AN14281

Channel State Information (CSI) on FreeRTOS

Rev. 4.0 — 7 January 2026 Application note

Document information

Information	Content
Keywords	Channel state information, CSI, ambient motion index, AMI, API, event, configuration files, log files
Abstract	The application note explains how to get the channel state information from a received OFDM packet and move the CSI to the host for post-processing.

1 About this document

Channel State Information (CSI) in Wi-Fi is an indication of how the wireless channel affects the signal between two devices. Details such as the signal strength and phase of each subcarrier are captured. CSI can be used to improve the connection quality and sensing environment.

CSI is the known channel properties of a communication link. CSI is used in Wi-Fi OFDM PHY layers (including 11a/g/n/ac/ax) to achieve reliable communication with high data rates. As such, CSI is generated every time the Wi-Fi device receives a packet seen over the air; the firmware and driver allow for filtering and selecting of CSI captures to send to the host.

This document explains how to get the CSI records from the Wi-Fi packets in STA mode. It covers:

- The format of CSI records
- CSI configuration
- How to receive and transmit packets to generate CSI.
- How to capture CSI logs.
- CSI-enabled sample applications
- How to generate an ambient motion index (AMI) – a measure of motion

Note: *This document assumes that you are familiar with ref.[4] or ref.[5] and that you have used the SDK release to bring up the radios on your device (ref.[3]).*

1.1 Supported products

Table 1. Supported Wi-Fi 6 products and features

Product	CSI	AMI
AW611	Yes	Yes
IW610	Yes	Yes
IW611	Yes	Yes
IW612	Yes	Yes
RW610	Yes	Yes
RW612	Yes	Yes

2 Modules and flow

The host and the wireless device exchange CSI data using:

- **NXP Wi-Fi firmware:** provides the raw CSI data to NXP driver in the form of CSI/Netlink events.
- **Bus driver:** driver that communicates with the Wi-Fi, Bluetooth, and 802.15.4 radios of the supported product. ***Note:*** *only IW610, IW612, and RW612 support 802.15.4 radio.*
- **Wi-Fi driver:** applies CSI configurations and executes CSI commands. Receives CSI header data through CSI/Netlink events. Transfers the headers to wifi_cli for processing.
- **Application (wifi_cli)**: issues start/stop CSI commands and passes the CSI configuration file to the Wi-Fi driver. The application also processes the CSI/Netlink events received by the Wi-Fi driver, displays CSI data, and saves CSI data on the host.

Upon receiving a packet, the wireless device generates and sends the CSI record to the host.

Figure 1 illustrates the data transfer between the Wi-Fi firmware and wifi_cli application.

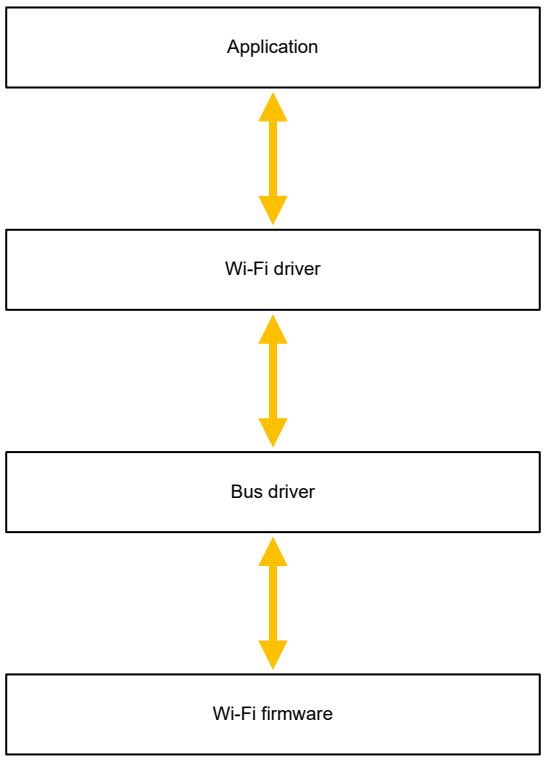

Figure 1. Data transfer between the Wi-Fi firmware and the application

3 CSI record

The CSI record includes the CSI header and CSI data.

3.1 CSI record format

Table 2 shows the format of CSI record.

Table 2. CSI record format

Dword	Byte			
	3	2	1	0
0	Signature[15:0]		Length[15:0]	
1	Header Signature ID			
2	PKT_info[31:0]			
3	TSF[63:32]			
4	TSF[31:0]			
5	Dst_MAC[31:0]			
6	Src_MAC[15:0]		Dst_MAC[47:32]	
7	Src_MAC[47:16]			
8	RX_NF_B	RX_NF_A	RX_RSSI_B	RX_RSSI_A
9	Chip ID	AP_TYPE	Channel	SINR
10	RSVD[7:0]	Total Gain	FCF[15:0]	
11	RSVD[15:0]		CSI Data Length[15:0]	
–	CSI Data			
–	Tail Signature ID			

3.2 Field descriptions

Table 3 described the fields of CSI record.

Table 3. Fields of CSI record

Field	Description
Length[15:0]	Actual buffer used (in Dword). Indicates the CSI record length which includes CSI header and CSI data.
Signature[15:0]	16-bit signature. Always 0xABCD (not configurable). 0xABCD = indicates the beginning of a CSI record
Header Signature ID	Header ID: 4 Byte user-defined value configurable using CSI configuration file
PKT_info[31:0]	See Table 4 and Table 5.
TSF[63:0]	Time stamp value from the timing synchronization function (TSF) of the supported product. TSF is the local clock of the Wi-Fi device, the units are microseconds.
Dst_MAC[47:0]	Destination MAC address
SrcMAC[47:0]	Source MAC address
RX_RSSI_A RX_RSSI_B	Received Signal Strength Indication (RSSI): total gain for the whole receiver chain plus average power of CSI data. Signed integer (-128 to +127) in dBm steps. For 1x1 products, RSSI_B is set to 0.
RX_NF_A RX_NF_B	RX Noise Floor (NF): total gain recorded in the receiver chain before the start of a packet. For 1x1 products, RX_NF_B is set to 0.
SINR	Signal to interference noise ratio (SINR). Signed integer (-128 to +127) in dB steps.
Channel	802.11 channel number
AP_TYPE	Type of access point 0x0 = legacy 0x2 = HT 0x3 = VHT 0x4 = HE Others = Reserved
Chip ID	Chip ID: 1 Byte user-defined value configurable using *CSI.conf* configuration file
FCF[15:0]	802.11 Frame Control Field (FCF) or carried FCF in control wrapper
Total Gain	Gain applied to the CSI data in dB.
CSI Data Length[15:0]	CSI data length, which is the actual length (in Dwords) + 1. • $L = \left\lceil \frac{nTones \times Nr \times Nc}{2} \right\rceil + 1$ Where: Nr = Number of rows in the CSI matrix, equal to the number of receiver antennas. Nc = Number of columns in the CSI matrix, equal to the number of spatial streams. nTones = Determined from the bandwidth and tone group (Ng). Refer to Table 6 for Wi-Fi 5 product category, and Table 7 for Wi-Fi 6 product category.
CSI Data	Refer to Section 3.3 for CSI data format.
Tail Signature ID	Tail ID : 4-Byte user-defined value configurable using *CSI.conf* configuration file

Table 4. PKT_INFO[31:0] signals for Wi-Fi 5 product category

Signal	Description
CSI format [1:0]	Adjust the CSI fixed-point format as: 00 = s8.3 01 = s8.4 10 = s8.5 11 = s8.6
Common AGC flag[0]	Applies to 2x2 products only. Indicates that common AGC is active. That is, the weaker RX path backs off gain to match the gain on the stronger path. 0 = common AGC not active 1 = common AGC active
RSVD[12:0]	Reserved
devBW[1:0]	The current bandwidth of the STA. 00 = 20 MHz 01 = 40 MHz 10 = 80 MHz 11 = Reserved
nRx [2:0] or Nr	Number of receiver antennas (constant unless antennas are deactivated using `mlanutl mlan0 antcfg`) 000 = 1 001 = 2
nTx[2:0] or Nc	Number of spatial streams (SS) in a given packet. 000 = 1 001 = 2 010 = 3 011 = 4
Ng[0]	Tone grouping (Table 6) 0 = 2 1 = 4
sigBW[1:0]	The bandwidth (BW) of a given received packet. The value of sigBW must be less than or equal to the value of devBW. 00 = 20 MHz 01 = 40 MHz 10 = 80 MHz 11 = Reserved
Primary Subband (PSB)[2:0]	If devBW[2:0] = 00 (20 MHz), not applicable If devBW[2:0] = 01 (40 MHz): Bit[0] = 0: PSB is -1 Bit[0] = 1: PSB is 1 If devBW[2:0] = 10 (80 MHz): Bit[1:0] = 00: PSB is -2 Bit[1:0] = 01: PSB is -1 Bit[1:0] = 10: PSB is 1 Bit[1:0] = 11: PSB is 2

Table 4. PKT_INFO[31:0] signals for Wi-Fi 5 product category*...continued*

Signal	Description
pktType[1:0]	Packet type 00 = legacy orthogonal frequency division multiplexing (OFDM) 01 = High Throughout (HT) 10 = Greenfield (GF)[1] 11 = Very High Throughout (VHT)

[1] Deprecated by WFA.

Table 5. PKT_INFO[31:0] signals for Wi-Fi 6 product category

Signal	Description
CSI format [1:0]	Adjust the CSI fixed-point format as: 00 = s8.3 01 = s8.4 10 = s8.5 11 = s8.6
Common AGC flag[0]	Applies to 2x2 products only. Common automatic gain control (AGC) Indicates that common AGC is active. That is, the weaker RX path backs off gain to match the gain on the stronger path. 0 = common AGC not active 1 = common AGC active
RSVD[8:0]	Reserved
HeLTF[1:0]	Long training field (LTF) duration 0 = 1x LTF 1 = 2x LTF 2 = 4x LTF
MU[0]	Number of users 0 = SU (single user) 1 = MU (multiple users)
devBW[1:0]	The current bandwidth of the STA. 00 = 20 MHz 01 = 40 MHz 10 = 80 MHz 11 = reserved
nRx [2:0] or Nr	Number of receiver antennas (constant unless antennas are deactivated using `mlanutl mlan0 antcfg`) 000 = 1 001 = 2
nTx[2:0] or Nc	Number of spatial streams (SS) in a received packet. For each CSI collection, the number can vary based on the SS used. 000 = 1 001 = 2 010 = 3 011 = 4
Reserved[0]	Reserved[0]

Table 5. PKT_INFO[31:0] signals for Wi-Fi 6 product category ***...continued***

Signal	Description
sigBW[1:0]	The bandwidth (BW) of a given received packet. The value of sigBW must be less than or equal to the value of devBW. 00 = 20 MHz 01 = 40 MHz 10 = 80 MHz 11 = Reserved
Primary Subband (PSB)[2:0]	If devBW[2:0] = 00 (20 MHz), not applicable If devBW[2:0] = 01 (40 MHz): Bit[0] = 0: PSB is -1 Bit[0] = 1: PSB is 1 If devBW[2:0] = 10 (80 MHz): Bit[1:0] = 00: PSB is -2 Bit[1:0] = 01: PSB is -1 Bit[1:0] = 10: PSB is 1 Bit[1:0] = 11: PSB is 2
pktType[2:0]	Packet type 000 = legacy Orthogonal Frequency Division Multiplexing (OFDM) 001 = High Throughout (HT) 010 = Greenfield (GF)[1] 011 = Very High Throughout (VHT) 100 = High Efficiency (HE)

[1] Deprecated by WFA.

Table 6. nTones parameter values for Wi-Fi 5 product category

nTones are based on subcarrier indices for compressed beamforming feedback. Refer to VHT and HT subcarrier and number of matrices/carrier grouping tables in ref.[2].

Bandwidth	Tone grouping (Ng)	nTones
20 MHz	2	30
40 MHz	2	58
80 MHz[1]	4	62

[1] For Wi-Fi 5 product category: Maximum CSI data length for 80 MHz 2x4 (two RX antennas receiving data from a 4-antenna device) = 249 Dwords.

Table 7. nTones parameter values for Wi-Fi 6 product category

nTones are based on subcarrier indices for compressed beamforming feedback. Refer to VHT, HT, and HE subcarrier and number of matrices/carrier grouping tables in ref.[1] and ref.[2].

Bandwidth	Tone grouping (Ng)	nTones
20 MHz	1 (VHT[1] + legacy)	52
	2 (HT[2])	30
	4 (HE[3])	64
40 MHz	1 (VHT[1] + legacy)	108
	2 (HT[2])	58
	4 (HE[3])	122
80 MHz[4]	1 (VHT[1] + legacy)	234
	4 (HE[3])	250

[1] Subcarrier indices for which a compressed beamforming feedback matrix is sent back in Table 9-70 in ref.[2].
[2] Number of matrices and carrier grouping in Table 9-54 in ref.[2].
[3] Subcarrier indices for compressed beamforming feedback matrix in Table 9-91e in ref.[1].
[4] For Wi-Fi 6 product category: Maximum CSI data length for 80 MHz 2x4 (2 RX antennas receiving data from a 4-antenna device) = 1,033 Dwords.

3.3 CSI data format

CSI data is a set of complex values with:

- Nr x Nc complex values for each subcarrier. For example:
  - Nr = 2 and Nc = 1 translates as two complex CSI values per subcarrier.
  - Nr = 1 and Nc = 1 translates as one complex CSI values per subcarrier.
- Each complex value is stored as 2-byte value: 1 byte for the real part and 1 byte for the imaginary part. The imaginary part is in the format configured in `CSI format [1:0]` signal of PKT_info field signals (s8.3/4/5/6).

Table 8 shows the number of complex CSI values per subcarrier.

Table 8. Number of complex CSI values per subcarrier

	Nc = 1	Nc = 2	Nc = 3	Nc = 4
Nr = 1	1	2	3	4
Nr = 2	2	4	6	8

Figure 2 shows the example of CSI matrix with Nr = 2, Nc = 1, nTones = number of subcarriers = 52.

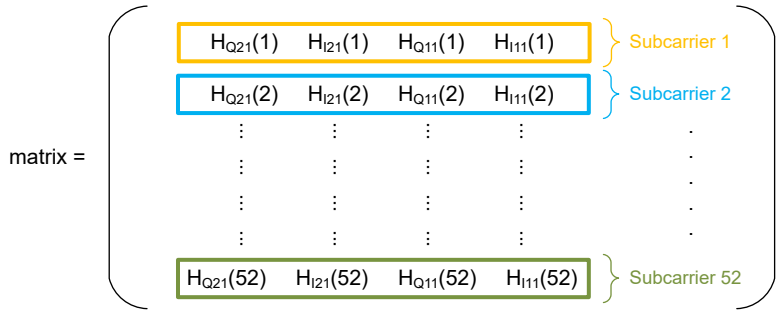

Figure 2. Example of CSI matrix

Where in CSI $matrix_{\alpha\beta y}(\delta)$:

- α: I = real part of the complex value. Q = imaginary part of the complex value
- β: Nr = Nrx = receiver antenna in the order (1, 2, ..., Nr)
- y: Nc = Ntx = spatial streams in the order (1, 2, ..., Nc)
- δ: subcarrier in the order (1, 2, 3, ..., nTones)

Table 9 shows an example of CSI matrix (Nc = 1 and Nr = 2).

Table 9. Example of CSI matrix (Nc = 1 and Nr = 2)

Dword	Byte 3	Byte 2	Byte 1	Byte 0
1	Q21(1)	I21(1)	Q11(1)	I11(1)[1]
2	Q21(2)[2]	I21(2)	Q11(2)	I11(2)
3	...	...	...	...
…	…	…	…	…
52	Q21(52)	I21(52)	Q11(52)	I11(52)

[1] I11(1): I = real part—1 = receiver antenna 1—1 = transmitter antenna 1—(1) = subcarrier 1
[2] Q21(2): Q = imaginary part—2 = receiver antenna 2—1 = transmitter antenna 1—(2) = subcarrier 2

Table 10 shows an example of CSI matrix (Nc = 2 and Nr = 2).

Table 10. Example of CSI matrix (Nc = 2 and Nr = 2)

Dword	Byte 3	Byte 2	Byte 1	Byte 0
1	Q12(1)	I12(1)	Q11(1)	I11(1)
2	Q22(1)	I22(1)	Q21(1)	H21(1)
3	Q12(2)	I12(2)	Q11(2)	I11(2)
4	Q22(2)	I22(2)	Q21(2)	H21(2)
…	…	…	…	…
51	Q12(51)	I12(51)	Q11(51)	I11(51)
52	Q22(52)	I22(52)	Q21(52)	I21(52)

Note: *The CSI data matrix is zero-padded to be DWORD aligned.*

4 CSI generation

The wifi_cli application is used to configure CSI generation and collection. The application is included in the SDK release. To generate CSI data, flash wi-fi_cli application onto the Wi-Fi device and run the following commands:

- `wlan-set-csi-param-header`
- `wlan-set-csi-filter`
- `wlan-csi-cfg`
- `wlan-auto-null-tx`

4.1 wlan-set-csi-param-header

The command is used to configure CSI data (headID, chipID, and channel) and to disable or enable CSI collection. The command is used in the wifi_cli example project. The command is not an API in the Wi-Fi driver.

Note: *Use the command* `wlan-csi-config` *to set the CSI data configurations.*

Syntax:

```
wlan-set-csi-param-header <mode> <csi_enable> <head_id> <tail_id> <chip_id> <band_config>
 <channel> <csi_monitor_enable> <ra4us>
```

Table 11. Command parameters

Parameter	Description
mode	sta = STA mode uap = uAP mode
csi_enable	Enable/disable CSI 1 = enable 2 = disable
head_id	User-defined 4-byte identifier in hexadecimal to represent the beginning of the CSI records. For example, 00010203.
tail_id	User-defined 4-byte identifier in hexadecimal to represent the end of the CSI records For example, 00010203
chip_id	User-defined 1-byte identifier in hexadecimal to represent the Wi-Fi device. For example 170.
band_config	1-byte configuration for the bandwidth. Used to set the channel band and device bandwidth in CSI monitor mode, when the device is not connected to an AP. Bit[0:1] = channel band 00 = 2.4 GHz 01 = 5 GHz Bit[2:3] = channel width (chanWidth) 00 = 20MHz 10 = 40MHz 11 = 80MHz Bit[4:5] = Secondary channel offset (chan2Offset) 00 = None 01 = Above 11 = Below Bit[6:7] = scan mode (scanMode) 00 = manual 01 = Auto channel select (ACS) 02 = Adoption mode
channel	Channel number. Channel used in CSI monitor mode when the device is not connected to an AP.

Table 11. Command parameters*...continued*

Parameter	Description
csi_monitor_enable	0 = normal mode (MAC filter enabled). CSI is generated from packets that match the transmitter address (TA) configured in one of the CSI filters. 1 = CSI monitor mode enabled. Generates CSI from any packet addressed to the device or broadcast, independent of the TA. Parameter value also used to configure the channel to use to collect CSI when the device is not connected to an AP.
ra4us	Receiver address (RA) for us. The RA field is part of the MAC header. 0 = normal mode. Generates CSI only if the receiver address (RA) is set to the MAC address of either the device or the broadcast address. 1 = monitor mode. Generates CSI independently of the receiver address.

Example of command:

```
#wlan-set-csi-param-header sta 1 66051 66051 170 0 1 1 1
```

Example of output:

```
The current csi_param is:
bss_type      : sta
csi_enable    : 1
head_id       : 66051
tail_id       : 66051
csi_filter_cnt: 0
chip_id       : 170
band_config   : 0
channel       : 1
csi_monitor_enable : 1
ra4us         : 1
```

4.2 `wlan-set-csi-filter`

This command is used to add, delete, clear, and list the CSI filters. Run the command multiple times to apply multiple CSI filters.

Note: *Use the command* `wlan-csi-config` *to set the CSI data configurations.*

Syntax:

```
#wlan-set-csi-filter <opt> <macaddr> <pkt_type> <subtype> <flag>
```

Table 12. Command parameters

Parameter	Description
`opt`	Option field used to define the action on the CSI filters. add = add up to 16 CSI filters delete = delete the most recent CSI filter clear = clear all CSI filters dump = list the CSI configurations and filters
`macaddr`	6 byte value that defines the MAC address of the device with which the firmware is communicating (in hexadecimal).
`pkt_type`	1 byte value that defines the Wi-Fi packet type (in hexadecimal) 0xFF = the firmware does not filter packets using the packet type.
`subtype`	1 byte value that defines the Wi-Fi packet subtype (in hexadecimal) 0xFF = the firmware does not filter packets using the packet subtypes.
`flag`	1 byte (in hexadecimal) • Bit[0] = 0, reserved, must be 0. • Bit[1] = 1, the Wi-Fi firmware waits for a triggering packet before sending a CSI event to the host. • Bit[2] = 1, the Wi-Fi firmware sends a CSI error event when a timeout occurs.

Example of command to add a CSI filter:

```
# wlan-set-csi-filter add 00:18:E7:ED:2D:C1 255 255 0
```

Example of command to dump the CSI filters:

```
# wlan-set-csi-filter dump
```

Example of output:

```
The current csi_param is:
bss_type           : sta
csi_enable         : 1
head_id            : 66051
tail_id            : 66051
csi_filter_cnt     : 1
chip_id            : 170
band_config        : 0
channel            : 1
csi_monitor_enable : 1
ra4us              : 1
mac_addr           : 00:18:E7:ED:2D:C1
pkt_type           : 255
subtype            : 255
flags              : 0
```

4.3 `wlan-csi-cfg`

The command is used to apply the CSI configurations to the firmware.

Syntax:

```
#wlan-csi-cfg
```

4.4 wlan-auto-null-tx

The command is used to configure the auto transmit and one shot quality of service data packets and stop.

Syntax:

```
wlan-auto-null-tx <mode> <opt> interval <interval> dst_mac <dst_mac>
```

Table 13. Command parameters

Parameter	Description
mode	Wi-Fi radio mode sta = station uap = access point
opt	Option start = start quality of service data packets stop = stop quality of service data packets
interval	Time interval and unit in hexadecimal. bit[15:14]: time interval unit 00 = seconds 01 = microseconds (us) 10 = milliseconds (ms) 11 = one shot bit[13:0] = time interval value
dst_mac	Destination MAC address Not applicable if connected to an AP.

Example of command to start auto transmit at 100 ms:

- Bit[15:14] = 10 = ms
- Bit[13:0] = 0x64 = 100

```
#wlan-auto-null-tx uap start interval 0x8064
```

Command to transmit one shot (single generation of CSI data):

- Bit[15:14] = 11
- Bit[13:0] = 0x64 = 100

```
wlan-auto-null-tx sta start interval 0xC064
```

Command to stop auto transmit:

```
#wlan-auto-null-tx stop
```

5 Processing CSI – Ambient motion index (AMI)

The CSI records can be processed to calculate the ambient motion index (AMI). AMI is a measure of change in CSI used to detect motion in the vicinity of the Wi-Fi STA and/or AP. AMI is expressed in dB.

Note: *Verify that the product supports CSI in Section 1.1.*

The SDK release includes the wifi_cli application. Two commands are used:

- `wlan-set-ami-cfg` is used to configure the AMI.
- `wlan-start-stop-ami` is used to calculate the AMI.

5.1 Enabling ambient motion index (AMI)

AMI is disabled by default in the SDK.

Note: *The feature is only supported with wifi_cli and wifi_wpa_supplicant applications.*

To enable AMI feature:

Step 1 – Edit *<SDK_PATH>/ boards/<HOST>/wifi_examples/wifi_cli/wifi_config.h*.

- Set `CONFIG_CSI_AMI=1`.

Step 2 – Update the properties in MCUXpresso IDE for RT1060-EVKC and RT1170-EVKB platforms.

- Go to *Project properties* > *C/C++ Build* > *Settings* > *Preprocessor*.
- Add `PRINTF_FLOAT_ENABLE=1`.

Step 3 – Build and flash wifi_cli application.

5.2 `wlan-set-ami-cfg`

The command is used to configure AMI.

Syntax:

```
wlan-set-ami-cfg mac <mac_address> type  <packet_type> ref <update_ref> bw <band_width>
 num <CSI_number>
```

Table 14. Command parameters

Parameters	Description
`mac_address`	Source MAC address of the packets to process CSI from. Should match one of the CSIfilterN MAC addresses defined by wlan-set-csi-filter.
`packet_type`	Select the packet type to process: 0 = legacy (11a/g), default 1 = HT (11n) 2 = VHT (11ac) 3 = HE (11ax)
`band_width`	Select the bandwidth to process: 0 = 20 MHz, default 1 = 40 MHz 2 = 80 MHz
`update_ref`	Reference update 0 = static (uses first CSI record with no updates) 1 = infinite impulse response (IIR) filter (initializes the reference with the first CSI record and updates the reference with the IIR filter using IIR filter coefficient alpha) 2 = Kalman filter (uses the first CSI record as reference and updates the record with the Kalman filter)
`CSI_number`	Number of CSI records to use: 0 = runs until CSI generation is stopped. Range = 1 to 255 (runs for this many CSI records)

Example of command:

```
# wlan-set-ami-cfg mac 7C:10:C9:02:DA:4C type 2 ref 2 bw 1 num 10
```

5.3 wlan-start-stop-ami

The command is used to start or stop the calculation of AMI.

Syntax:

```
wlan-start-stop-ami <action>
```

Table 15. Command parameters

Parameters	Description
action	1 = start to calculate Ambient Motion Index 0 = stop to calculate Ambient Motion Index

Example of command:

```
# wlan-start-stop-ami 1
```

Example of output:

```
Compare CSI filter set MAC: 7c.10.c9.02.da.4c, sig BW/format 1|2
NUM 1 CSI Processing Results: VHT(40), RX/TX 1/1, -440.08 TSF 19a285d7, Ambient Motion
 Index -3.2 dB
NUM 2 CSI Processing Results: VHT(40), RX/TX 1/1, 103.29 TSF 1a45073a, Ambient Motion
 Index -3.1 dB
NUM 3 CSI Processing Results: VHT(40), RX/TX 1/1, -482.94 TSF 1a450887, Ambient Motion
 Index -3.3 dB
NUM 4 CSI Processing Results: VHT(40), RX/TX 1/1, -441.77 TSF 1a864347, Ambient Motion
 Index -4.3 dB
NUM 5 CSI Processing Results: VHT(40), RX/TX 1/1, -519.32 TSF 1b75b749, Ambient Motion
 Index -3.2 dB
NUM 6 CSI Processing Results: VHT(40), RX/TX 1/1, 168.90 TSF 1bb771ef, Ambient Motion
 Index -2.7 dB
NUM 7 CSI Processing Results: VHT(40), RX/TX 1/1, -549.59 TSF 1c03e5b5, Ambient Motion
 Index -4.2 dB
NUM 8 CSI Processing Results: VHT(40), RX/TX 1/1, 221.89 TSF 1ca668c5, Ambient Motion
 Index -4.5 dB
NUM 9 CSI Processing Results: VHT(40), RX/TX 1/1, 208.22 TSF 1ce89f14, Ambient Motion
 Index -8.0 dB
NUM 10 CSI Processing Results: VHT(40), RX/TX 1/1, -10.70 TSF 1dd716e8, Ambient Motion
 Index -4.1 dB
```

6 CSI dump details

CSI data is not dumped to the console by default. Users must register a callback to receive the CSI data in their application. Users can add the callback in */wifi/wifidriver/wifi.c*.

```
csi_data_recv_user (void* buffer, t_u16 data_len)
{
pcsi_record_ds data = pcsi_record_ds(buffer);
(void)PRINTF("Len :%d \r\n", data ->Len);
}
register_csi_user_callback(csi_data_recv_user)
```

Once enabled, the CSI data is displayed on the console. The data contains:

- All the CSI records printed in Dword text format. Each Dword is separated with a space.
- A signature with:
  - CSI record = 0xABCD
  - Length (in Dwords)
- (CSI header + CSI data) * N, where N = number of CSI records

CSI event console print example:

```
# CSI user callback: Event CSI data
**** Dump @ 20027EDC Len: 156 ****
27 00 cd ab 03 02 01 00 00 00 40 00 58 92 90 2f
00 00 00 00 ff ff ff ff ff ff c8 7f 54 de 20 74
e0 00 a1 00 3f 64 03 aa 80 00 00 00 1b 00 13 01
06 f5 06 f5 06 f5 01 f3 fc f3 f7 f5 f4 f9 f3 01
f4 05 f6 08 f9 0a fc 0b ff 0b 02 0a 04 09 05 07
06 05 05 04 05 02 04 01 01 01 00 02 00 04 00 05
01 07 03 09 07 0a 0a 0a 0d 09 10 07 12 04 14 01
13 f8 13 f4 0f f1 0c ed 07 ec 02 ea fd eb f9 ee
f4 f1 f1 f5 f1 fb f2 00 f4 04 fd 08 01 07 04 04
05 00 03 fd 03 fd 03 fd 03 02 01 00
******** End Dump *******
```

Decoding of the above CSI record example (refer to Section 3.2 for the parameter definitions):

- `0027` = CSI record length in hexadecimal
- `abcd` = signature
- `03020100` = HeaderID—same as what is defined in *csi.conf* file
- `00004000` = 0x0000: reserved, 0x4000: Packet_INFO in hexadecimal (refer to Table 3)
- `2f90925800 00000000` = TSF fields
- `ffffffffffff` = Destination MAC for the CSI packets
- `7420de547fc8` = Source MAC ID of CSI packets
- `0xe0` = RX_RSSI_A
- `0x00` = RX_RSSI_B
- `0xa1` = RX_NF_A
- `0x00` = RX_NF_B
- `0x3f` = SINR
- `0x64` = channel
- `0x03` = AP_type
- `0xaa` = chip ID
- `0x0080` = FCF
- `0x00` = total gain

- `0x00` = reserved
- `0x001b` = CSI data length
- `fd03fd03 fd030005 04040701 08fd04f4 00f2fbf1 f5f1f1f4 eef9ebfd ea02ec07 ed0cf10f f413f813 01140412 0710090d 0a0a0a07 09030701 05000400 02000101 01040205 04050506 07050904 0a020bff 0bfc0af9 08f605f4 01f3f9f4 f5f7f3fc f301f506 f506f506` = CSI data
- `03020100` = TailID

7 Examples

This section includes examples of configurations to generate CSI data. A CSI record is generated each time the Wi-Fi device receives a packet that matches the filters specified with the command `wlan-set-csi-filter .`

7.1 Beacons

This example generates CSI data for every beacon received.

Note: *CSI data is generated from OFDM packets. 802.11b beacons cannot be used.*

Step 1 – Flash wifi_cli application onto the Wi-Fi device.

Step 2 – Set the device in STA mode and connect to an AP.

```
#wlan-add test ssid AX6wpa3
#wlan-connect test
```

Step 3 – Configure the CSI parameters and add CSI filters.

```
#wlan-set-csi-param-header sta 1 66051 66051 170 1 40 0 0
#wlan-set-csi-filter add C8:7F:54:DE:20:70 255 08 0
```

Where:

- `C8:7F:54:DE:20:70` = MAC address that the firmware is communicating with.
- `255` = no packet type filter
- `08` = subtype (subtype filter for the beacon)
- `00` = no other filters are applied

Step 4 – Enable CSI.

```
#wlan-csi-cfg
```

Step 5 – Disable CSI.

```
#wlan-set-csi-param-header sta 2 66051 66051 170 0 11 0 1
#wlan-csi-cfg
```

7.2 Management packets

The example uses received management packets to generate CSI data. The rate at which the CSI is generated depends on the packet type being filtered.

Step 1 – Flash wifi_cli application onto the Wi-Fi device.

Step 2 – Set the device in STA mode and connect to an AP.

```
#wlan-add test ssid AX6wpa3
#wlan-connect test
```

Step 3 – Configure the CSI parameters and add CSI filters.

```
#wlan-set-csi-param-header sta 1 66051 66051 170 1 40 0 0
#wlan-set-csi-filter add C8:7F:54:DE:20:70 00 255 0
```

Where:

- `C8:7F:54:DE:20:70` = MAC address that the firmware is communicating with.
- `00` = management packet type
- `255` = no subtype filter
- `00` = no other filters

Step 4 – Enable CSI.

```
#wlan-csi-cfg
```

Step 5 – Disable CSI.

```
#wlan-set-csi-param-header sta 2 66051 66051 170 0 11 0 1
#wlan-csi-cfg
```

7.3 Data packets

The example uses received data packets to generate CSI data. The rate at which the CSI is generated depends on the packet type being filtered.

Step 1 – Flash wifi_cli application onto the Wi-Fi device.

Step 2 – Set the device in STA mode and connect to an AP.

```
#wlan-add test ssid AX6wpa3
#wlan-connect test
```

Step 3 – Configure the CSI parameters and add CSI filters.

```
#wlan-set-csi-param-header sta 1 66051 66051 170 1 40 0 0
#wlan-set-csi-filter add C8:7F:54:DE:20:70 02 255 0
```

Where:

- `C8:7F:54:DE:20:70` = MAC address that the firmware is communicating with
- `02` = data packet
- `255` = no subtype filter
- `00` = no other filters

Step 4 – Enable CSI.

```
#wlan-csi-cfg
```

Step 5 – Ping the AP for data traffic.

The ping command is used to send data packets over the air and generate CSI data.

Step 6 – Disable CSI.

```
#wlan-set-csi-param-header sta 2 66051 66051 170 0 11 0 1
#wlan-csi-cfg
```

7.4 QoS null packets

The uAP sends null data packets to the STA. The STA receives the packets to generate CSI data. Device #1 acts as a uAP and device #2 acts as the STA.

7.4.1 Set up the uAP to send QoS null packets

Step 1 – Flash wifi_cli application onto the Wi-Fi device.

Step 2 – Configure device #1 in uAP mode and associate with the STA.

Step 3 – Configure and start the null data packet transmission. Refer to Section 4.4.

```
#wlan-auto-null-tx uap start interval 0x8064
```

Step 4 – Disable null data packet transmission. Refer to Section 4.4.

```
#wlan-auto-null-tx uap stop
```

7.4.2 Configure the STA to generate CSI from the received QoS null packets

The example uses received data packets to generate CSI data. The rate at which the CSI is generated depends on the packet type being filtered.

Step 1 – Flash wifi_cli application onto the Wi-Fi device.

Step 2 – Set the device in STA mode and connect to an AP.

```
#wlan-add test ssid AX6wpa3
#wlan-connect test
```

Step 3 – Configure the CSI parameters and add CSI filters.

```
#wlan-set-csi-param-header sta 1 66051 66051 170 1 40 0 0
#wlan-set-csi-filter add C8:7F:54:DE:20:70 02 255 0
```

Where:

- `C8:7F:54:DE:20:70` = MAC address that the firmware is communicating with
- `02` = data packet
- `255` = no subtype filter
- `00` = no other filters

Step 4 – Enable CSI.

```
#wlan-csi-cfg
```

Step 5 – Verify that the uAP is sending null data packets to the STA (step 3 in Section 7.4.1).

Step 6 – Disable CSI.

```
#wlan-set-csi-param-header sta 2 66051 66051 170 0 11 0 1
#wlan-csi-cfg
```

7.5 Connectionless or monitor mode

The STA generates CSI data from management packets received in an connectionless state. The parameter `csi_monitor_enable` must be enabled with `wlan-set-csi-param-header` command (Section 4.1).

Step 1 – Flash wifi_cli application onto the Wi-Fi device.

Step 2 – Configure the CSI parameters and add CSI filters.

```
#wlan-set-csi-param-header sta 1 66051 66051 170 1 40 1 0
#wlan-set-csi-filter add C8:7F:54:DE:20:70 00 255 0
```

Where:

- `C8:7F:54:DE:20:70` = MAC address that the firmware is communicating with
- `00` = management packet type
- `255` = no subtype filter
- `00` = no other filters

Step 4 – Enable CSI.

```
#wlan-csi-cfg
```

Step 6 – Disable CSI.

```
#wlan-set-csi-param-header sta 2 66051 66051 170 0 11 1 1
#wlan-csi-cfg
```

7.6 Multi-APs

The STA generates CSI data from multiple AP beacons received in an connectionless state. The parameter `csi_monitor_enable` must be enabled in *csi.conf* file.

Step 1 – Flash wifi_cli application onto the Wi-Fi device.

Step 2 – Configure the CSI parameters and add CSI filters.

Note:

- *To enable the CSI data collection from the beacons sent by all the nearby access points, set the MAC address to `00:00:00:00:00:00`.*
- *To restrict the CSI data collection is restricted to the beacons from a specific access point, specify the MAC address of the access point.*

```
#wlan-set-csi-param-header sta 1 66051 66051 170 1 40 1 0
#wlan-set-csi-filter add 00:00:00:00:00:00 00 08 0
```

Where:

- `00:00:00:00:00:00` = any MAC address
- `00` = management packet
- `08` = beacon
- `00` = no other filters

Step 3 – Enable CSI.

```
#wlan-csi-cfg
```

Step 4 – Disable CSI.

```
#wlan-set-csi-param-header sta 2 66051 66051 170 0 11 1 1
#wlan-csi-cfg
```

7.7 Multi-clients

The uAP generates CSI data from data packets received from two clients.

Step 1 – Flash the wifi_cli application onto the Wi-Fi device.

Step 2 – Configure the device #1 in uAP mode and associate the uAP with two clients.

Step 3 – Configure the CSI parameters and add CSI filters.

```
#wlan-set-csi-param-header sta 1 66051 66051 170 1 40 0 0
#wlan-set-csi-filter add 4E:2C:ED:B9:47:27 02 255 0
#wlan-set-csi-filter add D4:54:8B:61:8E:8D 02 255 0
```

Parameters of csifilter0:

- `4E:2C:ED:B9:47:27` = MAC address of the first client that the uAP is generating CSI from
- `02` = data packet
- `255` = no subtype filter
- `00` = no other filters

Parameters of csifilter1:

- `D4:54:8B:61:8E:8D` = MAC address of the second client that the uAP is generating CSI from
- `02` = data packet
- `255` = no subtype filter
- `00` = no other filters

Step 3 – Enable CSI.

```
#wlan-csi-cfg
```

Step 4 – Disable CSI.

```
#wlan-set-csi-param-header sta 2 66051 66051 170 0 11 1 1
#wlan-csi-cfg
```

7.8 Ambient motion detection

The section includes examples of AMI calculation with motion and with no motion.

7.8.1 No motion

The low AMI value is displayed to indicate that there is little to no motion near the STA.

Step 1 – Flash wifi_cli application onto the Wi-Fi device.

Step 2 – Set the device in STA mode and connect to an AP.

```
# wlan-add test ssid ASUS_5G wpa2 psk 12345678
# wlan-connect test
```

Step 3 – Configure the CSI parameters and add CSI filters.

```
# wlan-set-csi-param-header sta 1 66051 66051 170 141 36 0 1
# wlan-set-csi-filter add 7C:10:C9:02:DA:4C 2 8 0
```

Step 4 – Enable CSI.

```
# wlan-csi-cfg
```

Step 5 – Set AMI.

```
# wlan-set-ami-cfg mac 7C:10:C9:02:DA:4C type 3 ref 2 bw 1 num 10
```

Step 6 – Start/stop the calculation of the ambient motion index (AMI).

```
# wlan-start-stop-ami 1
```

Example of output:

```
Compare CSI filter set MAC: 7c.10.c9.02.da.4c, sig BW/format 1|3
NUM 1 CSI Processing Results: HE(40), RX/TX 1/1, -426.30  TSF 13dbebff, Ambient Motion
 Index -30.1 dB
NUM 2 CSI Processing Results: HE(40), RX/TX 1/1, -421.78  TSF 14bd6632, Ambient Motion
 Index -24.5 dB
NUM 3 CSI Processing Results: HE(40), RX/TX 1/1, -419.56  TSF 150d163b, Ambient Motion
 Index -30.5 dB
NUM 4 CSI Processing Results: HE(40), RX/TX 1/1, -419.80  TSF 1559a5fe, Ambient Motion
 Index -28.0 dB
NUM 5 CSI Processing Results: HE(40), RX/TX 1/1, 161.04   TSF 15ee1743, Ambient Motion
 Index -30.0 dB
NUM 6 CSI Processing Results: HE(40), RX/TX 1/1, -414.79  TSF 163e4731, Ambient Motion
 Index -10.8 dB
NUM 7 CSI Processing Results: HE(40), RX/TX 1/1, 169.94   TSF 17205721, Ambient Motion
 Index -25.0 dB
NUM 8 CSI Processing Results: HE(40), RX/TX 1/1, 159.91   TSF 176f7614, Ambient Motion
 Index -19.2 dB
NUM 9 CSI Processing Results: HE(40), RX/TX 1/1, -62.99   TSF 17bc95ef, Ambient Motion
 Index -25.7 dB
NUM 10 CSI Processing Results: HE(40), RX/TX 1/1, 554.75   TSF 1851076c, Ambient Motion
 Index -17.8 dB
```

The AMI outputs range from -17 dB to -30 dB, which is considered as a small AMI. No motion was detected.

7.8.2 Motion

The high AMI value is displayed to indicate that there is some motion near the STA.

Step 1 – Flash wifi_cli application onto the Wi-Fi device.

Step 2 – Set the device in STA mode and connect to an AP.

```
# wlan-add test ssid ASUS_5G wpa2 psk 12345678
# wlan-connect test
```

Step 3 – Configure the CSI parameters and add CSI filters.

```
# wlan-set-csi-param-header sta 1 66051 66051 170 141 36 0 1
# wlan-set-csi-filter add 7C:10:C9:02:DA:4C 2 8 0
```

Step 4 – Enable CSI.

```
# wlan-csi-cfg
```

Step 5 – Set AMI configurations.

```
# wlan-set-ami-cfg mac 7C:10:C9:02:DA:4C type 3 ref 2 bw 1 num 10
```

Step 6 – Walk back and forth by the STA.

Step 7 – Start/stop the calculation of ambient motion index (AMI).

```
# wlan-start-stop-ami 1
```

Example of output:

```
Compare CSI filter set MAC: 7c.10.c9.02.da.4c, sig BW/format 1|2
NUM 1 CSI Processing Results: VHT(40), RX/TX 1/1, -440.08 TSF 19a285d7, Ambient Motion
 Index -3.2 dB
NUM 2 CSI Processing Results: VHT(40), RX/TX 1/1, 103.29 TSF 1a45073a, Ambient Motion
 Index -3.1 dB
NUM 3 CSI Processing Results: VHT(40), RX/TX 1/1, -482.94 TSF 1a450887, Ambient Motion
 Index -3.3 dB
NUM 4 CSI Processing Results: VHT(40), RX/TX 1/1, -441.77 TSF 1a864347, Ambient Motion
 Index -4.3 dB
NUM 5 CSI Processing Results: VHT(40), RX/TX 1/1, -519.32 TSF 1b75b749, Ambient Motion
 Index -3.2 dB
NUM 6 CSI Processing Results: VHT(40), RX/TX 1/1, 168.90 TSF 1bb771ef, Ambient Motion
 Index -2.7 dB
NUM 7 CSI Processing Results: VHT(40), RX/TX 1/1, -549.59 TSF 1c03e5b5, Ambient Motion
 Index -4.2 dB
NUM 8 CSI Processing Results: VHT(40), RX/TX 1/1, 221.89 TSF 1ca668c5, Ambient Motion
 Index -4.5 dB
NUM 9 CSI Processing Results: VHT(40), RX/TX 1/1, 208.22 TSF 1ce89f14, Ambient Motion
 Index -8.0 dB
NUM 10 CSI Processing Results: VHT(40), RX/TX 1/1, -10.70 TSF 1dd716e8, Ambient Motion
 Index -4.1 dB
```

The AMI outputs range from -3 dB to -8 dB, which corresponds to a larger AMI. Motion was detected.

8 Sample applications

Matlab, Python, or similar tools can be used to process CSI data records. The following are examples of applications using the CSI data.

8.1 Amplitude and phase graphs

To get the graphs:

Step 1 - Identify the nRx, nTx, and nTones values from the CSI records (Section 3.1).

Step 2 - Get r, t, and k.

- r = nRx
- t = nTx
- k = nTones

Step 3 - Calculate the gain in dB.

$$G_{dB}^{(rt)}(k) = 10\log_{10}\left(\left(H_{Irt}(k)\right)^2 + \left(H_{Qrt}(k)\right)^2\right)$$

Step 4 - Calculate the phase in radians.

$$\varphi_{rad}^{(rt)}(k) = \tan^{-1}\left({H_{Qrt}(k)}/{H_{Irt}(k)}\right)$$

Figure 3 shows the relative gain at the reported subcarriers (in dB) for both receiver antennas.

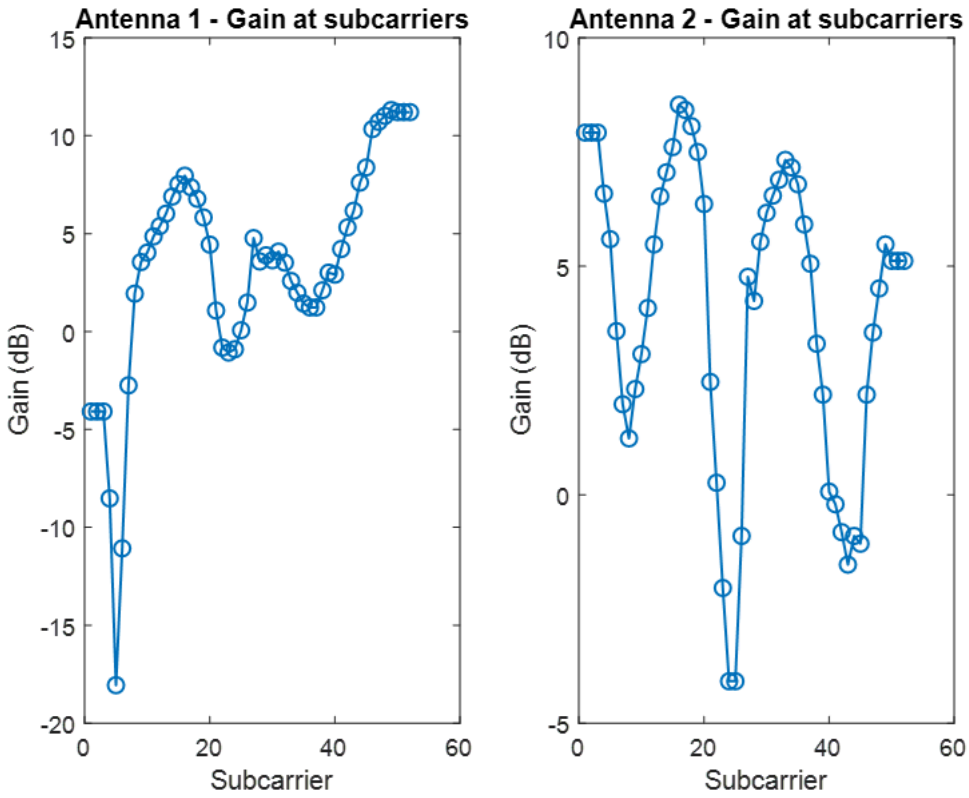

Figure 3. Relative gain at the reported subcarriers

Figure 4 shows the phase across the reported subcarriers for both receiver antennas.

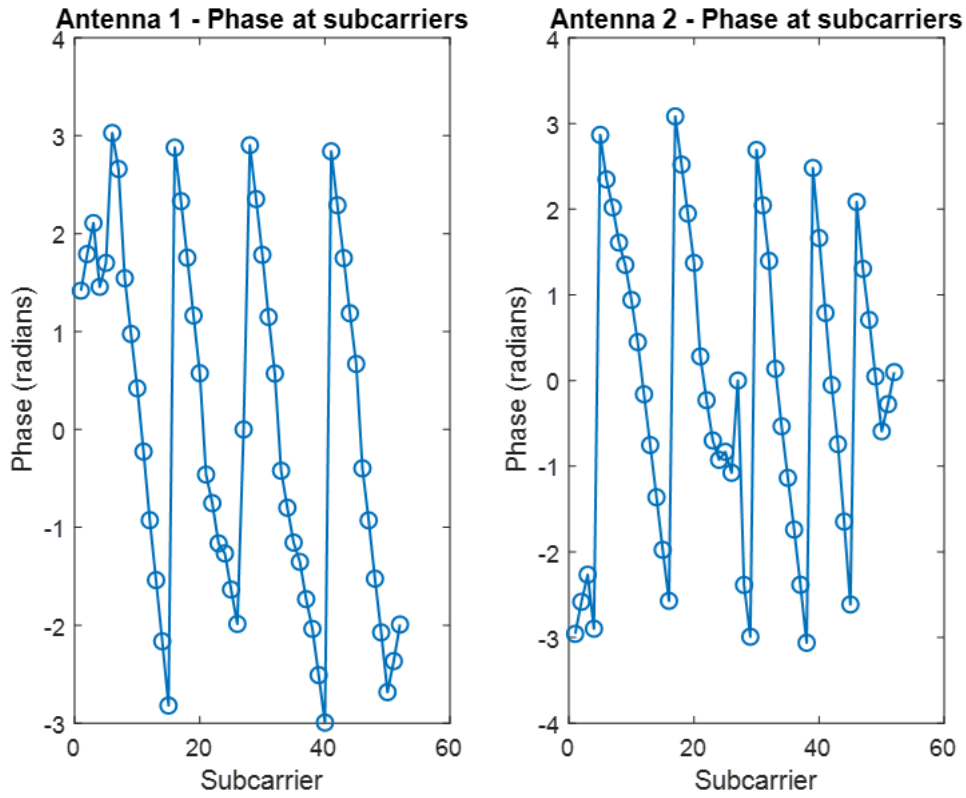

Figure 4. Phase across reported subcarriers for receive antennas

8.2 Impulse response and delay

Channel Impulse Response (CIR) = 0-padded Inverse Fast Fourier Transform (IFFT) of CSI data.

Figure 5 shows the channel impulse response for the two antennas.

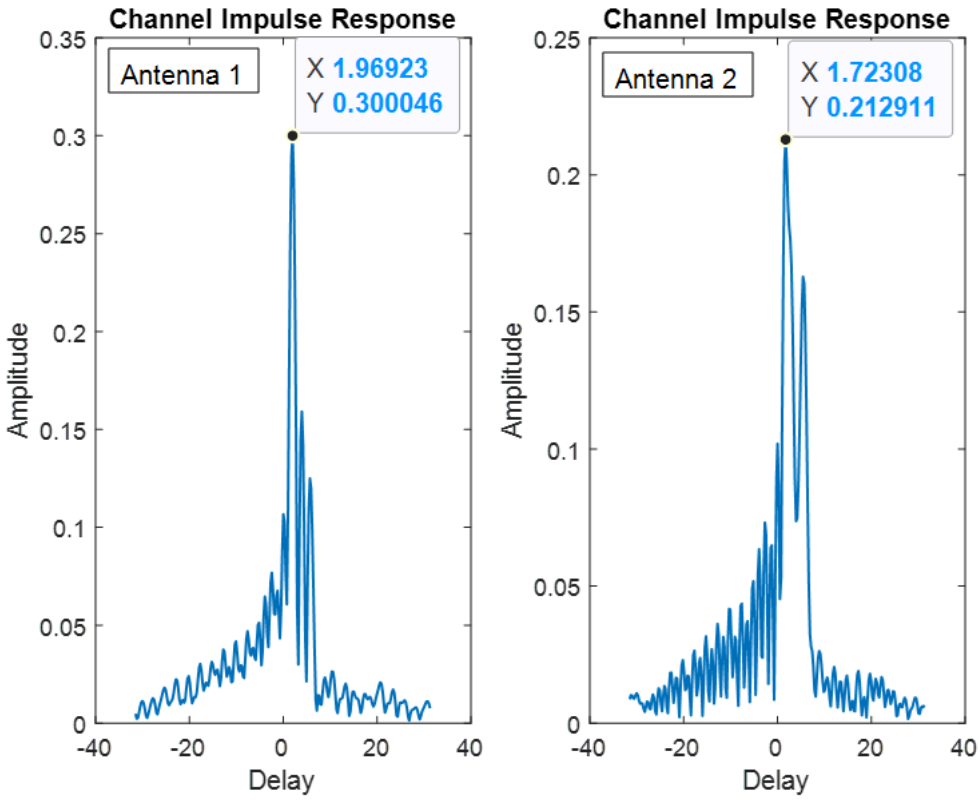

Figure 5. Channel impulse response

9 Appendix

9.1 s8.n fixed-point format

Fixed-point format is a way to present fractional numbers using integers and fixed decimals. Based on *csi.conf*, the CSI format is in s8.3, s8.4, s8.5, or s8.6 fixed-point format.

- s8.3 fixed-point format: 8 bits total with 5 bits as integers and 3 bits as decimal
- s8.4 fixed-point format: 8 bits total with 4 bits as integers and 4 bits as decimal
- s8.5 fixed-point format: 8 bits total with 3 bits as integers and 5 bits as decimal
- s8.6 fixed-point format: 8 bits total with 2 bits as integers and 6 bits as decimal

Figure 6 shows a visual representation of s8.3 fixed-point format.

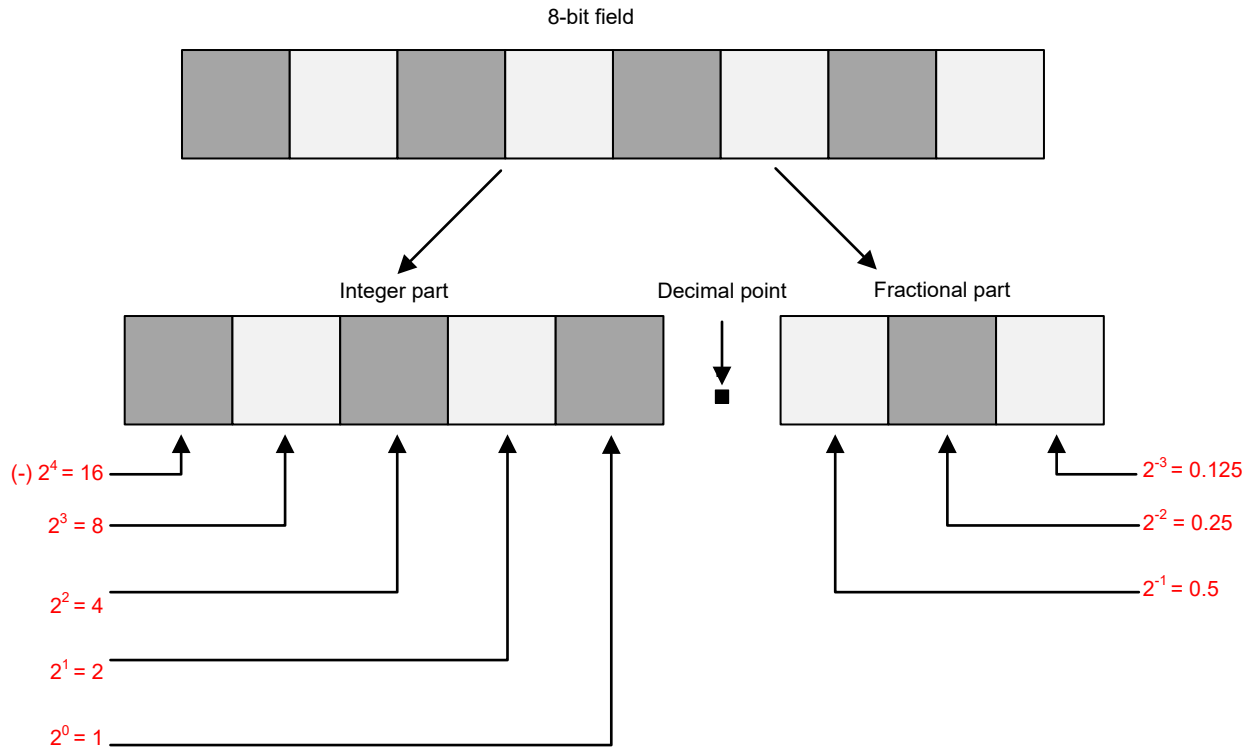

Figure 6. S8.3 fixed-point format

Example 1 – Positive number

0x16 = 0001 0110 (can be padded with leading 0s to make 8 bits)

The decimal point is placed 3 bits from the right.

00010 . 110

= 2.75

The left part of the decimal includes the positive exponents. The first bit indicates if the number is positive (0) or negative (1).

The right part of the decimal includes the negative exponents.

This example is for a positive number (0).

$2^1 + 2^{-1} + 2^{-2}$

= 2 + 0.5 + 0.25

Example 2 – Negative number

0xCA = 1100 1010

The decimal point is placed 3 bits from the right.

11001 . 010

The left part of the decimal includes the positive exponents. The first bit indicates if the number is positive (0) or negative (1).

The right part of the decimal includes the negative exponents.

Example 2 is for a negative number (1).

$-2^4 + 2^3 + 2^0 + 2^{-2}$

= -16 + 8 + 1 + 0.25 (starts as -16 because of 2’s complement)

= -6.75

10 References

[1] Published amendment to a standard – IEEE 802.11ax-2021 (link)
[2] Standard – IEEE 802.11-2016 (link)
[3] Web page – MCUXpresso SDK Builder (link)
[4] User manual – UM11799: NXP Wi-Fi and Bluetooth Demo Applications for RW61x (link)
[5] User manual – UM11442: NXP Wi-Fi and Bluetooth Demo Applications for i.MX RT platforms (link)

11 Abbreviations

Table 16. Abbreviations

Abbreviation	Description
ACS	auto channel select
AMI	ambient motion index
BCC	binary convolutional coding
BW	bandwidth
CIR	channel impulse response
CSI	channel state information
DCM	dual carrier modulation
GF	greenfield
GI	guard interval
HE	high efficiency
HT	high throughput
IFFT	inverse fast Fourier transform
LDPC	low-density parity-check
LG	legacy
LTF	long training field
MU	multiple users
NF	noise floor
OFDM	orthogonal frequency division multiplexing
PKT	packet
PSB	primary subband
RA	receiver address
RSSI	received signal strength indication
RSVD	reserved
SINR	signal to interference noise ratio
SS	spatial stream
STA	station
STBC	space time block code
SU	single user
TA	transmitter address
TSF	time synchronization function
uAP	mobile access point
VHT	very high throughput

12 Note about the source code in the document

The example code shown in this document has the following copyright and BSD-3-Clause license:

Copyright 2026 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials must be provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

13 Revision history

Table 17. Revision history

Document ID	Release date	Description
AN14281 v.4.0	7 January 2026	• Initial public release
AN14281 v.3.0	23 September 2025	• Section 1 "About this document": updated. • Section 1.1 "Supported products": updated. • Section 2 "Modules and flow ": updated. • Section 3.1 "CSI record format": updated Dword2 and Dword10. • Section 3.2 "Field descriptions": updated. • Section 3.3 "CSI data format": updated. • Section 4.2 "wlan-set-csi-filter": updated. • Section 4.4 "wlan-auto-null-tx": updated. • Section 5 "Processing CSI – Ambient motion index (AMI)": added. • Section 6 "CSI dump details": updated. • Section 7.1 "Beacons": updated. • Section 7.2 "Management packets": updated. • Section 7.3 "Data packets": added. • Section 7.4 "QoS null packets": added. • Section 7.5 "Connectionless or monitor mode": added. • Section 7.6 "Multi-APs": added. • Section 7.7 "Multi-clients": added. • Section 7.8 "Ambient motion detection": added. • Section 9.1: updated the section title and introduction.
AN14281 v.2.0	7 May 2025	• Section 2 "Modules and flow ": removed host-based supplicant. • Section 3.2 "Field descriptions": – Table 3: corrected TSF, moved FCF, updated the definitions of RX_RSSI and SINR. • Section 3.3 "CSI data format": added Table 10. • Table 11 "Command parameters": corrected the bit fields in `band_config` parameter description. • Section 4.4 "wlan-auto-null-tx": updated the description of `interval` parameter.
AN14281 v.1.0	22 August 2024	• Initial version

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Suitability for use in automotive applications — This NXP product has been qualified for use in automotive applications. If this product is used by customer in the development of, or for incorporation into, products or services (a) used in safety critical applications or (b) in which failure could lead to death, personal injury, or severe physical or environmental damage (such products and services hereinafter referred to as "Critical Applications"), then customer makes the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, safety, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. As such, customer assumes all risk related to use of any products in Critical Applications and NXP and its suppliers shall not be liable for any such use by customer. Accordingly, customer will indemnify and hold NXP harmless from any claims, liabilities, damages and associated costs and expenses (including attorneys' fees) that NXP may incur related to customer's incorporation of any product in a Critical Application.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Amazon Web Services, AWS, the Powered by AWS logo, and FreeRTOS — are trademarks of Amazon.com, Inc. or its affiliates.

Bluetooth — the Bluetooth wordmark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by NXP Semiconductors is under license.

MATLAB — is a registered trademark of The MathWorks, Inc.

Tables

Figures

Contents