RF Front End Characterization for Narrow and Wide Band Radios

Major Project Report

Submitted in partial fulfillment of the requirements for the degree of

Master of Technology
in
Electronics & Communication Engineering
(Embedded Systems)

By

Darshan P Shah
(13MECE22)

Electronics & Communication Engineering Branch
Electrical Engineering Department
Institute of Technology
Nirma University
Ahmedabad-382 481
May 2015

RF Front End Characterization for Narrow and Wide Band Radios

Major Project Report

Submitted in partial fulfillment of the requirements
for the degree of

Master of Technology
in
Electronics & Communication Engineering
(Embedded Systems)

By

Darshan P Shah
(13MECE22)

Under the guidance of

External Project Guide:

Mr. Preetham Raj
Manager, Hardware Development Engineering,
Broadcom Communications Technologies Pvt. Ltd.,
Banglaore.

Internal Project Guide:

Prof. Manisha Upadhyay
Associate Professor, EC Department,
Institute of Technology,
Nirma University, Ahmedabad.

Electronics & Communication Engineering Branch
Electrical Engineering Department
Institute of Technology
Nirma University
Ahmedabad-382 481

May 2015

Declaration

This is to certify that

a. The thesis comprises my original work towards the degree of Master of Technology in Embedded Systems at Nirma University and has not been submitted elsewhere for a degree.

b. Due acknowledgment has been made in the text to all other material used.

- Darshan P Shah

13MECE22

Disclaimer

"The content of this thesis does not represent the technology, opinions, beliefs, or positions of Broadcom Communications Technologies Pvt. Ltd, its employees, vendors, customers, or associates."

Certificate

This is to certify that the Major Project entitled **"RF Front End Characterization for Narrow and Wide Band Radios"** submitted by **Darshan P Shah (13MECE22)**, 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

Prof. Manisha Upadhyay
Internal Guide

Dr. N.P. Gajjar
Program Coordinator

Dr. D.K.Kothari
Section Head, EC

Dr. P.N.Tekwani
Head of EE Dept.

Dr. K. Kotecha
Director, IT

Certificate

This is to certify that Major Project entitled **"RF Front End Characterization for Narrow and Wide Band Radios"** submitted by **Darshan P Shah (13MECE22)**, towards the partial fulfilment of the requirements for the degree of Master of Technology in Electronics and Communication Engineering (Embedded Systems), Institute of Technology, Nirma University, Ahmedabad is the record of work carried by him under my supervision and guidance. In our opinion, the submitted work has reached the level required for being accepted for the examination.

Date: Place: Bangalore

Mr. Preetham Raj,

Manager, Hardware Development Engineering,

Broadcom Communications Technologies Pvt. Ltd.,

Bangalore

Acknowledgements

I would like to express my gratitude and sincere thanks to **Dr. P.N.Tekwani**, Head of Electrical Engineering Department and **Dr. N.P.Gajjar**, PG Coordinator of M.Tech Embedded Systems program for allowing me to undertake this thesis work and for his guidelines during the review process.

I take this opportunity to express my profound gratitude and deep regards to **Prof. Manisha Upadhyay**, guide of my major project for her exemplary guidance, monitoring and constant encouragement throughout the course of this thesis. The blessing, help and guidance given by her time to time shall carry me a long way in the journey of life on which I am about to embark.

I would take this opportunity to express a deep sense of gratitude to Project Manager **Mr. Preetham Raj**, Hardware Development Engineering, Broadcom Communications Technologies Pvt. Ltd. for his cordial support, constant supervision as well as for providing valuable information regarding the project and guidance, which helped me in completing this task through various stages.

I am obliged to **Vishwas Bhat** and **Sudarsh Mallaya**, staff members of Bluetooth RFDVT team, Broadcom Communications Technologies Pvt. Ltd. for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

Lastly, I thank almighty, my parents, brother, sisters and friends for their constant encouragement without which this assignment would not be possible.

- Darshan P Shah

13MECE22

Abstract

In the design and performance evaluation of radio communication systems, it is necessary to have an accurate characterization of RF front end. Front end is an important part of the design. It has Band Pass Filter, Power Amplifier, Low Noise Amplifier, Tx-Rx switch and matching network. Filter non-idealities, nonlinearity of power amplifier, high noise figure of Low Noise Amplifier and many other impairments affect the ideal behavior of the Front End design. Consequently, desired performance of the whole design is degraded. To improve degraded performance, it is necessary to know what exactly is causing the problem. This is the reason why characterization of RF Front End is essential.

RF Characterization of front end supports various tests. These tests can be divided in two parts: Transmitter tests and Receiver tests. Transmitter tests involve tests like Transmit Signal Strength Indicator, Error Vector Magnitude, Harmonic Distortion, Out of Band Transmission, etc. At receiver side, tests like Receiver Sensitivity and Noise Figure are required. Out of these, EVM test, TSSI test and Receiver Sensitivity (Rx BER and PER) test are carried out.

Nonlinearity, phase and gain imbalance, DC offsets, spurious tones, phase noise, etc. are degrading factors of EVM. So, a single EVM test result gives information about all these factors. That is why EVM is the most important test to evaluate transmitter section of the front end. Noise added by the receiver system affects the receiver sensitivity. Therefore it is necessary to perform Sensitivity and Noise Figure tests for receiver system and evaluate their results.

To accomplish these tests, automated test set-up (station) plays an important role. Bringing up the station from non connected test instruments to final working condition of station is an important part in testing phase. Station bring-up which requires loss calibration and sanity tests to validate test result data is carried out. Result data of the tests executed on automated test stations is stored in station controller/computer. Arranging this raw result data in a proper format and plotting it, is an important task to make result analysis easy. The automation tool named Report Generator has already been developed to accomplish the same task. Python automaton scripts written for this tool had bugs which are identified and removed. Moreover, existing code for report generator is modified to add a new functionality.

Contents

List of Tables

List of Figures

Abbreviation Notation and Nomenclature

PER	Packet Error Rate
EDR	Enhanced Data Rate
NBR	Narrow Band Radio
WBR	Wide Band Radio
DUT	Device Under Test
HCI	Host Controller Interface
USB	Universal Serial Bus
UART	Universal Asynchronous Receiver/Transmitter
GPIB	General Purpose Interface Bus
RF	Radio Frequency
BT	Bluetooth
TSSI	Transmitter signal strength indicator
SA	Spectrum Analyzer
Vdet	Voltage Detector
ADC	Analog to Digital Convertor
PA	Power Amplifier
EVM	Error Vector Magnitude
Tx	Transmission
Rx	Reception
CW	Continuous Wave
IM	Intermodulation
HS	High Speed
HD	Harmonic Distortion
LOFT	Local Oscillator FeedThrough
IQ	In phase Quadrature phase
PAPR	Peak to Average Power Ratio
BR	Basic Rate

BLE .... Bluetooth Low Energy

ISM .... Industrial, Scientific and Medical

BPF .... Band Pass Filter

OP .... Output Power

FEM .... Front End Module

Chapter 1

Introduction

1.1 Motivation

The growing demand for high date rate communication in WLAN has driven the market toward the 5GHz IEEE 802.11n and 802.11ac standards. These standards are based on orthogonal frequency division multiplexing modulation consisting of 52 subcarriers, each subcarrier being 256QAM (11ac), 64QAM (11n), 16QAM, QPSK and BPSK modulated. Orthogonal frequency division multiplexing systems demand high peak to average power ratio for the modulated signal so that it satisfies IEEE 802.11a linearity requirement. Therefore, maximum output power of the power amplifier which is required should be much higher than the needed average power of the PA [1].

In modern radio telecommunication systems, a fully integrated power amplifiers design with a reasonable output power, high gain and high efficiency remains challenging considering the linearity aspect of the power amplifier. Linearity and efficiency are the two important characteristics of power amplifiers, either of them has to be improved by trading-off for the other parameter [1].

Receiver systems are normally required to process very small signals. The weak signals cannot be processed if the noise magnitude added by the receiver system is larger than that of the received signal. Increasing the desired signals amplitude is one method of raising the signal above the noise of the receiver system. Signal amplitude can be increased by raising the transmitters output power. Alternately, increasing the antenna aperture of the receiver, the transmitter, or both allows a stronger signal at the receiver input terminals. Increasing the physical size of an antenna is one method to increase its aperture.

Higher heat dissipation is typically required to increase transmitter output power. Cost, government regulations, and interference with other channels also limit the transmitter power available for a given application. Physically increasing transmitter antenna size may cause weight and wind load problems on the tower to which the antenna is mounted.

Increasing receiver antenna size obviously increases the housing size for a portable product enclosing the antenna structure, such as a cellular telephone or pager. Because raising the desired signal amplitude above the noise added by the receiver may not always be practical, a weak signal might be processed by lowering the added noise. In this case, the noise must be decreased such that the noise amplitude is somewhat below the weak signal amplitude [2].

1.2 Problem Definition

RF front-end is a responsible part in any radio communication system, for delivering the proper signal coming out of chip to the antenna-port. It is important to characterize this Front End Module for its compatibility with the Wi-Fi/BT chip driving it. In practical scenario, when a particular FEM is used along with the Wi-Fi/BT chip driving it, FEMs inherent properties tend to change as what is specified by its datasheet. This may happen because of nonlinearity of Power Amplifier, filter non-idealities, high noise figure of Low Noise Amplifier and many other impairments that may affect the ideal behavior of the Front End Module. To improve degraded performance, it is necessary to know what exactly is causing the problem. Characterization of RF Front End helps to discover cause of the problem and find possible solution.

1.3 Contribution

RF Characterization of front end is an essential part in design of WiFi/BT chip. It supports various tests. These tests can be divided in two parts: Transmitter test and Receiver test.

In transmitter test, I have performed tests like Error Vector Magnitude, Harmonic Distortion, Out of Band Transmission, Transmit Signal Strength Indicator. In receiver tests, tests like Sensitivity (RxPER) and Noise Figure are performed.

I have also analyzed these tests' results for how the result of a particular test looks, whether it needs improvement or not, and if improvement is needed, then what exactly is causing the problem, etc. Some of these tests are explained in detail.

I worked on Vector Network Analyzer to calibrate automated test set-ups (stations). Station calibration is required for correctness of test results for the tests performed on these stations.

1.4 Thesis Organization

Rest of the thesis is organized as follows:

Chapter 2 contains brief detail of various IEEE WLAN standards, comparison between standards and some modulation schemes falling into IEEE 802.11b standard. It also describes the basics of Bluetooth wireless technology and different versions of Bluetooth, Bluetooth core system Architecture in which BR/EDR technology is briefly discussed. The essential part in chip design i.e. front end module, is describes in this chapter.

Chapter 3, Front End characterization of Transmitter, presents TSSI and EVM tests. Test set up for both the tests are shown. It explains steps for performing the tests and their result analysis.

Chapter 4, Front End characterization of Receiver, covers detailed description of sensitivity test. It is the most important test on receiver side. It characterizes behavior of the chip and helps improve the design of receiver.

Chapter 5, Bluetooth automated test setup is described in brief. How this setup is brought up from non-connected devices to its final working state is also explained. Sanity test is an important part to validate results of all the stations.

Chapter 6, Python Report Generator Automation Tool is developed to organize different DUT data for different temperature in a single file to make result analysis easy. This chapter explains result file structure, functionality of report generator, and tools basic code architecture with pseudo code. It also explains what are the issues found with default codebase and their workaround.

In Chapter 7, Conclusion and Scope for future work is presented.

Chapter 2

Literature Survey

2.1 Study of various WLAN standards

In 1990, the IEEE 802 executive committee established the 802.11 working group to create a WLAN standard. The standard specified an operating frequency in the 2.4-GHz ISM (industrial, scientific, and medical) band. The standard supports about 1 to 2 Mbps data rates. But after that the need for higher data rates raised which lead to the invention of more WLAN standards. Authorization requests of two projects for higher rate physical layer extensions to 802.11 were approved by working group. IEEE 802.11b5 GHz and IEEE 802.11a2.4 GHz were the two extensions [1].

IEEE 802.11b operates at 2.4-GHz ISM band and it supports two modulation schemes frequency hopping spread spectrum (FHSS) and direct sequence spread spectrum (DSSS) implementations. 802.11b is a PHY extension to the original 802.11 standard. It also operates at the 2.40-GHz band and allows for higher data rates of 5.5 and 11 Mbps. It uses a technique known as complementary code keying (CCK). The 802.11a is another PHY extension to the 802.11 standard. It operates at the 5-GHz unlicensed national infrastructure for information (UNII) band and allows for data rates of 654 Mbps. It uses a technique known as orthogonal frequency division multiplexing. The next extension to the 802.11 standard was the 802.11g. Operation of it is done at the 2.4-GHz ISM band and it supports data rates from 1 to 54 Mbps. The 1 and 2Mbps data rates are provided by the DSSS mode whereas the 51, 2 and 11Mbps rates are provided by CCK mode. Moreover, rates at 6 to 54 Mbps are exercised in OFDM mode. The 802.11g standard uses the OFDM technique and 802.11a standard data rates but operation band is the 2.4-GHz ISM band. Therefore, very high data rates are operable while achieving backward compatibility with the 802.11b standard [1].

2.1.1 Comparison of 802.11: b, a, g, n, ac

802.11b, 802.11a, and 802.11g are the three known versions of the 802.11 physical. As explained before, the 802.11g & 802.11a standards provide much higher speed than 802.11b. However, the emergence of 802.11g and a will not cause the end of 802.11b in the near future. Applications are there where the low system cost and low power consumption are required. In these cases 802.11b might yet be the best solution in the near future. At the other side, most system traders have moved to 802.11g standard, which is backward compatible with 802.11b [1].

As an option to 802.11g and b, if one requires higher density of user, higher network capacity and data rate, 802.11a is the one that he or she has to choose because of wider spectrum at 5 GHz and higher data rate offered by this standard. For applications where higher data rates and longer ranges are required, 802.11g is the most suitable. The 802.11g gives the added advantage of backward compatibility with 802.11b, which has the largest existing base [1].

Increase in rate over 802.11a/g can be achieved in IEEE standard 802.11n. This can be achieved through wider signal bandwidth of 40 MHz, MIMO implementation

Table 2.1: Comparison of IEEE Standards

Standard	**Frequency Band**	**Modulation Scheme**	**Data Rate (Mbps)**
802.11	2.4 GHz	DSSS, FHSS	1,2
802.11b	2.4 GHz	CCK	5.5, 11
802.11a	5 GHz	OFDM	6 to 54
802.11g	2.4 GHz	DSSS+CCK+OFDM	1 to 54
802.11n	2.4 + 5 GHz	OFDM	Up to 600
802.11ac	5 GHz	OFDM	Up to 6.933

and higher coding rates. The signal is modulated using OFDM modulation scheme. It uses 77 Modulation and Coding Schemes (MCSs). Up to 600 Mbps data rate can be achieved.

Standard IEEE 802.11ac incorporates MIMO mode configurations, where we have multiple antennas on transmitter and receiver side (2x2 or higher). The total band of emission is 160 MHz, where each antenna emits in 80 MHz band for 2x2 mode. High data rate (in Gigabits) can be achieved.

2.1.2 IEEE 802.11b modulation schemes

2.1.2.1 Direst Sequence Spread Spectrum (DSSS)

In this scheme, each bit in original signal is represented by multiple bits in the transmitted signal. This is called chips. No. of chips per bit is called chip rate. Spreading code spreads signal across a wider frequency band. It is in direct proportion to number of bits used.

One technique combines digital information stream with the spreading code bit stream using exclusive-OR Gate. This is illustrated in the Figure 2.1.

DSSS using BPSK:
DSSS is generated by multiplying BPSK signal to c(t).

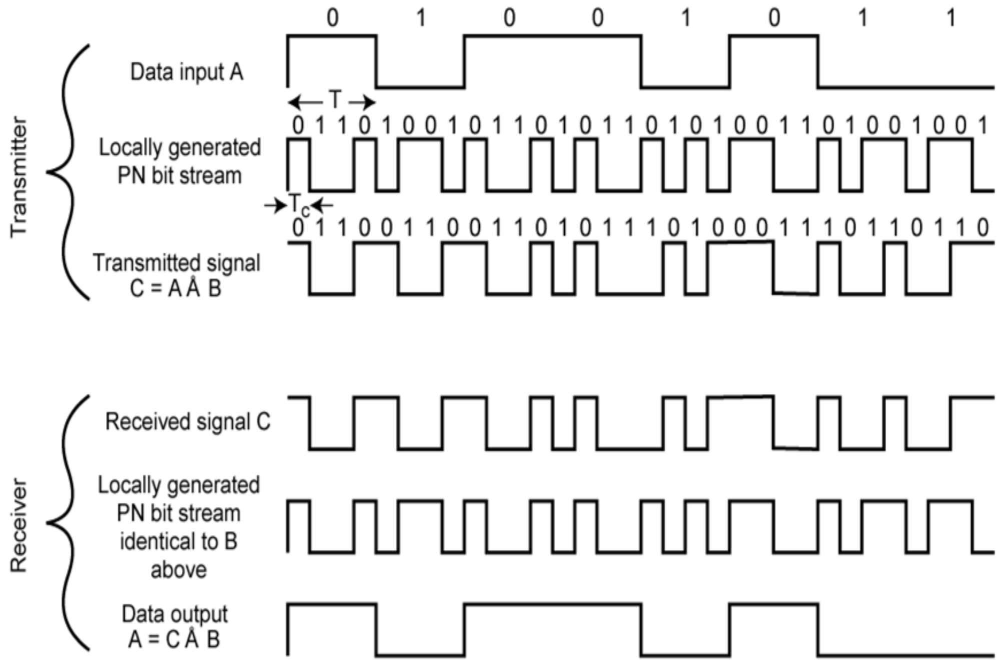

Figure 2.1: Example of Direct Sequence Spread Spectrum

For BPSK signal,

$$s_d(t) = A\,d(t)\,cos(2\pi f_c t)$$

Spread signal is given by,

$$s(t) = A\,d(t)\,c(t)\,cos(2\pi f_c t)$$

where,

- A= Amplitude of signal
- fc= Carrier Frequency
- d(t) = Discrete function, which can take values between [+1,-1]

This is illustrated in Fig.2.2.

At receiver, incoming signal multiplied by c(t). Since c(t) * c(t) = 1, incoming signal is recovered. The idea is shown in the Fig.2.3.

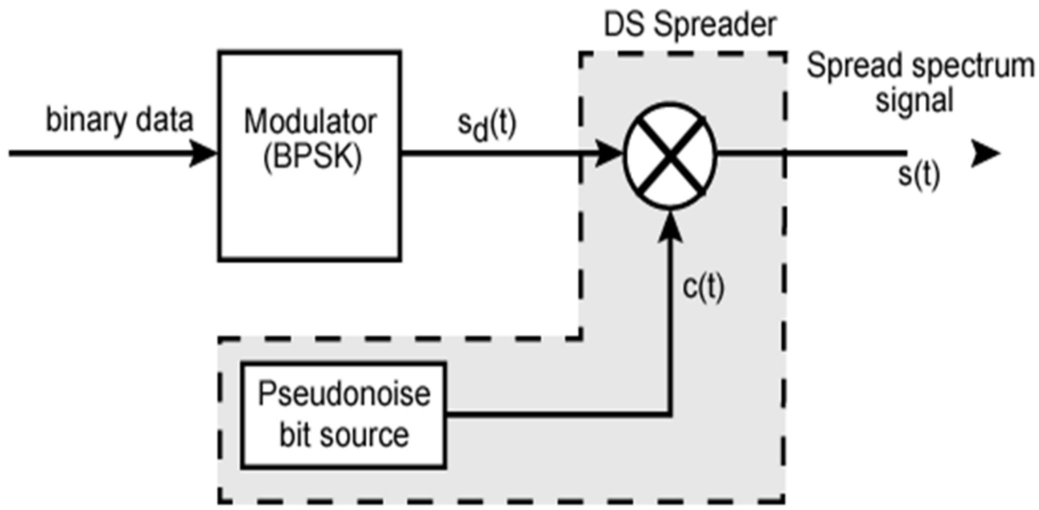

Figure 2.2: DSSS Transmitter

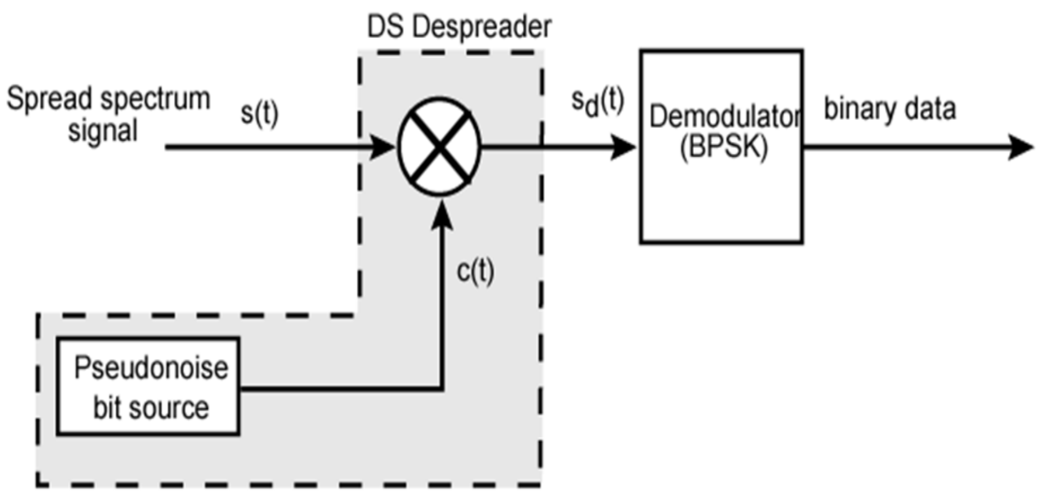

Figure 2.3: DSSS Receiver

2.1.2.2 Complementary Code Keying

The IEEE 802.11 complementary spreading codes have a code length 8 and a chipping rate of 11 Mchip/s. The 8 complex chips comprise a single symbol. By making the symbol rate 1.375 MS/s the 11Mbps waveform ends up occupying the same approximate bandwidth as that for the 2Mbps 802.11 QPSK waveform thereby allowing for 3 non-overlapping channels in the ISM band. This is important for maximizing aggregate system throughput in a wireless LAN network and was one reason for choosing CCK as the modulation technique. The 8-bit CCK code words are derived from the following formula:

$$\begin{aligned} c \;=\; & \{e^{j(\phi 1+\phi 2+\phi 3+\phi 4)},\ e^{j(\phi 1+\phi 3+\phi 4)},\ e^{j(\phi 1+\phi 2+\phi 4)},\ -e^{j(\phi 1+\phi 4)},\ e^{-j(\phi 1+\phi 2+\phi 3)}, \\ & e^{j(\phi 1+\phi 3)},\ -e^{j(\phi 1+\phi 2)},\ e^{j(\phi 1)}\} \end{aligned} \tag{2.1}$$

where, C is the code word with LSB first to MSB last. This strange looking formula is used to generate the code sets for both 11 and 5.5 Mbps data rates. Thus a subset of the 11Mbps code set is used at the 5.5Mbps data rate. The parameters $\phi 1$ - $\phi 4$ determine the phase values of the complex code set and are defined in the 802.11 high rate standards. For the 11Mbps data rate each symbol represents 8 bits of information. At 5.5Mbps 4 bits per symbol are transmitted. For the purpose of this discussion the 11Mbps mode will be described [3].

The data bit stream is partitioned into bytes as (d7, d6, d5, ..., d0) where d0 is the LSB and is first in time. The 8 bits are used to encode the phase parameters $\phi 1$ - $\phi 4$ according to scheme shown in Table 2.2. The encoding is based on differential QPSK modulation as specified in Table 2.3

Referring to Equation 2.1, we see that phase parameter $\phi 1$ is contained in all 8 chips of the code word so it essentially rotates the whole vector. This is important

Table 2.2: Phase Parameter Encoding Scheme

DIBT	**Phase Parameter**
(d1, d0)	c
(d3, d2)	$\phi 2$
(d5, d4)	$\phi 3$
(d7, d6)	$\phi 4$

Table 2.3: DQPSK Modulation of Phase Parameters

DIBT(di+1,di)	**Phase**
00	0
01	π
10	$\pi/2$
11	$-\pi/2$

in the circuit implementation of the CCK modulation.

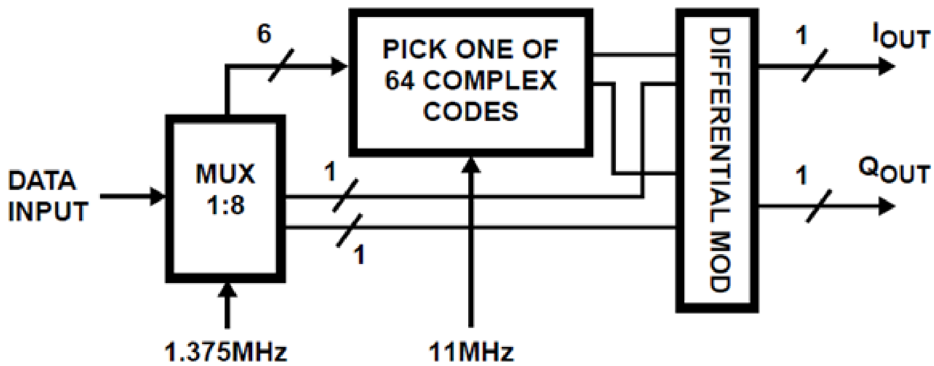

Figure 2.4: Block Diagram of Modulator Circuit

Figure 2.4 shows the block diagram of the CCK modulator circuit. The output of the data scrambler is partitioned into bytes and fed to a serial in parallel out mux circuit that gets clocked at the symbol rate of 1.375 MHz. Six bits of the mux output are used to select one of 64 complex codes which are fed to a differential modulator circuit. The other 2 bits of the mux output are used to QPSK modulate, i.e., rotate, the 8 chip complex code word. The outputs of the differential modulator are the I and Q outputs in accordance with Equation 2 for generating complex codes. And that is essentially CCK modulation in a nutshell.

2.2 Bluetooth Wireless Technology and Specifications

Bluetooth technology is a short-range communications technology that is simple, secure, and everywhere. We can find it in billions of devices ranging from mobile phones and computers to medical devices and home entertainment products. It is intended to replace the cables connecting devices, while maintaining high levels of security.

The key features of Bluetooth technology are robustness, low power, and low cost. The Bluetooth Specification defines a uniform structure for a wide range of devices to connect and communicate with each other.

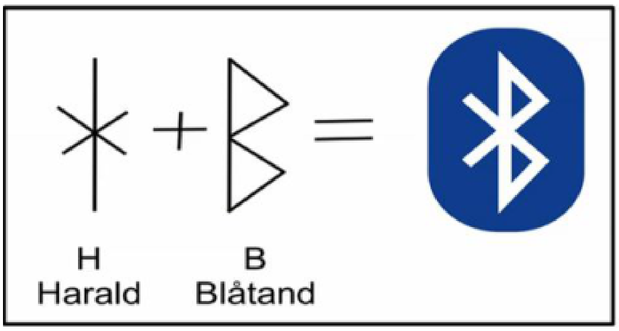

Figure 2.5: Bluetooth Logo named from Danish King Harald Blatand

When two Bluetooth enabled devices connect to each other, this is called pairing. The structure and the global acceptance of Bluetooth technology means any Bluetooth enabled device, almost everywhere in the world, can connect to other Bluetooth enabled devices located in proximity to one another.

A fundamental strength of Bluetooth wireless technology is the ability to simultaneously handle data and voice transmissions. which provides users with a variety of innovative solutions such as hands-free headsets for voice calls, printing and fax capabilities, and synchronization for PCs and mobile phones, just to name a few.

The range of Bluetooth technology is application specific. The Core Specification mandates a minimum range of 10 meters or 30 feet, but there is no set limit and manufacturers can tune their implementations to provide the range needed to support the use cases for their solutions.

2.2.1 Bluetooth Technology Specifications

Bluetooth specifications are formalized by the Bluetooth Special Interest Group (SIG).Bluetooth Core Specification provides product developers both link layer and application layer definitions, which support data and voice applications [4].

- **Spectrum:**

Bluetooth technology operates in the unlicensed industrial, scientific and medical(ISM) band at 2.4 to 2.485 GHz, using a spread spectrum, frequency hopping, full-duplex signal at a nominal rate of 1600 hops/sec. The 2.4 GHz ISM band is available and unlicensed in most countries.

- **Interference:**

Bluetooth technology's adaptive frequency hopping (AFH) capability was designed to reduce interference between wireless technologies sharing the 2.4 GHz spectrum like with Wi-Fi. AFH works within the spectrum to take advantage of the available frequency. This is done by the technology detecting other devices in the spectrum and avoiding the frequencies they are using. This adaptive hopping among 79 frequencies at 1 MHz intervals gives a high degree of interference immunity and also allows for more efficient transmission within the spectrum. For users of Bluetooth technology this hopping provides greater performance even when other technologies are being used along with Bluetooth technology.

- **Power Emission and Control:**

Bluetooth uses spread spectrum transmission scheme and thus FCC permits the transmit power to reach a maximum of 20 dBm, resulting in a range of 100 meters. However, most Bluetooth devices transmit at a much low level in order to save battery power.

Table 2.4: Bluetooth Power Classes

Power Class	**Output Power(Max)**	**Range(meters)**
1	100 mW (20 dBm)	100
2	2.5 mW (4 dBm)	22
3	1 mW (0 dBm)	6

Three power classes are defined in the Bluetooth specification as shown in Table 2.4. Power Class 3 is the most common scheme adopted by manufacturers (for cell phones and other battery devices). Laptops generally contain Class 2 devices. Power Control (optional) operates by the receiver monitoring the Received Signal Strength Indicator (RSSI) and sending LMP (Link Manager Protocol) control commands back to the transmitter to appropriately increase or decrease the transmission power level.

2.2.2 Different Versions of Bluetooth Technology

There are multiple versions of Bluetooth which is IEEE Standard 802.15.1 and not all Bluetooth specifications are the same. All of the newer Bluetooth versions are backward-compatible.

- **Bluetooth Version 1.0 and 1.0B:**

  - Includes Bluetooth hardware device address (BD ADDR) transmission in the Connecting process
  - Had many problems
  - Difficulty to make product inter-operable

- **Bluetooth Version 1.1 (BR- Basic Rate):**
  - Error fixed which found in V1.0B
  - Includes RSSI(Received Signal Strength Indicator)
  - Offers such basic features as voice dialing, call mute and last-number redial L2CAP packets

- **Bluetooth Version 1.2 (BR- Basic Rate):**
  - Faster Discovery and Connection
  - Added AFH Adaptive frequency-hopping spread spectrum Feature which improves resistance to radio frequency interference by avoiding the use of crowded frequencies in the hopping sequence
  - Added Flow Control and Retransmission Modes for L2CAP.

- **Bluetooth Version 2.0 + EDR (Enhanced Data Rate):**
  - Introduce Enhanced Data Rate (EDR) for faster data transfer. (2.1 to 3 Mbit/s)
  - Provide lower power consumption through a reduced duty cycle

- **Bluetooth Version 2.1 + EDR (Enhanced Data Rate):**
  - Secure simple pairing (SSP): Improves the pairing with Bluetooth devices, while increasing the use and strength of security

  - Extended inquiry response (EIR): Provides more information during the inquiry procedure to allow better filtering of devices before connection and sniff mode which reduces the power consumption in low-power mode

- **Bluetooth Version 3.0 + HS (High Speed):**
  - Theoretical data transfer speed up to 24 Mbit/s (Not on BT Link but on collocated 802.11 link)
  - High speed alternative MAC PHY 802.11 AMP (Alternative MAC/-PHY):

    Proves low power connection models of Bluetooth are used when the system is idle, and the faster radio is used when large quantities of data need to be sent

- **Bluetooth Version 4.0 (BLE- Bluetooth(Smart) Low Energy):**
  - Ability to run for years on standard coin-cell batteries
  - Ultra-low peak, average and idle mode power consumption
  - Low cost
  - Multi-vendor interoperability
  - Enhanced range
  - Secure encrypted connections at the lowest possible cost

- **Bluetooth Version 4.1:**
  - Co-ordination of transmission with LTE(Long Term Evolution)and reduced interference [5]
  - Better Connections: By making the reconnection time interval flexible and variable
  - Improved Data Transfer
  - Single device acts as both a Bluetooth Smart peripheral and a Bluetooth Smart Ready hub at the same time

Figure 2.6: Lithium Button cell power source for BLE Peripherals [7]

2.2.3 Overview of BR/EDR Technology

For BR Technology, the symbol rate is 1 Mega symbol per second (Ms/s) supporting the bit rate of 1 Megabit per second (Mb/s). For EDR Technology, a gross air bit rate of 2 or 3 Mb/s. These modes are known as Basic Rate and Enhanced Data Rate respectively [6].

2.2.4 Basics of Bluetooth Low Energy (BLE)

Bluetooth Low Energy is the new Technology which optimized for 'Ultra Low Power'. The Lithium Button cells are the main power source for peripherals as shown in Figure 2.6. It expects to operate without battery change for an extended period of time (more than one year).

It supports very low data rates. For BLE, data throughput is not a meaningful parameter. It does not support streaming. It has a data rate of 1 Mbps, but is not optimized for file transfer. It is designed for sending small chunks of data like Temperature reading, Time, Speed Measures etc. as shown in Figure 2.7.

The reason for consuming lower power is no connection needed before transmitting. BLE is the best alternative for applications that are 'always-off'.

In BLE, Receiving is more expensive and Transmitting is cheap. So that it

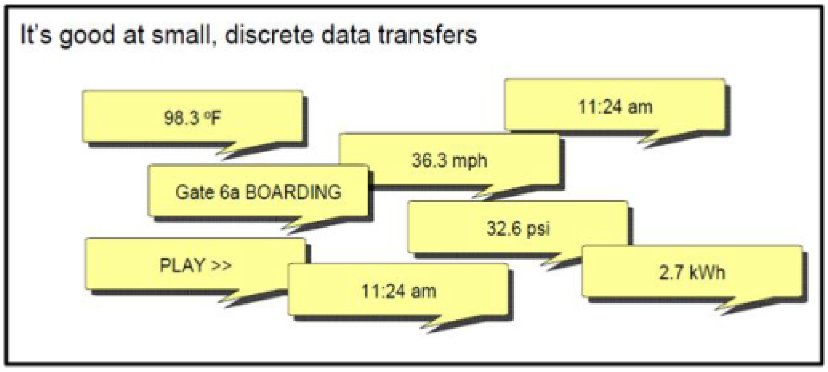

Figure 2.7: BLE designed for sending small chunk of data[7]

optimizes to reduce Rx time as much possible.

2.2.5 Comparison of LE with BR/EDR Technology

Comparison of LE Technology with BR/EDR Technology is shown in Table 2.5.

2.2.5.1 Differences between LE and BR/EDR Technology

The differences in LE Technology compared to BR/EDR Technology which makes it efficient are as follow:

- Efficient discovery / connection procedures: Just use 3 Advertising Channels
- Very short packets around 32 bytes
- Asymmetric design for peripherals: Master-Time Sensitive and Slave-Power Sensitive
- Connections are entered and always exist in a sniff like mode: Low Power mode- listening activity of the slave is reduced

2.2.5.2 Similarities between LE and BR/EDR Technology

The basic similarities between LE and BR/EDR Technology are as follow:

Table 2.5: Comparison of LE with BR/EDR Technology

Feature	**BR/EDR**	**LE**	**Notes for LE**
RF Channels	79	40	1 MHz Spacing in BT; 2 MHz Spacing in LE
Modulation	GFSK	GFSK	Simple and Effective
Modulation Index	0.25 to 0.35	0.45 to 0.55	Wider signal More Robust
Range (Typical)	30 meters	50 meters	Modulation Index, Increased Power for Class 2
Packet format	6	2	ID, FHS, DM, DH, 2-DH, 3-DH; Advertising / Data
Ack Packet Length	126 s	80 s	63

- Reuse existing
  - BR architecture
  - HCI logical and Physical transports
  - L2CAP packets
- Uses the same 2.4 GHz ISM License-free Band

2.3 RF Front End

2.3.1 Overview of RF Front End

In a radio receiver circuit, the RF front end is a generic term for all the circuitry between the antenna and the first intermediate frequency (IF) stage. It consists of all the components in the receiver that process the signal at the original incoming radio frequency (RF), before it is converted to a lower intermediate frequency (IF).

For most super-heterodyne architectures, the RF front end consists of:

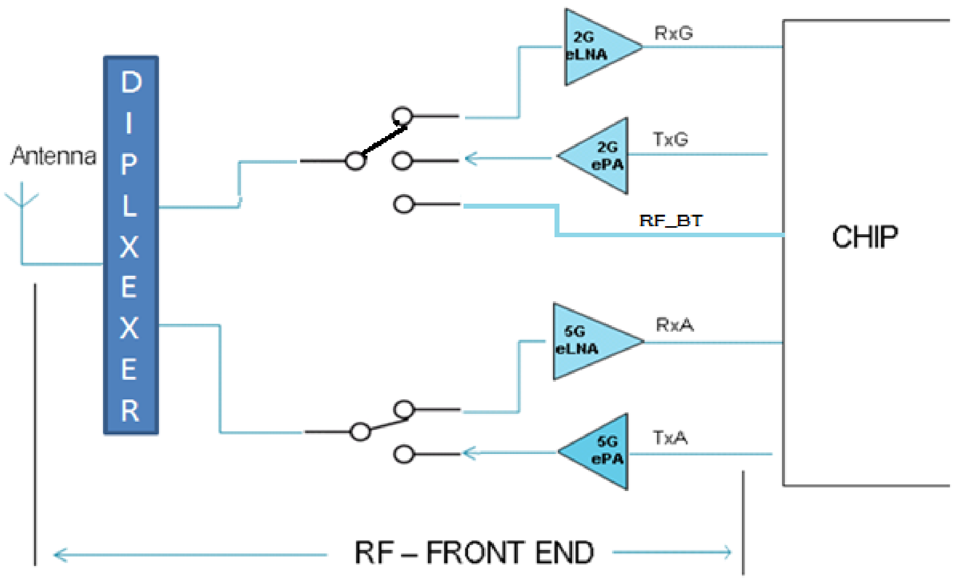

Figure 2.8: RF Front-End Architecture

- An impedance matching circuit to match the input impedance of the receiver with the antenna, so the maximum power is transferred from the antenna;
- A 'gentle' band-pass filter (BPF) to reduce strong out-of-band signals and image frequency response;
- An RF amplifier often called the low-noise amplifier (LNA). Its primary responsibility is to increase the sensitivity of the receiver by amplifying weak signals without contaminating them with noise, so they are above the noise level in succeeding stages. It must have a very low noise figure (NF). The RF amplifier can be omitted (or switched off) on frequencies below 30 MHz, where the signal-to-noise ratio is defined by atmospheric and man-made noise.
- The mixer, which mixes the incoming signal with the signal from a local oscillator (LO) to convert the signal to the intermediate frequency (IF).

A typical RF Front-End is as shown in the figure 2.1. The RF front end is usually defined as everything between the digital baseband system and the antenna.

For receiver, this region contains down-conversion mixer(s), filters, low-noise amplifiers (LNAs) required to process the modulated signals received at the antenna into signals appropriate for input into the baseband analog to digital convertor (ADC). This is the reason why the front end is usually known as the analog to digital or RF to baseband section of a receiver.

Radios work by receiving RF waves containing formally modulated data sent by a RF transmitter. The receiver is essentially a low noise amplifier that down converts the incoming signal. Hence, selectivity and sensitivity are the main concerns in receiver design.

An outgoing signal is up converted by transmitter before power amplifier passage. In such case non linearity of power amplifier is the main concern.

2.3.2 Higher Levels of Integration

In any system which relies on the transmission and/or reception of wireless signal such as multimedia devices and mobile phone, RF front end would be there. This demands integration which leads to add more functionality into a single chip. The reason behind this is growing demand of smaller product size, low power consumption and lower costs. Basic RF architecture is filtering, detection, amplification and demodulation regardless of integration level.

Modulated signals are passed through antenna to RF front end of receiver. Output signals of the front end are in the form of analog baseband signal which are ready to be converted from analog into digital form. In digital form, information is extracted from digitized carrier waveform.

Before the origin of integrated modules, there were separate components for each

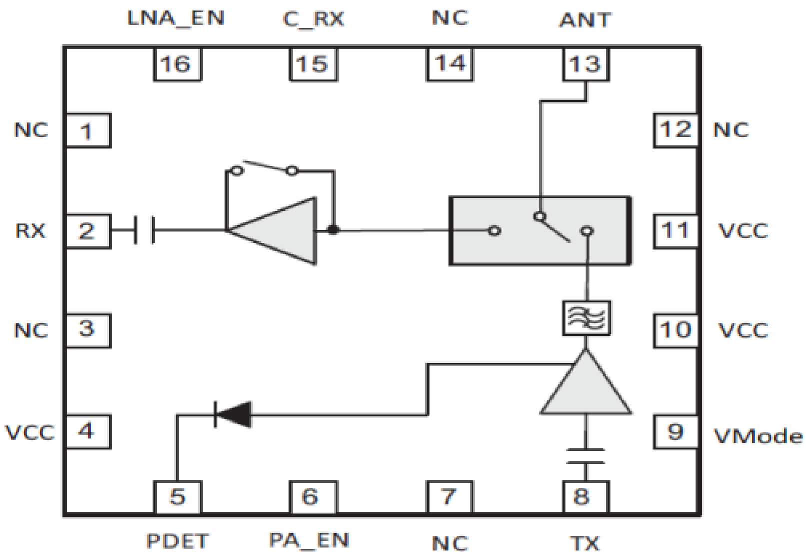

Figure 2.9: Functional Block Diagram of a FEM (Front End-Module)

functional block of the RF front end i.e. separate components for the amplifier, mixer-demodulator, detector and RF filter.

2.3.3 Front-End Module

WLAN (Wi-Fi) Front-End Modules basically integrate multiple devices required to implement an RF front-end. Typically included are one or more power amplifiers with input/output impedance matching (usually matched to 50 ohms), a switch and one or more low noise amplifiers.

Different vendors provide a complete integrated solution in a single front end module (FEM) for Wi-Fi 802.11a/n systems. Figure 2.9 shows the functional block diagram of a FEM. The ultra-small form factor and integrated matching minimizes the layout area in the customer's application and greatly reduces the number of external components. This simplifies the total front end solution by reducing the bill of materials, system footprint, and manufacturability cost.

The FEMs usually integrates a power amplifier (PA), single pole double throw switch (SP2T), LNA with bypass, and a power detector coupler for improved accuracy. These devices are provided in a 2.5mm x 2.5mm x 0.45mm, 16-pin QFN package. Different module meets the RF front end needs of IEEE 802.11 a/n/ac WiFi RF systems.

2.4 Basic Transceiver Block Diagram

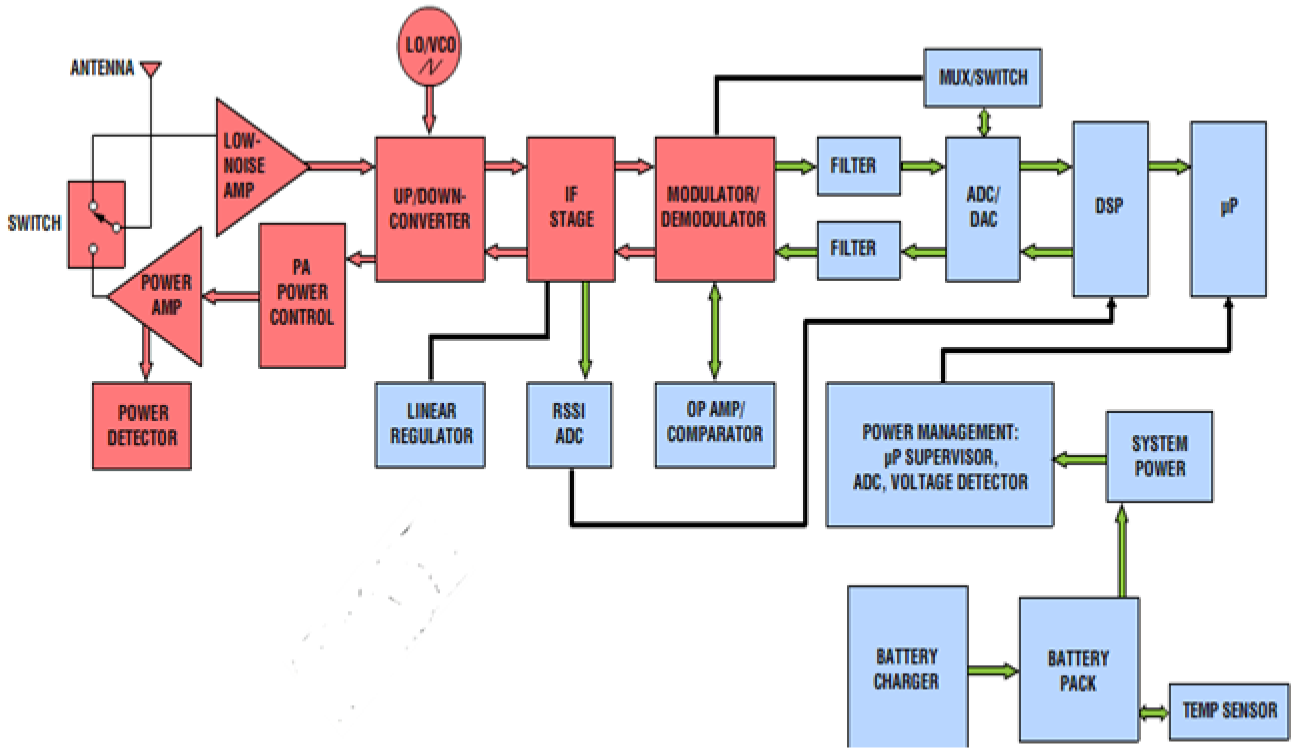

Figure 2.10: Transceiver Block Diagram

Chapter 3

Front End Characterization of Transmitter

3.1 For Narrow Band Radios

Bluetooth frequencies are all located within the 2.4 GHz ISM band. The ISM band typically extends from 2400 MHz to 2483.5 MHz (i.e. 2.4000 - 2.4835 GHz) (including guard bands). The Bluetooth channels are spaced 1 MHz apart, starting at 2 402 MHz and finishing at 2 480 MHz. This can be calculated as 2401 + n, where n varies from 1 to 79 [2]. Each channel has a bandwidth of 1 MHz. Due to its narrower bandwidth, Bluetooth technology is Narrow Band Radios.

3.1.1 Transmit Signal Strength Indicator (TSSI) Test

TSSI (Transmit signal strength indicator) is an indicator of the power level being transmitted by the antenna. Higher the TSSI number, the stronger is the signal being transmitted from the device. This parameter is very critical in making the chip know what power it is transmitting actually.

Vdet (V-detector) is the analog form of the indicator of power being transmit-

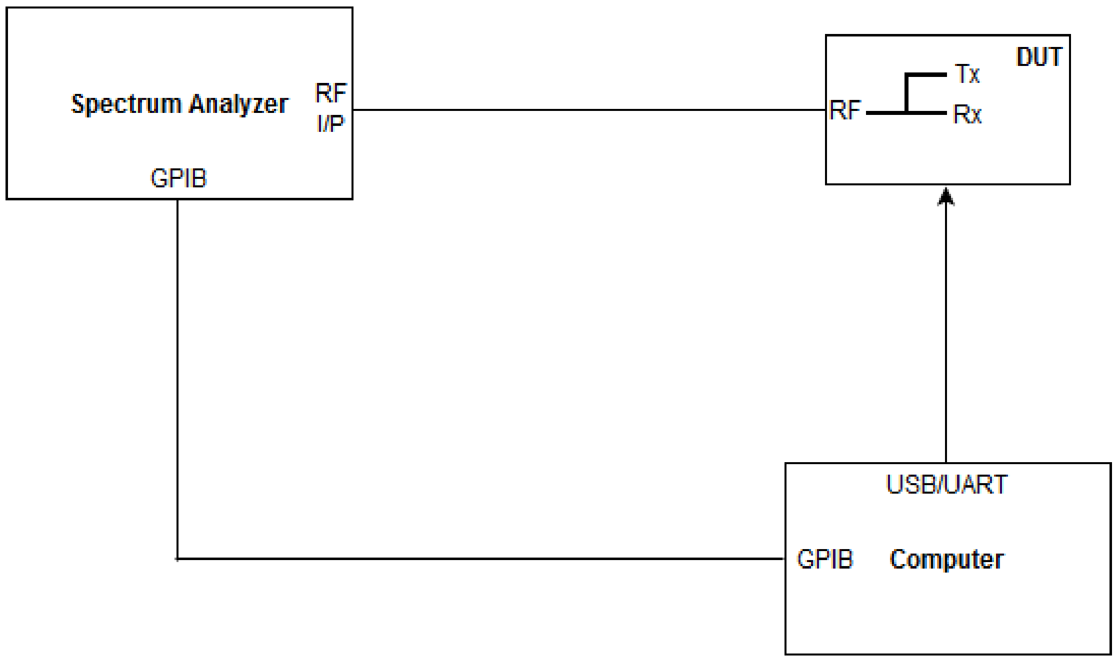

Figure 3.1: Setup for measuring TSSI

ted. It is difficult to have power couplers implemented in the FEMs hence the corresponding voltage (Vdet) is fed back to the ADC inside the chip. ADC converts the analog voltage into digital raw form and this output of ADC is what we use as TSSI.

SA is used to measure power of the transmitted RF signal from DUT at a particular frequency. The setup for measuring TSSI is shown in Figure 3.1. The commands to DUT from computer are sent by means of HCI interface via USB/UART connection. The computer sends commands to control programmable test and measurement device like spectrum analyzer via GPIB bus.

DUT transmits the tone at a particular frequency for a given Power Amplifier (PA) gain. For that frequency, as we discussed through feedback via Vdet we will get TSSI reading at the output of the ADC. After TSSI reading, PA gain value will be incremented by one. Power and TSSI value will be measured again for that PA gain. This iteration will be repeated until PA gain maximum limit is reached.

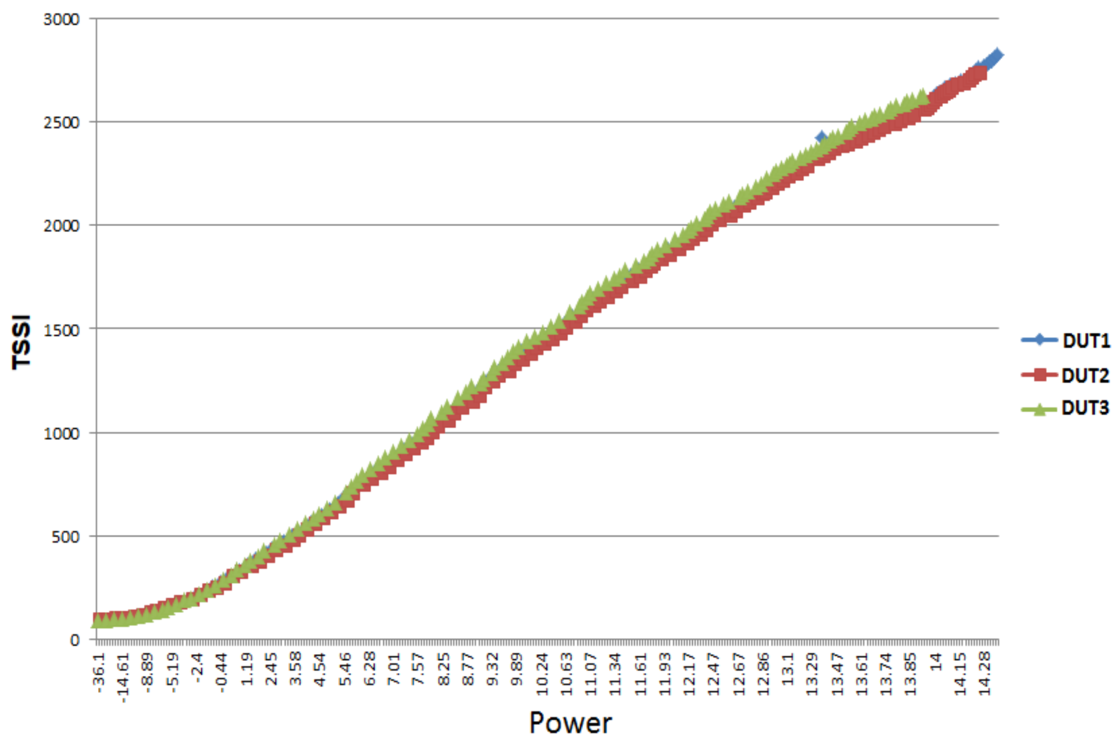

Figure 3.2: TSSI Vs Power

TSSI should ideally increase linearly with increase in power. Figure 3.2 compares TSSI reading for different DUTs at different output power levels.

TSSI log to excel template automation:

Excel template for tssi is having info sheet and data sheets for different frequencies of 2GHz band and different modulations (GFSK and EDR). Info sheet will be containing information about DUT and multiple frequencies and modulations used to generate results. Data sheets will be having test result data and plots.

Results are plotted in excel template and we get our result into raw log format, which is not in excel format. So we have to import raw data into excel template and plot the results to analyze. Raw to excel automation script is used to do the same task. Flow chart of the automation script is illustrated in Figure 3.3.

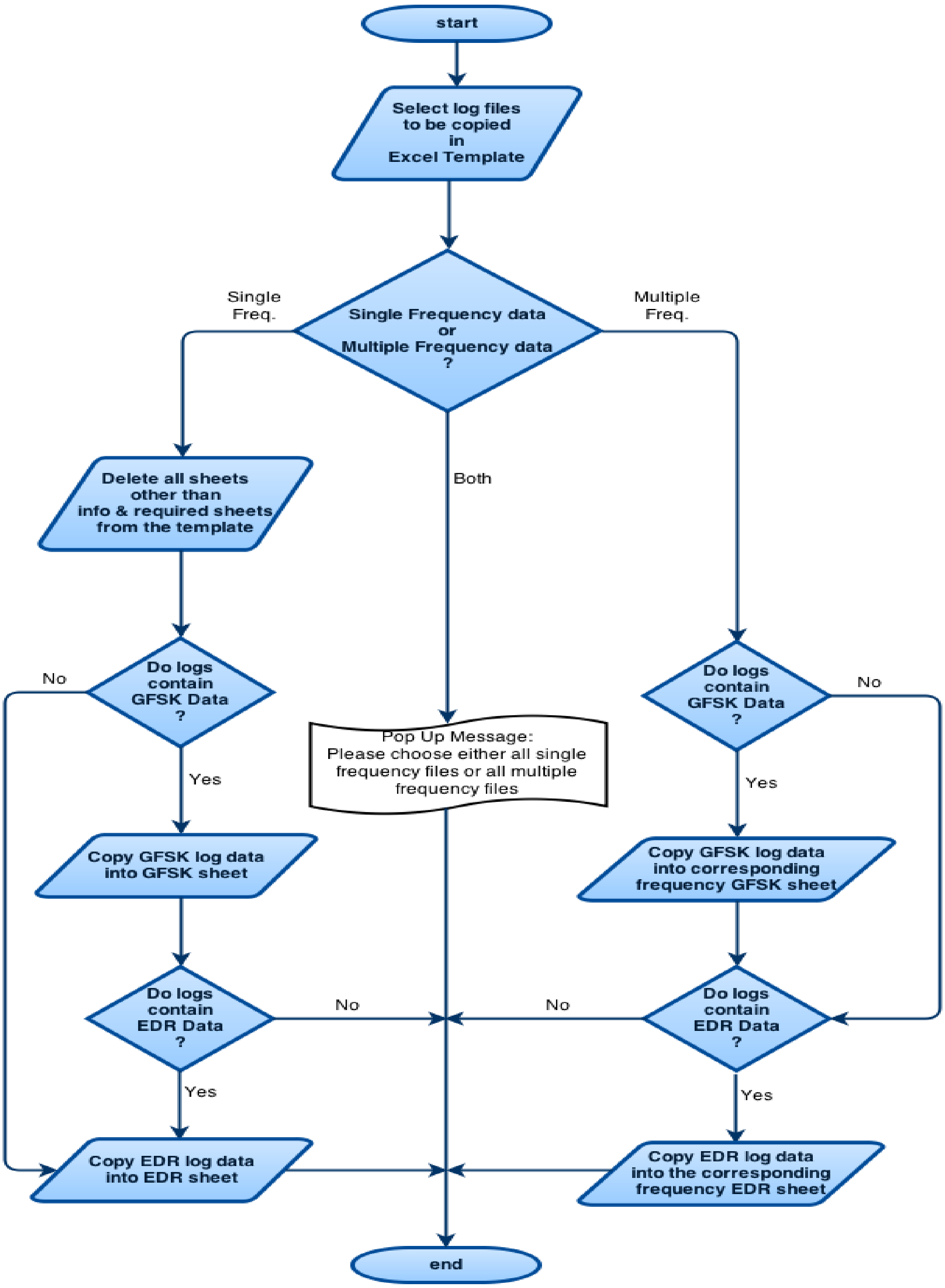

Figure 3.3: log to excel automation script flow-chart

3.2 For Wide Band Radios

3.2.1 Error Vector Magnitude (EVM) Test

One of the important tests which characterize the front end while in transmission (Tx) mode is transmitter EVM, which, in general is a single scalar number that is an indication of the modulation quality of the signal. To calculate the EVM, one needs to compare the actual symbols with their ideal impairment-free symbols on the constellation diagram and compute the error vectors as shown in Figure 3.4. The real symbol will have a different phase and amplitude as compared to the ideal symbol constellation points. Systematic and deterministic errors would simply offset the real constellation points as compared to the ideal ones. Nonsystematic impairments such as noise, however, would cause an "error ball" or "error cloud" of uncertainty in the constellation points about the ideal constellation points. For the 802.11a and g standards, at 54 Mbps, the transmitter EVM (or transmitter quality) is required to be a minimum of -25 dB [4]. Mathematically, for a given symbol, EVM is defined as,

$$\mathrm{EVM} = \sqrt{\frac{\sum_{i=1}^{M} |Z(i)-R(i)|^2}{\sum_{i=1}^{M} |R(i)|^2}}$$

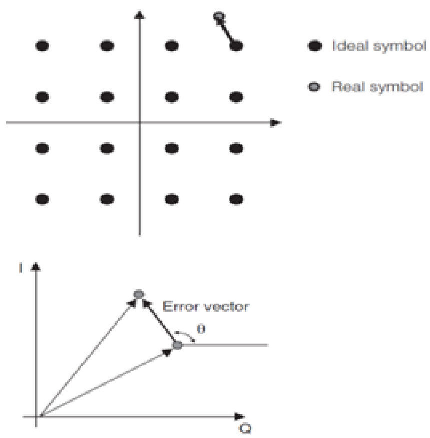

Figure 3.4: Pictorial description of the concept of EVM[4]

Where Z is the measured signal, R is the reference (ideal) signal, M is the number of measurements, and i is the measurement index. This definition can be extended to all the symbols of a modulation by averaging over all these symbols.

It can be seen from the definition that the EVM is an estimate of the magnitude of the error signal as compared to the magnitude of the ideal signal. As such, it is clear that that the maximum value for the EVM is 1, or 100%, and that the minimum value of the EVM is 0. It is clear that the smaller the value of the EVM in percent, the higher the quality of the modulated signal. The EVM can be expressed in decibels as 20 log (EVM). For example, 1% EVM can be expressed as 40 dB EVM.

The EVM is affected by a variety of impairments. These include nonlinearities, phase noise, quadrature imbalances, and filter shapes and bandwidth. The reason that the EVM is so commonly used as a measure of the quality of the transmitter is that it is essentially impacted by much such impairment [4].

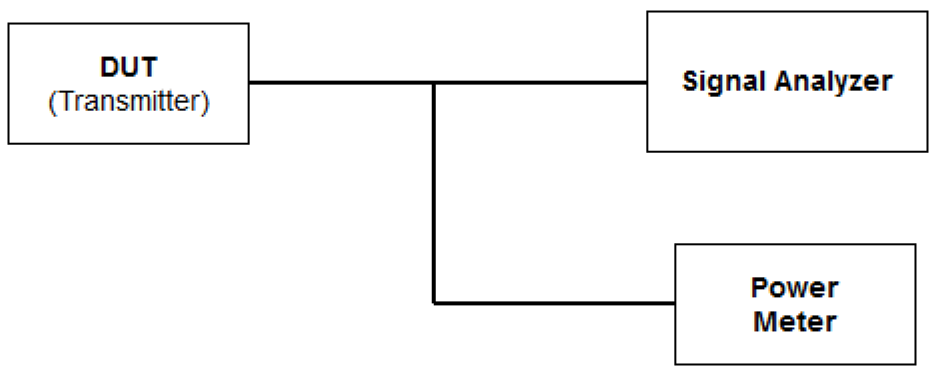

Figure 3.5: Setup for measuring EVM

This test is performed on transmitter board station. Test setup for measuring EVM is illustrated in Fig. 3.5. Spectrum Analyzer will calculate EVM and other required parameters and Power Meter will measure transmitter output power.

Single EVM test gives information about plenty of parameters like EVM vs power, Local oscillator feed through (LOFT), IQ parameters, PAPR (peak to average power ratio), frequency offset (ppm), etc.

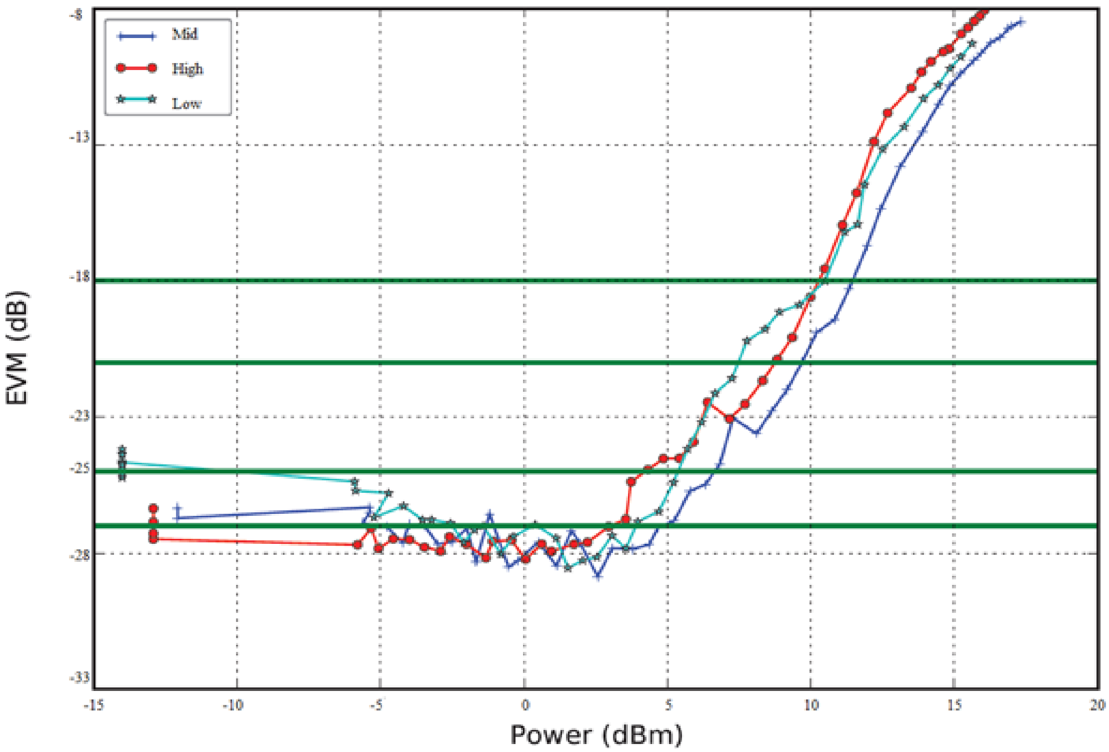

Figure 3.6: EVM Vs Output Power

Figurte 3.6 shows EVM test result by plotting EVM Vs Output Power. In EVM Vs Transmit Power plot, EVM noise floor and spread i.e. difference of transmit power at a particular EVM level for low, mid and high channels are observed. Lower EVM noise floor and lesser spread are desired. Increment of EVM with increase in output power is due to nonlinearity of PA. As we start increasing power above certain level, PA will start moving into nonlinear (Saturation) region.

LOFT is amount of local oscillator frequency power radiated at the output of the mixer. The results are better if we get lesser LOFT value. I-Q imbalance gives idea about fluctuation of the constellation points around reference (ideal) constellation points.

Frequency offset is measured in parts per million (ppm). 1 ppm corresponds to 1Hz frequency change per 1 MHz center frequency. It indicates how much actual center frequency is shifted (offset) with respect to desired one.

PAPR is peak to average power ratio. The large PARs significantly complicate the design of the radio and the mixed-signal blocks. The signal path will have to be designed with much more severe linearity constraints than traditional one. In particular, on the transmit signal path, the design of the power amplifier becomes quite challenging. Not only are designing high linearity power amplifiers quite challenging, but such amplifiers have much worse efficiencies than their nonlinear counterparts.

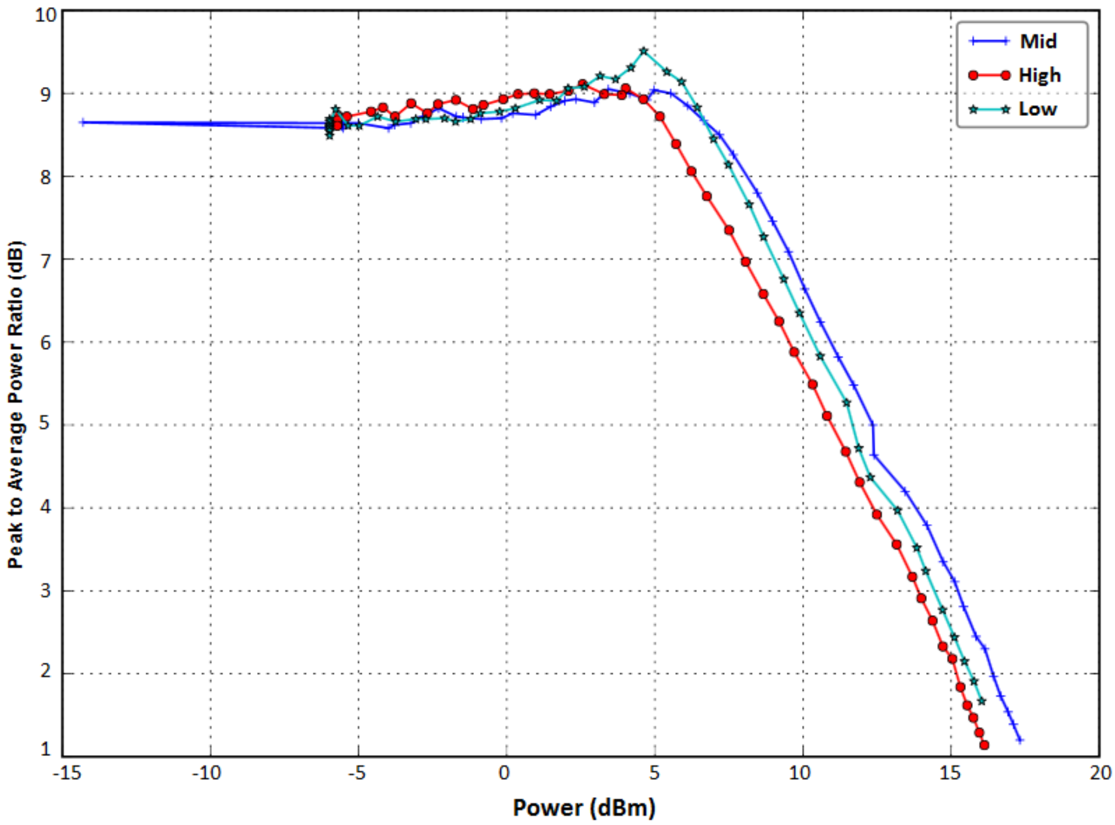

Figure 3.7: PAPR Vs Output Power

Effect of nonlinearity of power amplifier is shown in Fig. 3.7. As power increases, PA will gradually start moving into nonlinear region. So peak of the signal will be clipping the saturation region of the amplifier. As a result, for higher power levels PAPR will decrease significantly. This is shown in Fig. 3.7. Ideally it should be constant.

Chapter 4

Front End Characterization of Receiver

4.1 For Narrow Band Radios

Bluetooth frequencies are all located within the 2.4 GHz ISM band. The ISM band typically extends from 2400 MHz to 2483.5 MHz (i.e. 2.4000 - 2.4835 GHz) (including guard bands). The Bluetooth channels are spaced 1 MHz apart, starting at 2 402 MHz and finishing at 2 480 MHz. This can be calculated as 2401 + n, where n varies from 1 to 79 [2]. Each channel has a bandwidth of 1 MHz. Due to its narrower bandwidth, Bluetooth technology is Narrow Band Radios.

4.1.1 EDR Sensitivity Test

Receiver sensitivity is one of the important tests which characterize the front end while in receiving mode. Receiver sensitivity is defined as minimum amount of power for which receiver is able to detect the received message with BER less than or equal to 0.1 percent. A Bluetooth receiver must have sufficient sensitivity to sustain communication at low input power levels and tolerate a degree of poor transmitter performance [9].

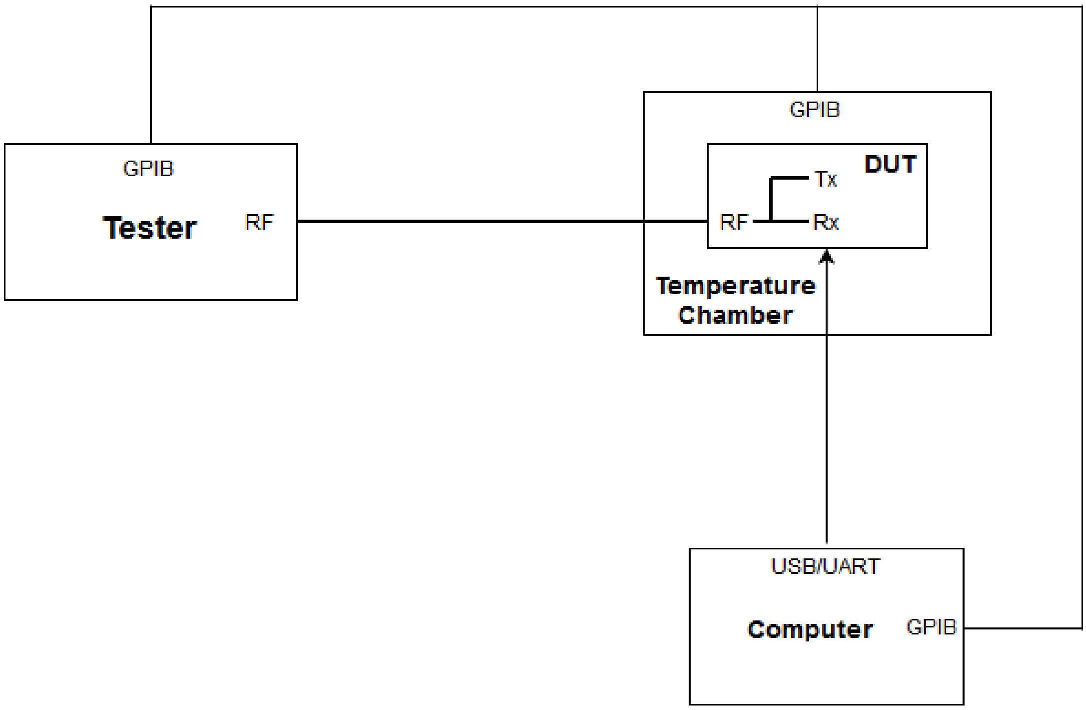

Figure 4.1: Sensitivity Test Setup

The sensitivity is tested using tester referred as Bluetooth test set. The set up for measuring sensitivity is shown in Figure 4.1. DUT is connected with tester via RF connection. The commands to DUT from computer are sent by means of HCI interface via USB/UART connection. DUT is kept inside of temperature chamber so that we can measure the sensitivity at different temperatures i.e. for different test conditions. The computer sends commands to control programmable test and measurement devices like tester and temperature chamber via GPIB bus.

Test Procedure:

1. DUT first receivers from tester at lowest frequency 2402 MHz. The tester continuously sends packets to the DUT. The tester's transmit power is chosen such that the input power to the DUT receiver is 70 dBm.

2. The returned packets from DUT to tester (via loop back) are received and the

BER is measured by the tester according to Bit error rate (BER) measurements. BER measurements are carried out by comparing data in the payload fields transmitted by the tester with data in the payload fields received from the DUT.

Let the number of payload bits counted in error be α, and let the number of payload bits received (via loop back) from the DUT be β, then:

$$\text{BER} = \alpha \;/\; \beta$$

This BER should be less than or equal to 0.001 after minimum number of samples, 16,00,000 payload bits are returned.

3. The input power to the DUT is decreased by 1 dB and step 2 is repeated again for the same frequency until BER becomes equal to 0.001. The power level at this stage is the sensitivity of the receiver for that frequency i.e. channel.

4. Repeat steps 2 to 4 while the DUT receives at 2402 + n, where n varies from 1 to 78 and is incremented by one after each iteration.

The EDR Sensitivity test is performed for 2 Mbps and 3 Mbps data rate. The tester continuously sends $\pi/4$-DQPSK packets for 2 Mbps EDR sensitivity test. While for 3 Mbps data rate, it uses 8 DPSK packet type. This test produces the plot of Sensitivity (dBm) Vs. Channel.

This measurement tests the minimum signal level required to produce a maximum value of BER. This can be limited by noise in the test environment so this must be taken into consideration when constructing a manufacturing test. It tests the performance of the filter and demodulator for an EDR signal [3].

Figure 4.2 shows sensitivity result for 3 Mbps while test environment is noisy. We can see fluctuation of sensitivity over the span of 2 GHz band. Ideally BT re-

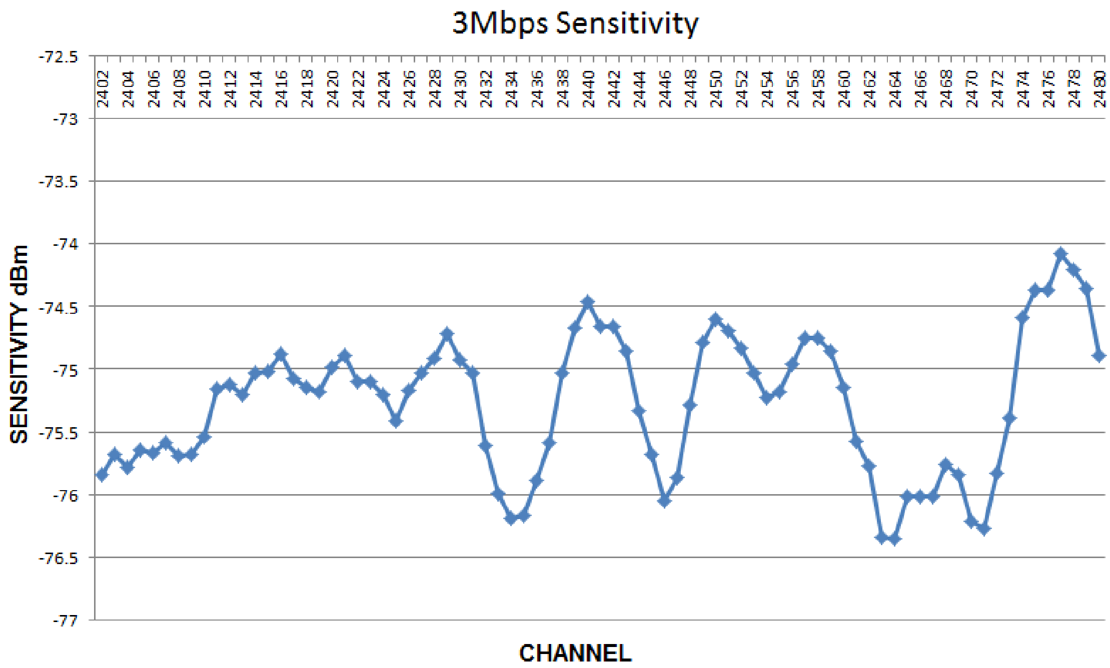

Figure 4.2: 3 Mbps Sensitivity in Noisy Environment

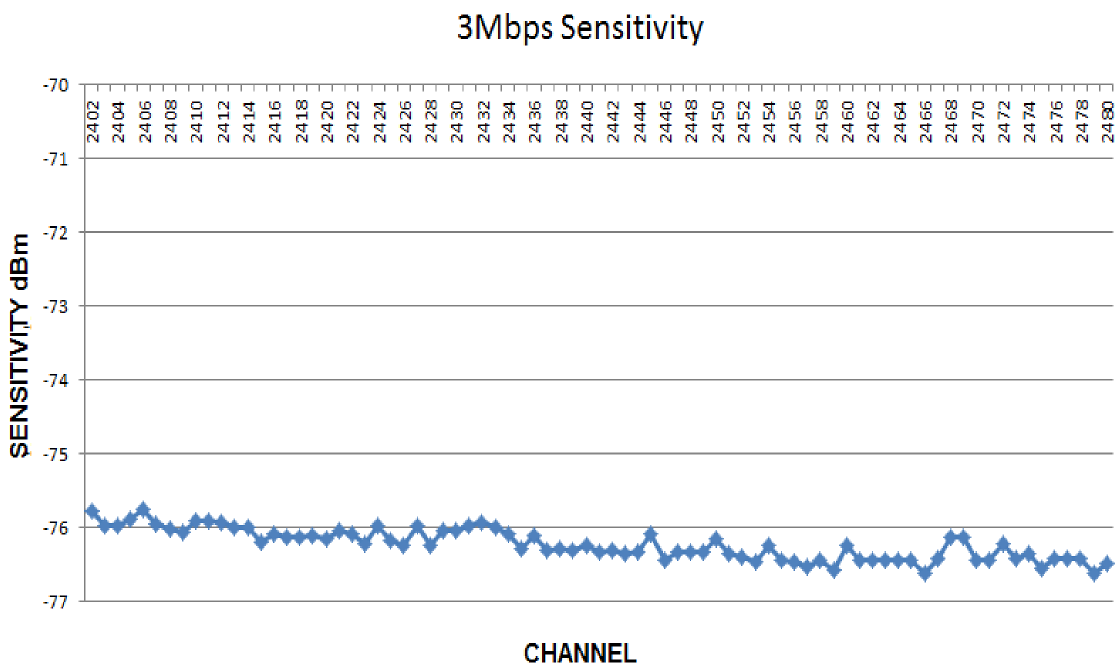

Figure 4.3: 3Mbps Sensitivity in Noise Free Environment

ceiver should not have this much fluctuations in sensitivity.

Now we can see in Figure 4.3 that Sensitivity result across whole BT 2GHz spectrum is almost constant.

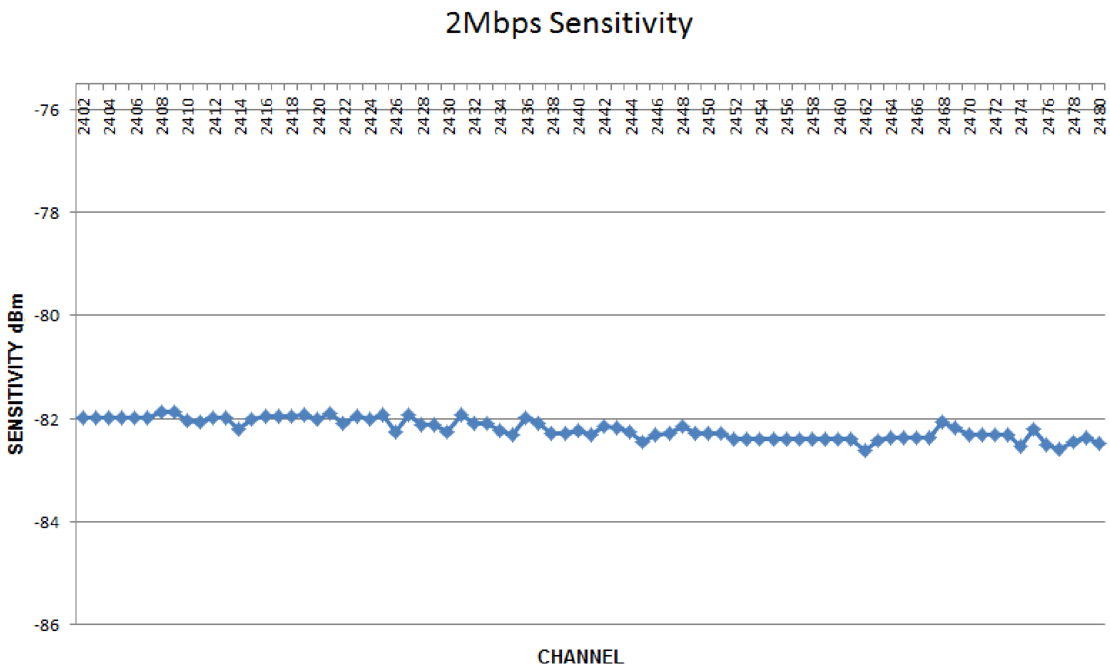

Figure 4.4: 2 Mbps Sensitivity in Noise Free Environment

The results in Figure 4.2, Figure 4.3 and Figure 4.4 are derived for the same DUT. So noisy environment can be misleading while measuring sensitivity of the receiver. It is good practice to make sure that the test environment is noise free for sensitivity test.

Other observation is the difference in overall sensitivity for different data rates i.e. for 2 Mbps and for 3 Mbps. From Figure 4.3 and Figure 4.4 we can see that sensitivity for 3 Mbps and for 2 Mbps is around -76 dBm and -82 dBm respectively.

4.2 For Wide Band Radios

4.2.1 Receiver Sensitivity Test

Receiver systems are normally required to process very small signals. The weak signals cannot be processed if the noise magnitude added by the receiver system is larger than that of the received signal. Increasing the desired signals amplitude is one method of raising the signal above the noise of the receiver system. Signal amplitude can be increased by raising the transmitters output power. Alternately, increasing the antenna aperture of the receiver, the transmitter, or both allows a stronger signal at the receiver input terminals. Increasing the physical size of an antenna is one method to increase its aperture [5].

Higher heat dissipation is typically required to increase transmitter output power. Cost, government regulations, and interference with other channels also limit the transmitter power available for a given application. Physically increasing transmitter antenna size may cause weight and wind load problems on the tower to which the antenna is mounted.

Increasing receiver antenna size obviously increases the housing size for a portable product enclosing the antenna structure, such as a cellular telephone or pager. Because raising the desired signal amplitude above the noise added by the receiver may not always be practical, a weak signal might be processed by lowering the added noise. In this case, the noise must be decreased such that the noise amplitude is somewhat below the weak signal amplitude. Depending on the RxPER the sensitivity of the receiver keeps on varying [6].

Receiver sensitivity is defined as minimum signal power required to detect the signal with Packet Error Rate (PER) less than or equal to 10%. This test is performed on receiver station. Receiver station will be having Reference machine and DUT

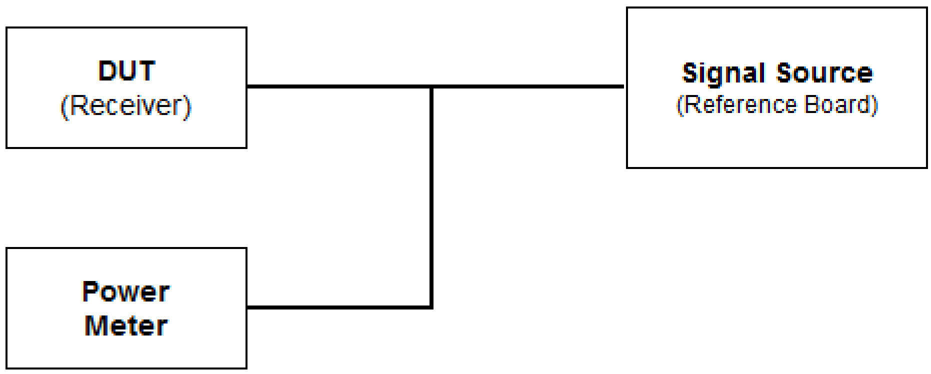

Figure 4.5: Setup for measuring Receiver Sensitivity

machine. Setup for measuring receiver sensitivity is shown in Figure 4.5. Reference is to transmit the signal with specific characteristics like power, rate, frequency, etc. and DUT machine will be having DUT connected to it to receive the transmitted signal from the Reference.

First the reference is initialized and ready to transmit packet and then DUT is

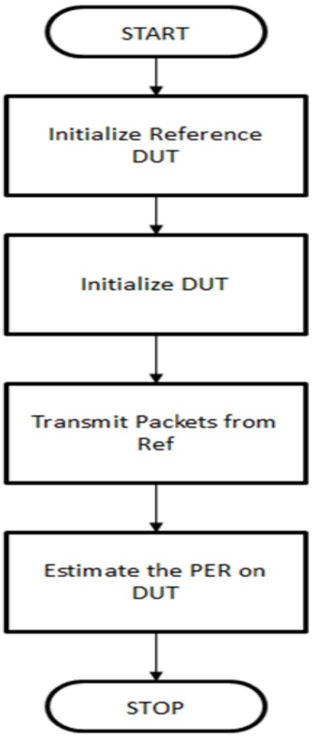

Figure 4.6: Steps for measuring receiver sensitivity

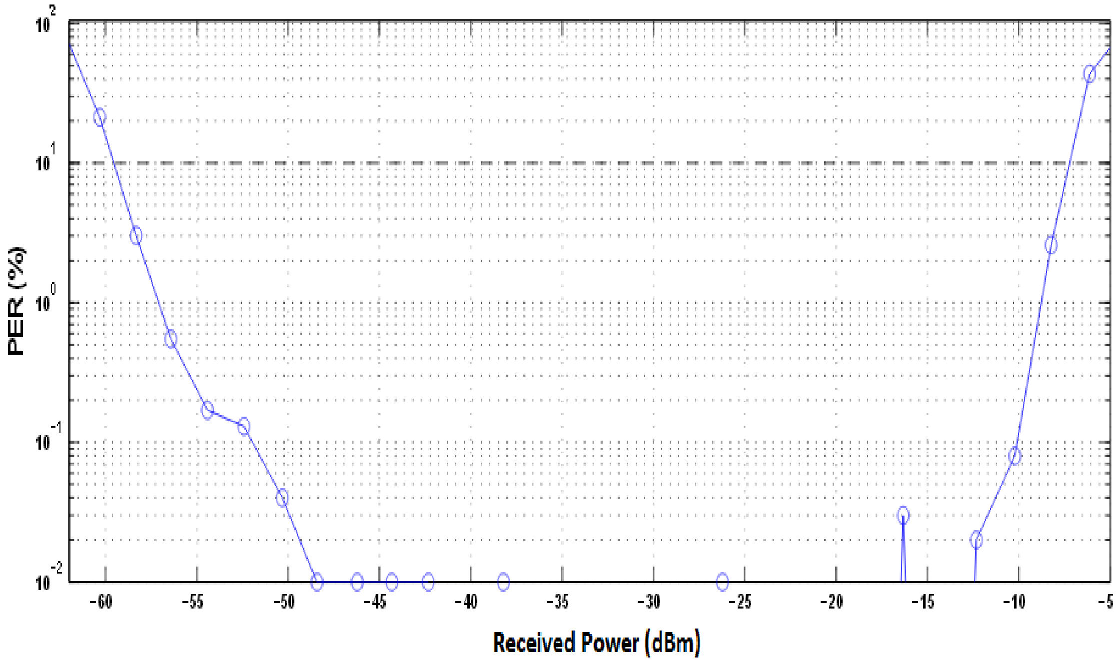

Figure 4.7: Packet Error Rate for Different Received Power Levels

initialized to receive the packet from reference DUT. These steps are shown by flowchart in Figure 4.6.

Reference transmits packets within a particular power range in discrete manner and at DUT side, for each power level PER will be calculated. Result of the test is a plot of PER Vs Received Power.

Receiver sensitivity is defined from the Plot of PER Vs Received Power. As we can see from Figure 4.7 that power corresponding to cross over point of actual curve and 10% PER threshold is sensitivity of the receiver.

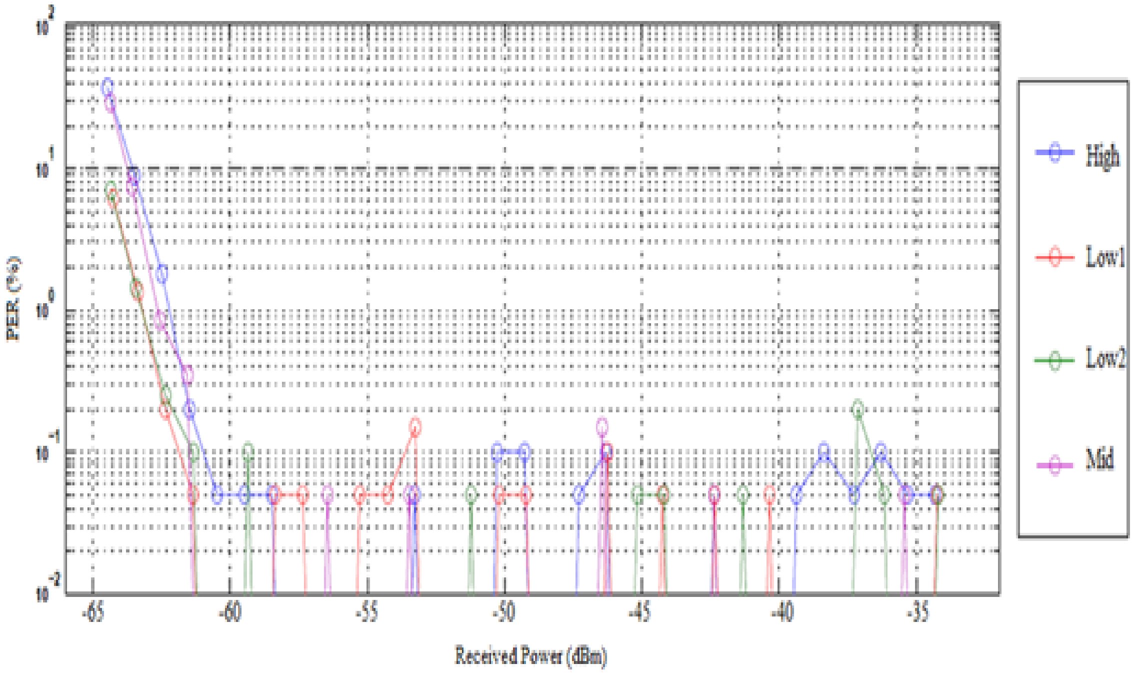

Figure 4.8: PER Vs Received Power for different WLAN channels

Due to fading, high frequency signal undergoes more attenuation with distance. As a result of that high frequency channel will be having less sensitivity as compared to Mid and Lower frequency channels. This can be observed from the Figure 4.8.

Chapter 5

Bluetooth Station Bring-up

5.1 Bluetooth Automation Test Setup (Station)

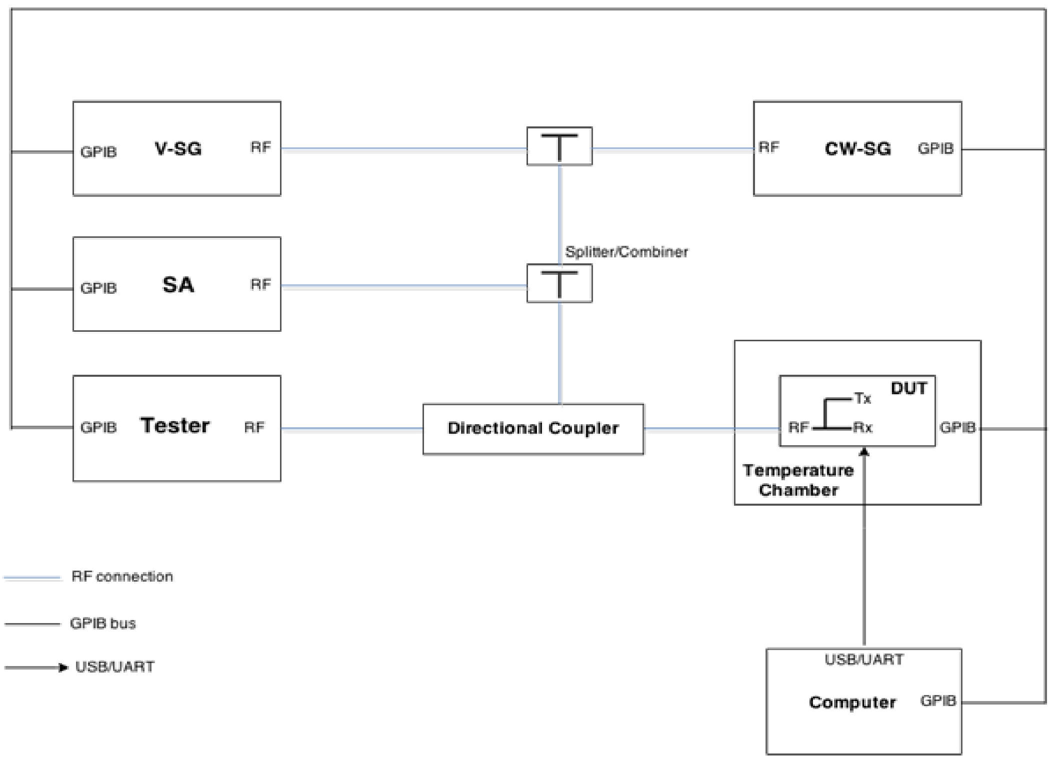

Figure 5.1: Automation test setup for Bluetooth test cases

The Figure 5.1 above shows typical automation test setup for Bluetooth test cases. Our DUT is kept under temperature chamber to test it under different temperature conditions. All the internal connections are RF connection and communication to all these devices for automation and fetching the result data is done through GPIB bus as shown in figure. Different test commands are sent from computer to DUT through USB/UART protocol.

5.2 Station Bring-up

Bringing up the station means to get the station up from non-connected devices to final working condition. The task is divided into three parts, Cabling, Calibration and Sanity Test.

Calibration is again divided into two parts, VNA calibration and then station calibration.

A network analyser has connectors on its front panel, but the measurements are seldom made at the front panel. Usually some test cables will connect from the front panel to the device under test (DUT). The length of those cables will introduce a time delay and corresponding phase shift (affecting VNA measurements); the cables will also introduce some attenuation (affecting SNA and VNA measurements). The same is true for cables and couplers inside the network analyser. All these factors will change with temperature. Calibration usually involves measuring known standards and using those measurements to compensate for these errors.

Station calibration is carried out by measuring S21/S12 parameters from DUT to all the instruments for getting the path loss.

We can take connections from DUT to any devices as a DUT block shown in

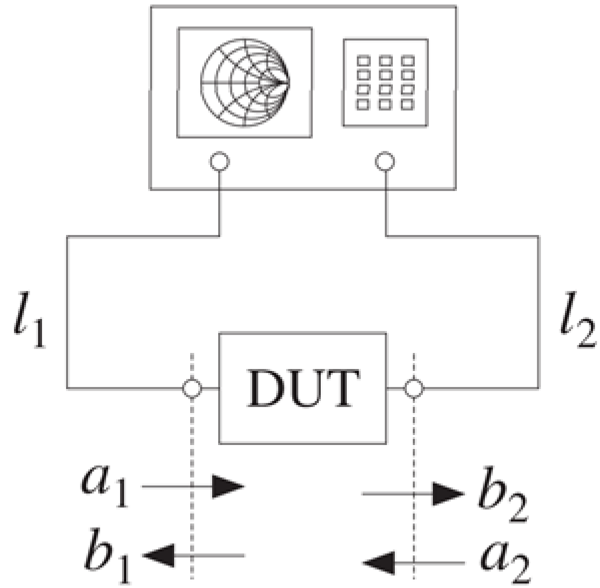

Figure 5.2: calibration setup

Figure 5.2. The obtained path losses measured for all the RF connections are then compensated once we get the test output.

5.3 Sanity Test

A sanity test or sanity check is a basic test to quickly evaluate whether a claim or the result of a calculation can possibly be true. It is mainly done to check the balance of the test results performed on all the stations under same cases and conditions.

Output power test is performed to measure output power at different frequencies at antenna out. This is the basic test used for sanity.

Figure 5.3 shows result of output power test performed on different test stations for BT frequency range. Variation in OP with varying frequency for station 06 is due to bad S11 seen from DUT side to the tester due to improper design of MUX box or improper connections. It can be improved by connecting suitable value attenuator

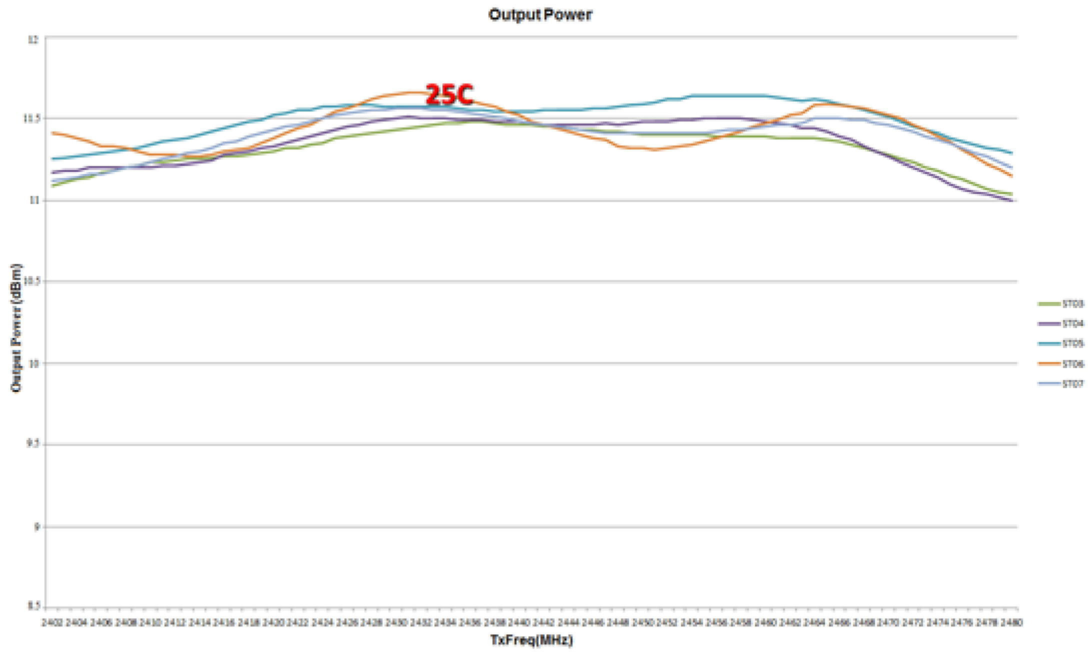

Figure 5.3: OP test result performed on different stations

at the antenna port of DUT. Improvement is shown in Figure 5.4.

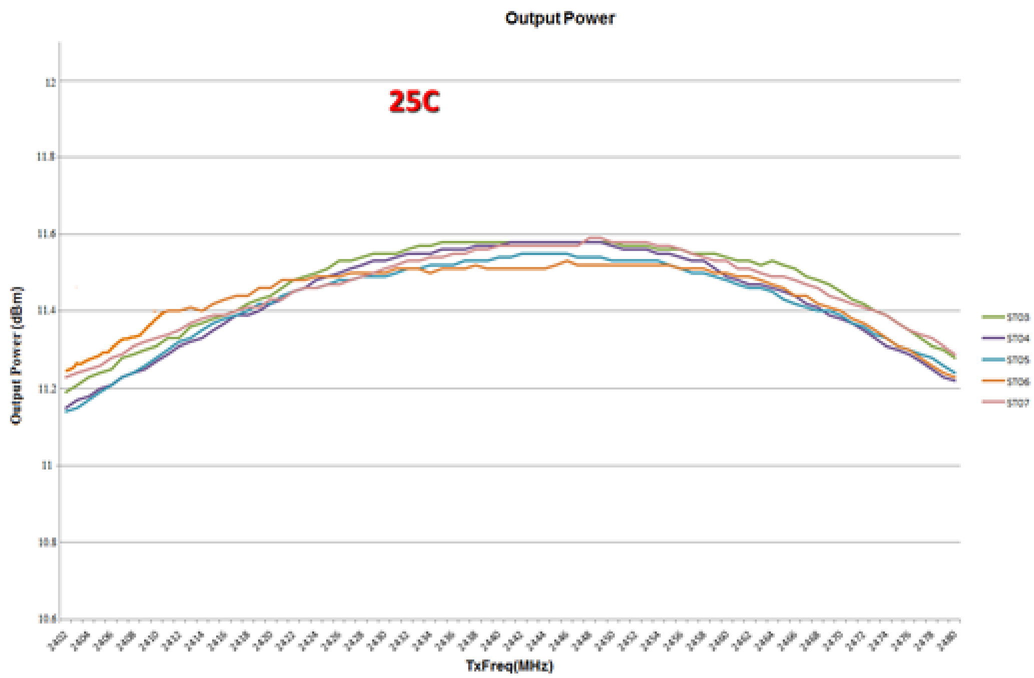

Figure 5.4: OP test result performed on different stations with 10dB attenuator

The above figure shows improvement in OP result for Station 06 and for all the other stations as well.

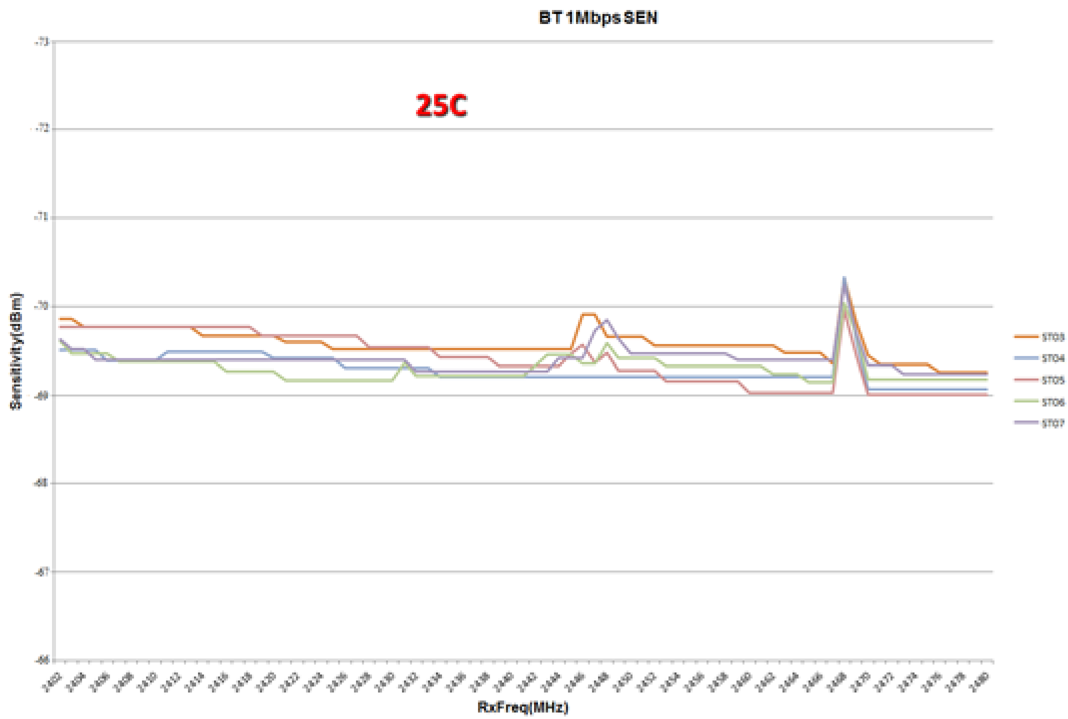

Figure 5.5: Sensitivity test result performed on different stations

The other test result which is preferred for sanity check is sensitivity. Figure 5.5 shows result of sensitivity test performed on different stations.

Chapter 6

Python Report Generator Automation Tool

6.1 Result File Structure

As we have seen, various tests performed on Bluetooth chip are automated. DUT and all the test instruments are controlled remotely by sending test command for DUT through USB/UART protocol and automation command for test instruments through GPIB bus. Corresponding test result is fetched through GPIB bus. Arranging this result data in a proper format and plotting it, is an important part to make result analysis easy.

All the raw result data is stored in an excel sheet. We make use of Microsoft excel to store, arrange and plot the result data to make the analysis easy. We have made different template files for different kind of test results. As discussed in Chapter 2 (subtopic 2.2.2) Bluetooth has different versions, Basic Rate (BR), Enhanced Data Rate (EDR), and Bluetooth Low Energy (BLE). Therefore, there are three different kinds of templates, one for BR, one for EDR and one for BLE. Each of these template files is having multiple sheets for storing results of various transmitter and receiver tests (for example, EDR Sensitivity in 4.1.1 and Output Power in 5.3) corresponding to particular Bluetooth version. Each sheet is dedicated for a particular test and will have test result only if that particular test is performed. Further, all these tests are performed for different DUTs on different test stations across different temperatures (low temperature, room temperature and high temperature).

Likewise, we will be having many such result files for different BT versions, different DUTs, different stations and different temperatures. These result files are distinguished by their name. Result files name structure is as follows:

BTVersion_*DUT*_TestStationName_*Temperature*_TimeStamp.xls

This much data will need summarization and comparison of a particular tests result for different DUTs and/or for different temperature.

6.2 Basic Function of Report Generator Tool

Report Generator takes multiple raw result files and processes those files to generate summary file. Single sheet in summary file will have data of a common BT version and common test compared for different DUTs and/or for different temperature.

It processes raw files sheet by sheet i.e. it takes data for the particular sheet (test) from all imported raw files and dumps it into a single sheet corresponding to that test and then moves to next sheet (test).

Data from different files will then be plotted for comparison purpose. Figure 6.1, 6.2 and 6.3 shows Output power results for DUT1, DUT2 and DUT3 respectively. It is very difficult to compare and conclude anything just by looking at each of these

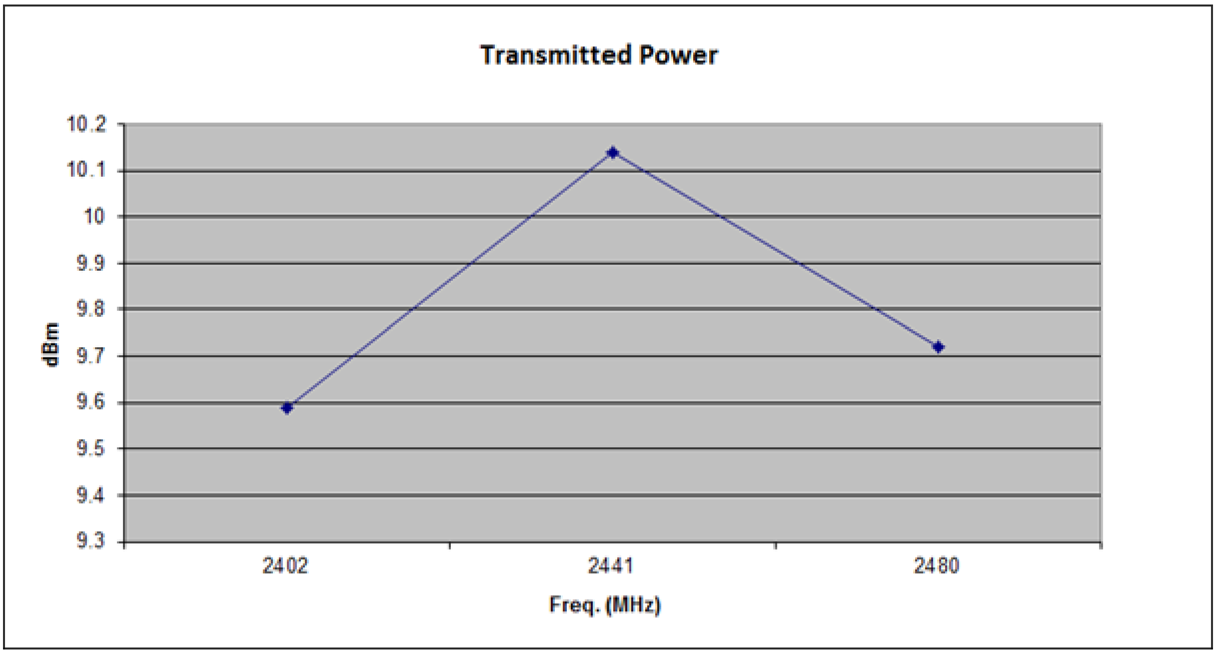

Figure 6.1: Output power result for DUT1

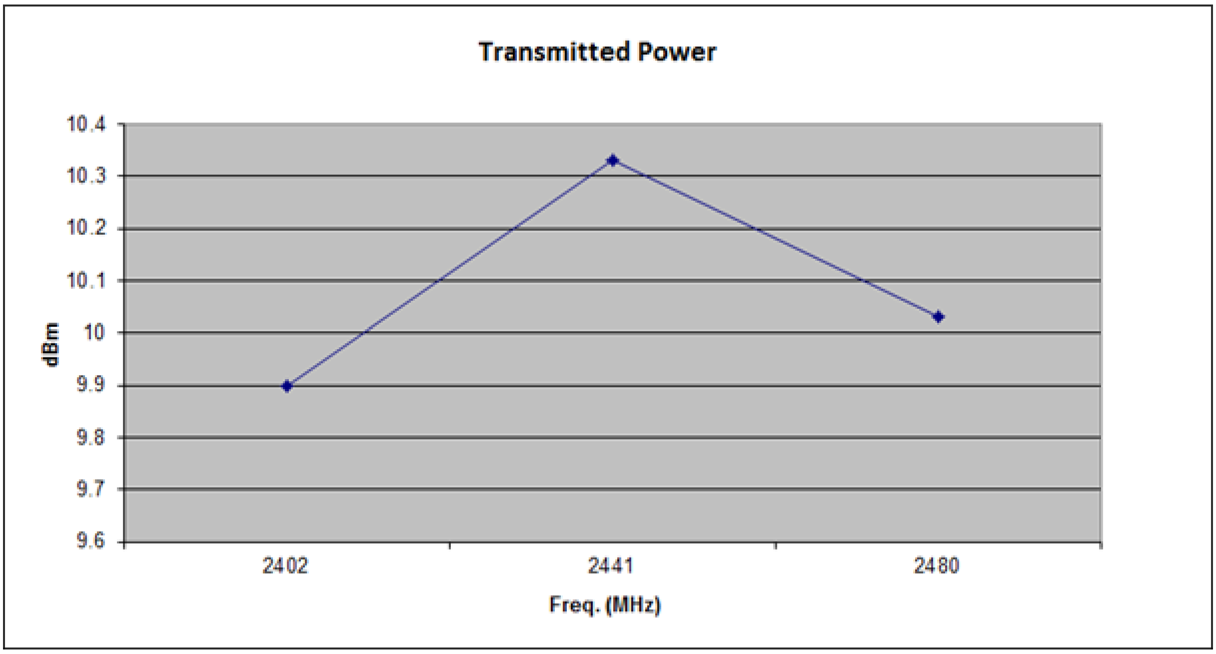

Figure 6.2: Output power result for DUT2

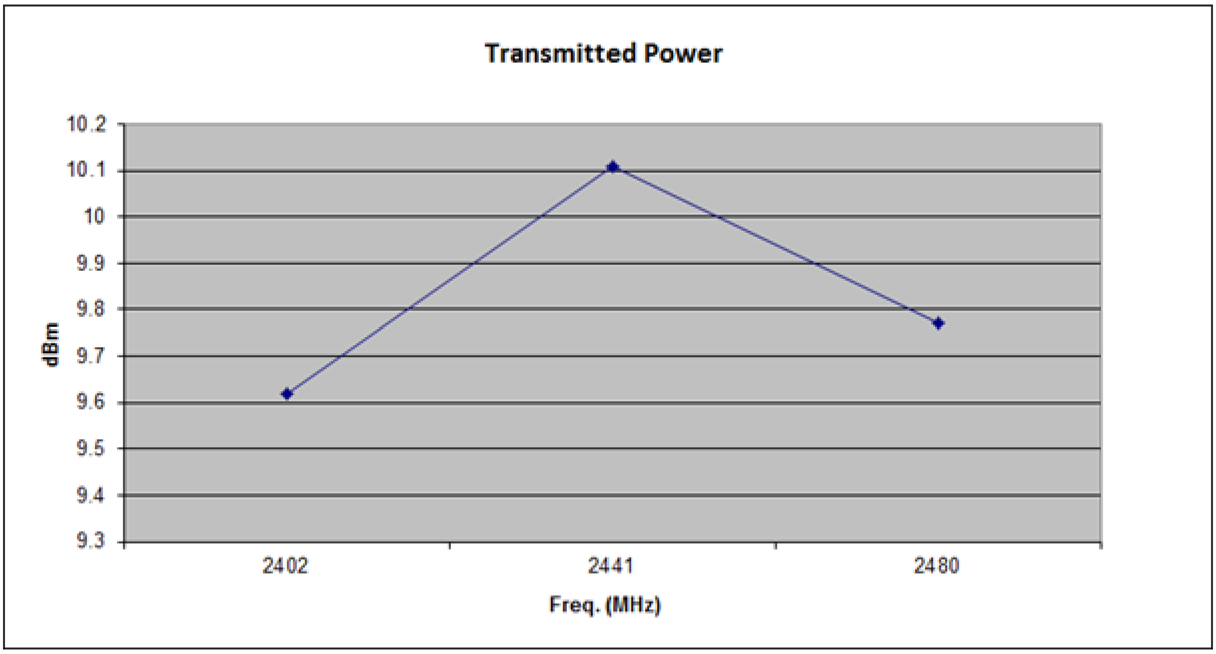

Figure 6.3: Output power result for DUT3

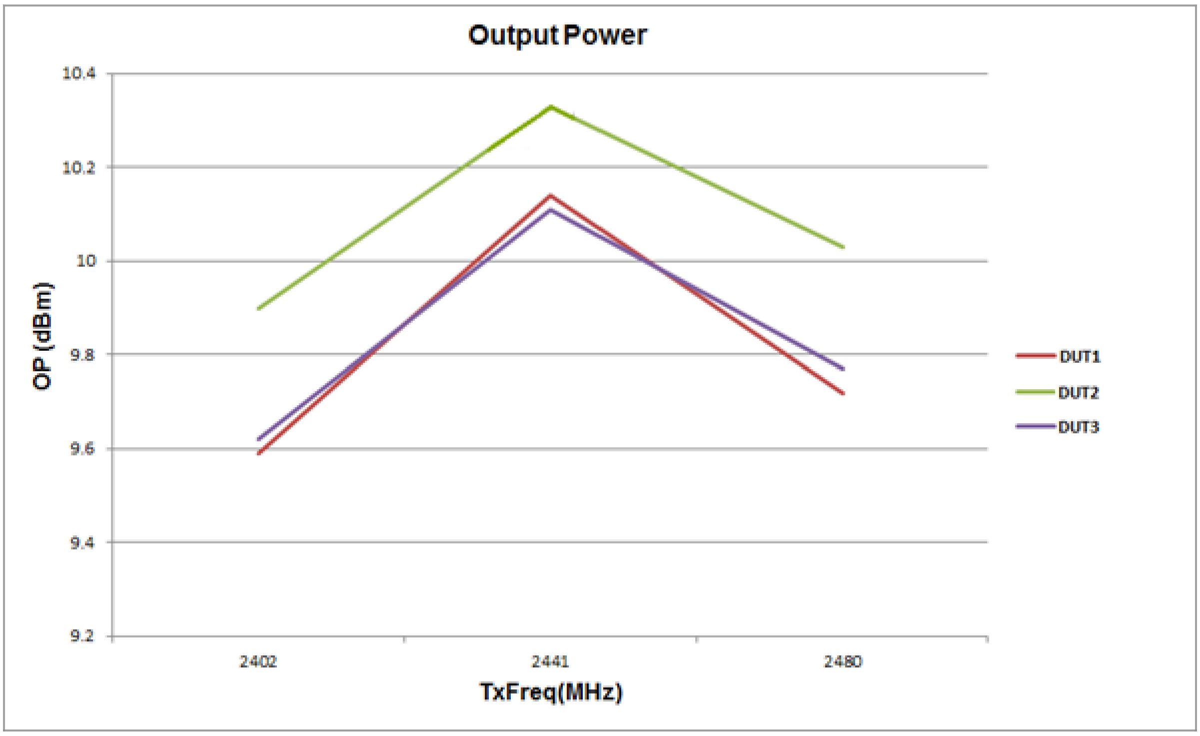

Figure 6.4: OP result for DUT1, DUT2, and DUT3

figures individually. In this scenario, report generator tool will help by taking data from each of those sheets and dumping into one sheet and plotting results for all the DUTs simultaneously. Figure 6.4 shows summary sheet plot which is output of report generator while importing raw files of DUT1, DUT2, and DUT3 for Output Power test.

6.3 Basic Code Architecture

Report Generator python code is organized in three python files, Driver.py, SummaryFileSheets.py, and SummaryFileTarget.py. Driver.py is at the top layer through to SummaryFileTarget.py, at the bottom layer. Pseudo codes of these files are shown below.

Pseudo code of Driver.py:

```
Useful Functions
Def BTReportGen(BR_FilesList , EDR_FilesList , BLE_FilesList):
    workbook object wb defined here
    IF (BR_FilesList is empty and EDR_FilesList is empty):
        pass
    ELSE:
        IF (BR_FilesList is not empty and EDR_FilesList is
        empty):
            testClassList for BR defined here
            FOR each testClass in the testClassList list:
                Get Attribute of SummaryFileSheets.testClass
                with wb into wx
                CALL wx.generateReport with BR_FilesList
            save wb
        ELIF (BR_FilesList is empty and EDR_FilesList is
```

```
        not−empty):
            testClassList for EDR defined here
            FOR each testClass in the testClassList list:
                Get Attribute of SummaryFileSheets.testClass
                with wb into wx
                CALL wx.generateReport with EDR_FilesList
            save wb
        ELSE:
            testClassList for BR defined here
            FOR each testClass in the testClassList list:
                Get Attribute of SummaryFileSheets.testClass
                with wb into wx
                CALL wx.generateReport with BR_FilesList
            testClassList for EDR defined here
            FOR each testClass in the testClassList list:
                Get Attribute of SummaryFileSheets.testClass
                with wb into wx
                CALL wx.generateReport with EDR_FilesList
            save wb
    IF BLE_FilesList is not−empty:
        testClassList for BLE defined here
        FOR each testClass in the testClassList list:
            Get Attribute of SummaryFileSheets.testClass with
            wb into wx
            CALL wx.generateReport with BLE\_FilesList
        save wb
```

Pseudo code of SummaryFileSheets.py:

```
Class common(ReportSheet):
```

```
Class testClass1(common):
Class testClass2(common):
Class testClass3(common):
Class testClass4(common):
.
.
.
Class testClassN(common):
```

Pseudo code of SummaryFileTarget.py:

```
Useful Functions
Class ReportSheet(object):
    Def __init__(currentsheet,workbook):
        Make currentsheet a global variable with name
        self.currentsheet
        Make workbook a global object/variable with name
        self.workbook
        Def DumpRawData
    Def getTempFromFileName
        Def GroupFilesAccordingToDut
        Def SortFilesofOneChip(templist,dutSet):
        FOR each temp in the templist list:
            FOR each dut in dutSet:
                CALL getTempFromFileName with dut RETURNING
                temperature
                IF temp is equal to temperature:
                    Append dut to NewdutSet
                    Break
        Return NewdutSet
```

```
Def ExtractDataFromOneDut(DutSet, currentsheet,
templist):
CALL SortFilesofOneChip with templist DutSet
RETURNING NewDutSet
FOR each DutFile in NewDutSet:
    CALL ReadRawDataFromWorkbook with DutFile
    RETURNING result
    Return result
Def ReadRawDataFromWorkbook
Def generateReport(self,RawFilesList):
Assign self.currentsheet to currentsheet to use it
for the scope of current function
Assign self.workbook to workbook to use it for the
scope of current function
CALL getTempFromFileName with RawFilesList
RETURNING templist
        CALL GroupFilesAccordingToDut with RawFilesList
        RETURNING GroupedFileSet
        FOR each filesofoneDut in GroupedFileSet:
    CALL ExtractDataFromOneDut with filesofoneDut,
    currentsheet templist RETURNING OneDutData
    Append OneDutData to AllDutDataList
CALL DumpRawData with AllDutDataList, currentsheet
workbook
Process on data written into currentsheet of workbook
for plotting
```

Driver.py grabs the raw result files added into report generator GUI in 'BR_FilesList', 'EDR_FilesList', and 'BLE_FilesList'. Later on it will check these lists. If a particular list is not empty i.e. it has result files in it, code will pass control to bottom level files, SummaryFileSheets.py and SummaryFileTarget.py. Processing of raw result files is done by SummaryFileTarget.py and SummaryFileSheets.py. It loops on 'testClassList' and switches between different test classes and turn by turn sends corresponding BT version raw result files to bottom layer files for processing for corresponding test class.

SummaryFileSheets.py defines all BR, EDR and BLE test classes. These classes have functions for plotting the data. Class 'common' is the parent class of all the test classes. In this class, some useful functions are defined which are common to all the test classes. Class 'common' is itself a child class of 'ReportSheet' class, which is defined in SummaryFileSheets.py.

SummaryFileTarget.py is the main engine to grab the raw data for each test class. Function 'generateReport' is called from Driver.py with list of raw files to be processed. Function 'GroupFilesAccordingToDut' groups the raw files according to DUT name such that it will be having list of lists and all the lists inside that outer list will be having files of a common DUT only. E.g. [[dut1_v1,dut1_v2,...], [dut2_v1,dut2_v2,],], where v1, v2,... can any combination of station name, temperature and time stamp. So raw result files will be processed DUT by DUT for the current test class i.e. for example, it will first process raw files of dut1 then dut2 and so on. Further, it will grab data from every single same DUT raw file present in 'NewDutSet' list turn by turn [function 'ExtractDataFromOneDut'] and appends it to 'AllDutDataList'. Data in 'AllDutDataList' will then be dumped into current sheet ('currentsheet') and processed to get plot.

6.4 Problem Statement and Motivation for Code Enhancements

As we have seen so far, on a broader prospective that Report Generator takes raw result files and processes these files by taking files of one chip at a time. But, it is not able to process all the files from a group of files which has files having same DUT name and same temperature in their name fields. It will simply omit all the other files amongst this kind of files by keeping one file which comes first in order by its name. For example, let's suppose we have 'GroupedFileSet' is equal to [[DUT1_Temp1_TimeStamp1, DUT1_Temp1_TimeStamp2, DUT1_Temp1_TimeStamp3, DUT1_Temp2_TimeStamp, DUT1_Temp3_TimeStamp], [DUT2_Temp1_TimeStamp, DUT2_Temp2_TimeStamp, DUT2_Temp3_ TimeStamp], [DUT3_Temp1_TimeStamp, DUT3_Temp2_TimeStamp, DUT3_Temp3_TimeStamp],...]. Code will omit DUT1_Temp1_TimeStamp2 and DUT1_Temp1_TimeStamp3 named files from that DUT list. This is due to 'SortFilesofOneChip' function called from 'ExtractDataFromOneDut' function. Only one file, which comes first in 'dutSet' list for a given temperature will be appended to 'NewdutSet' list and files in this list will then be processed by ExtractDataFromOneDut function. This will be solved if we comment out 'break' statement in 'SortFilesofOneChip' function. But only commenting 'break' statement will not help, there are many other modifications required to be done for getting files which are having same DUT name and same temperature to be processed. Those modifications, we will see in the following topics. By doing those modifications, we will be able to have data from all the raw files of that DUT in summary file.

A single result file will be having test results for all the test cases related to its BT version. Now, in case of test crash, we attempt to rerun tests by omitting executed tests from the automation tool. This generates result files which are having same DUT name and same temperature number in their name fields. So, ideally there should be only one result file, which has all test case data for a particular DUT and temperature. If we want to merge these multiple crash files into a single file, we have to do it manually by copy-pasting as the modified code of report generator for getting the summary from all the files will generate summary file, which will be having data in different format and might have data from all the other DUT files if we have imported it in Report Generator GUI. So there is a need to add functionality to generate merged file from the result files which are having same DUT name and same temperature in their name fields.

6.5 Issues faced and their workaround for summary file

As discussed in previous topic, only modifying SortFilesofOneChip function will not solve the whole problem. We have seen in topic 6.1 that we have predefined templates to store test results. The template is saved with different name by the naming convention that we have seen in topic 6.1 while running automatic test.

By default some of the test sheets inside the template file will be having raw data in it. Due to existing data in some of the sheets in default template, it will cause the plot to have an unnecessary blank space.

By comparing Figure 6.4 and Figure 6.5 is it clearly seen that Figure 6.5 is having unusual plot. This is because sheet for output power test in template file is not having OP readings for 2402, 2441 and 2480 MHz channels at all and still the sheet is having a 3x1 column filled with channel numbers.

So actually this sheet is empty. There is no valid data in it. But due to that filled 3x1 column in the sheet, code is considering it as a valid data and takes it

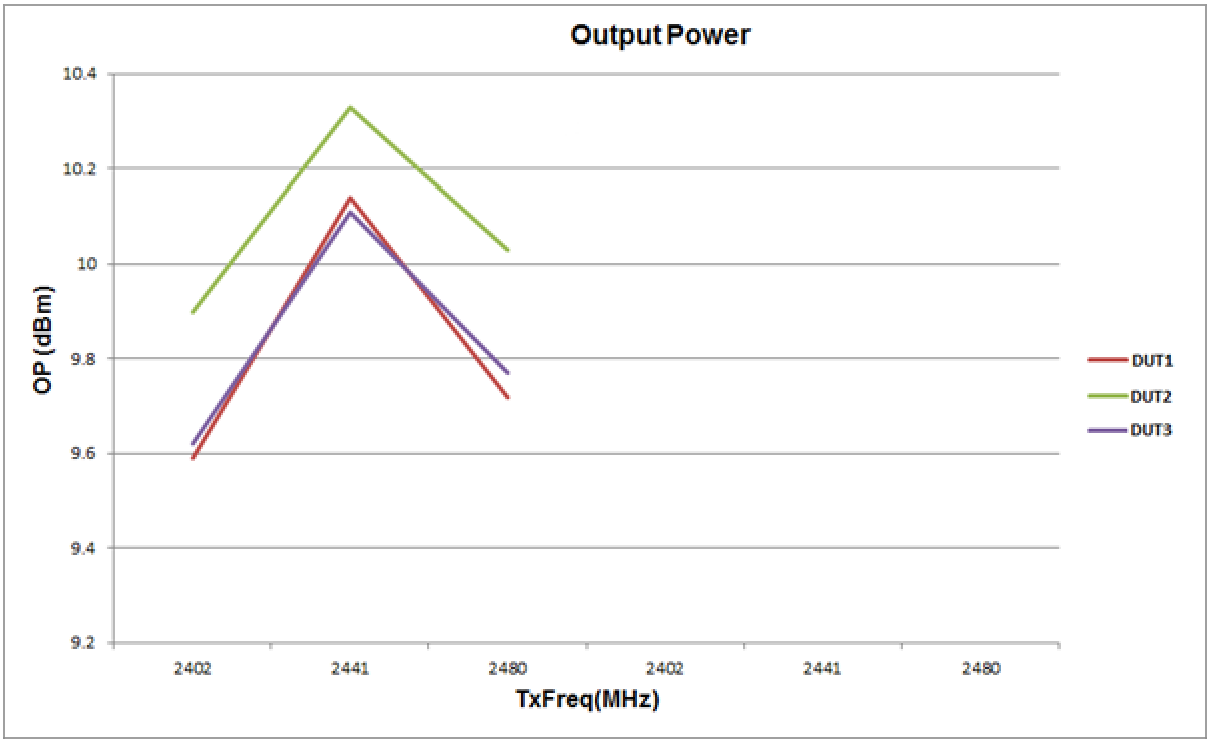

Figure 6.5: First case of improper plot after modifying code

to process. Instead of considering it as a valid data, code should recognize the sheet as an empty sheet due to no OP readings for low, mid, and high channels. Workaround for this problem is comparing such sheets with corresponding template file's sheets. If both the sheets are matching then it will be considered as an empty sheet, else the sheet will be considered as a filled sheet. Therefore, exceptions for the sheets which are having raw data in the default template is put in 'ReadRawDataFromWorkbook' function in SummaryFileTarget.py file. Whenever such sheets are encountered, code will compare the sheet with corresponding template file sheet and then decide whether the sheet is empty or not.

Now let's suppose we have result files for two DUTs, DUT1 and DUT2 and group of files for DUT1 and for DUT2, both are having files of same temperature number. All files having same temperature for DUT1 are having data for a particular test sheet whereas, for DUT2 not all such files are having data for that particular sheet. This case will make the code taking random data for the sheets in which we don't have data. This will result in a random unusual plot as shown in Figure 6.6.

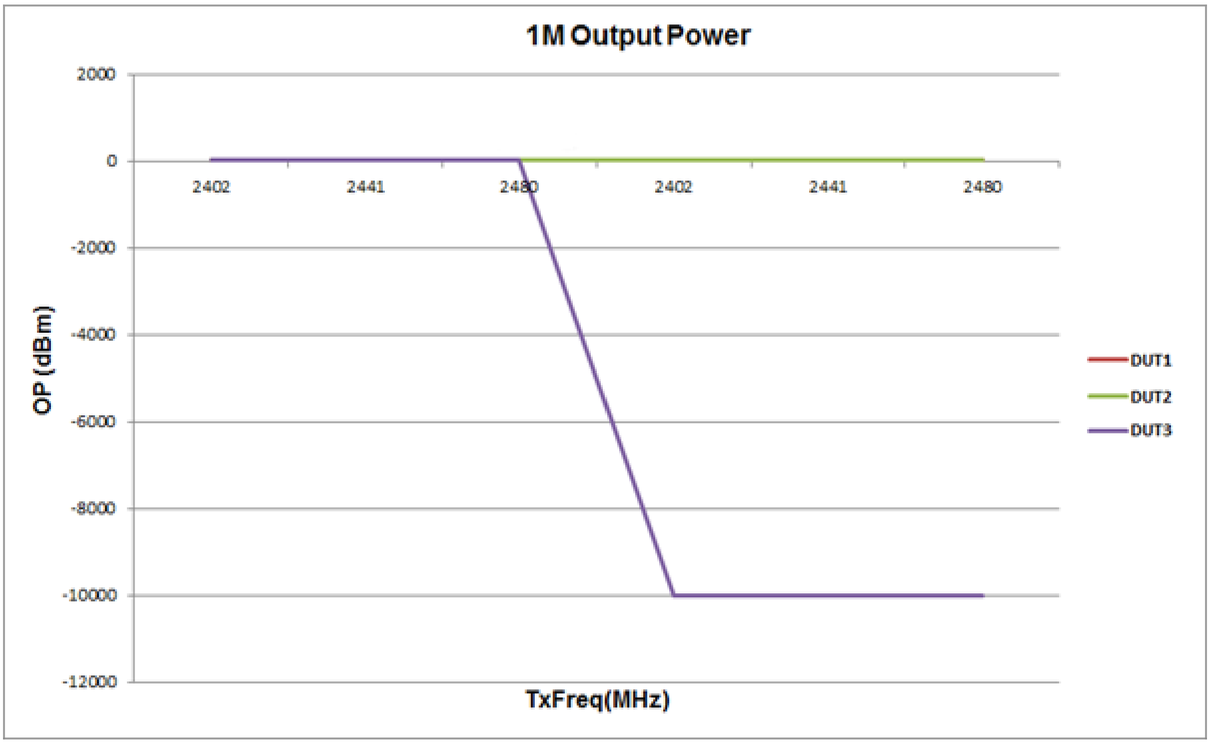

Figure 6.6: Second case of improper plot after modifying code

This is happened due to random data taken for the DUT3. All files of DUT1 and DUT2 are having data in their output power sheets, but one of the files for DUT3 is not having data in the sheet for output power.

This issue can be solved by taking only one file's data and omitting other files data, as having multiple files of data for a particular test case for the same DUT and same temperature is of no use. Code inside 'ExtractDataFromOneDut' function in SummaryFileTarget.py is modified to take data of the file which comes first when shorted by name amongst 'DutSetofOneTemp'. One more function is defined to achieve the same task, which is 'GroupChipSetAccordingtoTemp'. This function parts 'NewDutSet' into three groups. Each group will be having result files of one temperature only. Processing DUT set, temperature by temperature is required because we want data from the file which comes first is common temperature DUT Set.

All the modifications which we have discussed for summary file are shown in modified pseudo codes in topic 6.7.

6.6 Code modifications for generating merged file

As we have seen, generating merged file out of multiple files having same DUT name and same temperature is necessary. To achieve the same, two new functions are defined in SummaryFileTarget.py file. Function 'getFilesTobeMerged' returns 'DutSetAccordingtotemp' into which, files are separated in terms of temperature. Files will be there inside the 'DutSetAccordingtotemp' list for a particular temperature, if we have more than one files for that temperature, else there will be 'None' instead. Structure of 'DutSetAccordingtotemp' is as given [[DUT1_temp1_TimeStemp1, DUT1_temp1_TimeStemp2], [DUT1_temp2_TimeStemp1, DUT1_temp2_TimeStemp2], None].

Function 'MergeFiles' takes each list in 'DutSetAccordingtotemp' list for processing. Each file in list of 'DutSetAccordingtotemp' list i.e. each file of that DUT and temperature will be taken turn by turn to fetch data from it. It will write data into merged file only if current sheet is not empty in raw result file and if corresponding current sheet is empty in merged file. To grab the data from raw result files to put into merged files, function 'ReadRawDataFromWorkbookforMerging' is defined. This function also checks for empty or filled sheets by using the same method of comparing which we have seen in topic 6.5. These two functions drive whole merging thing and calls other useful functions like 'ReadRawDataFromWorkbookforMerging' for merging operation. These two functions i.e. merging operation will be bypassed if we don't have files of same DUT name and same temperature for a particular DUT.

Saving merged files after having data in it is a crucial part. 'generateReport' function is called sheet by sheet, so program control will come out of this function every time processing for currentsheet is done. Now if we define workbook object for merge file here then we have to save the workbook before program control comes out of this function, as scope of the workbook object will be limited to this function only. Saving merge file after processing for every sheet will consume more time and might cause the excel to work improperly. This is the main reason why 'MergeFiles_wb_List' is defined in Driver.py file. 'MergeFiles_wb_List' is a list of lists. Each list inside this list is dedicated for a single DUT files. Function 'GetwbObjListforMergeFiles' inside Driver.py file takes one 'DutSet' at a time. It checks for no. of files having same temperature greater than one. If it is greater than one then it will 'save as' corresponding BT version template file to the location where raw result files are kept and renames it as the naming convention which we have seen in topic 6.1 by using same DUT name same temperature and appends workbook object of the copied file into 'MergeFiles_wb_List'. If we do not have files having same temperature and same DUT name more than one then 'None' will be appended to the 'MergeFiles_wb_List' list. If we do not have multiple same temperature files at all then 'GetwbObjListforMergeFiles' function will simply return 'None'. Structure of 'MergeFiles_wb_List' is shown: [DUT1_Temp1, DUT1_Temp2, DUT1_Temp3], [DUT2_Temp1, DUT2_Temp2, DUT2_Temp3], None, [DUT4_Temp1, None, None],...].

'MergeFiles_wb_List' for merge file is passed from Driver.py through SummaryFileSheets.py to SummaryFileTarget.py along with 'wb' for summary file. As discussed saving of workbook objects inside 'MergeFiles_wb_List' list will be done in Driver.py file.

All the modifications which we have discussed for generating merge file are shown in modified pseudo codes in topic 6.7.

6.7 Modified Pseudo Codes of Report Generator

All the modifications/additions which are discussed are shown below.

6.7.1 Modified pseudo code of Driver.py

```
Useful Functions defined here
Def GroupFilesAccordingToChip
Def GetwbObjListforMergeFiles(DutSet):
        IF not more than one same temp file:
                Return None
        ELSE:
            IF no. of files having temp1 is greater than one:
                Check for BT version of the files in DutSet
                Save As the corresponding BT version template
                file to raw files  directory
                Append workbook object to MergeFiles_wb_List
            ELSE:
            Append None to MergeFiles_wb_List
                IF no. of files having temp2 is greater than one:
                    Check for BT version of the files in DutSet
                    Save As the corresponding BT version template
                    file to raw files  directory
                    Append workbook object to MergeFiles_wb_List
                ELSE:
            Append None to MergeFiles_wb_List
            IF no. of files having temp3 is greater than one:
```

```
                Check for BT version of the files in DutSet
                Save As the corresponding BT version template
                file to raw files  directory
                Append workbook object to MergeFiles_wb_List
            ELSE:
                Append None to MergeFiles_wb_List
        Return MergeFiles_wb_List
Def BTReportGen(BR_FilesList, EDR_FilesList, BLE_FilesList):
    workbook object wb defined here
    IF (BR_FilesList is empty and EDR_FilesList is emoty):
        pass
    ELSE:
        IF (BR_FilesList is not empty and EDR_FilesList is
        empty):
            CALL GroupFilesAccordingToChip with BR_FilesList
            RETURNING GroupedFileSet
            FOR  each filesofoneDut in GroupedFileSet:
                Append workbook object list RETURNED by CALLING
                GetwbObjListforMergeFiles with filesofoneDut
                into MergeFiles_wb_List
            testClassList for BR defined here
            FOR each testClass in the testClassList list:
                Get Attribute of SummaryFileSheets.testClass
                with wb & MergeFiles_wb_List into wx
                CALL wx.generateReport with BR_FilesList
            save wb
            save MergeFiles_wb_List
        ELIF (BR_FilesList is empty and EDR_FilesList is
        not-empty):
```

```
          CALL GroupFilesAccordingToChip with EDR_FilesList
          RETURNING GroupedFileSet
          FOR each filesofoneDut in GroupedFileSet:
          Append workbook object list RETURNED by CALLING
          GetwbObjListforMergeFiles with filesofoneDut into
          MergeFiles_wb_List
      testClassList for EDR defined here
      FOR each testClass in the testClassList list:
          Get Attribute of SummaryFileSheets.testClass with
          wb & MergeFiles_wb_List into wx
          CALL wx.generateReport with EDR_FilesList
      save wb
      save MergeFiles_wb_List
      ELSE:
      CALL GroupFilesAccordingToChip with BR_FilesList
      RETURNING GroupedFileSet
      FOR each filesofoneDut in GroupedFileSet:
          Append workbook object list RETURNED by CALLING
          GetwbObjListforMergeFiles with filesofoneDut into
          MergeFiles_wb_List
      testClassList for BR defined here
      FOR each testClass in the testClassList list:
          Get Attribute of SummaryFileSheets.testClass with
          wb & MergeFiles_wb_List into wx
          CALL wx.generateReport with BR_FilesList
      CALL GroupFilesAccordingToChip with EDR_FilesList
      RETURNING GroupedFileSet
      FOR each filesofoneDut in GroupedFileSet:
          Append workbook object list RETURNED by CALLING
```

```
            GetwbObjListforMergeFiles with filesofoneDut into
            MergeFiles_wb_List
        testClassList for EDR defined here
        FOR each testClass in the testClassList list:
            Get Attribute of SummaryFileSheets.testClass with
            wb & MergeFiles_wb_List into wx
            CALL wx.generateReport with EDR_FilesList
                save wb
        save MergeFiles_wb_List
    IF BLE_FilesList is not-empty:
        CALL GroupFilesAccordingToChip with BLE_FilesList
        RETURNING GroupedFileSet
        FOR  each filesofoneDut in GroupedFileSet:
            Append workbook object list RETURNED by CALLING
            GetwbObjListforMergeFiles with filesofoneDut into
            MergeFiles_wb_List
    testClassList for BLE defined here
    FOR each testClass in the testClassList list:
        Get Attribute of SummaryFileSheets.testClass with
        wb & MergeFiles_wb_List into wx
        CALL wx.generateReport with BLE_FilesList
    save wb
    save MergeFiles_wb_List
```

6.7.2 Modified pseudo code of SummaryFileTarget.py

```
Useful Functions defined here
Class ReportSheet(object):
    Def __init__(currentsheet,workbook):
        Make currentsheet a global variable with names
```

```
        self.currentsheet
        Make workbook a global object/variable with name
        self.workbook
        Def DumpRawData
    Def getTempFromFileName
        Def GroupFilesAccordingToDut
        Def SortFilesofOneChip(templist,dutSet):
        FOR each temp in the templist list:
            FOR each dut in dutSet:
                CALL getTempFromFileName with dut RETURNING
                temperature
                IF temp is equal to temperature:
                    Append dut to NewdutSet
                    # Break
        Return NewdutSet
        Def getFilesTobeMerged(OneDutFileSet):
                IF no. of files having temp1 is greater than one:
                        Append files having temp1 to
                        DutSetAccordingtotemp
                ELSE:
                        Append None to DutSetAccordingtotemp
                IF no. of files having temp2 is greater than one:
                        Append files having temp2 to
                        DutSetAccordingtotemp
                ELSE:
                        Append None to DutSetAccordingtotemp
                IF no. of files having temp3 is greater than one:
                        Append files having temp3 to
                        DutSetAccordingtotemp
```

```
        ELSE:
                Append None to DutSetAccordingtotemp
        Return DutSetAccordingtotemp
Def MergeFiles(MergeFiles_wb_List, DutSetAccordingtotemp,
currentsheet):
        FOR each DutSetofOneTemp in DutSetAccordingtotemp:
        IF DutSetofOneTemp is not equal to None:
                FOR each DutFile in DutSetofOneTemp:
                Pull data from currentsheet of DutFile by
                CALLing ReadRawDataFromWorkbookforMerging
                and store into result1
                IF currentsheet of DutFile is having data
                (result1 != None):
                IF currentsheet of MergeFile is empty:
                    Write result1 to currentsheet of
                    MergeFile
Def GroupChipSetAccordingtoTemp(DutSet):
Def ExtractDataFromOneDut(DutSet, currentsheet, templist):
CALL SortFilesofOneChip with templist & DutSet RETURNING
NewDutSet
CALL GroupDutSetAccordingtoTemp with NewDutSet RETURNING
GroupedDutSetAcctoTemp
FOR each DutSetofOneTemp in GroupedDutSetAcctoTemp list:
    IF DutSetofOneTemp is not empty:
        FOR each DutFile in DutSetofOneTemp:
            CALL ReadRawDataFromWorkbook with DutFile
            RETURNING result2
            Append result2 to resultSetofOneTemp
        Append first file data amongst all files in
```

```
            DutSetofOneTemp to result
    Return result
    Def ReadRawDataFromWorkbook
Def ReadRawDataFromWorkbookforMerging
    Def generateReport(self,RawFilesList):
    Assign self.currentsheet to currentsheet to use it
    for the scope of current function
    Assign self.workbook to workbook to use it for the
    scope of current function
    CALL getTempFromFileName with RawFilesList RETURNING
    templist
    CALL GroupFilesAccordingToDut with RawFilesList RETURNING
    GroupedFileSet
    FOR each filesofoneDut in GroupedFileSet:
        IF MergeFiles_wb_List is not equal to None:
            CALL getFilesTobeMerged with filesofoneDut
            RETURNING DutSetAccordingtotemp
            CALL MergeFiles with MergeFiles_wb_List,
            DutSetAccordingtotemp & currentsheet
        CALL ExtractDataFromOneDut with filesofoneDut,
        currentsheet & templist RETURNING OneDutData
        Append OneDutData to AllDutDataList
    CALL DumpRawData with AllDutDataList, currentsheet
    & workbook
    Process on data written into currentsheet of workbook
    for plotting
```

Chapter 7

Conclusion and Future Scope

Nonlinearity of power amplifier, high noise figure of Low Noise Amplifier, filter non-idealities, and many other impairments affect the ideal behavior of the design and degrades the desired performance. RF Front End characterization helps to evaluate how and at what extent these imperilments affect the design and help to improve the design accordingly.

In the RF Front End characterization, automated test stations play an important part to evaluate different performance parameters of the design. Calibration and sanity test are the essential parts in the process of bringing these stations up.

In future, we focus on improving test techniques and setups for performing tests wherever it is possible. Constant improvement in the design is also desired. Apart from performance parameters which are discussed we have its other side, power consumption. In todays era, power consumption is the most essential and focused aspect for any design. Different practical issues are faced while chip is in idle or listen or in transmitter mode. We focus on finding those practical issues and their possible solutions.

Bibliography

[1] Arya Behzad, ”Wireless LAN Radios, System definition to Transistor design”, IEEE Press, 445 Hoes Lane, Piscataway, NJ 088854, 2008.

[2] Matt Loy (1999, May), Understanding and Enhancing Sensitivity in Receivers for Wireless Applications [Online]. Available: http://www.ti.com/lit/an/swra030/swra030.pdf

[3] Bob Pearson, ”Complementary Code Keying Made Simple”, May 2000.

[4] Bluetooth SIG, Inc (2015). Bluetooth Basics [Online]. Available: http//www.bluetooth.com/Pages/Basics.aspx

[5] Bluetooth SIG, Inc (2015). Press Releases: Updated Bluetooth 4.1 Extends the Foundation of Bluetooth Technology for the Internet of Things [Online]. Available: http://www.bluetooth.com/Pages/Press-Releases-Detail.aspx?ItemID=197

[6] Bluetooth SIG, Inc (2015). Basic Rate/Enhanced Data Rate (BR/EDR) [Online]. Available: https://developer.bluetooth.org/TechnologyOverview/Pages/BREDR.aspx

[7] Bluetooth Low Energy: Broadcom Presentation.

[8] Ian Poole, Bluetooth radio interface, modulation, and channels [Online]. Available: http://www.radio-electronics.com/info/wireless/bluetooth/radio-interface-modulation.php

[9] Sensitivity Test [Online]. Available: http://rfmw.em.keysight.com/sensitivity/test.htm

[10] Bluetooth Manufacturing Test, A Guide to Getting Started [Online]. Available: http://literature.cdn.keysight.com/litweb/pdf/5988-5412EN.pdf