

IITK-CS630A

ADVANCED OS FOR EMBEDDED SYSTEMS

PROJECT PROPOSAL

Wi-Fi Analyzer

Supervisor:

MR NITIN JHANWAR

Submitted By :

NISHIT MAJITHIA

ROHIT SEHGAL

February 27, 2016



1 Project Idea

The Basic idea of the project is to develop an Wi-Fi Analyzer app for the tizen TV.This App will be capable enough to scan Nearby wifi Access Points, provide the user with the Following information about the Wifi Access Point.

- Wi-Fi signal Strength of the Respective AP.
- The MAC address of Each AP.
- The IP address provided by the AP to the device.

Once the user is connected to a specific AP then,in order to test the proper functionality of that AP,a simple file transfer will be initiated from this connected device to AP.

2 Functional Attributes

2.1 Access Points Listing

1. Initialize local wifi device using `wifi_initialize()`,this allows us to access all device capable wifi functions.This method returns error code corresponding to the error listed in `wifi_error_e` enum present in `wifi.h`
2. Activate a local Wi-Fi device using the `wifi_activate()` function. This function take callback function as the first parameter.In our Sequence diagram we have used `activated_cb` function as the same callback which is called after the activation is completed.The next argument is temporarily set to use the `NULL` value, in case we want no notifications.
3. `wifi_scan()` is used, after returning from `activated_cb` in order to scan AP nearby.This method takes a callback function, here it is `scan_request` which enumerates all scanned AP, traverse through each of them to query about their MAC addresses,ssid and other state specific information.

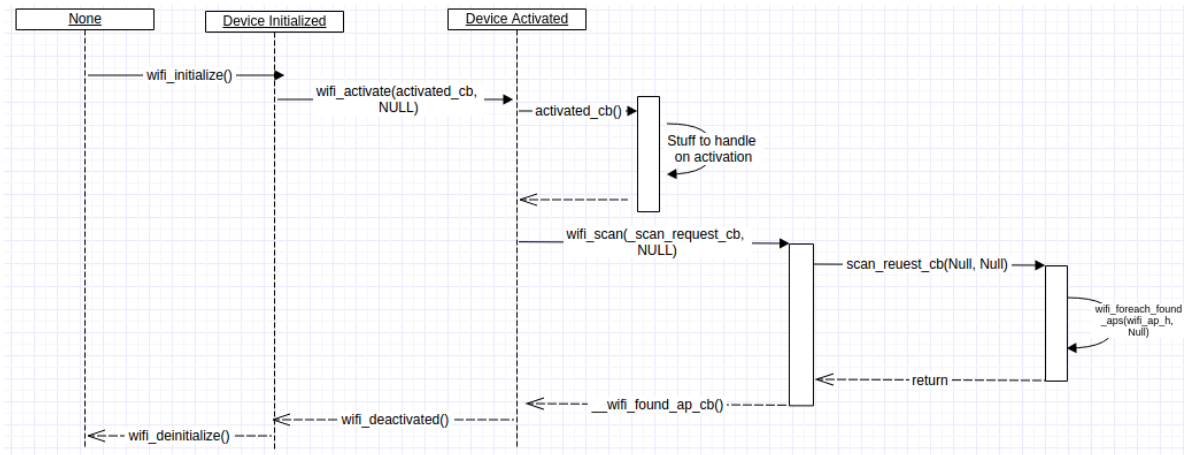


Figure 1: Sequence Diagram showing steps to Initialize and list all APs

2.2 Connection With AP

1. Since `wifi_found_ap_callback` displays the all the activated AP,so as to connect with a specific AP,`wifi_foreach_found_ap` is used with `found_connect_cb` callback as first parameter,that handles the request of user specific AP is found which is still activated(cheked using `wifi_is_activated`)
2. The `wifi_connect()` function called within the `found_connect_cb()` callback invokes the `connected_cb()` function,passed as second argument, which can be used to notify the user of the connection result. `wifi_connect` takes three different arguments,first the ap details,which are passed as handle of type `wifi_ap_h` object,that is being received form `found_connect_cb` as an argument.And third argument is pointer to the user_data,here it is not being used so passed as NULL.

If `WIFI_CONNECTION_STATE_CONNECTED` of the enum object `wifi_connection_state_e` (definition in `wifi.h`) is returned by `wifi_connect`.Then it indicate the success of the connection between the two.

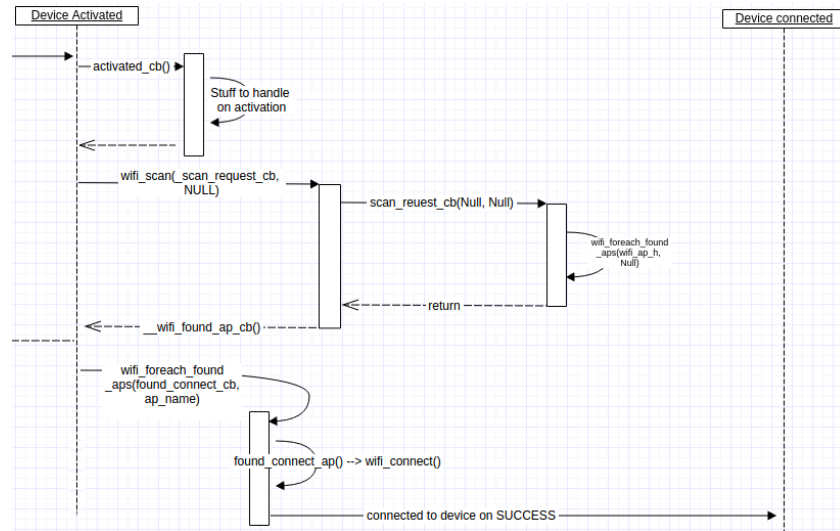


Figure 2: Sequence Diagram showing the process flow of how to connect with a specific AP

3 File Transfer

- Normal P2P socket system call is initiated if both the device are provided MAC addresses. But before a particular device initiate the file transfer, A MD5 Check sum of the file is computed and transfer along with the file.
- At receiving side the file check sum is computed again and checked for the equality of the two. Equality ensures the consistency of the file.

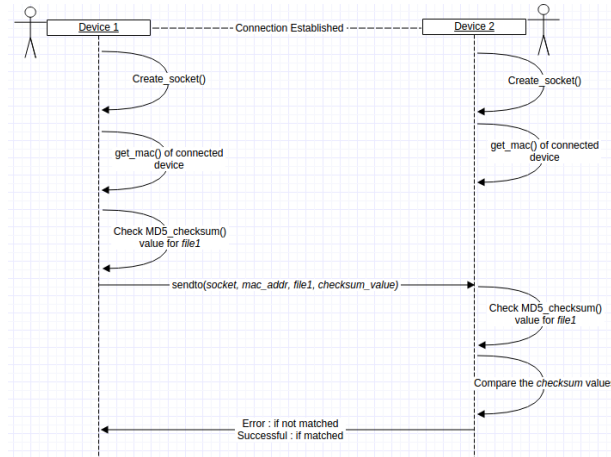


Figure 3: Sequence Diagram For file transfer

4 Scanning Wifi-Direct Stations

- Scanning Wifi direct stations is similar to that of scanning with wifi APIs. The only difference is that api calls are appended by the wifi_direct.

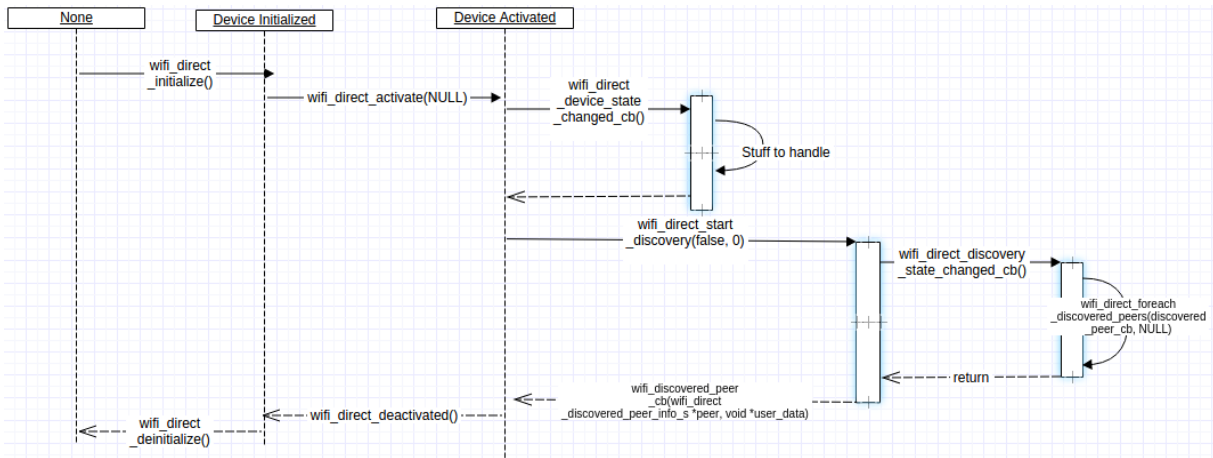


Figure 4: Sequence Diagram for scanning wifi direct stations

5 Connecting with Wifi-Direct STA

- Creating a connection with wifi-Direct station is in way very much similar to connecting a wifi device with wifi AP. The process flow is represented below by the sequence diagram. With the listing of API calls.

12. `int wifi_direct_foreach_connected_peers (wifi_direct_connected_peer_cb callback, void *user_data)` Gets the information of connected peers.

6.1.1 References

- https://developer.tizen.org/dev-guide/2.4/org.tizen.native.mobile.apireference/group__CAPI__NETWORK__WIFI__DIRECT__MODULE.html
- https://developer.tizen.org/dev-guide/2.4/org.tizen.native.mobile.apireference/group__CAPI__NETWORK__WIFI__MODULE.html

6.2 Data Structure Description

- These C APIs calls uses the Inbuilt data Structure defined in `wifi.h`, `wifi_direct.h`. For example all the error numbers that could be return by the calls is enumerated into **wifi_error_e**. E.g `WIFI_ERROR_NOT_SUPPORTED` is returned when the wifi device do not support the called functionality.
- **wifi_device_state_e** and **wifi_connection_state_e** enums store the state properties of the device and connection.
- **wifi_ap_h** handle is used to store all the information related to a specific ap. E.g the call `wifi_connect` takes `wifi_ap_h` object as an argument. So this handle abstract out the ap details.

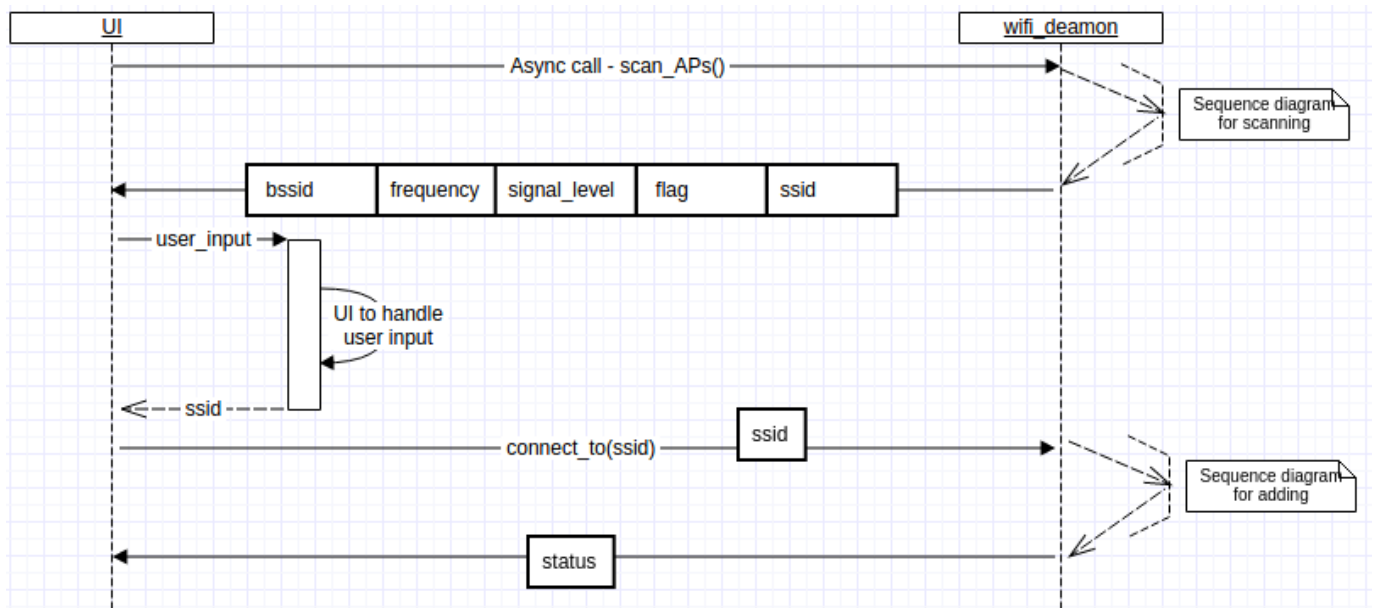


Figure 6: UI and Wifi Daemon Interaction With data Structure Flow between the two

6.2.1 References

<https://developer.tizen.org/dev-guide/2.4/org.tizen.native.mobile.apireference>