

How to use Smart Connection library on MT8516

- MTK provide libsmtcn.so to implement Smart Connection.
- libsmtcn.so support MC and BC packet method.
- Usaging:
 - Link libsmtcn.so librt.so(If setting timeout)
 - Include smt_api.h
 - API function is introduced in smt_api.h
 - You could refer to src/apps/aud-base/main/wifi_setting_proc.c



Smart Connection Usage Guide in smt_api.h

```
/**
 * @addtogroup Smart_Connection
 * @{}
 * This section introduces MediaTek Smart Connection APIs including details on how to use the APIs, enums, structures and functions.
 *
 * @section Smart_Connection_Usage_Chapter How to use these APIs
 *
 * - Connecting to the target AP with Smart Connection\n
 * - Step 1. Define a #wifi_smart_connection_callback_t callback function to process a Smart Connection event.
 * - Step 2. Call #wifi_smart_connection_init() to apply for required resources.
 * Note: The predefined callback, such as, the smtcn_evt_handler(), should be passed to the #wifi_smart_connection_init().
 * - Step 3. Call #wifi_smart_connection_start() to start the Smart Connection.
 * - Step 4. Call #wifi_smart_connection_get_result() to get the SSID and the password, when receiving #WIFI_SMART_CONNECTION_EV...
 * - Step 5. Connect to the target AP.
 * - Step 6: Call #wifi_smart_connection_stop() to exit the Smart Connection. Or call #wifi_smart_connection_deinit() if the con...
```

```

* - sample code:
* @code
* void smtcn_evt_handler(wifi_smart_connection_event_t event, void *data)
* {
*     uint8_t passwd[WIFI_LENGTH_PASSPHRASE + 1] = {0};
*     uint8_t ssid[WIFI_MAX_LENGTH_OF_SSID + 1] = {0};
*     uint8_t ssid_len = 0;
*     uint8_t passwd_len = 0;
*
*     switch (event)
*     {
*         case WIFI_SMART_CONNECTION_EVENT_CHANNEL_LOCKED:
*             //Report the locked channel.
*             break;
*         case WIFI_SMART_CONNECTION_EVENT_TIMEOUT:
*             //Report the timeout to the upper layers.
*             break;
*         case WIFI_SMART_CONNECTION_EVENT_INFO_COLLECTED:
*             //Save the received information and connect to the target AP, and then call #wifi_smart_connection_deinit() to release the resources.
*             wifi_smart_connection_get_result(ssid, &ssid_len, passwd, &passwd_len, NULL, NULL);
*             mtk_scan_connect_ap(ssid, passwd);
*             wifi_smart_connection_deinit();
*             break;
*     }
* }
*
* int32_t mtk_smart_connect(void)
* {
*     if(wifi_smart_connection_init(NULL, 0, smtcn_evt_handler) < 0){
*         return -1;
*     }
*
*     wifi_smart_connection_start(0);
*     return 0;
* }
*
* int32_t mtk_smart_stop(void)
* {
*     wifi_smart_connection_stop();
*     return 0;
* }

```



everyday genius