ALPS ALPINE CV2X AMI Specification

Document Version v1.0.7

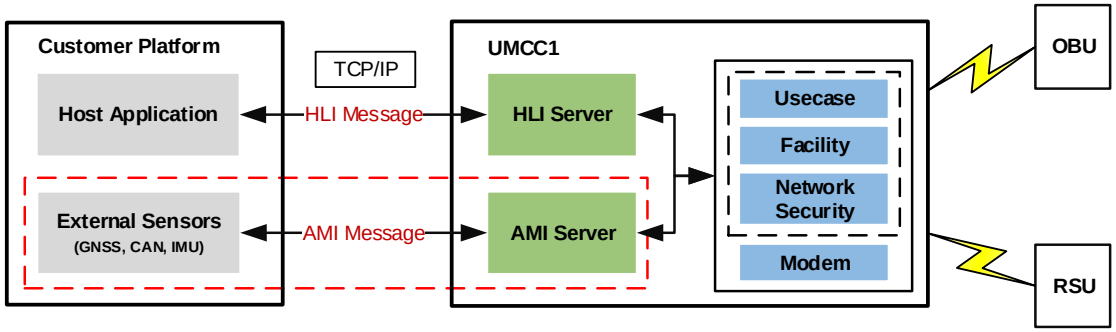
Contents

1. Introduction

The work of CV2X Use case on UMCC1 requires position and vehicle data from external sensors, such as GNSS, CAN, IMU. An interface for transferring data from external sensors is defined as AMI (ALAP Module Interface). The sensor data transmission is implemented by standard Socket interface. This is convenient for communication between UMCC1 and other AP. This document will introduce the specification of AMI Message Protocol and Application Message Protocol for external sensors.

1.1 System Diagram

The following figure shows the concept of the CV2X system. The part inside the red dot line will be described in this document.

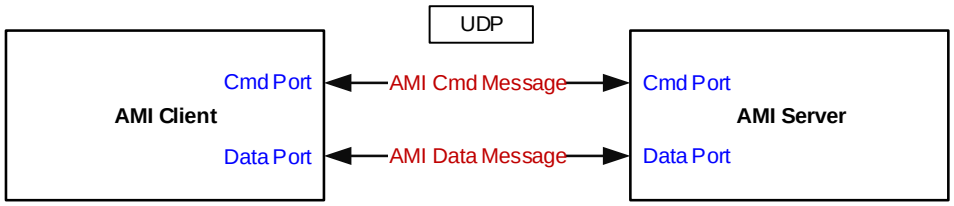


The communication interface between UMCC1 and other AP is defined as UDP socket. UMCC1 receives external sensor data such as GNSS, CAN, IMU sensor. These data will be processed on the Use case and transfer BSM to OBU or RSU.

2. AMI Message Protocol

The AMI message should be transferred between client and server via UDP. The UMCC1 serves as an AMI Server, and the External Sensors serves as an AMI Client.

The AMI support to transfer **Command** and **Data** in two different UDP ports. All the data for AMI message is based on the TLV structure for message compatibility.



The default **Command Port** is 6001, and **Data Port** is 6002.

2.1 AMI Data Type

The AMI Data Type is structured as a series of 8bit bytes packed data. The following table shows the AMI Data Type:

Data Type	Data Length	Meaning
uint8_t	1 Byte	Unsigned 8bit integer.
int8_t	1 Byte	Signed 8bit integer.
uint16_t	2 Byte	Unsigned 16bit integer.
int16_t	2 Byte	Signed 16bit integer.
uint32_t	4 Byte	Unsigned 32bit integer.
int32_t	4 Byte	Signed 32bit integer.
uint64_t	8 Byte	Unsigned 64bit integer.
int64_t	8 Byte	Signed 64bit integer.

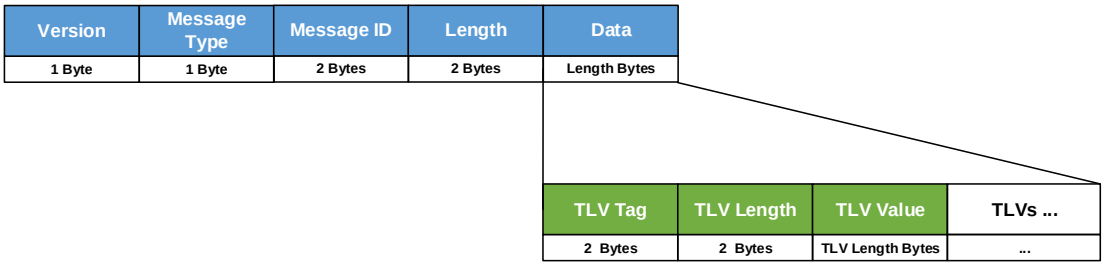
2.2 AMI Message Types

There are two message types include **Command** and **Data**. Types of AMI messages have been defined as shown in the following table:

Message Type	Direction	Type	Meaning
Command Request	Client → Server	Command	Command Request from AMI Client to perform a particular action. And Command Request and Command Response is a pair of sequence.
Command Response	Client ← Server	Command	The Command Response is corresponding to the previous Command Request.
Command Indication	Client ← Server	Command	The Command Indication will be sent to show the status is changed in AMI server.
Data	Client ←→ Server	Data	Both AMI Client and AMI Server could send data to peer device. The Data message has no response.

2.3 AMI Message Format

AMI message with all the message types has the same message format. All the message fields are presented in BIG-ENDIAN. The AMI message consists a number of fields as shown in the following diagram.



The detail fields of AMI message define as below:

Message Field	Data Type	Range	Description
Version	uint8_t	1	The version of AMI message protocol. The value should be version 1.
Message Type	uint8_t	1…4	The message type identifies the message content. 1 - Command Request 2 - Command Response 3 - Command Indication 4 - Data
Message ID	uint16_t	0…65535	This is used to identify messages; this is global scope: 0 - Reserved 1…1023 - Common Message ID 1024…65535 - Application Message ID
Length	uint16_t	1…1400	The length of message payload, excluding message header above.
Data	uint8_t[n]	-	The message payload is structured as a series of bytes packed TLV format. The array length “n” is defined in the above Length field.

The detail of TLV (Tag-Length-Value) format defines as below:

TLV Field	Data Type	Range	Description
TLV Tag	uint16_t	0…65535	This is used to identify TLV Tag: 0…1023 - Common TLV Tag (Global scope) 1024…65535 - Application TLV Tag (Application scope depends on the Application Message ID)
TLV Length	uint16_t	1…1400	The length of TLV Value.
TLV Value	uint8_t[n]	-	The TLV value is the content of TLV Tag. The array length “n” is defined in the above TLV Length field.

2.4 AMI Message Field

AMI message field include AMI Common Message ID, AMI Common TLV etc. defines as below.

2.4.1 AMI Common Message ID

The AMI Common Message ID is used for AMI protocol common Interaction between AMI Client and AMI Server. The AMI Common Message ID is defined in global scope. The detail of Common Message ID defines as below:

Message ID	Message Type	Value	Description
AMI_MSGID_SESSION_ATTACH	Command Request/Response	1	The message is used to bind the AMI Client IP address and cmd/data port to the AMI Server. When multiple AMI Client bind to the AMI Server, the AMI Server could send correspond Command Indication to the specified AMI Client.
AMI_MSGID_SESSION_DETACH	Command Request/Response	2	The message is used to release the AMI Client IP address and port from the AMI Server. Then the AMI Server will not send Command Indication to the AMI Client.
AMI_MSGID_RESULT_INFO	Command Indication	3	The AMI Server will send the message to the specified AMI Client after binding to the AMI Server.
AMI_MSGID_SERVICE_REGISTER	Command Request/Response	4	To register one or multiple interested service to AMI Server by service ID. The service data or indication will be sent to AMI Client after register success.
AMI_MSGID_SERVICE_UNREGISTER	Command Request/Response	5	To unregister one or multiple service to AMI Server by service ID.
AMI_MSGID_KEEPALIVE_PROBE	Command Request/Response	6	To test keepalive between AMI Client and AMI Server, this command should be sent every 1s. When keepalive maximum timeout 5s arrived, AMI Server will delete the session create by AMI_MSGID_SESSION_ATTACH command.

2.4.2 AMI Common TLV Tag

The AMI Common TLV Tag could be used for Common Messages and Application Messages. The AMI Common TLV Tag is defined in global scope. The detail of Common TLV Tag defines as below:

Note:
Parameters with (Optional) in the Description column are optional.

Parameter	TLV Tag	Data Type	Range	Description
ip_address	0	uint8_t[16]	-	IPv4 address in String.
cmd_port	1	uint16_t	0…65535	IPv4 port for transferring Command.
data_port	2	uint16_t	0…65535	IPv4 port for transferring Data.

result_code	3	uint16_t	0…65535	The result code to inform the AMI Client when error happens in AMI Server. Result Code will be defined in the following section, this is defined in global scope: 0…1023 - Common Result Code 1024…65535 - Application Result Code
result_description	4	uint8_t[1…128]	-	The result description in string to inform the AMI Client when error happens in AMI Server. The result string could be defined by user in debug mode.
session_id	5	uint32_t	-	The session ID is generated by AMI Server and return to AMI Client after attach to the AMI Server.
service_id	6	uint16_t	0…65535	The service ID defined in global scope: 0…1023 - Common Service ID 1024…65535 - Application Service ID
channel_type	7	uint8_t	0…1	The AMI physical channel type: 0 - UDP 1 - UART
session_name	8	uint8_t[1…32]	-	The unique name for the session. This is used for attach command via UART channel.

2.4.3 AMI Common Result Code

The AMI Common Result Code will be included in the Command Response or Command Indication to report AMI Client the result status. If the result code is not RESULT_SUCCESS, user could refer to the TLV **result_description** for result detail information if the TLV exists. The Result Code defines as below:

Result Code Name	Value	Description
RESULT_SUCCESS	0	The result is success for the command.
RESULT_FAILURE	1	The result is failure for the command.
RESULT_NOT_SUPPORT	2	The result is not support for the command.
RESULT_TLV_TAG_MISSING	3	The mandatory TLV tag should be missing in the AMI message.
RESULT_TLV_LENGTH_MISMATCH	4	The received TLV length is mismatch from the predefined TLV format.
RESULT_TLV_VALUE_OUT_OF_RANGE	5	The received TLV value is out of range.
RESULT_SERVICE_ALREADY_REGISTERED	6	Register service already registered for one AMI Client. Multiple AMI Clients could register the same service.
RESULT_SERVICE_UNSUPPORT	7	The service ID is not supported.
RESULT_SERVICE_NOT_FOUND	8	The service ID has not been found when do unregister a service.
RESULT_SESSION_NOT_FOUND	9	The session ID has not been found. That means the AMI Client has not been attached.
RESULT_SESSION_ALREADY_EXIST	10	

2.4.4 AMI Common Service ID

The AMI Common Service ID is used to provide internal service for AMI Client. The Service ID defines as below:

Service Name	Value	Description
TBD	-	

2.5 AMI Message Details

The detail AMI Message ID with corresponding TLV structure defines as below:

2.5.1 AMI_MSGID_SESSION_ATTACH

2.5.1.1 AMI_MSGID_SESSION_ATTACH Request

The detail of AMI_MSGID_SESSION_ATTACH Command Request TLV defines as below:

Parameter	TLV Tag	Data Type	Range	Description
channel_type	7	uint8_t	0…1	The AMI physical channel type: 0 - UDP 1 - UART
channel information etc.	-	-	-	The channel information depends on the channel_type.

- For channel_type is UDP, the request is constructed as below:

Parameter	TLV Tag	Data Type	Range	Description
channel_type	7	uint8_t	0	The AMI physical channel type: 0 - UDP
ip_address	0	uint8_t[16]	-	The AMI Client ready to bind IPv4 address in String.
cmd_port	1	uint16_t	0…65535	The AMI Client ready to bind Ipv4 port for transferring Command.
data_port	2	uint16_t	0…65535	The AMI Client ready to bind Ipv4 port for transferring Data.

The message is used to bind the AMI Client IP address and cmd/data port to the AMI Server. When multiple AMI Client bind to the AMI Server, the AMI Server could send correspond Command Indication to the specified AMI Client.

- For channel_type is UART, the request is constructed as below:

Parameter	TLV Tag	Data Type	Range	Description
channel_type	7	uint8_t	1	The AMI physical channel type: 1 - UART
session_name	8	uint8_t[1…32]	-	The unique name for the session. This is used for attach command via UART channel.

2.5.1.2 AMI_MSGID_SESSION_ATTACH Response

The detail of AMI_MSGID_SESSION_ATTACH Command Response TLV defines as below:

Parameter	TLV Tag	Data Type	Range	Description
result_code	3	uint16_t	0…65535	The result code to inform the AMI Client when error happens in AMI Server. Result Code will be defined in the following section, this is defined in global scope: 0…1023 - Common Result Code 1024…65535 - Application Result Code
result_description	4	uint8_t[1…128]	-	The result description in string to inform the AMI Client when error happens in AMI Server. The result string could be defined by user in debug mode. (Optional)
session_id	5	uint32_t	-	The session ID is generated by AMI Server and return to AMI Client after attach to the AMI Server. (Optional)

2.5.2 AMI_MSGID_SESSION_DETACH

2.5.2.1 AMI_MSGID_SESSION_DETACH Request

The detail of AMI_MSGID_SESSION_DETACH Command Request TLV defines as below:

Parameter	TLV Tag	Data Type	Range	Description
session_id	5	uint32_t	-	The session ID is generated by AMI Server and return to AMI Client after attach to the AMI Server.

The message is used to release the AMI Client IP address and port from the AMI Server. Then the AMI Server will not send Command Indication to the AMI Client.

2.5.2.2 AMI_MSGID_SESSION_DETACH Response

The detail of AMI_MSGID_SESSION_DETACH Command Response TLV defines as below:

Parameter	TLV Tag	Data Type	Range	Description

result_code	3	uint16_t	0…65535	The result code to inform the AMI Client when error happens in AMI Server. Result Code will be defined in the following section, this is defined in global scope: 0…1023 - Common Result Code 1024…65535 – Application Result Code
result_description	4	uint8_t[1…128]	-	The result description in string to inform the AMI Client when error happens in AMI Server. The result string could be defined by user in debug mode. (Optional)
session_id	5	uint32_t	-	The session ID is generated by AMI Server and return to AMI Client after attach to the AMI Server.

2.5.3 AMI_MSGID_RESULT_INFO

2.5.3.1 AMI_MSGID_RESULT_INFO Indication

The detail of AMI_MSGID_RESULT_INFO Command Indication TLV defines as below:

Parameter	TLV Tag	Data Type	Range	Description
result_code	3	uint16_t	0…65535	The result code to inform the AMI Client when error happens in AMI Server. Result Code will be defined in the following section: 0…1023 - Common Result Code 1024…65535 - Application Result Code
result_description	4	uint8_t[1…128]	-	The result description in string to inform the AMI Client when error happens in AMI Server. The result string could be defined by user in debug mode. (Optional)
session_id	5	uint32_t	-	The session ID is generated by AMI Server and return to AMI Client after attach to the AMI Server.

2.5.4 AMI_MSGID_SERVICE_REGISTER

2.5.4.1 AMI_MSGID_SERVICE_REGISTER Request

The detail of AMI_MSGID_SERVICE_REGISTER Command Request TLV defines as below:

Parameter	TLV Tag	Data Type	Range	Description
session_id	5	uint32_t	-	The session ID is generated by AMI Server and return to AMI Client after attach to the AMI Server.
service_id	6	uint16_t	0…65535	The service ID defined in global scope: 0…1023 - Common Service ID 1024…65535 - Application Service ID

2.5.4.2 AMI_MSGID_SERVICE_REGISTER Response

The detail of AMI_MSGID_SERVICE_REGISTER Command Response TLV defines as below:

Parameter	TLV Tag	Data Type	Range	Description
result_code	3	uint16_t	0…65535	The result code to inform the AMI Client when error happens in AMI Server. Result Code will be defined in the following section, this is defined in global scope: 0…1023 - Common Result Code 1024…65535 - Application Result Code
result_description	4	uint8_t[1…128]	-	The result description in string to inform the AMI Client when error happens in AMI Server. The result string could be defined by user in debug mode. (Optional)
session_id	5	uint32_t	-	The session ID is generated by AMI Server and return to AMI Client after attach to the AMI Server.

2.5.5 AMI_MSGID_SERVICE_UNREGISTER

2.5.5.1 AMI_MSGID_SERVICE_UNREGISTER Request

The detail of AMI_MSGID_SERVICE_UNREGISTER Command Request TLV defines as below:

Parameter	TLV Tag	Data Type	Range	Description

session_id	5	uint32_t	-	The session ID is generated by AMI Server and return to AMI Client after attach to the AMI Server.
Service_id	6	uint16_t	0…65535	The service ID defined in global scope: 0…1023 - Common Service ID 1024…65535 - Application Service ID

2.5.5.2 AMI_MSGID_SERVICE_UNREGISTER Response

The detail of AMI_MSGID_SERVICE_UNREGISTER Command Response TLV defines as below:

Parameter	TLV Tag	Data Type	Range	Description
result_code	3	uint16_t	0…65535	The result code to inform the AMI Client when error happens in AMI Server. Result Code will be defined in the following section, this is defined in global scope: 0…1023 - Common Result Code 1024…65535 - Application Result Code
result_description	4	uint8_t[1…128]	-	The result description in string to inform the AMI Client when error happens in AMI Server. The result string could be defined by user in debug mode. (Optional)
session_id	5	uint32_t	-	The session ID is generated by AMI Server and return to AMI Client after attach to the AMI Server.

2.5.6 AMI_MSGID_KEEPALIVE_PROBE

2.5.6.1 AMI_MSGID_KEEPALIVE_PROBE Request

The detail of AMI_MSGID_KEEPALIVE_PROBE Command Request TLV defines as below:

Parameter	TLV Tag	Data Type	Range	Description
session_id	5	uint32_t	-	The session ID is generated by AMI Server and return to AMI Client after attach to the AMI Server.

2.5.6.2 AMI_MSGID_KEEPALIVE_PROBE Response

The detail of AMI_MSGID_KEEPALIVE_PROBE Command Response TLV defines as below:

Parameter	TLV Tag	Data Type	Range	Description
result_code	3	uint16_t	0…65535	The result code to inform the AMI Client when error happens in AMI Server. Result Code will be defined in the following section, this is defined in global scope: 0…1023 - Common Result Code 1024…65535 - Application Result Code
result_description	4	uint8_t[1…128]	-	The result description in string to inform the AMI Client when error happens in AMI Server. The result string could be defined by user in debug mode. (Optional)
session_id	5	uint32_t	-	The session ID is generated by AMI Server and return to AMI Client after attach to the AMI Server.

3. AMI Application Protocol

The Sensor Application called AMI Client is used to do GNSS, CAN and IMU sensor collection and build several AMI messages. Then AMI Client will transfer the AMI message to AMI Server.

For CV2X Use case, there are three sensor type include GNSS, CAN and IMU sensor data need to be collected. The senser data should be transferred in AMI Message Type with Data.

3.1 AMI Application Message Field

Application message field include Application Message ID, Application TLV etc. defines as below.

3.1.1 AMI Application Message ID

The Application Message ID is used for Application protocol interaction between AMI Client and AMI Server. The Application Message ID is defined in global scope. The detail of Application Message ID defines as below:

Message ID	Message Type	Value	Description
AMI_MSGID_GNSS_DATA	Data (Client Tx)	1024	The message is used to send GNSS data which read from GNSS module to the AMI Server.
AMI_MSGID_CAN_DATA	Data (Client Tx)	1025	The message is used to send CAN data which read from Vehicle to the AMI Server.
AMI_MSGID_IMU_DATA	Data (Client Tx)	1026	The message is used to send IMU data which read from IMU module to the AMI Server.
AMI_MSGID_GNSS_RTCM_DATA	Data (Server Tx)	1027	The message is used to send GNSS RTCM data which read from RTK server to the AMI Client.
AMI_MSGID_GNSS_RESULT_INFO	Command Indication	1028	The AMI Server will send the message to the specified AMI Client after registering GNSS service to AMI Server.
AMI_MSGID_CAN_RESULT_INFO	Command Indication	1029	The AMI Server will send the message to the specified AMI Client after registering CAN service to AMI Server.
AMI_MSGID_IMU_RESULT_INFO	Command Indication	1030	The AMI Server will send the message to the specified AMI Client after registering IMU service to AMI Server.
AMI_MSGID_PC5_PROFILE_CONFIG	Command Request/Response	1031	The PC5 Profile could be selected by the message. After configuration, the UMCC1 need to be reboot.
AMI_MSGID_MODULE_STATUS_INFO	Command Indication	1032	The AMI Server will report module status to the specified AMI Client after registering MODULE_STATUS service to AMI Server.
AMI_MSGID_VEHICLE_EXT_DATA	Data (Client Tx)	1033	The message is used to send vehicle extension data which read from extension module to the AMI Server.
AMI_MSGID_VEHICLE_EXT_RESULT_INFO	Command Indication	1034	The AMI Server will send the message to the specified AMI Client after registering VEHICLE_EXT service to AMI Server.

3.1.2 AMI Application Service ID

The AMI Application Service ID is used to provide application service for AMI Client. The Service ID defines as below:

Service Name	Value	Description
AMI_SERVICE_GNSS	1024	The service is used to provide GNSS service and transfer AMI_MSGID_GNSS_RESULT_INFO indication to AMI Client when error happened.
AMI_SERVICE_CAN	1025	The service is used to provide CAN service and transfer AMI_MSGID_CAN_RESULT_INFO indication to AMI Client when error happened.
AMI_SERVICE_IMU	1026	The service is used to provide IMU service and transfer AMI_MSGID_IMU_RESULT_INFO indication to AMI Client when error happened.
AMI_SERVICE_MODULE_STATUS	1027	The service is used to provide MODULE_STATUS service and transfer AMI_MSGID_MODULE_STATUS_INFO indication to AMI Client when module status changed.
AMI_SERVICE_VEHICLE_EXT	1028	The service is used to provide VEHICLE_EXT service and transfer AMI_MSGID_VEHICLE_EXT_RESULT_INFO indication to AMI Client when error happened.
AMI_SERVICE_GNSS_RTCM	1029	The service is used to provide GNSS_RTCM service and transfer AMI_MSGID_GNSS_RTCM_DATA to AMI Client.

3.1.3 AMI Application Structure

The Application Structure defines as below:

3.1.3.1 utc_time_t

The utc_time_t structure defines as below:

Parameter	Data Type	Range	Description
year	uint16_t	1900…2099	The year.
month	uint8_t	1…12	The month
day	uint8_t	1…31	The day
hour	uint8_t	0…23	The hour
minute	uint8_t	0…59	The minute
second	uint8_t	0…59	The second
millisecond	uint16_t	0…999	The millisecond

3.2 AMI Application Message Details

The detail Application Message ID with corresponding TLV structure defines as below:

3.2.1 AMI_MSGID_GNSS_DATA

The detail of AMI_MSGID_GNSS_DATA Data TLV defines as below:

Note:
When parameters is_vaild is 1, parameters with (Optional1) in the Description column are required; When parameters is_vaild is 0, parameters with (Optional1) in the Description column are optional.

Parameter	TLV Tag	Data Type	Range	Unit	Description
altitude	1024	uint16_t	0x0000…0xEFFF 0xF001…0xFFFF 0xF000	0.1 meter	Global Positioning System Fix Data: altitude (unit: meter) 0x0000…0xEFFF - 0…6143.9 0xF001…0xFFFF - -409.5…-0.1 0xF000 - Unknown
gps_state	1025	uint8_t	-	-	Global Positioning System Fix Data: gps_state
satellite_num	1026	uint8_t	0…12	-	DOP value of GNSS and effective satellite information: satellite_num
pdop	1027	uint16_t	0…999	0.1	DOP value of GNSS and effective satellite information: pdop 0.0…99.9
hdop	1028	uint16_t	0…999	0.1	DOP value of GNSS and effective satellite information: hdop 0.0…99.9
vdop	1029	uint16_t	0…999	0.1	DOP value of GNSS and effective satellite information: vdop 0.0…99.9
semi_major	1030	uint16_t	0…999	0.1 meter	GNSS Pseudorange Error Statistics: semi_major (unit: meter) 0.0…99.9
semi_minor	1031	uint16_t	0…999	0.1 meter	GNSS Pseudorange Error Statistics: semi_minor (unit: meter) 0.0…99.9
orientation	1032	uint16_t	0…35999 36000	0.01 degree	GNSS Pseudorange Error Statistics: orientation (unit: degree) 0…35999 - 0.00…359.99 36000 - Unavailable
is_vaild	1033	uint8_t	0…1	-	Recommended Minimum Specific GNSS Data: is_vaild False(0) - invaild: fill in the default value for Optional1 items. True(1) - vaild: the Optional1 items are mandatory.
utc_time	1034	-	-	-	Recommended Minimum Specific GNSS Data: utc time Refer to utc_time_t for details. (Optional1)
zone	1035	int8_t	-11…12	-	Recommended Minimum Specific GNSS Data: offset Distinguish time zones (Optional1)

latitude	1036	int32_t	-900000000… 900000000 900000001	0.1 micro degree	Recommended Minimum Specific GNSS Data: latitude (unit: degree) Providing a range of plus-minus 90 degrees -900000000…900000000 - -90.0000000…90.0000000 900000001 – Unavailable (Optional1)
longitude	1037	int32_t	-1800000000… 1800000000 1800000001	0.1 micro degree	Recommended Minimum Specific GNSS Data: longitude (unit: degree) Providing a range of plus-minus 180 degrees: -1800000000…1800000000 - -180.0000000…180.0000000 1800000001 - Unavailable (Optional1)
speed	1038	uint16_t	0…20000	0.01 m/s	Recommended Minimum Specific GNSS Data: speed (unit: m/s) 0.00…200.00 (Optional1)
heading	1039	uint16_t	0…35999 36000	0.01 degree	Recommended Minimum Specific GNSS Data: heading (unit: degree) 0…35999 - 0.00…359.99 36000 - Unavailable (Optional1)
leap_second	1040	uint16_t	-	-	The GNSS Data: leap second (Optional)

3.2.2 AMI_MSGID_CAN_DATA

The detail of AMI_MSGID_CAN_DATA Data TLV defines as below:

Parameter	TLV Tag	Data Type	Range	Unit	Description
transmission	1024	uint8_t	0…4	-	The current vehicle gear status from CAN bus: 0 - Unavailable 1 - Neutral 2 - Park 3 - Forward Gears 4 - Reverse Gears
wheelbrake	1025	uint8_t	0…2	-	The current Brake pedal status from CAN bus: 0 - Unavailable 1 - OFF 2 - ON
TCS	1026	uint8_t	0…3	-	The Traction Control Status from CAN bus: 0 - Unavailable 1 - OFF 2 - ON 3 - Engaged

ABS	1027	uint8_t	0…3	-	The Anti-Lock Brake Status from CAN bus: 0 - Unavailable 1 - OFF 2 - ON 3 - Engaged
SCS	1028	uint8_t	0…3	-	The Stability Control Status from CAN bus: 0 - Unavailable 1 - OFF 2 - ON 3 - Engaged
low_beam_head_lights	1029	uint8_t	0…3	-	The status of low beam head lights from CAN bus: 0 - Unavailable 1 - OFF 2 - ON 3 – Error (Optional)
high_beam_head_lights	1030	uint8_t	0…3	-	The status of high beam head lights from CAN bus: 0 - Unavailable 1 - OFF 2 - ON 3 - Error (Optional)
left_turn_signal	1031	uint8_t	0…3	-	The status of left turn signal lights from CAN bus: 0 - Unavailable 1 - OFF 2 - ON 3 - Error
right_turn_signal	1032	uint8_t	0…3	-	The status of right turn signal lights from CAN bus: 0 - Unavailable 1 - OFF 2 - ON 3 - Error
hazard_signal	1033	uint8_t	0…3	-	The status of hazard signal lights from CAN bus: 0 - Unavailable 1 - OFF 2 - ON 3 - Error
automatic_light_control	1034	uint8_t	0…3	-	The status of automatic light control from CAN bus: 0 - Unavailable 1 - OFF 2 - ON 3 – Error (Optional)

daytime_running_lights	1035	uint8_t	0…3	-	The status of daytime running lights from CAN bus: 0 - Unavailable 1 - OFF 2 - ON 3 – Error (Optional)
fog_light	1036	uint8_t	0…3	-	The status of fog light from CAN bus: 0 - Unavailable 1 - OFF 2 - ON 3 – Error (Optional)
parking_lights	1037	uint8_t	0…3	-	The status of parking lights from CAN bus: 0 - Unavailable 1 - OFF 2 - ON 3 – Error (Optional)
utc_time	1038	-	-	-	utc time: Refer to utc_time_t for details. (Optional)

3.2.3 AMI_MSGID_IMU_DATA

The detail of AMI_MSGID_IMU_DATA Data TLV defines as below:

Parameter	TLV Tag	Data Type	Range	Unit	Description
lateral	1024	int16_t	-19620…19620	0.001 m/s^2	accelerations of vehicle lateral (unit:m/s^2) -19.620…19.620
longitudinal	1025	int16_t	-19620…19620	0.001 m/s^2	accelerations of vehicle longitudinal (unit:m/s^2) -19.620…19.620
vertical	1026	int16_t	-19620…19620	0.001 m/s^2	accelerations of vehicle Vertical (unit:m/s^2) -19.620…19.620
roll	1027	int32_t	-300000…300000	0.001 dps	rotation around the front-to-back axis (unit:dps) Positive number is clockwise. -300.000…300.000 (Optional)
pitch	1028	int32_t	-300000…300000	0.001 dps	rotation around the side-to-side axis (unit:dps) Positive number is clockwise. -300.000…300.000 (Optional)
yaw	1029	int32_t	-300000…300000	0.001 dps	rotation around the vertical axis (unit:dps) Positive number is clockwise. -300.000…300.000

utc_time	1030	-	-	-	utc time: Refer to utc_time_t for details. (Optional)

3.2.4 AMI_MSGID_GNSS_RTCM_DATA

The detail of AMI_MSGID_GNSS_RTCM_DATA Data TLV defines as below:

Parameter	TLV Tag	Data Type	Range	Description
session_id	5	uint32_t	-	The session ID is generated by AMI Server and return to AMI Client after attach to the AMI Server.
rtcm_raw_data	1024	int8_t[1…1400]	1…1400	RTCM raw data from RTK server.

3.2.5 AMI_MSGID_GNSS_RESULT_INFO

3.2.5.1 AMI_MSGID_GNSS_RESULT_INFO Indication

The detail of AMI_MSGID_GNSS_RESULT_INFO indication TLV defines as below:

Parameter	TLV Tag	Data Type	Range	Description
result_code	3	uint16_t	0…65535	The result code to inform the AMI Client when error happens in AMI Server. Result Code will be defined in the following section, this is defined in global scope: 0…1023 - Common Result Code 1024…65535 - Application Result Code
result_description	4	uint8_t[1…128]	-	The result description in string to inform the AMI Client when error happens in AMI Server. The result string could be defined by user in debug mode. (Optional)
session_id	5	uint32_t	-	The session ID is generated by AMI Server and return to AMI Client after attach to the AMI Server.

3.2.6 AMI_MSGID_CAN_RESULT_INFO

3.2.6.1 AMI_MSGID_CAN_RESULT_INFO Indication

The detail of AMI_MSGID_CAN_RESULT_INFO indication TLV defines as below:

Parameter	TLV Tag	Data Type	Range	Description
result_code	3	uint16_t	0…65535	The result code to inform the AMI Client when error happens in AMI Server. Result Code will be defined in the following section, this is defined in global scope: 0…1023 - Common Result Code 1024…65535 - Application Result Code
result_description	4	uint8_t[1…128]	-	The result description in string to inform the AMI Client when error happens in AMI Server. The result string could be defined by user in debug mode. (Optional)
session_id	5	uint32_t	-	The session ID is generated by AMI Server and return to AMI Client after attach to the AMI Server.

3.2.7 AMI_MSGID_IMU_RESULT_INFO

3.2.7.1 AMI_MSGID_IMU_RESULT_INFO Indication

The detail of AMI_MSGID_IMU_RESULT_INFO indication TLV defines as below:

Parameter	TLV Tag	Data Type	Range	Description
result_code	3	uint16_t	0…65535	The result code to inform the AMI Client when error happens in AMI Server. Result Code will be defined in the following section, this is defined in global scope: 0…1023 - Common Result Code 1024…65535 - Application Result Code
result_description	4	uint8_t[1…128]	-	The result description in string to inform the AMI Client when error happens in AMI Server. The result string could be defined by user in debug mode. (Optional)

session_id	5	uint32_t	-	The session ID is generated by AMI Server and return to AMI Client after attach to the AMI Server.

3.2.8 AMI_ PC5_PROFILE_CONFIG

3.2.8.1 AMI_MSGID_PC5_PROFILE_CONFIG Request

The detail of AMI_MSGID_PC5_PROFILE_CONFIG Command Request TLV defines as below:

Parameter	TLV Tag	Data Type	Range	Description
session_id	5	uint32_t	-	The session ID is generated by AMI Server and return to AMI Client after attach to the AMI Server.
pc5_profile_id	1024	uint8_t	0…1	PC5 profile ID: 0 - 2019 4-layer PC5 profile 1 - 2020 4-layer PC5 profile

3.2.8.2 AMI_MSGID_PC5_PROFILE_CONFIG Response

The detail of AMI_MSGID_PC5_PROFILE_CONFIG indication TLV defines as below:

Parameter	TLV Tag	Data Type	Range	Description
result_code	3	uint16_t	0…65535	The result code to inform the AMI Client when error happens in AMI Server. Result Code will be defined in the following section, this is defined in global scope: 0…1023 - Common Result Code 1024…65535 - Application Result Code
result_description	4	uint8_t[1…128]	-	The result description in string to inform the AMI Client when error happens in AMI Server. The result string could be defined by user in debug mode. (Optional)
session_id	5	uint32_t	-	The session ID is generated by AMI Server and return to AMI Client after attach to the AMI Server.

3.2.9 AMI_MSGID_MODULE_STATUS_INFO

3.2.9.1 AMI_MSGID_MODULE_STATUS_INFO Indication

The detail of AMI_MSGID_MODULE_STATUS_INFO Command Indication TLV defines as below:

Parameter	TLV Tag	Data Type	Range	Description
module_status	1024	uint16_t	0…65535	TBD.
session_id	5	uint32_t	-	The session ID is generated by AMI Server and return to AMI Client after attach to the AMI Server.

3.2.10 AMI_MSGID_VEHICLE_EXT_DATA

The detail of AMI_MSGID_VEHICLE_EXT Data TLV defines as below:

Parameter	TLV Tag	Data Type	Range	Unit	Description
response_type	1024	uint8_t	0…6	-	Driving state of vehicle extended information: 0 - Unavailable 1 - Emergency 2 - Non_emergency 3 - Pursuit 4 - Stationary 5 - Slow_moving 6 - Stop_and_go_movement (Optional)
siren_use	1025	uint8_t	0…2	-	Siren of vehicle extended information: 0 - Unavailable 1 - Not_in_use 2 - In_use (Optional)
lights_use	1026	uint8_t	0…7	-	Indicator lights for vehicle extended information: 0 - Unavailable 1 - Not_in_use 2 - In_use 3 - Yellow_caution_lights 4 - Schoold_bus_lights 5 - Arrow_signs_active 6 - Slow_moving_vehicle 7 - Freq_stops (Optional)
utc_time	1027	-	-	-	utc time: Refer to utc_time_t for details. (Optional)

3.2.11 AMI_MSGID_VEHICLE_EXT_RESULT_INFO

3.2.11.1 AMI_MSGID_VEHICLE_EXT_RESULT_INFO Indication

The detail of AMI_MSGID_VEHICLE_EXT_RESULT_INFO indication TLV defines as below:

Parameter	TLV Tag	Data Type	Range	Description
result_code	3	uint16_t	0…65535	The result code to inform the AMI Client when error happens in AMI Server. Result Code will be defined in the following section, this is defined in global scope: 0…1023 - Common Result Code 1024…65535 - Application Result Code
result_description	4	uint8_t[1…128]	-	The result description in string to inform the AMI Client when error happens in AMI Server. The result string could be defined by user in debug mode. (Optional)
session_id	5	uint32_t	-	The session ID is generated by AMI Server and return to AMI Client after attach to the AMI Server.

4. Appendix

4.1 Example Data of AMI Message Packet

We take a Sensor Data Message as an example.

4.1.1 AMI_MSGID_GNSS_DATA Sample

The raw data in hex for AMI_MSGID_GNSS_DATA Message is below:

01 04 **04 00** 00 67 **04 00** 00 02 **00 00** 04 01 **00 01** 08 **04 02** 00 01 **00** 04 03 **00 02** 00 00 **04 04** 00 02 **00 00** 04 05 **00 02** 00 00 **04 06** 00 02 **00 00** 04 07 **00 02** 00 00 **04 08** 00 02 **00 00** 04 09 **00 01** 01 **04 0A** 00 09 **07 E5 02 07 0A 01 1E 00 C8** 04 0B **00 01** 00 **04 0C** 00 04 **12 84 65 B9** 04 0D **00 04** 48 5C 2B 83 **04 0E** 00 02 **00 00** 04 0F **00 02** 00 00

The message field explanation of raw data is below:

Message Field		Data Type	Raw Data	Description
Version		uint8_t	**01**	The version **0x01**
Message Type		uint8_t	04	The message type is 0x04 (Data)
Message ID		uint16_t	**04 00**	Application Message ID is **0x0400 (GNSS Data)**
Length		uint16_t	00 67	The length of message payload is 0x0067 (All the TLV data length is 103)
Data				The following TLV data
altitude	Tag	uint16_t	**04 00**	Tag is **0x0400 (1024)**
	Length	uint16_t	00 02	Length is 0x0002
	Value	uint8_t[2]	**00 00**	Value is **0x0000 (0.0 meter)**
gps_state	Tag	uint16_t	04 01	Tag is 0x0401 (1025)
	Length	uint16_t	**00 01**	Length is **0x0001**
	Value	uint8_t[1]	08	Value is 0x08
satellite_num	Tag	uint16_t	**04 02**	Tag is **0x0402 (1026)**
	Length	uint16_t	00 01	Length is 0x0001
	Value	uint8_t[1]	**00**	Value is **0x00**
pdop	Tag	uint16_t	04 03	Tag is 0x0403 (1027)
	Length	uint16_t	**00 02**	Length is **0x0002**
	Value	uint8_t[2]	00 00	Value is 0x0000 (0.0)
hdop	Tag	uint16_t	**04 04**	Tag is **0x0404 (1028)**
	Length	uint16_t	00 02	Length is 0x0002
	Value	uint8_t[2]	**00 00**	Value is **0x0000 (0.0)**
vdop	Tag	uint16_t	04 05	Tag is 0x0405 (1029)
	Length	uint16_t	**00 02**	Length is **0x0002**
	Value	uint8_t[2]	00 00	Value is 0x0000 (0.0)
semi_major	Tag	uint16_t	**04 06**	Tag is **0x0406 (1030)**
	Length	uint16_t	00 02	Length is 0x0002
	Value	uint8_t[2]	**00 00**	Value is **0x0000 (0.0 meter)**
semi_minor	Tag	uint16_t	04 07	Tag is 0x0407 (1031)
	Length	uint16_t	**00 02**	Length is **0x0002**
	Value	uint8_t[2]	00 00	Value is 0x0000 (0.0 meter)
orientation	Tag	uint16_t	**04 08**	Tag is **0x0408 (1032)**
	Length	uint16_t	00 02	Length is 0x0002
	Value	uint8_t[2]	**00 00**	Value is **0x0000 (0.00 degree)**
is_valid	Tag	uint16_t	04 09	Tag is 0x0409 (1033)

	Length	uint16_t	**00 01**	Length is **0x0001**
	Value	uint8_t[1]	01	Value is 0x01
utc_time	Tag	uint16_t	**04 0A**	Tag is **0x040A (1034)**
	Length	uint16_t	00 09	Length is 0x0009
	Value	uint8_t[9]	**07 E5 02 07 0A 01 1E 00 C8**	Value is utc_time_t structure: **year - 0x07E5 (2021)** **month - 0x02 (2)** **day - 0x07 (7)** **hour - 0x0A (10)** **minute - 0x01 (1)** **second - 0x1E (30)** **millisecond - 0x00C8 (200)**
zone	Tag	uint16_t	04 0B	Tag is 0x040B (1035)
	Length	uint16_t	**00 01**	Length is **0x0001**
	Value	uint8_t[1]	00	Value is 0x00
latitude	Tag	uint16_t	**04 0C**	Tag is **0x040C (1036)**
	Length	uint16_t	00 04	Length is 0x0004
	Value	uint8_t[4]	**12 84 65 B9**	Value is **0x128465B9 (31.0666681 degree)**
longitude	Tag	uint16_t	04 0D	Tag is 0x040D (1037)
	Length	uint16_t	**00 04**	Length is **0x0004**
	Value	uint8_t[4]	48 5C 2B 83	Value is 0x485C2B83 (121.4000003 degree)
speed	Tag	uint16_t	**04 0E**	Tag is **0x040E (1038)**
	Length	uint16_t	00 02	Length is 0x0002
	Value	uint8_t[2]	**00 00**	Value is **0x0000 (0.00 m/s)**
heading	Tag	uint16_t	04 0F	Tag is 0x040F (1039)
	Length	uint16_t	**00 02**	Length is **0x0002**
	Value	uint8_t[2]	00 00	Value is 0x0000 (0.00 degree)

4.1.2 AMI_MSGID_CAN_DATA Sample

The raw data in hex for AMI_MSGID_CAN_DATA Message is below:

01 04 **04 01** 00 46 **04 00** 00 01 **03** 04 01 **00 01** 00 **04 02** 00 01 **00** 04 03 **00 01** 00 **04 04** 00 01 **00** 04 05 **00 01** 00 **04 06** 00 01 **01** 04 07 **00 01** 01 **04 08** 00 01 **00** 04 09 **00 01** 00 **04 0A** 00 01 **03** 04 0B **00 01** 00 **04 0C** 00 01 **00** 04 0D **00 01** 00

The message field explanation of raw data is below:

Message Field		**Data Type**	**Raw Data**	**Description**
Version		uint8_t	**01**	The version **0x01**
Message Type		uint8_t	04	The message type is 0x04 (Data)
Message ID		uint16_t	**04 01**	Application Message ID is **0x0401 (CAN Data)**
Length		uint16_t	00 46	The length of message payload is 0x0046 (All the TLV data length is 70)
Data				The following TLV data
transmission	Tag	uint16_t	**04 00**	Tag is **0x0400 (1024)**
	Length	uint16_t	00 01	Length is 0x0001
	Value	uint8_t[1]	**03**	Value is **0x03**
wheelbrake	Tag	uint16_t	04 01	Tag is 0x0401 (1025)
	Length	uint16_t	**00 01**	Length is **0x0001**

	Value	uint8_t[1]	00	Value is 0x00
TCS	Tag	uint16_t	**04 02**	Tag is **0x0402 (1026)**
	Length	uint16_t	00 01	Length is 0x0001
	Value	uint8_t[1]	**00**	Value is **0x00**
ABS	Tag	uint16_t	04 03	Tag is 0x0403 (1027)
	Length	uint16_t	**00 01**	Length is **0x0001**
	Value	uint8_t[1]	00	Value is 0x00
SCS	Tag	uint16_t	**04 04**	Tag is **0x0404 (1028)**
	Length	uint16_t	00 01	Length is 0x0001
	Value	uint8_t[1]	**00**	Value is **0x00**
low_beam_head_lights	Tag	uint16_t	04 05	Tag is 0x0405 (1029)
	Length	uint16_t	**00 01**	Length is **0x0001**
	Value	uint8_t[1]	00	Value is 0x00
high_beam_head_lights	Tag	uint16_t	**04 06**	Tag is **0x0406 (1030)**
	Length	uint16_t	00 01	Length is 0x0001
	Value	uint8_t[1]	**01**	Value is **0x01**
left_turn_signal	Tag	uint16_t	04 07	Tag is 0x0407 (1031)
	Length	uint16_t	**00 01**	Length is **0x0001**
	Value	uint8_t[1]	01	Value is 0x01
right_turn_signal	Tag	uint16_t	**04 08**	Tag is **0x0408 (1032)**
	Length	uint16_t	00 01	Length is 0x0001
	Value	uint8_t[1]	**00**	Value is **0x00**
hazard_signal	Tag	uint16_t	04 09	Tag is 0x0409 (1033)
	Length	uint16_t	**00 01**	Length is **0x0001**
	Value	uint8_t[1]	00	Value is 0x00
automatic_light_control	Tag	uint16_t	**04 0A**	Tag is **0x040A (1034)**
	Length	uint16_t	00 01	Length is 0x0001
	Value	uint8_t[1]	**03**	Value is **0x03**
daytime_running_lights	Tag	uint16_t	04 0B	Tag is 0x040B (1035)
	Length	uint16_t	**00 01**	Length is **0x0001**
	Value	uint8_t[1]	00	Value is 0x00
fog_light	Tag	uint16_t	**04 0C**	Tag is **0x040C (1036)**
	Length	uint16_t	00 01	Length is 0x0001
	Value	uint8_t[1]	**00**	Value is **0x00**
parking_lights	Tag	uint16_t	04 0D	Tag is 0x040D (1037)
	Length	uint16_t	**00 01**	Length is **0x0001**
	Value	uint8_t[1]	00	Value is 0x00

4.1.3 AMI_MSGID_IMU_DATA Sample

The raw data in hex for AMI_MSGID_IMU_DATA Message is below:

01 04 **04 02** 00 2A **04 00** 00 02 **01 3F** 04 01 **00 02** FF 4D **04 02** 00 02 **26 F8** 04 03 **00 04** FF FF FF 9D **04 04** 00 04 **00 00 00 3D** 04 05 **00 04** FF FF FF F9

The message field explanation of raw data is below:

Message Field	Data Type	Raw Data	Description

Version		uint8_t	**01**	The version **0x01**
Message Type		uint8_t	04	The message type is 0x04 (Data)
Message ID		uint16_t	**04 02**	Application Message ID is **0x0402 (IMU Data)**
Length		uint16_t	00 2A	The length of message payload is 0x002A (All the TLV data length is 42)
Data				The following TLV data
lateral	Tag	uint16_t	**04 00**	Tag is **0x0400 (1024)**
	Length	uint16_t	00 02	Length is 0x0002
	Value	uint8_t[2]	**01 3F**	Value is **0x013F (0.319 m/s^2)**
longitudinal	Tag	uint16_t	04 01	Tag is 0x0401 (1025)
	Length	uint16_t	**00 02**	Length is **0x0002**
	Value	uint8_t[2]	FF 4D	Value is 0x FF4D (-0.179 m/s^2)
vertical	Tag	uint16_t	**04 02**	Tag is **0x0402 (1026)**
	Length	uint16_t	00 02	Length is 0x0002
	Value	uint8_t[2]	**26 F8**	Value is **0x26F8 (9.976 m/s^2)**
roll	Tag	uint16_t	04 03	Tag is 0x0403 (1027)
	Length	uint16_t	**00 04**	Length is **0x0004**
	Value	uint8_t[4]	FF FF FF 9D	Value is 0x FFFFFF9D (-0.099 dps)
pitch	Tag	uint16_t	**04 04**	Tag is **0x0404 (1028)**
	Length	uint16_t	00 04	Length is 0x0004
	Value	uint8_t[4]	**00 00 00 3D**	Value is **0x0000003D (0.061 dps)**
yaw	Tag	uint16_t	04 05	Tag is 0x0405 (1029)
	Length	uint16_t	**00 04**	Length is **0x0004**
	Value	uint8_t[4]	FF FF FF F9	Value is 0x FFFFFFF9 (-0.007 dps)

4.1.4 AMI_MSGID_VEHICLE_EXT_DATA Sample

The raw data in hex for AMI_MSGID_VEHICLE_EXT_DATA Message is below:

01 04 **04 09** 00 0F **04 00** 00 01 **04** 04 01 **00 01** 02 **04 02** 00 01 **02**

The message field explanation of raw data is below:

Message Field		**Data Type**	**Raw Data**	**Description**
Version		uint8_t	**01**	The version **0x01**
Message Type		uint8_t	04	The message type is 0x04 (Data)
Message ID		uint16_t	**04 09**	Application Message ID is **0x0409 (VEHICLE_EXT Data)**
Length		uint16_t	00 0F	The length of message payload is 0x000F (All the TLV data length is 15)
Data				The following TLV data
response_type	Tag	uint16_t	**04 00**	Tag is **0x0400 (1024)**
	Length	uint16_t	00 01	Length is 0x0001
	Value	uint8_t[1]	**04**	Value is **0x04**
siren_use	Tag	uint16_t	04 01	Tag is 0x0401 (1025)
	Length	uint16_t	**00 01**	Length is **0x0001**
	Value	uint8_t[1]	02	Value is 0x02
lights_use	Tag	uint16_t	**04 02**	Tag is **0x0402 (1026)**

	Length	uint16_t	00 01	Length is 0x0001
	Value	uint8_t[1]	**02**	Value is **0x02**

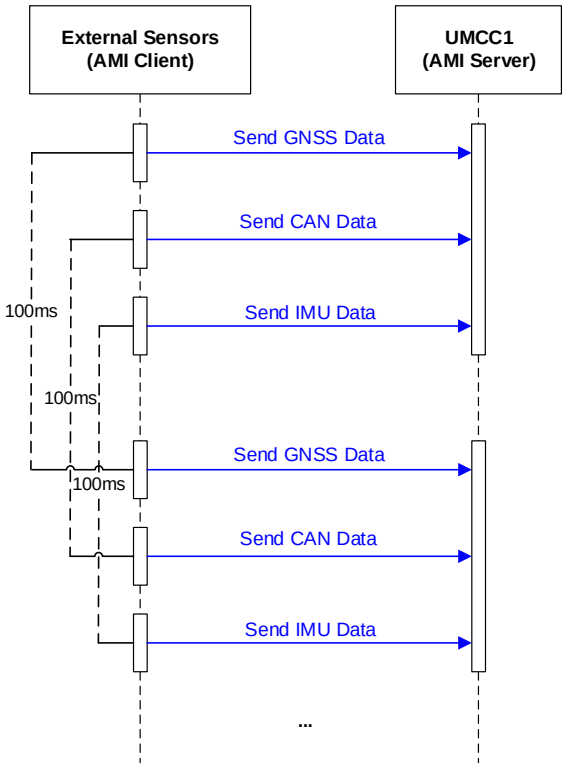
4.2 Message Sequence Chart

The following message sequence chart shows the usage for AMI message.

4.2.1 External Sensors Requirement

The External Sensors should send GNSS, CAN, IMU data within AMI Messages to UMCC1 in every 100ms (Frequency is 10Hz) strictly on UDP Data port.

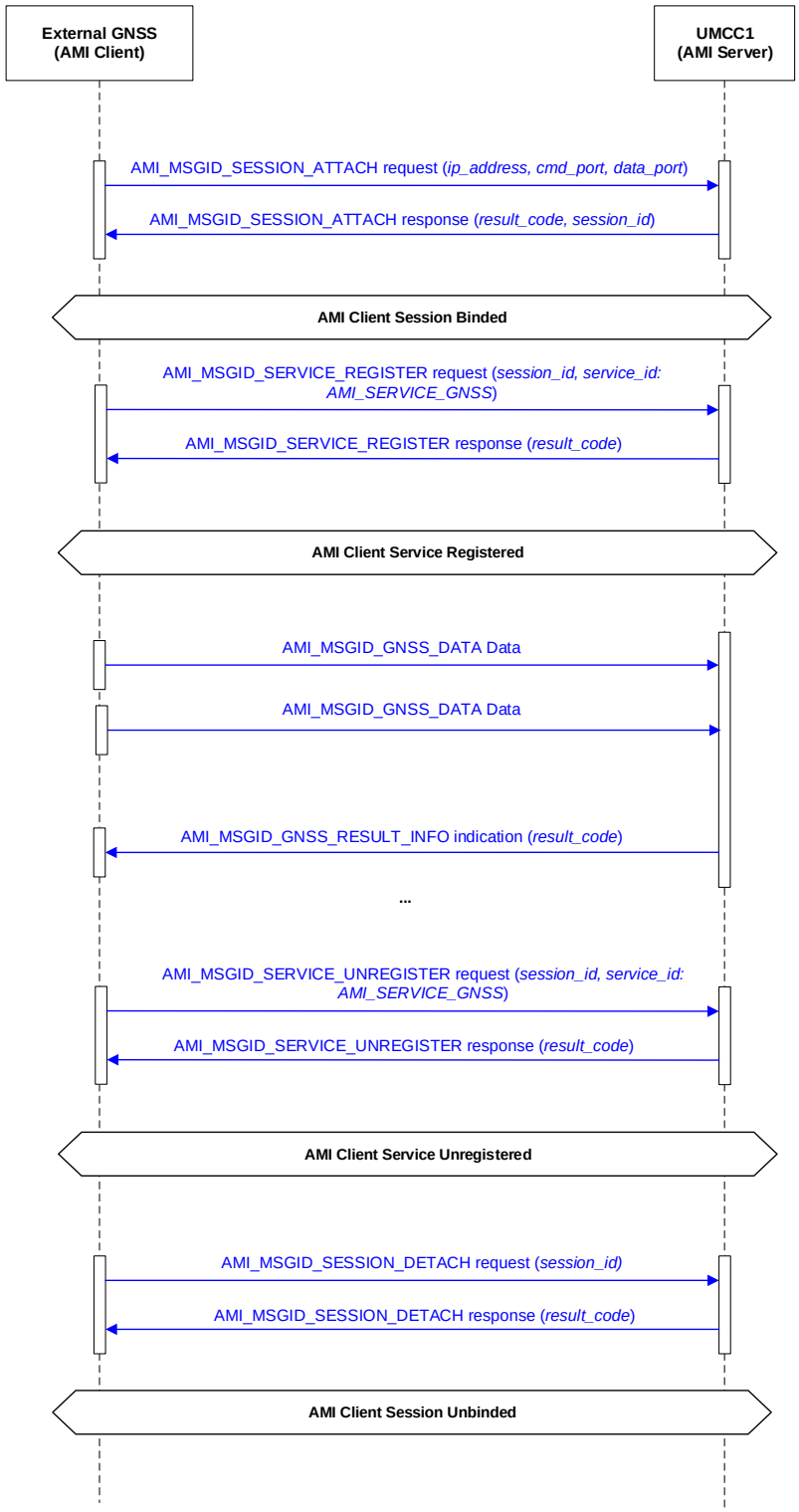
Note:
If CAN message changes, External Sensors should send CAN data immediately.



4.2.2 External Sensors how to get indication

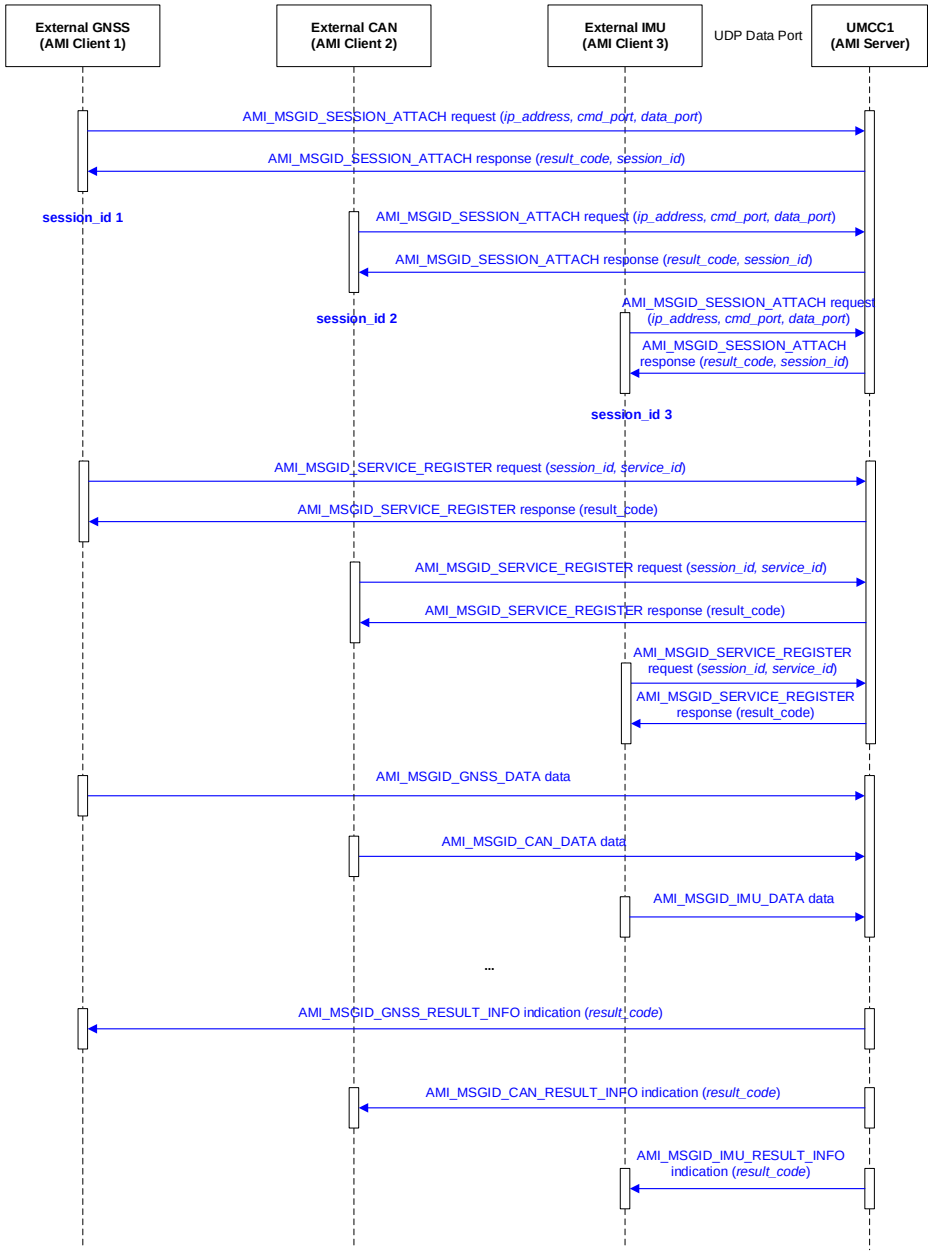
If the External Sensors have interest in specified service, AMI client should register the corresponding service by AMI_MSGID_SERVICE_REGISTER command, after attaching to the AMI server by AMI_MSGID_SESSION_ATTACH

command. When service status change, the AMI_MSGID_XXX_RESULT_INFO indication could be sent to the attached and registered AMI clients.



4.2.3 External Sensors in Multiple AMI Clients

The External Sensors include GNSS, CAN, IMU could be separated to multiple AMI clients. Each AMI client should attach to the AMI server by AMI_MSGID_SESSION_ATTACH command and register the corresponding service by AMI_MSGID_SERVICE_REGISTER command. When error happen, the AMI_MSGID_XXX_RESULT_INFO indication could be sent to the attached and registered AMI clients.



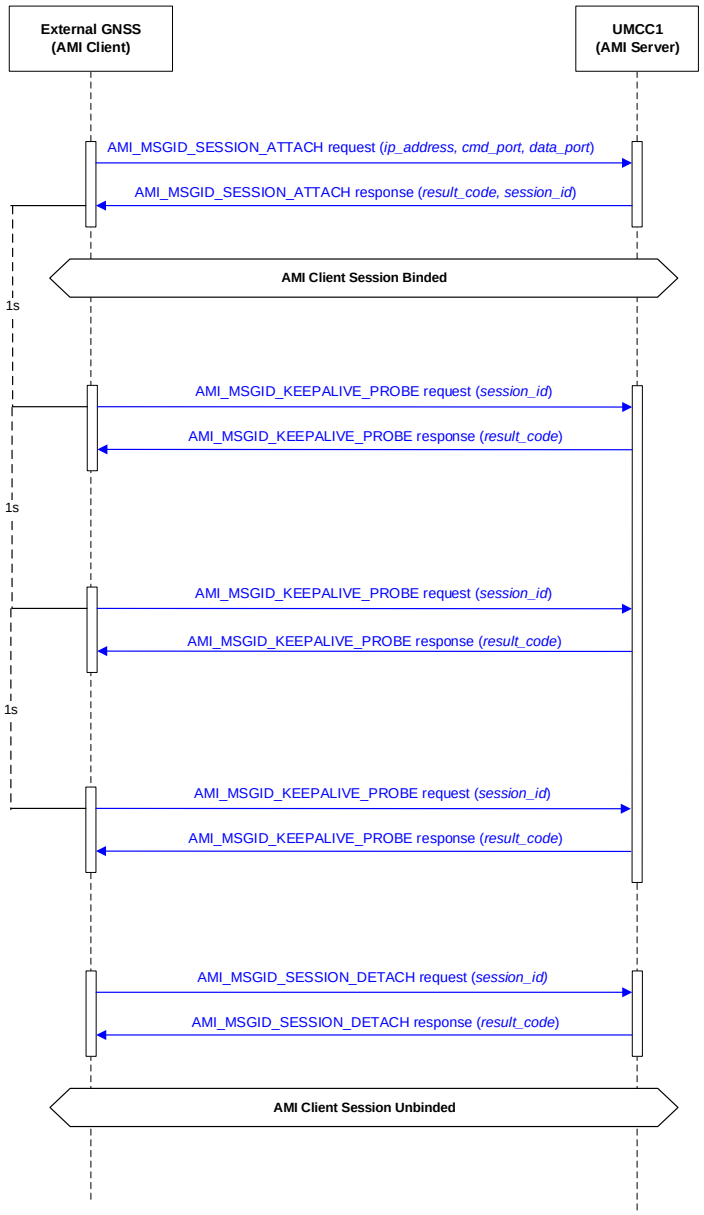
4.2.4 External Sensors Keep Alive

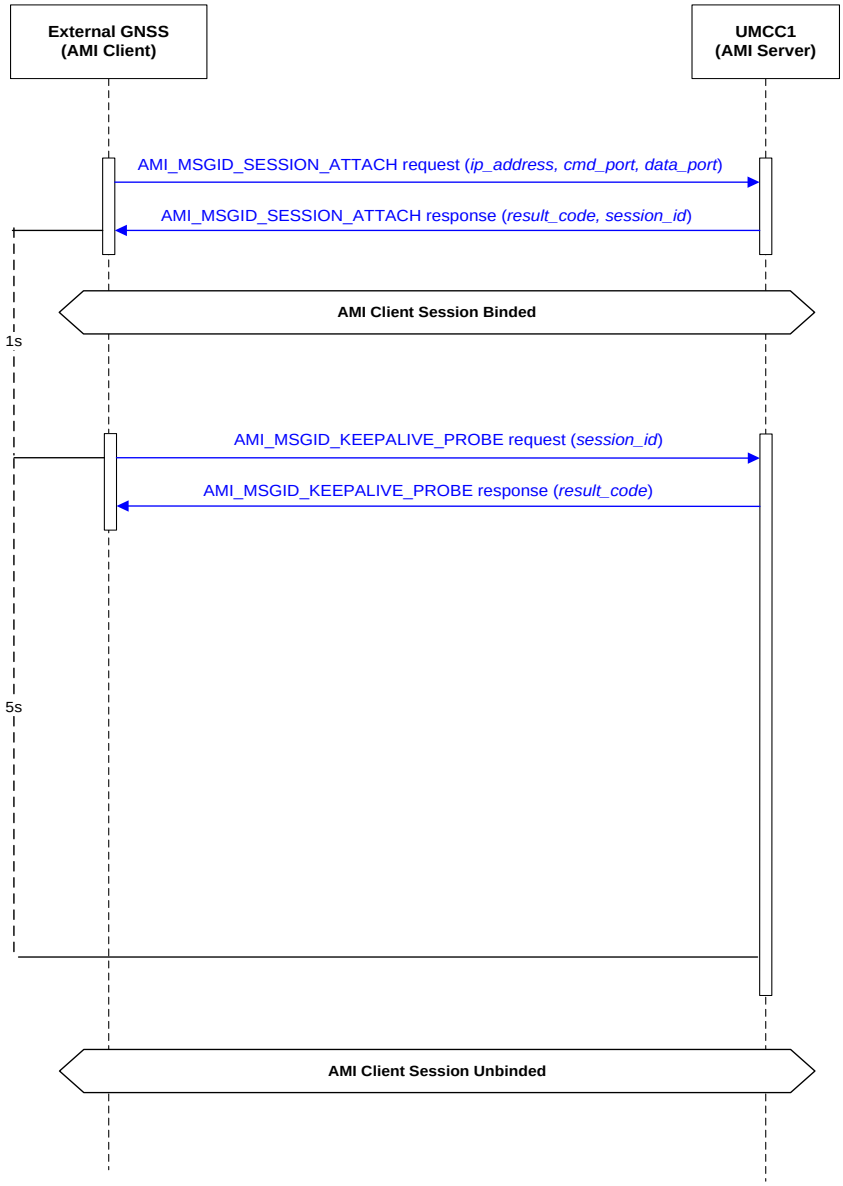
It is necessary to keep the communication with AMI Client and AMI Server. Due to UDP is connectionless, the keep alive mechanism should be supported.

AMI support keep alive message AMI_MSGID_KEEPALIVE_PROBE command. The rules for keep alive shows below:

1. Keep Alive message transmit interval: 1 msg/second. AMI Client should keep sending AMI_MSGID_KEEPALIVE_PROBE request to AMI Server after attached success.

2. Keep Alive maximum timeout: 5 seconds. When AMI Server detect keep alive timeout, the corresponding session with AMI Client will be detached.





4.3 Transport format for UART

AMI message also could be transferred via UART interface. The transport header should be added to ensure reliable transmission on UART.

The transport header consists a number of fields as shown in the following diagram.

Preamble Low	Preamble High	Length	Payload (AMI message)	Checksum
1 Byte	1 Byte	2 Bytes	Length Bytes	1 Byte

The detail fields of AMI message define as below:

Message Field	Data Type	Range	Description
Preamble Low	uint8_t	0x55	The Preamble is for identifying the starting of a packet, where Preamble Low and Preamble High are fixed magic numbers.
Preamble High	uint8_t	0xAA	
Length	uint16_t	1…1400	The length of payload, excluding preamble and checksum.
Payload	uint8_t[n]	-	The transport payload is structured as AMI message. The array length “n” is defined in the above Length field.
Checksum	uint8_t	0…255	The checksum is a value which represents accumulated value for preamble and length and shown in byte. If the value is more than 0xFF, the lower byte of the value will be stored.

4.3.1 Example AMI_MSGID_SERVICE_REGISTER Data of UART packet

The raw data in hex for AMI_MSGID_SESSION_ATTACH Message is below(UART channel):

55 AA **00 14** 01 **01** 00 04 **00 0E** 00 05 **00 04** 64 1F 4A 55 **00 06** 00 02 **04 00** 5E

The message field explanation of raw data is below:

Message Field		Data Type	Raw Data	Description
Preamble Low		uint8_t	**55**	The preamble Low **0x55**
Preamble High		uint8_t	AA	The preamble Low 0xAA
Length		uint16_t	**00 14**	The length of UART payload (excluding preamble and checksum) is **0x0014 (20)**
Payload				The following AMI_MSGID_SERVICE_REGISTER data
Version		uint8_t	01	The version 0x01
Message Type		uint8_t	**01**	The message type is **0x01** (Request)
Message ID		uint16_t	00 04	Application Message ID is 0x0004 (Service register)
Length		uint16_t	**00 0E**	The length of message payload is **0x000E** (All the TLV data length is **14**)
Data				The following TLV data
session_id	Tag	uint16_t	00 05	Tag is 0x0005 (5)
	Length	uint16_t	**00 04**	Length is **0x0004**
	Value	uint8_t[4]	64 1F 4A 55	Value is 0x641F4A55
service_id	Tag	uint16_t	**00 06**	Tag is **0x0006 (6)**
	Length	uint16_t	00 02	Length is 0x0002
	Value	uint8_t[2]	**04 00**	Value is **0x0400(1024)**
Checksum		uint8_t	5E	The checksum is 0x5E

5. Revision History

Version	Date	Section	Details
V1.0.0	2021-2-5	All Sections	1st release.
V1.0.1	2021-2-18	Section 2, 3, 4	1. Add AMI_SERVICE_REGISGER/UNREGISTER command. 2. Add Service ID definition. 3. Add MSC in Appendix.
V1.0.2	2021-2-20	Section 2, 3, 4	1. Add AMI_MSGID_KEEPALIVE_PROBE command. 2. Add TLV item utc_time to AMI_MSGID_CAN_DATA and AMI_MSGID_IMU_DATA.. 3. Add MSC for some scene.
V1.0.3	2021-2-26	Section 2, 3, 4	1. Combine result code and reason code. 2. Add PC5 Profile config command. 3. Add Module status indication. 4. Add session_id for command response.
V1.0.4	2021-3-2	Section 1, 2, 3	1. Change ALPS to ALAP. 2. Add optional description. 3. Some typo.
V1.0.5	2021-3-5	All Sections	Change Font format etc.
V1.0.6	2021-3-10	Section 2, 5	1. Modify AMI_MSGID_SESSION_ATTACH Request parameters. 2. Add Transport format for UART interface.
V1.0.7	2021-4-26	Section 2, 3, 4	1.Add TLV item leap_second to AMI_MSGID_GNSS_DATA. 2. Add AMI_MSGID_VEHICLE_EXT_DATA.