Roadside ITS Station - RIS-9x60

RIS-9x60-Platform Integrators Manual

Doc No. 47000001691

Version 02-00

Author:	Rob Baily, R. Tugrul Güner
DocReviewer:	J. Büger, M. Claesson, H. Liebhart, A. Neustifter
Release:	R. Tugrul Güner

Overview of changes.

No.	Version	Status	Date	Contributor	Type of the change
1	00-01	Draft	2018-02-15	T. Güner	Initial version
2	00-02	Draft	2018-02-22	T. Güner	Review
3	00-03	Draft	2018-03-23	T. Güner	Progress
4	00-04	Draft	2018-05-15	T. Güner	Review
5	00-05	Draft	2018-06-13	T. Güner	Corrections, functional extensions
6	00-06	Draft	2019-09-18	T. Güner	Corrections, functional extensions
7	00-07	Draft	2020-02-10	R. Baily	Improvements, functional extensions
8	01-00	release	2020-03-10	T. Güner	Review and release
9	01-01	Draft update	2020-07-22	R. Baily / S. Khosravi	Updates
10	01-02	Draft update	2020-08-14	T. Güner	Review
11	02-00	Release	2020-09-03	T. Güner	Release

Table 1 Overview of changes

Reference to the status, versions and data classification.

Status:	
Draft	the document is being processed
Released	the document has been checked and released, it can only be modified if the version number is updated.
Obsolete	the document is not valid anymore
Versions:	
00-01, 00-02, etc.	draft versions
01	**first released version with the status “Released“**
01-01, 01-02, etc.	draft versions, that supplement version A
02	**second released version with the status “Released”**
Data classification	
Public	No restriction
Internal	Restricted to internal and external Kapsch employees
Confidential	**Restricted to selected active directory and/or sharepoint groups (default)**
Secret	Restricted to selected employees, server encryption needed

Kapsch TrafficCom

RIS-9x60-Platform Integrators Manual

Doc No. 47000001691
Version 02-00

Document number:	47000001691
Document type:	Integrators Manual
Document issue:	02-00
Document status:	release
Date of issue:	2020-09-03

Table of Contents.

Page

List of Tables.

1 General.

This document is applicable to Roadside ITS Station 9x60 platform, RIS-9x60 including the North American I2V and V2I functionality according to US DOT RSU Spec. V4.1 in terms of general requirements and supports Dedicated Short Range Communication and/or C-V2X communication functionality.

1.1 Purpose of the document.

V2X communication is a core solution in state of the art “connected vehicle” environments. Road operators, infrastructure and road users must cooperate to deliver the most efficient, safe, secure and comfortable journey.

Kapsch TrafficCom specified, designed and developed a new generation Roadside Unit (RSU), the Roadside ITS Station 9x60 platform, or RIS-9x60, as a response to North American, European and Asia-Pacific market demand.

This document is the V2X roadside equipment Integrators Manual to operate RIS-9x60 with IEEE WAVE and SAE J2735 based communication stack software for DSRC and/or C-V2X communication systems and US DOTs RSU specification V4.1.

1.2 Abbreviations.

The following table contains a list of most important abbreviations used within this document to enable an easy reading.

Abbreviation	Description
C-V2X	Cellular V2X communication
CMCC	Connected Mobility Control Center
DSRC	Dedicated Short Range Communication
IFM	Immediate Forward Message
MIB	Management Information Base
PDU	Protocol Data Units
RSU	Road Side Unit
OBU	Onboard unit
SRM	Store and Repeat Message
V2X	Vehicle to everything communication

Table 2 List of used abbreviations

1.3 List of referenced documents.

Ref. No	Document Title
[1]	DSRC Roadside Unit (RSU) Specifications Document v4.1(October 31, 2016)
[2]	SAE J2735 03.2016, Dedicated Short Range Communications (DSRC) Message Set Dictionary
[3]	Simple Network Management Protocol Version 3 (SNMPv3)

Table 3 List of referenced documents

1.4 Exceptions using this document and applying existing specifications.

The methods implemented according to US DOT RSU Specification V4.1 are not fully applicable to C-V2X communication.
Today C-V2X communication mostly focused on safety related V2X communication and applying only one single communication channel. As a consequence there is no need to select a specific radio channel except the one regulated. Radio channel, bandwidth and transmit power to be used are pre-programmed on the radio device depending radio frequency regulations and/or permissions e.g. experimental license and must not be changed by user and application.
The channel numbers and other radio parameters in the US DOT MIB doesn’t have any effect on C-V2X.
There is no second service or control channel available therefore the WSA concept is not applicable at the moment same applies for IP V6 communication.

2 Security Configuration

The RIS-9x60 is equipped with an HSM (Hardware Security Module) and software that communicates with an SCMS (Security Credential Management System) to enable secure and trusted message transmit and reception.

2.1 Process Overview

The general process for a V2X device to obtain security credential information is:

- The device generates a security enrollment request on its hardware, using the HSM to secure private keys. The request will include the services (PSIDs) to be used and the location where the unit is to be installed.
- The enrollment request is submitted to the SCMS for it to generate an enrollment response. The responses contains information about the DNS name for the SCMS and the root certificates used. This submission can happen by one of these methods:
  - The enrollment request file is transferred to a computer and a user logs on to a portal to upload the enrollment request. The SCMS generates an enrollment response file which is transferred to the device and then processed locally.
  - The enrollment request is sent through a web API to the SCMS. The response is processed from the return value of the API call. **NOTE:** There is currently not a standard for this interaction or API so each SCMS needs to be handled differently.
- Once the enrollment response is processed the device is now ready to request the generation of certificates. When the V2X software starts up and determines there is network connectivity the security module contacts the SCMS to indicate that certificates should be generated for the device. The SCMS responds with the time that the certificates will be ready.
- When the time comes the security module downloads the certificates from the SCMS and stores them locally.
- The device may now sign messages that are transmitted so that other devices can trust they came from a valid source.
- When the security module detects that the certificates are due to expire is requests more certificates to be generated by the SCMS and downloads them when they become available.

2.2 RIS-9x60 Setup

On the RIS-9xv60 security is set up when running the “***roadside admin configure***” command. As part of this command the user is asked about setting up security. When security is enabled the enrollment request is generated and can be processed using the method of logging into the relevant SCMS portal to get a response file or using a direct network connection if supported by the SCMS vendor. Both methods are described below in more detail.

Important: Be sure that the system time is correct before running the enrollment. Security systems are sensitive to time accuracy for correct processing. Failure to have the correct time on the device may result in an unusable security state!

Important: If you experience problems during the enrollment request indicating “SCI2C error” the HSM on the device may need to be unlocked. If you believe this is the case please contact the Kapsch support team.

2.2.1 Portal Request Processing

Most SCMS vendors support enrollment through a secure portal. Copy the enrollment request file from the RSU to a local machine, upload the request to the portal and wait for the response. Once the response file is obtained use following steps to process it:

- Copy the enrollment response file (should be a .zip file) to the RIS-9x60.
- Use sudo to run the ***security_process_enrollment.sh*** script passing the path to the enrollment response file. See example below.

```
sudo security_process_enrollment.sh RESPONSE_FILE.zip
```

- If this is successful then reboot the device and it is ready to start requesting certificates.

2.2.2 ISS REST services

When ISS is the SCMS provider based on the security profile the enrollment process will ask if you want to use immediate enrollment or not, if you choose this option it will use ISS rest services and automatically enroll the device.
In order for this to work you must have the following ready,

- An internet connection to connect to the SCMS
- The appropriate cert and key files to validate the request for the ISS system

Please perform following steps:

- Copy the cert and key files to the /home/<username> directory
- During the enrollment process you will be requested if you want to " proceed with immediate enrollment?” Enter Y to continue with the process.
- Next you will be requested if you want to use pilot or pre-production server. During testing choose pre-production and for field deployments select pilot.
- When prompted to enter “the location of the directory with the correct cert.pem and key.pem files” use /home/<username> which should be where the files were copied in the first step.

It will start to access ISS SCMS server and if the request is successfully processed you will get the confirmation. If the request does not work because of a communications error you can submit the enrollment manually via the relevant portal.

2.3 Security Status

At any point in time you can look at the status of the security certificate processing to see where things are in the process. Run the ***security_status.sh*** script using sudo.

Important: If the security enrolled in hardware mode it will ask some permissions to stop applications in order to access HSM module.
Make sure to reboot the device if you choose to stop applications.

The output will include the following major status indicators at the end of the report:

- Enrollment: Indicates how far in the enrollment process the device is. Possible values are:
    - NOT STARTED : Security has not been set up so enrollment has not started.
    - INITIATED : Security has been set up so that enrollment can be set up.
    - REQUEST GENERATED : The enrollment request has been generated.
    - COMPLETED : The enrollment response has been processed and the device is enrolled successfully.
- Enrollment Start: indicates the date that the enrollment request has been submitted. Possible values are :
    - Unknown: The enrollment response has not been processed.
    - “Date”: The date that the enrollment request has been submitted.
- Cert Database : Indicates whether the root certificate database is set up. These certificates are not set up until the device contacts the SCMS after enrollment. Possible values are:
    - EMPTY : Root certificates are not present.
    - ROOT CERTS PRESENT : Root certificates are present. They are listed above the overall status section.
- CRL Downloads : Indicates whether and CRL lists have been downloaded. This is one of the first steps with communication to the SCMS so the presence of these typically indicates the unit has the correct network setup to connect to the SCMS. Possible values are:
    - NOT PRESENT : No CRLs have been downloaded. Network may not allow communication to the SCMS.
    - AVAILABLE : The device has downloaded CRLs from the SCMS.
- Application Certs : Indicates whether or not application certificates are ready. Possible values are:
    - UNAVAILABLE : Application certificates are not present.
    - INACTIVE : Indicates that the device was set up with certificates at one point but is no longer. This can happen if the device is unable to refresh its list of certs over time.
    - PRESENT : Application certs are installed. Some information about them is listed above the overall status section.

2.4 Re-Enrollment

In some cases you may need to re-enroll a device to a different SCMS or using different PSIDs and permissions.

Important:

When you re-enroll a device the previous enrolment is permanently lost! Be ABSOLUTELY SURE you want to re-enroll before initiating this process.

If you are sure you want to reset the enrollment run the ***security_reenroll.sh*** script using sudo. This will stop the running applications, reset the security and then run the enrollment process again.

Whenever this command is run you need to reboot the unit for changes to take effect.

2.5 Disable and Reenable

After security has been set up you may disable it and run with no security. This should only be in the event of security issues that need to be resolved and you are not running any critical applications where security is required. To disable security run the following command:

```
sudo security_enable.sh 0
```

Once issues are resolved and you are ready to enable security run the following command.

```
sudo security_enable.sh 1
```

Whenever this command is run you need to reboot the unit for changes to take effect.

3 CMCC Management Setup

3.1 General.

Kapsch RIS-9x60 units can be configured to connect to the Connected Mobility Control Center for monitoring and managing connected vehicle infrastructure environments. During the “***roadside admin configure***” command You will need a subscription to CMCC and the appropriate id and password for the device.

4 Interface to Backoffice Service

4.1 General.

DSRC RSU(s) transmits messages (Application Layer PDUs) formatted in accordance with SAE J2735 [2] using one of two mechanisms

- Store and Repeat (SR)
- Immediate Forward (IF)

Store and Repeat messages (SRM) are received from a back office service and stored on the specified SNMPv3 [3] OID of the RSU, see also ANNEX B. Transmit Instructions are included with each message that defines how often the message should be transmitted, when the message should start being transmitted, when the message should stop being transmitted, the channel that should be used for the transmission, the Provider Service Identifier (PSID) the message is associated with, and whether the message should be signed and/or encrypted. These transmission instructions should be extracted from the received SNMPv3 message and written to the specified Object Identifier (OID). Once the message expires, it should be removed from RSU MIB / OID storage and the associated SNMPv3 OID should be cleared.

The RSU transmits Immediate Forward messages (IFM) as they are received by the RSU from a back office service. Transmission instructions accompany IF messages, including the channel that should be used for the transmission, the PSID the message is associated with, and whether the message should be signed and/or encrypted. These transmission instructions should be extracted from the IF message and written to the appropriate SNMPv3 OID. The associated SNMPv3 OID should be cleared once the IF messages cease.

4.2 SNMP Credentials

SNMP V3 credentials must be used for every operation. The credentials must match the setup for user id, keys and the encryption levels used. Best practice calls for the users of an RSU to change the SNMP credentials upon installation. See the Administrators Manual for information on SNMP user management.

For all examples below it is assumed that the correct user, auth and priv options and host name are used with the command and set in the environment variable SNMP_ARGS.

Typically walking the MIB is good check to see if you at least have read access to the data. To walk the MIB using snmpwalk use this command:

```
snmpwalk ${SNMP_ARGS} RSU-MIB:rsuMIB
```

One other important note about the RSU MIB is that modification can be made ***ONLY*** when it is set to “standby mode”. If you attempt to set a value when it is in “operate” mode you will get a vague error. To check the mode use:

```
snmpget ${SNMP_ARGS} RSU-MIB:rsuMode.0
```

To set the device to standby:

```
snmpset ${SNMP_ARGS} RSU-MIB:rsuMode.0 i 2
```

To set the device back to operate:

```
snmpset ${SNMP_ARGS} RSU-MIB:rsuMode.0 i 4
```

4.3 Store and Repeat-Encoded Payload Messages (SRM).

The roadside unit will transmit DSRC messages based on SNMP OIDs received via SNMPv3 from back office service. Each OID entry and related SNMP message will contain the transmission instructions and encoded payload for 1 DSRC message. Transmit Instructions included with each message define how often the message should be transmitted, when the message should start being transmitted, when the message should stop being transmitted, the channel that should be used for the transmission, the Provider Service Identifier (PSID) the message is associated with, and whether the message should be signed and/or encrypted.

Once the message expires, it will be removed from the RSU and the associated SNMPv3 OID will be cleared.

Each OID entry will and related SNMPv3 message shall contain

- Message Type/Description
- Message PSID
- Message Priority
- Transmission Channel Mode
- Transmission Channel
- Transmission Interval
- Message Delivery (transmission) start time
- Message Delivery (transmission) stop time
- Signature
- Encryption
- Payload
- Enable

See detailed format in Annex A..

4.3.1 Store & Repeat Message Start of Transmission.

The RSU begins transmitting the payload of an OID Store and Repeat entry (rsuSRMStatusEntry) over a DSRC interface on or after the start time specified in rsuSRMDeliveryStart. The start value must be valid See detailed rsuSRMStatusEntry structure in ANNEX B.

4.3.2 Store & Repeat Message End of Transmission.

The RSU stops transmitting the payload of an OID Store and Repeat entry (rsuSRMStatusEntry) over a DSRC interface at the end time specified in rsuSRMDeliveryStop.
See detailed rsuSRMStatusEntry structure in ANNEX B.

4.3.3 Store & Repeat Message Storage.

The RSU has minimum storage for at least 100 active SRMs in its MIB (config.db).

4.3.4 Store & Repeat Message Add Entry.

The RSU allows authorized users to add an OID Store and Repeat entry (rsuSRMStatusEntry) through SNMPv3 OID 1.0.15628.4.1.4.x.
See detailed rsuSRMStatusEntry structure in ANNEX B.

4.3.5 Store & Repeat Message Remove Entry.

The RSU allows authorized users to remove an OID Store and Repeat entry (rsuSRMStatusEntry) through SNMPv3 OID 1.0.15628.4.1.4.x.
See detailed rsuSRMStatusEntry structure in ANNEX B.

4.3.6 Store & Repeat Message View Entry.

The RSU allows authorized users to view (read) the content of an OID Store and Repeat entry (rsuSRMStatusEntry) through SNMPv3 OID 1.0.15628.4.1.4.x.
See detailed rsuSRMStatusEntry structure in ANNEX B.

4.3.7 Store & Repeat Message Modify Entry.

The RSU allows authorized users to modify the content of an OID Store and Repeat entry (rsuSRMStatusEntry) through SNMPv3 OID 1.0.15628.4.1.4.x.
See detailed rsuSRMStatusEntry structure in ANNEX B.

4.3.8 Store & Repeat Message Authorized Access Log Entry.

The RSU writes an INFO entry in the custom log file for each authorized access to an OID Store and Repeat entry (rsuSRMStatusEntry) through SNMPv3 OID 1.0.15628.4.1.4.x
See detailed rsuSRMStatusEntry structure in ANNEX B.

4.3.9 Store & Repeat Message Failed Access Log Entry.

The RSU writes a WARNING entry in the custom log file for each failed access attempt to an OID Store and Repeat entry (rsuSRMStatusEntry) through SNMPv3 OID 1.0.15628.4.1.4.x
See detailed rsuSRMStatusEntry structure in ANNEX B.Store & Repeat Message Transmission Log Entry.

The RSU writes a NOTICE entry in the custom log file or changes in transmission status resulting from a user initiated device shut down, device boot up, message start time or message end time.

4.3.11 SNMP Examples

4.3.11.1 Adding an Entry

The commands below show an example to set up a TIM (SAE J2735, Traveller Information Message) using the SNMP interface. You can modify this yourself to send different messages. The key thing to note is that the rsuSRMDsrcMsgId should match the content of the message. This can usually be checked by ensuring the first 2 bytes in the payload match the message type. For example here 31 = 0x001f.

```
snmpset ${SNMP_ARGS} \

RSU-MIB:rsuSRMPsid.1 x 8003 \

RSU-MIB:rsuSRMDsrcMsgId.1 i 31 \

RSU-MIB:rsuSRMTxMode.1 i 1 \

RSU-MIB:rsuSRMTxChannel.1 i 176 \

RSU-MIB:rsuSRMTxInterval.1 i 1000 \

RSU-MIB:rsuSRMDeliveryStart.1 x 07e20b071610 \

RSU-MIB:rsuSRMDeliveryStop.1 x 07e40b07161a \

RSU-MIB:rsuSRMPayload.1 x
001f4d2010000000000266bccdb082b28e6568c461045380342800002fc25445f0e030800200393205a200ba3
174a062df5b290f93d901d05dc036e7ec066877d0c34eba16e3d408364010c189408840 \

RSU-MIB:rsuSRMEnable.1 i 1 \

RSU-MIB:rsuSRMStatus.1 i 4
```

4.3.11.2 Updating an Entry

If you want to update any field in the message you just need to put the RSU in standby mode and modify the fields you want to change. For example, in the example above if you want to change the hex part of the payload you can use following command:

```
snmpset ${SNMP_ARGS} RSU-MIB:rsuSRMPayload.1 x
001F34200312020202020202020000802427F440E0FC000660D693A401AD2747FC0B41000100000013360BC6CF
506007C045800000805C0
```

Please note that rsuSRMStatus is not be included and this only works if the row already exists.

4.3.11.3 Deleting an Entry

If you want to delete a SRM entry you can use following command:

```
snmpset ${SNMP_ARGS} udp:localhost:161 \

RSU RSU-MIB:rsuSRMStatus.1 i 6
```

4.4 Immediate Forward-Encoded Payload Messages (IFM).

The RSU transmits Immediate Forward messages which are a specific form of the SRM. Transmission instructions accompany IF messages contains the same parameters as SRM except that the Transmission Interval, Message Delivery (transmission) start and stop time are set to Null.

The associated SNMPv3 OID should be cleared once the Immediate Forward message cease.

Each OID entry will and related SNMPv3 message shall contain

- Message Type/Description
- Message PSID
- Message Priority
- Transmission Channel Mode
- Transmission Channel
- Transmission Interval (set to Null)
- Message Delivery (transmission) start time (set to Null)
- Message Delivery (transmission) stop time (set to Null)
- Signature
- Encryption
- Payload
- See detailed data format in Annex A..

Caveat: OIDs for Message Priority, Transmission Interval, Message Delivery (transmission) start time, Message Delivery (transmission) stop time, Signature, Encryption and Payload are missing in the MIB [1]. A solution need to be defined.
Enable is an OID for each IFM entry [1] but not listed above. A solution need to be defined.

4.4.1 Immediate Forward Message Receive.

The RSU receive messages for Immediate Forward from back office service, a network host, on default UDP port 1516.
See detailed rsuIFMStatusEntry structure in ANNEX B for setup and the content of an IFM message in ANNEX A.

4.4.2 Immediate Forward Message Transmit.

The RSU transmits over a DSRC interface each message payload received from back office service, a network host, after extracting it from IFM and upon writing to the specified SNMPv3 OID.
See detailed rsuIFMStatusEntry structure in ANNEX B.

4.4.3 SNMP Examples

4.4.3.1 Adding an Entry

```
snmpset ${SNMP_ARGS} \
RSU-MIB:rsuIFMPsid.1 x 8001 \
RSU-MIB:rsuIFMDsrcMsgId.1 i 19 \
RSU-MIB:rsuIFMTxMode.1 i 1 \
RSU-MIB:rsuIFMTxChannel.1 i 182 \
RSU-MIB:rsuIFMEnable.1 i 1 \
RSU-MIB:rsuIFMStatus.1 i 4
```

4.4.3.2 Deleting an Entry

```
snmpset ${SNMP_ARGS} \
RSU-MIB:rsuIFMStatus.1 i 6
```

4.5 DSRC Message Forwarding.

The RSU may receive messages broadcast by a DSRC-equipped mobile device and forward them to a remote host. Messages are forwarded based on the PSID. The PSID of the message to be forwarded, the IP address and port number of the remote host, the transport protocol to use, the receive signal strength, the interval at which to forward, and the period to forward messages are all configurable. The configurable parameters are stored in the specified SNMPv3 OID.

4.5.1 DSRC Message Forwarding.

The RSU forwards WSMP messages received on any DSRC interface, containing a specified PSID, to a specified network host, as configured in SNMPv3 MIB OID 1.0.15628.4.1.7.

The WSMP Message Forwarding SNMPv3 MIB Object contains the following information

- PSID
- Dest_IP Address
- Dest_Port
- TransPort_Protocol
- RSSI
- MsgForwardInterval (RSU forwards every nth message received)
- DeliveryStart
- DeliveryStop
- ForwardEnable

See detailed SNMPv3 MIB OID 1.0.15628.4.1.7 in ANNEX B.

4.5.2 SNMP Examples

4.5.2.1 Adding an entry

```
snmpset ${SNMP_ARGS} \
RSU-MIB:rsuDsrcFwdPsid.1 x 8002 \
RSU-MIB:rsuDsrcFwdDestIpAddr.1 x 200100bb0000000000000000000000ff \
RSU-MIB:rsuDsrcFwdDestPort.1 i 47563 \
RSU-MIB:rsuDsrcFwdProtocol.1 i 2 \
RSU-MIB:rsuDsrcFwdRssi.1 i -100 \
RSU-MIB:rsuDsrcFwdMsgInterval.1 i 1 \
RSU-MIB:rsuDsrcFwdDeliveryStart.1 x 07e20b071610 \
RSU-MIB:rsuDsrcFwdDeliveryStop.1 x 07e40b07161a \
RSU-MIB:rsuDsrcFwdEnable.1 i 1 \
RSU-MIB:rsuDsrcFwdStatus.1 i 4
snmpset ${SNMP_ARGS} \
RSU-MIB:rsuDsrcFwdStatus.1 i 6
```

5 IPV6 Integration

5.1 Overview

The WAVE specification provides a mechanism for an OBU to communicate to an RSU using IPV6 over the 802.11 interface. This communication can be directly to the RSU or as a gateway to a backhaul on the ethernet side of the RSU.

For the RSU you will need to set up the following:

- The second radio set to alternate between the control channel (178) and a designated service channel.
- Routing advertisement information (WRA) to indicate what the IPV6 network configuration will be between the RSU and OBU.
- A wave service entry (part of WSA) for the IPV6 service.
- To test routing across the RSU ethernet interface you will need it configured with an IPV6 network that does not overlap with the one provided in the WRA and appropriate routing information.

The sections below assume the user is familiar with IPV6 networks or can get help from a networking guru to confirm the network parameters are correct and fit within the connected network.

When SNMP is used it is assumed the appropriate credentials are available and the RSU is put in standby mode as discussed above.

5.2 Channel Configuration

Typically we will want the first radio on the safety channel and the second radio alternating between the control channel and the service channel. The example below uses channel 176 for the service channel.

```
# set first radio to continuous on the safety channel

snmpset ${SNMP_ARGS} RSU-MIB:rsuDCMMode.1 i 0 RSU-MIB:rsuDCMCCH.1 i 172

# set second radio to alternating on the control and service channels

snmpset ${SNMP_ARGS} RSU-MIB:rsuDCMMode.2 i 1 RSU-MIB:rsuDCMCCH.2 i 178 RSU-MIB:rsuDCMSCH.2 i 176
```

5.3 Wave Routing Advertisement (WRA)

Set up the network on the DSRC interface with its own network. This should be a different network than is used on the ethernet side. Typically a network mask of 64 bits (0x40) is used for allowing units to create their own IP addresses based on their MACs.

The example below sets the following configuration:

- A network prefix of 2001:BB::0 using 64 (0x40) bits
- A default gateway of 2001:BB:1. This is will be the IP address the RSU uses on the DSRC interface.
- A DNS entry of 2001:BB:2. The RSU does not provide DNS so that must be obtained from a different device.

```
snmpset ${SNMP_ARGS} RSU-MIB:rsuWraIpPrefix.0 x 200100BB000000000000000000000000 RSU-
MIB:rsuWraIpPrefixLength.0 x 40 \
RSU-MIB:rsuWraGateway.0 x 200100BB000000000000000000000001 RSU-MIB:rsuWraPrimaryDns.0 x
200100BB000000000000000000000002
```

5.4 WSA Service

Set up a service advertisement for IPV6 service using the service channel set up before.

```
snmpset ${SNMP_ARGS} RSU-MIB:rsuWsaStatus.1 i 4 RSU-MIB:rsuWsaPsid.1 x EFFFFFFE RSU-
MIB:rsuWsaPriority.1 i 0 RSU-MIB:rsuWsaChannel.1 i 176
```

5.5 Wave Device IP Configuration

After the WRA and WSA service has been set up you should see the default gateway from the WRA information set as a global IPV6 address on the appropriate wave interface. The output below comes from using the command "***ip addr show dev wave1***" The fe80 address is the automatically created link local address.

```
21: wave1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen
1000

    link/ether 76:b7:c6:c9:8b:3a brd ff:ff:ff:ff:ff:ff

    inet6 2001:bb::1/64 scope global

       valid_lft forever preferred_lft forever

    inet6 fe80::74b7:c6ff:fec9:8b3a/64 scope link

       valid_lft forever preferred_lft forever
```

5.6 Ethernet

In order to enable traffic to be passed through to the other side of the device through the ethernet port you will need to set up an IPV6 address that is NOT on the same network as the DSRC interface. The update_ipconfig.sh command for ipv6_global takes an optional argument for gateway at the end if you have one.

```
# short term to test

ip addr add 3001:bb::1/64 dev eth0

# longer term to persist

update_ipconfig.sh ipv6 enable

update_ipconfig.sh ipv6_global static 3001:bb::1 64
```

If using the commands to persist you will need to reboot the Kapsch RSU after setting the IPV6 configuration for it to take effect.

6 Monitoring Traffic on Radios

In order to check what messages has been received and/or broadcasted by the RSU radio, you can get pcap files for each radio under: **/var/log/pcap** . There are pcap files for RadioA and RadioB which are set up for sending and receiving messages. You can use Wireshark application to open files.

Also you may use tcpdump on the specific interface to see traffic in real-time. ***cw-mon-txa*** and ***cw-mon-txb*** are interfaces that radioA and radioB use respectively to send out messages. ***cw-mon-rxa*** and ***cw-mon-rxb*** are interfaces that radioA and radioB use respectively to receive messages. In order to use tcpdump first you need to enable the interface by running:

sudo <interface_name> up and then you can use ***sudo tcpdump -X -i <interface_name>*** to monitor the traffic.

For RIS-9260 RSU please refer to (Annex D)

7 ANNEX A

The format for both encoded Store & Repeat Messages and encoded Immediate Forward messages is contained below

```
# Message File/Data Format
# Modified Date: 04/10/2014
# Version: 0.7
Version=0.7
#
# Message Dispatch Items
#
# All line beginning with # shall be removed in file sent to radio
#
# Message Type
# Values: SPAT, MAP, TIM, (other message types)
Type=<Type>
#
# Message PSID as a 2 Byte Hex value (e.g. 0x8003)
PSID=<PSID>
#
# Message Priority in the range of 0 (lowest) through 7
Priority=<priority>
#
# Transmission Channel Mode
# Allowed values: CONT, ALT
TxMode=<txmode>
# Allowed values: 172, CCH, SCH (note: “CCH” refers to DSRC Channel 178 and SCH refers to the
#operator configured DSRC Service Channel)
TxChannel=<channel>
#
# Transmission Broadcast Interval in Seconds
# Allowed values: 0 for Immediate-Forwarding, 1 to 5 for Store-and-Repeat
TxInterval=<txinterval>
#
# Message Delivery (broadcast) start time (UTC date and time) in the form:
# "mm/dd/yyyy, hh:mm”
# Leave value blank if Immediate Forward mode
DeliveryStart=<mm/dd/yyyy, hh:mm>
#
# Message Delivery (broadcast) stop time (UTC date and time) in the form:
# "mm/dd/yyyy, hh:mm”
# Leave value blank if Immediate Forward mode
DeliveryStop=<mm/dd/yyyy, hh:mm>
#
# Message Signature/Encryption
Signature=<True\False>
Encryption=<True\False>
#
# Message Payload (encoded according to SAE J2735 or other definition)
Payload=<DSRC message payload>
```

8 ANNEX B

Please see the RSU 4.1 MIB specification for normative information.

All time fields are supplied in the format of the first 6 octets in the DateAndTime field as defined in RFC2579. Example: October 7, 2017 at 11:34 PM UTC would be encoded as 07e10a071722” Broken down:

- x07e1 = 2017
- x0a = 10 (Oct)
- x07 = Day 7
- x17 = 2300 hour of day
- x22 = 34 minutes past the hour

Name	OID 15628.4.x	Max-Access	Syntax	Range / Remarks
Store and Repeat				
rsuSRMStatusTable	1.4	not-accessible	SEQUENCE OF	
rsuSRMStatusEntry	1.4.1	not-accessible	RsuSRMStatusEntry	
rsuSRMIndex	1.4.1.1	not-accessible	RsuTableIndex	
rsuSRMPsid	1.4.1.2	read-create	RsuPsidTC	p-encoded hex value of 1-4 octets
rsuSRMDsrcMsgId	1.4.1.3	read-create	Integer32	n.a. already part of payload
rsuSRMTxMode	1.4.1.4	read-create	INTEGER	0 continuous
rsuSRMTxChannel	1.4.1.5	read-create	Integer32	172..184 up to 2 channels simultaneously
rsuSRMTxInterval	1.4.1.6	read-create	Integer32	in milliseconds
rsuSRMDeliveryStart	1.4.1.7	read-create	OCTET STRING	SIZE(0|6) See note above regarding time
rsuSRMDeliveryStop	1.4.1.8	read-create	OCTET STRING	SIZE(0|6) See note above regarding time
rsuSRMPayload	1.4.1.9	read-create	OCTET STRING	SIZE(0..1500)
rsuSRMEnable	1.4.1.10	read-create	INTEGER	0|1 stop/start
rsuSRMStatus	1.4.1.11	read-create	RowStatus	Use 4 to create rows and 6 to delete
Immediate Forward				
rsuIFMStatusTable	1.5	not-accessible	SEQUENCE OF	

rsuIFMStatusEntry	1.5.1	not-accessible	RsuIFMStatusEntry	
rsuIFMIndex	1.5.1.1	not-accessible	RsuTableIndex	
rsuIFMPsid	1.5.1.2	read-create	RsuPsidTC	p-encoded hex value of 1-4 octets
rsuIFMDsrcMsgId	1.5.1.3	read-create	Integer32	n.a. already part of payload
rsuIFMTxMode	1.5.1.4	read-create	INTEGER	0 continuous
rsuIFMTxChannel	1.5.1.5	read-create	Integer32	172..184 up to 2 channels simultaneously
rsuIFMEnable	1.5.1.6	read-create	INTEGER	0|1
rsuIFMStatus	1.5.1.7	read-create	RowStatus	Use 4 to create rows and 6 to delete
DSRC Forwarding				
rsuDsrcForwardTable	1.7	not-accessible	SEQUENCE OF	
rsuDsrcForwardEntry	1.7.1	not-accessible	RsuDsrcForwardEntry	
rsuDsrcForwardIndex	1.7.1.1	not-accessible	RsuTableIndex	
rsuDsrcFwdPsid	1.7.1.2	read-create	RsuPsidTC	1-2 octets 0..65535
rsuDsrcFwdDestIPAddr	1.7.1.3	read-create	Ipv6Address	4 or 16 octets IPv4, IPv6
rsuDsrcFwdDestPort	1.7.1.4	read-create	INTEGER	1024..65535
rsuDsrcFwdProtocol	1.7.1.5	read-create	INTEGER	1|2
rsuDsrcFwdRssi	1.7.1.6	read-create	INTEGER	-100..-60 Not supported
rsuDsrcFwdMsgInterval	1.7.1.7	read-create	INTEGER	1..9
rsuDsrcFwdDeliveryStart	1.7.1.8	read-create	OCTET STRING	SIZE(0|6) See note above regarding time
rsuDsrcFwdDeliveryStop	1.7.1.9	read-create	OCTET STRING	SIZE(0|6) See note above regarding time
rsuDsrcFwdEnable	1.7.1.10	read-create	INTEGER	0|1
messageForwardingRowStatus	1.7.1.11	read-create	RowStatus	Use 4 to create rows and 6 to delete

Table 4 Mapping of the RSU specific MIB OIDs from [1] Appendix B.1 .

9 ANNEX C

RSU 4.1 Specification Requirement	Relevant Manual Section	Remarks
USDOT_RSU-Req_468-v001	Store & Repeat Message Start of Transmission.	
USDOT_RSU-Req_470-v001	Store & Repeat Message End of Transmission.	
USDOT_RSU-Req_452-v002	Store & Repeat Message Storage.	
USDOT_RSU-Req_453-v002	Store & Repeat Message Add Entry.	
USDOT_RSU-Req_454-v003	Store & Repeat Message Remove Entry.	
USDOT_RSU-Req_455-v003	Store & Repeat Message View Entry.	
USDOT_RSU-Req_457-v003	Store & Repeat Message Modify Entry.	
USDOT_RSU-Req_459-v001	Store & Repeat Message Authorized Access Log Entry.	
USDOT_RSU-Req_469-v001	Store & Repeat Message Failed Access Log Entry.	
USDOT_RSU-Req_462-v001	Store & Repeat Message Transmission Log Entry.	
USDOT_RSU-Req_554-v001	Immediate Forward Message Receive.	
USDOT_RSU-Req_471-v003	Immediate Forward Message Transmit.	
USDOT_RSU-Req_437-v005	DSRC Message Forwarding.	

Table 5 Mapping of the RSU 4.1 specification requirements.

10 ANNEX D

RIS-9260 RSUs are capable of C-V2X and DSRC communication either in dual mode active i.e. in both radios working simultaneously or in single mode i.e. only one radio is active.

C-V2X modules are basically LTE modems which are using PC5 sidelink communication technology according 3GPP LTE-V2X Rel.14. Currently RIS-9260 is using an C-V2Xmodule supporting only direct communication between the devices without the involvement of a cellular networks. The module does have its own configuration file (v2x.xml) which defines the frequency, bandwidth and many more radio parameters depending on a geographical zone and related regulations the module is operated in. The configuration file is part of the C-V2X module defined and delivered by the manufacturer together with the current available module firmware and must typically not be changed by any customer. Access C-V2X Configuration

During board startup (after the module has been found and initialized) the current configuration set will be read out and stored in /var/log/cca_capabilities.txt.

You can retrieve the information about channel and bandwidth reading out the file:

```
cat /var/log/cca_capabilities.txt
maxTxPower 23
minFreq 54965
maxFreq 55015
Channel 180
BandWidth 10 MHz
```

10.1 C-V2X Module Firmware

You should make sure the device that has latest C-V2X module firmware (currently **Post-CS 0.0.120.1** is the latest version). To get C-V2X module firmware information reading out the file:

```
cat /var/log/cca_version_info.txt
Linux mdm9150-cv2x 3.18.71 #1 PREEMPT Mon Nov 11 18:50:11 CST 2019 armv7l GNU/Linux
mdm9150-cv2x
v00.02.00.00 (Post-CS 0.0.120.1)
```

Important: If the module on the device is not update please contact support team to help you through the update.

10.2 Communication Mode

You can get the current radio operation mode reading out the file:

```
cat /etc/eeprom-cb/a.dualradio.mode
1,:NXP::CCA: // which it means NXP (DSRC) and CCA (C-V2X) communication modes are available
```

To change the RSU mode you run the command below and follow the prompts to enable the NXP (means DSRC mode), CCA or both as desired.

```
sudo roadside admin al-config
1,:NXP::CCA: // which it means NXP (DSRC) and CCA (C-V2X) communication modes are available
```

Important: You will need to reboot the unit for the changes to take effect.

10.3 Update C-V2X Channel

You can update the channel setting by applying the following command:

```
sudo update_channel.sh -c xxx // xxx channel # like 183
```

Important: You will need to reboot the unit for the changes to take effect.

10.4 C-V2X Data Traffic Monitoring

You can monitor the traffic on the air to and from the RIS-9x60. You will see traffic to different IPv6 ports for RX/TX. TX on port# 2602 and RX on port# 9000. You can run this command to get traffic on radios:

```
sudo tcpdump -X -i rmnet_data1
```

After couple of seconds you can monitor ingoing and outgoing messages.

- END OF DOCUMENT -