/DENM/iGAME messages that will be forwarded to other vehicles. The LMS follows the ETSI specification as closely as possible, but makes some changes to make is possible to create the messages in Simulink.

Different network ports will be used for CAM/DENM/iGAME messages in order to distinguish them.

All data is in network byte order, which is identical to big endian.

2 CAM

CAM messages are created by the vehicle adapter using the data fields present in a local CAM message as detailed below. CAM messages have both mandatory and optional data fields, where the optional data fields are contained in the low frequency container. The container mask is a bit mask that indicates whether this container is present. Note that the optional fields in the local CAM message always are present. However their values are undefined if they are not indicated as present.

Bytes:	Data:	Notes
1	Header	= 2 for CAM
1	Container Mask	
4	GenerationDeltaTime	See D3.2
4	Station Type	See D3.2
4	(opt) Vehicle Role	See D3.2
4	Vehicle Length	See D3.2
4	Vehicle Width	See D3.2
4	Latitude	See D3.2
4	Longitude	See D3.2
4	Semi Major Confidence	See D3.2
4	Semi Minor Confidence	See D3.2
4	Semi Major Orientation	See D3.2
4	Heading	See D3.2
4	Heading confidence 95%	See D3.2
4	Altitude	Not in D3.2?
4	Speed	See D3.2
4	Speed Confidence 95%	See D3.2
4	Yaw Rate	See D3.2
4	Yaw Rate Confidence 95%	See D3.2
4	Longitudinal vehicle acceleration	See D3.2
4	Longitudinal vehicle acceleration confidence 95%	See D3.2

3 DENM

New proposal for DENM local message set. This implementation is more standards compliant. Most of the DENM specification is implemented in the LMS, however the LMS is structured such that it's easy to implement only a subset of the functionality in Simulink. This is done by setting the respective bit masks to indicate that those values aren't used and then fill the data fields with arbitrary data.

The first part of the message, after the header, is a bit mask that indicates which of the optional containers that are present. The containers also start with a bit mask to indicate which of the optional data fields inside that container are used. Data fields marked as unused by the bit mask can have arbitrary values as they are ignored by the communication stack. This also means that every local DENM message has the same size, making it easier to use in Simulink.

Message part:	Bytes:	Data:	Notes:
Header	1	Message ID	= 1 for DENM
Container Mask	1	ContainerMask	
Management Container	1	ManagementMask	
	8	DetectionTime	
	8	ReferenceTime	
	4	(opt) Termination	
	24	EventPosition	
	4	(opt) RelevanceDistance	
	4	(opt) RelevanceTrafficDirection	
	4	(opt) ValidityDuration	
	4	(opt) TransmissionIntervall	
	4	StationType	
(opt)Situation Container	1	SituationMask	
	4	InformationQuality	
	8	CauseCode	
	8	(opt) LinkedCauseCode	
	0	(opt) EventHistory	Not implemented
(opt) Location Container	0	LocationMask	Not implemented
	0	(opt) EventSpeed	Not implemented
	0	(opt) EventPositionheading	Not implemented
	0	Traces	Not implemented
	0	(opt) RoadType	Not implemented
(opt) Alacarte Container	1	AlacarteMask	
	4	(opt) LanePosition	See D3.2
	0	(opt) ImpactReducationContainer	Not implemented
	4	(opt) ExternalTemperature	
	0	(opt) RoadWorksContainerExtended	Not implemented
	4	(opt) PositioningSolution	
	0	(opt) StationaryVehicleContainer	Not implemented

4 DENM (deprecated)

These messages are deprecated. See the section above for the local DENM messages.

DENM messages are sent on-demand. D3.2 describes four different DENM messages, but due to how DENM messages are structured with several data containers in every message, it turns out that there are only two distinct

messages used for GCDC16.

The roadworks message is used to notify the vehicles in scenario one that there are roadworks ahead, and what lane is blocked because of that. The emergency vehicle message is used to notify the vehicles in scenario three that there's an emergency vehicle approaching, and what lane it is requesting.

The LMS for DENM messages follows the specification very loosely as it's not worth the time to implement the full standard when only two distinct messages are used.

The fields marked "dummy field" are presented in the spec but doesn't contain any data, and as such are 0 bytes long.

DENM Message:	Bytes:	Data:	Notes::
Roadworks	1	Message ID	= 38
	8	Reference Time	See D3.2
	0	Event Type	Dummy field
	1	Cause Code	See D3.2
	1	Sub Cause Code	See D3.2
	0	Closed Lanes	Dummy field
	1	Driving Lane Status	See D3.2
Emergency Vehicle	1	Message ID	= 40
	8	Reference Time	See D3.2
	0	Event Type	Dummy field
	1	Cause Code	See D3.2
	1	Sub Cause Code	See D3.2
	1	Lane Position	See D3.2

5 iGAME

The iGAME message set is still under proposal. Details on this set will be presented in a future release of this document. As it looks now the structure is very similar to CAM.

Message part:	Bytes:	Data:	Notes:
Header	1	Message ID	= 10 for iGAME
Container Mask	1	Container mask	
High frequency container		Rear axle location	
		Controller type	
		Response time constant	
		Response time delay	
		Target longitudinal acceleration	
		Time headway	
		Cruise speed	
(opt) Low frequency container		Low frequency mask	
		(opt) Participants ready	
		(opt) Start platoon	
		(opt) End-of-scenario	
MIO		Mio ID	
		Mio Range	
		Mio Bearing	
		Mio Range rate	
Lane		Lane	
Pair ID		Forward ID	
		Backward ID	
		Acknowledgement flag	
Merge		Merge request	
		Safe-to-merge	
		Flag	
		Flag tail	
		Flag head	
Intersection		Platoon ID	
		Distance travelled in CZ	
		Intention	
		Counter	