Automation Framework and TriRadio Connectivity

Major Project Report

Submitted in fulfillment of the requirements

for the degree of

Master of Technology

in

Electronics & Communication Engineering

(Embedded Systems)

By

Vyom Chanpura

(20mece05)

Electronics & Communication Engineering Department

Institute of Technology

Nirma University

Ahmedabad-382 481

Certificate

This is to certify that the Major Project entitled **"Automation Framework and TriRadio Connectivity"** submitted by **Vyom Chanpura (20MECE05)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in Embedded Systems, Nirma University, Ahmedabad is the record of work carried out by him under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination.The results embodied in this major project, to the best of our knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Date: Place: Ahmedabad

Dr.Dhaval Pujara
Internal Guide

Dr. N.P. Gajjar
Program Coordinator

Dr. Dhaval Pujara
Head of Department,EC

Dr. Rajesh Patel
Director, IT

Acknowledgements

I would like to express my gratitude and sincere thanks to **Dr. N.P.Gajjar**, PG Coordinator of M.Tech Embedded Systems and **Dr. Dhaval Pujara** for guidelines during the review process.

I take this opportunity to express my profound gratitude and deep regards to **Dr. Dhaval Pujara**, guide of my internship project for his exemplary guidance, monitoring and constant encouragement.

I would also like to thank **Mr.Mahesh Nahar**, external guide of my internship project from **NXP,Pune**, for guidance, monitoring and encouragement regarding the project.

- Vyom Chanpura

20MECE05

Abstract

As of now the people having the smartphone / mobile phone will connect to their friends family and also share the photos, videos, files, etc through the WLAN / Bluetooth technologies. This connectivity in our phone has been possible because of the WLAN / Bluetooth SoCs.

Every phone has a separate SoC other than processor SoC. So, before going to the customer we have to test the SoCs properly by running the test cases in it. It will be done in two ways one is manually(in this we write the commands manually for testing Bluetooth WLAN) another is automation (In this the automation script has to be run by selecting appropriate test cases).

In the automation framework , work is to add some feature to track the result of the test-cases of the different executions with there time date called as the Report file comparison implementation.Also this data is being send in the mail with the specific format having a hyperlink with the particular result of the execution

In bluetooth testing , tested on some of the profiles like A2DP , AVRCP , HFP , HID , LEHID etc , also we tested the WLAN profile like STA , MMH , P2P. This two technologies are being tested simultaneously in the COEX testing part.Also we explored the open-thread protocol by doing testing on it.This testing is being done on the Tri-Radio chipset.

Company Profile

NXP Semiconductors is well known for their wireless SoCs in the Automotive field, In most vehicles, the entertainment part is powered by the NXP's wireless SoCs. Also, NXP has solutions for various fields like IoT, Industrial IoT, Home automation, and communication infrastructure markets.

NXP is also a co-inventor in the NFC technology with the Sony and they supply NFC chipsets for secure and powerful communication between any two devices supporting this technology, by this, NXP's product is also being embedded in the mobile phone, debit credit cards for contactless payment. NXP owns approx. 9,500 patents in the semiconductor industry.

NXP chipset has various Product line :

1. **NXP MIFARE contactless** smart cards and proximity cards
2. **NXP LPC** Micro controller
3. **NXP QorIQ** Micro processors
4. **NXP GreenChip**

Contents

Certificate ii

Acknowledgements iii

Abstract iv

Company Profile v

1 Introduction 1

1.1 Domain Of Work 1

1.2 Objective 3

1.3 Problem Statement 3

2 Literature Review 1

2.1 Literature Review in Bluetooth 1

2.1.1 Bluetooth Architecture 1

2.1.2 Bluetooth Profiles 3

2.2 Literature Review In Automation 5

2.2.1 Process of Automation 5

2.2.2 Automation Scripting Language 6

2.2.3 Difference Between Automation testing and Manual Testing . 6

2.2.4 Advantages of using Automation : 7

3 Test Report File Comparison Implementation **1**

3.1 Generation Of Report File 1

3.2 Need of Report File comparison Implementation 2

3.3 Block Diagram of Report File Comparison Implementation 2

4 Cloud-Based Automation (Block diagram) **1**

5 Tri-Radio Chipset Testing **1**

5.1 i.MX Linux 1

5.2 Ubuntu 16 2

5.3 i.MX Android 2

6 Open-Thread Testing **1**

6.1 Introduction Open thread 1

6.2 Role of the Devices 3

6.2.1 Border Router 3

6.2.2 Leader 3

6.2.3 Router(Parent) 4

6.2.4 End Device 4

7 Conclusion and Future Scope **1**

7.1 Conclusion 1

7.2 Future Scope 2

References **3**

List of Figures

2.1 Bluetooth Architecture Block Diagram 2

3.1 RPT Implementation Block Diagram 3

4.1 Block Diagram of Cloud-Based Automation 2

6.1 Mesh Network in Open Thread[5] 1

6.2 Remote Co-processor Stack[4] 2

6.3 Network Co-Processor Stack[4] 2

Chapter 1

Introduction

1.1 Domain Of Work

Firstly my domain of work is in Bluetooth technology in that testing the Bluetooth on the android OS means with the help of the android GUI, not by writing the command manually.

In the Bluetooth testing team, learn to load the drivers, and firmware into the respective chipset and then test the quality of the packets that are sent to another device via Bluetooth. For eg, connect the Bluetooth speaker to the board through the android GUI by scanning and pairing the nearby device, then by playing the audio on the GUI side(Board side) to check the quality of the audio packets. So, this is how the testing of the Bluetooth technology on the android side.

Then after working in the automation department, the automation department does the scripting in the Perl language run the particular test cases automatically and the result has been generated at the end of the automation script.

In this automation department, it is further divided into two parts one is for

the Bluetooth automation testing and second is WLAN /Wi-Fi automation testing and another is the daily sanity (means whenever the new build is available in the server this daily sanity script gets a load that builds in the chipset and then the test cases get triggered via automation script of WiFi / Bluetooth and it also sends the report in the mail).

Most of the work is in Wifi-Automation testing and in daily sanity. In daily sanity it also sends the mail, in that mail, you have the test cases with the there results, my work is to compare the previous current execution test cases results and display it in the mail.

Another work of the domain is to validate the server-side from the user perspective. In this server there is the information of all the testbed with respective execution, user can also run the automation from the server GUI itself and the user can see the logs of each of the executions at the end of that particular execution.

So, this Server-based GUI is used when the user wants to run the automation in the online mode so, the information(test case name, number, result, comments) of that particular test case is being reflected in the server-based GUI.

Currently working on the triradio chip-set testing in the coex department in that work is to test the WLAN , Bluetooth and open-thread simultaneously on the different host like i.MX Linux, i.MX android and U16 , based on the observation we can write the results of the testing like for eg in the testing of the WLAN and Bluetooth so, we can observe in the throughput of WLAN by running the traffic between your DUT and router or DUT and MMH client and also in the background Bluetooth traffic is enable so, if there is the link loss in the WLAN throughput we say that the tescase is not working properly in this scenario.

1.2 Objective

Our objective is to make the hardware testing process easier and feasible by developing the automation script for different profiles and also after running the automation the summary is created of the selected test cases. So, by this summary, the user knows the result of the selected test cases with the particular comment.

So, by this process of automation, it is easy to know the error of the particular test cases after running the automation. The objective is to make the testing system more reliable and easy to use by running the automation script in the console host (Main system).

1.3 Problem Statement

As there was a problem in testing the chipset with so many test cases manually. We have to give the command manually for Eg for connecting the Bluetooth to the speaker, phone another reference device the command is to be given from the host to the controller, after that controller will give the user back with the event.

In short manual testing of the profiles takes time when there are so many testcase to be done so, for the solution to this problem automation testing is to be done , in that we have to just select the particular test cases and then want to trigger the test cases by running the automation script from the main host.

Chapter 2

Literature Review

2.1 Literature Review in Bluetooth

2.1.1 Bluetooth Architecture

As we all know Bluetooth is used for sending the data packets to the remote device wirelessly. It sends the data packets in the 2.4Ghz frequency band. It uses the specific channels usually uses 79 channels and having 16000 hops per second means the data can be switched among the 79 channels 16000 times per second.This can be done because wifi is also working on the 2.4Ghz band to avoid the overlap between the channels of Bluetooth wifi. So, these data packets are in the form of audio data, video data, image data PDF, etc, each data packet type has its profile or its feature to transmit the packets to the remote device.

Bluetooth has host plus controller architecture means there are two layers in the Bluetooth stack one is the Host that initiates the command to the controller via HCI (Host controller interface) and then the controller sends an event to the host that the command is successfully received and the task has been performed by the controller.

The host has various other layers in it. There are some protocols like SMP, SDP, ATT, L2CAP. This protocol is used to format the data and send it to the controller. The L2CAP layer that is just above the HCI interface is used to segregate or combine the data in one specific format and give it to the controller.

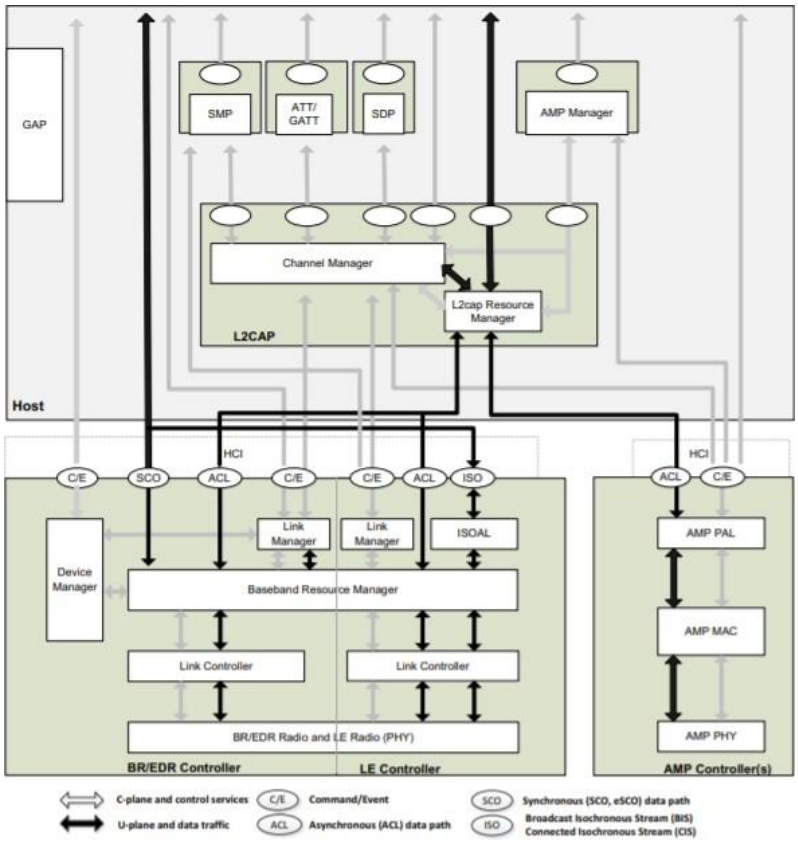

Figure 2.1: Bluetooth Architecture Block Diagram

After the controller receives the data from the host it initiates the link to the remote device and establishes the link between them, the link should be ACL, SCO link based on your profile(for eg your device is connected to the Bluetooth speaker so in that case, it establishes ACL link).

This link establishment and data transfer between two devices are done with the help of link manager link controller that is in the controller part and at last

PHY layer is used to send the data packets in the air using some modulation techniques and also channel management is done by this controller part.

2.1.2 Bluetooth Profiles

In Bluetooth, there are various types of profiles. Profiles mean what kind of data has been transferred between the devices (for eg if we want to play the song in the BT speaker then Bluetooth uses the A2DP profile to stream the audio from your phone to the speaker).

Advanced Audio Distribution protocol(A2DP): A2DP (Advanced audio distribution profile) is used when we want to transfer the audio data to the remote device meaning it is used to stream the audio data packets and uses an ACL connection between two devices.

In this the communication is in between two devices one of them is source(device that transmit the audio packets) and one is the sink (device that Receive the audio packets).

For eg : The audio streaming is done on the phone/latop/tablet means the audio data packets is being transmitted from the phone to the Sink device that is in the network of that phone / laptop / tablet , the sink device are of various types BT speaker, BT headphones etc.

Audio Video Remote Control(AVRCP): In this AVRCP(Audio Video Remote Control Profile) we should control the audio and video stream means if we want to play, pause, rewind the song or the video so this is done with the help of this profile and it has all the features of the A2DP profile.

Human Interface Device(HID) : HID(Human Interface Device) profile is used to transfer the data from your input devices like keyboards , mouse , speaker etc.LEHID is also the part of the HID profile in this low power devices are being connected to the tablet / laptop.

General Attribute Profile(GATT) : GATT (General Attribute profile) profile is used to transfer the data like battery level of the remote device, sensor data like heart rate data.GATT defines the standard way to discover the services, characteristics descriptors and then uses them.

GATT services mean it is a grouping of one or more attributes(attributes mean some sort of information). It performs services to the client and means to give the information to the client. For example, battery level services give the level of the battery in the remote device to the client and the same as temperature services give temp data.

GATT characteristic has a piece of information/data that is given to the client(phone, laptop, tab are the GATT client) from the server(speaker, headphone smartwatch are the GATT server).

Hands-free profile(HFP): In this type of profile, we can take a call, end the call and also talk from the remote device. This means it directly goes to the baseband of the controller side from the host site and sets up the link SCO(synchronous connection orientation) with the remote device like a phone, tab. In this, the data of the audio is real-time like real-time communication.

2.2 Literature Review In Automation

Generally, automation testing is used when we have to run so many test cases and it is not possible to do it manually then only automation comes in. It has its own automation script for different profiles like for Bluetooth there will be one automation script with their respective test cases, each test case has one script related to it, same as for the wifi but for WLAN it has a different script.

2.2.1 Process of Automation

The process of the automation is simple, we have to give first the configuration of the devices that have to be tested means to configure that device by giving/assigning the IP to it and we have to select the test cases for the respective test suites.

For eg., if we want to test for the Wi-FI(WLAN)on to the chipset then that Chipset acts as an STA(station), MMH(Mobile hotspot), etc and on the basis of the testing we have to select the test cases and configure them, after that, we have to run the script.

After all the configuration and selection of the test cases, we have to run the particular automation script for Bluetooth, if you are testing the Bluetooth and Wi-Fi if testing is WLAN.

At last, the result is saved in one folder and the user has to observe the result that has come for each of the test cases and in that folder, there are many files like RPT files, XLS files, YAML files, etc.

2.2.2 Automation Scripting Language

This automation script has to be written in Perl language, this language has been used for text manipulation and other features are also there. I also learned the Perl language for doing the automation scripting. I learned about the data types in Perl, array in Perl, creating the hash multi hashes, file handling in Perl, Regex in Perl.

2.2.3 Difference Between Automation testing and Manual Testing

Manual Testing:

This test is done by the QA itself by writing the specific command in the Linux host so, this is also called profile-based testing in terms of Bluetooth testing. So, In this one is the controller and the second is the host, we have to write the command in the Linux host to the controller and then the controller gives the event back with the status for eg Bluetooth is being scanned by writing a specific command in the Linux host controller give the scan result back to the Linux host.

Automotive Testing:

It is one type of controller-based testing, in this, we have to write the scripts in Perl, python, etc languages, and then we have to only run the script by specifying the test cases from the excel sheets, and then the results are to be generated in one folder and it consists of RPT file, YAML file, output Xls file, etc.

In automation testing, we did not have to write the command in the Linux host just we had to run the automation script of the particular profiles (like for Bluetooth profile we have to run the Bluetooth automation script and for the WLAN profile we have to run the WLAN automation script) in the Linux host.

For example, if we want to test the Bluetooth on the particular chipset we have to run the Bluetooth automation script in that host and at last, we get all the related files of that test suite in one folder.

2.2.4 Advantages of using Automation :

1. No need to write commands in the Linux host for testing.

2. Results are ready after each test case.

3. Each test case running is being saved means logs of each of the test cases are being saved.

4. We can also go through the other execution (previous execution) for just checking the history.

5. It is a feasible and time management process.

6. If test cases are not working or do not pass then it will also write the comment so the user can understand the error and it is useful for debugging.

7. We can use any Soc that is in the Setup for testing.

Chapter 3

Test Report File Comparison Implementation

3.1 Generation Of Report File

The report file is being generated at the last of the automation script, it has too much information about the particular test cases of the specific execution (information like test cases ID, test cases name, the result of the particular test cases and also there will be comment of that particular test cases). At the last of that file, it has a summary table in that we have calculated the failed/passed test cases.

Generation of the report file is done at the last of the automation but it is in real-time means if one test case is being executed of the particular execution and at the end of that test case, the information of that particular test case is being reflected in the RPT file, then only second test case is being executed and so on, at last, all the selected test cases are being listed in that generated RPT file with there information like test cases ID, result, comment, etc.

3.2 Need of Report File comparison Implementation

As the automation is carried out for each of the execution, each execution is having the same or different test cases with having different builds to it which means this Report file comparison implementation is mainly developed for testing the device on a daily basis whenever the new builds of that particular chipset is being there in the server (called as the daily sanity).

When new builds are available by the developer on the server the script takes that build and loads the drivers and firmware on the particular chipset, then triggers the test cases by running the WLAN / Bluetooth automation then the results are available in the form of an RPT file and the mail also. But if we want to compare the result of the current RPT file having new builds loaded on the chipset with the previous RPT file having old builds, by doing this we can have a record/track of the test cases of the particular execution.

3.3 Block Diagram of Report File Comparison Implementation

We have to compare the five previous execution report files with the current report file (means current RPT has test cases running with new builds and previous RPT files having the test cases with old builds).

As shown in the diagram we have to compare each of the previous RPT files with the current RPT file individually then we have a result of that in the output (an eg result of the previous 1 RPT file with the current RPT file).

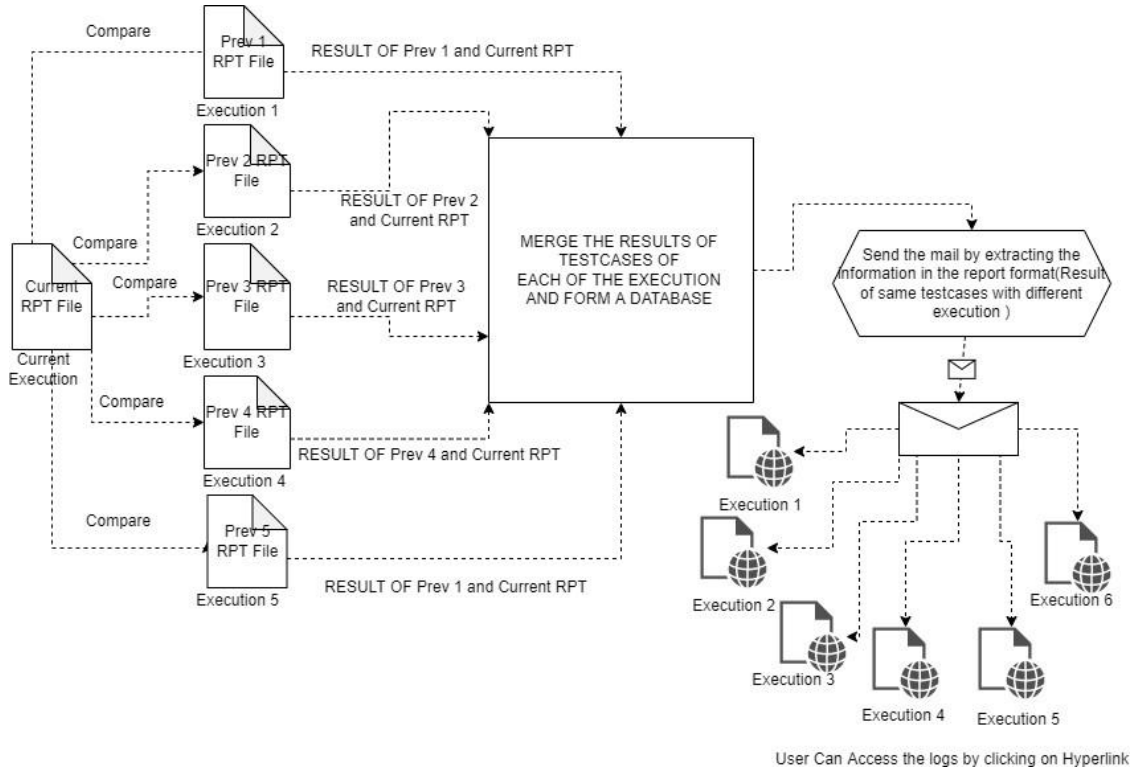

Figure 3.1: RPT Implementation Block Diagram

All the results are to be merged in the hash form like a database and then by extracting the data from the hashes or from the database we have to send the result to the mail in the form of the table (for Eg first column is test case name, the second column has the current result of that test case and the third column has the result of the five previous execution of that same test case).

In the third column(previous execution results) each execution result has a hyperlink attached with it so, user can directly go to the web page and access the logs of that particular execution of the particular test cases.

Chapter 4

Cloud-Based Automation (Block diagram)

In cloud-based automation, the test cases running in the online mode means the information of each of the test cases is being reflected the server/cloud means the automation is being triggered in the online mode.

As shown in the diagram that one controller host is there from which we have to execute/trigger the automation of any profile, it is connected to the UUT / REf via test network through the switch (means the controller host will available to give the commands via test network and also UUT / REF give the event to the controller host back).

When we trigger the automation in the online mode the data will push the data of the particular test cases to the cloud and it will display all the information in the dashboard(GUI). Users can also compare the execution in the server/cloud itself and also generate the report of the same.

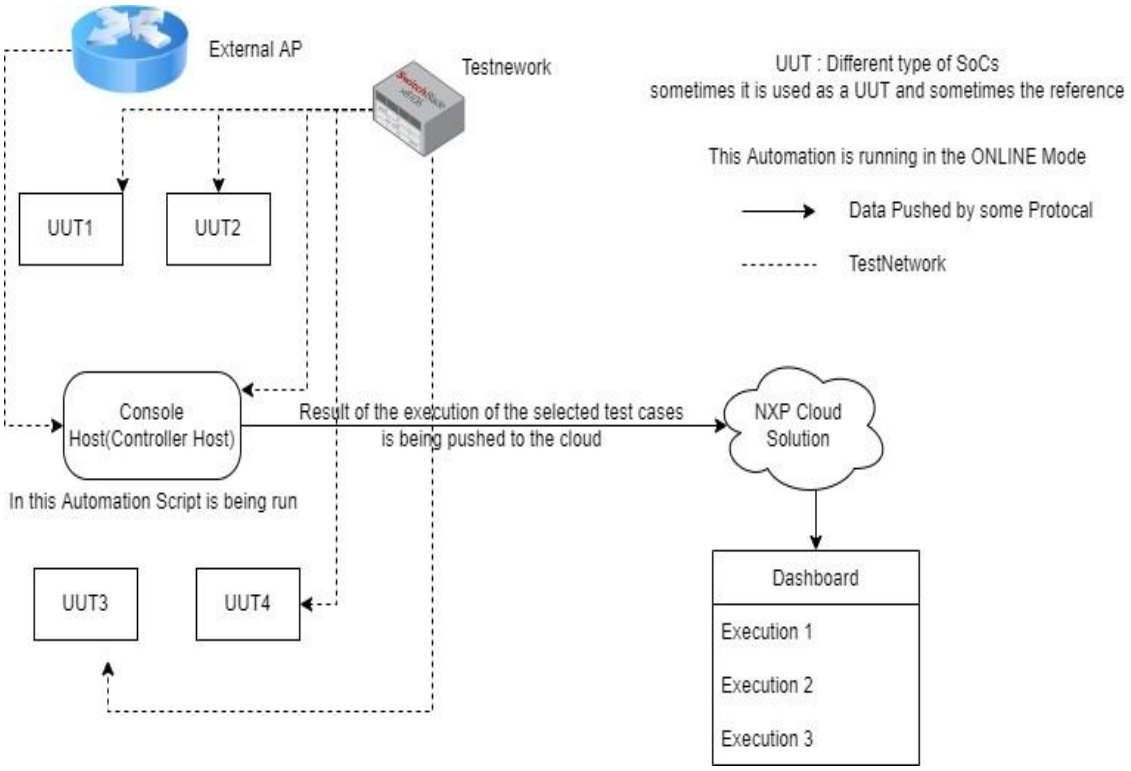

Figure 4.1: Block Diagram of Cloud-Based Automation

Chapter 5

Tri-Radio Chipset Testing

Tri radio chipset means three wireless technologies are being embedded inside the chipset, one is Bluetooth, the second is WLAN and the third is Open thread. so, in the coex testing of this chipset, we have to test all of the three technologies(WLAN+BT+OT) simultaneously.

This tri radio chipset is mostly used in smart home automation products and it is the first chipset with WLAN, BT, and OT altogether. Testing of this tri radio chipset is done via three types of host.

5.1 i.MX Linux

i.MX linux is the NXP proprietary OS, we have to flash the OS kernel in the i.MX8M mini board via Linux /windows and then attach the tri radio chipset with SDIO and UART interface to load the WLAN drivers/firmware and Bluetooth firmware respectively (WLAN Drivers and firmware are being loaded via SDIO, Bluetooth/open-thread firmware is being loaded via UART interface).

5.2 Ubuntu 16

In the U16 host, we have to just connect the tri radio chipset to the laptop having U16 OS via SDIO interface for loading the WLAN parameters(drivers firmware) and also connect the USB cable for loading the Bluetooth firmware but before that, we have to copy that related firmware in the particular location and for WLAN drivers we have to write the command in the user space called "insmod driver name. ko" so that it can insert the drivers in the kernel space and then after we have to start testing the WLAN and Bluetooth profiles.

5.3 i.MX Android

In this, we have to flash the Android BSP (currently version is 11) in the i.MX 8M mini board via Linux /windows laptop and then need to connect the tri radio chipset via SDIO(for loading the WLAN) and UART(for loading the Bluetooth) interface but before attaching the chipset we have to copy/push the related files of WLAN Bluetooth (drivers/firmware) in the shell at the particular location.

Chapter 6

Open-Thread Testing

6.1 Introduction Open thread

The open thread is the protocol by Google, this protocol is used in smart home automation. it is like the ZigBee protocol in the Bluetooth but it is an IP-based protocol (IPv6), like ZigBee only it creates the mesh network of the devices and each device knows the information of the other device, and also each device (REF) has its own IP address.

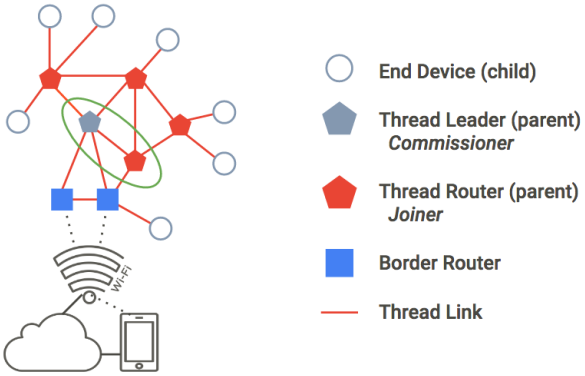

Figure 6.1: Mesh Network in Open Thread[5]

In the open thread protocol, the devices that create the mesh network have

their independent roles like a leader, router, child, and border router, so while testing the open thread protocol in the tri radio chip-set we have to assign them different roles by giving some specific OT command after running the OT demon at the host side.

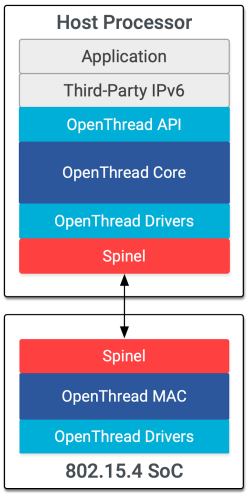

Figure 6.2: Remote Co-processor Stack[4]

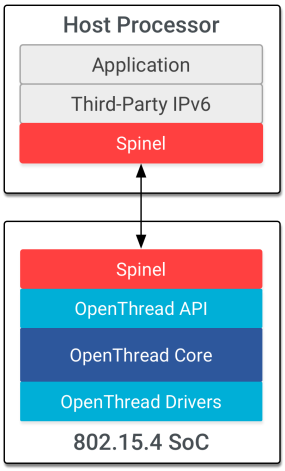

Figure 6.3: Network Co-Processor Stack[4]

Before assigning the role of the device we have to start the OT demon at the host side for eg . For the i.MX Linux host we have to first start the OT demon in the background that is used to communicate to the controller (chip-set) from your host so, the controller is your co-processor and the main processor is your host

processor. there are two types of co-processor stack for the open thread they are remote co-processor and network co-processor as shown in above figure.

6.2 Role of the Devices

6.2.1 Border Router

It is the main source of the devices from where the data is being pushed to the cloud / server or also being received from the server / cloud , then after receiving the data it is pushed to the leading device in the network.

6.2.2 Leader

Leader from which the connection gets established but for that we have to assign that role to the device by giving some specific command after that the leader has their network key and pan ID for the connection of router and child(end device).

[6]**For assigning leader role commands are as follows :**

1. **dataset init new** // for initiate the new connection so the leader gives the network key , panID and channel in the dataset itself.
2. **dataset channel 11** // change the channel from default to 11.
3. **dataset commit active**.
4. **ifconfig up** // assigning the IP to the leader.
5. **thread start** // start the thread network so that router / end device can scan the leader.

By writing the dataset command in the OT demon we know the network key and panID of that particular leader.

6.2.3 Router(Parent)

Router is connected to the leader through the network key and PanId that is generated while assigning the leader role to the device router is used to communicate other routers and end devices.

So, between the leader and end devices we have to connect the router for mesh networking.

[6]**Commands for assigning the Router role :**

1. **dataset channel 11**//append the channel 11 in the dataset.

2. **dataset networkkey ¡keys¿**//append the network key of the leader in the dataset of the router.

3. **dataset panid IDno**// append the PanID of the leader in the dataset of the router.

4. **dataset commit active**

5. **ifconfig up**// give the IP to the router.

6. **thread start** // start the thread and connected to the leader.

7. **state**// this command is used to know the role of the devices whether it is router / leader /end device.

6.2.4 End Device

End device is also get connected to the router by the network key and PanID of the leader, this end device knows the information of all other end device that is connected in the network through the router.

For knowing the connection between the leader ,router and end devices we have to ping the leader from the end device as each of them having there own IP so, we can run the UDP traffic from any of the devices

[6]**Commands for assigning the Child role :**

1. **mode r**//this is done because it does not acts as a router , it acts as a end device only. 1. **dataset channel 11**//append the channel 11 in the dataset.

2. **dataset networkkey ¡keys¿**//append the network key of the leader in the dataset of the end device.

3. **dataset panid IDno**// append the PanID of the leader in the dataset of the end device.

4. **dataset commit active**

5. **ifconfig up**// give the IP to the end device.

6. **thread start** // start the thread and connected to the router.

7. **state**// this command is used to know the role of the devices whether it is router / leader /end device.

But Before starting the thread network of the end device at the leader side we have to blacklist the end device mac address so that the end device is not connected to the leader , it is only connected to the router.

Chapter 7

Conclusion and Future Scope

7.1 Conclusion

As I work on the two projects, one is in the Automation testing and the second is in the COEX testing (in the particular chipset called as Tri-Radio chipset).

So, In the Automation testing, we implemented one feature called a Report file comparison, in which we compare the previous five executions with the current executions test cases result and make one table of that, then after the system itself sends the mail to the users with all the data included like test case name, current execution result, previous executions results with there specific date time. Also, understand the automation framework process and its advantages compared to manual testing.

Then after explored the tri-radio chipset by doing the testing on it of WLAN profiles , Bluetooth profiles, and open thread protocol simultaneously. So , during the testing I explored most of the profiles of WLAN , Bluetooth and open thread, like STA , MMH , P2P are for WLAN and HFP , A2DP , AVRCP , HID , LEHID etc for bluetooth and at last did some work in the open thread protocol , it is the

protocol that is used for the home automation and there testing is done by running the OT-daemon on it. The open thread is an open source so, anyone can explored that topic ,also there is the commands in the Github for testing from that command only we do our testcases

Then after I learned about automation testing and also understand the difference between automation testing manual testing. In the automation department, I learned about writing the script in Perl language for each of the profiles and also learned to set up the devices from the controller host (from where the automation script is being run).

7.2 Future Scope

1. As the new Tri-Radio chipset that is being tested is in the RD part so, it has lots more features to come like LE audio, dual HFP, and dual A2DP.

2. Currently the Builds / Release of the chipset is not as per the expectation so, in the future, there may be changes in the Builds to resolve the bugs/issue that are facing with current builds.

3. Most of the changing is being done at the open thread side because currently there had been an issue in that only the issue came when we test the open thread with the other protocol like WLAN and Bluetooth simultaneously.

4. In future the testing of the tri radio chipset will be done via an IMX android(Host) with version 12 currently the version is 11.

References

[1] Tutorialspoint.com. 2021. Perl Tutorial“[online] Available at: https://www.tutorialspoint.com/perl/index.htm

[2] GeeksforGeeks. 2021. Perl,Regular Expressions - GeeksforGeeks. “, [online] Available at: https://www.geeksforgeeks.org/perl-regular-expressions/.

[3] Moquality.com. 2021 “[online] Available at: https://www.moquality.com/blog/How-to-automate-testing-of-Bluetooth-features

[4] . OpenThread. 2022. Co-Processor Designs “OpenThread [online] Available at: https://openthread.io/platforms/co-processor.

[5] OpenThread. 2022. Simulating a Thread network with OpenThread “[online] Available at: https://openthread.io/codelabs/openthread-simulation-posix0.

[6] GitHub. 2022. GitHub - openthread/openthread:OpenThread released by Google is an open-source implementation of the Thread networking protocol “[online] Available at: ¡https://github.com/openthread/openthread¿.