



# Wi-Fi location demo reference design

Version: 1.0

Release date: 15 November 2016

---

© 2015 - 2016 MediaTek Inc.

This document contains information that is proprietary to MediaTek Inc. ("MediaTek") and/or its licensor(s). MediaTek cannot grant you permission for any material that is owned by third parties. You may only use or reproduce this document if you have agreed to and been bound by the applicable license agreement with MediaTek ("License Agreement") and been granted explicit permission within the License Agreement ("Permitted User"). If you are not a Permitted User, please cease any access or use of this document immediately. Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited. THIS DOCUMENT IS PROVIDED ON AN "AS-IS" BASIS ONLY. MEDIATEK EXPRESSLY DISCLAIMS ANY AND ALL WARRANTIES OF ANY KIND AND SHALL IN NO EVENT BE LIABLE FOR ANY CLAIMS RELATING TO OR ARISING OUT OF THIS DOCUMENT OR ANY USE OR INABILITY TO USE THEREOF. Specifications contained herein are subject to change without notice.

## Document Revision History

---

Revision	Date	Description
1.0	15 November 2016	Porting Wi-Fi location demo to SDK4.0.0.

## Table of contents

---

<b>1.</b>	<b>Introduction.....</b>	<b>1</b>
1.1.	Objective.....	1
<b>2.</b>	<b>Hardware.....</b>	<b>2</b>
2.1.	MT7687 HDK.....	2
<b>3.</b>	<b>Software.....</b>	<b>3</b>
3.1.	Wi-Fi positioning principle based on cloud service .....	3
3.2.	Cloud service .....	3
3.3.	Location implement.....	4
<b>4.</b>	<b>Test.....</b>	<b>9</b>
4.1.	Software merge .....	9
4.2.	Device .....	9
4.3.	PC.....	9

## **1. Introduction**

---

### **1.1. Objective**

The purpose of this document is to describe the design of Wi-Fi location demo.

In this document, you can create your own Wi-Fi location application by using MT7687 and cloud service. This design is also can be used in production sale's area sales statistics and other scene that need a location information.

In this document, SW coding should be introduced for reader. And reader could build the Wi-Fi location application in a short time.

## 2. Hardware

### 2.1. MT7687 HDK

MCU: Mediatek MT7687

Wi-Fi: embedded in Mediatek MT7687 (Wi-Fi SOC)



**Figure 1. MT7687 HDK**

### 3. Software

#### 3.1. Wi-Fi positioning principle based on cloud service

Collect information about the Wi-Fi hotspots around the device, upload these information to cloud service. Cloud service return the result after location calculation, device parse the result and get location information.



Figure 2. Wi-Fi positioning principle

#### 3.2. Cloud service

In this design, cloud service provide by haoservice, detail information as below.

Service interface link: <http://api.haoservice.com/api/LocationByWifiData>

Result support format: JSON/XML

Service request method: GET/POST

Service request example: `http://api.haoservice.com/api/LocationByWifiData?requestdata=[{"mac_address": "f0:b4:29:8b:e2:ee", "singal_strength":40, "age":0}, {"mac_address": "40:e3:d6:8f:69:24", "singal_strength":53, "age":0}]&type=2&key=your apply APPKEY`

Table 1. Service request parameters

Name	Type	Required	Note
key	String	Yes	APP KEY
requestdata	List	Yes	Wi-Fi hotspots data object
mac_address	String	Yes	Wi-Fi hotspots mac address
age	int	No	Wi-Fi hotspots detect timestamp
singal_strength	int	No	Wi-Fi hotspots signal strength, unit: dbm
type	int	No	Return coordinate type, 0: google coordinate, 1: baidu coordinate, 2: GPS coordinate, default type is 0.

**Table 2. Service return result define**

Name	Type
address	Object
region	String
county	String
street	String
street_number	String
city	String
country	String
addressDescription	String
longitude	Double
latitude	Double
accuracy	Int

Service return result example (JSON format):

```
{
  "location": {
    "address": {
      "region": "广东省",
      "county": "南山区",
      "street": "科技中二路",
      "street_number": "嘉余坊",
      "city": "深圳市",
      "country": "中国"
    },
    "addressDescription": "广东省深圳市南山区科技中二路南头街道嘉余坊,深圳软件园东区北 131 米,科技中二路北 1 米",
    "longitude": 113.9338781414848,
    "latitude": 22.5494486738287,
    "accuracy": "131 米"
  },
  "access_token": null,
  "ErrCode": "0"
}
```

### 3.3. Location implement

#### 3.3.1. Collect information about the Wi-Fi hotspots

Collect information about the Wi-Fi hotspots around the device, save these information in an array and sort by signal strength (descending order).

```
static int32_t scan_ap(void)
```

```

{
    scan_result_number = 0;
    os_memset(&g_tmp_scan_result_list, 0, MAX_SCAN_RESULT * sizeof(wifi_scan_list_item_t));

    wifi_connection_register_event_notifier(WIFI_EVENT_IOT_REPORT_BEACON_PROBE_RESPONSE,
        (wifi_event_handler) scan_ap_prb_rsp_handler);

    wifi_connection_register_event_notifier(WIFI_EVENT_IOT_SCAN_COMPLETE,
        (wifi_event_handler) scan_ap_done);
    wifi_connection_start_scan(NULL, 0, NULL, 0, 0);

    scan_done = 0;

    return 0;
}

```

```

static int scan_ap_prb_rsp_handler(wifi_event_t event_id, unsigned char *payload, unsigned int len)
{
    if (event_id != WIFI_EVENT_IOT_REPORT_BEACON_PROBE_RESPONSE) {
        return 1;
    }

    if (len != 0) {
        wifi_scan_list_item_t ap_data;
        os_memset(&ap_data, 0, sizeof(wifi_scan_list_item_t));
        if (wifi_connection_parse_beacon(payload, len, &ap_data) >= 0)
            if (ap_data.ssid_length > 0) {
                add_scan_result(&ap_data);
            }
    }
    return 1;
}

```

### 3.3.2. Build cloud service request

According to the cloud service request rule, build a cloud service request URL based on the information about Wi-Fi hotspots.

```

static void build_location_url(void)
{
    char *url_template =
"http://api.haoservice.com:80/api/LocationByWifiData?key=168de53dc3ee4ea8816ad9322a530935&type=2&re
questdata=";

    printf("build_location_url: %d , %d\n", sizeof(url), sizeof(char));
    os_memset(url, 0, MAX_URL_LEN);
    url_len = 0;
}

```



```

snprintf(url+url_len, MAX_URL_LEN-url_len, "%s", url_template);
url_len = strlen(url);

if(scan_result_number >0 )
    snprintf(url+url_len, MAX_URL_LEN-url_len,
"{\"mac_address\": \"%2x:%2x:%2x:%2x:%2x:%2x\", \"singal_strength\": %d, \"age\": 0}, \
g_tmp_scan_result_list[0].bssid[0],g_tmp_scan_result_list[0].bssid[1],g_tmp_scan_result_list[0].bssid[2
], \
g_tmp_scan_result_list[0].bssid[3],g_tmp_scan_result_list[0].bssid[4],g_tmp_scan_result_list[0].bssid[5
], \
-g_tmp_scan_result_list[0].rssi);

url_len = strlen(url);

if(scan_result_number >1 )
    snprintf(url+url_len, MAX_URL_LEN-url_len,
\", {\"mac_address\": \"%2x:%2x:%2x:%2x:%2x:%2x\", \"singal_strength\": %d, \"age\": 0}, \
g_tmp_scan_result_list[1].bssid[0],g_tmp_scan_result_list[1].bssid[1],g_tmp_scan_result_list[1].bssid[2
], \
g_tmp_scan_result_list[1].bssid[3],g_tmp_scan_result_list[1].bssid[4],g_tmp_scan_result_list[1].bssid[5
], \
-g_tmp_scan_result_list[1].rssi);

url_len = strlen(url);

if(scan_result_number >2 )
    snprintf(url+url_len, MAX_URL_LEN-url_len,
\", {\"mac_address\": \"%2x:%2x:%2x:%2x:%2x:%2x\", \"singal_strength\": %d, \"age\": 0}, \
g_tmp_scan_result_list[2].bssid[0],g_tmp_scan_result_list[2].bssid[1],g_tmp_scan_result_list[2].bssid[2
], \
g_tmp_scan_result_list[2].bssid[3],g_tmp_scan_result_list[2].bssid[4],g_tmp_scan_result_list[2].bssid[5
], \
-g_tmp_scan_result_list[2].rssi);

url_len = strlen(url);

snprintf(url+url_len, MAX_URL_LEN-url_len, "%s", "]}");

url_len = strlen(url);
}

```

### 3.3.3. Send cloud service request

Send cloud service request URL based on httpclient\_get and get the service result.

```
static void wifi_location_demo(void)
{
    httpclient_t client = {0};
    httpclient_data_t client_data = {0};
    char *buf;

    build_location_url();

    printf("url:%s\r\n",url);
    printf("url_len :%d\r\n",url_len);

    buf = pvPortMalloc(BUF_SIZE);
    if (buf == NULL) {
        printf("wifi_location_demo malloc failed.\r\n");
        return;
    }
    client_data.response_buf = buf;
    client_data.response_buf_len = BUF_SIZE;

    //httpclient_get(&client, url, HTTP_PORT, &client_data); //for SDK3.3.x
    httpclient_get(&client, url, &client_data); //for SDK4.0.x

    printf("wifi_location_dmeo data received: %s\r\n", client_data.response_buf);
    location_json_doit(client_data.response_buf);

    vPortFree(buf);

    printf("wifi_location_demo end.\r\n");
}
```

### 3.3.4. Get the location

Parse the service result based on cJSON and get location information (e.g., longitude, latitude).

```
void location_json_doit(char *text)
{
    char *out;
    cJSON *json,*json_errcode;

    json = cJSON_Parse(text);
    if (!json) {
        printf("Error before: [%s]\n", cJSON_GetErrorPtr());
    } else {
        printf("JSON Parse longitude.\r\n");
    }
}
```

```

json_Errcode=cJSON_GetObjectItem(json,"ErrCode");
if(!json_Errcode) printf("JSON json_Errcode none.\r\n");
else {
    printf("json_Errcode:%d %s\n",json_Errcode->type,json_Errcode->string);
    switch ((json_Errcode->type)&255)
    {
        case cJSON_NULL:      printf("cJSON_NULL.\r\n"); break;
        case cJSON_False:     printf("cJSON_False.\r\n"); break;
        case cJSON_True:      printf("cJSON_True.\r\n"); break;
        case cJSON_Number:    printf("valueint:%d.\r\n",json_Errcode-
>valueint); break;
        case cJSON_String:
printf("valuestring:%s.\r\n",json_Errcode->valuestring); break;
    }

    printf("%s\n",cJSON_Print(json_Errcode));
}

printf("%s\n",cJSON_Print(cJSON_GetObjectItem(json,"location")));
printf("%s\n",cJSON_Print(cJSON_GetObjectItem(json,"access_token")));
printf("%s\n",cJSON_Print(cJSON_GetObjectItem(json,"ErrCode")));

printf("Get Location Longitude, Latitude:
%s",cJSON_Print(cJSON_GetObjectItem(cJSON_GetObjectItem(json,"location"),"longitude")));
printf("
%s\n",cJSON_Print(cJSON_GetObjectItem(cJSON_GetObjectItem(json,"location"),"latitude")));

printf("JSON Parse Result.\r\n");
out = cJSON_Print(json);
cJSON_Delete(json);
printf("%s\n", out);
cJSON_free(out);
}
}

```

## 4. Test

---

### 4.1. Software merge

Copy the demo project to the following path:

Copy http\_client\_get to project\mt7687\_hdk\apps\http\_client\

Copy httpclient to middleware\third\_party\

Redefine WIFI\_SSID & WIFI\_PASSWORD in project\mt7687\_hdk\apps\http\_client\http\_client\_get\src\main.c according your AP setting.

And then run the build command:

```
./build.sh mt7687_hdk http_client_get
```

### 4.2. Device

Download the software BIN into MT7687, for how to download, refer to “LinkIt for RTOS get started”,

Then power on the MT7687 HDK.

### 4.3. PC

PC can get UART log from MT7687 HDK and get the location information.

```
Get Location Longitude, Latitude: 113.9338781414848, 22.5494486738287
JSON Parse Result.
{
  "location": {
    "address": {
      "region": "广东省",
      "county": "南山区",
      "street": "科技中二路",
      "street_number": "嘉余坊",
      "city": "深圳市",
      "country": "中国"
    },
    "addressDescription": "广东省深圳市南山区科技中二路南头街道嘉余坊,深圳软件园东区北131米,科技中二路北1米",
    "longitude": 113.9338781414848,
    "latitude": 22.5494486738287,
    "accuracy": "131米"
  }
}
```

**Figure 3. Location information in UART log**