Host RPU Comm Interface

Copyright © Imagination Technologies Limited. All Rights Reserved.

‑This document is confidential. Neither the whole nor any part of the information contained in, nor the product described in, this document may be adapted or reproduced in any material form except with the written permission of Imagination Technologies Limited. Imagination Technologies, the Imagination logo, PowerVR, MIPS, Meta, Ensigma and Codescape are trademarks or registered trademarks of Imagination Technologies Limited. All other logos, products, trademarks and registered trademarks are the property of their respective owners. This document can only be distributed subject to the terms of a Non‑Disclosure Agreement or Licence with Imagination Technologies Limited.‑

Filename : Host RPU Comm Interface  
Version : 1.0.3 Not Issued - Live Document  
Issue Date : 25 Aug 2016  
Author : Imagination Technologies Limited

Contents

[1. Introduction 4](#_Toc459908061)

[1.1. Last Version Target Audience 4](#_Toc459908062)

[1.2. Feedback 4](#_Toc459908063)

[1.3. Glossary 4](#_Toc459908064)

[2. Host-Firmware Communication Overview 5](#_Toc459908065)

[2.1. Host-Firmware Message Format 5](#_Toc459908066)

[2.2. Commands 6](#_Toc459908067)

[2.3. Events 10](#_Toc459908068)

[Appendix A. Data Structures 13](#_Toc459908069)

[A.1. Commands 13](#_Toc459908070)

[A.1.1. Cmd\_reset 13](#_Toc459908071)

[A.1.2. Cmd\_scan 14](#_Toc459908072)

[A.1.3. Cmd\_scan\_abort 15](#_Toc459908073)

[A.1.4. Cmd\_set\_key 15](#_Toc459908074)

[A.1.5. Cmd\_tx\_ctrl 16](#_Toc459908075)

[A.1.6. Cmd\_frag 18](#_Toc459908076)

[A.1.7. Cmd\_rate 18](#_Toc459908077)

[A.1.8. Cmd\_disconnect 18](#_Toc459908078)

[A.1.9. Cmd\_ps 19](#_Toc459908079)

[A.1.10. Cmd\_vifctrl 19](#_Toc459908080)

[A.1.11. Cmd\_set\_beacon 20](#_Toc459908081)

[A.1.12. Cmd\_set\_mode 20](#_Toc459908082)

[A.1.13. Cmd\_ht\_ba 20](#_Toc459908083)

[A.1.14. Cmd\_vht\_beamform 21](#_Toc459908084)

[A.1.15. Cmd\_roc 21](#_Toc459908085)

[A.1.16. Cmd\_nw\_selection 22](#_Toc459908086)

[A.1.17. Cmd\_set\_defaultkey 22](#_Toc459908087)

[A.1.18. Cmd\_set\_rekey 22](#_Toc459908088)

[A.1.19. Cmd\_tx\_pwr 22](#_Toc459908089)

[A.1.20. Cmd\_mcst\_addr\_cfg 23](#_Toc459908090)

[A.1.21. Cmd\_mcst\_filter\_ctrl 23](#_Toc459908091)

[A.1.22. Cmd\_channel 23](#_Toc459908092)

[A.1.23. Cmd\_vif\_cfg 24](#_Toc459908093)

[A.1.24. Cmd\_sta 25](#_Toc459908094)

[A.1.25. Cmd\_txq\_params 26](#_Toc459908095)

[A.1.26. Cmd\_aux\_adc\_chain\_sel 27](#_Toc459908096)

[A.1.27. Cmd\_detect\_radar 27](#_Toc459908097)

[A.1.28. Umac\_cmd\_tx\_enable 27](#_Toc459908098)

[A.1.29. Cmd\_discard\_pkts 27](#_Toc459908099)

[A.1.30. Cmd\_msrmnt\_start 28](#_Toc459908100)

[A.1.31. Cmd\_bt\_info 28](#_Toc459908101)

[A.1.32. Cmd\_chanctx\_time\_config 28](#_Toc459908102)

[A.1.33. Cmd\_cont\_tx 28](#_Toc459908103)

[A.2. Events 29](#_Toc459908104)

[A.2.1. wlan\_rx\_pkt 29](#_Toc459908105)

[A.2.2. umac\_event\_tx\_done 30](#_Toc459908106)

[A.2.3. umac\_event\_ch\_prog\_complete 30](#_Toc459908107)

[A.2.4. umac\_event\_ch\_switch 31](#_Toc459908108)

[A.2.5. host\_event\_mgmt\_rx 31](#_Toc459908109)

[A.2.6. host\_event\_command\_complete 31](#_Toc459908110)

[A.2.7. host\_event\_connect\_result 31](#_Toc459908111)

[A.2.8. host\_event\_rssi 32](#_Toc459908112)

[A.2.9. host\_event\_rekey\_data 33](#_Toc459908113)

[A.2.10. umac\_event\_ch\_switch\_complete 33](#_Toc459908114)

[A.2.11. host\_event\_disconnect 34](#_Toc459908115)

[A.2.12. host\_event\_scanres 34](#_Toc459908116)

[A.2.13. host\_event\_reset\_complete 35](#_Toc459908117)

[A.2.14. umac\_event\_mib\_stats 35](#_Toc459908118)

[A.2.15. host\_event\_phy\_stats 38](#_Toc459908119)

[A.2.16. umac\_event\_noa 38](#_Toc459908120)

[A.2.17. umac\_event\_ps\_econ\_cfg\_complete 38](#_Toc459908121)

[A.2.18. umac\_event\_ps\_econ\_wake 39](#_Toc459908122)

[A.2.19. umac\_event\_mac\_stats 39](#_Toc459908123)

[A.2.20. umac\_event\_rf\_calib\_data 41](#_Toc459908124)

[A.2.21. host\_event\_radar\_detected 41](#_Toc459908125)

[A.2.22. host\_event\_msrmnt\_complete 41](#_Toc459908126)

[A.2.23. Umac\_event\_roc\_status 42](#_Toc459908127)

[A.3. User Defined Types 42](#_Toc459908128)

[A.3.1. CIPHER\_TYPE\_T 42](#_Toc459908129)

[A.3.2. KEY\_TYPE\_T 43](#_Toc459908130)

[A.3.3. RESET\_TYPE\_T 43](#_Toc459908131)

[A.3.4. POWER\_SAVE\_T 43](#_Toc459908132)

[A.3.5. SCAN\_TYPE\_T 43](#_Toc459908133)

[A.3.6. host\_mac\_msg\_hdr 44](#_Toc459908134)

[A.3.7. ssid\_t 44](#_Toc459908135)

[A.3.8. bgscan\_params 44](#_Toc459908136)

[A.3.9. bssres 45](#_Toc459908137)

[A.3.10. CONSTANTS 45](#_Toc459908138)

List of Figures

**No table of figures entries found.**

IF{{Internal}}

Document History

| Issue | Date | Changes/Comments |
| --- | --- | --- |
| 1.0.3 | 25 Aug 2016 | External Issue. |

END{{Internal}}

# Introduction

This document describes the command / event communication between Host and RPU Firmware. Also includes description of all commands and events, set of legal values that each field in command/event etc.

## Last Version Target Audience

This Document is intended for Host driver and firmware developers

## Feedback

If you have any comments regarding this document, please contact your Imagination Technologies representative. Please provide the document title and revision with a description of the problem or suggestion for improvement.

## Glossary

The following abbreviations are used in this document.

AP Access Point

BSS Basic Service Set

IBSS Independent BSS

MSDU MAC Service Data Unit

QoS Quality of Service

RSSI Receive Signal Strength Indicator

RX Reception

STA Station

TX Transmission

UAPSD Unscheduled Automatic Power save Delivery

IE Information Element

P2P Peer to Peer operation

SSID Service Set Identifier

NOA Notice of absence

MIB Management Information Base

HAL Hardware Abstraction Layer

BK Background priority traffic

BE Best effort priority traffic

VI Video priority traffic

VO Voice priority traffic

VIF Virtual Interface number

MIC Message Integrity Check

# 

# Host-Firmware Communication Overview

## Host-Firmware Message Format

The format of messages exchanged between host and RPU is given in the diagram below:

7

0

31

23

15

|  |  |
| --- | --- |
| HAL\_PRIV\_DATA | |
| Queue Number | Descriptor Number |
| Payload length | |
| Command/Event Data | |

Messages from host to firmware are called commands while those from firmware to host are called events. Every message has a reserved area at the top for use by the HAL. The size of this area is decided at compile time by the HAL in use. For example, the Hostport HAL may require 8 bytes here while a different HAL may not require this at all. Following the HAL reserved area, every command/event has 8 bytes in which the descriptor number, queue number and payload length fields are encoded as shown in the figure above. These fields are valid only when the command/event has a payload associated with it. The payload pointer itself is passed to the HAL as one of the arguments of the hal\_send () API. This is indicated when the descriptor number is encoded as a value other than 0xFFFF. The queue number, descriptor number and payload length are related to the payload and this information can be used by the HAL to maintain a mapping between descriptors and actual payloads. The actual command or event data follows the payload length field and its length can be variable. The little-endian format is assumed for all the commands and the events.

|  |  |  |
| --- | --- | --- |
| Field Name | Length (in Bytes) | Description |
| HAL\_PRIV\_DATA | HAL Dependent | For use by the HAL |
| Descriptor Number | 2 | Descriptor ID – If not 0xFFFF, indicates the descriptor ID of a command which has a payload |
| Queue Number | 2 | Queue Num: Indicates which queue this packet belongs to. Valid only when Descriptor Number is not equal to 0xFFFF |
| Payload length | 4 | The length of payload. Valid only when Descriptor Number is not equal to 0xFFFF |
| more\_cmd\_data | 4 | = 1 if command length exceeds MAX\_CMD\_SIZE.  =0 otherwise |
| id | 4 | Event or command id |
| length | 4 | Size of command or event |
| Command/Event Data | Command/Event dependent | Contains the actual Host<->RPU message content |

## Commands

A brief description of each command is given below. For more details of each command, click on the hyper link in **command structure** column.

|  |  |  |
| --- | --- | --- |
| **Command ID** | **Command Structure** | **Command description** |
| UMAC\_CMD\_RESET | Cmd\_reset | This command is used to enable/disable WLAN interface. A RESET\_COMPLETE event from the firmware indicates successful completion of this command. After sending the RESET, the host should not issue further commands until a RESET\_COMPLETE is received. |
| UMAC\_CMD\_SCAN | Cmd\_scan | This command is used to initiate a SCAN operation on the target. The results are returned in the SCAN\_COMPLETE event. |
| UMAC\_CMD\_SCAN\_ABORT | cmd\_scan\_abort | This command is used to abort the scan request raised from host . The results are returned in the SCAN\_ABORT\_COMPLETE event. |
| UMAC\_CMD\_CONNECT | Unused | Unused |
| UMAC\_CMD\_SETKEY | Cmd\_set\_key | This command is used to program WEP40/104/ pairwise, group keys when corresponding security modes are enabled. |
| UMAC\_CMD\_SET\_DEFAULT\_KEY | cmd\_set\_defaultkey | unused |
| UMAC\_CMD\_REKEY\_DATA | cmd\_set\_rekey | unused |
| UMAC\_CMD\_TX | Cmd\_tx\_ctrl | This command is used to transmit a TX packet. A TX packet is divided into two parts. First part consists of TX control information and mac header bytes. Second part is payload.  802.11 mac header and crypto header should are considered as first part. Second part consist data portion of payload. Fist part is copied to RPU memory by using host port interface and second part is fetched by using DMA engine.  TX done event is generated after completing packet transmission. |
| UMAC\_CMD\_MGMT\_TX | Unused | Unused |
| UMAC\_CMD\_FRAG | Unused | Unused |
| UMAC\_CMD\_TX\_POWER | **Error! Reference source not found.** | This command is used to specify max transmit power. It takes values from 0-20; units in dbm. |
| UMAC\_CMD\_RATE | Cmd\_rate | This command is used to specify the fixed rate. . It is in units of Mbps. |
| UMAC\_CMD\_DISCONNECT | Cmd\_disconnect | This command is used to inform the RPU that Host is not interested in communicating with AP. Roaming decision triggers this command. |
| UMAC\_CMD\_PS | Cmd\_ps | This command only applicable for BSS mode and will be ignored when not associated with the AP. |
| UMAC\_CMD\_PS\_ECON\_CFG | cmd\_ps | This command is used to indicate that the host is going to power save mode or waking up from power save. The results are returned in the PS\_ECON\_CFG\_DONE event. |
| UMAC\_CMD\_VIF\_CTRL | Cmd\_vifctrl | This command is used to add or remove a virtual interface. Maximum number of virtual interfaces supported in firmware is 2. |
| UMAC\_CMD\_SET\_BEACON | Cmd\_set\_beacon | This command is used to inform start of AP / IBSS mode operation. After this command, Host transmits a beacon after every beacon interval. This command is not used in station mode operation. |
| UMAC\_CMD\_SET\_MODE | Cmd\_set\_mode | This command is used to set the operation mode. Valid modes are Station, AP and IBSS. |
| UMAC\_CMD\_BA\_SESSION\_INFO | Cmd\_ht\_ba | This command is used to pass block ack parameters to RPU firmware. |
| UMAC\_CMD\_MCST\_ADDR\_CFG | **Error! Reference source not found.** | This command is used to add or remove a multi cast address to firmware. |
| UMAC\_CMD\_MCST\_FLTR\_CTRL | **Error! Reference source not found.** | This command is used to enable / disable  Multi cast filtering in firmware. If multicast filtering is enabled, then multi cast packets with MAC addresses added using UMAC\_CMD\_MCST\_ADDR\_CFG only are allowed. If disabled all multi cast packets are received by host. |
| UMAC\_CMD\_VHT\_BEAMFORM\_CTRL | cmd\_vht\_beamform | This command is used to pass beam forming related parameters . |
| UMAC\_CMD\_ROC\_CTRL | cmd\_roc | This command is used to pass the ROC related information to firmware. |
| UMAC\_CMD\_CHANNEL | Cmd\_channel | This command is used to program the channel. A UMAC\_EVENT\_CH\_PROG\_DONE event from firmware indicates completion of channel programming. |
| UMAC\_CMD\_VIF\_CFG | Cmd\_vif\_cfg | This command is used to change configuration parameters like basic rate set, power save mode etc., in firmware. This command is issued by the Host driver, whenever configuration parameters are changed. |
| UMAC\_CMD\_STA | Cmd\_sta | This command is used to add or remove station information in firmware. |
| UMAC\_CMD\_TXQ\_PARAMS | Cmd\_txq\_params | This command is used to set transmission queue parameters. There are five transmission queues; they are AC\_BK, AC\_BE, AC\_VI, AC\_VO and AC\_BCN queues. |
| UMAC\_CMD\_MIB\_STATS | There are no parameters in MIB stats and hence no structure is needed | This command is used to get MIB stats. These values can be retrieved at any point of time. |
| UMAC\_CMD\_PHY\_STATS | There are no parameters in MIB stats and hence no structure is needed | This command is used to get PHY stats. These values can be retrieved at any point of time. |
| UMAC\_CMD\_NW\_SELECTION | cmd\_nw\_selection | unused |
| UMAC\_CMD\_AUX\_ADC\_CHAIN\_SEL | cmd\_aux\_adc\_chain\_sel | This command is used to set aux path in phy. Used to calibrate power in production mode. This need to be called in only production mode |
| UMAC\_CMD\_DETECT\_RADAR | Cmd\_detect\_radar | This command is used to start radar detection. This is valid only when 802.11h (DFS) support is enabled. |
| UMAC\_CMD\_ENABLE\_TX | Umac\_cmd\_tx\_enable | This Command is used to enable TX packets after RADAR detection and CMD\_DISCARD\_PKTS. This is valid only when 802.11h (DFS) support is enabled. |
| UMAC\_CMD\_DISCARD\_PKTS | Cmd\_discard\_pkts | This command is used to DISCARD packets that are pending at firmware.  Firmware drops all packets and issue TX Done per each packet with reason code as Discard. Host driver should issue this command, when Radar is detected in current operating channel. TX packets are re enabled after UMAC\_CMD\_ENABLE\_TX command to firmware. This is valid only when 802.11h (DFS) support is enabled. |
| UMAC\_CMD\_MEASURE | Cmd\_msrmnt\_start | This command is used to start the channel measurement. It checks for radar detection and if radar is detected, channel measurement report is send to AP. This is valid only when 802.11h (DFS) support is enabled. |
| UMAC\_CMD\_BT\_INFO | cmd\_bt\_info | This command is used to to provide BT related information. |
| UMAC\_CMD\_CLEAR\_STATS | host\_mac\_msg\_hdr | This command is used to clear all the previous mac and phy stats to get fresh reading when get stats is used. |
| UMAC\_CMD\_CHANCTX\_TIME\_INFO | cmd\_chanctx\_time\_config | This command is used to pass the channel context info . |
| UMAC\_CMD\_CONT\_TX | cmd\_cont\_tx | This command is used in production mode for looping the tx packet. |

## Events

A brief description of each event is given below. For more details of each event, click on the hyper link in **event structure** column.

|  |  |  |
| --- | --- | --- |
| **Event** | **Event structure** | **Description** |
| UMAC\_EVENT\_RX | wlan\_rx\_pkt | This event is used to inform host about the RX packet |
| UMAC\_EVENT\_TX\_DONE | umac\_event\_tx\_done | This event is used to signal the completion of transmission of one or more packets. It also contains the transmission success/failure information for each packet. |
| UMAC\_EVENT\_DISCONNECTED | host\_event\_disconnect | This event is used to inform the Host of disconnection. |
| UMAC\_EVENT\_CONNECT\_RESULT | Unused | Unused |
| UMAC\_EVENT\_MIC\_FAIL | Unused | Unused |
| UMAC\_EVENT\_SCAN\_COMPLETE | host\_event\_scanres | This event is used to inform the Host, about response of CMD\_SCAN received before. |
| UMAC\_EVENT\_SCAN\_ABORT\_COMPLETE | cmd\_scan\_abort | This event is used to inform the Host, about response of CMD\_SCAN\_ABORT received before. After aborting the scan request received from host we send a success in scan abort event |
| UMAC\_EVENT\_MGMT\_FRAME | Unused | Unused |
| UMAC\_EVENT\_RESET\_COMPLETE | host\_event\_reset\_complete | This event is generated in response to CMD\_RESET from the host. |
| UMAC\_EVENT\_RSSI | Unused | Unused |
| UMAC\_EVENT\_STA\_INFO | Unused | Unused |
| UMAC\_EVENT\_REKEY\_DATA | Unused | Unused |
| UMAC\_EVENT\_MIB\_STAT | Unused | Unused |
| UMAC\_EVENT\_PHY\_STAT | host\_event\_phy\_stats | This event is used to inform phy statistics like number of packets received, CRC failures etc., |
| UMAC\_EVENT\_NW\_FOUND | unused | unused |
| UMAC\_EVENT\_NOA | umac\_event\_noa | This event is used to inform Host about the absence of AP due to AP mode power save in P2P. After receiving this event with AP absent true, station should refrain from transmitting frames. In this mode, a station can start TX only after NOA event with AP absent is set to false. |
| UMAC\_EVENT\_CTRL\_POOL\_ACK | Unused | Unused |
| UMAC\_EVENT\_COMMAND\_PROC\_DONE | There is no parameter in this event. | This event is generated after completing command from host. This event will not be generated for reset and TX command. |
| UMAC\_EVENT\_CH\_PROG\_DONE | There is no parameter in this event. | This event is used to inform Host about the channel programming completion for the UMAC\_CMD\_CHANNEL command issued before. During channel switch Host should not give any other commands. |
| UMAC\_EVENT\_PS\_ECON\_CFG\_DONE | umac\_event\_ps\_econ\_cfg\_complete | This event is send success if there are no frames destined to host when host is in doze mode or sends failure in this event. |
| UMAC\_EVENT\_PS\_ECON\_WAKE | umac\_event\_ps\_econ\_wake | If host receives any data packet or if any disconnection is observed then firmware sends a wake up event to host before transmission of packet or sending disconnection event. |
| UMAC\_EVENT\_MAC\_STATS | umac\_event\_mac\_stats | This event is used to inform mac statistics like cmd counts ,done counts. isr counts etc., |
| UMAC\_EVENT\_RF\_CALIB\_DATA | umac\_event\_rf\_calib\_data | This event is used to inform host about the calib parameters by reading them from phy. |
| UMAC\_EVENT\_RADAR\_DETECTED | host\_event\_radar\_detected | This event is used to inform host about Radar detection. This event is in response to the command UMAC\_CMD\_DETECT\_RADAR. Firmware is responsible for detecting the Radar and informs the channel number to host. This event is meaningful only when DFS (dot11h) is enabled. |
| UMAC\_EVENT\_MSRMNT\_COMPLETE | host\_event\_msrmnt\_complete | This event is to inform measurement results to the host. This event is in response to the command HOST\_EVENT\_MSRMNT\_COMPLETE. This is valid only when 802.11h (DFS) support is enabled. |
| UMAC\_EVENT\_ROC\_STATUS | umac\_event\_roc\_status | This event is used to inform host about the status of the ROC. |
| UMAC\_EVENT\_CHAN\_SWITCH | umac\_event\_ch\_switch\_t | This event is used to send the channel switch in time slice to the host.channel channel\_no refers to the switched channel. |

1. Data Structures
   1. Commands
      1. Cmd\_reset

|  |  |  |
| --- | --- | --- |
| Field Name | Types | Description |
| Hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of hdr structure is set to UMAC\_CMD\_RESET |
| Type | RESET\_TYPE\_T | 0 = Firmware initialization  1 = Firmware shutdown |
| ed\_sensitivity | int32 | Sensitivity threshold/Nose floor of the board. |
| auto\_sensitivity | uint32 | This is used to control sensitivity adjusting Algorithm in RPU. |
| rf\_params[369] | uint8 | RF calibration data |
| include\_rxmac\_hdr | uint32 | 1 = Host receives RX data packets along with IEEE80211 header. This is required in case of windows native driver.  0 = Host receives RX data packets without IEEE80211 header. |
| bg\_scan | bgscan\_params | Background scan parameters, for more details click on the bgscan\_params link. |
| num\_spatial\_streams | uint8 | Used to define number of spatial streams defined. |
| system\_rev | uint32 | unused |
| lmac\_mode | uint32 | LMAC\_MODE\_NORMAL 0  LMAC\_MODE\_FTM 1 |
| antenna\_sel | uint32 | Used to define the antenna selection information. |

* + 1. Cmd\_scan

|  |  |  |
| --- | --- | --- |
| Field Name | Types | Description |
| Hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_SCAN |
| If\_index | uint32 | This should be equal to the scan command’s virtual interface number.  0 <= if index <= MAX\_VIF\_NUM |
| type | SCAN\_TYPE\_T | 0 = Passive  1 = Active |
| n\_channel | uint32 | Number of channels to scan. n\_channel and channel\_list parameters decides the number of channels and channel numbers to scan. |
| n\_ssids | uint8 | Number of SSIDs to scan; as of now the firmware supports single ssid scan. This should be set to one for wildcard (no SSID) too. |
| channel\_list[50] | uint8 | List of channel numbers to scan. List can include from channel number can be either 2.4GHz or 5GHz. |
| chan\_max\_power[50] | uint8 | The maximum power limitation that can be used to send a packet in current channel. |
| chan\_flags [50] | uint8 | This defines scan type that can be used in current channel. |
| ssids[MAX\_NUM\_SSID] | ssid\_t | SSID list to scan; as of now only one SSID at a time so MAX\_NUM\_SSID is always 1. |
| p2p\_probe | uint32 | =0 for normal scan command  =1 In case scan command is issued as part of P2P search operation. |
| extra\_ies\_len | uint32 | If there are any extra information elements that needs to be part of probe request frames, **extra\_ies\_len** should be set to the length of those IEs. |
| extra\_ies[0] | uint8 | As Extra IEs are varying in size they are placed at the end of scan command payload. **extra\_ies\_len** should be set to total length of IEs contained in this field. Allocated memory should be of size  sizeof (Cmd\_scan\_t) + extra\_ies\_len. |

* + 1. Cmd\_scan\_abort

|  |  |  |
| --- | --- | --- |
| Field Name | Types | Description |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_SETKEY. |
| if\_index | uint32 | This is the virtual interface number.  0 <= if index <= MAX\_VIF\_NUM |

* + 1. Cmd\_set\_key

|  |  |  |
| --- | --- | --- |
| Field Name | Types | Description |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_SETKEY. |
| if\_index | uint32 | Key’s virtual interface index number. |
| Ctrl | uint32 | 0 = Add key  1 = Delete key |
| key\_type | KEY\_TYPE\_T | 0=unicast  1=broadcast |
| cipher\_type | CIPHER\_TYPE\_T | Cipher type of the key |
| key\_id | uint8 | Key index, values (0-3) used for WEP; For other cases it can be any value starting from zero. |
| key\_len | int32 | Key length; its value depends on security mechanism chosen. |
| rsc\_len | int32 | Length of the rsc value |
| mac\_addr[ETH\_ALEN] | uint8 | Peer station address of respective key. In case of group key, it is set to broadcast address, i.e. all 6 bytes to 0xFF. |
| key[MAX\_KEY\_LEN] | uint8 | Encryption / Decryption key for Unicast/Multi cast packets depending on key type |
| rsc[ RX\_SEQ\_SIZE ] | uint8 | Receive sequence count (RSC). Used in detecting RX replay attacks. |

* + 1. Cmd\_tx\_ctrl

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to CMD\_TX. For detailed explanation of other TX related fields in this header click on host\_mac\_msg\_hdr. |
| if\_index | Uint8 | Virtual interface number this packet is tied to. |
| queue\_num | Uint8 | There are four queues named AC\_BK, AC\_BE, AC\_VI and AC\_VO. The queue number will be decided based on IP packet’s TOS field. |
| descriptor\_id | Uint32 | In firmware based on the descriptor id received we differentiate between internally generated frames and frames received from host |
| num\_frames\_per\_desc | Uint32 | We get the number of frames received in this descriptor id applies in case of ampdu. |
| pkt\_length[32] |  | Gives the packets lengths for each frame in the descriptor. |
| more\_frms | Uint8 | This is used in AP mode operation; while transmitting a packet to a station which is in the power save mode, if the AP has more data in destination’s station queue, this field is set to 1;  0 in all other cases. |
| force\_tx | Uint32 | If this field is set for any packet, need to be transmitting even though TX has been disabled. This field is only meaningful in DFS. |
| tx\_flags |  | unused |
| num\_rates | Unit8 | This field is used to specify the number of rates a TX packet can transmit. A TX can be transmitted at max with 4 possible rates. |
| rate\_protection\_type[4] | uint8 | 0 = USE\_PROTECTION\_NONE  1 = USE\_PROTECTION\_RTS  2 = USE\_PROTECTION\_CTS2SELF |
| rate\_preamble\_type[4] | uint8 | 0 =USE\_SHORT\_PREAMBLE  1 = DONT\_USE\_SHORT\_PREAMBLE |
| rate\_retries[4] |  | This field is number of times a packet should be transmitted at each possible rate. Ex {(1, 3),( 5.5, 2), (36,1) }, this example is a set of (rate, rate\_retries) couple with num\_rates as 3. This means, transmit a TX packet with 3 possible rates 1, 5.5 and 36.If the first rate is not successful after trying 3 times, try the second rate twice. If it is still failing it will try to transmit at rate 36 once. |
| rate[4] | uint8 | rate value, if the most significant bit is one it’s 11n rate. |
| rate\_flags[4] |  | 0x01 = ENABLE\_GREEN\_FIELD  0x2 = ENABLE\_CHNL\_WIDTH\_40MHZ  0x04 = ENABLE\_SGI  0x08 = ENABLE\_11N\_FORMAT  0x10 = ENABLE\_VHT\_FORMAT  0x20= ENABLE\_CHNL\_WIDTH\_80MHZ |
| num\_spatial\_streams[4] | uint8 | number of spatial streams |
| stbc\_enabled | uint8 | enable/disable STBC |
| bcc\_or\_ldpc | uint8 | frame encoding |
| aggregate\_mpdu | uint8 | In 11n or 11ac, this field informs firmware whether to club the packet with other packets or not. For management, control and broadcast frames this field is not set. |
| encrypt | uint8 | If the frame is encrypted frame this is set to 1 else this is 0 |
| pkt\_gram\_payload\_len | uint32 | This is variable in length. Due to in line DMA and encryption, it’s not possible to transmit MAC header and unencrypted part of pay load as part of TX payload. This value  =24 bytes for plain data  =26 bytes for QoS data  =30 bytes for A4 address frames  =28 bytes for WEP 40  =30 bytes for QoS WEP frames etc., |
| gram\_payload[0] | uint8 | pkt\_gram\_payload\_len bytes should be copied into gram\_payload from TX packet. |

* + 1. Cmd\_frag

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_FRAG. |
| If\_index | uint32 | Virtual interface number of this frag command. |
| frag\_threshold | uint16 | Fragmentation threshold value ranging from 256 to 2347; It must be an even value. |

* + 1. Cmd\_rate

|  |  |  |
| --- | --- | --- |
| Field Name | Types | Description |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_RATE. |
| is\_mcs | uint32 | Informs whether this rate is 11n rate or 11G rate. |
| rate | uint32 | = MCS index if is\_mcs field is true  = rate if is\_mcs field is false. |

* + 1. Cmd\_disconnect

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_DISCONNECT. |
| If\_index | uint32 | Virtual interface number of disconnect command; This will not remove the vif, instead it will disconnect from the Access Point. |
| reason\_code | int32 | Reason for disconnecting. |

* + 1. Cmd\_ps

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_PS. |
| If\_index | uint32 | Virtual interface number of power save command. |
| power\_save | [POWER\_SAVE\_T](file:///C:\img\11n-work\HOST-Specification.doc#_POWER_SAVE_T) | 0 = DISABLE  1=ENABLE |

* + 1. Cmd\_vifctrl

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_VIF\_CTRL. |
| if\_ctrl | uint32 | 0=Add interface  1=Delete Interface |
| if\_index | uint32 | = 0 <= if\_index <= MAX\_VIF\_NUM.  Host has to remember if\_index value for future communication of commands / events to / from firmware. |
| mode | uint32 | 0 = IF\_MODE\_STA\_BSS  1 = IF\_MODE\_STA\_IBSS  2 = IF\_MODE\_AP  3 = IF\_MODE\_INVALID |
| mac\_addr[ETH\_ALEN] | uint32 | Virtual interface’s mac address; It should be noted that each vif will have a different mac address. |

* + 1. Cmd\_set\_beacon

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_SET\_BEACON |
| if\_index | uint32 | Virtual interface number. |
| interval | uint32 | This is beacon interval value; normally this value is set to 100 ms. |
| dtim\_period | uint32 | = 0 means AP will indicate both unicast and multicast traffic on every beacon.  =1 means AP will indicate unicast traffic on every beacon and multicast traffic on every alternative beacon. |
| len | uint32 | Length of this beacon frame |
| mac\_addr[6] | uint8 | Unused |
| channel | uint32 | Unused |
| beacon\_buf[0] | uint8 | Beacon frame payload starting from fc field will be copied here. Size of this command should be equal to  Sizeof (cmd\_set\_beacon\_t) + length of the frame. |

* + 1. Cmd\_set\_mode

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_SET\_BEACON |
| If\_index | uint32 | Virtual interface number. |
| type | uint32 | 0 = IF\_MODE\_STA\_BSS  1 = IF\_MODE\_STA\_IBSS  2 = IF\_MODE\_AP  3 = IF\_MODE\_INVALID |

* + 1. Cmd\_ht\_ba

|  |  |  |
| --- | --- | --- |
| Field Name | Types | Description |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_BA\_SESSION\_INFO |
| If\_index | uint32 | virtual interface index number |
| op | uint32 | 0 = BLOCK\_ACK\_SESSION\_STOP  1 = BLOCK\_ACK\_SESSION\_START |
| tid | uint32 | Block ack session TID |
| ssn | uint32 | Start sequence number |
| policy | uint32 | Block ack policy. 0 – Immediate block ack   1. Delayed block ack |
| vif\_addr[6] | uint32 | MAC address of virtual interface |
| peer\_addr[6] | uint32 | MAC address of peer station that this session is established with. |

* + 1. Cmd\_vht\_beamform

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_BA\_SESSION\_INFO |
| vht\_beamform\_status | uint32 | Status is used to inform firmware if to enable or disable beam forming |
| vht\_beamform\_period | uint32 | Informs the period for configuring the timer in firmware .we get it in milliseconds. |

* + 1. Cmd\_roc

|  |  |  |
| --- | --- | --- |
| Field Name | Types | Description |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_BA\_SESSION\_INFO |
| roc\_ctrl | uint32 | ROC\_STOP 0  ROC\_START 1  This flag indicates if to start or stop ROC. |
| roc\_channel | uint32 | This indicates which channel to be in ROC. |
| roc\_duration | uint32 | Used to configure the time using hardware timer in firmware. |
| roc\_type | uint32 | ROC\_TYPE\_NORMAL 0  ROC\_TYPE\_OFFCHANNEL\_TX 1 |

* + 1. Cmd\_nw\_selection

|  |  |  |
| --- | --- | --- |
| Field Name | Types | Description |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_BA\_SESSION\_INFO |
| if\_index | uint32 | Virtual interface number |
| p2p\_selection | uint32 | unused |
| ssid | ssid\_t | Service Set Identifier |
| scan\_req\_ie\_len | uint32 | Length of the scan request IEs |
| scan\_resp\_ie\_len | uint32 | Length of the scan response IEs |
| scan\_req\_ie[200] | uint8 | Scan request Information elements |
| scan\_resp\_ie[200] | uint8 | Scan response Information elements |

* + 1. Cmd\_set\_defaultkey

|  |  |  |
| --- | --- | --- |
| Field Name | Types | Description |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_BA\_SESSION\_INFO |
| if\_index | uint32 | Virtual interface number |
| key\_id | uint32 | Default key index to use |

* + 1. Cmd\_set\_rekey

|  |  |  |
| --- | --- | --- |
| Field Name | Types | Description |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_BA\_SESSION\_INFO |
| if\_index | uint32 | Virtual interface number |
| Kek[16] | uint8 | Key encryption key |
| Kck[16] | uint8 | Key confirmation key |
| replay\_ctr[8] | uint8 | replay counter |

* + 1. Cmd\_tx\_pwr

|  |  |  |
| --- | --- | --- |
| Field Name | Types | Description |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_BA\_SESSION\_INFO |
| if\_index | uint32 | Virtual interface number |
| tx\_pwr | uint32 | power level in dBm (NA) |

* + 1. Cmd\_mcst\_addr\_cfg

|  |  |  |
| --- | --- | --- |
| Field Name | Types | Description |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_BA\_SESSION\_INFO |
| op | uint32 | 0 -- ADD multicast address  1 -- Remove multicast address |
| mac\_addr[6] | uint32 | addr to add or delete.. |

* + 1. Cmd\_mcst\_filter\_ctrl

|  |  |  |
| --- | --- | --- |
| Field Name | Types | Description |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_BA\_SESSION\_INFO |
| ctrl | uint32 | 0 - disable multicast filtering in LMAC  1 - enable multicast filtering in LMAC |

* + 1. Cmd\_channel

|  |  |  |
| --- | --- | --- |
| Field Name | Types | Description |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_BA\_SESSION\_INFO |
| channel\_bw | uint32 | 0 = 20MHz  1 = 40MHz  2 = 80MHz  3 = 160MHz |
| primary\_ch\_number | uint32 | Primary channel centre frequency. |
| channel\_number1 | uint32 | First band centre frequency For non-contiguous bands |
| channel\_number2 | uint32 | centre frequency of secondary band |
| freq\_band | uint32 | 0 = 2.4GHz  1 = 5GHz |
| vif\_index |  |  |

* + 1. Cmd\_vif\_cfg

|  |  |  |
| --- | --- | --- |
| Field Name | Types | Description |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_VIF\_CFG |
| changed\_bitmap | uint32 | 1 = BASICRATES\_CHANGED,  2 = SHORTSLOT\_CHANGED  4 = POWERSAVE\_CHANGED  8 = UAPSDTYPE\_CHANGED  16 = ATIMWINDOW\_CHANGED  32 = AID\_CHANGED  64 = CAPABILITY\_CHANGED  128 = SHORTRETRY\_CHANGED  256 = LONGRETRY\_CHANGED  512 = BSSID\_CHANGED  1024 = RCV\_BCN\_MODE\_CHANGED  2048=BCN\_INT\_CHANGED  4096=DTIM\_PERIOD\_CHANGED  8192=SMPS\_CHANGED  16384=CONNECT\_STATE\_CHANGED  32768=OP\_CHAN\_CHANGED |
| basic\_rate\_set | uint32 | bitmap of supported basic rates. |
| use\_short\_slot | uint32 | 0 – long slot  1 – short slot |
| atim\_window | uint32 | TBD |
| aid | uint32 | Association ID, useful in recognizing the buffered packets at AP when station is in power save. |
| capability | uint32 | When capabilities of current network changed, the latest capability information is programmed to firmware. |
| short\_retry | uint32 | Short retry count. This is used only for the packets initiated by LMAC. |
| long\_retry | uint32 | Long retry count. This is used only for the packets initiated by LMAC. |
| bcn\_mode | uint32 | 0 = Receive all beacons  1 = Receive current network beacons only  2 = Receive no beacons |
| dtim\_period | uint8 | This value is in terms of multiple of beacon intervals. A station in power save mode should wake up from sleep and listen to DTIM beacons. |
| beacon\_interval | uint32 | in units of milliseconds, this field informs beacon sending time. Useful in AP /IBSS mode |
| if\_index | uint32 | Virtual interface number |
| vif\_addr[6] | uint8 | Mac address of virtual interface. |
| bssid[6] | uint8 | Current network’s bssid. |
| smps\_info |  | 0= SMPS\_ENABLED BIT  1= SMPS\_MODE BIT  Indicates SMPS information. |
| connect\_state |  | States the connection status of the virtual interface |
| op\_channel |  | Indicates the operation channel of the virtual interface. |

* + 1. Cmd\_sta

|  |  |  |
| --- | --- | --- |
| Field Name | Types | Description |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_STA |
| if\_index | uint32 | Virtual interface number. |
| op | uint32 | 0 = ADD  1 = REM |
| supp\_rates[STA\_NUM\_BANDS] | uint32 | Supported rate set of the STA command. |
| ht\_cap | uint32 | high throughput (11n)capability information |
| ht\_supported | uint32 | Whether a station supports High throughput (11n) data rates. |
| vht\_cap | uint32 | Very high throughput (11ac) capability. |
| vht\_supported | uint32 | Whether the adding station supports very high throughput. |
| ampdu\_factor | uint32 | Unused |
| ampdu\_density | uint32 | Unused |
| rx\_highest | uint32 | Unused |
| tx\_params | uint32 | Unused |
| rx\_mask[HT\_MCS\_MASK\_LEN] | uint8 | Unused |
| addr[ETH\_ALEN] | uint8 | MAC address of the station |
| dot11\_mode | uint8 | Unused |
| no\_of\_streams | uint8 | Unused |
| preamble | uint8 | Unused |
| stbc\_enable | uint8 | Unused |
| ldpc\_enable | uint8 | Unused |
| guard\_interval | uint8 | Unused |
| aggregation | uint8 | Unused |
| tid | uint8 | Unused |
| band\_width | unit8 | Unused |

* + 1. Cmd\_txq\_params

|  |  |  |
| --- | --- | --- |
| Field Name | Types | Description |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_STA |
| queue\_num | uint32 | AC\_BK = 0  AC\_BE = 1  AC\_VI = 2  AC\_VO = 3  AC\_BCN = 4 |
| aifsn | uint32 | =7 for AC\_BK  =3 for AC\_BE  =2 for AC\_VI  =2 FOR AC\_VO |
| txop | uint32 | Transmission opportunity for this queue in units of 32 micro seconds.  = 0 for AC\_BK  =0 for AC\_BE  =188 for AC\_VI  = 102 for AC\_VO |
| cwmin | uint32 | Contention window minimum value. Back off value should be a random number from 2 power cwmin to 2 power cwmax.  =31 for AC\_BK  =31 for AC\_BE  = 15 for AC\_VI  = 7 for AC\_VO |
| cwmax | uint32 | = 1023 for AC\_BK  =1023 for AC\_BE  = 31 for AC\_VI  = 15 for AC\_VO |
| uapsd |  |  |
| if\_index | uint32 | Virtual interface number |
| vif\_addr[6] | uint8 | mac address of virtual interface. |

* + 1. Cmd\_aux\_adc\_chain\_sel

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Description |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_DETECT\_RADAR |
| Chain\_id | uint32 | AUX\_ADC\_CHAIN1=1  AUX\_ADC\_CHAIN=2 |

* + 1. Cmd\_detect\_radar

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Description |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_DETECT\_RADAR |
| radar\_detect\_op | uint32 | 0 = RADAR\_DETECT\_OPERATION\_START  1 = RADAR\_DETECT\_OPERATION\_STOP |

* + 1. Umac\_cmd\_tx\_enable

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Description |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_ENABLE\_TX |

* + 1. Cmd\_discard\_pkts

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Description |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_ENABLE\_TX |

* + 1. Cmd\_msrmnt\_start

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Description |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_ENABLE\_TX |
| start\_time[8] | uint8 | when to start the measurement |
| msr\_dur | unsigned short | How long measurement to be carried out |

* + 1. Cmd\_bt\_info

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Description |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_DETECT\_RADAR |
| bt\_state | uint32 | BT\_STATE\_OFF 0  BT\_STATE\_ON 1 |

* + 1. Cmd\_chanctx\_time\_config

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Type | | Description | |
| hdr | host\_mac\_msg\_hdr | | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_DETECT\_RADAR | |
| Struct chanctx\_time\_info[2] | | struct chanctx\_time\_info | | Uint32 channel  Uint32 percentage |

* + 1. Cmd\_cont\_tx

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Type | | Description | |
| hdr | host\_mac\_msg\_hdr | | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_CMD\_DETECT\_RADAR | |
| op | | uint32 | | used in production mode. |

* 1. Events
     1. wlan\_rx\_pkt

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_EVENT\_RX. |
| pkt\_length | uint32 | Length of the packet; this should be header length + payload bytes. |
| rate\_or\_mcs | uint8 | Most significant bit set to 1 implies rate is MCS rate otherwise legacy rate. |
| rssi | uint8 | rssi strength for this packet. |
| rx\_pkt\_status | uint8 | 0 = Successful  1 = MIC Error |
| rate\_flags | uint8 | 0x01 = ENABLE\_GREEN\_FIELD  0x02 =ENABLE\_CHNL\_WIDTH\_40MHZ 0x04 = ENABLE\_SGI  0x08 = ENABLE\_11N\_FORMAT  0x10 = ENABLE\_VHT\_FORMAT  0x20 = ENABLE\_CHNL\_WIDTH\_80MHZ |
| nss | uint8 | Number spatial streasm |
| num\_sts | uint8 | Number of space time streams |
| timestamp[8] | uint8 | packet received time |
| stbc\_enabled | uint8 | 1 – if the Received packet is STBC enabled |
| ldpc\_enabled | uint8 | 1 – if the Received packet is LDPC enabled |
| link\_margin | uint8 | unused |
| channel | uint8 | channel number |
| reserved[2] | uint8 | reserved bytes for padding |
| payload[0] | uint8 | Payload starting from frame control. Unlike TX command, header is combined along with payload |

* + 1. umac\_event\_tx\_done

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to EVENT\_TX\_DONE |
| queue | uint8 | 0 = AC\_BK  1 = AC\_BE  2=AC\_VI  3=AC\_VO |
| pdout\_voltage | uint32 | Unused |
| frm\_status | uint32 | 0 = TX\_DONE\_SUCCESS  1 = TX\_DONE\_ERROR\_RETRY\_LIMIT  2 =TX\_DONE\_MSDU\_LIFETIME  3 =TX\_DONE\_KEY\_NOT\_FOUND |
| retries\_num | uint32 | Number of times the TX packet retried. |
| rate[32] | uint8 | This is the rate that TX packet is successfully transmitted.  rate = Units of 500 Kbps or mcs index = 0 to 7 |
| descriptor\_id | uint32 | packet id of the transmitted frame |
| reserved[12] | uint8 | reserved bytes for padding |

* + 1. umac\_event\_ch\_prog\_complete

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_EVENT\_DISCONNECTED |

* + 1. umac\_event\_ch\_switch

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_EVENT\_DISCONNECTED |
| chan | uint32 | channel number |

* + 1. host\_event\_mgmt\_rx

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_EVENT\_DISCONNECTED |
| if\_index | uint32 | vif index |
| rssi | int32 | signal strength |
| rate | uint32 | data rate |
| padding | uint32 | padding |

* + 1. host\_event\_command\_complete

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_EVENT\_DISCONNECTED |

* + 1. host\_event\_connect\_result

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_EVENT\_DISCONNECTED |
| if\_index | uint32 | vif index |
| result\_code |  | connection result code  CONNECT\_SUCCESS = 0,  CONNECT\_UNSPECIFIED\_FAILURE=1,  CONNECT\_AUTH\_FAILURE=2,  CONNECT\_AUTH\_TIMEOUT=3,  CONNECT\_ASSOC\_TIMEOUT=4,  CONNECT\_ASSOC\_FAILURE=5,  CONNECT\_START\_IBSS=6 |
| aid | uint32 | association id |
| cap\_info | uint32 | capability info |
| ht\_supported | int32 | HT capable ? |
| ht\_cap\_info | uint16 | HT capabilities |
| vht\_supported | int32 | VHT capable ? |
| vht\_cap\_info | uint16 | VHT capabilities |
| qos\_capability | uint32 | QOS enabled ? |
| wmm\_acm | uint32 | WMM Access category |
| channel | uint32 | channel number |
| bssid[ETH\_ALEN] | uint8 | BSSID |
| ie\_len | uint32 | Length of IE data |
| ie[MAX\_IE\_LEN] | uint8 | Information Elements |
| bss\_frame | struct bssres | Response frame from BSS (NA) |

* + 1. host\_event\_rssi

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_EVENT\_DISCONNECTED |
| if\_index | int32 | vif index |
| rssi | int32 | Signal strength |

* + 1. host\_event\_rekey\_data

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_EVENT\_DISCONNECTED |
| if\_index | uint32 | vif index |
| cipher | int32 | Signal strength |
| key\_idx | uint32 | Key cache index |
| key\_len | uint32 | Key length |
| rsc\_len | uint32 | Replay sequence counter len |
| rsc[8] | uint8 | Replay sequence counter |
| key[16] | uint8 | Key value |

* + 1. umac\_event\_ch\_switch\_complete

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_EVENT\_DISCONNECTED |
| status | int32 | Channel switch status |

* + 1. host\_event\_disconnect

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_EVENT\_DISCONNECTED |
| if\_index | uint32 | virtual interface number |
| reason\_code | uint32 | reason for disconnection |
| mac\_addr[ETH\_ALEN] | uint8 | MAC address of current virtual interface. |

* + 1. host\_event\_scanres

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_EVENT\_SCAN\_COMPLETE |
| if\_index | uint32 | virtual interface number |
| scanres\_len | uint32 | This is total length of number of beacons or probe responses that are included in this event. Each frame is with it’s rssi value and channel number. |
| status\_code | uint32 | 0 = Scanning operation successful  1 = Unable to do scan |
| no\_of\_bss | uint32 | Total number of bss (Beacon/Probe response frames) along with their channel number and rssi. |
| more\_results | uint32 | = 1 scan is note completed, and firmware is continuing scan.  =0 Scan operation completed |
| bss\_res[0] | uint8 | Actual bss (beacon / probe response) frames starting for their rssi and channel values. |

* + 1. host\_event\_reset\_complete

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_EVENT\_RESET\_COMPLETE |
| cap | uint32 | Unused |
| ht\_supported | uint32 | Unused |
| ampdu\_factor | uint32 | Unused |
| ampdu\_density | uint32 | Unused |
| rx\_mask[HT\_MCS\_MASK\_LEN] | uint32 | Unused |
| rx\_highest | uint32 | Unused |
| tx\_params | uint32 | Unused |
| version[6] | uint8 | Firmware version number |

* + 1. umac\_event\_mib\_stats

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_EVENT\_MIB\_STAT |
| ed\_cnt | uint32 | Number of energy detections. |
| mpdu\_cnt | uint32 | Number of mpdus received |
| ofdm\_crc32\_pass\_cnt | uint32 | Successfully received OFDM packets |
| ofdm\_crc32\_fail\_cnt | uint32 | Number OFDM CRC32 fail packets |
| dsss\_crc32\_pass\_cnt | uint32 | Successfully received DSSS packets |
| dsss\_crc32\_fail\_cnt | uint32 | Number DSSS CRC32 fail packets |
| mac\_id\_pass\_cnt | uint32 | Successfully received correct ma id packets |
| mac\_id\_fail\_cnt | uint32 | Number mac id fail packets |
| ofdm\_corr\_pass\_cnt | uint32 | Successfully received OFDM correlation packets |
| ofdm\_corr\_fail\_cnt | uint32 | Number OFDM correlation fail packets |
| dsss\_corr\_pass\_cnt | uint32 | Successfully received DSS correlation packets |
| dsss\_corr\_fail\_cnt | uint32 | Number DSS correclation fail packets |
| ofdm\_s2l\_fail\_cnt | uint32 | OFDM Short to Long preamble fail count |
| lsig\_fail\_cnt | uint32 | Lsig failure count |
| htsig\_fail\_cnt | uint32 | Htsig failure count |
| vhtsiga\_fail\_cnt | uint32 | Vht siga failure count |
| vhtsigb\_fail\_cnt | uint32 | Vht sigb failure count |
| nonht\_ofdm\_cnt | uint32 | Non ht ofdm count |
| nonht\_dsss\_cnt | uint32 | Non ht dsss count |
| mm\_cnt | uint32 | Mm count |
| gf\_cnt | uint32 | Gf count |
| vht\_cnt | uint32 | VHTcount |
| aggregation\_cnt | uint32 | Aggregation count |
| non\_aggregation\_cnt | uint32 | Non aggregation count |
| ndp\_cnt | uint32 | ndp count |
| ofdm\_ldpc\_cnt | uint32 | Ofdm ldpc count |
| ofdm\_bcc\_cnt | uint32 | Ofdm bcc count |
| midpacket\_cnt | uint32 | Midpacket count |
| dsss\_sfd\_fail\_cnt | uint32 | Dsss sfd fail count |
| dsss\_hdr\_fail\_cnt | uint32 | Dsss hdr fail count |
| dsss\_short\_preamble\_cnt | uint32 | Dsss short preamble count |
| dsss\_long\_preamble\_cnt | uint32 | Dsss long preamble count |
| sifs\_event\_cnt | uint32 | Sifs event count |
| cts\_cnt | uint32 | Cts count |
| ack\_cnt | uint32 | Ack count |
| sifs\_no\_resp\_cnt | uint32 | Sifs no resp count |
| unsupported\_cnt | uint32 | Unsupported count |
| l1\_corr\_fail\_cnt | uint32 | 11 corre failure count |
| phy\_stats\_reserved22 | uint32 | Unused |
| phy\_stats\_reserved23 | uint32 | Unused |
| phy\_stats\_reserved24 | uint32 | Unused |
| phy\_stats\_reserved25 | uint32 | Unused |
| phy\_stats\_reserved26 | uint32 | Unused |
| phy\_stats\_reserved27 | uint32 | Unused |
| phy\_stats\_reserved28 | uint32 | Unused |
| phy\_stats\_reserved29 | uint32 | Unused |
| phy\_stats\_reserved30 | uint32 | Unused |
| tx\_pkts\_from\_lmac | uint32 | Tx packtes received from lmac count |
| tx\_pkts\_tx2tx | uint32 | Tx2tx packet count |
| tx\_pkts\_from\_rx | uint32 | Tx packets from rx count |
| tx\_pkts\_ofdm | uint32 | Tx packets from ofdm count |
| tx\_pkts\_dsss | uint32 | Tx packets from dss count |
| tx\_pkts\_reached\_end\_of\_fsm | uint32 | Tx packets that reached end of fsm count |
| tx\_unsupported\_modulation | uint32 | Tx unsupported modulation count |
| tx\_latest\_pkt\_from\_lmac\_or\_sifs | uint32 | Tx packets received from sifs count |
| tx\_abort\_bt\_confirm\_cnt | uint32 | Tx abort bt confirmation count |
| tx\_abort\_txstart\_timeout\_cnt | uint32 | Tx abort tx start timeout count |
| tx\_abort\_mid\_bt\_cnt | uint32 | Tx abort mid bt count |
| tx\_abort\_dac\_underrun\_cnt | uint32 | Tx abort dac underrun count |
| tx\_ofdm\_symbols\_master | uint32 | Tx ofdm synbols master |
| tx\_ofdm\_symbols\_slave1 | uint32 | Tx ofdm symbols slave1 |
| tx\_ofdm\_symbols\_slave2 | uint32 | Tx ofdm symbols slave 1 |
| tx\_dsss\_symbols | uint32 | Tx dsss symbols |
| cts\_received\_mcp\_cnt | uint32 | Number of cts frames received |

* + 1. host\_event\_phy\_stats

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_EVENT\_PHY\_STAT |
| phy\_stats[64] | uint32 | Unused |

* + 1. umac\_event\_noa

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. id parameter of the hdr structure is set to UMAC\_EVENT\_NOA |
| if\_index | uint32 | Virtual interface index number |
| vif\_addr[ETH\_ALEN] | uint8 | mac address of VIF |
| noa\_active | uint32 | 0 = NOA feature is not active  1 = NOA feature is active |
| ap\_present | uint32 | 0 = AP is absent  1 = AP is present |

* + 1. umac\_event\_ps\_econ\_cfg\_complete

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. Id parameter of the hdr structure is set to UMAC\_EVENT\_RADAR\_DETECTED. This command is only valid in |
| status | uint8 | -1= donot enter powersave as the host has some rx packets .  0=indictes that successfully configured economy powersave . |

* + 1. umac\_event\_ps\_econ\_wake

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. Id parameter of the hdr structure is set to UMAC\_EVENT\_RADAR\_DETECTED. This command is only valid in |
| trigger | uint32 | Trigger to wakeup from sleep .  TRIG\_PKT\_RCV  TRIG\_DISCONNECT |

* + 1. umac\_event\_mac\_stats

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. Id parameter of the hdr structure is set to UMAC\_EVENT\_RADAR\_DETECTED. This command is only valid in |
| roc\_start | uint32 | Num of ROC start commands |
| roc\_stop | uint32 | Num of ROC stop commands |
| roc\_complete | uint32 | Num of ROC start command completes |
| roc\_stop\_complete | uint32 | Num of ROC stop command completes |
| tx\_cmd\_cnt | uint32 | Num of TX commands received from host |
| tx\_done\_cnt | uint32 | Num of Tx done events sent to host |
| tx\_edca\_trigger\_cnt | uint32 | Num of times EDCA engine was triggered |
| tx\_edca\_isr\_cnt | uint32 | Num of times EDCA ISR was generated |
| tx\_start\_cnt | uint32 | Num of TX starts to MCP |
| tx\_abort\_cnt | uint32 | Num of TX aborts detected |
| tx\_abort\_isr\_cnt | uint32 | Num of TX aborts received from MCP |
| tx\_underrun\_cnt | uint32 | Num of under-runs |
| tx\_rts\_cnt | uint32 | Num of RTS frames Tx’d |
| tx\_ampdu\_cnt | uint32 | Num of AMPDU’s tx’d –incremented by 1 for each A-MPDU (consisting of one or more MPDUs) |
| tx\_mpdu\_cnt | uint32 | Num of MPDU’s tx’d – incremented by 1 for MPDU (1 for each A-MPDU subframe) |
| rx\_isr\_cnt | uint32 | Num of RX ISRs |
| rx\_ack\_cts\_to\_cnt | uint32 | Num of timeouts ACK |
| rx\_cts\_cnt | uint32 | Num of CTS frames received |
| rx\_ack\_resp\_cnt | uint32 | Num of Ack frames received |
| rx\_ba\_resp\_cnt | uint32 | Num of BA frames received |
| rx\_fail\_in\_ba\_bitmap\_cnt | uint32 | Num of BA frames indicating at least one failure in the BA bitmap |
| rx\_circular\_buffer\_free\_cnt | uint32 | Num of entries returned to RX circular buffers |
| rx\_mic\_fail\_cnt | uint32 | Num of MIC failures |
| hal\_cmd\_cnt | uint32 | Num of commands received by HAL from the host |
| hal\_event\_cnt | uint32 | Num of events sent by HAL to the host |
| hal\_ext\_ptr\_null\_cnt | uint32 | Num of packets dropped due to lack of Ext Ram buffers from host |

* + 1. umac\_event\_rf\_calib\_data

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. Id parameter of the hdr structure is set to UMAC\_EVENT\_RADAR\_DETECTED. This command is only valid in |
| rf\_calib\_data\_length | uint32 | Length of the calibration data |
| rf\_calib\_data | uint8 | calibration data |

* + 1. host\_event\_radar\_detected

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. Id parameter of the hdr structure is set to UMAC\_EVENT\_RADAR\_DETECTED. This command is only valid in |
| freq | uint32 | Central frequency of the detected Radar. |

* + 1. host\_event\_msrmnt\_complete

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. Id parameter of the hdr structure is set to UMAC\_EVENT\_MSRMNT\_COMPLETE |
| msrmnt\_status | uint32 | 0x01 = UMAC\_EVENT\_MSRMNT\_STATUS\_BSS  0x02 = UMAC\_EVENT\_MSRMNT\_STATUS\_OFDM\_PREAMBLE  0x04 = UMAC\_EVENT\_MSRMNT\_STATUS\_UNIDENTIFIED\_SIGNAL  0x8 = UMAC\_EVENT\_MSRMNT\_STATUS\_RADAR\_SIGNAL  0x10 = UMAC\_EVENT\_MSRMNT\_STATUS\_NO\_MSRMNT  0x20 = UMAC\_EVENT\_MSRMNT\_STATUS\_LATE  0x40 = UMAC\_EVENT\_MSRMNT\_STATUS\_INCAPABLE  0x80 = UMAC\_EVENT\_MSRMNT\_STATUS\_REFUSE |

* + 1. Umac\_event\_roc\_status

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Types** | **Description** |
| hdr | host\_mac\_msg\_hdr | 28 byte host-firmware header. Id parameter of the hdr structure is set to UMAC\_EVENT\_RADAR\_DETECTED. This command is only valid in |
| Roc\_status | uint32 | Defines the status of roc |

* 1. User Defined Types
     1. CIPHER\_TYPE\_T

|  |  |
| --- | --- |
| Expected Values | Description |
| 0 = SUIT\_WEP40 | 64 bit WEP encryption algorithm |
| 1 = SUIT\_WEP104 | 128 bit WEP encryption algorithm |
| 2 = SUIT\_TKIP | TKIP encryption algorithm is enhanced security over WEP. It is mandatory in WPA. |
| 3 = SUIT\_CCMP | (CCMP) encryption algorithm is IEEE802.11i. Protocol, it is mandatory in WPA2, RSNA but optional in WPA. |
| 4 = SUIT\_WAPI | WAPI uses 16 byte MIC value and it is Chinese encryption algorithm. |

* + 1. KEY\_TYPE\_T

|  |  |
| --- | --- |
| **Expected Values** | **Description** |
| 0=UNICAST | The security key generated for communicating with the peer. Used to encrypt/decrypt unicast packets. |
| 1=BROADCAST | The security key generated for communicating with all members in BSS. Used to encrypt /decrypt multicast packets. |

* + 1. RESET\_TYPE\_T

|  |  |
| --- | --- |
| **Expected Values** | **Description** |
| 0=Shutdown | Shutdown the device. |
| 1=Initialize | Initializes the device. |

* + 1. POWER\_SAVE\_T

|  |  |
| --- | --- |
| **Expected Values** | **Description** |
| 0=DISABLE | Disable 802.11 power save on target; Nothing to do with Host power save. |
| 1=ENABLE | Enable 802.11 power save on target, nothing to do with Host power save |

* + 1. SCAN\_TYPE\_T

|  |  |
| --- | --- |
| **Expected Values** | **Description** |
| 0= Active | Active scan probes the channel for BSS information. |
| 1=Passive | Passive scan listens to channel for BSS information. |

* + 1. host\_mac\_msg\_hdr

|  |  |  |
| --- | --- | --- |
| Field name | Type | Description |
| hal\_data | hal\_data | Hardware Abstraction Layer’s (HAL) private data. This is 8 bytes information; first 4 bytes differentiate between HAL command and UMAC command. next four bytes are valid for TX command and carry address of TX payload that has to be DMAed to the firmware |
| descriptor\_id | uint32 | least significant 2 bytes are TX pool id, most significant 2 bytes are queue number. pooid of 0xffff indicates no TX payload. |
| payload\_length | uint32 | TX payload length excluding mac header bytes. |
| id | uint32 | Host firmware Command/Event id |
| length | uint32 | Size of cmd\_tx\_ctrl. It doesn’t include any payload length |
| more\_cmd\_data | uint32 | =1 , if current command is split into two or more. This is used for commands whose size greater than MAX\_CMD\_SIZE.  =0 for normal commands |

* + 1. ssid\_t

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Description |
| len | uint8 | The actual size of ssid that is being used. |
| ssid[MAX\_SSID\_LEN] | uint8 | SSID character buffer; according to IEEE it is of size varying from 1- 32 . |

* + 1. bgscan\_params

|  |  |  |
| --- | --- | --- |
| Field name | Type | Description |
| enabled | uint32 | =0 Background scan enabled  =1 Background scan disabled |
| channel\_list[50] | uint8 | list of channels to scan |
| channel\_flags[50] | uint8 | channel flags for each channel |
| scan\_intval | uint32 | Back ground scan is done at regular intervals. This value is set to interval value. Units are in mille seconds |
| channel\_duration | uint32 | Time spends on each channel .In units of mille seconds |
| serv\_channel\_dur | uint32 | In connected state scanning, we need to share the time between operating channel and non-operating channels. After scanning each channel, firmware spends “serv\_channel\_dur” in operating channel. Units are in milli seconds. |
| num\_channels | uint32 | number of channels configured |

* + 1. bssres

|  |  |  |
| --- | --- | --- |
| Field name | Type | Description |
| channel | uint32 | =0 Background scan enabled  =1 Background scan disabled |
| rssi | int32 | list of channels to scan |
| frame\_len | uint32 | channel flags for each channel |
| frame\_buf | uint8 | Back ground scan is done at regular intervals. This value is set to interval value. Units are in mille seconds |

* + 1. CONSTANTS

MAX\_SSID\_LEN 32

ETH\_ALEN 6

MAX\_KEY\_LEN 16

RX\_SEQ\_SIZE 6

MAX\_VIF\_NUM 2

HAL\_PRIV\_DATA\_SIZE 8