Realtek 802.11k Development guide

Date: 2014/12/15
Version: 1.0

CHANGE HISTORY

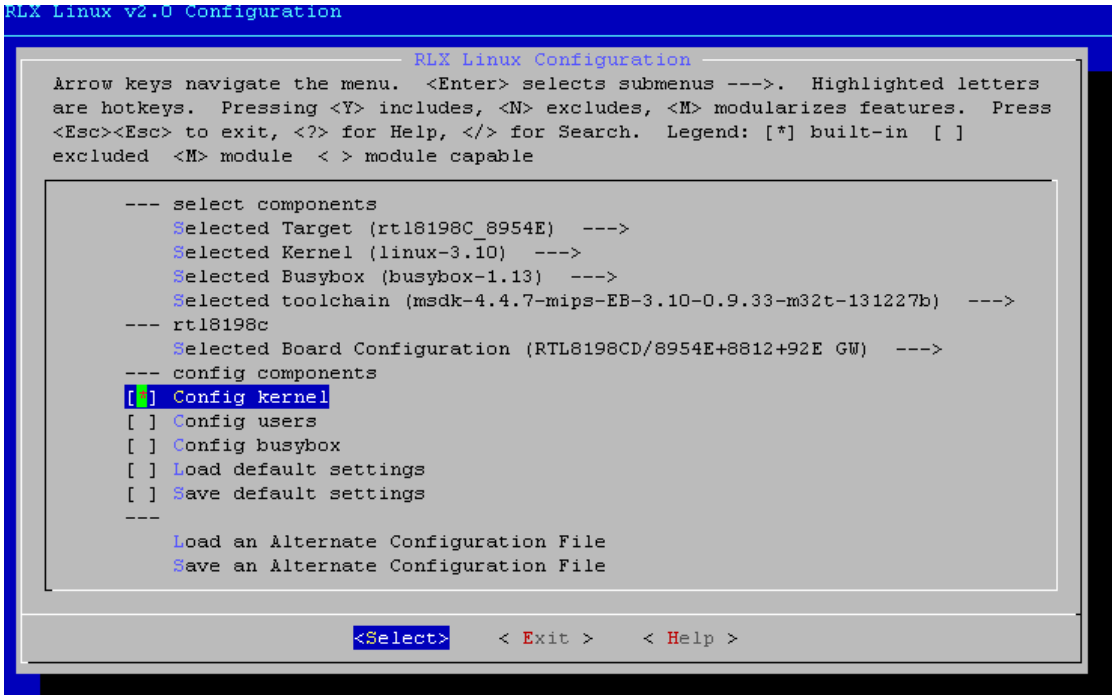
VERSION	DATE	REMARKS
1.0	2014/12/15	INITIAL RELEASE

TABLE OF CONTENTS

1. How to enable 802.11k function

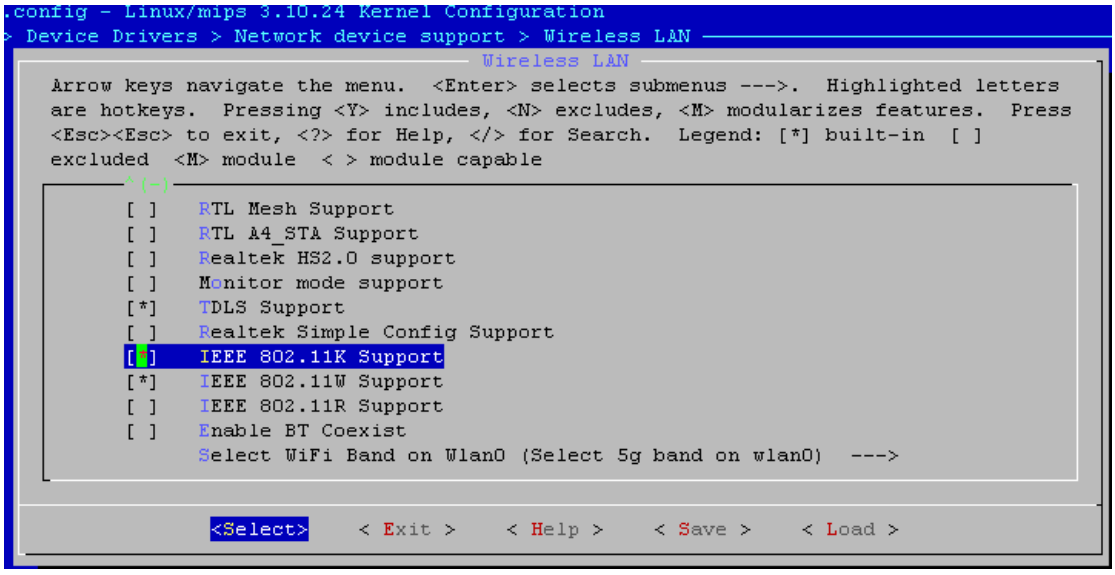
To enable 802.11k support, select the “IEEE 802.11K Support“ in menuconfig, as following demonstrated.

Select “Config kernel”



, and then exist.

Select option of “Device Drivers”->“Network device support”->“Wireless LAN”, you would see bellow figure.



Make sure “IEEE 802.11K Support” is selected

Compile SDK to generate new image.

2. Related MIBs

Name	Meaning	Value	Default	Comment
rm_activated	Activate/Deactivate 802.11k feature	0 – activate, 1 - deactivate		
rm_link_measure	Enable/Disable link measurement feature	0 – disable, 1 - enabl	1	This MIB is valid only when rm_activated is 1
rm_neighbor_report	Enable/Disable neighbor report feature	0 – disable, 1 - enabl	1	This MIB is valid only when rm_activated is 1
rm_beacon_passive	Enable/Disable passive beacon measurement feature	0 – disable, 1 - enabl	1	This MIB is valid only when rm_activated is 1
rm_beacon_active	Enable/Disable active beacon measurement feature	0 – disable, 1 - enabl	1	This MIB is valid only when rm_activated is 1
rm_beacon_table	Enable/Disable table beacon measurement feature	0 – disable, 1 - enabl	1	This MIB is valid only when rm_activated is 1
rm_ap_channel_report	Enable/Disable AP channel report ie brought in beacon frame	0 – disable, 1 - enabl	1	This MIB is valid only when rm_activated is 1
cubeaconintval	The number of consecutive beacon intervals during which the channel busy time is measured. The channel busy time is used by BSS Load ie which is carried in beacon and probe rsp frame.	1 ~ 20	10	This MIB is valid only when rm_activated is 1
countrystr	Country string	“string_value”, 2 characters in max	“US”	This MIB is valid only when countrycode is 1 OR rm_activated is 1 OR tpc_enable is 1
tpc_tx_power	The transmit power carried in the TPC report element	Two’s complement integer value in units of dBm	12	This MIB is valid only when tpc_enable is 1 OR rm_activated is 1
tpc_link_margin	The link margin carried in TPC report element	Two’s complement integer value in units of dBm		This MIB is valid only when tpc_enable is 1 OR rm_activated is 1
lpwc	The local power constraint	Unsigned integer		This MIB is valid only when

	carried in the power constraint element	value in units of dBm		tpc_enable is 1 OR rm_activated is 1
min_tx_power	The minimum transmit power carried in power capability element	Two's complement integer value in units of dBm		This MIB is valid only when tpc_enable is 1 OR rm_activated is 1
max_tx_power	The maximum transmit power carried in power capability element	Two's complement integer value in units of dBm	20	This MIB is valid only when tpc_enable is 1 OR rm_activated is 1

*The default value of MIB will be '0' if it is not specified

Table 1

3. Activate/Deactivate 802.11k Example

- Bellow is an example to activate 802.11k function and all supported measurement methods.

```
ifconfig wlan0 down
iwpriv wlan0 set_mib rm_activated =1
iwpriv wlan0 set_mib rm_link_measure=1
iwpriv wlan0 set_mib rm_beacon_passive=1
iwpriv wlan0 set_mib rm_beacon_active=1
iwpriv wlan0 set_mib rm_beacon_table=1
iwpriv wlan0 set_mib rm_neighbor_report = 1
iwpriv wlan0 set_mib rm_ap_channel_report=1
ifconfig wlan0 up
```

- Bellow is an example to deactivate 802.11k function.

```
ifconfig wlan0 down
iwpriv wlan0 set_mib rm_activated=0
ifconfig wlan0 up
```

4. PROC

- To add an AP channel report to AP side

#echo add [op_class] [channel1] [channel2] [channel3] … >
/proc/wlan0/rm_ap_channel_report

For example:

```
#echo add 115 36 48 > /proc/wlan0/rm_ap_channel_report
```

Current supported operating class for AP channel reports are as following:

Operating classes in the United States	
Operating class	**Channel set**
1	36, 40, 44, 48
2	52, 56, 60, 64
3	149, 153, 157, 161
4	100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
5	149, 153, 157, 161, 165
12	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

Operating classes in Europe	
Operating class	**Channel set**
1	36, 40, 44, 48
2	52, 56, 60, 64
3	100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
4	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13
17	149, 153, 157, 161, 165, 169

Operating classes in Japan	
Operating class	**Channel set**
1	36, 40, 44, 48
30	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13
32	52, 56, 60, 64
33	52, 56, 60, 64
34	100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
35	100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140

Global Operating classes	
Operating class	**Channel set**
81	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13
115	36, 40, 44, 48
118	52, 56, 60, 64
121	100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
124	149, 153, 157, 161
125	149, 153, 157, 161, 165, 169

Table 2.

- To delete an AP channel report from AP side

#echo del op_class > /proc/wlan0/rm_ap_channel_report
For example:

```
#echo del 115 > /proc/wlan0/rm_ap_channel_report
```

Above two commands are only available on AP side. If the MIB rm_ap_channel_report is “1”, the AP channel reports would be brought in beacon and probe rsp frame.

- Show current AP channel reports

```
# cat /proc/wlan0/rm_ap_channel_report
  -- AP CHANNEL REPORT info --
    Operating Class:115
          Channel List:44 48
    Operating Class:125
          Channel List:149 157
```

This command is available both on AP and Client side. On client side, the AP channel reports shown are collected by client from the associated AP.

- To add a neighbor report to AP side

#echo add [mac address] [bss info] [op_class] [channel] [phy type] [ssid] > /proc/wlan0/rm_neighbor_report
For example:

```
#echo add 00e04c000001 0xE7 115 36 0 RTK_11k >
/proc/wlan0/rm_neighbor_report
```

The bss info parameter is 4 octets in length and contains the subfields as shown in next section defined as dot11k_neighbor_report_bssinfo union.
The ssid parameter can be omitted. If the ssid parameter is omitted, it means this neighbor belongs to the same ESS as the AP.

- To del a neighbor report from AP side

#echo del [mac address] > /proc/wlan0/rm_neighbor_report
For example:

```
#echo del 00e04c000001 > /proc/wlan0/rm_neighbor_report
```

- To del all neighbor reports from AP side

#echo delall > /proc/wlan0/rm_neighbor_report

- Show neighbor reports

```
# cat /proc/wlan0/rm_neighbor_report
  -- NEIGHBOR REPORT info --
    [1]BSSID: 00e04c000001
       bss info:0x00E7
       Operating Class:115
       Channel:36
       phy type:0
       SSID:
    [2]BSSID: 00e04c000002
       bss info:0x00E7
       Operating Class:115
       Channel:44
       phy type:0
       SSID:
    [3]BSSID: 00e04c000003
       bss info:0x00E7
       Operating Class:125
       Channel:149
       phy type:0
       SSID:RTK_11k
```

Above four commands are only available on AP side. If the MIB rm_neighbor_report is “1”, client can request neighbor reports by neighbor report request frame.

5. IOCTL

Following are the IOCTL commands list for upper layer application to issue 802.11k measurement request to and retrieve report from other STA. These ioctl commands are available only if the MIB rm_activated is 1

- **0x8BD0 -** Issue link measurement request to other STA

  Input: 6 bytes of mac address of the STA to be requested

  Output: None

  Comment: This ioctl only available when MIB rm_link_measure is 1

- **0x8BD1 –** Get the link measurement report from a STA in response to a link measurement request which is issued by ioctl command 0x8BD0

  Input: 6 bytes of mac address of the STA that transmit the report.

  Output: 1 byte of measurement result (type of MEASUREMENT_RESULT) followed by a dot11k_link_measurement structure if the result is MEASUREMENT_SUCCEED.

  Comment: This ioctl only available when MIB rm_link_measure is 1. The possible values of result byte are MEASUREMENT_UNKNOWN, MEASUREMENT_PROCESSING, and MEASUREMENT_SUCCEED.

  The result MEASUREMENT_PROCESSING means the measurement process is still on-going; while the result MEASUREMENT_SUCCEED means the link measurement process is completed and the link measurement result is in dot11k_link_measurement structure follow the result byte. After the result is retrieved from upper layer, the result byte would change from EASUREMENT_SUCCEED to MEASUREMENT_UNKNOWN.

- **0x8BD2 -** Issue beacon measurement request to other STA

  Input: 6 bytes of mac address of the STA to be requested followed by a dot11k_beacon_measurement_req structure.

  Output: None

  Comment: This ioctl only available when MIB rm_beacon_passive is 1 OR rm_beacon_active is 1. If previous beacon measurement to that STA is still on-going, then the request would not succeed.

- **0x8BD3 -** Get the beacon measurement report from a STA in response to a beacon measurement request which is issued by ioctl command 0x8BD2.

  Input: 6 bytes of mac address of the STA that transmit the report.

  Output: 1 byte of measurement result (type of MEASUREMENT_RESULT) followed by 1 byte of report number and an array of dot11k_beacon_measurement_report structure if the result is MEASUREMENT_SUCCEED.

Comment: This ioctl only available when MIB rm_beacon_passive is 1 OR rm_beacon_active is 1. The possible values of result byte are MEASUREMENT_UNKNOWN, MEASUREMENT_PROCESSING, MEASUREMENT_SUCCEED, MEASUREMENT_INCAPABLE, and MEASUREMENT_REFUSED.

The result MEASUREMENT_PROCESSING means the measurement process is still on-going; while the result MEASUREMENT_SUCCEED means the beacon measurement process is completed and the result is in dot11k_beacon_measurement_report structure follow the result byte. After the result is retrieved from upper layer, the result byte would change from EASUREMENT_SUCCEED to MEASUREMENT_UNKNOWN. The MEASUREMENT_REFUSED means measurement may not currently be executed by the STA; while the MEASUREMENT_INCAPABLE means the STA do not support the beacon measurement or some of the parameter in dot11k_beacon_measurement_req is not supported. Please refer 802.11k spec for more details.

- **0x8BD4 -** Issue neighbor report request to associated AP

Input: None or SSID which identify the ESS that the APs belong to this ESS should be reported.

Output: None

Comment: This ioctl only available when MIB rm_neighbor_report is 1 and the requesting STA is a client which is associated to an AP. If the SSID parameter is omitted, the neighbor report response shall contain information concerning neighbor APs that belong to the same ESS as the requesting STA. If SSID contains the string “ANY”, it means the wildcard SSID is used and the neighbor report response shall contain information concerning all neighbor APs.

- **0x8BD5 -** Get neighbor report response from associated AP in response to a neighbor report request which is issued by ioctl command 0x8BD4.

Input: None.

Output: 1 byte of measurement result (type of MEASUREMENT_RESULT) followed by 1 byte of report number and an array of dot11k_neighbor_report structure if the result is MEASUREMENT_SUCCEED.

Comment: This ioctl only available when MIB rm_neighbor_report is 1 and the requesting STA is a client which is associated to an AP. The possible values of result byte are MEASUREMENT_UNKNOWN, MEASUREMENT_PROCESSING, MEASUREMENT_SUCCEED.

The result MEASUREMENT_PROCESSING means the measurement process is still on-going; while the result MEASUREMENT_SUCCEED means the

neighbor report response is received and the result is in the array of dot11k_neighbor_report structure. After the result is retrieved from upper layer, the result byte would change from MEASUREMENT_SUCCEED to MEASUREMENT_UNKNOWN.

```
typedef enum {
	MEASUREMENT_UNKNOWN = 0,
	MEASUREMENT_PROCESSING = 1,
	MEASUREMENT_SUCCEED = 2,
	MEASUREMENT_INCAPABLE = 3,
	MEASUREMENT_REFUSED = 4,
}MEASUREMENT_RESULT;

struct dot11k_link_measurement
{	unsigned char tpc_tx_power;
	unsigned char tpc_link_margin;
	unsigned char recv_antenna_id;
	unsigned char xmit_antenna_id;
	unsigned char RCPI;
	unsigned char RSNI;
};

struct dot11k_beacon_measurement_req
{	unsigned char op_class;
	unsigned char channel;
	unsigned short random_interval;
	unsigned short measure_duration;
	unsigned char mode;
	unsigned char bssid[MACADDRLEN];
	char ssid[MAX_SSID_LEN + 1];
	unsigned char report_detail;
	unsigned char request_ie_len;
	unsigned char request_ie[MAX_REQUEST_IE_LEN];
	struct dot11k_ap_channel_report
		ap_channel_report[MAX_AP_CHANNEL_REPORT];
};
```

```
struct dot11k_ap_channel_report
{
    unsigned char len;
    unsigned char op_class;
    unsigned char channel[MAX_AP_CHANNEL_NUM];
};
```

- **op_class**: the operating class that indicates the channel set for which the measurement request applies. The supported operating class value are shown in Table 2.
- **channel**: indicates the channel number for which the measurement request applies. The value of 0 indicates a request to make iterative measurements for all supported channels in op_class. The value 255 indicates a request to make iterative measurements for all supported channels in ap_channel_report field.
- **random_interval**: specifies the upper bound of the random delay prior to making the measuremen in unit of TUs.
- **measure_duration**: the duration of requested measurement in unit of TUs.
- **mode**: indicates the beacon measurement mode.

Mode	**Value**
Passive	0
Active	1
Table	2

- **bssid**: indicates the BSSID of BSSs for which a beacon report is requested. If this field is all zero or contains the wildcard BSSID, then all BSSs on the channel would be reported.
- **ssid**: indicateds the ESSs for which a beacon report is request. If this field is all zero, the default “wildcard SSID” is used and all possible SSIDs would be reported. The max ssid length is 32 bytes.
- **report_detail**: indicates the level of detail the requested STA should report. The values are shown as follow.

Level of detail requested	**Reported Detail**
No fixed-length fields of elements	0
All fixed-length fields and any request element in request_ie field	1
All fixed-length fields and elements	2

- **request_ie_len**: the length of request_ie field.
- **request_ie**: an array of request element IDs that are requested to be reported. This field is only valid when the report_detail field is 1. The max number of request_ie is 16.
- **ap_channel_report**: an array of dot11k_ap_channel_report structure that is use to make iterative measurement when channel number is 255. Currently up to 4 AP channel reports can be specified. The len field in dot11k_ap_channel_report is the total valid byte number includes op_class and channel fields. The len 0 means the AP channel report is empty.

```
struct dot11k_beacon_measurement_report
{
    unsigned char op_class;
    unsigned char channel;
    unsigned int    measure_time_hi;
    unsigned int    measure_time_lo;
    unsigned short measure_duration;
    unsigned char frame_info;
    unsigned char RCPI;
    unsigned char RSNI;
    unsigned char bssid[MACADDRLEN];
    unsigned char antenna_id;
    unsigned int    parent_tsf;
    unsigned short subelements_len;
    unsigned char subelements[MAX_BEACON_SUBLEMENT_LEN];
};
```

- **measure_time_hi**: the higher 4 bytes of the measuring STA's TSF at the time the measurement starts.
- **measure_time_lo**: the lower 4 bytes of the measuring STA's TSF at the time the measurement starts.
- **measure_duration**: the accrual duration the beacon report was measured in unit of TUs.
- **frame_info**: the value 0 indicates the report is measured by Beacon or Probe response frame.
- **RCPI**: the received channel power indicator
- **RSNI**: the received signal to noise indication
- **antenna_id**: identify number for the antenna used for this measurement
- **parent_tsf**: the lower 4 bytes of the measuring STA's TSF timer upon receiving the reported beacon or probe response frame.
- **subelements_len**: the length of the subelements field.
- **subelements**: contains the reported frame body element. The existence of this element and the detail level of the frame body depend on the value of report_detail field in the dot11k_beacon_measurement_req structure. The max length of this field is 226 bytes.

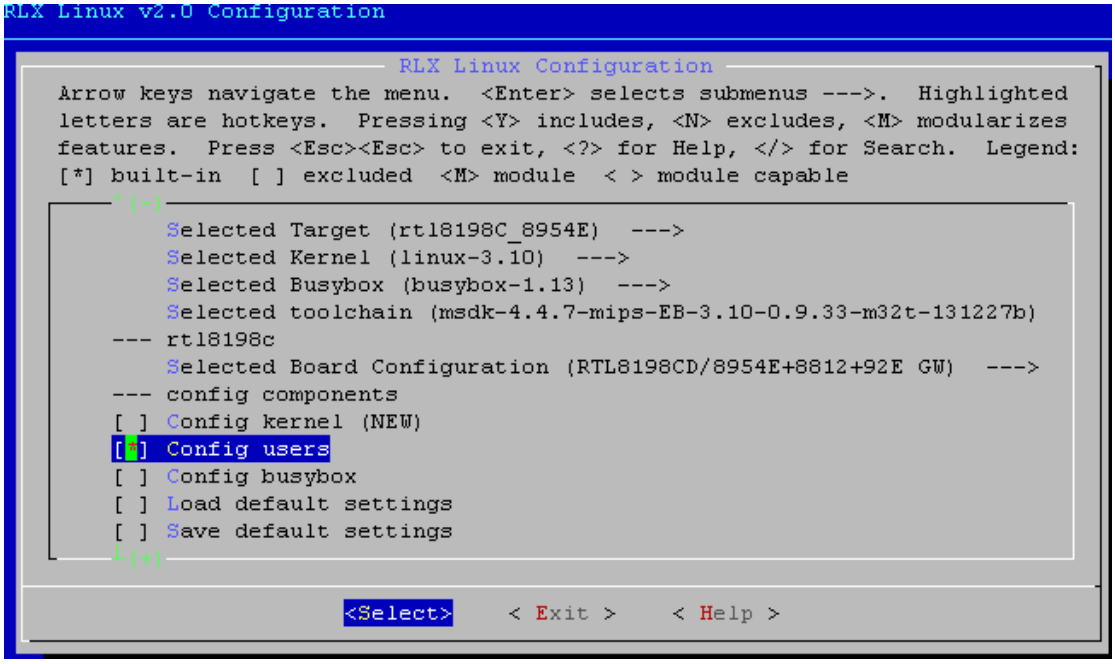
```
struct dot11k_neighbor_report
{
	unsigned char bssid[MACADDRLEN];
	union dot11k_neighbor_report_bssinfo bssinfo;
	unsigned char op_class;
	unsigned char channel;
	unsigned char phytype;
};

union dot11k_neighbor_report_bssinfo {
	unsigned int value;
	struct {
#ifdef _BIG_ENDIAN_
		unsigned int reserved:20;
		unsigned int high_tp:1;
		unsigned int mde:1;
		unsigned int cap_im_ba:1;
```

```
            unsigned int cap_delay_ba:1;
            unsigned int cap_rm:1;
            unsigned int cap_apsd:1;
            unsigned int cap_qos:1;
            unsigned int cap_spectrum:1;
            unsigned int key_scope:1;
            unsigned int security:1;
            unsigned int ap_reachability:2;
#else
            unsigned int ap_reachability:2;
            unsigned int security:1;
            unsigned int key_scope:1;
            unsigned int cap_spectrum:1;
            unsigned int cap_qos:1;
            unsigned int cap_apsd:1;
            unsigned int cap_rm:1;
            unsigned int cap_delay_ba:1;
            unsigned int cap_im_ba:1;
            unsigned int mde:1;
            unsigned int high_tp:1;
            unsigned int reserved:20;
#endif
      } field;
};
```

6. Use test_11k sample code to test 802.11k feature

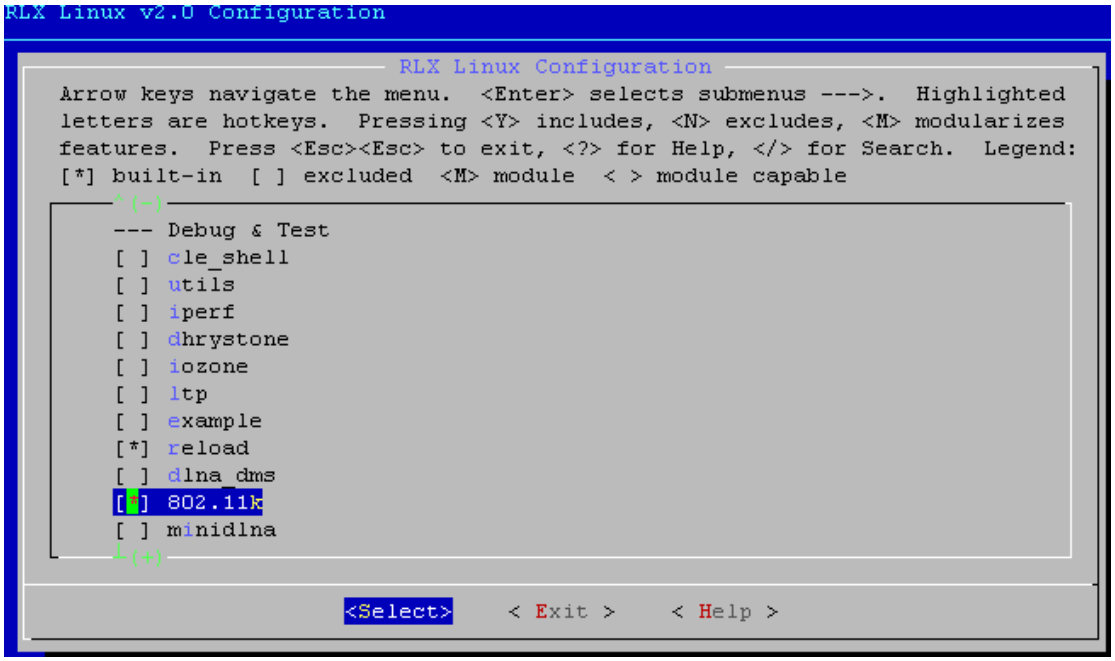
To enable test_11k application, select the “802.11k “ in menuconfig, as following demonstrated.

Select “Config users”



, and then exist.

Find and select option “802.11k” in --- Debug & Test section



Compile SDK to generate new image.

The test_11k sample code is located at /released SDK patch/users/dot11k/test_11k.c

- **Show test_11k command list**

  Type test_11k would show all commands and the description of parameters

```
# test_11k
test_11k [-c command] [-i interface_name] [-a mac_addr]

Available commands:
link    : link measurement
beacon  : beacon measurement: [-o operating_class] [-n channel_number] [-m mode] [-b BSSID] [-s SSID] [-r report_detail]
        OPTION
          -o
             set operating class
               Operating classes in United States
                 1: 5g channel 36, 40, 44, 48
                 2: 5g channel 52, 56, 60, 64
                 3: 5g channel 149, 153, 157, 161
                 4: 5g channel 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
                 5: 5g channel 149, 153, 157, 161, 165
                 12: 2g channel 1~11
               Operating classes in Europe
                 1: 5g channel 36, 40, 44, 48
                 2: 5g channel 52, 56, 60, 64
                 3: 5g channel 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
                 4: 2g channel 1~13,
                 17: 5g channel 149, 153, 157, 161, 165, 169
               Operating classes in Japan
                 1: 5g channel 36, 40, 44, 48
                 30: 2g channel 1~13
                 32: 5g channel 52, 56, 60, 64
                 33: 5g channel 52, 56, 60, 64
                 34: 5g channel 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
                 35: 5g channel 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
               Global Operating classes
                 81: 2g channel 1~13
                 115(default): 5g channel 36, 40, 44, 48
                 118: 5g channel 52, 56, 60, 64
                 121: 5g channel 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
                 124: 5g channel 149, 153, 157, 161,
                 125: 5g channel 149, 153, 157, 161, 165, 169
          -n
             set beacon measurement channel,
             0(default): for iterative measuring all supported channel in the operating class
              in the operating class
             255: for iterative measuring all supported channel liseted in the AP Channel Report
             other: the channel number for which the measuremnet request applies
          -m
             set beacon measurement mode,
             0(default): passive mode,
             1: active mode
             2: table mode
          -b
             set BSSID for whitch a beacon report is requested,
             the default is wildcard BSSID for all BSSs
          -s
             set SSID for whitch a beacon report is requested,
             the default is wildcard SSID that represent all possible SSID
          -r
             set report detail,
             0: no-fixed len field and element,
             1: all fixed len field and elements in Request ie,
             2(default): all fixed len field and elements
neighbor        : get neighbor report [-s SSID]
```

- **Issue link measurement request to and get the report from a STA**

```
#
# test_11k -c link -i wlan0 -a 00e04c66d252
tpc_tx_power: 12
tpc_link_margin: 0
recv_antenna_id: 0
xmit_antenna_id: 0
RCPI: 0xA0
RSNI: 0xFF
#
#
```

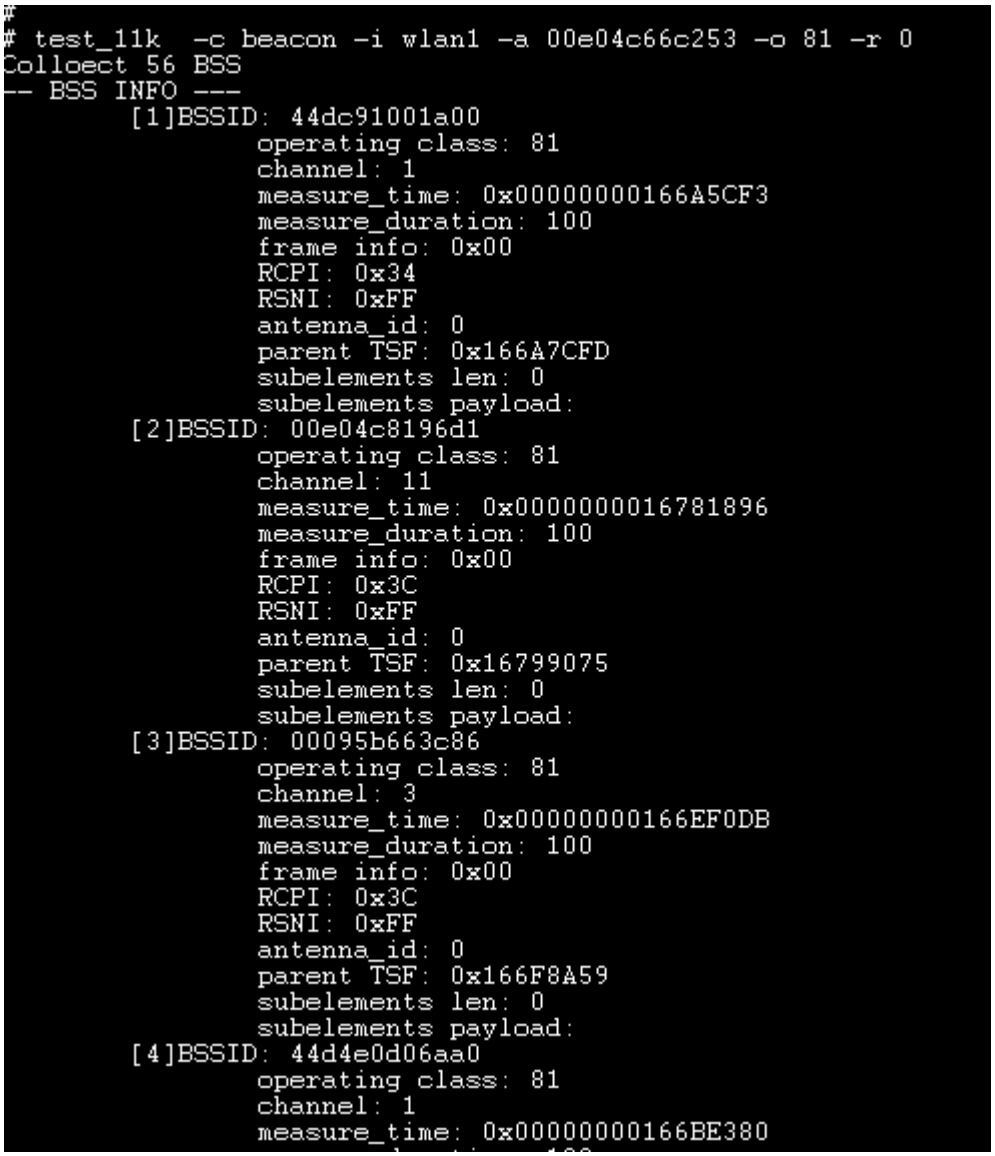
- **Issue passive beacon measurement request to and get the report from iphone with channel 153 and report_detail 1**

```
# test_11k -c beacon -i wlan0 -a f4f15a882aec -o 124 -n 153 -r 1
Colloect 5 BSS
-- BSS INFO ---
        [1]BSSID: 00e081975b37
                operating class: 124
                channel: 151
                measure_time: 0x00000000003E475D
                measure_duration: 100
                frame info: 0x00
                RCPI: 0xBE
                RSNI: 0x18
                antenna_id: 0
                parent TSF: 0x34312D74
                subelements len: 35
                subelements payload:
                 01 21 45 E0 CF 87 6C 01 00 00 64 00 01 00 00 09
                 30 78 53 52 41 50 2D 35 47 01 08 8C 12 98 24 B0
                 48 60 6C
        [2]BSSID: 10bf48d8c96c
                operating class: 124
                channel: 151
                measure_time: 0x00000000003E475D
                measure_duration: 100
                frame info: 0x00
                RCPI: 0xB3
                RSNI: 0x10
                antenna_id: 0
                parent TSF: 0x00000000
                subelements len: 36
                subelements payload:
                 01 22 43 F1 FE F0 51 00 00 00 64 00 01 00 00 0A
                 41 41 5F 41 53 55 53 5F 35 47 01 08 8C 12 98 24
                 B0 48 60 6C
        [3]BSSID: 00e04c000025
                operating class: 124
                channel: 151
                measure_time: 0x00000000003E475D
                measure_duration: 100
                frame info: 0x00
                RCPI: 0xB3
                RSNI: 0x10
                antenna_id: 0
                parent TSF: 0x0586EC00
                subelements len: 44
                subelements payload:
                 01 2A 41 80 5D 08 82 01 00 00 64 00 11 00 00 0A
                 52 65 61 6C 4B 75 6E 67 46 75 01 08 8C 12 98 24
                 B0 48 60 6C 07 06 55 53 20 01 0B 1E
        [4]BSSID: 00e04c58881a
                operating class: 124
                channel: 151
                measure_time: 0x00000000003E475D
```

- **Issue active beacon measurement request to and get the report from iphone with channel 36 and report_detail 2 and the report matches a specific bssid**

```
# test_11k -c beacon -i wlan0 -a f4f15a882aec -m 1 -n 36 -b 00e04c897711
Colloect 1 BSS
-- BSS INFO ---
        [1]BSSID: 00e04c897711
                operating class: 115
                channel: 36
                measure_time: 0x00000000005F3793
                measure_duration: 100
                frame info: 0x00
                RCPI: 0xAC
                RSNI: 0x10
                antenna_id: 0
                parent TSF: 0x00000000
                subelements len: 207
                subelements payload:
                 01 E7 24 D0 2B C3 B3 00 00 00 64 00 01 00 00 07
                 52 54 57 35 47 41 43 01 08 8C 12 98 24 B0 48 60
                 6C 05 04 00 01 00 00 2D 1A 2C 10 1B FF FF 00 00
                 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                 00 00 00 3D 16 24 00 00 00 00 00 00 00 00 00 00
                 00 00 00 00 00 00 00 00 00 00 00 DD 18 00 50 F2
                 02 01 01 00 00 03 A4 00 00 27 A4 00 00 42 43 5E
                 00 62 32 2F 00 DD 1E 00 90 4C 33 2C 10 1B FF FF
                 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                 00 00 00 00 00 DD 1A 00 90 4C 34 24 00 00 00 00
                 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                 00 DD 07 00 E0 4C 02 02 60 02 BF 0C 20 40 C1 03
                 FA FF 2C 01 FA FF 2C 01 C0 05 00 00 00 F0 FF
#
#
#
```

- **Issue iterative passive beacon measurement request to and get the report from a STA to all channel in op_class 81 (2g: 1~13) with report_detail 0**



- **Issue iterative active beacon measurement request to and get the report from a STA to the channels 36, 48 that are contained in the AP channel report with report_detail 1**

Add the channel report to AP first

```
#
# echo add 115 36 48 > /proc/wlan0/rm_ap_channel_report
#
```

Use test_11k to issue beacon measurement request. The test_11k would use the AP channel reports in the /proc/wlan0/rm_ap_channel_report.

```
# test_11k -c beacon -i wlan0 -a 00e04c8881aa -m 1 -n 255 -r 1
Colloect 14 BSS
-- BSS INFO ---
        [1]BSSID: 00e04caa0917
                operating class: 115
                channel: 36
                measure_time: 0x000000008521369A
                measure_duration: 100
                frame info: 0x00
                RCPI: 0x60
                RSNI: 0xFF
                antenna_id: 0
                parent TSF: 0x85213E7F
                subelements len: 70
                subelements payload:
                 01 44 F0 3C 90 06 02 00 00 00 64 00 11 05 00 06
                 48 53 32 5F 35 47 30 14 01 00 00 0F AC 04 01 00
                 00 0F AC 04 01 00 00 0F AC 02 00 00 DD 18 00 50
                 F2 02 01 01 00 00 03 A4 00 00 27 A4 00 00 42 43
                 5E 00 62 32 2F 00
        [2]BSSID: 100d7f877910
                operating class: 115
                channel: 36
                measure_time: 0x000000008521369A
                measure_duration: 100
                frame info: 0x00
                RCPI: 0x44
                RSNI: 0xFF
                antenna_id: 0
                parent TSF: 0x852140A9
                subelements len: 169
                subelements payload:
                 01 A7 33 B7 30 B2 40 00 00 00 64 00 11 00 00 05
                 36 33 30 30 35 30 14 01 00 00 0F AC 04 01 00 00
                 0F AC 04 01 00 00 0F AC 02 0C 00 DD 7C 00 50 F2
                 04 10 4A 00 01 10 10 44 00 01 02 10 3B 00 01 03
                 10 47 00 10 48 A3 91 EF 08 60 2B 46 C1 E6 27 29
                 84 C7 07 4A 10 21 00 0D 4E 45 54 47 45 41 52 2C
                 20 49 6E 63 2E 10 23 00 05 52 36 33 30 30 10 24
                 00 05 52 36 33 30 30 10 42 00 04 34 35 33 36 10
                 54 00 08 00 06 00 50 F2 04 00 01 10 11 00 05 52
                 36 33 30 30 10 08 00 02 20 08 10 3C 00 01 03 10
                 49 00 06 00 37 2A 00 01 20
        [3]BSSID: 00904c131dc7
                operating class: 115
                channel: 36
                measure_time: 0x000000008521369A
                measure_duration: 100
```

- **Issue table beacon measurement request to and get the report from a STA with report_detail 0**

```
# test_11k -c beacon -i wlan0 -a 00e04c8881aa -m 2 -r 0
Colloect 20 BSS
-- BSS INFO ---
        [1]BSSID: 00e04c897711
                operating class: 115
                channel: 36
                measure_time: 0x0000000010BF5BFF
                measure_duration: 100
                frame info: 0x00
                RCPI: 0x4A
                RSNI: 0xFF
                antenna_id: 0
                parent TSF: 0x10BF8591
                subelements len: 0
                subelements payload:
        [2]BSSID: 00904c131dc7
                operating class: 115
                channel: 36
                measure_time: 0x0000000008521369A
                measure_duration: 100
                frame info: 0x00
                RCPI: 0x24
                RSNI: 0xFF
                antenna_id: 0
                parent TSF: 0x85215D54
                subelements len: 0
                subelements payload:
        [3]BSSID: 1cb17fe1d5ff
                operating class: 115
                channel: 36
                measure_time: 0x0000000013C8460B
                measure_duration: 100
                frame info: 0x00
                RCPI: 0x18
                RSNI: 0xFF
                antenna_id: 0
                parent TSF: 0x13C8598D
                subelements len: 0
                subelements payload:
        [4]BSSID: 00e04c975b9d
                operating class: 115
                channel: 36
                measure_time: 0x0000000008521369A
                measure_duration: 100
                frame info: 0x00
                RCPI: 0x44
                RSNI: 0xFF
                antenna_id: 0
                parent TSF: 0x85219BA6
                subelements len: 0
                subelements payload:
        [5]BSSID: 00e04c474847
```

- **Issue neighbor report request to and get the neighbor report response from the associated AP**

Prepare two DUTs. One is using AP mode, and the other is using client mode.

Connect the client to the AP.

On the AP side, add neighbor reports as following

```
#
# echo add 00e04c000001 0xE7 115 44 0 > /proc/wlan0/rm_neighbor_report
# echo add 00e04c000002 0xE7 124 149 0 > /proc/wlan0/rm_neighbor_report
#
#
```

On the client side, use test_11k to issue neighbor report request and get neighbor report response from AP.

```
# test_11k -c neighbor -i wlan0
NEIGHBOR AP NUM: 2
-- NEIGHBOR AP INFO ---
        [1]BSSID: 00e04c000001
                bss info: 0x00E7
                operating class: 115
                channel: 44
                phytype: 0
        [2]BSSID: 00e04c000002
                bss info: 0x00E7
                operating class: 124
                channel: 149
                phytype: 0
#
#
```