



# **OBU-201U Quick Start Guide**

**HW Version: 0B**

**U-Boot: 1.3.3**

**SDK: 4.11.0-sc**

**Stack: US, 914**

**V1.0**

# Table of Contents

1. OBU at a Glance.....	1
2. Get Started – Environment Setup .....	4
3. WSM – Tx & Rx Demo (Using Channel & WSM services).....	5
4. WSM – Tx & Rx Demo (Using Channel, Provider, User, WSM services)...	11
5. Checking MIB status.....	18
6. BSM – Tx & Rx Demo.....	21
7. Firmware Upgrade .....	24
8. Appendix .....	29
A: Channel service .....	29
B: Provider service .....	30
C: User service .....	32
D: WSM service.....	34
E: BSM Tx & Rx .....	34

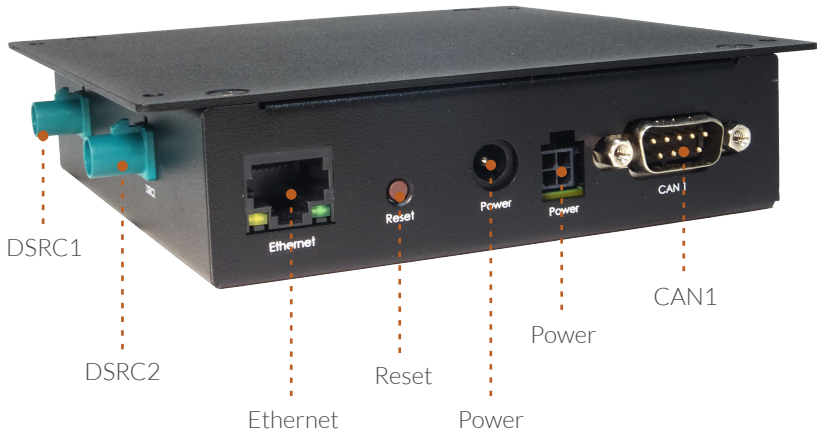
## Abbreviations

DSRC	Dedicated Short Range Communication
CCH	Control Channel
SCH	Service Channel
PSC	Provider Service Context
PSID	Provider Service Identifier
WAVE	Wireless Access in Vehicular Environments
WSA	WAVE Service Advertisements
WSMP	WAVE Short Message Protocol
WME	WAVE Management Entity
BSM	Basic Safety Message
SPAT	Signal Phase And Timing Message
MAP	Map Data
TIM	Traveler Information Message
RSA	Road Side Alert
RTCM	Radio Technical Commission For Maritime Services
CSR	Common Safety Request Message

NMEA	National Marine Electronics Association
ICA	Intersection Collision Alert Message
EVA	Emergency Vehicle Alert Message
PDM	Probe Data Message
SSM	Signal Status Message
SRM	Signal Request Message
PVD	Probe Vehicle Data Message
PSM	Pedestrian Safety Message

# 1. OBU at a Glance

## Front View:



## Rear View:



## Accessories

Following accessories are included with OBU-201:

1. Round jack power adapter:



2. 4-pin automotive power connector:



3. GPS antenna:



#### 4. DSRC antennas:



## 2. Get Started – Environment Setup

### Prerequisites:

1. Terminal emulator program (Example: PuTTY, Speed: 115200 bits per second)
2. PL2303 Prolific Driver
3. Connect the key components (DSRC antenna, GPS antenna, USB to serial adapter and power adapter) as shown in figure below.

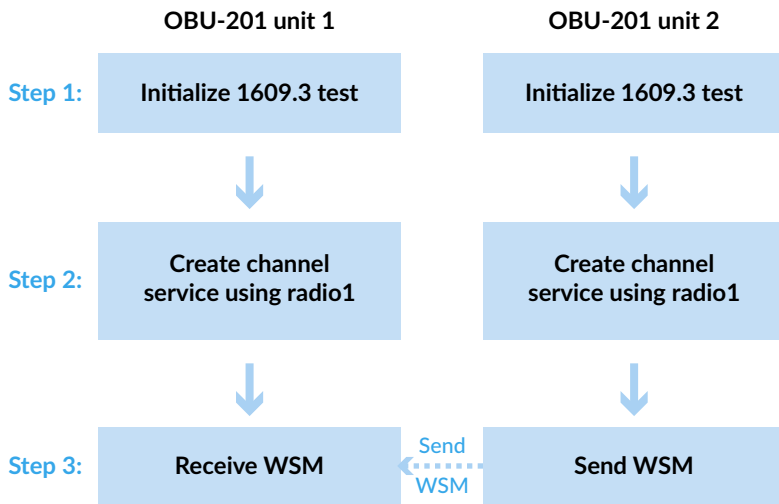


**Note:** For application flexibility in lab and vehicle, two kinds of power cables are provided. One is the round jack power adapter and the other one is 4-pin automotive power connector. It is recommended to use any ONE of the power cables at one time.



### 3. WSM – Tx & Rx Demo (Using Channel & WSM services)

#### Overview:



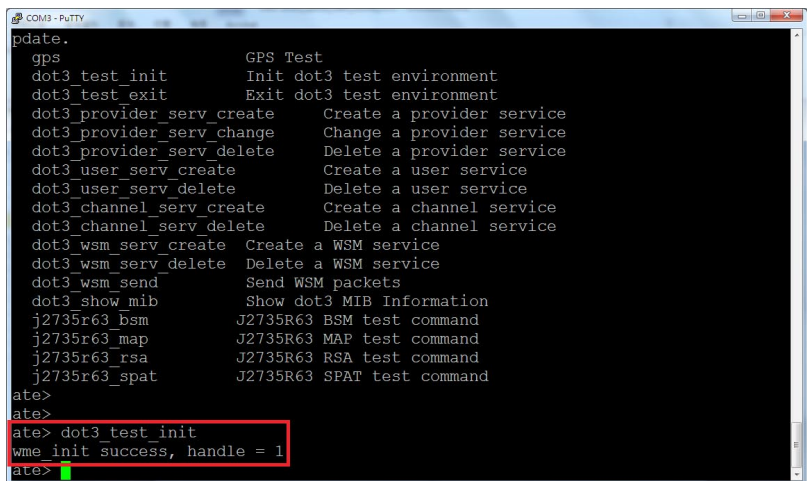
**Note:** This demo requires configuring two OBU-201 units. Only radio number 1 is used. For detailed command description, check section 8 (Appendix).

**Step 1:** On OBU-201 unit 1, initialize 1609.3 test environment

**Example:** `dot3_test_init`

**Description:** This command will get a handler id from 1609.3 to register other services such as: Provider, User, Channel, WSM

**Sample snapshot:**



```
COM3 - PuTTY
pdate.
gps                GPS Test
dot3_test_init     Init dot3 test environment
dot3_test_exit     Exit dot3 test environment
dot3_provider_serv_create  Create a provider service
dot3_provider_serv_change  Change a provider service
dot3_provider_serv_delete  Delete a provider service
dot3_user_serv_create      Create a user service
dot3_user_serv_delete      Delete a user service
dot3_channel_serv_create   Create a channel service
dot3_channel_serv_delete   Delete a channel service
dot3_wsm_serv_create       Create a WSM service
dot3_wsm_serv_delete       Delete a WSM service
dot3_wsm_send              Send WSM packets
dot3_show_mib              Show dot3 MIB Information
j2735r63_bsm              J2735R63 BSM test command
j2735r63_map              J2735R63 MAP test command
j2735r63_rsa              J2735R63 RSA test command
j2735r63_spat             J2735R63 SPAT test command
ate>
ate>
ate> dot3_test_init
wme_init success, handle = 1
ate>
```

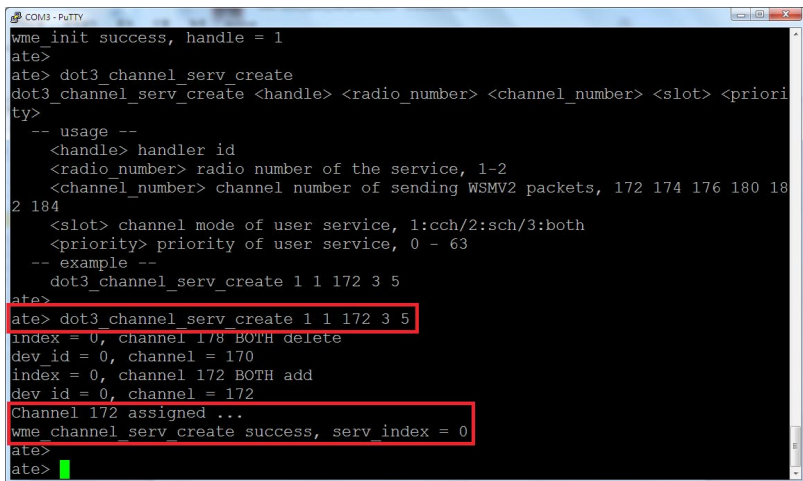
Similarly, on OBU-201 unit 2, initialize 1609.3 test environment (as shown in step 1).

**Step 2:** On OBU-201 unit 1, create a channel service to receive WSM

**Example:** `dot3_channel_serv_create 1 1 172 3 5`

**Description:** This command creates a channel service on Channel 172 using radio number 1.

**Sample snapshot:**



```
COM3 - PuTTY
wme_init success, handle = 1
ate>
ate> dot3_channel_serv_create
dot3_channel_serv_create <handle> <radio_number> <channel_number> <slot> <priority>
-- usage --
<handle> handler id
<radio_number> radio number of the service, 1-2
<channel_number> channel number of sending WSMV2 packets, 172 174 176 180 182 184
<slot> channel mode of user service, 1:cch/2:sch/3:both
<priority> priority of user service, 0 - 63
-- example --
dot3_channel_serv_create 1 1 172 3 5
ate>
ate> dot3 channel serv create 1 1 172 3 5
index = 0, channel 178 BOTH delete
dev id = 0, channel = 170
index = 0, channel 172 BOTH add
dev id = 0, channel = 172
Channel 172 assigned ...
wme channel_serv_create success, serv_index = 0
ate>
ate>
```

Similarly, on OBU-201 unit 2, create a channel service to send WSM (as shown in step 2).

**Note:** A channel service can be deleted using `<handle>` and `<serv_index>`.

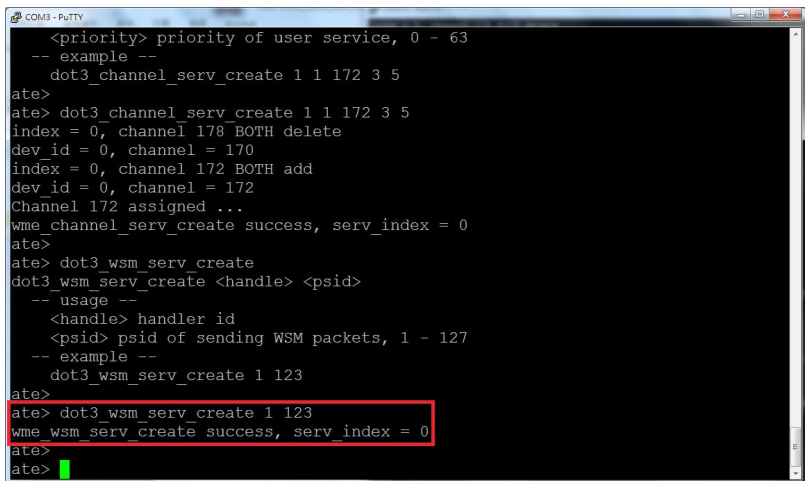
**Example:** `dot3_channel_serv_delete 1 0`

**Step 3:** On OBU-201 unit 1, create a WSM service

**Example:** `dot3_wsm_serv_create 1 123`

**Description:** This command will receive all WSM packets with PSID 123

**Sample snapshot:**

A screenshot of a PuTTY terminal window titled 'COM3 - PuTTY'. The terminal shows a series of commands and their outputs. The command 'dot3\_wsm\_serv\_create 1 123' is entered, and the output 'wme\_wsm\_serv\_create success, serv\_index = 0' is displayed. This line of output is highlighted with a red rectangular box. The terminal also shows other commands like 'dot3\_channel\_serv\_create' and their corresponding outputs, including channel assignments and service creation success messages.

```
<priority> priority of user service, 0 - 63
-- example --
dot3_channel_serv_create 1 1 172 3 5
ate>
ate> dot3_channel_serv_create 1 1 172 3 5
index = 0, channel 178 BOTH delete
dev_id = 0, channel = 170
index = 0, channel 172 BOTH add
dev_id = 0, channel = 172
Channel 172 assigned ...
wme_channel_serv_create success, serv_index = 0
ate>
ate> dot3_wsm_serv_create
dot3_wsm_serv_create <handle> <psid>
-- usage --
<handle> handler id
<psid> psid of sending WSM packets, 1 - 127
-- example --
dot3_wsm_serv_create 1 123
ate>
ate> dot3_wsm_serv_create 1 123
wme_wsm_serv_create success, serv_index = 0
ate>
ate>
```

**Note:** A WSM service can also be deleted using <handle> and <serv\_index>.

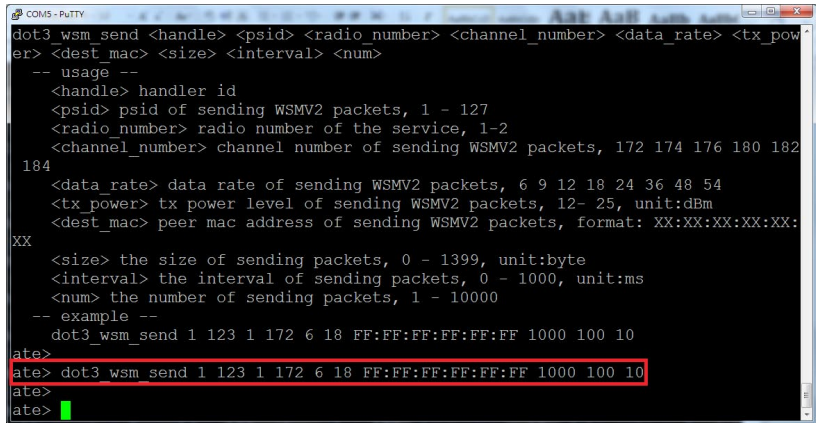
**Example:** `dot3_wsm_serv_delete 1 0`

On OBU-201 unit 2, send WSM packets using radio number 1

**Example:** `dot3_wsm_send 1 123 1 172 6 18 FF:FF:FF:FF:FF:FF 1000 100 10`

**Description:** This command sends 10 WSM packets with PSID 123 on Channel 172 using radio number 1.

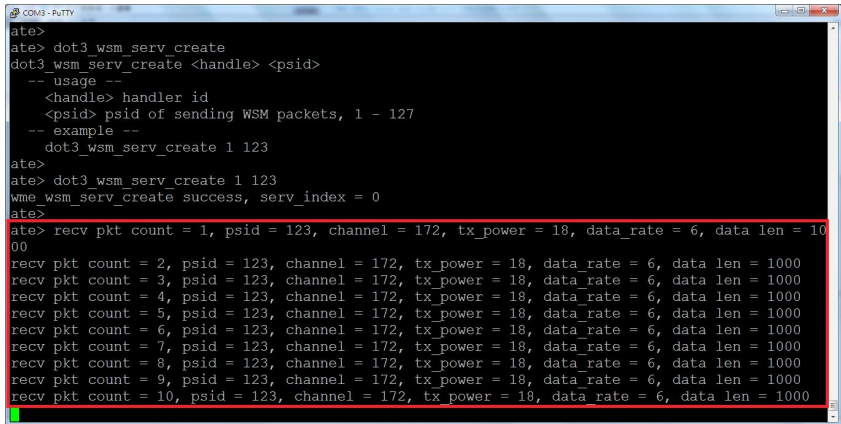
### Sample snapshot:



```
COM5 - PuTTY
dot3_wsm_send <handle> <psid> <radio_number> <channel_number> <data_rate> <tx_power>
er> <dest_mac> <size> <interval> <num>
-- usage --
<handle> handler id
<psid> psid of sending WSMV2 packets, 1 - 127
<radio_number> radio number of the service, 1-2
<channel_number> channel number of sending WSMV2 packets, 172 174 176 180 182
184
<data_rate> data rate of sending WSMV2 packets, 6 9 12 18 24 36 48 54
<tx_power> tx power level of sending WSMV2 packets, 12- 25, unit:dBm
<dest_mac> peer mac address of sending WSMV2 packets, format: XX:XX:XX:XX:XX:
XX
<size> the size of sending packets, 0 - 1399, unit:byte
<interval> the interval of sending packets, 0 - 1000, unit:ms
<num> the number of sending packets, 1 - 10000
-- example --
dot3_wsm_send 1 123 1 172 6 18 FF:FF:FF:FF:FF:FF 1000 100 10
ate>
ate> dot3_wsm_send 1 123 1 172 6 18 FF:FF:FF:FF:FF:FF 1000 100 10
ate>
ate>
```

After execution of above steps, 10 WSM packets are successfully received on OBU-201 unit 1, as shown below

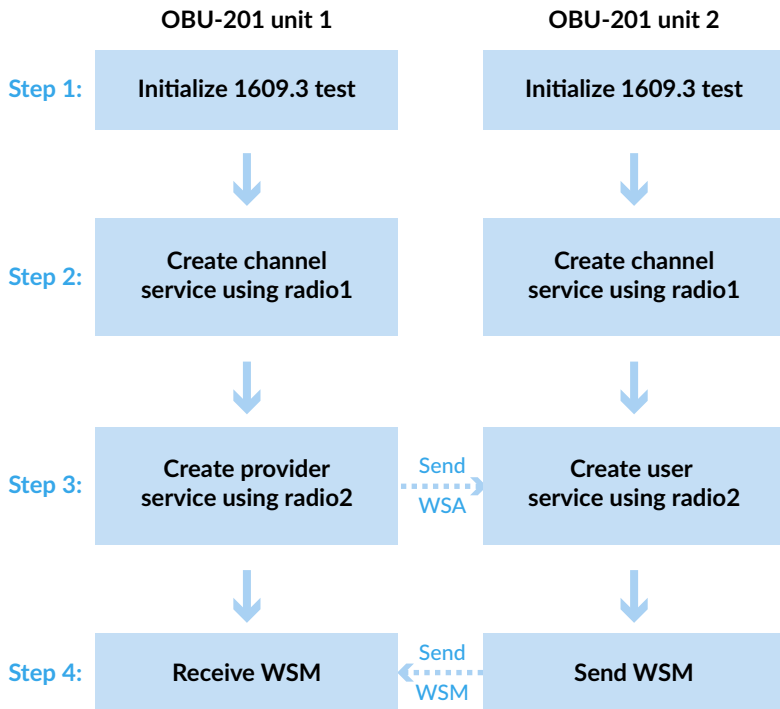
(*recv pkt count* = 10):



```
COM3 - PuTTY
ate>
ate> dot3_wsm_serv_create
dot3_wsm_serv_create <handle> <psid>
-- usage --
  <handle> handler id
  <psid> psid of sending WSM packets, 1 - 127
-- example --
dot3_wsm_serv_create 1 123
ate>
ate> dot3_wsm_serv_create 1 123
wme_wsm_serv_create success, serv_index = 0
ate>
ate> recv_pkt count = 1, psid = 123, channel = 172, tx_power = 18, data_rate = 6, data len = 1000
recv_pkt count = 2, psid = 123, channel = 172, tx_power = 18, data_rate = 6, data len = 1000
recv_pkt count = 3, psid = 123, channel = 172, tx_power = 18, data_rate = 6, data len = 1000
recv_pkt count = 4, psid = 123, channel = 172, tx_power = 18, data_rate = 6, data len = 1000
recv_pkt count = 5, psid = 123, channel = 172, tx_power = 18, data_rate = 6, data len = 1000
recv_pkt count = 6, psid = 123, channel = 172, tx_power = 18, data_rate = 6, data len = 1000
recv_pkt count = 7, psid = 123, channel = 172, tx_power = 18, data_rate = 6, data len = 1000
recv_pkt count = 8, psid = 123, channel = 172, tx_power = 18, data_rate = 6, data len = 1000
recv_pkt count = 9, psid = 123, channel = 172, tx_power = 18, data_rate = 6, data len = 1000
recv_pkt count = 10, psid = 123, channel = 172, tx_power = 18, data_rate = 6, data len = 1000
```

## 4. WSM – Tx & Rx Demo (Using Channel, Provider, User, WSM services)

### Overview:



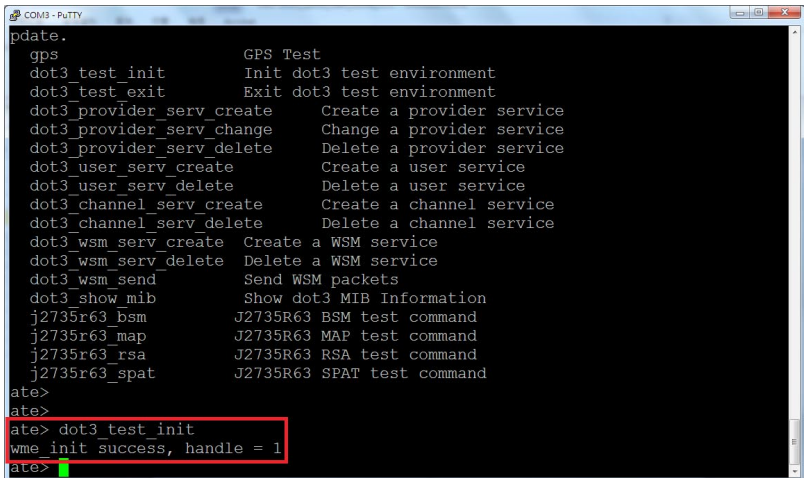
**Note:** This demo requires configuring two OBU-201 units. Both radio number 1 and 2 are used. For detailed command description, check section 8 (Appendix).

**Step 1:** On OBU-201 unit 1, initialize 1609.3 test environment

**Example:** `dot3_test_init`

**Description:** This command will get a handler id from 1609.3 to register other services such as: Provider, User, Channel, WSM

**Sample snapshot:**



```
COM3 - PUTTY
pdate.
gps                GPS Test
dot3_test_init     Init dot3 test environment
dot3_test_exit     Exit dot3 test environment
dot3_provider_serv_create  Create a provider service
dot3_provider_serv_change  Change a provider service
dot3_provider_serv_delete  Delete a provider service
dot3_user_serv_create      Create a user service
dot3_user_serv_delete      Delete a user service
dot3_channel_serv_create   Create a channel service
dot3_channel_serv_delete   Delete a channel service
dot3_wsm_serv_create       Create a WSM service
dot3_wsm_serv_delete       Delete a WSM service
dot3_wsm_send              Send WSM packets
dot3_show_mib              Show dot3 MIB Information
j2735r63_bsm              J2735R63 BSM test command
j2735r63_map              J2735R63 MAP test command
j2735r63_rsa              J2735R63 RSA test command
j2735r63_spat             J2735R63 SPAT test command
ate>
ate>
ate> dot3_test_init
wme init success, handle = 1
ate>
```

Similarly, on OBU-201 unit 2, initialize 1609.3 test environment (as shown in step 1).

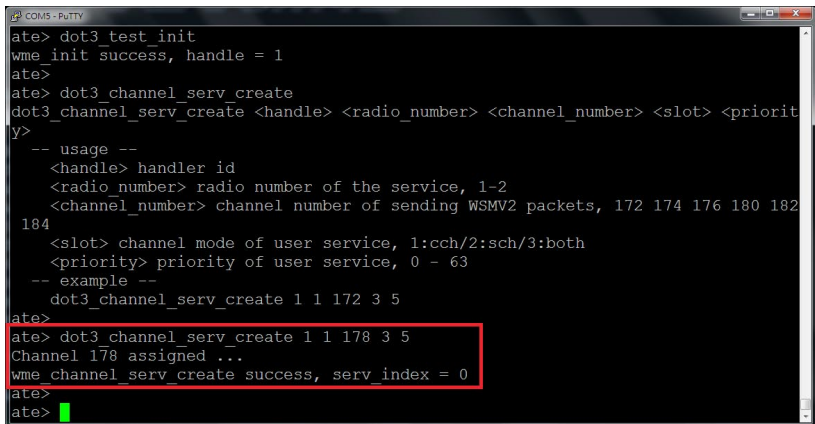


**Step 2:** On OBU-201 unit 1, create a channel service to send WSA

**Example:** `dot3_channel_serv_create 1 1 178 3 5`

**Description:** This command creates a channel service on Channel 178 using radio number 1.

**Sample snapshot:**

A screenshot of a terminal window titled 'COM3 - PUTTY'. The terminal shows the following sequence of commands and outputs:

```
ate> dot3 test init
wme_init success, handle = 1
ate>
ate> dot3_channel_serv_create
dot3_channel_serv_create <handle> <radio_number> <channel_number> <slot> <priority>
y>
-- usage --
<handle> handler id
<radio_number> radio number of the service, 1-2
<channel_number> channel number of sending WSMV2 packets, 172 174 176 180 182
184
<slot> channel mode of user service, 1:cch/2:sch/3:both
<priority> priority of user service, 0 - 63
-- example --
dot3_channel_serv_create 1 1 172 3 5
ate>
ate> dot3 channel serv create 1 1 178 3 5
Channel 178 assigned ...
wme_channel_serv_create success, serv_index = 0
ate>
ate>
```

The last three lines of the terminal output are highlighted with a red rectangular box.

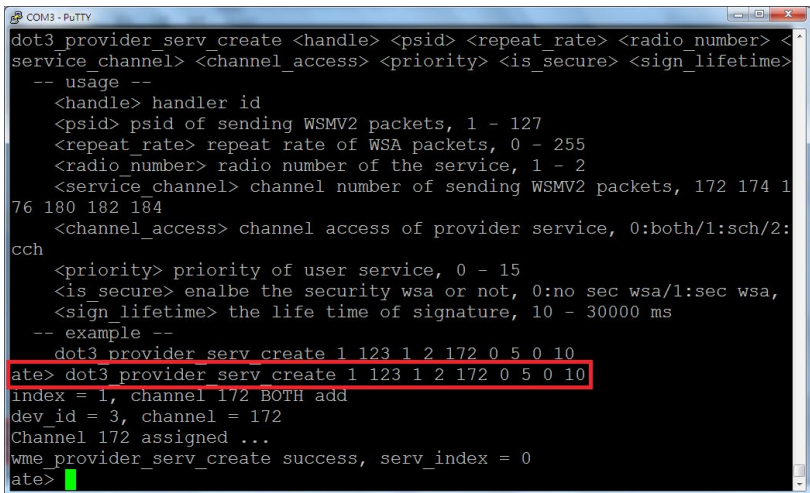
On OBU-201 unit 2, create a channel service to receive WSA (as shown in step 2).

**Step 3:** On OBU-201 unit 1, create a Provider service to send WSA

**Example:** `dot3_provider_serv_create 1 123 1 2 172 0 5 0 10`

**Description:** This command will send WSA on channel 178 and configure channel 172 to receive WSM using radio number 2.

**Sample snapshot:**



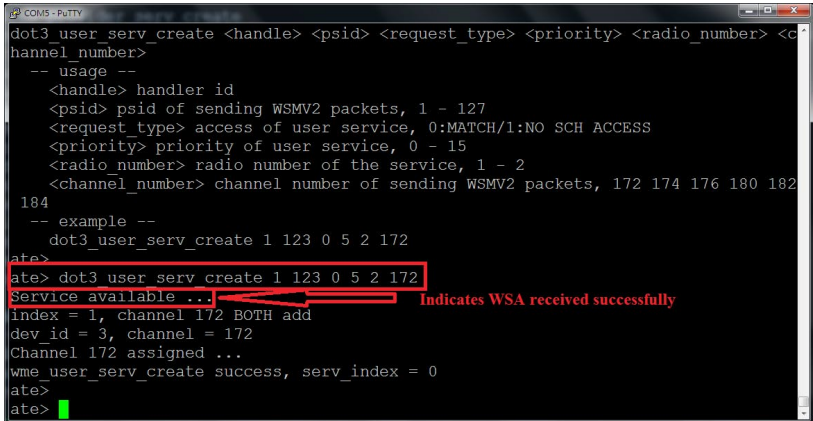
```
COM3 - PuTTY
dot3_provider_serv_create <handle> <psid> <repeat_rate> <radio number> <
service_channel> <channel_access> <priority> <is_secure> <sign_lifetime>
-- usage --
<handle> handler id
<psid> psid of sending WSMV2 packets, 1 - 127
<repeat_rate> repeat rate of WSA packets, 0 - 255
<radio number> radio number of the service, 1 - 2
<service_channel> channel number of sending WSMV2 packets, 172 174 1
76 180 182 184
<channel_access> channel access of provider service, 0:both/1:sch/2:
cch
<priority> priority of user service, 0 - 15
<is_secure> enable the security wsa or not, 0:no sec wsa/1:sec wsa,
<sign_lifetime> the life time of signature, 10 - 30000 ms
-- example --
dot3_provider_serv_create 1 123 1 2 172 0 5 0 10
ate> dot3_provider_serv_create 1 123 1 2 172 0 5 0 10
index = 1, channel 172 BOTH add
dev_id = 3, channel = 172
Channel 172 assigned ...
wme_provider_serv_create success, serv_index = 0
ate>
```

On OBU-201 unit 2, create a User service to receive WSA

**Example:** `dot3_user_serv_create 1 123 0 5 2 172`

**Description:** This command will receive WSA (if PSID & channel number match) and configure channel 172 to send WSM using radio number 2.

## Sample snapshot:



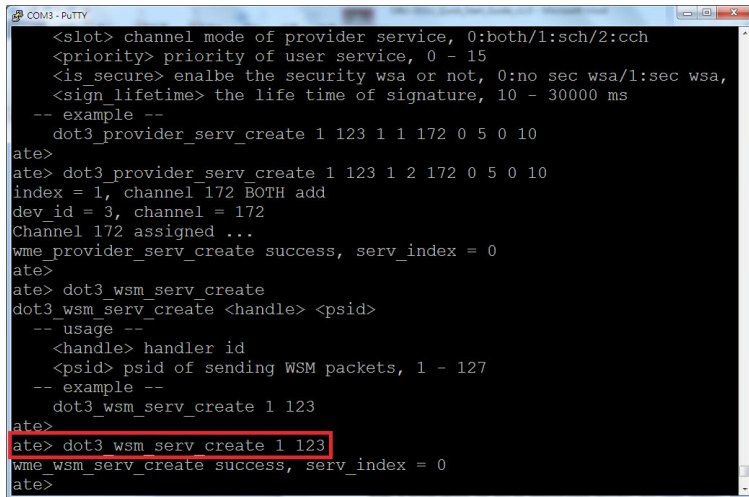
```
COM5 - PuTTY
dot3_user_serv_create <handle> <psid> <request_type> <priority> <radio_number> <channel_number>
-- usage --
  <handle> handler id
  <psid> psid of sending WSMV2 packets, 1 - 127
  <request_type> access of user service, 0:MATCH/1:NO SCH ACCESS
  <priority> priority of user service, 0 - 15
  <radio number> radio number of the service, 1 - 2
  <channel_number> channel number of sending WSMV2 packets, 172 174 176 180 182
184
-- example --
dot3_user_serv_create 1 123 0 5 2 172
ate>
ate> dot3 user serv create 1 123 0 5 2 172
Service available ... Indicates WSA received successfully
index = 1, channel 172 BOTH add
dev_id = 3, channel = 172
Channel 172 assigned ...
wme user_serv_create success, serv_index = 0
ate>
ate>
```

**Step 4:** On OBU-201 unit 1, create a WSM service

**Example:** `dot3_wsm_serv_create 1 123`

**Description:** This command will receive all WSM packets with PSID 123

## Sample snapshot:



```
COM3 - PuTTY
<slot> channel mode of provider service, 0:both/1:sch/2:cch
<priority> priority of user service, 0 - 15
<is_secure> enable the security wsa or not, 0:no sec wsa/1:sec wsa,
<sign lifetime> the life time of signature, 10 - 30000 ms
-- example --
dot3_provider_serv_create 1 123 1 1 172 0 5 0 10
ate>
ate> dot3_provider_serv_create 1 123 1 2 172 0 5 0 10
index = 1, channel 172 BOTH add
dev_id = 3, channel = 172
Channel 172 assigned ...
wme_provider_serv_create success, serv_index = 0
ate>
ate> dot3_wsm_serv_create
dot3_wsm_serv_create <handle> <psid>
-- usage --
<handle> handler id
<psid> psid of sending WSM packets, 1 - 127
-- example --
dot3_wsm_serv_create 1 123
ate>
ate> dot3 wsm_serv create 1 123
wme_wsm_serv_create success, serv_index = 0
ate>
```

On OBU-201 unit 2, send WSM packets using radio number 2

**Example:** *dot3\_wsm\_send 1 123 2 172 6 18 FF:FF:FF:FF:FF:FF 1000 100 10*

**Description:** This command sends 10 WSM packets with PSID 123 on Channel 172 using radio number 2.

## Sample snapshot:

```
COM3 - PUTTY
ate> dot3_wsm_send
dot3_wsm_send <handle> <psid> <radio_number> <channel_number> <data_rate> <tx_power>
er> <dest_mac> <size> <interval> <num>
-- usage --
<handle> handler id
<psid> psid of sending WSMV2 packets, 1 - 127
<radio number> radio number of the service, 1-2
<channel_number> channel number of sending WSMV2 packets, 172 174 176 180 182
184
<data_rate> data rate of sending WSMV2 packets, 6 9 12 18 24 36 48 54
<tx_power> tx power level of sending WSMV2 packets, 12- 25, unit:dBm
<dest_mac> peer mac address of sending WSMV2 packets, format: XX:XX:XX:XX:XX:
XX
<size> the size of sending packets, 0 - 1399, unit:byte
<interval> the interval of sending packets, 0 - 1000, unit:ms
<num> the number of sending packets, 1 - 10000
-- example --
dot3_wsm_send 1 123 1 172 6 18 FF:FF:FF:FF:FF:FF 1000 100 10
ate> dot3_wsm_send 1 123 2 172 6 18 FF:FF:FF:FF:FF:FF 1000 100 10
ate>
ate>
```

After execution of above steps, 10 WSM packets are successfully received on OBU-201 unit 1, as shown below

(*recv\_pkt count* = 10):

```
COM3 - PUTTY
ate> dot3_wsm_serv_create
dot3_wsm_serv_create <handle> <psid>
-- usage --
<handle> handler id
<psid> psid of sending WSM packets, 1 - 127
-- example --
dot3_wsm_serv_create 1 123
ate>
ate> dot3_wsm_serv_create 1 123
wsm_wsm_serv_create success, serv_index = 0
ate>
ate> recv_pkt count = 1, psid = 123, channel = 172, tx_power = 18, data_rate = 6, data len = 1000
recv_pkt count = 2, psid = 123, channel = 172, tx_power = 18, data_rate = 6, data len = 1000
recv_pkt count = 3, psid = 123, channel = 172, tx_power = 18, data_rate = 6, data len = 1000
recv_pkt count = 4, psid = 123, channel = 172, tx_power = 18, data_rate = 6, data len = 1000
recv_pkt count = 5, psid = 123, channel = 172, tx_power = 18, data_rate = 6, data len = 1000
recv_pkt count = 6, psid = 123, channel = 172, tx_power = 18, data_rate = 6, data len = 1000
recv_pkt count = 7, psid = 123, channel = 172, tx_power = 18, data_rate = 6, data len = 1000
recv_pkt count = 8, psid = 123, channel = 172, tx_power = 18, data_rate = 6, data len = 1000
recv_pkt count = 9, psid = 123, channel = 172, tx_power = 18, data_rate = 6, data len = 1000
recv_pkt count = 10, psid = 123, channel = 172, tx_power = 18, data_rate = 6, data len = 1000
ate>
```

## 5. Checking MIB status

To retrieve MIB setting, we have the following primitive:

```
dot3_show_mib
```

### **Illustration:**

On OBU, initialize 1609.3 test and create a channel service

```
ate> dot3_test_init
```

```
wme_init success, handle = 1
```

```
ate> dot3_channel_serv_create 1 1 172 3 5
```

```
index = 0, channel 178 BOTH delete
```

```
dev_id = 0, channel = 170
```

```
index = 0, channel 172 BOTH add
```

```
dev_id = 0, channel = 172
```

```
Channel 172 assigned ...
```

```
wme_channel_serv_create success, serv_index = 0
```

```
ate>
```

### **View MIB status:**

```
ate> dot3_show_mib

wme_init success, handle = 2

*** Provider entry ***

*** User entry ***

*** Channel entry ***

index = 0

priority = 5

channel = 172

slot = 3

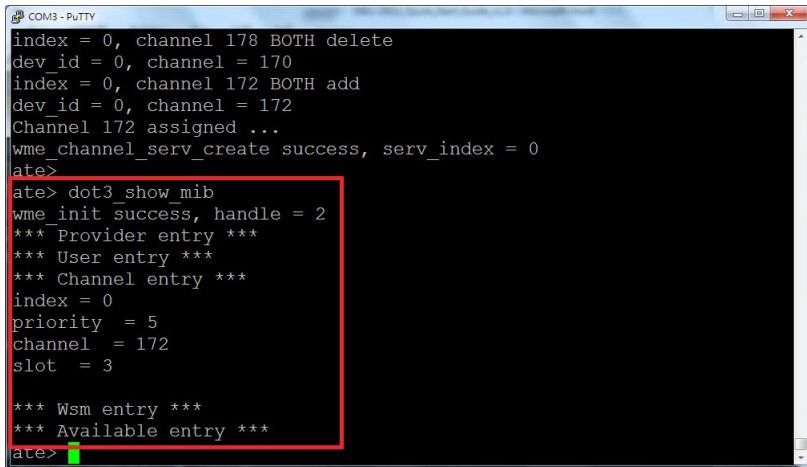
*** Wsm entry ***

*** Available entry ***

ate>
```

As shown above, we can see a Channel entry in MIB.

## Sample snapshot:



```
COM3 - PUTTY
index = 0, channel 178 BOTH delete
dev_id = 0, channel = 170
index = 0, channel 172 BOTH add
dev_id = 0, channel = 172
Channel 172 assigned ...
wme_channel_serv_create success, serv_index = 0
ate>
ate> dot3 show mib
wme_init success, handle = 2
*** Provider entry ***
*** User entry ***
*** Channel entry ***
index = 0
priority = 5
channel = 172
slot = 3

*** Wsm entry ***
*** Available entry ***
ate>
```



## 6. BSM – Tx & Rx Demo

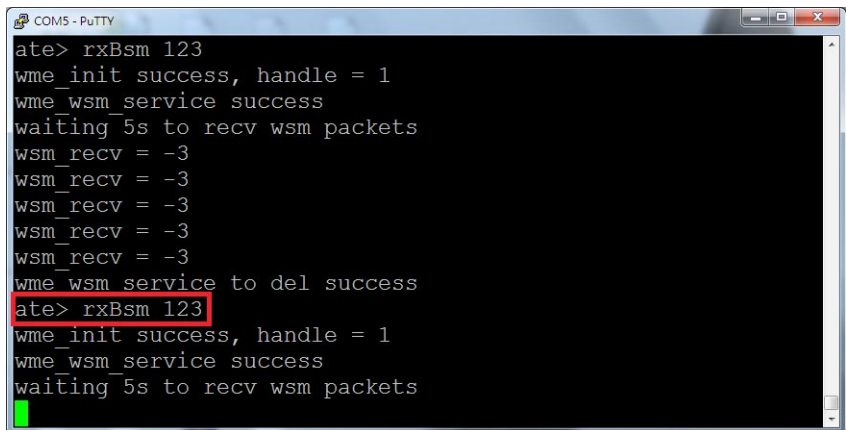
**Note:** This demo requires configuring two OBU-201 units. For detailed command description, check section 8 (Appendix) and for acronyms used, check abbreviations on Page 2.

**Step 1:** On OBU-201 unit 1, receive BSM packets

**Example:** `rxBsm 123`

**Description:** This command will start to receive BSM packets with matching PSID (for 5 seconds)

**Sample snapshot:**



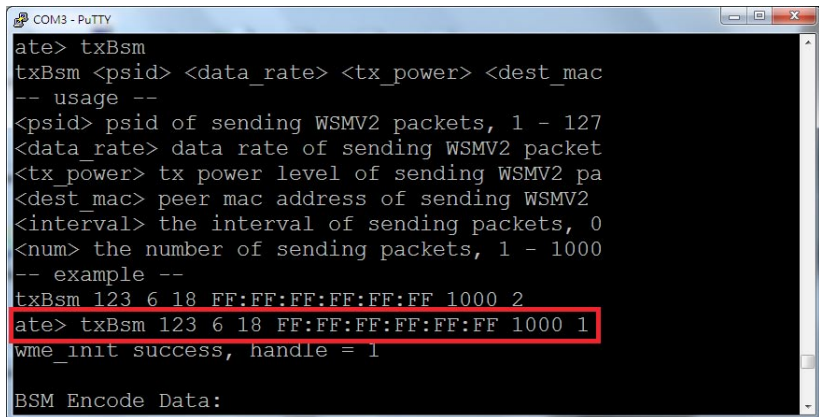
```
COM5 - PuTTY
ate> rxBsm 123
wme_init success, handle = 1
wme_wsm_service success
waiting 5s to recv wsm packets
wsm_rcv = -3
wsm_rcv = -3
wsm_rcv = -3
wsm_rcv = -3
wsm_rcv = -3
wme_wsm_service to del success
ate> rxBsm 123
wme_init success, handle = 1
wme_wsm_service success
waiting 5s to recv wsm packets
```

**Step 2:** On OBU-201 unit 2, send BSM packets

**Example:** `txBsm 123 6 18 FF:FF:FF:FF:FF:FF 1000 1`

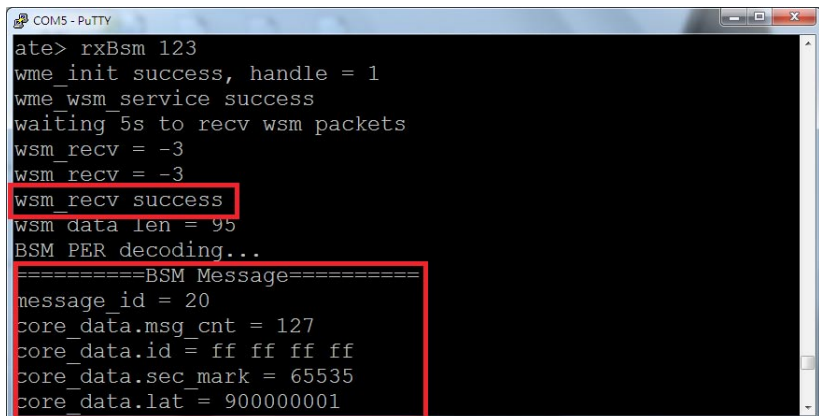
**Description:** This command will send 1 BSM packet with PSID 123

**Sample snapshot:**



```
COM3 - PuTTY
ate> txBsm
txBsm <psid> <data_rate> <tx_power> <dest_mac>
-- usage --
<psid> psid of sending WSMV2 packets, 1 - 127
<data_rate> data rate of sending WSMV2 packet
<tx_power> tx power level of sending WSMV2 pa
<dest_mac> peer mac address of sending WSMV2
<interval> the interval of sending packets, 0
<num> the number of sending packets, 1 - 1000
-- example --
txBsm 123 6 18 FF:FF:FF:FF:FF:FF 1000 2
ate> txBsm 123 6 18 FF:FF:FF:FF:FF:FF 1000 1
wme_init success, handle = 1
BSM Encode Data:
```

After execution of step 1 and step 2, we have successfully received BSM packet:



```
COM5 - PuTTY
ate> rxBsm 123
wme_init success, handle = 1
wme_wsm_service success
waiting 5s to recv wsm packets
wsm_rcv = -3
wsm_rcv = -3
wsm_rcv success
wsm_data len = 95
BSM PER decoding...
=====BSM Message=====
message_id = 20
core_data.msg_cnt = 127
core_data.id = ff ff ff ff
core_data.sec_mark = 65535
core_data.lat = 900000001
```

Similarly, we have all the other J2735 message sets following the same syntax, viz.:

SPAT/MAP/TIM/RSA/RTCM/CSR/NMEA/ICA/EVA/PDM/SSM/  
SRM/PVD/PSM.

## 7. Firmware Upgrade

1. Setup a serial connection to the OBU-201 unit
2. Ensure that the unit is connected to an Ethernet LAN. Also ensure that PC and unit are on the same subnet
3. Reset the unit and enter U-Boot console by pressing any key during the 3 second countdown. You should see the U-Boot console prompt:

```
U-Boot>
```

4. To check the IP address of the unit:

```
U-Boot> printenv (look for ipaddr field)
```

5. To modify the IP address:

```
U-Boot> setenv ipaddr 10.10.10.x
```

6. We will assume the IP address of your TFTP server is 10.10.10.10. Perform ping test

```
U-Boot> ping 10.10.10.10
```

7. If the ping was successful, you should see output similar to:

```
Link: UP
```

```
Duplex: FULL
```

```
Speed 100BASE-X
```

*Using device*

*host 10.10.0.10 is alive*

8. Configure the TFTP server's IP address & gateway ip address:

*U-Boot> setenv serverip 10.10.10.10*

*U-Boot> setenv gatewayip 10.10.10.254*

9. We will assume that the image you would like to boot is served by the TFTP server under the name uImage. Issue the following command to load the image via TFTP:

*U-Boot> tftp uImage*

10. Erasing flash memory is done differently depending on flash size. When using a 8 MB flash device:

*U-Boot> protect off 80000 7fffff*

*U-Boot> erase 80000 7fffff*

11. When using a 32 MB flash device:

*U-Boot> mw.l 0x41016850 0x001f0000*

*U-Boot> protect off 80000 1fdfffff*

*U-Boot> erase 80000 1fdfffff*

12. Replace existing firmware image with the new one:

*U-Boot> cp.b \${fileaddr} 80000 \${filesize}*

```
U-Boot> setenv bootcmd cp.b 80000 50000000  
${filesize}\; bootm
```

```
U-Boot> saveenv
```

13. You should see output similar to:

```
U-Boot> tftp uImage
```

```
Link: UP
```

```
Duplex: FULL
```

```
Speed 100BASE-X
```

```
Using device
```

```
TFTP from server 10.10.0.10; our IP address is  
10.10.0.50
```

```
Filename 'uImage'.
```

```
Load address: 0x50000000
```

```
Loading: TftpRemotePort=69
```

```
#####  
#####
```

```
#####
```

```
done
```

```
Bytes transferred = 552100 (86ca4 hex)
```

*U-Boot> protect off 80000 7fffff*

*.....  
..... done*

*Un-Protected 120 sectors*

*U-Boot> erase 80000 7fffff*

*.....  
..... done*

*Erased 120 sectors*

*U-Boot> cp.b \${fileaddr} 80000 \${filesize}*

*Copy to Flash... done*

*U-Boot> setenv bootcmd cp.b 80000 50000000  
\${filesize}\; bootm*

*U-Boot> saveenv*

*Saving Environment to Flash...*

*.. done*

*Un-Protected 2 sectors*

*Erasing Flash...*

*.. done*

*Erased 2 sectors*

*Writing to Flash... done*

*.. done*

*Protected 2 sectors*

*Done*

14. Reset the unit via the command:

*U-Boot> reset*



# 8. Appendix

## A: Channel service

To create a channel service, we have the following primitive:

*dot3\_channel\_serv\_create*

**Syntax:** *dot3\_channel\_serv\_create* <handle>

<radio\_number> <channel\_number> <slot>  
<priority>

The parameters are described in the following table:

Parameter	Valid Range	Description
handle	0 ~ 255	Handler id
radio_number	1 - 2	the radio number
channel_number	172 174 176 178 180 182 184	The channel that users want to reserve
slot	1: CCH 2: SCH 3: BOTH	The time slot for the desired channel

priority	0 ~ 15	Priority of user service. Lower value has higher priority
----------	--------	---

**Example:** `dot3_channel_serv_create 1 1 172 3 5`

To delete a channel service, we have the following primitive:

`dot3_channel_serv_delete`

**Syntax:** `dot3_channel_serv_delete <handle>`

`<serv_index>`

NOTE: The parameters <handle> and <serv\_index> are assigned by the system and are not configurable by the user.

**Example:** `dot3_channel_serv_delete 1 0`

## B: Provider service

To create a provider service, we have the following primitive:

`dot3_provider_serv_create`

**Syntax:** `dot3_provider_serv_create <handle> <psid>  
<repeat_rate> <radio_number> <service_channel>  
<channel_access> <priority> <is_secure>  
<sign_lifetime>`

The parameters are described in the following table:

Parameter	Valid Range	Description
handle	0 ~ 255	Handler id
psid	1 ~ 127	psid of desired service
repeat rate	0 ~ 255	repeat rate of WSA packets
radio_number	1 - 2	radio number of the service
service_channel	172 174 176 180 182 184	Channel number of desired service
channel_access	1: CCH 2: SCH 3: BOTH	The time slot for the desired channel
priority	0 ~ 15	Priority of user service. Lower value has higher priority
is_secure	0: no secured WSA 1: secured WSA	Security enabling status
sign_lifetime	10 ~ 30000 ms	life time of signature

**Example:** `dot3_provider_serv_create 1 123 1 1 172 0 5 0 10`

To delete a provider service, we have the following primitive:

*dot3\_provider\_serv\_delete*

**Syntax:** *dot3\_provider\_serv\_delete* <handle>  
<serv\_index>

NOTE: The parameters <handle> and <serv\_index> are assigned by the system and are not configurable by the user.

**Example:** *dot3\_provider\_serv\_delete 1 0*

## C: User service

To create a user service, we have the following primitive:

*dot3\_user\_serv\_create*

**Syntax:** *dot3\_user\_serv\_create* <handle> <psid>  
<request\_type> <priority> <radio\_number>  
<channel\_number>

The parameters are described in the following table:

Parameter	Valid Range	Description
handle	0 ~ 255	Handler id
psid	1 ~ 127	psid of desired service
repeat rate	0: MATCH 1: NO SCH Access	Access mode of user service
priority	0 ~ 15	Priority of user service. Lower value has higher priority
radio_number	1 - 2	radio number of the service
channel_number	172 174 176 180 182 184	Channel number of desired service

**Example:** *dot3\_user\_serv\_create 1 123 0 5 2 172*

To delete a user service, we have the following primitive:

*dot3\_user\_serv\_delete*

**Syntax:** *dot3\_user\_serv\_delete <handle> <serv\_index>*

NOTE: The parameters <handle> and <serv\_index> are assigned by the system and are not configurable by the user.

**Example:** *dot3\_user\_serv\_delete 1 0*

# D: WSM service

To create a WSM service, we have the following primitive:

```
dot3_wsm_serv_create
```

**Syntax:** *dot3\_wsm\_serv\_create* <handle> <psid>

The parameters are described in the following table:

Parameter	Valid Range	Description
handle	0 ~ 255	Handler id
psid	1 ~ 127	psid of desired service

**Example:** *dot3\_wsm\_serv\_create 1 123*

To delete a wsm service, we have the following primitive:

```
dot3_wsm_serv_delete
```

# E: BSM Tx & Rx

To send a BSM packet, we have the following primitive: *txBsm*

**Syntax:** *txBsm* <psid> <data\_rate> <tx\_power>  
<dest\_mac> <interval> <num>

The parameters are described in the following table:

Parameter	Valid Range	Description
psid	1 ~ 127	psid of desired service
data_rate	6 9 12 18 24 36 48 54	data rate of sending WSM packets
tx_power	12- 25, unit:dBm	tx power level of sending WSM packets
dest_mac	format: XX:XX:XX:XX:XX:XX	peer mac address of sending WSM packets
interval	0 - 1000, unit:ms	the interval of sending packets
num	1 - 10000	the number of sending packets

**Example:** *txBsm 123 6 18 FF:FF:FF:FF:FF:FF 1000 1*

To receive BSM packets, we have the following primitive: *rxBsm*

**Syntax:** *rxBsm <psid>*

**Example:** *rxBsm 123*

**Note:** Syntax is the same for other J2735 message sets which are:

SPAT/MAP/TIM/RSA/RTCM/CSR/NMEA/ICA/EVA/PDM/SSM/  
SRM/PVD/PSM.

Unex Technology Corporation  
11F-3, No. 100, Sec. 1, Jiafeng 11th Rd., Zhubei City,  
Hsinchu County 302, Taiwan, R.O.C.  
Tel. 886-3-657-8188  
Fax. 886-3-657-8199

[www.unex.com.tw](http://www.unex.com.tw)