

**AUTOMATED TESTING
OF
WIRELESS LINK STANDARDS**

by

Ajay Ogirala

B.Tech, Jawaharlal Nehru Technological University, 2006

Submitted to the Graduate Faculty of
School of Engineering in partial fulfillment
of the requirements for the degree of
Master of Science in Electrical Engineering

University of Pittsburgh

2007

UNIVERSITY OF PITTSBURGH
SCHOOL OF ENGINEERING

This thesis was presented

by

Ajay Ogirala

It was defended on

July 23rd, 2007

and approved by

Dr. William Stanchina, Chairman and Professor, Department of Electrical and Computer
Engineering

Dr. J.T. (Tom) Cain, Professor, Department of Electrical and Computer Engineering

Dr. Ronald G. Hoelzeman, Associate Professor, Department of Electrical and Computer
Engineering

Thesis Advisor: Marlin H. Mickle, Professor, Department of Electrical and Computer
Engineering

Copyright © by Ajay Ogirala
2007

AUTOMATED TESTING OF WIRELESS LINK STANDARDS

Ajay Ogirala, M.S

University of Pittsburgh, 2007

With the increase of the wireless market, there is a trend to equip a hand held wireless device with all the features of a desktop. The wireless devices are becoming more and more integrated into our everyday lives. As these devices become less of a technology toy, and more of a necessity, we will find that our dependence on the correct operation and non-interference of the device rises to the level that errors or defects will not be tolerated readily.

This means that the devices we will use in the future will not malfunction at all, which is not practical in engineering, or that the wireless devices developed will have to be extensively and rigorously tested. These tests have to follow a standard based approach.

The ISO/ICE 18000 - Part 7 standard defines the air interface for radio-frequency identification (RFID) devices operating in the 433.92MHz Industrial, Scientific and Medical (ISM) band used in item management applications. The test procedures are divided into two parts. Testing the interrogator (reader) and testing the tag (both active and passive).

This thesis document explains the development of generalized test procedures to test a wireless RFID device, and for conformance to the ISO/ICE 18000 – Part 7 standard as a particular example. A brief description on how to configure and use the test procedures is also included. The test procedures were developed using Labview software, which is a versatile graphical programming language that can be used for programming hardware devices. The test procedures were developed so that they can be easily updated if the standard itself changes and also they can be modified to develop new test procedures for different standards.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	XV
1.0 INTRODUCTION.....	1
1.1 STATEMENT OF THE PROBLEM.....	4
2.0 INTRODUCTION TO RFID	6
2.1 RFID TAG	6
2.2 RFID READER.....	7
2.3 RFID TECHNOLOGY	8
3.0 STANDARDS	14
4.0 RF & COMMUNICATIONS TERMINOLOGY AND GLOSSARY.....	17
4.1 MODULATION.....	17
4.1.1 Analog Modulation	18
4.1.2 Digital Modulation	18
4.2 DE-MODULATION	19
4.3 CARRIER FREQUENCY AND MESSAGE SIGNAL.....	19
4.4 NOISE.....	21
4.5 SIGNAL TO NOISE RATIO (SNR)	21
4.6 CARRIER TO NOISE RATIO	22
4.7 MODULATION INDEX.....	23

4.8	SYMBOL RATE.....	23
4.9	FREQUENCY SHIFT KEYING.....	24
4.10	FREQUENCY DEVIATION.....	25
4.11	AMPLITUDE SHIFT KEYING.....	25
4.12	PHASE SHIFT KEYING.....	26
4.13	PULSE WIDTH.....	27
4.14	RISE TIME AND FALL TIME.....	28
4.15	BANDWIDTH.....	29
4.16	RSSI.....	30
4.17	I-Q DATA.....	30
5.0	WIRELESS AND RFID TESTING OVERVIEW.....	32
6.0	BLOCK DIAGRAM OF A WIRELESS TEST SETUP.....	35
6.1	TEST SETUP IMPLEMENTED FOR TESTING ISO 18000 – 7.....	36
7.0	OVERVIEW OF LABVIEW.....	39
7.1	FRONT PANEL.....	40
7.2	BLOCK DIAGRAM.....	42
7.3	ICON AND CONNECTOR PANE.....	47
8.0	WIRELESS DEVICES AND RFSG.....	50
8.1	RHODES AND SCHWARZ SMJ100A.....	51
8.2	REMOTE CONFIGURATION OF RFSG SMJ100A.....	57
8.3	WINIQSIM.....	61
9.0	RADIO FREQUENCY SIGNAL ANALYZER.....	71
9.1	NI – 5660 RFSA.....	72

10.0	DEMODULATOR	78
10.1	DEMODULATION IN LABVIEW.....	78
11.0	UPSAMPLER.....	82
11.1	UPSAMPLING IN LABVIEW.....	82
12.0	ANALYZER	84
13.0	SYNCHRONIZATION	86
13.1	SYNCHRONIZATION IN THE TEST SETUP FOR ISO 18000-7	86
14.0	FORMAT OF A SIGNAL	89
14.1	PREAMBLE.....	90
14.2	SYNCHRONIZATION PULSES (SYNC PULSE).....	90
14.3	START OF DATA PULSES	90
14.4	DATA PULSES	91
14.5	ERROR CHECK AND CORRECTION PULSES.....	91
14.6	END OF DATA PULSE	92
14.7	COLLECTION COMMAND FOR ISO 18000-7 STANDARD	92
15.0	DESCRIPTION OF VIS DEVELOPED.....	93
15.1	FSK FREQUENCY DEVIATION TEST VI	95
15.2	FSK FREQUENCY DEVIATION TEST VI – LOAD AND TEST	101
15.3	CARRIER FREQUENCY TEST VI.....	104
15.4	CARRIER FREQUENCY TEST VI – LOAD AND TEST	108
15.5	WAKEUP SIGNAL TEST VI	111
15.6	WAKEUP SIGNAL TEST VI – LOAD AND TEST	115
15.7	PREAMBLE TEST VI.....	118

15.8	PREAMBLE TEST VI – LOAD AND TEST.....	123
15.9	DATA TRANSMITTED TEST VI.....	126
15.10	DATA TRANSMITTED TEST VI – LOAD AND TEST	130
15.11	DECODE DATA - LOAD AND TEST	133
15.12	LOAD AND ANALYZE A LONG CAPTURE VI.....	136
15.13	RECEIVER BANDWIDTH TEST VI.....	138
16.0	RESULTS	141
17.0	CONCLUSION.....	155
18.0	FUTURE DEVELOPMENTS.....	156
18.1	ADAPTING TO THE ISO 18000-6C STANDARD	159
18.2	ADAPTING TO THE ISO 18185 – 7 TYPE A STANDARD	160
	BIBLIOGRAPHY	161

LIST OF TABLES

Table 2.1: RFID Class Structure.....	9
Table 2.2: RFID Operating Frequencies.....	10
Table 8.1: Function of the four blocks in SMJ100A	53
Table 8.2: Working with Data Editor Window.....	64
Table 8.3: Modulation Settings.....	66
Table 8.4: SMU Waveform Transmission settings.....	69
Table 9.1: Key Terms in RFSA	71
Table 15.1: Common Inputs to all VIs.....	94
Table 16.1: Example Log File	148
Table 16.2: Test Timings	153
Table 16.3: Test Memory Requirements	154

LIST OF FIGURES

Figure 1.1: Complex wireless devices. Courtesy: Google Images	1
Figure 2.1: RFID Tags. Courtesy: Google Images	7
Figure 2.2: RFID Reader. Courtesy: Google Images.....	8
Figure 2.3: The Electromagnetic Spectrum	12
Figure 4.1: Modulation. Courtesy: Google Images	17
Figure 4.2: Frequency Translation.....	20
Figure 4.3: FSK Modulation. Courtesy: Wikipedia.....	24
Figure 4.4: ASK Modulation	26
Figure 4.5: PSK Modulation.....	27
Figure 4.6: Pulse Width	28
Figure 4.7: Rise time and Fall time.....	29
Figure 4.8 Bandwidth. Courtesy: Wikipedia	30
Figure 4.9: IQ data. Courtesy: NI	31
Figure 6.1: General Test setup for wireless standard conformation test.....	35
Figure 6.2: Test setup implemented for testing ISO 18000 – 7	37
Figure 7.1: Generate Sine Wave VI – Front Panel	40
Figure 7.2 Controls Window.....	41

Figure 7.3: Generate Sine Wave VI – Block Diagram	43
Figure 7.4: Functions Window	45
Figure 7.5 Generate Sine Wave VI – Block Diagram (Unwired).....	46
Figure 7.6 Structures in Labview	47
Figure 7.7: Icon Window	48
Figure 7.8: Generate Sine and calculate RMS	49
Figure 8.1 Rhodes and Schwarz SMJ100A. Courtesy: Rhodes and Schwarz	51
Figure 8.2: Four Blocks of SMJ100A.....	52
Figure 8.3: ARB Sub Menu	54
Figure 8.4: Trigger/Marker Menu.....	55
Figure 8.5: RSSMU Initialize VI.....	57
Figure 8.6: RSSMU Wait to Continue VI.....	58
Figure 8.7: RSSMU Write From File To Instrument VI	58
Figure 8.8: RSSMU ARB Waveform Select	59
Figure 8.9: R&S SMJ100A Single Mode – Sub VI.....	59
Figure 8.10: RSSMU Set ARB State VI.....	59
Figure 8.11: RSSMU Set Amplitude VI.....	60
Figure 8.12: RSSMU Set RF Frequency	60
Figure 8.13: RSSMU Set Output State	60
Figure 8.14: RSSMU ARB Execute Trigger	61
Figure 8.15: RSSMU Close	61
Figure 8.16: WinIQSim welcome page	62
Figure 8.17: Data Source window	63

Figure 8.18: Data Editor Window.....	64
Figure 8.19: Modulation Settings	66
Figure 8.20: SMU Waveform Transmission window.....	69
Figure 8.21: WinIQSim Indicator bar.....	70
Figure 9.1: NI – 5660 (NI – 5600 + NI – 5620)	73
Figure 9.2: ni5660 Initialize VI	74
Figure 9.3: ni5660 Configure for IQ VI.....	75
Figure 9.4: ni5660 Read IQ VI	75
Figure 9.5: ni5660 Initiate IQ	76
Figure 9.6: ni5660 Fetch IQ VI.....	76
Figure 9.7: ni5660 Close VI.....	77
Figure 10.1: ASK Demodulation VI.....	78
Figure 10.2: FSK Demodulator VI	79
Figure 10.3: PSK Demodulator VI	79
Figure 10.4: Demodulate AM VI.....	80
Figure 10.5: Demodulate FM VI	80
Figure 10.6: Demodulate PM VI	81
Figure 11.1: Resample VI.....	82
Figure 11.2: Median Filter VI.....	83
Figure 12.1: Pulse Measurements VI.....	85
Figure 13.1: Typical signal capture from RFSA.....	87
Figure 13.2: Obtain Notifier VI	87
Figure 13.3: Send Notification VI	88

Figure 13.4: Wait on Notification VI.....	88
Figure 14.1: Collection command for the ISO 18000-7 standard.....	92
Figure 15.1: Capture Window of FSK Frequency Deviation Test VI.....	96
Figure 15.2: Test Window of FSK Frequency Deviation Test VI.....	97
Figure 15.3: Front panel of FSK Frequency Deviation Test VI – Load and Test	101
Figure 15.4: Capture window of Carrier Frequency Test VI.....	105
Figure 15.5: Test window of Carrier Frequency Test VI.....	106
Figure 15.6: Front Panel of Carrier Frequency Test VI – Load and Test.....	109
Figure 15.7: Capture Window of Wakeup signal Test VI	112
Figure 15.8: Test Window of Wakeup signal Test VI.....	113
Figure 15.9: Front Panel of Wakeup signal Test VI – Load and Test	116
Figure 15.10: Capture Window of Preamble Test VI.....	119
Figure 15.11: Test Window of Preamble Test VI.....	121
Figure 15.12: Front panel of Preamble Test VI – Load and Test	124
Figure 15.13: Capture Window of Data Transmitted Test VI	126
Figure 15.14: Test Window of Data Transmitted Test VI.....	128
Figure 15.15 Front panel of Data Transmitted Test VI – Load and Test.....	131
Figure 15.16: Front panel of Decode Data - Reader to tag - Load and Test.....	134
Figure 15.17: Front panel of Load and analyze a long capture VI	137
Figure 15.18: Receiver Bandwidth Test VI.....	139
Figure 16.1: A signal that passes the FSK Deviation Test	141
Figure 16.2: A signal that fails the FSK Deviation Test.....	142
Figure 16.3: A signal that passes the Carrier Frequency Test	143

Figure 16.4: A signal failing Carrier Frequency Test.....	144
Figure 16.5: A signal passing Preamble Test.....	145
Figure 16.6: A signal that fails Preamble Test.....	146
Figure 16.7: A signal decoded correct	147
Figure 16.8: Signal with improper data format that fails the test	148
Figure 16.9: Command that passes data test.....	151
Figure 16.10: Analyzing a long capture.....	152
Figure 18.1: Clip command SubVI for ISO 18000-7	157
Figure 18.2: How many commands - SubVI	158
Figure 18.3: VIs that convert I and Q data into waveform	158

ACKNOWLEDGEMENTS

A leaflet of acknowledgement cannot show my feelings to the people who made this endeavor a reality.

I would like to express my sincere gratitude to Dr. Marlin H. Mickle, for giving me an opportunity to work with him and be a part of the RFID Center of Excellence. It was his guidance, support, patience and constant encouragement that made this project a wonderful research experience. He has been a great source of inspiration for my research. Working for him has been a pleasant and wonderful learning experience. His ability to breakdown the most complex of problems into simple subsets and solve them has always inspired me. I thank him for all the helpful technical discussions that have made me have a profound understanding of RF devices.

I would like to thank Dr. J.T. (Tom) Cain for all the academic help provided during my stay at the university. I consider it a great privilege being his student and I thank him for all the invaluable suggestions and advice that was definitely needed for the successful completion of this thesis. His knowledge was a guiding light in this wonderful journey and he was a constant source of motivation during the entire course of the project. I specially thank him for giving me the opportunity to work with Dr. Alex K. Jones on the business plan for the commercialization of RFID tags.

I express my profound thanks to Dr. Peter J. Hawrylak, for helping me analyze every step of my thesis. His insights on the project were of great importance. I thank him for taking time to help me debug the test procedures and for clarifying the technical ambiguity.

I thank Dr. William Stanchina and Dr. Ronald G. Hoelzeman for being a part of my thesis defense committee.

I am obliged and grateful to my parents, who are always there for me. I wish to thank my friend, Gerold Joseph Dhanabalan for his constant support and encouragement. I thank all my friends in 407 Denniston, without whom life would have been more difficult.

I apologize for any oversights and inaccuracies in my acknowledgements and thank all the professionals for guiding me.

1.0 INTRODUCTION

Wireless devices are a part of our day-to-day lives. They will become more complex and more integrated into our lives. There will be more wireless devices to come. These new devices being more complex, both in the design phase and in the manufacturing test phase, require specialized testing. Statistics has shown this to be the case. As these devices become less of a technology toy and more of an expected and also required part of our life, we will find that our dependence on the correct operation and non-interference of the device rises to the level that we will not tolerate errors or defects readily.



Figure 1.1: Complex wireless devices. Courtesy: Google Images

This means that the devices we will use in the future will not malfunction at all, which is not practical in engineering, or that the wireless devices developed and released into the market for daily customer use will have to be extensively and rigorously tested.

Another concern for aggressive testing of wireless devices is that the availability of unused channels of transmission which is free space is becoming more sparse everyday. This is most susceptible to interference when compared to all other channels of transmission (coaxial cable, Fiber optic cable or twisted pair wire). Hence, there is a need to make sure that the wireless device is by all means working in the band of frequencies it is allotted.

Again the susceptibility of the channel to noise is the reason wireless standard testing is more complicated when compared to all other standards not involving wireless devices. Special equipment designed for capturing wireless transmissions must be used. The equipment should be definitely fast enough to capture RF energy traveling at the speed of light. Transient response should be as short as possible to minimize noise caused by the test equipment itself.

As wireless devices become more complex, most of the new functionalities will be added through software modules. These software modules are either developed or bought from the many companies providing software solutions. The addition of new software modules will not increase the testing time as much as a new hardware upgrade would. But still additional time is required to test and certify the product as having passed or failed. Also one would want to spend enough time testing the software module for compatibility when integrated into the present device and also that the new software module is providing the new additional features for which it is added.

In addition, due to the complexity of the wireless systems, not every manufacturer is going to be able to design all of the hardware parts required for their product. Instead they will license designs to other companies or purchase completed modules from among the many companies competing with each other. Unfortunately, this does not reduce the testing requirements of the manufacturer. It, in fact, makes it harder to accomplish. This is because

without the innate knowledge of the design of the hardware, it is harder to determine test plans. Also it is difficult to develop test routines that minimize testing complexity and time.

Beyond the requirements for production testing, lies the need for interoperability testing between devices from different manufacturers. One should test for both the ability to operate with similar products from other sources, and in the ability not to interfere with the operation of other devices.

The foundation for all of these different test strategies is a standards based approach. A standards based approach will utilize a series of specifications developed within the industry that specifies nearly all aspects of the products operation and the environment within which it must operate. This approach has previously been used and is currently being used with great success in almost all industries. These specifications for testing wireless devices can generally be described in three categories.

Air Interface Specifications

These describe the operation of the system from the RF characteristics. A few typical examples are, the way that bits are converted into RF energy, the composition of the messages that are sent across the RF interface, and the use or purpose of the messages.

Minimum Performance Specifications

These describe the minimal characteristics of the device so that it may function as described by the Air Interface Specification. These generally describe RF (hardware) characteristics, but will occasionally address the minimum characteristics of the upper layer of the seven OSI layers. The upper layer is generally software oriented. These specifications are a good foundation for the product test plan.

Interoperability Specifications

These describe how the product will work in the rest of the world. These typically describe interfaces to external systems, but have been used to address expected interferences, although this is occasionally part of the Minimum Performance Specification.

With well-defined industry wide standards, it is possible to design efficient test plans that address the requirements of the market and ease the integration with the rest of the world. In addition, with a widely used industry specification, it is possible for test equipment vendors to build innovative test solutions, at a practical cost, that can minimize test plan development time and shorten the per unit test time.

1.1 STATEMENT OF THE PROBLEM

The standard conformance testing strategies require complex and expensive equipment. It is cumbersome to setup the test equipment every time the test is performed. The user performing the tests will require a certain level of skill and training to operate the test equipment and to understand the technical ambiguity involved. Hence there is a necessity for automated test procedures which are easy to understand and operate.

This thesis document explains the development of automated test procedures for wireless RFID devices conforming to the standard ISO/ICE 18000 – Part 7. These test procedures can be easily maneuvered to obtain the desired outputs. The test procedures setup the test equipment to the same configuration, every time the test is performed, minimizing human errors. A text file is

recorded describing the protocol adopted for each test. The outputs of each test can be saved to memory and can be tested anew for conformation. The testing strategies implemented are upgradeable and flexible.

2.0 INTRODUCTION TO RFID

We have understood the importance of wireless devices and testing them. Most of the information in this document applies to all wireless devices. The document in many sections quotes ISO 18000 – Part 7 standard tested on RFID tags as an example. ISO 18000 – Part 7 standard defines the test methods for active air interface communication at 433.92MHz. As most of the testing examples explained are about RFID tags, this section gives a brief description about Radio Frequency Identification Tags (RFID Tags), their types, working and uses.

At the very simplest level, Radio Frequency Identification (RFID) technologies allow the transmission of a unique serial number wirelessly, using radio waves. The two key parts of the system that are needed to do this are the RFID tag and the RFID reader. Attaching an RFID tag to a physical object allows the object to be “seen” and monitored by existing computer networks and office administration systems.

2.1 RFID TAG

There are two main components present in the RFID tag. Firstly, a small silicon chip or integrated circuit which contains a unique identification number (ID). Secondly, an antenna that can send and receive radio waves. These two components are usually attached to a flat plastic tag that can be fixed to a physical item. These tags can be quite small, thin and, increasingly, easily embedded within packaging, plastic cards, tickets, clothing labels, pallets and books. There are two main types of tags: passive and active. Passive tags are currently the most widely deployed as they are the cheapest to produce. More about active and passive tags is discussed in later sections.



Figure 2.1: RFID Tags. Courtesy: Google Images

2.2 RFID READER

The reader is a handheld or fixed unit that can interrogate nearby RFID tags and obtain their ID numbers using radio frequency (RF) communication. These ID numbers are unique to each tag. This enables the packet to be easily tracked when a tag is attached to it.

RFID readers can be classified into two main classes: read-only and read-write. A read only reader can only interrogate the tag and get the ID number of the tag. No information in the tag is modified. Read-write tags can interrogate the tag and also can write new information into the tag. These tags are equipped with a read-write memory. Many models of reader are handheld devices and resemble the pricing guns or barcode scanners used in supermarkets, but readers can also be fixed in place and even hidden.



Figure 2.2: RFID Reader. Courtesy: Google Images

2.3 RFID TECHNOLOGY

There are three key elements that need to be borne in mind in any discussion of RFID systems.

- i. Energy Source (which determines if a tag is passive or active)
- ii. Frequency
- iii. Memory

RFID Class Structure

The basic structure defines five classes in ascending order as shown in the Table 2.1 below. The Classification is based on the energy source of RFID tag.

Table 2.1: RFID Class Structure

S.No.	Class Name	Description and Functionality
1	Identity Tags	These tags are purely passive and primarily used for identification
2	Higher Functionality Tags	They are also passive. They are used for identification and the data in these tags can be modified by the appropriate reader.
3	Semi – Passive Tags	These tags have on-board battery power. Semi-passive tags differ from fully active tags in that they still require stimulation by an RFID reader to broadcast information. In the absence of a reader signal, they lie dormant.
4	Active Tags	These tags have a battery much powerful than the semi passive tags. They can transmit RF data on their own. Also some active tags communicate with other active tags.
5	Reader Tags	These tags can provide power for other tags and communicate with them. They can act as a reader, transmitting and receiving radio waves.

Frequency

RFID is fundamentally based on wireless communication, making use of radio waves, which form part of the electromagnetic spectrum (i.e. frequencies from 300 kHz to 3 GHz). RFID operates in unlicensed spectrum space, sometimes referred to as ISM (Industrial, Scientific and Medical) but the exact frequencies that constitute ISM may vary depending on the regulations in different countries. These operating frequencies are generally considered to be organized into four main frequency bands and the table below shows these different radio wave bands and the more common frequencies used for RFID systems (IEE, 2005).

Within a given frequency band, the actual real-world communication range will vary widely depending on factors such as the operating environment, the detail of the antenna design and the available system power. The following Table 2.2 gives the RFID operating frequencies and their characteristics at a glimpse.

Table 2.2: RFID Operating Frequencies

Band	LF – Low Frequency	HF – High Frequency	UHF – Ultra High Frequency	Microwave
Frequency	30 – 300kHz	3 – 30MHz	300MHz – 3GHz	2 – 30GHz
Typical RFID Frequencies	125 – 134 kHz	13.56MHz	433MHz and 865 – 956MHz and 2.45GHz	2.45GHz
Approximate Read Range	Less than half a meter	Up to 1.5 meters	433 MHz – up to 100 meters, 865 – 956MHz – from 0.5m to 5m	Up to 10 meters

Table 2.2 (continued)

Typical Data Transfer Rate	Less than one kilo bit per second	Approximately 25 kilo bits per second	433 MHz – up to 30 kilo bits per second, 865 – 956MHz – up to 30 kilo bits per second, 2.45GHz – up to 100 kilo bits per second	Up to 100 kilo bits per second
Characteristics	These tags are used for short-range. They have a low data transfer rate. Their signals penetrate water but not metal.	These tags have higher ranges compared to LF tags. They have reasonable data transfer rate. Their signals also penetrate water but not metal.	These tags have much long ranges. They can transfer data at a higher data rate. Some systems can do a concurrent read of up to 100 items. Their signals cannot penetrate water or metals.	These tags have the longest range, highest data transfer rate. Their signals cannot penetrate water or metal.
Typical Use	Animal ID, Car, immobilizer	Smart Labels Contact-less travel cards Access & Security.	Specialist animal tracking Logistics.	Moving vehicle toll.

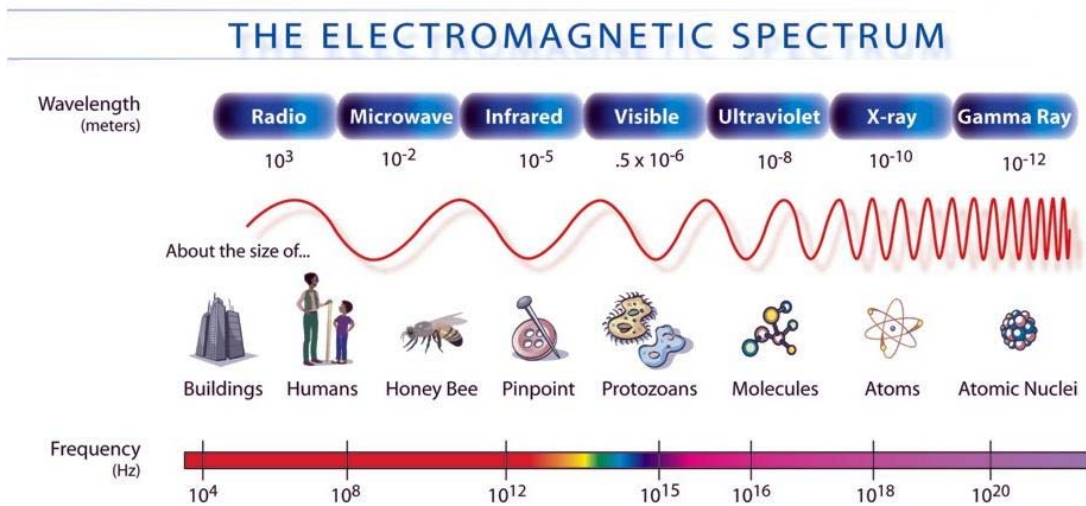


Figure 2.3: The Electromagnetic Spectrum

There are two types of RFID systems. Each of these systems uses different physical properties to enable communication between the reader and the tag. It is important to realize that it partly determines the operating range of the systems.

RFID systems based on LF and HF frequencies make use of near field communication and the physical property of inductive coupling from a magnetic field. The reader creates a magnetic field between the reader and the tag which induces an electric current in the tag's antenna, which is used to power the integrated circuit and obtain the ID. The ID is communicated back to the reader by varying the load on the antenna's coil which changes the current drawn on the reader's communication coil.

RFID systems based on UHF and higher frequencies use far field communication and the physical property of backscattering or reflected power. Far field communication is based on radio waves. The reader sends a continuous base signal frequency that is reflected back by the tag's antenna. During the process, the tag encodes the signal to be reflected with the information from the tag (the ID) using a technique called modulation.

Memory

As already described earlier, the type of reader determines the type of memory of the tag it is communication with. A read-only tag, as the name suggests, has a read-only memory. The unique ID code is permanently stored on the tag. This cannot be modified by the reader. These types of tags are also known as WORM meaning Write Once Read Many Tags.

A read-write tag, again as the name suggests has a read-write memory. This allows a user to change the ID and add additional data to the tag's memory through a read-write reader.

Some tags are a combination of the two types mentioned above. They may have a unique ID which cannot be modified. However the tag might also contain some additional memory that is user programmable.

Passive tags typically have anywhere from 64 bits to 1 kilobyte of non-volatile memory. Active tags tend to have larger memories with a range of typically between 16 bytes and 128 kilobytes.

3.0 STANDARDS

The number and use of standards within RFID and its associated industries are quite complex. They involve a number of bodies and are in a continuous process of development. Standards have been produced to cover four key areas of RFID application and use.

- Air interface standards (for basic tag-to-reader data communication)
- Data content and encoding (numbering schemes)
- Conformance (testing of RFID systems) and
- Interoperability between applications and RFID systems.

There are several standards bodies involved in the development and definition of RFID technologies including:

- International Organization of Standardization (ISO)
- EPCglobal Inc.
- European Telecommunications Standards Institute (ETSI)
- Federal Communications Commission (FCC)

Air Interface Standards

RFID frequencies are governed by the ISO 18000–RFID Air Interface family of standards, and a complete set of standards was released in September 2004:

1. ISO 18000-1 – Generic Parameters for the Air Interface for Globally Accepted Frequencies

2. ISO 18000-2 – for frequencies below 135 kHz
3. ISO 18000-3 – for 13.56 MHz
4. ISO 18000-4 – for 2.45 GHz
5. ISO 18000-6 – for 860 to 960 MHz
6. ISO 18000-7 – for 433 MHz

There are also earlier standards relating to, for example, cattle tracking systems (ISO 11785), tag-based payment proximity cards (ISO 14443) and electronic toll collection vicinity cards (ISO 15693). ISO 14443 and ISO 15693 both operate at 13.56MHz (HF), but the first standard has a read range of about 10cm whereas the later has a read range of 1 to 1.5 meters.

Examples in this document quote passages from ISO 18000-7 defined for 433 MHz wireless interface. Test automation programs have been developed for this standard in Labview 8.2. Understanding the details of the standard will help in understanding the test procedures and the overall test setup.

ISO 18000 – Part 7

ISO/ICE 18000 - Part 7 defines the air interface for radio-frequency identification (RFID) devices operating in the 433.92MHz Industrial, Scientific and Medical (ISM) band used in item management applications.

The test procedures are divided into two parts. Testing the interrogator (reader) and testing the tag (both active and passive). Tags conforming to this standard use FSK modulation for communication. The tests concentrate on:

1. Accuracy of center frequency
2. Accuracy of FSK frequency deviation
3. Wakeup signal
4. Preamble signal
5. Data portion of the command

Most of the tests check for the pulse widths to be within values specified in the standard. When measuring for frequency deviation we check that bit '1' is represented by a frequency within a band of frequencies and bit '0' is represented by a frequency within another band of frequencies.

Checking the data portion is most critical. All the data pulses are checked to be within specified values. The format of the data is also verified.

4.0 RF & COMMUNICATIONS TERMINOLOGY AND GLOSSARY

4.1 MODULATION

Modulation in simplest terms is the process of modifying a periodic waveform in accordance with the message signal. Normally a high-frequency sinusoid waveform is used as a carrier signal. A high frequency sinusoidal waveform has three key parameters – amplitude, frequency and phase. All of them can be modified according to the message signal to get the modulated waveform. A device that performs modulation is termed a modulator.

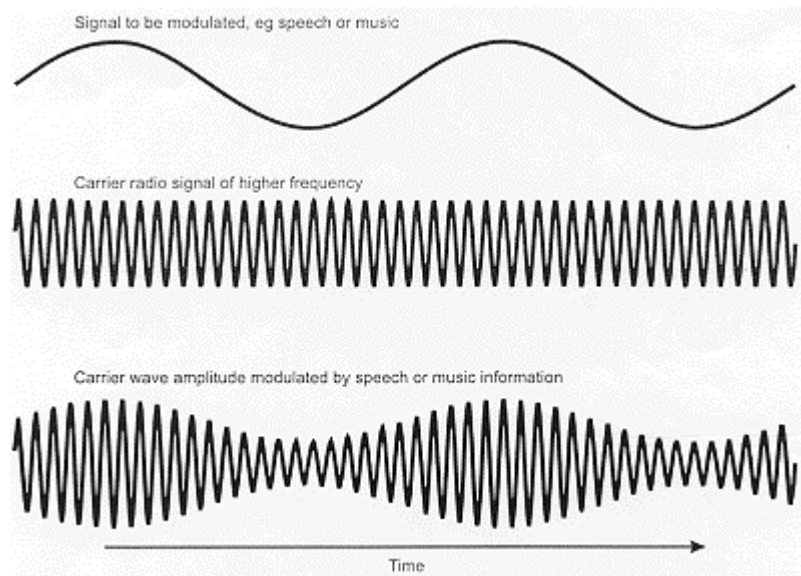


Figure 4.1: Modulation. Courtesy: Google Images

The aim of modulation is to convert a baseband or message signal into a high frequency signal. This has several advantages the most important being signals becoming more immune to noise. Modulation is the soul of modern communication systems. Modulation can be further divided into Analog modulation and Digital modulation schemes.

4.1.1 Analog Modulation

In analog modulation the message signal is an analog-continuous signal. The modulation is applied continuously in response to the analog information signal. Common analog modulation techniques are:

- Angular modulation
 - Phase modulation (PM)
 - Frequency modulation (FM)
- Amplitude modulation (AM)
 - Double-sideband modulation with unsuppressed carrier (used on the radio AM band)
 - Double-sideband suppressed-carrier transmission (DSB-SC)
 - Double-sideband reduced carrier transmission (DSB-RC)
 - Single-sideband modulation (SSB, or SSB-AM), very similar to single-sideband suppressed carrier modulation (SSB-SC)
 - Vestigial-sideband modulation (VSB, or VSB-AM)
 - Quadrature amplitude modulation (QAM)

4.1.2 Digital Modulation

In digital modulation the message signal is a digital signal (only 2 amplitude values). An analog carrier signal is modulated by a digital bit stream of either equal length signals or varying length signals. A simple way of visualizing digital modulation is when we have two carriers,

each representing bit '1' and bit '0' respectively. The two carriers may be different in any of the three key aspects of a carrier signal. The three most common digital modulation techniques are:

1. Frequency Shift Keying
2. Amplitude Shift Keying
3. Phase Shift Keying

Most of the wireless devices use modulation that can be grouped into one of the above three techniques.

4.2 DE-MODULATION

De-modulation is the exact opposite of modulation. Here the carrier signal is removed and the original baseband or the message signal is retrieved. Demodulating is necessary because the receiver system receives a modulated signal with specific characteristics, and it needs to turn it into base-band.

There are several ways of demodulating depending on what parameters of the base-band signal are transmitted in the carrier signal, such as amplitude, frequency or phase. But one must know in advance what type of modulation has been performed on the carrier to do demodulation.

4.3 CARRIER FREQUENCY AND MESSAGE SIGNAL

The frequency of the unmodulated electrical wave at the output of an amplitude modulated (AM), frequency modulated (FM), or phase modulated (PM) transmitter is the system's carrier frequency. This can also be viewed as the output of the modulator when

modulation is zero. A carrier signal's frequency is generally very high when compared to the message signal.

Message signal, as the name suggests, is a very low frequency signal which actually contains the data. Hence the name base band signals. But as most of the messages are low frequency signals, they cannot be communicated through free space without modulation. Modulation can be viewed as the process of converting a low frequency signal whose frequency spectrum is centered on zero, to a high frequency signal whose center is at the carrier frequency.

It is interesting to note that the modulated signal will occupy the same bandwidth as occupied by the unmodulated baseband signal.

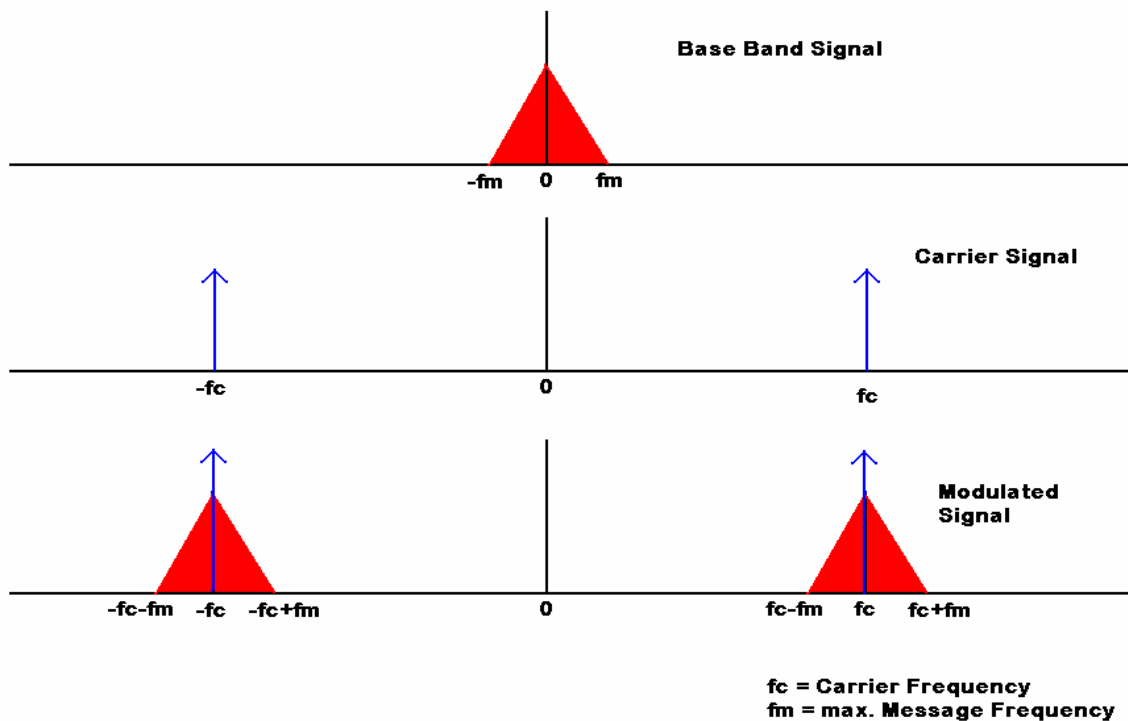


Figure 4.2: Frequency Translation

4.4 NOISE

In communication, noise refers to influences on the modulated carrier signal that affects the recovery of the message signal. In simpler words, what we receive that is useless is noise. Noise is what drives communication to be more efficient.

Noise can affect any of the three key parameters of the carrier. It is interesting to note that when one type of modulation is employed and noise affects another parameter, the message is not altered. This is this reason that many types of modulation were developed.

Sometimes the same message is modulated in different ways and sent from one place to another. The reason being, it is highly improbable that the same noise can affect the same signal in such a way that when demodulated, the effect will alter the message in the same place. This is called modulation diversity transmission.

4.5 SIGNAL TO NOISE RATIO (SNR)

Signal-to-noise ratio (abbreviated SNR) is an electrical engineering concept defined as the ratio of a signal power to the noise power corrupting the signal.

$$\text{SNR} = \frac{P_{\text{signal}}}{P_{\text{noise}}} = \left(\frac{A_{\text{signal}}}{A_{\text{noise}}} \right)^2$$

In less technical terms, signal-to-noise ratio compares the level of a desired signal (message) to the level of unwanted noise. The higher the ratio, the more clear the message is and the better is the communication system. One should note that SNR is not the only thing that rates a communication system. SNR is usually expressed in dB using the formula:

$$\text{SNR(dB)} = 10 \log_{10} \left(\frac{P_{\text{signal}}}{P_{\text{noise}}} \right) = 20 \log_{10} \left(\frac{A_{\text{signal}}}{A_{\text{noise}}} \right)$$

If reference (denominator) power or amplitudes are expressed in milli then SNR is said to be expressed in dBm.

4.6 CARRIER TO NOISE RATIO

Carrier to noise ratio is very similar to SNR. The carrier-to-noise ratio (CNR) is a measure of the received carrier strength relative to the strength of the received noise.

High CNR provides better quality of reception, and generally results in higher communications accuracy and reliability. Engineers specify the CNR in decibels (dB) between the power in the carrier of the desired signal and the total received noise power. If the incoming carrier strength in microwatts is P_{carrier} and the noise level, also in microwatts, is P_{noise} , then the carrier-to-noise ratio in dB is given by the formula:

$$\text{C/N(dB)} = 10 \log_{10} \left(\frac{P_{\text{carrier}}}{P_{\text{noise}}} \right)$$

The SNR specification is more meaningful in practical situations. The CNR is commonly used in satellite communications systems to point or align the receiving antenna. The best alignment is indicated by the maximum CNR.

4.7 MODULATION INDEX

The modulation index describes by how much the modulated variable of the carrier signal varies around its unmodulated level. It is defined differently in each modulation scheme.

In FM, this quantity indicates by how much the modulated variable varies around its unmodulated level. For FM, it relates to the variations in the frequency of the carrier signal:

$$h = \frac{\Delta f}{f_m} = \frac{f_{\Delta} |x_m(t)|}{f_m}$$

If $h \ll 1$, the modulation is called narrowband FM, and its bandwidth is approximately $2f_m$. If $h \gg 1$, the modulation is called wideband FM, and its bandwidth is approximately $2f_{\Delta}$. While wideband FM uses more bandwidth, it can improve signal-to-noise ratio significantly.

4.8 SYMBOL RATE

In digital communications, the symbol rate is the bit rate (bits per unit time) divided by the number of bits transmitted in each symbol. Symbol rate is measured in symbols per second, hertz (Hz), or baud. The term baud rate is synonymous with symbol rate.

There is no fixed relationship between the symbol rate and the bit rate. The simplest digital communication links typically have a symbol rate equal to the bit rate.

4.9 FREQUENCY SHIFT KEYING

Frequency Shift Keying (FSK) is a type of digital modulation. Here the message signal, which is digital, shifts the carrier signal frequency between two predefined values, each one corresponding to bit '1' and bit '0' respectively.

The carrier frequency is shifted between two discrete values termed the mark frequency and the space frequency. Usually mark frequency is greater than space frequency and represents bit '1'. Figure 4.3 illustrates a FSK modulated signal.

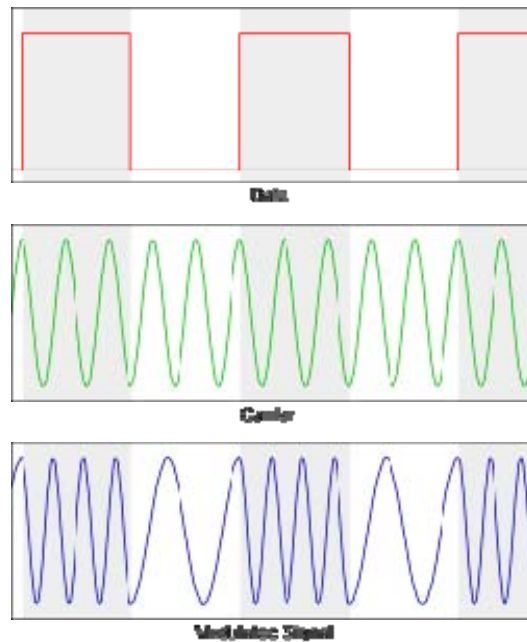


Figure 4.3: FSK Modulation. Courtesy: Wikipedia

4.10 FREQUENCY DEVIATION

Frequency deviation quantifies the amount by which a frequency differs from its specified value, as when measuring how much an oscillator frequency deviates from its nominal frequency.

In frequency modulation, frequency deviation refers to the maximum absolute difference, during a specified period, between the instantaneous frequency of the modulated wave and the carrier frequency.

In other words, it is the difference between mark frequency and carrier frequency or carrier frequency and space frequency.

4.11 AMPLITUDE SHIFT KEYING

The amplitude of an analog carrier signal varies in accordance with the bit stream (message signal), keeping frequency and phase constant. The level of amplitude can be used to represent binary logic '0' and '1'. We can think of a carrier signal as an ON or OFF switch. In the modulated signal, bit '0' is represented by the absence of a carrier, thus giving OFF-ON keying operation and hence the name. Figure 4.4 shows ASK modulated signal.

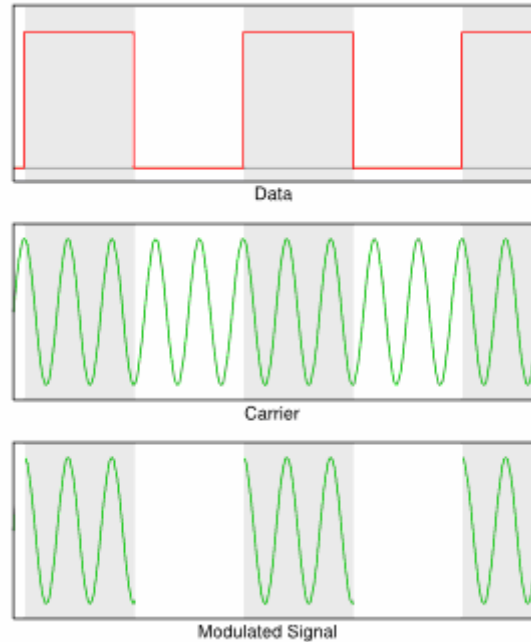


Figure 4.4: ASK Modulation

4.12 PHASE SHIFT KEYING

PSK is similar to FSK, the only difference is that the phase changes where the frequency would. In FSK the message signal, which is digital, shifts the carrier signal phase between two predefined values, each one corresponding to bit '1' and bit '0' respectively. Generally the two phases are 180 degrees apart. The frequency of the carrier remains the same. Figure 4.5 illustrates a PSK modulated signal.

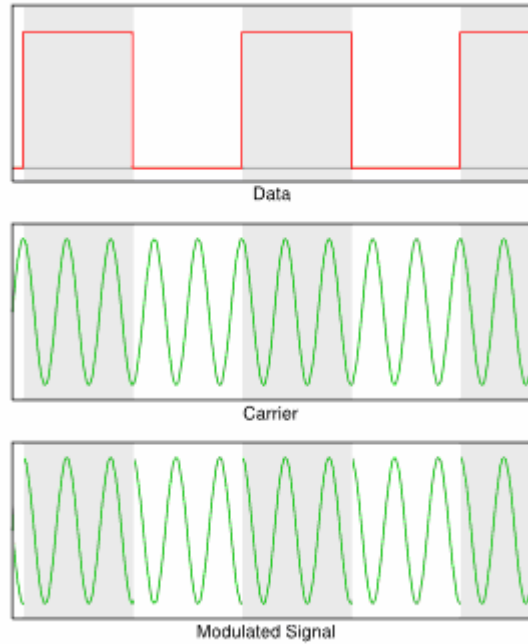


Figure 4.5: PSK Modulation

4.13 PULSE WIDTH

Pulse width is defined as the difference between the consecutive times when the signal crosses zero or some predetermined level.

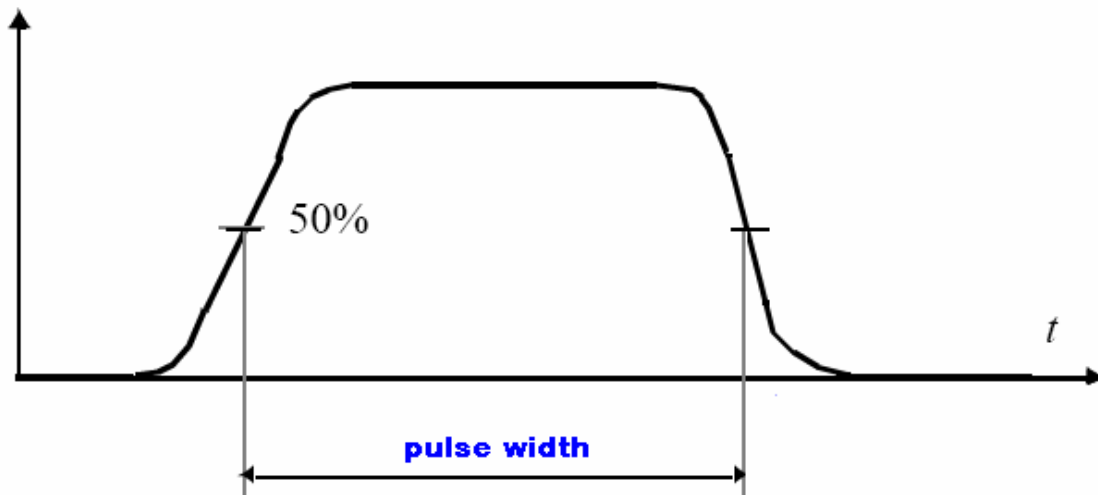


Figure 4.6: Pulse Width

4.14 RISE TIME AND FALL TIME

Rise time (t_r) is typically defined as the time required by the rising edge of the pulse to go from 10% of maximum value to 90% of maximum value.

Fall time (t_f) is typically defined as the time required by the falling edge of the pulse to go from 90% of maximum value to 10% of maximum value.

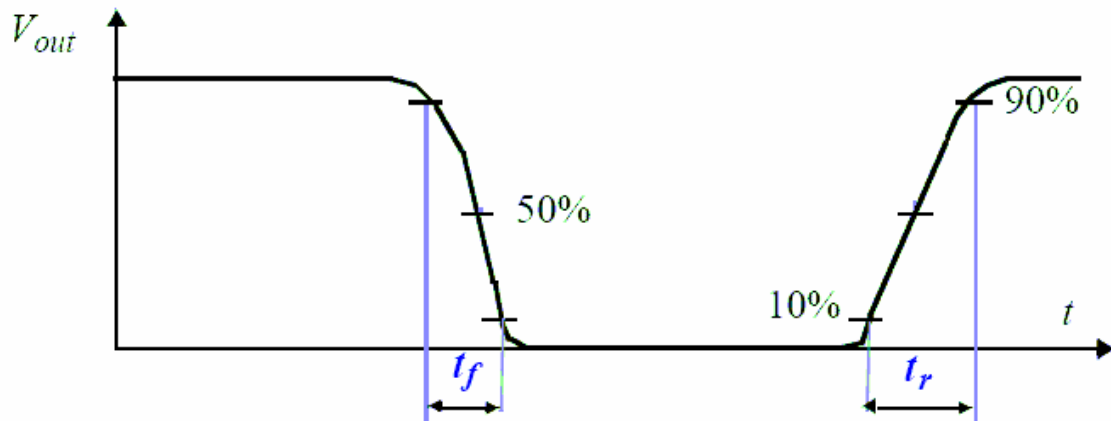


Figure 4.7: Rise time and Fall time

4.15 BANDWIDTH

A wireless device receives a maximum power at f_c and receives half the power at f_1 and f_2 , where $f_1 < f_2$. Then the bandwidth is defined as the difference between the half power frequencies ($f_2 - f_1$). This is also called the 3dB bandwidth.

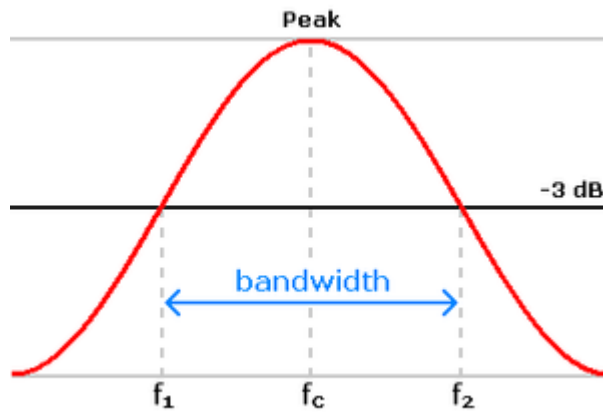


Figure 4.8 Bandwidth. Courtesy: Wikipedia

4.16 RSSI

RSSI (Received Signal Strength Indication) is a measurement of the received radio signal strength. RSSI output is often a DC analog level.

4.17 I-Q DATA

A sinusoidal wave of the form $f(t) = r \cdot \cos(\omega t + \theta)$, can be expressed in terms of its in-phase component (I) and Quadrature-phase component (Q). The I and Q components are orthogonal and hence do not interfere with each other.

$$I(t) = r \times \cos(\theta) \times \cos(\omega t)$$

$$Q(t) = r \times \sin(\theta) \times \sin(\omega t)$$

$$F(t) = I(t) - Q(t)$$

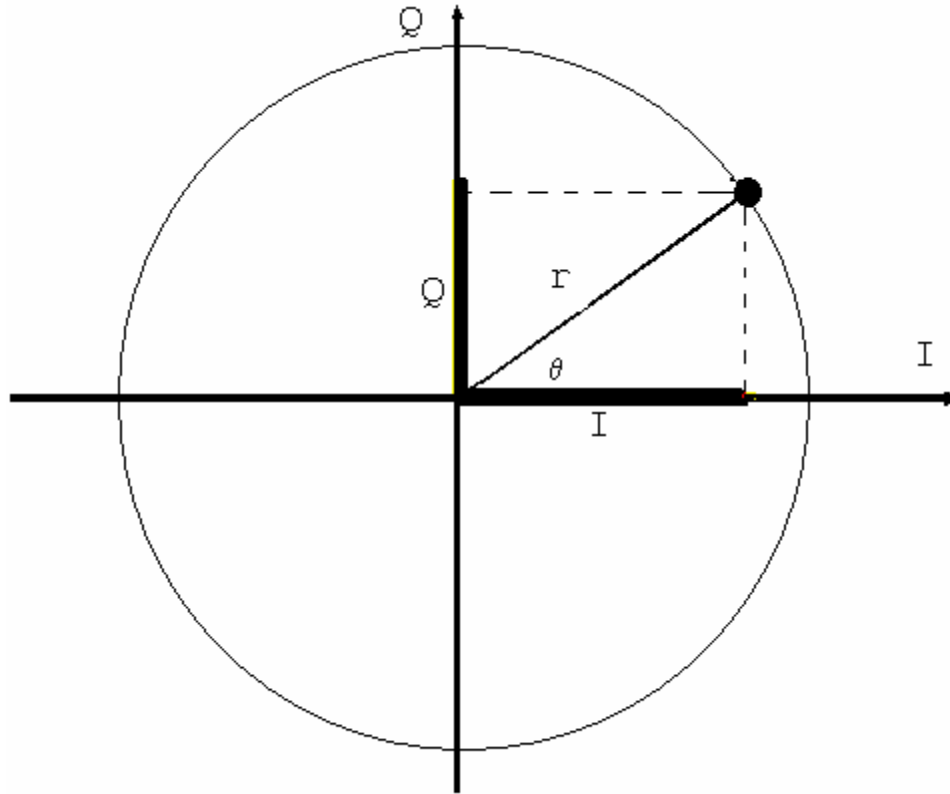


Figure 4.9: IQ data. Courtesy: NI

IQ data are important because most RFSAs give IQ as their output. This is fed to the demodulator as input.

5.0 WIRELESS AND RFID TESTING OVERVIEW

Every wireless communication system must pass regulatory requirements and must conform to whatever standard is applicable. Today, however, it is system optimization that separates the winners from the losers in this fast growing industry. This optimization at all levels must meet the industrial standards. The standard is as important to a developing engineer as it is to a testing engineer. A good understanding of the standard will allow the developer to stretch the device functionality to the limit while at the same time not breaking rules and optimizing cost of production and time to market. This section discusses the testing challenges facing the designer of a wireless communication system: regulatory testing, standards conformance and optimization.

RFID technologies present several uncommon engineering measurement challenges such as transient signals, bandwidth inefficient modulations and backscattered data. Traditionally, swept tuned spectrum analyzers, vector signal analyzers and oscilloscopes have been used for wireless data link development. However, each of these tools has disadvantages when used for RFID testing. Swept tuned spectrum analyzers have difficulty accurately capturing and characterizing transient RF signals. Vector signal analyzers have virtually no support for spectrally inefficient RFID modulations and their special decoding requirements and demodulation cannot be done in real time. Fast oscilloscopes have substantially less measurement dynamic range and lack modulation and decoding capability. We need signal analyzers that are optimized for transient signals and have ability to reliably trigger on specific spectral events in complex real-world spectral environments.

Regulatory Testing

Every producer of electronic equipment, wireless or not, must in general meet regulatory standards where the equipment will be sold or used. Regulatory tests are focused on confirming that the equipment is serving the purpose for which it is introduced. In other words, it tests the fact that the device is operating as it should in all three conditions – ideal, pragmatic and hostile. Boundary conditions should be taken into consideration when doing these tests.

The regulatory laws are changing in most countries, to catch up with the unique data link characteristics of a passive RFID tag. Most regulators prohibit CW transmissions from devices unless they are for a short term test. However, many regulations do not address non transmitter based modulation. A variety of spectral emission tests, which may not be explicitly contained in the RFID standard for the reader become requirements. For example, the minimum power that should be maintained by the reader, to sufficiently feed the passive tag, when the power radiated from reader is not directional. Also many other regulations might limit the power that can be transmitted by the reader.

Government regulations require that transmitted signals be controlled in power, frequency and bandwidth. These regulations prevent harmful interference between wireless devices. They also ensure that each transmitter is a spectrally good neighbor to other users of the frequency band. Power measurements of pulsed signals can be challenging for many spectrum analyzers, especially swept spectrum analyzers typically used for these types of measurements. An analyzer that can recognize the modulation of a transient RFID signal and make regulatory measurements of power, frequency and bandwidth at the touch of a button makes the process of pre-compliance testing swift and easy. Pre-compliance testing helps ensure that the product passes the compliance test the first time, eliminating the need for re-design and re-test.

Standards Conformance

Reliable reader and tag interaction requires conformance to industry standards such as the ISO 18000-6 type C and ISO 18000-7 specifications. This requirement adds many tests beyond those essential to meet government spectral emissions requirements. RF conformance tests are critical to assure reliable operation among tags and readers in both directions of communication. Pre-programmed measurements can reduce the setup time required to make these tests.

For example, one important measurement for ISO 18000-7 is the pulse width of the data pulses. The data are Manchester encoded with a stop bit every eight bits. Also, because the coding is Manchester, when the data changes from one bit to the other, the pulse widths add making a pulse double the original width. While testing the pulse widths, one must exercise extreme caution and check the starting and ending of each bit and measure pulse widths accordingly. Here knowing the data transmitted before decoding and testing it proves helpful.

Some RFID devices use proprietary communications schemes optimized for specific applications. In this case, engineers need an analyzer that provides multiple modulation and coding schemes that can be programmatically adjusted for the specific format in use.

6.0 BLOCK DIAGRAM OF A WIRELESS TEST SETUP

Figure 6.1 below shows a typical setup for wireless standards testing. The two wireless devices shown in the block diagram are the wireless devices under test. For effective testing of each device, it is sometimes required to test them separately using a “gold” standard for the second device. A Radio Frequency Signal Generator (RFSG) can be programmed to emulate the wireless device not in test. The RF communication between the wireless device under test and the RFSG are captured by a Radio Frequency Signal Analyzer (RFSA). Most RFSA's give an IQ output which is fed into a demodulator. It is typical that wireless devices use modulation like ASK, FSK and PSK for efficient utilization of bandwidth and also to minimize bit error rate. So the demodulator is necessary to demodulate the IQ into electrical pulses or digital data. The demodulated signal should be upsampled for effective testing purposes. Finally, the upsampled signal is sent to the analyzer which is programmed to test the signal. A synchronizer is necessary in most cases for effective testing.

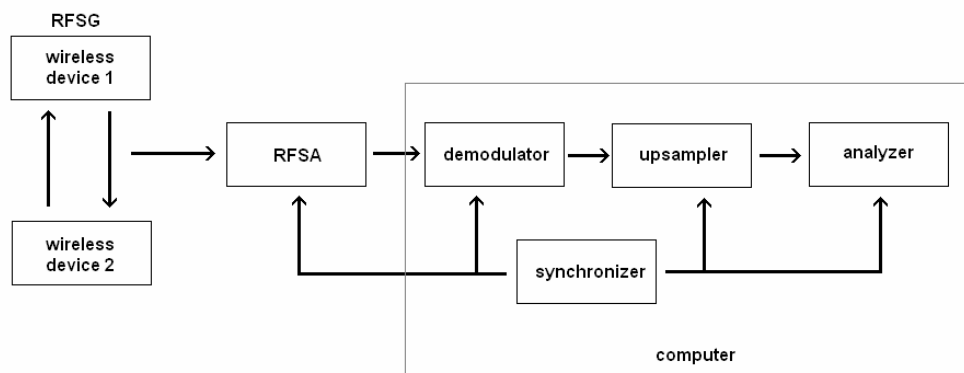


Figure 6.1: General Test setup for wireless standard conformation test

In a typical test setup, demodulation, upsampling and standard conformation analysis are done in software as it is easily programmable and cost effective. A computer is generally used for these three blocks. But sometimes demodulation can be done by hardware and the demodulated waveform can be taken as input by the computer. Synchronization can be in software or hardware. This choice mainly depends on the hardware being used and its specifications. Synchronization in hardware is definitely more accurate than in software.

The test setup specified in Figure 6.1 also contains wireless antennas, PXI bus, Ethernet cables, GPIB cables and probes.

It is always advised to have a Real Time Spectrum Analyzer in the test setup. It can be useful when tests are being developed to see if the RFSG is producing the correct signal. Once the tests are fully developed, the RTSA can be used as an observer. Then its function will be to observe the signals going back and forth between wireless device 1 and wireless device 2. Once a series of commands is sent between the wireless devices, the RTSA can be used to verify the sequence of commands sent and the time in-between each command.

6.1 TEST SETUP IMPLEMENTED FOR TESTING ISO 18000 – 7

Figure 6.2 below illustrates the test setup that has been used to develop the testing procedures for ISO 18000-7 standard reported in this thesis.

In the test setup, the Rhodes – Schwarz SMJ100A (RFSG) is used to generate RF signals. The RFSG can be programmed to emulate the reader or the tag depending on the device under test.

When testing a tag, the RFSG will be programmed to act as a reader. It will send out a command such as the collection command just like a commercial RFID reader. The tag will answer by sending the response signal. This response can be analyzed for standard conformance.

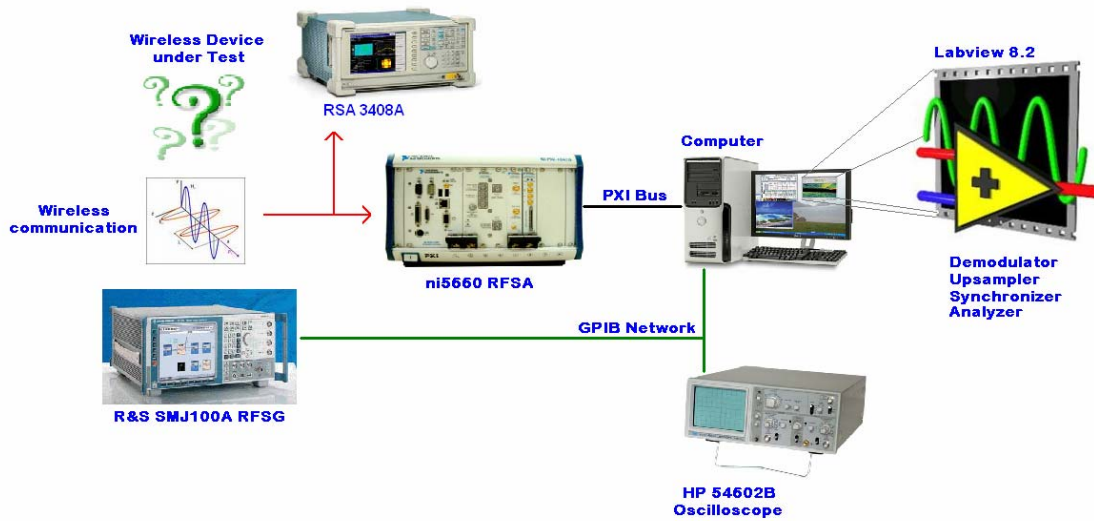


Figure 6.2: Test setup implemented for testing ISO 18000 – 7

All wireless RF data transmission is captured by the RFSA. The RFSA used is NI-5660. It will capture the command and its response and send them to the computer. The computer is the central unit that controls all of the other devices. The IQ data are demodulated into digital data by the software Labview 8.2. Labview is very powerful software that can be used to communicate with hardware devices. Here Labview not only demodulates the signal but it also acts as a synchronizer by configuring the RFSG to send the command and configuring the RFSA to capture the response at the correct time. Labview is also an excellent choice when analyzing and manipulating signals. Labview has an upsampler readily implemented in software.

The HP54062B Oscilloscope is used for one specific test where a DC voltage has to be read by the computer. Again Labview is used to configure the oscilloscope and to read from it.

RFSG and the oscilloscope are connected using the GPIB network cable. The RFSA is connected using the PXI bus. An Ethernet cable can be used when communicating with the RFSG, which has the advantage of faster data transfer, but the oscilloscope has only a GPIB port.

More about each of these devices and their controls is explained in detail in the following sections.

7.0 OVERVIEW OF LABVIEW

Labview can be viewed as any other conventional programming language, but here the program is written graphically rather than following syntax. It uses icons analogous to functions instead of text. Labview uses dataflow programming where the flow of data determines execution. Labview offers almost all the features of any other programming language and sometimes even more. It is an excellent choice when working with hardware devices through remote connection. Therefore, to understand remote operation of all the test setup through Labview, it is essential to get acquainted with the basics of the software first.

Labview is not only simple to program, but it is also very easy to understand. The main reason for this is that the user need not know any syntax to understand the code. Data flows through wires from one sub VI to another. Just by understanding inputs and outputs of each VI, one can get the overview of the program.

Labview programs are called virtual instruments, or VIs, because their appearance and operation imitate physical instruments. Labview has a variety of instruments like upsamplers, encoders, decoders, modulators, demodulators and filters all implemented in software. Every VI uses functions that manipulate input from the user interface or other sources and display that information or move it to other files or hardware devices. A VI can be separated into three main components:

1. Front Panel
2. Block Diagram
3. Icon and Connector Pane

7.1 FRONT PANEL

The front panel acts as the graphical user interface of the VI. The user gives inputs and can see the outputs in the front pane. We build front panels using controls, push buttons, numerical inputs, indicators (LEDs) and graphs.

Figure 7.1 shows the front panel of a VI that generates a sine wave based on amplitude and frequency inputs given by the user. The VI also outputs the RMS voltage value. Another input to the VI is the stop button which stops generating the sine wave.

In the example all the inputs have been grouped to the left and all outputs are grouped to the right. As we change the frequency and amplitude input, the waveform in the graph (output) also changes.

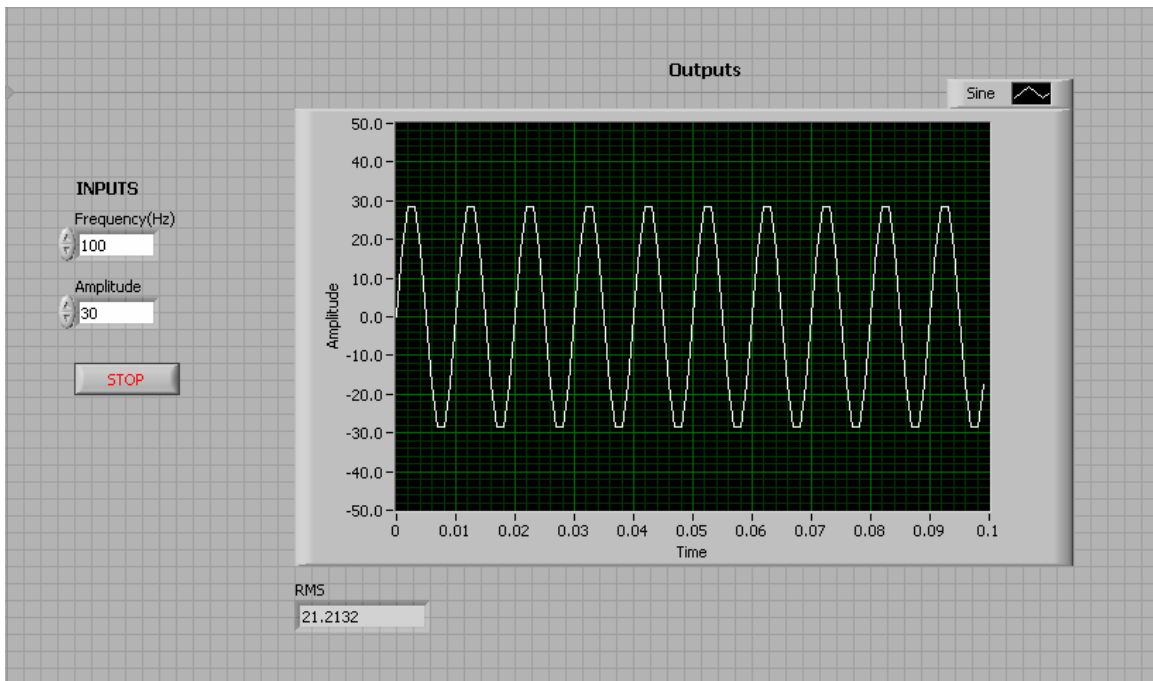


Figure 7.1: Generate Sine Wave VI – Front Panel

Building a Front panel

The front panel is built with controls and indicators, which are the interactive input and output terminals of the VI, respectively. Controls are knobs, push buttons, dials, and other input devices. Indicators are graphs, LEDs, and other displays. Controls simulate instrument input devices and supply data to the block diagram of the VI. Indicators simulate instrument output devices and display data the block diagram acquires or generates.

To add an indicator to the front panel, right click anywhere on the free portion of the front panel. A controls window is opened as shown in Figure 7.2.

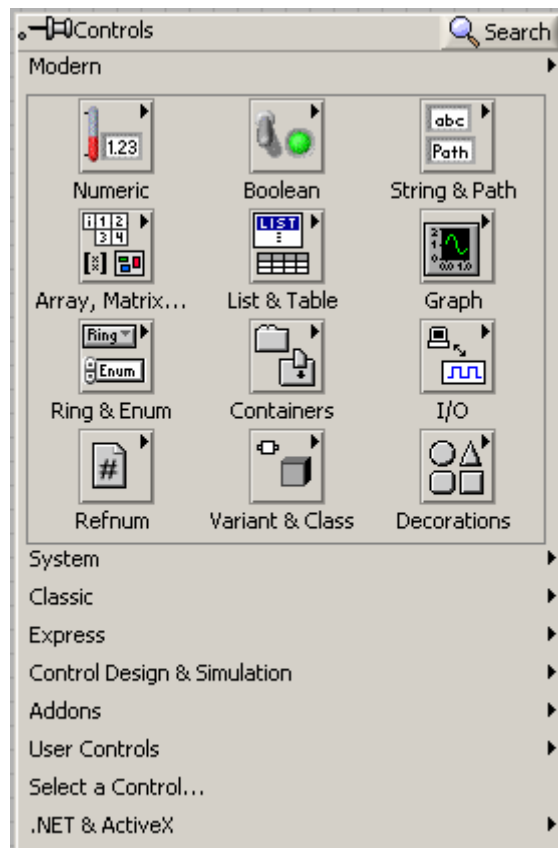


Figure 7.2 Controls Window

The controls window has all the terminals, divided and grouped into many types (Numeric, Boolean, Express, Graph etc.). A terminal can be found on the controls window in more than one place. The type of the terminal can be changed by modifying its properties.

Aligning, distributing, grouping, locking and resizing objects can be done in Labview just like any other windows based program.

Another interesting feature of Labview is the search option located in the upper right corner of the controls window. For example, if it is required to find an indicator, and it is not known where to look for it, an appropriate keyword can be entered in the search window to display a choice of indicators.

The terminals in Labview support all the variable types possible in any other conventional programming language such as “C”.

7.2 BLOCK DIAGRAM

Once the front panel is built (the inputs and outputs are decided), the next step is to add the code in the block diagram using graphical representation of functions. Front panel objects appear as terminals on the block diagram.

Figure 7.3 illustrates the corresponding block diagram for the Generate Sine Wave VI.

As shown, the entire source code is put in a *while* loop running continuously. The *while* loop breaks when stop button is pressed. The “Simulate Signal” VI takes the inputs Frequency and Amplitude. The output of the VI is a sine wave. This sine wave is branched into two components, one going to the waveform graph display and the other going as input to the ”Amplitude and Level Measurements” VI. The output of the VI is RMS Voltage.

Notice that all the terminals (inputs and outputs) of the code (Frequency, Amplitude, Waveform, RMS Voltage and Stop) are shown on the front panel. Double-clicking a block diagram terminal will highlight the corresponding control or indicator on the front panel.

Terminals are entry and exit ports that exchange information between the front panel and block diagram. Data entered into the front panel controls enter the block diagram through the control terminals. During execution, the output data flow to the indicator terminals, where they exit the block diagram, reenter the front panel, and appear in front panel indicators.

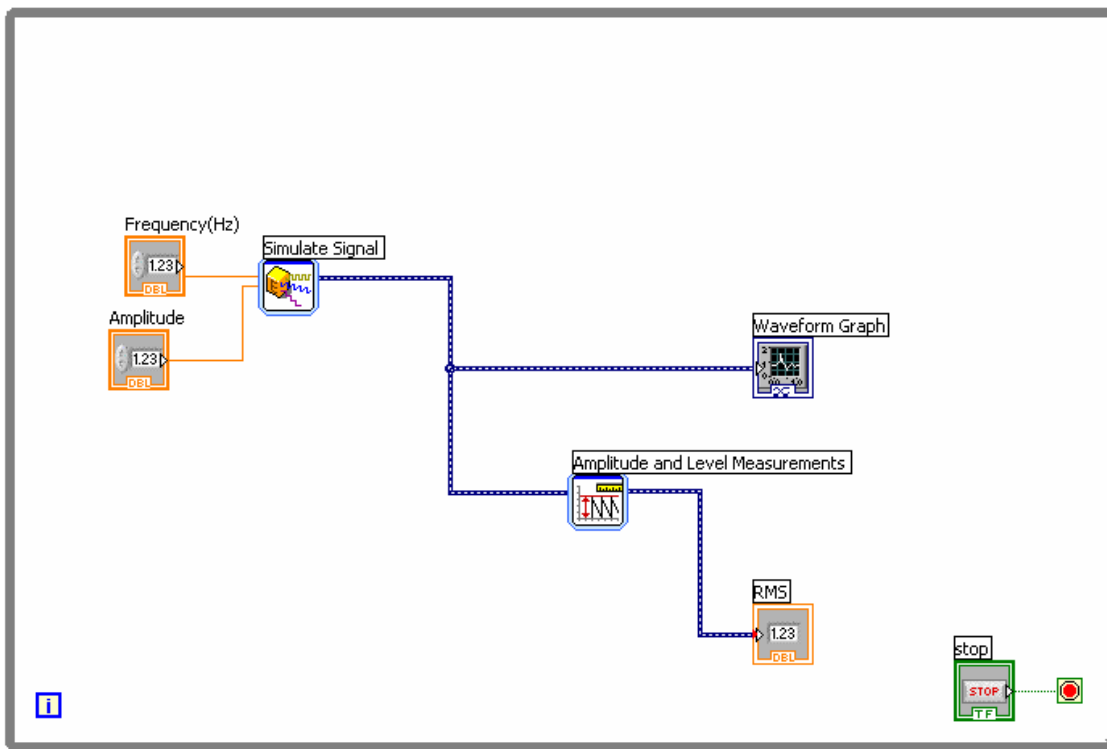


Figure 7.3: Generate Sine Wave VI – Block Diagram

Building a Block Diagram

Building a block diagram is very similar to building a front panel. As a practice example, we first decide on the inputs and outputs before writing a program. Similarly we decide what

should be on the front panel before developing the block diagram. In some cases, the opposite might make things easy. One can create a control or indicator by right clicking on the input or output of a VI and select create control or indicator to put the corresponding terminal (object) on the front panel. The advantage of using this is that all the properties (type, format and precision) of the object are already setup for use.

When the front panel is built, the block diagram would already have the terminals (in this case Frequency, Amplitude, Graph and RMS). The next step is to put in a function that will take in Frequency and Amplitude and produce a sine wave.

As in the front panel, right clicking anywhere on the white portion of the block diagram opens the functions window. The *functions* window is shown in Figure 7.4.

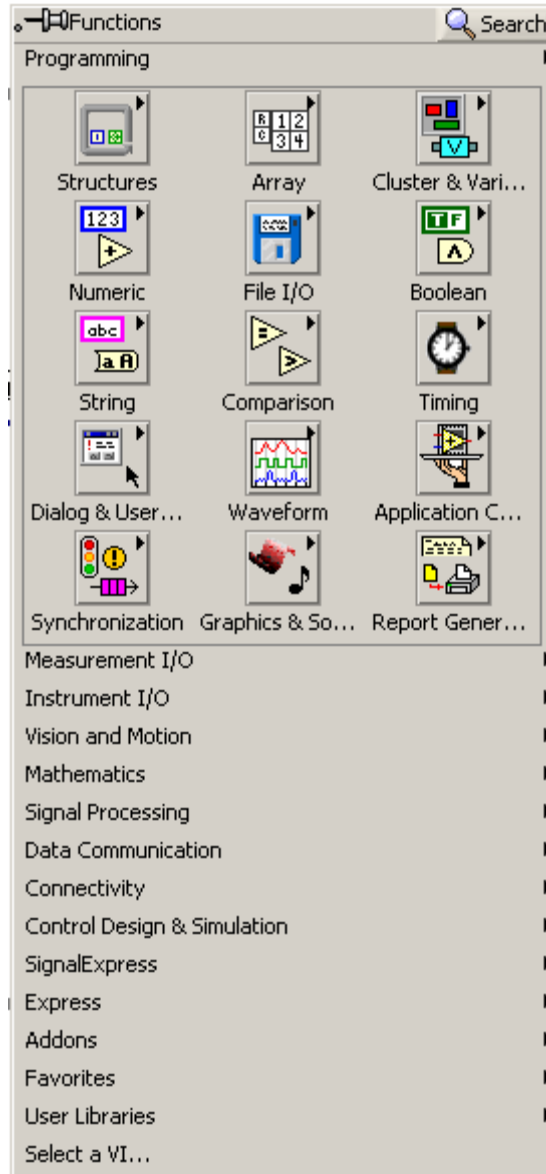


Figure 7.4: Functions Window

If it is known where to find a VI, one can find it manually and put on the block diagram. Otherwise we can make use of the search function. After putting in all the functions, the block diagram will look as shown in Figure 7.5.

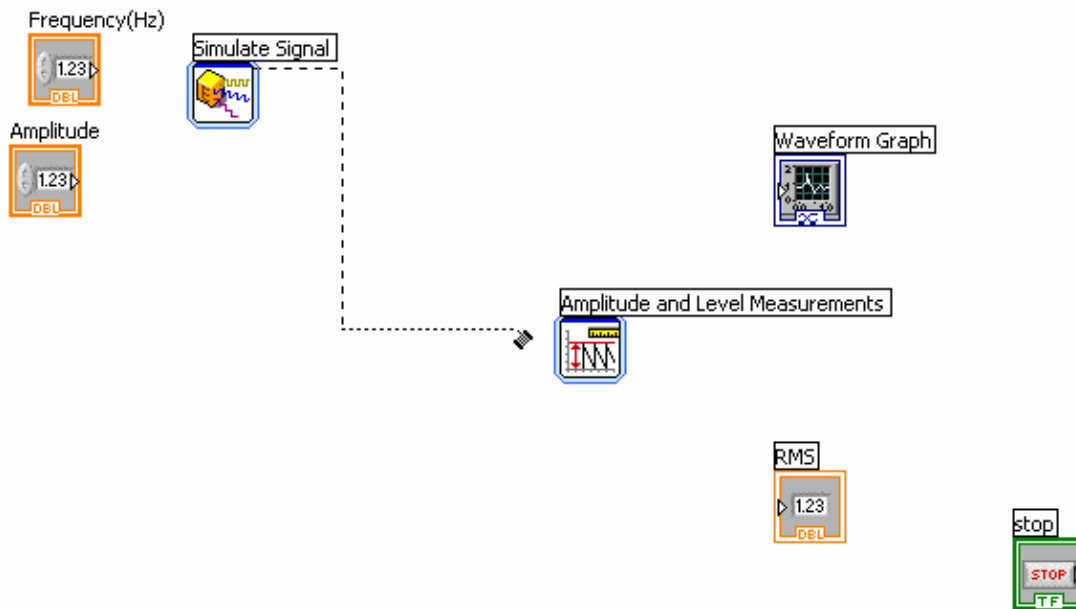


Figure 7.5 Generate Sine Wave VI – Block Diagram (Unwired)

Now it is necessary to connect (wire) the block diagram. The terminals with functions should be connected with wires because in Labview data are transmitted between functions through wires. Each wire has a single data source, but we can wire it to many VIs and functions that need to read the data. The color of the terminal is an indicator of its type which the user will get used to with experience. The user must wire all required block diagram terminals. Otherwise, the VI is broken and will not execute.

The wiring tool is used to manually connect the terminals on one block diagram node to the terminals on another block diagram node. The cursor point of the tool is the tip of the unwound wire spool. When the tip of the unwound wire is clicked on the terminal, it starts to unroll a broken wire. When the broken wire is connected to the correct input of a VI of compatible type, the wire becomes solid. The color of the wire is same as its terminal.

Labview also supports the structures of conventional programming languages like *while*, *for loop* etc. Labview also accepts ‘C’ code or Matlab code to be put into the block diagram. The structures in Labview are shown in Figure 7.6.

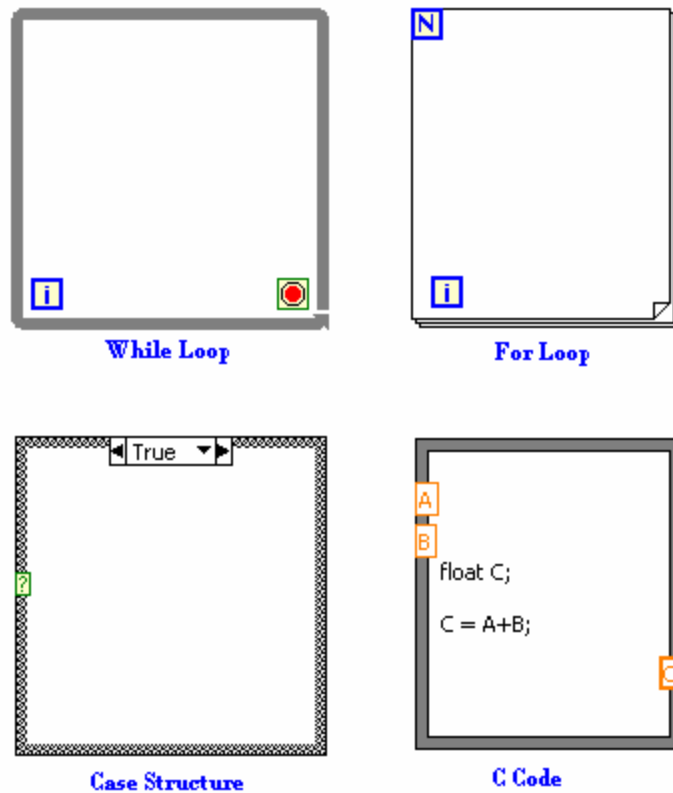


Figure 7.6 Structures in Labview

7.3 ICON AND CONNECTOR PANE

Icon is a graphical representation of the VI. It is always shown at the top right corner of the front panel window. Making an icon will make it easy to identify a particular VI when using it in another VI. After all, Labview is a graphical programming language.

For the Generate Sine Wave VI, if it is required to combine Simulate Signal and Amplitude and Level Measurements into a single VI called “Generate Sine and calculate RMS”, the first step is to define an icon. The user can edit icon using the “Icon Editor” window. Edit icon window is shown in Figure 7.7.

In a large design, a section of a VI can be converted into a Sub-VI by using the Positioning tool to select the section of the block diagram and selecting Create Sub –VI in the edit menu. An icon for the new Sub-VI replaces the selected section of the block diagram. Labview creates controls and indicators for the new Sub-VI and wires the Sub-VI to the existing wires.

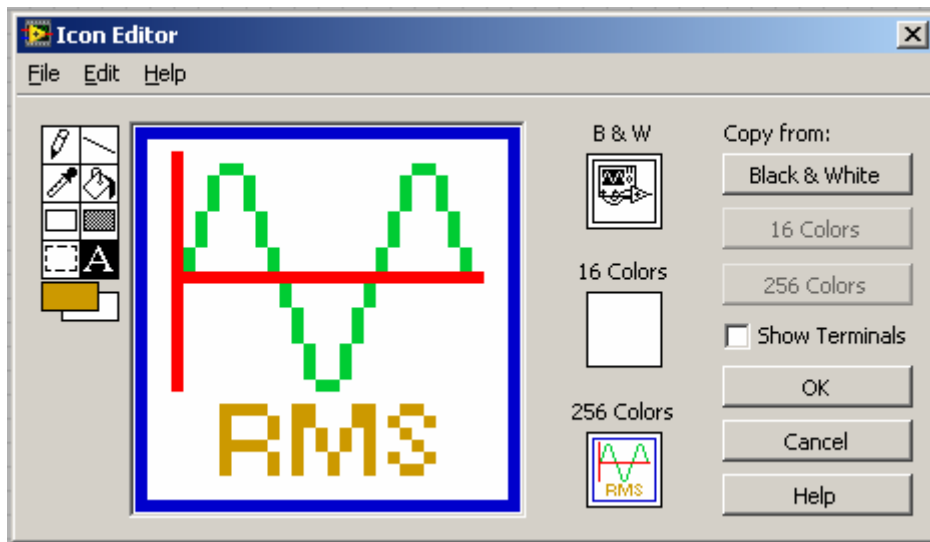


Figure 7.7: Icon Window

Once the new icon is defined, the user is required to define the input and outputs terminals of sub-VI. VI Connector can be used to define terminals of the sub-VI. A sub-VI is developed with terminals and icon as shown in Figure 7.8.

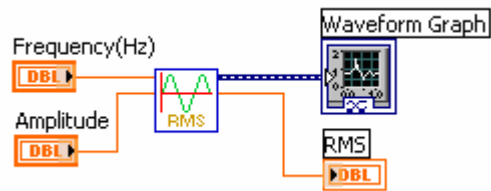


Figure 7.8: Generate Sine and calculate RMS

This VI can be used in other VIs just as functions in any other programming language. One must note that Labview does not allow a VI to call itself recursively.

8.0 WIRELESS DEVICES AND RFSG

Wireless devices that are to be tested are configured as shown in Figure 6.1. The direction of communication between the two wireless devices can be unidirectional (from transmitter to receiver) or bidirectional (both wireless devices have transmitter and receiver, and they exchange information both ways). There may be different standards defined for the transmitting device and the receiving device. The aim of the test setup is to test that the signal sent out of the device conforms to the standard.

Wireless devices include cellular phones, radio communication equipment, sensors, RFID tags and RFID readers. All equipment types that use the electromagnetic spectrum for communication come under wireless devices. Based on the frequency of carrier, different standards might apply to different wireless devices, but the test setup will be similar.

Most of the wireless devices receive and transmit information. When testing the signal from “wireless device 1”, as well as from “wireless device 2”, it is advisable to test each device separately. The Radio Frequency Signal Generator can be programmed to behave like the device not under test.

Another reason to use RFSG is because the flow of data between the wireless devices can be controlled. The RFSA can be triggered to capture the signal depending on the time when RFSG is triggered to transmit. When testing a series of data transfers, using an RFSG can provide sufficient time between each data transfer pair. An analysis can be done during this time before continuing with the next set of data transfers.

Many industry standard signal generators are available like NI5671, Rhodes and Schwarz SMJ100A and Keithley 2910. The testing programs developed in this thesis in Labview use the Rhodes and Schwarz SMJ100A. The signal transmitted is developed using software named WinIQSim.

8.1 RHODES AND SCHWARZ SMJ100A



Figure 8.1 Rhodes and Schwarz SMJ100A. Courtesy: Rhodes and Schwarz

The R&S SMJ100A can be used in generation of digitally modulated signals. The device uses I/Q (vector) modulation. Digital data streams are converted to an I/Q baseband signal. The baseband signal is then D/A-converted and modulated onto the desired RF carrier frequency with the aid of an I/Q modulator.

The front panel of the device is divided into in four different blocks. Generation of signals from RFSG requires configuring the four different blocks.

The four different blocks are:

1. Baseband configuration
2. AWGN/IMP configuration
3. IQ Modulation configuration
4. RF Configuration

The architecture of the RFSG and the signal flow are shown in a block diagram on the RFSG user interface. In the diagram, signal processing is performed from left to right. The function of each of these blocks can be summarized by the Table 8.1.

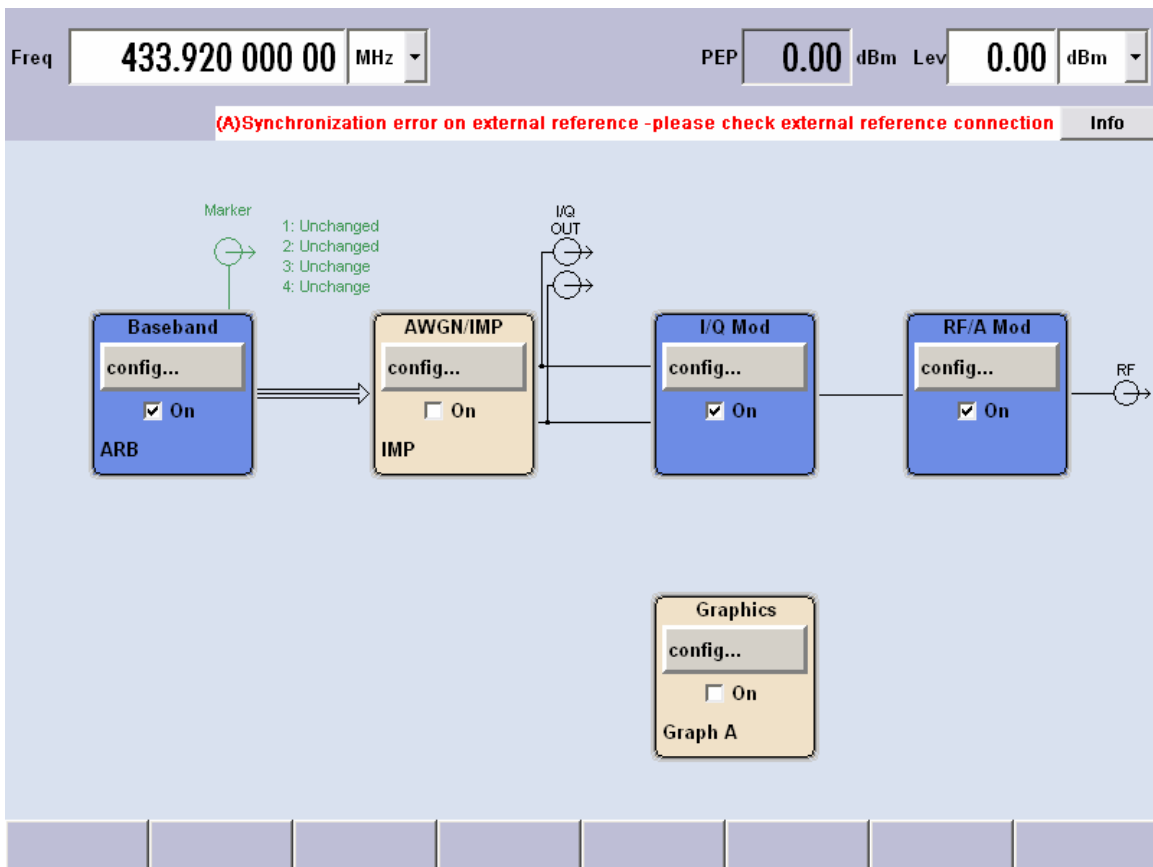


Figure 8.2: Four Blocks of SMJ100A

Table 8.1: Function of the four blocks in SMJ100A

Block	Function of Block	Status Display in the block	Effect of Toggle ON-OFF Switch
Baseband	This block is used to configure the baseband source and activate it	Selected modulation. Select waveform to transmit.	Switches the selected modulation (digital standard, digital modulation or ARB) on or off.
AWGN/IMP	This block is used to insert additive white gaussian noise and digital impairments into the generated signal.	Active functions of block.	Switches the active functions of the block on or off. The functions (AWGN, impairments or both) are activated in the respective menus.
I/Q Mod	This block is used to configure the I/Q modulator. I/Q impairments can be selected and activated.	Analog I/Q impairments, I/Q swap.	Switches the I/Q modulator on or off.
RF/A Mod	This block is for configuring and activating RF signal, analog modulations, list mode and sweeps.	Active analog modulation modes.	Switches the RF signal on or off.

Baseband Configuration Block

The baseband section of the R&S SMJ is fully digital and contains the hardware for generating and processing I/Q signals. The three sub menus in the configuration of Baseband block are:

1. Custom Digital Modulation
2. Multi carrier CW
3. ARB

An ARM menu is used when it is required to transmit an arbitrary waveform. This is the menu used when transmitting a waveform that is made in WinIQSim. Figure 8.3 illustrates the ARB sub menu.

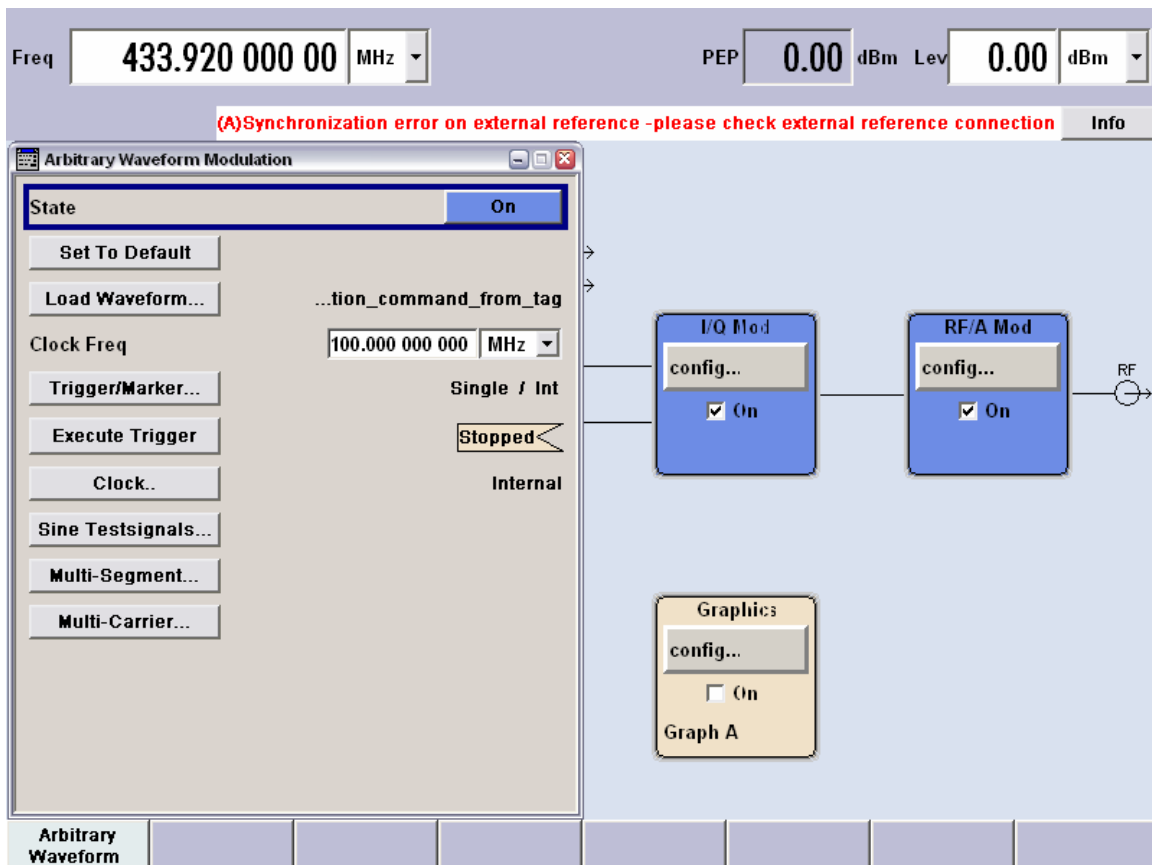


Figure 8.3: ARB Sub Menu

The “State” on top of the menu indicates if the block is active or not. There is a flag that displays “Stopped” in the figure. When the system is transmitting the signal, this flag will change to indicate “Running”.

The “Load Waveform” button is used to load a waveform. All files with extension “.wv” can be loaded. When a waveform is loaded, its name is shown next to the button.

The next important sub menu is the Trigger/Marker menu. Figure 8.4 illustrates the Trigger/Marker menu.

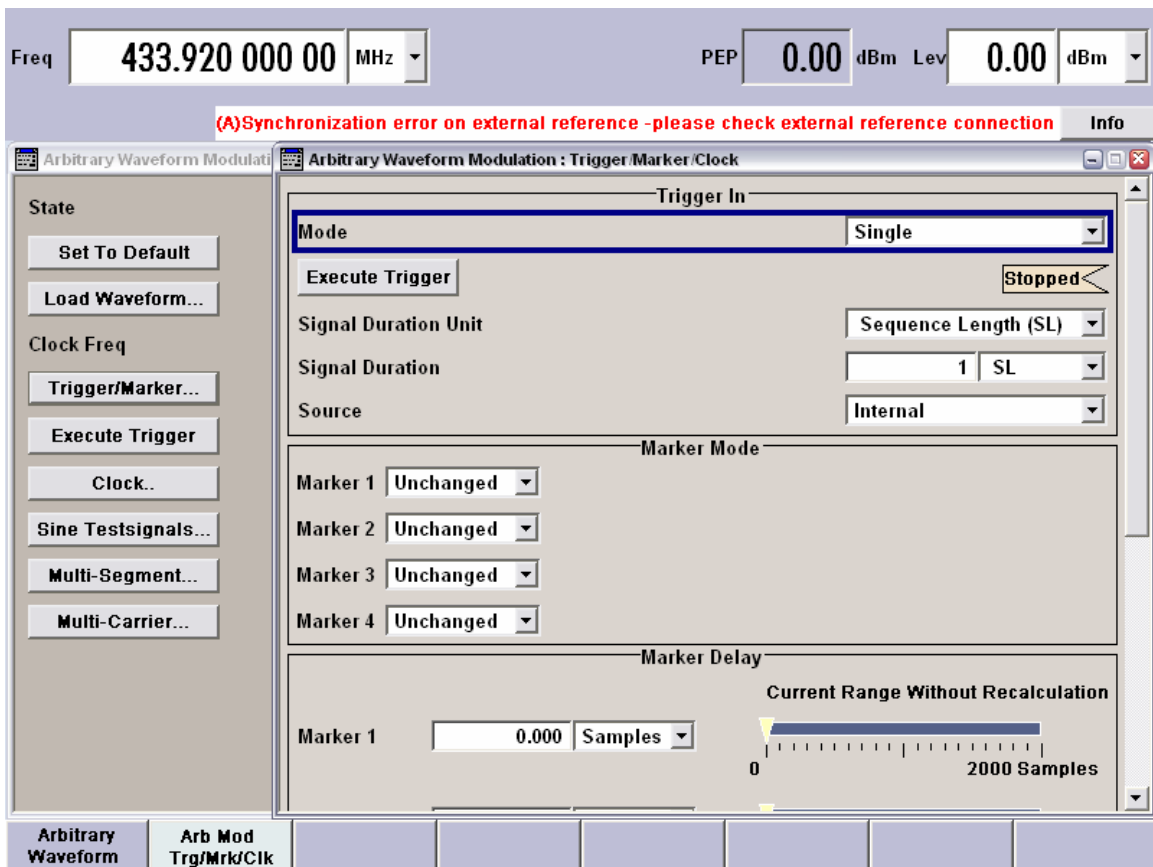


Figure 8.4: Trigger/Marker Menu

The first selection in this menu is the mode. There are five modes available – Auto, Retrigger, Armed auto, Armed Retrigger, Single. The default mode is “Auto”. In this mode the loaded waveform is repeated until RF is switched off.

The Single mode is used for transmission of a signal only once. The Signal Duration unit is set to Sequence Length, which means the system will transmit the entire sequence of the waveform loaded only once. Signal duration is set to 1 SL. This setting is one advantage with the SMJ100A. When transmitting a waveform that is only milli seconds long, it takes a short time (in seconds) to load this waveform. However, when transmitting a long signal (for example a wakeup signal for 2.5 seconds) the time taken to load the signal will be long and there is not enough memory to load all the waveform data. Thus the signal can be generated for a shorter time (10ms) and make the device transmit the same signal for many times (250 times) to form the longer signal (a wakeup of length 2.5 seconds).

AWGN/IMP Block

This block is used to generate realistic signals added with noise generated from the transmitter itself and also added by the channel. But in testing wireless devices it is required to test the response of the device under test to the signal generated by the RFSG. This block is not really useful while performing conformance tests. This block is turned OFF during the tests.

IQ modulation Block

The I/Q modulator is configured in this block. The three sub menus internal to configure IQ Mod are:

1. IQ Settings
2. Analog Wideband IQ In
3. Internal Baseband IQ In

All the IQ settings can be set in their default state. When generating an arbitrary waveform, “Internal baseband IQ In” should be selected.

RF/A Configuration Block

The parameters of the carrier (center frequency, maximum carrier offset and amplitude of carrier) are configured in this block. But as can be seen in Figure 8.2, carrier frequency and amplitude can also be set from the front display panel itself.

All these settings need not be done manually. Labview software can be used to remotely configure the RFGS to transmit signals. The test VIs developed can configure the RFSG as specified by the user.

8.2 REMOTE CONFIGURATION OF RFSG SMJ100A

RFSA SMJ100A can be remotely controlled using Labview. This section describes the sub VIs that are useful when configuring the device through Labview.

RSSMU Initialize

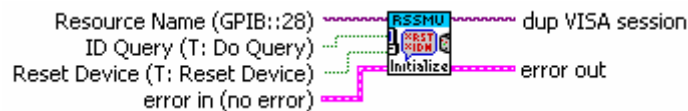


Figure 8.5: RSSMU Initialize VI

When writing a Labview program for SMJ100A, this is the first VI that should be used. This VI opens a session to the Default Resource Manager resource and a session to the specified device using the interface and address specified in the Resource Name control. In other words, it tells the RFSG that someone wants to communicate with it remotely.

RSSMU Wait to Continue



Figure 8.6: RSSMU Wait to Continue VI

WAIT-to-CONTINUE does not allow subsequent commands to be processed until all previous commands have been executed and all signals are in their transient condition. This is a very important VI because the GPIB bus is very slow, and it is very difficult to tell how much time it would take for a command to reach the RFSG and be executed by it. The next command should not be sent until the present command reaches the RFSG and is executed.

RSSMU Write From File To Instrument



Figure 8.7: RSSMU Write From File To Instrument VI

This VI is used to read data from the host computer and write it to a user specified file in the instrument. It is this VI that writes the “.wv” file to the RFSG SMJ100A.

RSSMU ARB Waveform Select

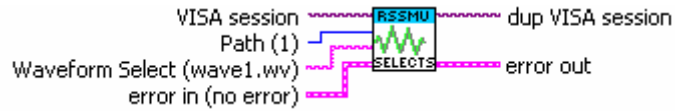


Figure 8.8: RSSMU ARB Waveform Select

This VI selects the waveform file. Only files with the file extension “.wv” will be created or loaded.

R&S SMJ100A Single Mode – Sub VI



Figure 8.9: R&S SMJ100A Single Mode – Sub VI

This VI configures the mode of transmission. It will set the mode to single and will also set the signal duration in SL, as specified by the user.

RSSMU Set ARB State

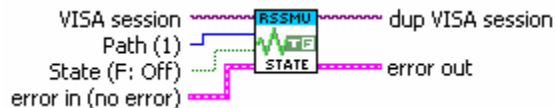


Figure 8.10: RSSMU Set ARB State VI

This VI switches the ARB generator ON. Any other standards or digital modulation that may be in the ON state are automatically turned OFF.

RSSMU Set RF Amplitude

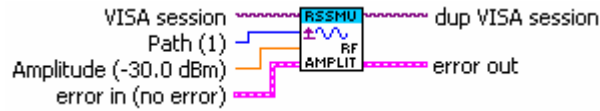


Figure 8.11: RSSMU Set Amplitude VI

This VI sets the RF output level in CW mode.

RSSMU Set RF Frequency

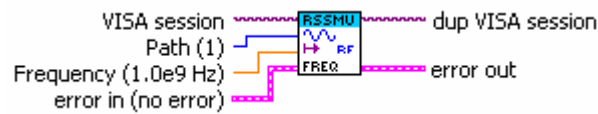


Figure 8.12: RSSMU Set RF Frequency

This VI sets the frequency of the RF output signal for CW mode. In other words, it sets the carrier frequency.

RSSMU Set Output State

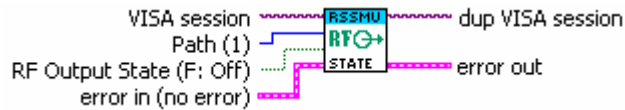


Figure 8.13: RSSMU Set Output State

This VI activates or deactivates the RF output.

RSSMU ARB Execute Trigger



Figure 8.14: RSSMU ARB Execute Trigger

This VI executes a trigger. It makes the RFGS send out the signal.

RSSMU Close

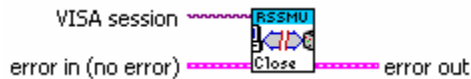


Figure 8.15: RSSMU Close

When writing a Labview program for SMJ100A, this program needs to be the last VI that should be used. It tells the RFSG that remote communication has been ended.

8.3 WINIQSIM

WinIQSim is the software that is used to generate the “.wv” files that are loaded into SMJ100A. When WinIQSim is started it will load all the setting last used. The welcome page is shown in Figure 8.16.

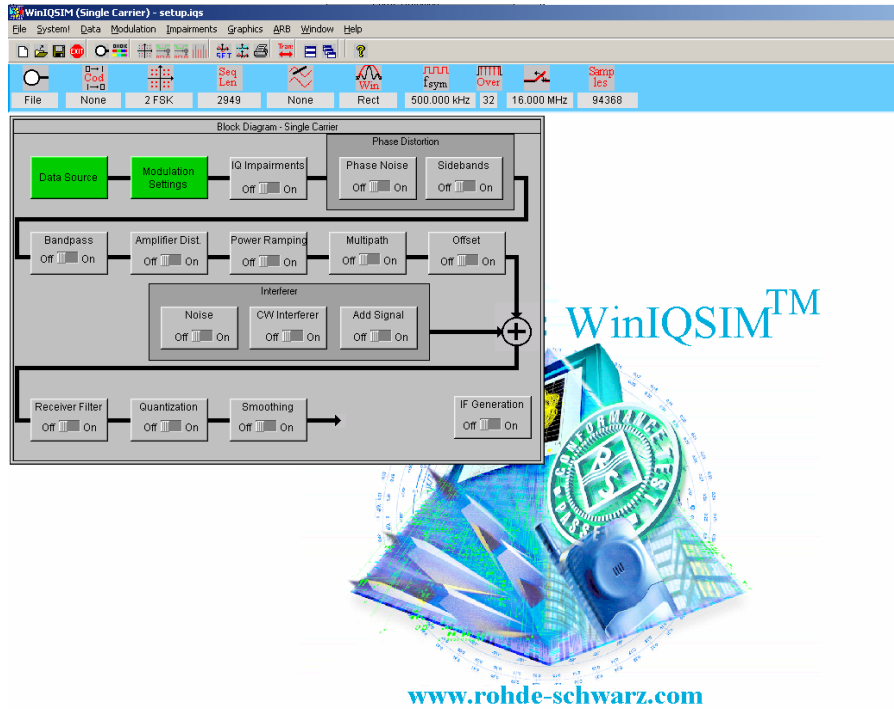


Figure 8.16: WinIQSim welcome page

The generation of signals is done by configuring each of the blocks shown. The two most important blocks are:

1. Data Source Block
2. Modulation Settings Block

Those two blocks are sufficient to generate the necessary waveform.

Data Source Block

Clicking the data source block will open the following window.

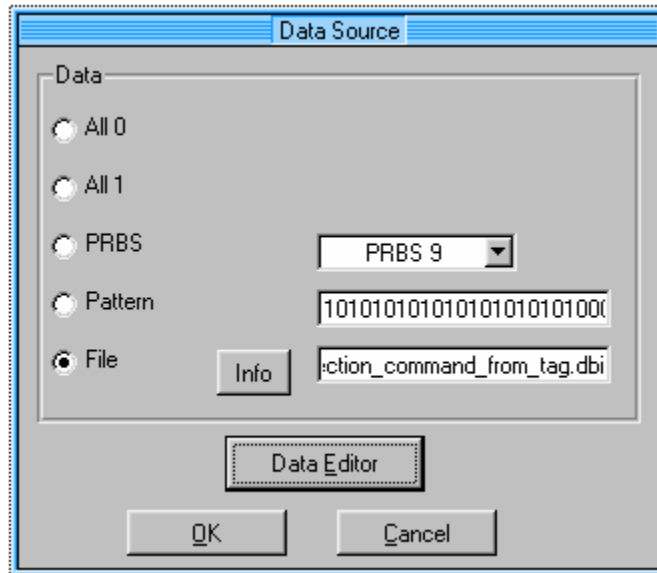


Figure 8.17: Data Source window

The Data Editor is selected to organize the baseband data to be transmitted. The following window shown in Figure 8.18 opens when Data Editor is selected.

This is a very easy to use interactive window used to make the digital data that has to be transmitted after modulation. Waveform generation is gone in five steps.

1. Configure data fields
2. Configure Slots
3. Configure Frames
4. Configure power ramp settings
5. Calculate and save sequence

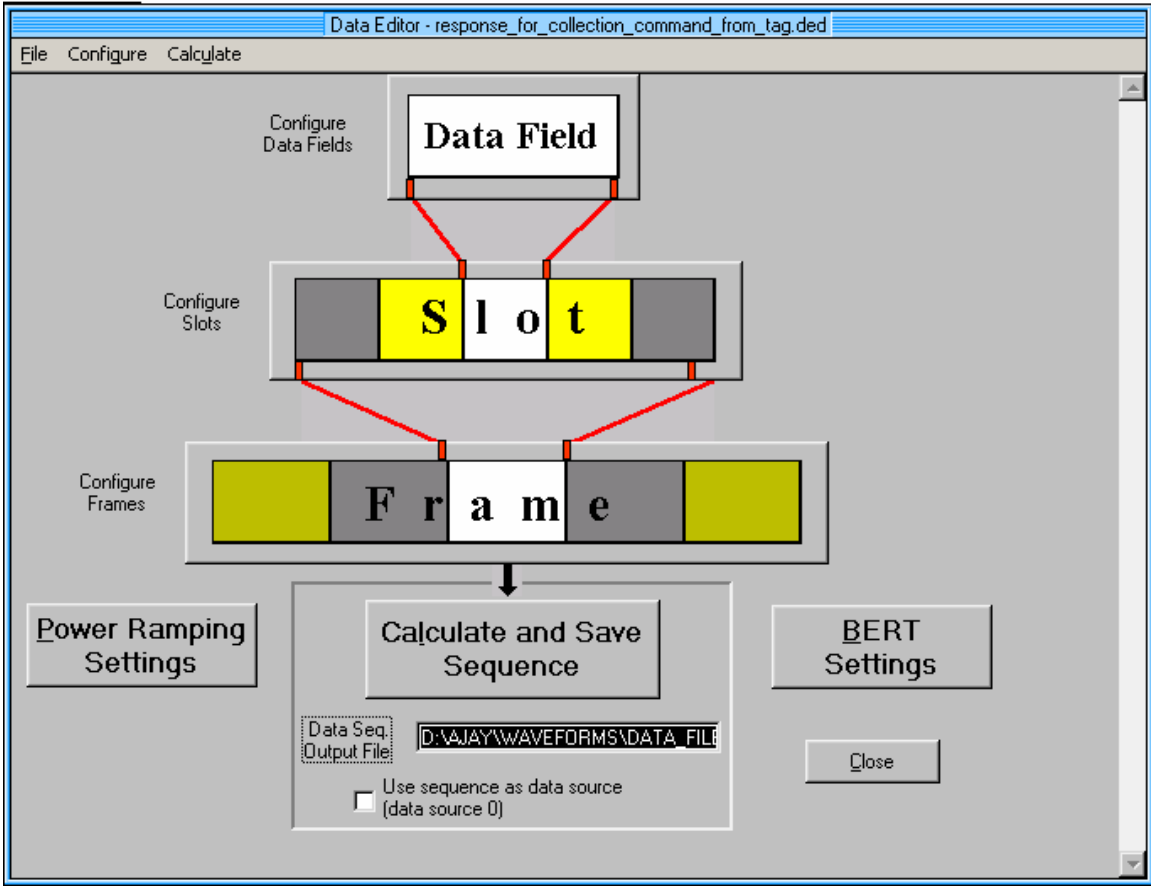


Figure 8.18: Data Editor Window

The Table 8.2 below introduces each of the steps.

Table 8.2: Working with Data Editor Window

Data Fields	Selecting this menu item opens the Configure Data Fields panel where the basic components of the data sequence can be generated.
-------------	--

Table 8.2 (continued)

Slots	Selecting this menu item opens the Configure Slots panel. Here the data structures of the different time slots can be prepared with the previously defined data fields. In this panel also the power/time templates of the slots can be configured and the markers output with the signal in the AMIQ can be set.
Frame	Calls the Configure Frame panel. One or several consecutive data frames are created in this panel with the aid of the defined slots. The frame length is calculated and shown here.
Power Ramping	The Power Ramping Settings panel is displayed. The basic parameters of power control such as ramp type and duration as well as the various signal levels in the on and off state can be set in this panel.
Calculate and save sequence	Save sequence Selecting this menu item triggers a calculation of the data sequence and of power ramping with the settings defined. Before calculation is started, a file selection window is opened where a file with the fixed extension *.dbi (data bits) has to be specified for saving the calculated data sequence. The calculated data sequence may also be used as a data source for I/Q calculation. It is this file that is loaded into data source window (see figure 8.6).

Once the data source block is configured the next step is to configure modulation settings.

Modulation Settings

The modulation settings block is relatively easy to understand. It is shown below in Figure 8.19.

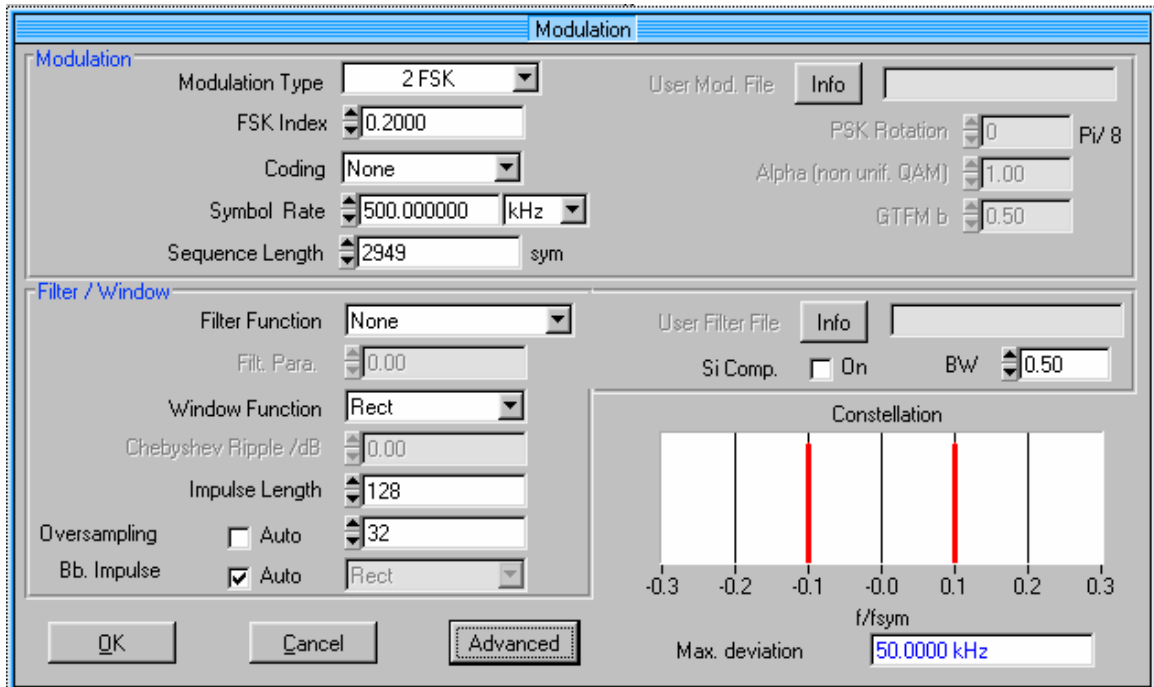


Figure 8.19: Modulation Settings

The Table 8.3 below explains each of the inputs. Only those inputs required for generating a FSK modulation are discussed.

Table 8.3: Modulation Settings

Modulation Type	This is the most important parameter of the panel. The selected type is indicated in the Modulation Type field. The modulation types possible are FSK, PSK and QAM and Dirac Impulse. Sending out the Dirac Impulse and analyzing the response of the receiver gives its impulse response.
-----------------	--

Table 8.3 (continued)

FSK Index	This parameter can only be defined for 2FSK and 4FSK modulation. It is displayed instead of the reference level. With the FSK or modulation index, the frequency deviation for FSK can be increased or reduced. An index between 0.1 and 12 can be set. Sufficiently high over sampling has to be selected for the selected FSK index
Coding	The binary data stream from the source can be recorded with the aid of a coder. The coding types supported by WinIQSim do not contain redundancies. The number of coder input bits corresponds to the number of output bits. Block coding is mainly used to adapt the mapping schemes supported by WinIQSim to the requirements of the respective communications standards. In most cases the coder only reorganizes the assignment of data symbols to the modulation states. WinIQSim does not contain elements like channel coding or interleaving. These procedures can however be performed via the data source File.
Symbol Rate	<p>The symbol rate must be adjusted so that a correct clock rate can be set for the I/Q source AMIQ. This setting is performed in the Symbol Rate field. The symbol rate is obtained by dividing the bit rate by the modulation level. The modulation level corresponds to the number of bits in the input stream from the data source transmitted in one symbol.</p> <p>Maximum Symbol Rate = 100MHz / Over Sampling</p>
Sequence Length	The sequence length specifies the length in data symbols of the I/Q signal to be calculated. A length between 1 and 4000000 can be set in the Sequence Length field. The sequence length is calculated in the frame panel of data editor window.

Table 8.3 (continued)

Filter Function	This section describes the selection of the baseband filter and of associated filter parameters. All baseband filters offered by WinIQSim are lowpass filters. The various communication systems use different bandpass filters. The most frequently used filter types can be selected here. To be able to use other filters, a user-specific filter can be defined with the aid of a file.
Window Function	The Window Function is an additional filter that can be placed over the impulse response for a period determined by the impulse length. This window function is simply an additional weighting factor for the coefficients of the baseband filters.
Impulse Length	The pulse length determines the number of symbol periods over which the filter impulse response is considered in the I/Q calculation. The pulse length has a considerable effect on the modulation accuracy.
Over sampling	Oversampling defines the number of samples to be calculated for each symbol. This factor helps to increase the frequency offset of the aliasing products produced by D/A conversion of the signal.
Baseband Impulse	In this field the baseband pulse for filter excitation can be selected. A Dirac pulse, a rectangular pulse or Auto can be selected.

SMU Waveform Transmission

The SMU Waveform Transmission window can be configured to build and save the “.wv” file. The Table 8.4 discusses the different inputs to the SMU Waveform Transmission window.

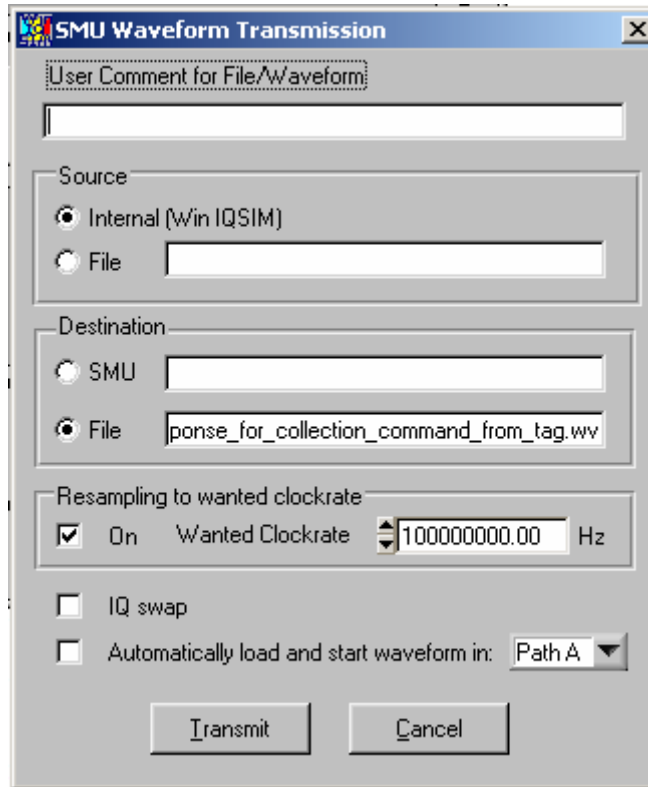


Figure 8.20: SMU Waveform Transmission window

Once the “.wv” file is built, it can be copied to RFSG by using a flash drive or we can load it remotely using Labview through GPIB or ETHERNET cable.

Table 8.4: SMU Waveform Transmission settings

Source	In this window a selection can be made whether the IQ signal defined by WinIQSim (Internal) or a signal file (File) previously stored on the hard disk of your PC should be transmitted. In the File selection field the currently selected source file is displayed. To select another source file click into the field with the left mouse key. A file selection
--------	--

Table 8.4 (continued)

	<p>window is opened where a file with the extension *.wv (waveform) can be selected for transmission.</p>
Destination	<p>In this window the destination of the transmission is set. The selected waveform can be directly transmitted to the SMU (SMU). There it is stored under the name entered in the field to the right.</p>
Resampling to wanted clockrate	<p>When this option is active, the waveform is being resampled before its transmission to SMU.</p>

WinIQSim has a blue indicator bar at the top of its main window. It is very useful as it shows the properties of the signal built.



Figure 8.21: WinIQSim Indicator bar

9.0 RADIO FREQUENCY SIGNAL ANALYZER

The Radio Frequency Signal Analyzer captures the communication between the wireless devices. It acts as a transducer, taking in the RF energy and converting it into digital data that can be processed and analyzed by the computer.

There are many RFSAs available in market like the NI5660 and Keithley 2810. There are RFSAs which have a wide frequency range of capture, and there are some made specifically to capture signals in a particular band of frequencies. Another major specification to look at while selecting a RFSA is the Resolution Bandwidth. The Resolution Bandwidth is related to the minimum sampling rate possible.

Most RFSAs give their output in IQ data format. This can be loaded into Labview or any other similar software for processing, demodulation and analyzing. The RFSA used in the test setup is NI5660. This is a product of National Instruments.

Key Terms in RFSA

Table 9.1: Key Terms in RFSA

Digitizer	A device that converts analog voltages into digital representations and stores the results in memory; refers to the NI 5620 IF digitizer hardware module.
Downconverter	A device that frequency-translates RF signals to center around a specified IF frequency; refers to the NI 5600 3-slot RF downconverter hardware module.

Table 9.1 (continued)

Intermediate Frequency (IF)	Refers to the signal passed from the NI 5600 RF downconverter module front panel OUTPUT connector to the NI 5620 IF digitizer module front panel INPUT connector
Module	Refers to one of the hardware components of the RF Signal Analyzer

9.1 NI – 5660 RFSA

The NI – 5660 is an industry standard RFSA from NI, with a very wide frequency range. It can capture signals with frequencies ranging from 9kHz to 2.7GHz. It has a wide adjustable Resolution Bandwidth from less than 1Hz to 10MHz. It has a 10 MHz oven-controlled crystal oscillator (OCXO) timebase. The full signal input range is 30dBm. It also has a 64MB built in memory. NI – RFSAs can be connected to the computer using a PCI bus which allows very high speed data transfers on the order of MBs.

The NI – 5660 is divided into two modules; the NI – 5600 which is the downconverter module and the NI – 5620 which is the digitizer module. The RF signal is taken as input by the downconverter. The NI 5600 downconverter module performs two primary functions. They are frequency shifting, or down conversion, and input signal conditioning. Frequency shifting is performed using a tunable oscillator in the superheterodyne signal chain. Input signal conditioning is accomplished using three stages of mixer conversion and two sets of gain attenuators whose levels are programmable. The first set of attenuators can be set to minimize distortion and other spurious signals when input levels are high and to minimize noise when input levels are low. The second set of attenuators is in the second intermediate frequency path before the third mixer and ensures an appropriate output signal level even if the first mixer is intentionally driven into compression. These attenuators are also set when performing linearity measurements. The NI 5620 is an IF high-speed digitizer module featuring a 14-bit analog-to-digital converter (ADC) with deep onboard sample memory. Offering sample rates of 1 kS to 64

MS/s and distortion-free performance, it complements the NI 5600 RF downconverter module in analysis applications. [Adapted from NI – RFSA Help]

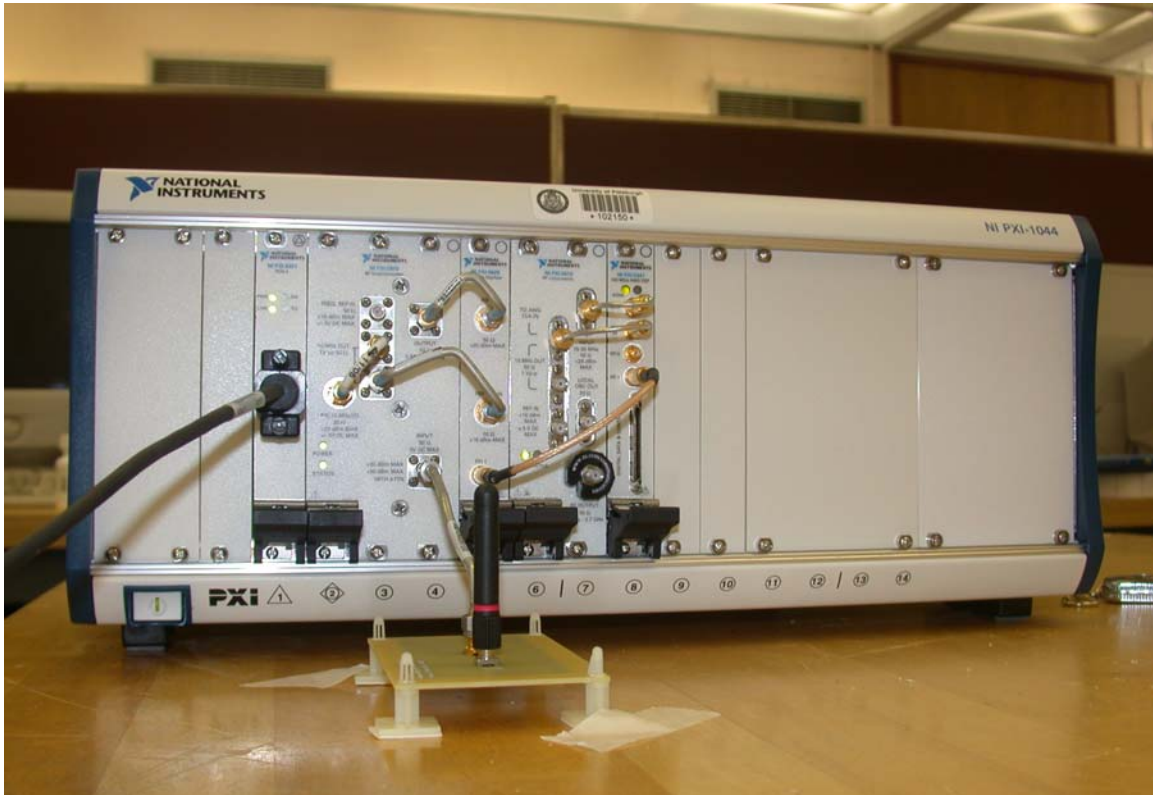


Figure 9.1: NI – 5660 (NI – 5600 + NI – 5620)

NI 5660 Signal Paths

A signal takes the following path from the RF Signal Analyzer wireless antenna to the PXI controller:

1. A signal enters the RF Signal Analyzer through the INPUT front panel connector of the NI 5600 RF downconverter module.

2. The NI 5600 RF downconverter module "zooms in" on a 20 MHz block of spectrum and frequency-translates it to center around 15 MHz. The translated IF signal is sent to the NI 5600 downconverter module OUTPUT connector.
3. The IF signal is passed from the NI 5600 RF downconverter module front panel OUTPUT connector to the NI 5620 IF digitizer module front panel INPUT connector.
4. The NI 5620 IF digitizer module filters and conditions the signal and applies gain and dither.
5. The Analog-to-Digital Converter (ADC) converts the signal from analog to digital.
6. The data are sent to onboard memory.
7. The data are transferred to the host computer. [Adapted from NI – RFSA Help]

Configuring NI – 5660

Unlike the SMJ100A, the NI – 5660 can only be configured using Labview. Now we will see the important VIs used in configuring NI – 5660.

ni5660 Initialize

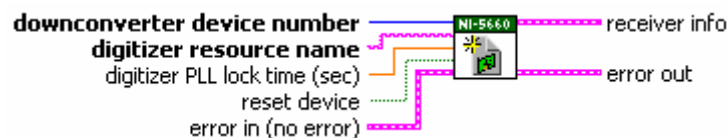


Figure 9.2: ni5660 Initialize VI

This VI creates a new instrument driver session to the RF signal analyzer, using the downconverter device number and the digitizer resource name specified. It sends initialization commands to reset both hardware modules to a known state necessary for NI-RFSA operation. Then it sets the RF downconverter module onboard clock as the RF Signal Analyzer reference

clock source. When all the PLLs lock correctly, the STATUS light on the RF downconverter module is activated. When programming NI – 5660, this VI must always be used first.

ni5660 Configure for IQ

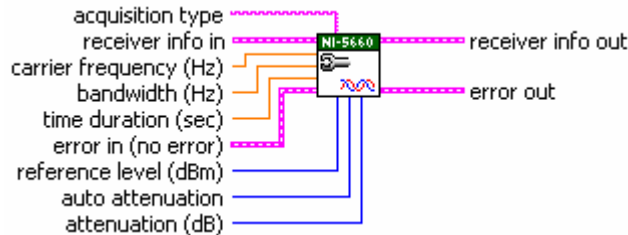


Figure 9.3: ni5660 Configure for IQ VI

This VI configures the RF Signal Analyzer hardware to acquire a time-domain signal with IQ settings you specify. Carrier frequency, bandwidth, and acquisition time settings define the IQ data. You can configure the acquisition to be finite or continuous. We use continuous acquisition type when we want to do streaming. This VI configures the reference level and attenuation settings used by the RF downconverter module, and the horizontal settings and acquisition type used by the IF digitizer module.

ni5660 Read IQ



Figure 9.4: ni5660 Read IQ VI

This VI returns the waveform acquired by the RF Signal Analyzer. This VI is used when acquiring a finite signal which can be stored in the buffer memory of the device. This VI initiates an acquisition and returns both a scaled voltage waveform with timing information, and

IQ data computed through a process of filtering, decimation, and downconversion. It is this IQ that is taken as input by the PC and used for demodulation.

ni5660 Initiate IQ



Figure 9.5: ni5660 Initiate IQ

This VI initiates an IQ waveform acquisition. This does roughly half the work done by ni5660 Read IQ. This is used when doing continuous acquisition.

ni5660 Fetch IQ

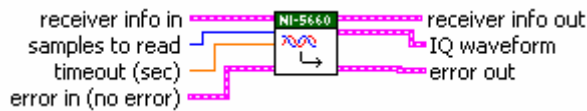


Figure 9.6: ni5660 Fetch IQ VI

This VI returns the IQ waveform acquired by the RF Signal Analyzer during a previously initiated acquisition. This VI returns IQ data computed through a process of filtering, decimation, and downconversion. This VI will finish the work started by ni5660 Initiate IQ. This VI is programmed in a while loop for continuous acquisition.

ni5660 Close

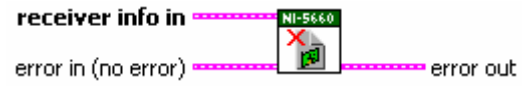


Figure 9.7: ni5660 Close VI

This VI aborts any signal acquisition in progress, terminates the instrument I/O session, destroys the instrument driver session and any attributes. We need to use this VI to end every session with the ni5660.

10.0 DEMODULATOR

As mentioned before, most all wireless devices use some form of modulation for various reasons. Once the signal has been captured, it needs to be demodulated. We may use hardware or software to demodulate the signal.

10.1 DEMODULATION IN LABVIEW

Labview supports almost all modulation and demodulation schemes. We will see the three most popular analog and digital demodulation schemes frequently used for RFID communication.

ASK Demodulation

The following VI is used to demodulate an ASK modulated complex baseband waveform and returns the time-aligned oversampled complex waveform, the demodulated bit stream, and the results of offset and drift measurements. This VI attempts to remove carrier and phase offset by locking to the carrier signal.

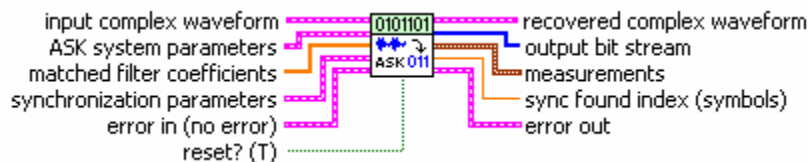


Figure 10.1: ASK Demodulation VI

The input complex waveform is the IQ data captured by RFSA. It is given as input to all the demodulator VIs.

FSK Demodulation

The following VI demodulates an FSK modulated complex baseband waveform and returns the time-aligned demodulated waveform, the demodulated information bit stream, and measurement results obtained during demodulation. This VI attempts to remove carrier and phase offset by locking to the carrier signal.

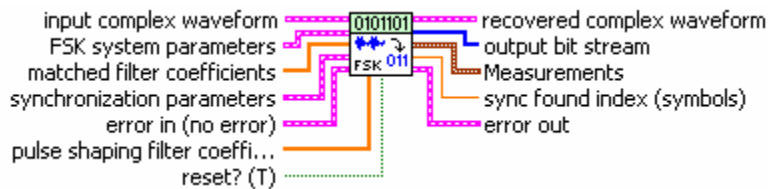


Figure 10.2: FSK Demodulator VI

PSK Demodulation

The following VI demodulates a PSK modulated complex baseband waveform and returns the time-aligned oversampled complex waveform, the demodulated bit stream, and measurement results obtained during demodulation. This VI attempts to remove carrier and phase offset by locking to the carrier signal.

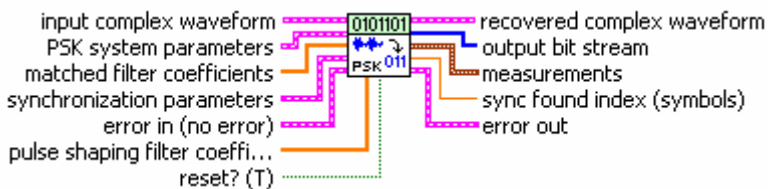


Figure 10.3: PSK Demodulator VI

Demodulate AM

The following VI demodulates an amplitude-modulated signal input to the AM modulated waveform input with optional suppression of the carrier wave. The demodulated signal is returned in the AM demodulated waveform output.

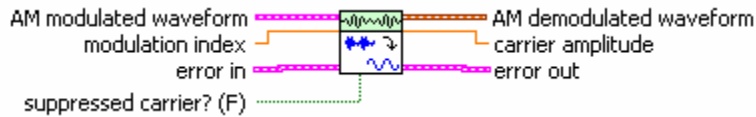


Figure 10.4: Demodulate AM VI

Demodulate FM

The following VI performs frequency demodulation on the incoming IQ signal. The recovered information signal is returned in the FM demodulated waveform output.

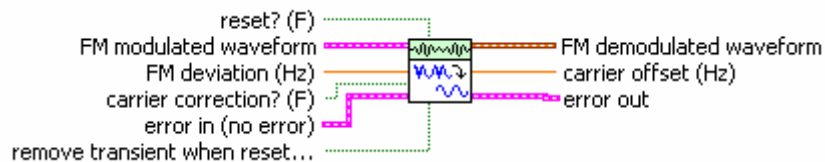


Figure 10.5: Demodulate FM VI

Demodulate PM

This VI Performs phase demodulation on the IQ signal.

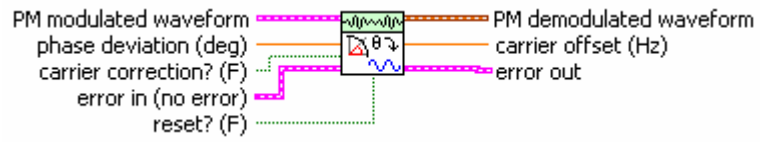


Figure 10.6: Demodulate PM VI

11.0 UPSAMPLER

Most standards require that the analysis of the signals should be done at a particular sampling rate. Capturing the signal at such very high sampling frequency presents two problems.

1. Data transfer time between RFSA and the Computer
2. Insufficient Memory in RFSA buffer

Thus, the obvious solution is to capture the signal at a reasonable rate (Niquist Sampling Frequency) and later upsample it. Again, implementing upsampling in hardware is very expensive, and it is preferable to do it in software.

11.1 UPSAMPLING IN LABVIEW

Labview offers many built in VIs for upsampling a signal. The VI used in this thesis is called “Resample Waveforms”.

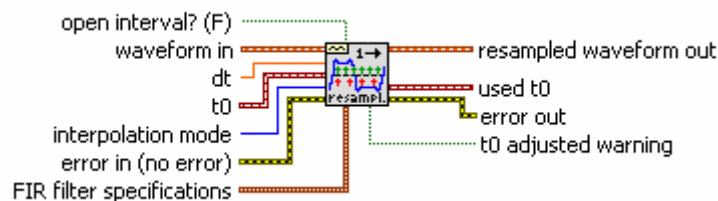


Figure 11.1: Resample VI

This is the simplest of all Resampling VIs pre-programmed in Labview. The VI resamples the input waveform at the “dt” specified.

After Resampling, it is good to pass the waveform through a median filter to remove any distortion caused by Resampling. The VI is called “Median Filter”.

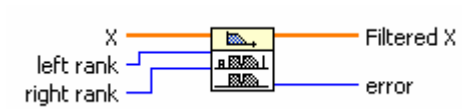


Figure 11.2: Median Filter VI

12.0 ANALYZER

The analyzer is where the signal is tested that it conforms to the standard. Testing is mostly done in software. It mainly consists of a series of comparative operations.

Labview as Analyzer

Labview is a good choice when analyzing and manipulating signals. It offers many built-in VIs to analyze signals. The output of the demodulator is a waveform signal. A waveform signal can be viewed as a structure in 'C' language (Non – Homogenous collection of data). A waveform consists of:

1. Start of waveform
2. Sampling period
3. Array of numbers.

In the case of demodulated FM signal, the array of numbers signify the frequency deviation of the baseband data. Using the array of numbers, we can write a 'C' code (which is the best way to work with arrays) to do a series of comparative operations.

Labview also has built-in VIs for pulse width measurements of periodic signals. This is a very useful VI, as most tests for conformance check for pulse widths to be within defined values.

Pulse Measurements

This VI accepts a periodic waveform or an array of periodic waveforms and returns the period, pulse duration (pulse width), duty cycle (duty factor), and pulse center of a selected pulse and period.

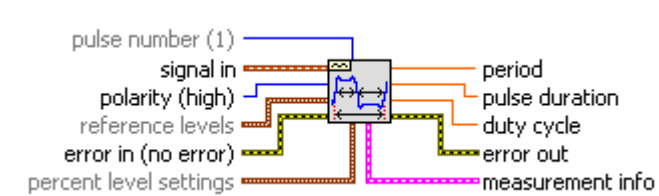


Figure 12.1: Pulse Measurements VI

13.0 SYNCHRONIZATION

Synchronization is essential between all the hardware devices. The RFSA must know when the RFSG is sending out the signal and when it should start capturing signals. Synchronization can be done in hardware as well as in software. Hardware synchronization is much more accurate. However software synchronization is cost effective and also easily programmable.

13.1 SYNCHRONIZATION IN THE TEST SETUP FOR ISO 18000-7

In the test procedures developed for verifying the ISO 18000-7 standard, software synchronization was used. This is because the NI5660 RFSA has no hardware trigger input. When we initialize the setup of ni5660, it may take anywhere between 10ms to 30ms for it to be ready for capture. NI-5660 does not give any output notifying that it is ready for capture. So in all the test procedures when capturing the signal, the procedure is to set up RFSG SMJ100A for transmission and then set it to wait. Next RFSA NI5660 is initialized and RFSG is triggered to transmit after some delay time that is dependent on the system specifications and the program written. This delay time should be calculated experimentally.

The signal is captured for time duration specified in “ni5660 configure for IQ” VI. During the time of capture RFSG transmits the command whose length is less than this time duration of capture. A typical signal capture will appear as shown Figure 13.1. Later the signal can be extracted or clipped in software.

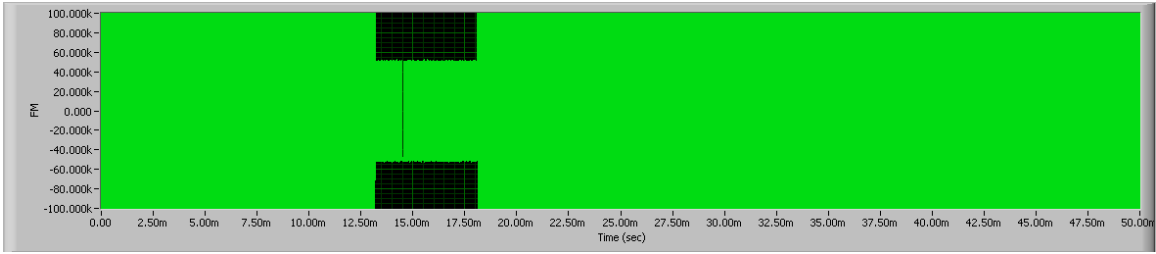


Figure 13.1: Typical signal capture from RFSA

The signal is captured for about 50ms. The command is from 13.2ms to 18.2ms (5ms).

Labview synchronization VIs

Labview offers a set of VIs used for synchronization.

Obtain Notifier

This VI is used to create a notifier or to refer to a created notifier.



Figure 13.2: Obtain Notifier VI

Send Notification

This VI sends a message to all functions waiting on a notifier. All functions currently waiting on the notifier stop waiting and continue to execute.

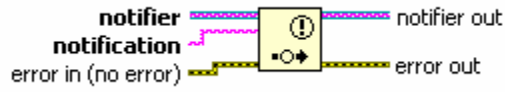


Figure 13.3: Send Notification VI

Wait on Notification

This VI waits until a notifier receives a message. When the notifier receives a message, this function continues to execute.

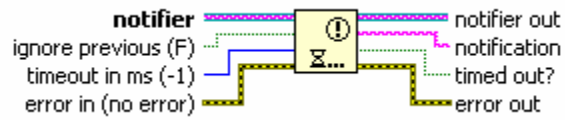


Figure 13.4: Wait on Notification VI

14.0 FORMAT OF A SIGNAL

As seen in Section 13.1, it is not possible for the RFSA to exactly capture the signal transmitted by the wireless device in a single frame without any other information. Hence after demodulation, it is required to develop a program to search through the captured waveform and find the signal. To successfully develop such a program, it is necessary to understand the composition of the signal.

The signal transmitted between wireless devices can be divide into parts. A typical signal or command consists of:

1. Preamble
2. Synchronization Pulses
3. Start of Data Pulses
4. Data Pulses
5. Error Check or Corrections Pulses
6. End of Data Pulses

The composition of the actual signal may vary depending upon the wireless device, modulation scheme used, standard it is built to conform to and security level required. However, every part of the signal can be classified into one of the six parts of a signal mentioned above.

14.1 PREAMBLE

A preamble is a signal used in wireless communications to synchronize the transmission timing between two or more systems. Proper timing ensures that all systems are interpreting the start of the information transfer correctly.

A preamble defines a specific series of transmission pulses that is understood by communicating systems to indicate the start of data transmission. This ensures that systems receiving the information correctly interpret when the data transmission starts. The number of pulses in the preamble and their pulse widths vary for different standards.

14.2 SYNCHRONIZATION PULSES (SYNC PULSE)

Synchronization pulses are used to synchronize the data transmission between the transmitter and the receiver. Synchronization pulses are generally transmitted before the start of data transfer to establish a fixed reference point, and sometimes they are transmitted after data transfer is completed.

Synchronization pulses are typically one or two pulses that are usually defined so that the receiver can distinguish them from all other pulses in the signal. This ensures that the receiver system clearly understands the start and end of data transfers. The physical properties of synchronization pulses differ for different standards and communication systems.

14.3 START OF DATA PULSES

Start of data pulses can be defined as redundant data bits transmitted to aid in signaling of start of data transfer. Sometimes the sync pulse itself can be used as start of data pulse. In such a condition any data after sync pulse is considered as data pulses. However, if the data bits

immediately follow the sync pulse, depending on the type of coding used, sometimes the first data bit might add to the length of sync pulse and alter sync pulse width. Hence, it is advisable to transmit redundant start of data pulses between sync pulse and data pulses. This ensures that the sync pulse width is never altered.

14.4 DATA PULSES

This is where the actual data is transmitted. It is stream of binary ones and zeros represented as electrical pulses. The representation of binary one and zero varies depending on the coding and number of symbols used.

The syntax and order of transmission of the data bits differ for different communication systems and the standards to which it conforms to. Some wireless devices transfer data from MSB to LSB, others transfer from LSB to MSB and others are undefined. Some standards require a redundant bit (either zero or one) to be transmitted after transmitting a particular number of data bits.

14.5 ERROR CHECK AND CORRECTION PULSES

As wireless communication is prone to noise, error check and correction algorithms are implemented on data being transmitted. An example for an error correction algorithm is Hamming code, and an example for an error detection algorithm is the CRC algorithm.

Error correction data bits are transmitted after all data bits are transmitted. They will be used by the receiver to determine if there is an alteration in data because of noise and, if possible, to correct the data received.

14.6 END OF DATA PULSE

End of data pulses define the end of transmission. Any data transmitted after this pulse (or these pulses), are not accepted by the receiving device. End of data pulses can also be followed by sync pulses depending on the communication system.

14.7 COLLECTION COMMAND FOR ISO 18000-7 STANDARD

Figure 14.1 below illustrates the collection command for the ISO 18000-7 standard. The portion marked in red is the preamble. The preamble is followed by the sync pulse indicated in blue. There is only one high pulse and one low pulse, which together form the sync pulse, defined in this standard. The sync pulse is immediately followed by data bits. There is no start of data bit(s) in this standard. Data bits are represented in yellow. There is a stop bit transmitted between every 8 data bits. The command ends with a long low pulse which indicated the end of transmitted data.

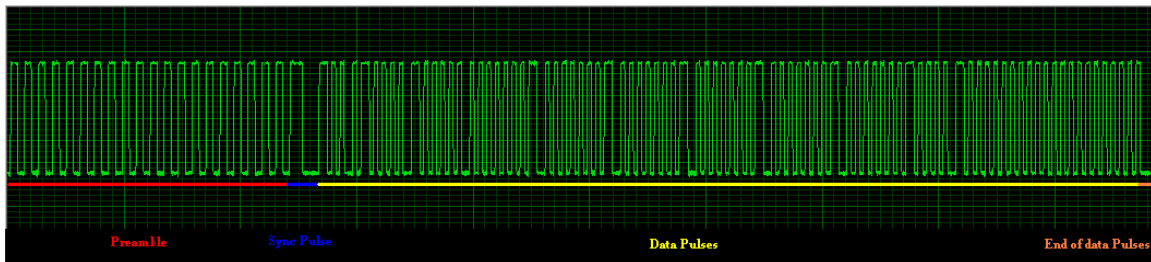


Figure 14.1: Collection command for the ISO 18000-7 standard

15.0 DESCRIPTION OF VIs DEVELOPED

This section explains the VIs developed to test the transmitted signal that it conforms to the ISO 18000 – 7 standard. There are in total 20 VIs testing different properties of the commands transmitted by the wireless devices. A list of the VIs developed is given below.

1. FSK Frequency Deviation Test – Reader to Tag
2. FSK Frequency Deviation Test – Tag to Reader
3. FSK Frequency Deviation Test – Load and Test
4. Carrier Frequency Test – Reader to Tag
5. Carrier Frequency Test – Tag to Reader
6. Carrier Frequency Test – Load and Test
7. Wakeup signal Test
8. Wakeup signal Test – Load and Test
9. Preamble Test – Tag to reader
10. Preamble Test – Reader to Tag
11. Preamble Test – Tag to reader – Load and Test
12. Preamble Test – Reader to Tag – Load and Test
13. Data Transmitted Test – Reader to tag
14. Data Transmitted Test – Tag to Reader
15. Data Transmitted Test – Reader to tag – Load and Test
16. Data Transmitted Test – Tag to Reader – Load and Test
17. Decode Data - Reader to tag - Load and Test
18. Decode Data - Tag to Reader - Load and Test
19. Load and analyze a long capture
20. Receiver Bandwidth Test

Each VI is divided into two parts – Capture window and Test window. In the capture window, the RFSG and RFSA are configured, the command is transmitted and the response is captured. Testing of response signal is done in the Test window. All VIs that load an already captured signal and test it have only one window.

The operation is same for the same test done from Tag to Reader or Reader to Tag. For example the procedure to operate FSK Frequency Deviation Test – Tag to Reader, will be the same as the procedure to operate FSK Frequency Deviation Test – Reader to Tag.

Common Inputs to all VIs

Table 15.1: Common Inputs to all VIs

Input name	Explanation
Downconverter Device Number	Open Measurement and automation Explorer. Look in: My System >> Devices and Interfaces >> NI-DAQmx Devices
Digitizer Resource Name	Open Measurement and automation Explorer. Look in: My System >> Devices and Interfaces >> Traditional NI-DAQ (Legacy) Devices
Carrier Frequency	Center frequency of input signal
Bandwidth	It is the BW of NI-RFSA 5660.
Reference level	Reference power level of input signal
Time Duration	Specifies for how long the signal is to be captured
Resource Name	Open Measurement and automation Explorer. Look in: My System >> Devices and Interfaces >> GPIB0
Waveform Select	Selects waveform to transmit

Table 15.1 (continued)

Amplitude	Amplitude of signal being transmitted
Frequency	Center frequency of signal to be transmitted
Transition time	Specifies the Rise time and Fall time of the signal pulse. This period of the signal will be ignored when testing.
Delay time	Used for synchronization between RFSG SMJ100A & RFSA NI5660. It must be set depending on speed of OS.
Force Exit	Stops executing the VI without transmitting or testing.

15.1 FSK FREQUENCY DEVIATION TEST VI

Test Objective

The objective of this test is that to ensure that the signal frequency deviation is within acceptable operating limits.

Inputs

1. Downconverter Device Number
2. Digitizer Resource Name
3. Carrier Frequency
4. Bandwidth
5. Reference level
6. Time Duration
7. Deviation limits - max freq & min freq
8. Transition Time
9. Resource Name
10. Waveform Select

11. Amplitude
12. Frequency
13. Delay time

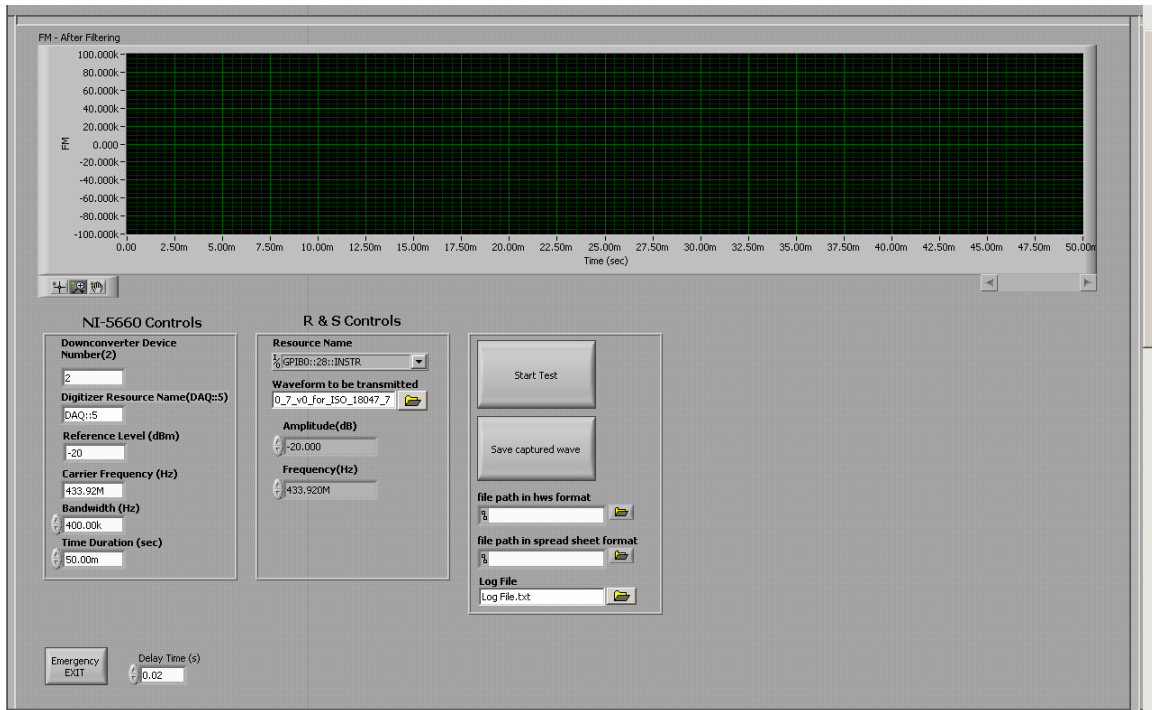


Figure 15.1: Capture Window of FSK Frequency Deviation Test VI

Outputs

1. Minimum Freq High Deviation
2. Maximum Freq High Deviation
3. Minimum Freq Low Deviation
4. Maximum Freq Low Deviation
5. Average Freq Deviation
6. Pass or Fail

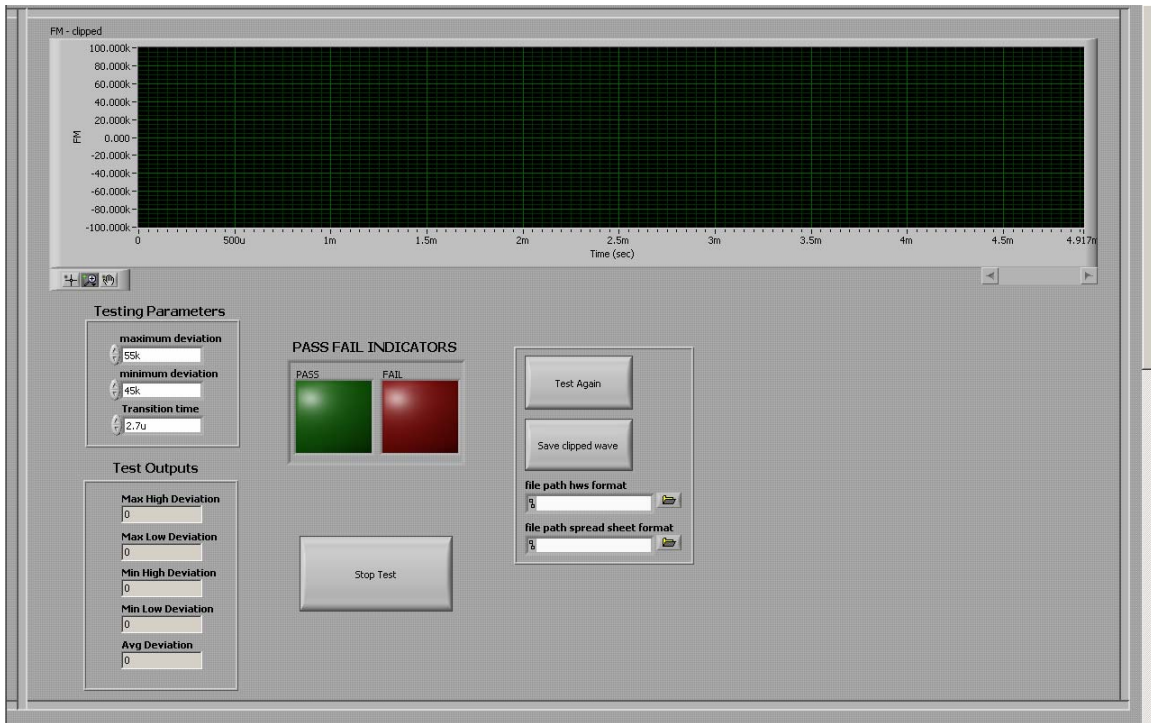


Figure 15.2: Test Window of FSK Frequency Deviation Test VI

Procedure to use the VI

1. User will input all the required parameters for testing mentioned above.
2. User will start executing VI.
3. User clicks on *start test* button.
4. After the signal is captured, a dialogue will appear on screen.
5. User has an option to save the entire captured signal in Labview format and Excel format by pressing *save captured wave* button.
6. The command or response is clipped from the entire capture and is shown on a graph indicator in the test window.
7. Test is performed.

8. User has an option to save the command in Labview format and Excel format by pressing the *save clipped waveform* button.
9. Pass or fail indicators (Green and Red LEDs) indicate if the test result is pass or fail.

Algorithm of test:

1. Signal is acquired from RFSA.
2. **Clip signal box:** For the first test ($i = 0$) the box calculates start of signal and length of signal values and saves them to memory. When the response is tested again (for all other values of i), the box outputs the same values calculated the first time.
3. **Finding Sync Pulse box:** A window of 130us is defined.
4. From the start to end of captured waveform, the window is shifted for alignment and the waveform is searched for the sync pulse.
5. Sync pulse is found if a high pulse of length specified by standard is followed by a low pulse of length specified by the standard occurs within the window.
6. Start of command is set to be 1210 us before start of sync pulse.
7. From end of low sync pulse, the signal is searched for the end of data bits by defining a window of 40us.
8. End bits are said to be found when there is a low pulse of length greater than or equal to 35us at a time multiple of 36us from start of data.
9. End of command is set to end of data end bits found.
10. Taking the start of command and end of command sample values, the signal is clipped.
11. **Resampling the signal to 10ns:** The signal is resampled at 10ns intervals. All logical analysis is done at 10ns intervals as the standard specifies.
12. The resampled signal is filtered by a median filter to eliminate pulse distortion caused by resampling.
13. **Check for signal within maximum deviation box:** First the signal is checked for a glitch. There is a glitch if there is a set of continuous points exceeding the maximum deviation value specified by the user.

14. If there is a glitch and its width is within specified value as set by the user, it is eliminated.
15. Then each sample above zero deviation is compared to higher maximum deviation tolerance value.
16. The signal is said to be within higher deviation limit if all samples are below max deviation limit specified.
17. Then each sample below zero deviation is compared to lower maximum deviation tolerance value which is $-$ (higher maximum deviation tolerance value).
18. Signal is said to be within lower maximum deviation limit if all samples are below max deviation limit specified.
19. **First and last point of signal on transition check box:** This box checks if the signal is clipped such that the first and last point are on transition period and will ignore the transition time as specified by the user from the beginning or end of signal. If the signal is clipped correctly, this box is void.
20. **Box that removes transition period of signal:** The signal has finite samples during rise time and fall time of each pulse. These points are ignored when checking for lower frequency tolerance. The user specifies the maximum transition time possible in the signal.
21. Number of transition points between high min deviation and low minimum deviation is calculated. This number is compared to the maximum number of transition points possible. If the number of transition points is less than or equal to maximum transition points possible, these transition points are ignored.
22. **Signal above minimum deviation box:** Next each sample above zero deviation is compared to higher minimum deviation tolerance value.
23. The signal is said to be within lower deviation limit if all samples above zero are above min deviation limit specified.
24. Then, each sample below zero deviation is compared to lower minimum deviation tolerance value which is $-$ (higher minimum deviation tolerance value).
25. The signal is said to be within lower deviation limit if all samples below zero are below the minimum deviation limit specified.

26. **Average deviation calculation box & maximum high deviation value, maximum low deviation value, minimum high deviation value, minimum low deviation value calculation box:** Average deviation value, maximum high deviation value, maximum low deviation value, minimum high deviation value, minimum low deviation value are calculated.
27. Pass or fail indicators indicate if the signal passed the test or not. The signal passes the test if it is between upper and lower deviation tolerance values.

15.2 FSK FREQUENCY DEVIATION TEST VI – LOAD AND TEST

Test Objective

The objective of this test is to ensure that the signal frequency deviation is within acceptable operating limits.

This test is done off-line without using a signal generator or the signal analyzer. The command captured by RFSA that is stored in memory can then be loaded into Labview and tested.

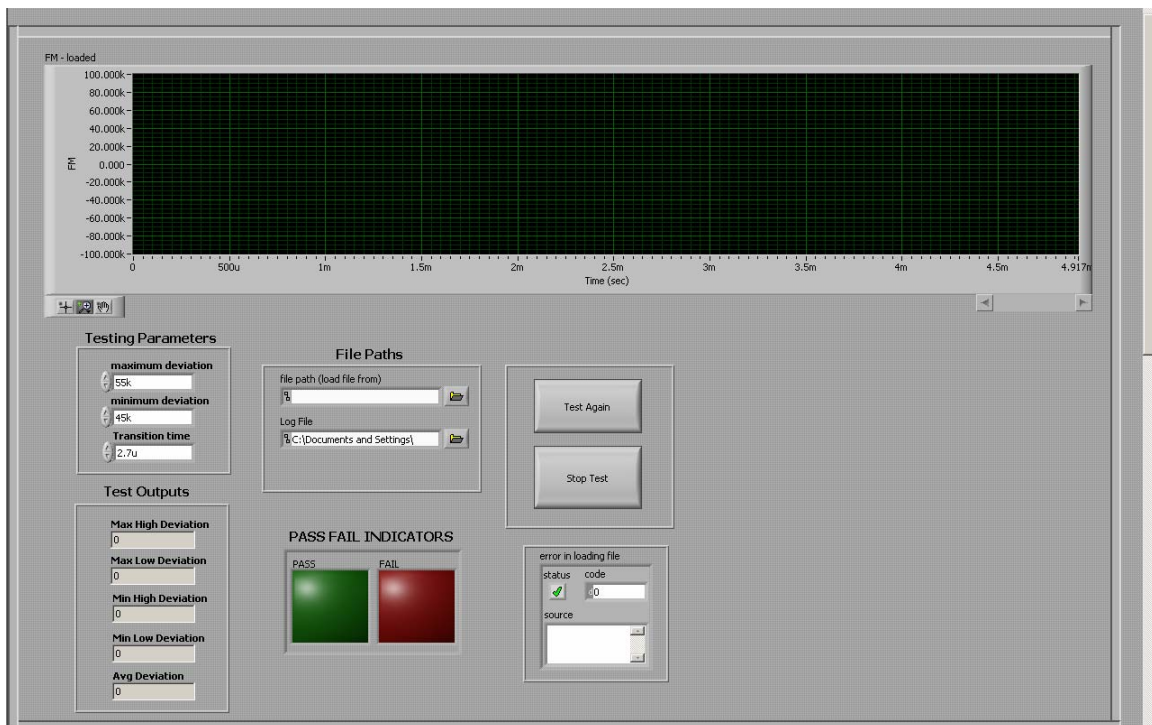


Figure 15.3: Front panel of FSK Frequency Deviation Test VI – Load and Test

Inputs

1. Deviation limits - maximum frequency & minimum frequency
2. Transition Time
3. File Path – specifies where the signal for testing should be loaded from

Outputs

1. Minimum Frequency High Deviation
2. Maximum Frequency High Deviation
3. Minimum Frequency Low Deviation
4. Maximum Frequency Low Deviation
5. Average Frequency Deviation
6. Pass or Fail

Procedure to use the VI

1. User will input all the required parameters for testing mentioned above.
2. User will start executing VI.
3. User selects the signal to be loaded and tested.
4. Test is performed.
5. Pass or fail indicators indicate if the test result is pass or fail.

Algorithm of test:

1. **Resampling the signal to 10ns:** The signal loaded from memory is resampled at 10ns intervals. All logical analysis is done at 10ns intervals as the standard specifies.
2. The resampled signal is filtered by a median filter to eliminate pulse distortion caused by resampling.

3. **Check for signal within maximum deviation box:** First the signal is checked for a glitch. There is a glitch if there is a set of continuous points exceeding the maximum deviation value specified by the user.
4. If there is a glitch and its width is within specified value as set by the user, it is eliminated.
5. Then each sample above zero deviation is compared to higher maximum deviation tolerance value.
6. The signal is said to be within higher deviation limit if all samples are below max deviation limit specified.
7. Then each sample below zero deviation is compared to lower maximum deviation tolerance value which is $-$ (higher maximum deviation tolerance value).
8. Signal is said to be within lower maximum deviation limit if all samples are below max deviation limit specified.
9. **First and last point of signal on transition check box:** This box checks if the signal is clipped such that the first and last point are on transition period and will ignore the transition time as specified by the user from the beginning or end of signal. If the signal is clipped correctly, this box is void.
10. **Box that removes transition period of signal:** The signal has finite samples during rise time and fall time of each pulse. These points are ignored when checking for lower frequency tolerance. The user specifies the maximum transition time possible in the signal.
11. Number of transition points between high minimum deviation and low minimum deviation is calculated. This number is compared to the maximum number of transition points possible. If the number of transition points is less than or equal to maximum transition points possible, these transition points are ignored.
12. **Signal above minimum deviation box:** Next each sample above zero deviation is compared to higher minimum deviation tolerance value.
13. The signal is said to be within lower deviation limit if all samples above zero are above minimum deviation limit specified.
14. Then, each sample below zero deviation is compared to lower minimum deviation tolerance value which is $-$ (higher minimum deviation tolerance value).

15. The signal is said to be within lower deviation limit if all samples below zero are below the minimum deviation limit specified.
16. **Average deviation calculation box & maximum high deviation value, maximum low deviation value, minimum high deviation value, minimum low deviation value calculation box:** Average deviation value, maximum high deviation value, maximum low deviation value, minimum high deviation value, minimum low deviation value are calculated.
17. Pass or fail indicators indicate if the signal passed the test or not. The signal passes the test if it is between upper and lower deviation tolerance values.

15.3 CARRIER FREQUENCY TEST VI

Test Objective

The objective of this test is to ensure that the interrogator communication frequency is in compliance with the ISO/IEC 18000-7 standard.

Inputs

1. Downconverter Device Number
2. Digitizer Resource Name
3. Carrier Frequency
4. Bandwidth
5. Reference level
6. Time Duration
7. Deviation limits of carrier in ppm
8. Transition Time
9. Resource Name
10. Waveform Select
11. Amplitude

12. Frequency
13. Delay time

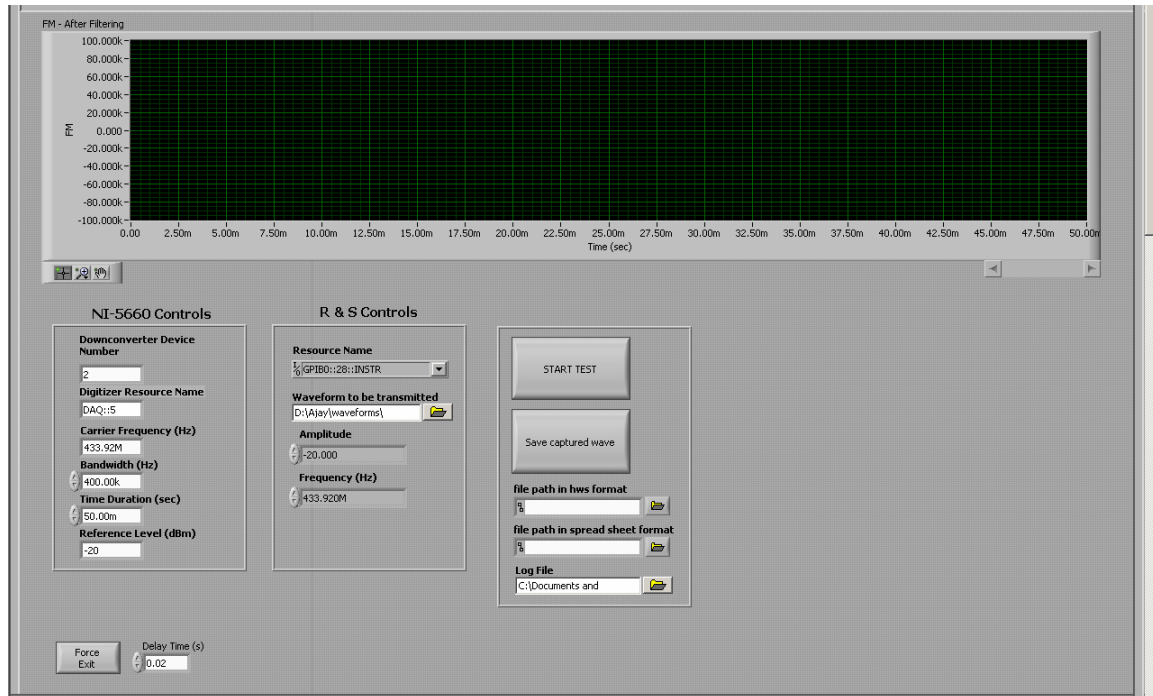


Figure 15.4: Capture window of Carrier Frequency Test VI

Outputs

1. Tolerance in Hz
2. Pass or Fail

Procedure to use the VI

1. User will input all the required parameters for testing mentioned above.
2. User will start executing VI.
3. User clicks on *start test* button.

4. After the signal is captured, a dialogue will appear on screen.
5. User has an option to save the entire captured signal in Labview format and Excel format by pressing *save captured wave* button.
6. The command or response is clipped from the entire capture and is shown on a graph indicator in the test window.
7. Test is performed.
8. User has an option to save the command in Labview format and Excel format by pressing the *save clipped waveform* button.
9. Pass or fail indicators indicate if the test result is pass or fail.

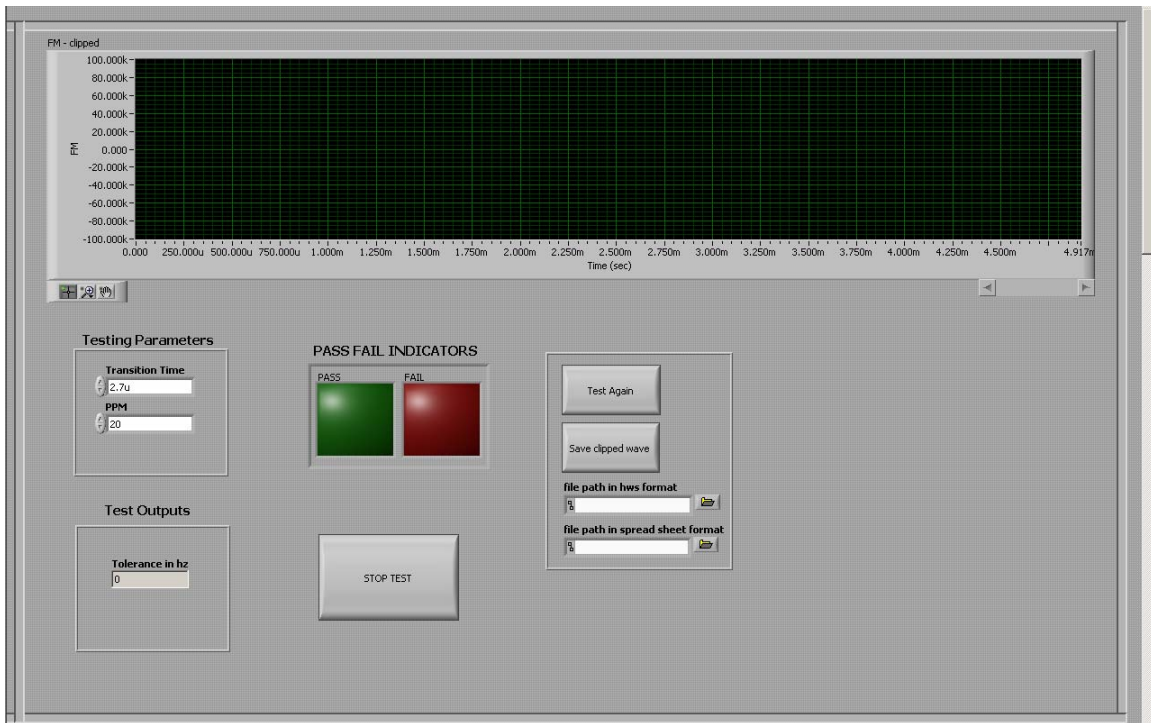


Figure 15.5: Test window of Carrier Frequency Test VI

Algorithm of test:

1. Signal is acquired from RFSA.
2. **Clip signal box:** For the first test ($i = 0$) the box calculates start of signal and length of signal values and saves them to memory. When the response is tested again (for all other values of i), the box outputs the same values calculated the first time.
3. **Finding Sync Pulse box:** A window of 130us is defined.
4. From the start to end of captured waveform, the window is shifted for alignment and the waveform is searched for the sync pulse.
5. Sync pulse is found if a high pulse of length specified by standard is followed by a low pulse of length specified by the standard occurs within the window.
6. Start of command is set to be 1210us before start of sync pulse.
7. From the end of low sync pulse, the signal is searched for the end of data bits by defining a window of 40us.
8. End bits are said to be found when there is a low pulse of length greater than or equal to 35us at a time multiple of 36us from start of data.
9. End of command is set to end of data end bits found.
10. Taking the start of command and end of command sample values, the signal is clipped.
11. **Resampling the signal to 10ns:** The signal is resampled at 10ns intervals. All logical analysis is done at 10ns intervals as the standard specifies.
12. The resampled signal is filtered by a median filter to eliminate pulse distortion caused by resampling.
13. **First and last point of signal on transition check box:** This box checks if the signal is clipped such that the first and last point are on the transition period and will ignore the transition time as specified by the user from the beginning or end of signal. If the signal is clipped correctly, this box is void.
14. **Box that removes transition period of signal and checks for glitch:** The signal has finite samples during rise time and fall time of each pulse. These points are ignored when checking for carrier frequency deviation tolerance. The user specifies “Transition Time” which is made zero when testing the signal for carrier frequency deviation tolerance.

15. The number of transition points between high minimum deviation and low minimum deviation is calculated. These points are compared to the maximum transition points possible. If the number of transition points is less than or equal to maximum transition points possible, these transition points are ignored.
16. Next, the signal is checked for a glitch. There is a glitch, if there is a set of continuous points exceeding the maximum deviation value as specified by the user.
17. If there exists a glitch, and it is within the specified value as set by the user, it is eliminated.
18. An assumption is made that there is no error in the deviation of signal and all error is in the deviation of the carrier frequency.
19. **Carrier Frequency Deviation at each point box:** The deviation of carrier is calculated by subtracting 50000 from all high samples and adding 50000 to all low samples.
20. A modulus operation is performed on the resulting array.
21. **Check if deviation is within limits box:** The signal is said to be within tolerance limit if all samples are below the maximum deviation limit specified.
22. Pass or fail indicators indicate if the signal passed the test or not. The signal is passed if all samples are below the maximum deviation allowed
23. ppm is converted into Hz using the formula : $\text{Hz} = \text{ppm} * \text{center freq} / 1\text{M}$

15.4 CARRIER FREQUENCY TEST VI – LOAD AND TEST

Test Objective

The objective of this test is that to ensure that the interrogator communication frequency is according to the ISO/IEC 18000-7 standard.

This test is done off-line without using the signal generator or the signal analyzer. The command captured by RFSA that is stored in memory can be loaded into Labview and tested.

Inputs

1. Deviation limits of the carrier in ppm
2. Transition Time
3. File Path – specifies where the signal for testing should be loaded from

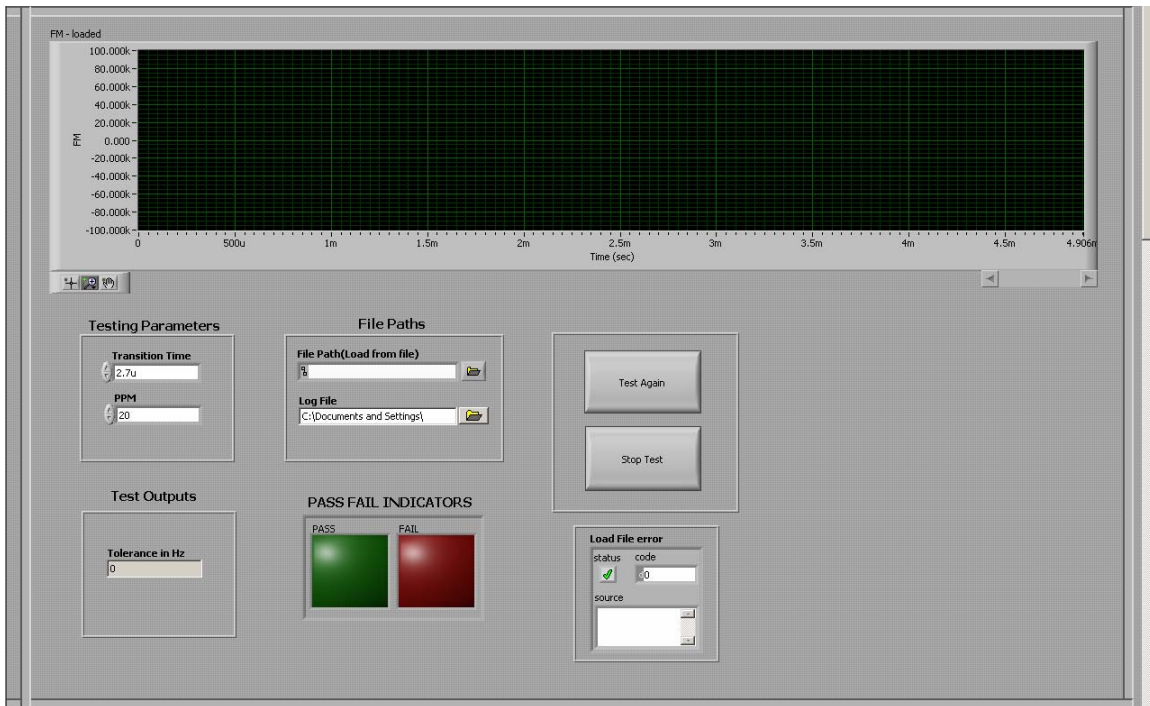


Figure 15.6: Front Panel of Carrier Frequency Test VI – Load and Test

Outputs

1. Tolerance in Hz
2. Pass or Fail

Procedure to use the VI

1. User will input all the required parameters for testing as mentioned above.
2. User will start executing the VI.
3. User selects the signal to be loaded and tested.
4. Test is performed.
5. Pass or fail indicators indicate if the test result is pass or fail.

Algorithm of test:

1. **Resampling the signal to 10ns:** The signal loaded from memory is resampled at 10ns intervals. All logical analysis is done at 10ns intervals as the standard specifies.
2. The resampled signal is filtered by a median filter to eliminate pulse distortion caused by resampling.
3. **First and last point of signal on transition check box:** This box checks if the signal is clipped such that the first and last points are on transition period and will ignore the transition time as specified by the user from the beginning or end of signal. If the signal is clipped correctly, this box is void.
4. **Box that removes transition period of signal and checks for glitch:** The signal has finite samples during rise time and fall time of each pulse. These points are ignored when checking for carrier frequency deviation tolerance. The user specifies “Transition Time” which is made zero when testing the signal for carrier frequency deviation tolerance.
5. The number of transition points between high minimum deviation and low minimum deviation is calculated. These points are compared to the maximum transition points possible. If the number of transition points is less than or equal to maximum transition points possible, these transition points are ignored.
6. Then the signal is checked for a glitch. There is a glitch, if there is a set of continuous points exceeding the maximum deviation value as specified by the user.

7. If there exists a glitch, and it is within specified value as set by the user, it is eliminated.
8. An assumption is made that there is no error in the deviation of signal and all error is in the deviation of carrier frequency.
9. **Carrier Frequency Deviation at each point box:** The deviation of carrier is calculated by subtracting 50000 from all high samples and adding 50000 to all low samples.
10. A modulus operation is performed on the resulting array.
11. **Check if deviation is within limits box:** The signal is said to be within tolerance limit if all samples are below the maximum deviation limit specified.
12. Pass or fail indicators indicate if the signal passed the test or not. The signal is passed if all samples are below the maximum deviation allowed
13. ppm is converted into Hz using the formula : $\text{Hz} = \text{ppm} * \text{center freq} / 1\text{M}$

15.5 WAKEUP SIGNAL TEST VI

Test Objective

The objective of this test is that to verify that wakeup signal duration is within nominal tolerance values and pulse widths are within nominal values.

Inputs

1. Downconverter Device Number
2. Digitizer Resource Name
3. Carrier Frequency
4. Bandwidth
5. Reference level
6. Time Duration
7. Deviation limits of pulse duration – Maximum pulse duration and minimum pulse duration

8. Resource Name
9. Waveform Select
10. Amplitude
11. Frequency
12. Delay time

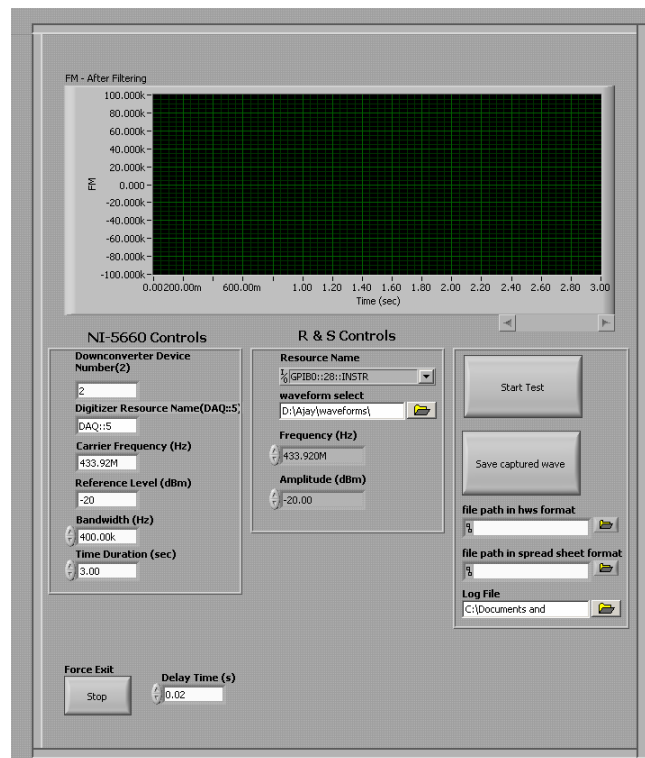


Figure 15.7: Capture Window of Wakeup signal Test VI

Outputs

1. Minimum high pulse width
2. Maximum high pulse width
3. Average high pulse width
4. Minimum low pulse width

5. Maximum low pulse width
6. Average low pulse width
7. Wakeup length
8. Points on glitch
9. Pass or Fail

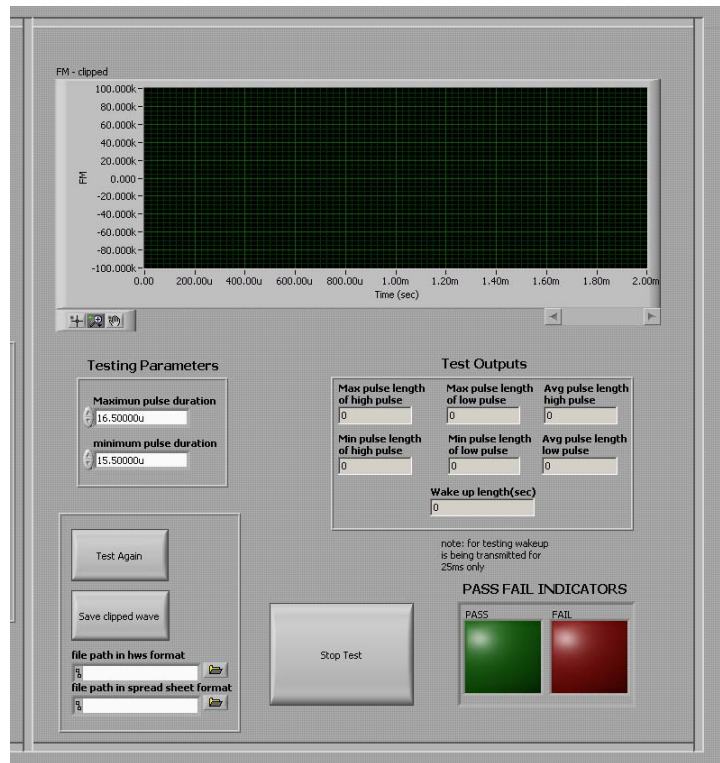


Figure 15.8: Test Window of Wakeup signal Test VI

Procedure to use the VI

1. User will input all the required parameters for testing mentioned above.
2. User will start executing VI.
3. User clicks on *start test* button.
4. After the signal is captured, a dialogue will appear on screen.

5. User has an option to save the entire captured signal in Labview format and Excel format by pressing *save captured wave* button.
6. The command or response is clipped from the entire capture and is shown on a graph indicator in the test window.
7. Test is performed.
8. User has an option to save the command in Labview format and Excel format by pressing the *save clipped waveform* button.
9. Pass or fail indicators indicate if the test result is pass or fail.

Algorithm of test:

1. Signal is acquired from RFSA.
2. **Clip signal box:** For the first test ($i = 0$) the box calculates start of signal and length of signal values and saves them to memory. When the response is tested again (for all other values of i), the box outputs the same values calculated the first time.
3. **Finding Wakeup Pulse box:** A window of 40us is defined.
4. From the start to end of captured signal, the window is shifted for alignment and signal is searched for the wakeup pulse.
5. Wakeup pulse is found if a high pulse of length specified by standard followed by a low pulse of length specified by the standard occurs in the window.
6. Start of command is set when the first wakeup pulse is encountered.
7. End of wakeup is set when a pulse not according the standard is encountered.
8. Taking the start of command and end of command values, the signal is clipped.
9. Because there is not enough memory to test the entire signal at a time it is broken into parts on 2ms each.
10. **Resampling the signal to 10ns:** The signal is resampled at 10ns intervals. All logical analysis is done at 10ns intervals as the standard specifies.
11. The resampled signal is filtered by a median filter to eliminate pulse distortion caused by resampling.

12. **Check for glitch and remove it box:** First the signal is checked for a glitch. There is a glitch, if there is a set of continuous points exceeding the maximum deviation value as specified by the user.
13. If there is a glitch, and its width is within specified value as set by the user, it is eliminated.
14. There are approximately 62 high pulses in a 2ms wakeup. Out of these pulses, the 3rd to 60th pulses are tested by Wakeup High Pulses Array – Sub VI.
15. Maximum, minimum and average pulse widths are calculated for each 2ms piece of wakeup signal.
16. The maximum value and minimum value calculated are checked to be within limits.
17. There are approximately 62 low pulses in a 2ms wakeup. Out of these pulses, the 3rd to 60th pulses are tested by Wakeup Low Pulses Array – Sub VI.
18. Maximum, minimum and average pulse widths are calculated for each 2ms piece of wakeup signal.
19. The maximum value and minimum value calculated are checked to be within limits.
20. When the test encounters a pulse that does not pass, the wakeup will fail the test and the test procedure will not test the rest of the wakeup signal.
21. Pass or fail indicators indicate if the signal passed the test or not.

15.6 WAKEUP SIGNAL TEST VI – LOAD AND TEST

Test Objective

The objective of this test is that to verify that the wakeup signal duration is within the nominal tolerance values and pulse widths are the within nominal values.

This test is done off-line without using the signal generator or the signal analyzer. The command captured by RFSA that is stored in memory can be loaded into Labview and tested.

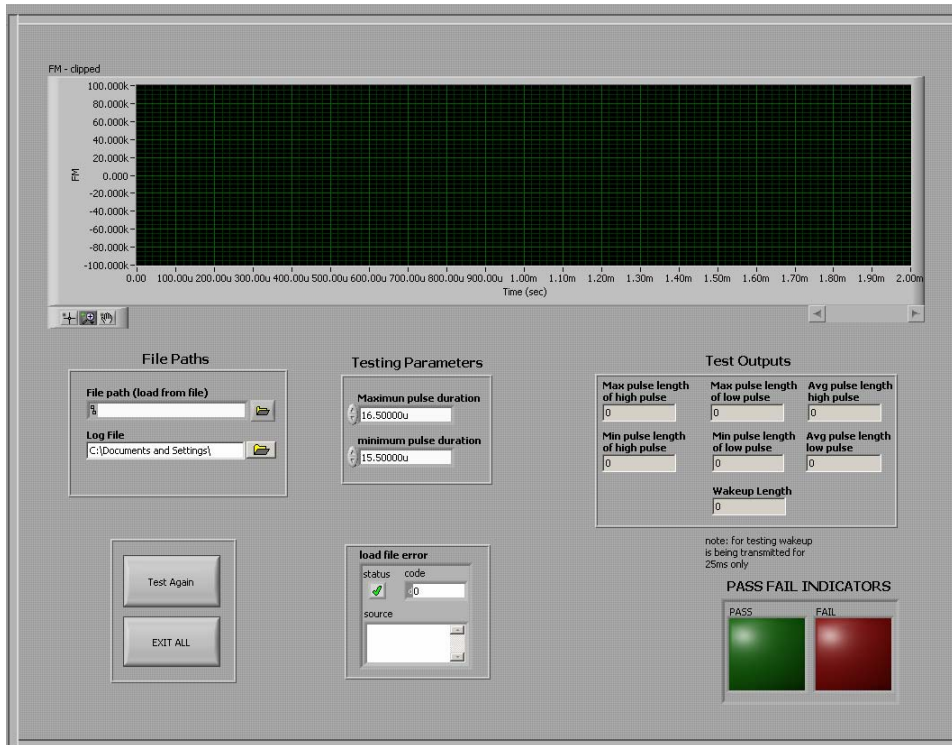


Figure 15.9: Front Panel of Wakeup signal Test VI – Load and Test

Inputs

1. Deviation limits of pulse duration – Maximum pulse duration and minimum pulse duration
2. File Path – specifies from where the signal for testing should be loaded.

Outputs

1. Minimum high pulse width
2. Maximum high pulse width
3. Average high pulse width
4. Minimum low pulse width
5. Maximum low pulse width

6. Average low pulse width
7. Wakeup length
8. Points on glitch
9. Pass or Fail

Procedure to use the VI

1. User will input all the required parameters for the test mentioned above.
2. User will start executing VI.
3. User selects the signal to be loaded and tested.
4. Test is performed.
5. Pass or fail indicators indicate if the test result is pass or fail.

Algorithm of test

1. Because there is not enough memory to test the entire signal at a time, it is broken into parts on 2ms each.
2. **Resampling the signal to 10ns:** The signal loaded from memory is resampled at 10ns intervals. All logical analysis is done at 10ns intervals as the standard specifies.
3. The resampled signal is filtered by a median filter to eliminate pulse distortion caused by resampling.
4. **Check for glitch and remove it box:** First the signal is checked for a glitch. There is a glitch, if there is a set of continuous points exceeding the maximum deviation value as specified by the user.
5. If there is a glitch, and if the width of the glitch is within specified value as set by the user, it is eliminated.
6. There are approximately 62 high pulses in a 2ms wakeup. Out of these pulses, the 3rd to 60th pulses are tested by Wakeup High Pulses Array – Sub VI.
7. Maximum, minimum and average pulse widths are calculated for each 2ms piece of wakeup signal.
8. The maximum value and minimum value calculated are checked to be within limits.

9. There are approximately 62 low pulses in a 2ms wakeup. Out of these pulses, the 3rd to 60th pulses are tested by Wakeup Low Pulses Array – Sub VI.
10. Maximum, minimum and average pulse widths are calculated for each 2ms piece of wakeup signal.
11. The maximum value and minimum value calculated are checked to be within limits.
12. When the test encounters a pulse that does not pass, the wakeup will fail the test and the test procedure will not test the rest of the wakeup signal.
13. Pass or fail indicators indicate if the signal passed the test or not.

15.7 PREAMBLE TEST VI

Test Objective

The objective of this test is to verify that the Preamble signal duration is within nominal tolerance values, and the preamble pulse widths are within nominal values.

Inputs

1. Downconverter Device Number
2. Digitizer Resource Name
3. Carrier Frequency
4. Bandwidth
5. Reference level
6. Time Duration
7. Deviation limits of high pulse duration – Maximum pulse duration and minimum pulse duration
8. Deviation limits of low pulse duration – Maximum pulse duration and minimum pulse duration

9. Deviation limits of preamble pulse duration – Maximum pulse duration and minimum pulse duration
10. Resource Name
11. Waveform Select
12. Amplitude
13. Frequency
14. Delay time

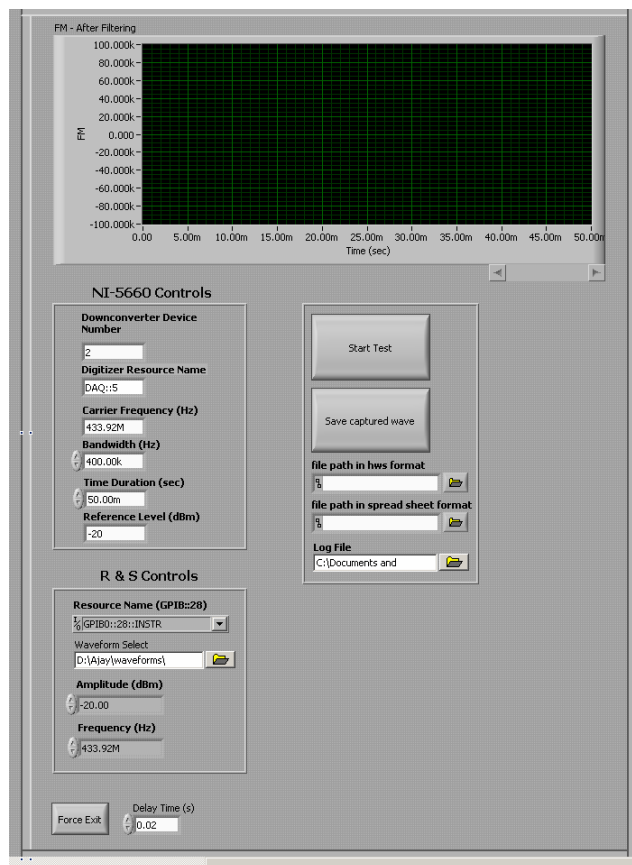


Figure 15.10: Capture Window of Preamble Test VI

Outputs

1. Minimum high pulse width
2. Maximum high pulse width
3. Average high pulse width
4. Minimum low pulse width
5. Maximum low pulse width
6. Average low pulse width
7. high sync pulse width
8. low sync pulse width
9. Preamble Length
10. Pass or Fail

Procedure to use the VI

1. User will input all the required parameters for testing mentioned above.
2. User will start executing VI.
3. User clicks on *start test* button.
4. After the signal is captured, a dialogue will appear on screen.
5. User has an option to save the entire captured signal in Labview format and Excel format by pressing *save captured wave* button.
6. The command or response is clipped from the entire capture and is shown on a graph indicator in the test window.
7. Test is performed.
8. User has an option to save the command in Labview format and Excel format by pressing the *save clipped waveform* button.
9. Pass or fail indicators indicate if the test result is pass or fail.

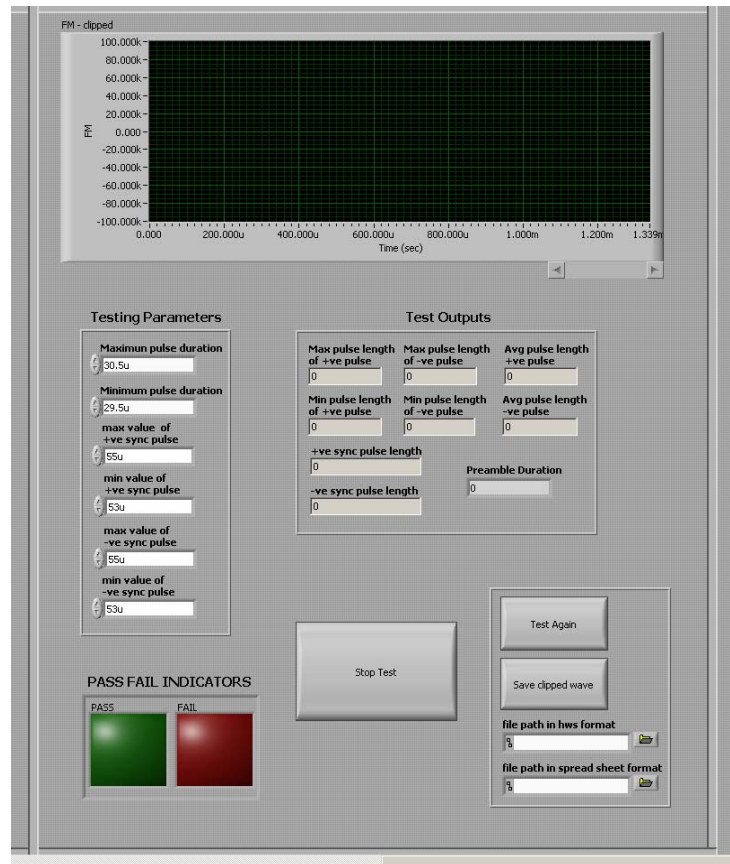


Figure 15.11: Test Window of Preamble Test VI

Algorithm of test

1. Signal is acquired from the RFSA.
2. **Clip signal box:** For the first test ($i = 0$) the box calculates start of signal and length of signal values and saves them to memory. When the response is tested again (for all other values of i), the box outputs the same values calculated the first time.
3. **Finding Sync Pulse box:** A window of 130us is defined.
4. From the start to end of captured waveform, the window is shifted for alignment and the waveform is searched for the sync pulse.
5. Sync pulse is found if a high pulse of length specified by standard is followed by a low pulse of length specified by the standard occurs within the window.
6. Start of command is set to be 1210 us before start of sync pulse.

7. End of command is set to the sample after last sample of the low sync pulse.
8. Taking the start of command and end of command values, the signal is clipped.
9. **Resampling the signal to 10ns:** The signal is resampled at 10ns intervals. All logical analysis is done at 10ns intervals as the standard specifies.
10. The resampled signal is filtered by a median filter to eliminate pulse distortion caused by resampling.
11. **Check for glitch and remove it box:** Then the signal is checked for a glitch. There is a glitch, if there is a set of continuous points exceeding the maximum deviation value as specified by the user.
12. If there is a glitch, and its width is within specified value set by the user, it is eliminated.
13. 1st to 20th pulses are tested by Preamble High Pulses Array – Sub VI.
14. Maximum, minimum and average pulse widths are calculated.
15. The maximum value and minimum value calculated are checked to be within limits.
16. 1st to 20th pulses are tested by Preamble Low Pulses Array – Sub VI.
17. Maximum, minimum and average pulse widths are calculated.
18. The maximum value and minimum value calculated are checked to be within limits.
19. **Clip 160us signal from end:** This box clips 160us of the signal from end of the low sync pulse.
20. **Calculate high and low sync pulse widths:** This box Calculates the pulse widths of the high sync pulse and the low sync pulse.
21. The length of high sync pulse is measured and tested.
22. The length of low sync pulse is measured and tested if it is greater than minimum value and less than the maximum value + 18.5us. An extra 18.5us is added to the maximum pulse width specified by the standard because if the first bit is '0', the low pulse of bit '0' will add to low sync pulse and increase pulse width of low sync pulse.
23. Pass or fail indicators indicate if the signal passed the test or not.

15.8 PREAMBLE TEST VI – LOAD AND TEST

Test Objective

The objective of this test is to verify that the Preamble signal duration is within nominal tolerance values and the pulse widths are within nominal values.

This test is done off-line without using signal generator or the signal analyzer. The command captured by RFSA that is stored in memory can be loaded into Labview and tested.

Inputs

1. Deviation limits of high pulse duration – Maximum pulse duration and minimum pulse duration
2. Deviation limits of low pulse duration – Maximum pulse duration and minimum pulse duration
3. Deviation limits of preamble pulse duration – Maximum pulse duration and minimum pulse duration
4. File Path – specifies where the signal for testing should be loaded from.

Outputs

1. Minimum high pulse width
2. Maximum high pulse width
3. Average high pulse width
4. Minimum low pulse width
5. Maximum low pulse width
6. Average low pulse width
7. High sync pulse width
8. Low sync pulse width

9. Preamble Length

10. Pass or Fail

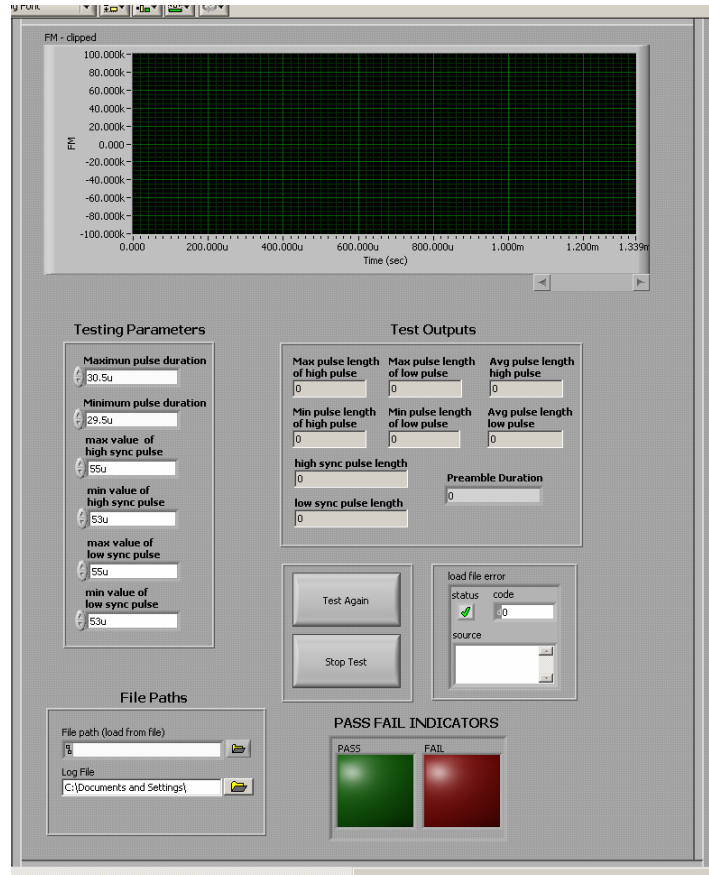


Figure 15.12: Front panel of Preamble Test VI – Load and Test

Procedure to use the VI

1. User will input all the required parameters for testing mentioned above.
2. User will start executing VI.
3. User selects the signal to be loaded and tested.
4. Test is performed.
5. Pass or fail indicators indicate if the test result is pass or fail.

Algorithm of test

1. **Resampling the signal to 10ns:** The signal loaded from memory is resampled at 10ns intervals. All logical analysis is done at 10ns intervals as the standard specifies.
2. The resampled signal is filtered by a median filter to eliminate pulse distortion caused by resampling.
3. **Check for glitch and remove it box:** Then the signal is checked for a glitch. There is a glitch if there is a set of continuous points exceeding the maximum deviation value as specified by the user.
4. If there is a glitch, and its width is within specified value set by the user, it is eliminated.
5. 1st to 20th pulses are tested by Preamble High Pulses Array – Sub VI.
6. Maximum, minimum and average pulse widths are calculated.
7. The maximum value and minimum value calculated are checked to be within limits.
8. 1st to 20th pulses are tested by Preamble Low Pulses Array – Sub VI.
9. Maximum, minimum and average pulse widths are calculated.
10. The maximum value and minimum value calculated are checked to be within limits.
11. **Clip 160us signal from end:** This box clips 160us of the signal from end of the low sync pulse.
12. **Calculate high and low sync pulse widths:** This box Calculates the pulse widths of the high sync pulse and the low sync pulse.
13. The length of high sync pulse is measured and tested.
14. The length of low sync pulse is measured and tested if it is greater than minimum value and less than the maximum value + 18.5us. An extra 18.5us is added to the maximum pulse width specified by the standard because if the first bit is '0', the low pulse of bit '0' will add to low sync pulse and increase pulse width of low sync pulse.
15. Pass or fail indicators indicate if the signal passed the test or not.

15.9 DATA TRANSMITTED TEST VI

Test Objective

The objective of this test is to verify that the interrogator transmission employs Manchester encoding and that the data timing is within limits.

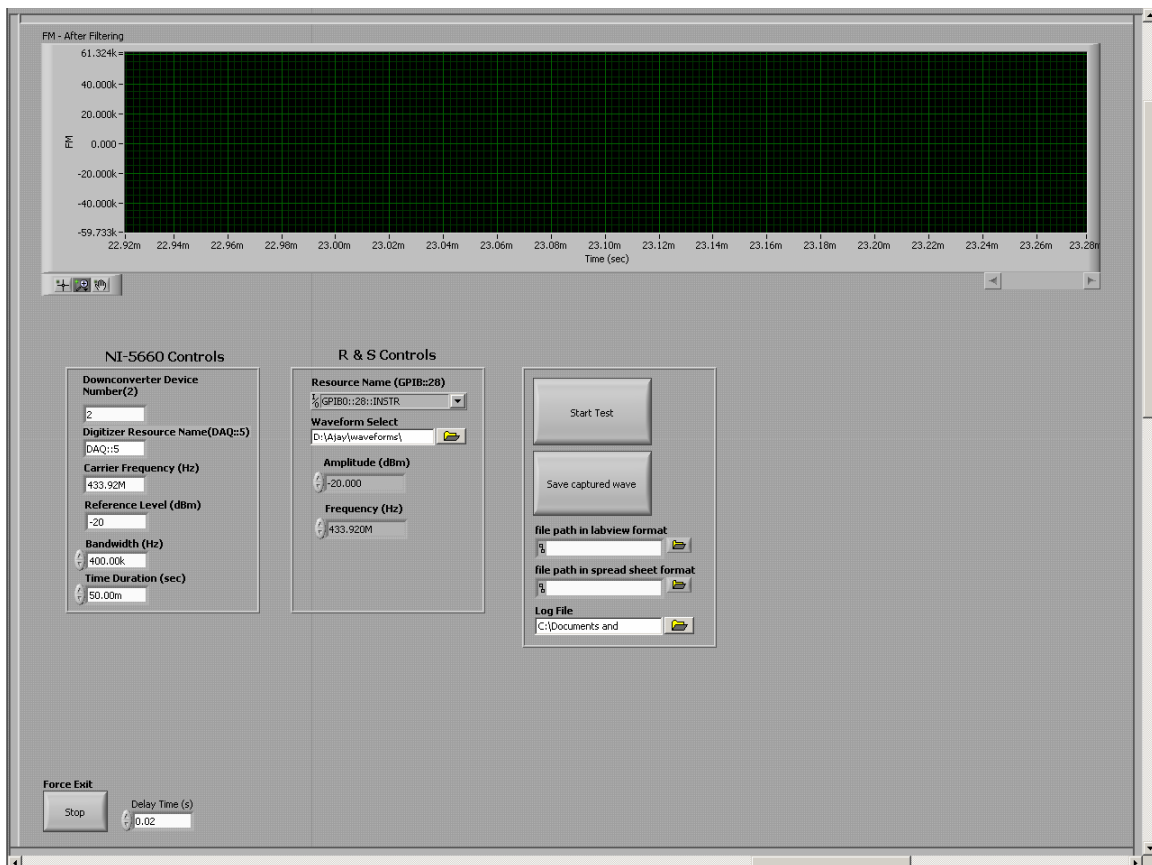


Figure 15.13: Capture Window of Data Transmitted Test VI

Inputs

1. Downconverter Device Number
2. Digitizer Resource Name
3. Carrier Frequency
4. Bandwidth
5. Reference level
6. Time Duration
7. Deviation limits of data pulse duration – Maximum pulse duration and minimum pulse duration
8. Resource Name
9. Waveform Select
10. Amplitude
11. Frequency
12. Delay time

Outputs

1. Average pulse width
2. Decoded Data in Hex
3. Index of last byte of data
4. Pass or Fail

Procedure to use the VI

1. User will input all the required parameters for testing mentioned above.
2. User will start executing VI.
3. User clicks on *start test* button.
4. After the signal is captured, a dialogue will appear on screen.
5. User has an option to save the entire captured signal in Labview format and Excel format by pressing *save captured wave* button.

6. The command or response is clipped from the entire capture and is shown on a graph indicator in the test window.
7. Test is performed.
8. User has an option to save the command in Labview format and Excel format by pressing the *save clipped waveform* button.
9. Data are displayed in Hex format.

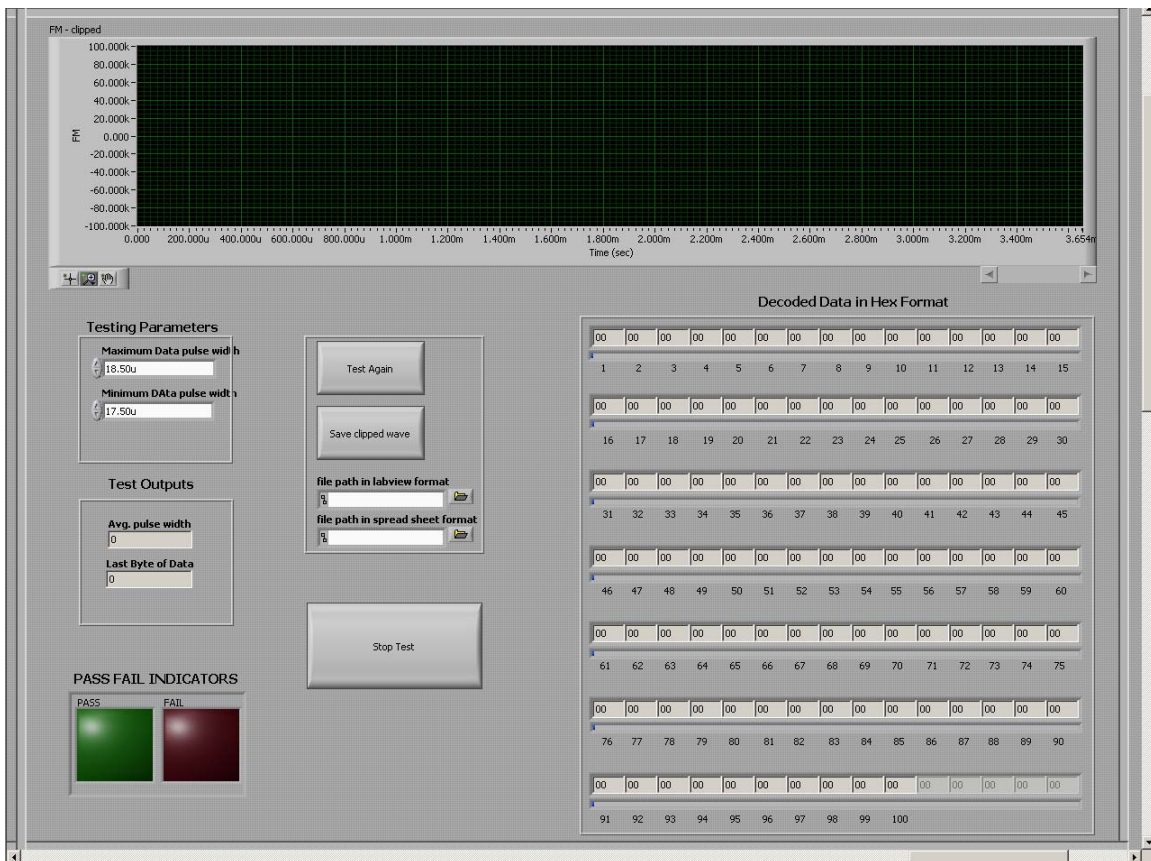


Figure 15.14: Test Window of Data Transmitted Test VI

Algorithm of test

1. Signal is acquired from RFSA.
2. **Clip signal box:** For the first test ($i = 0$) the box calculates start of signal and length of signal values and saves them to memory. When the response is tested again (for all other values of i), the box outputs the same values calculated the first time.
3. **Finding Sync Pulse box:** A window of 130us is defined.
4. From the start to end of the captured waveform, the window is shifted for alignment and the waveform is searched for the sync pulse.
5. Sync pulse is found if a high pulse of length specified by standard is followed by a low pulse of length specified by the standard occurs within the window.
6. Start of command is set to the start of the low wakeup pulse sample.
7. From end of low sync pulse, the signal is searched for the end of data bits by defining a window of 40us.
8. End bits are said to be found when there is a low pulse of length greater than or equal to 35us at a time multiple of 36us from start of data.
9. End of command is set after data end bits are found.
10. Taking the start of command and end of command values, signal is clipped.
11. **Resampling the signal to 10ns:** The signal is resampled at 10ns intervals. All logical analysis is done at 10ns intervals as the standard specifies.
12. The resampled signal is filtered by a median filter to eliminate pulse distortion caused by resampling.
13. **Check for glitch and remove it box:** Then the signal is checked for a glitch. There is a glitch if there is a set of continuous points exceeding the maximum deviation value as specified by the user.
14. If there is a glitch, and its width is within the specified value as set by the user, it is eliminated.
15. **Calculate what is 1st bit and low sync length box:** This box calculates the 1st bit value and low sync length. From the start of the clipped signal, when the signal deviation value goes from positive to negative, it is start of low sync pulse. In the for-loop, 'i' value is initialized to 0 and maximum iterations of the loop is set to 2500 ($i < 2500$). This implies

that the start of low sync will be found before 25us (2500 x 10ns) from start of clipped signal sent for testing. From start of low sync pulse, it is expected to find low sync pulse end before 7351 samples, i.e., before 73.51us. If length of low sync pulse is within 55us, first bit is 0, else first bit is 1.

16. **Calculate start of Data box:** This box gives the sample value where data starts. If the first data bit is 0, data starts after low sync pulse ends. Otherwise it starts 17.95us before end of low sync pulse.
17. **Calculate Avg, min and max data pulse width box:** From start of data to end of data, pulse width of each high and low pulse is calculated.
18. Each pulse is measured for tolerance values.
19. Average pulse width is calculated by adding the width of each pulse and dividing by number of pulses.
20. **Decode data into Binary Box:** From start of data to end of data, the amplitude at the center of each pulse is measured.
21. If amplitude is greater than zero, it is decoded as '1', and if it is less than zero it is decoded as '0'.
22. **Changes binary to Manchester Coding:** If '1' is followed by a '0', it is decoded as bit '0' and if '0' is followed by a '1', it is decoded as bit '1'.
23. **Check stop Bit:** Every 9th bit in the binary data is checked to be a stop bit (zero bit).
24. **Change Manchester to Hex:** Data decoded in binary is converted into hexadecimal format and displayed.
25. Pass or fail indicators indicate if the signal passed the test or not.

15.10 DATA TRANSMITTED TEST VI – LOAD AND TEST

Test Objective

The objective of this test is to verify that the interrogator transmission employs Manchester encoding and that the data timing is within limits.

This test is done off-line without using the signal generator or the signal analyzer. The command captured by the RFSA that is stored in memory can be loaded into Labview and tested.

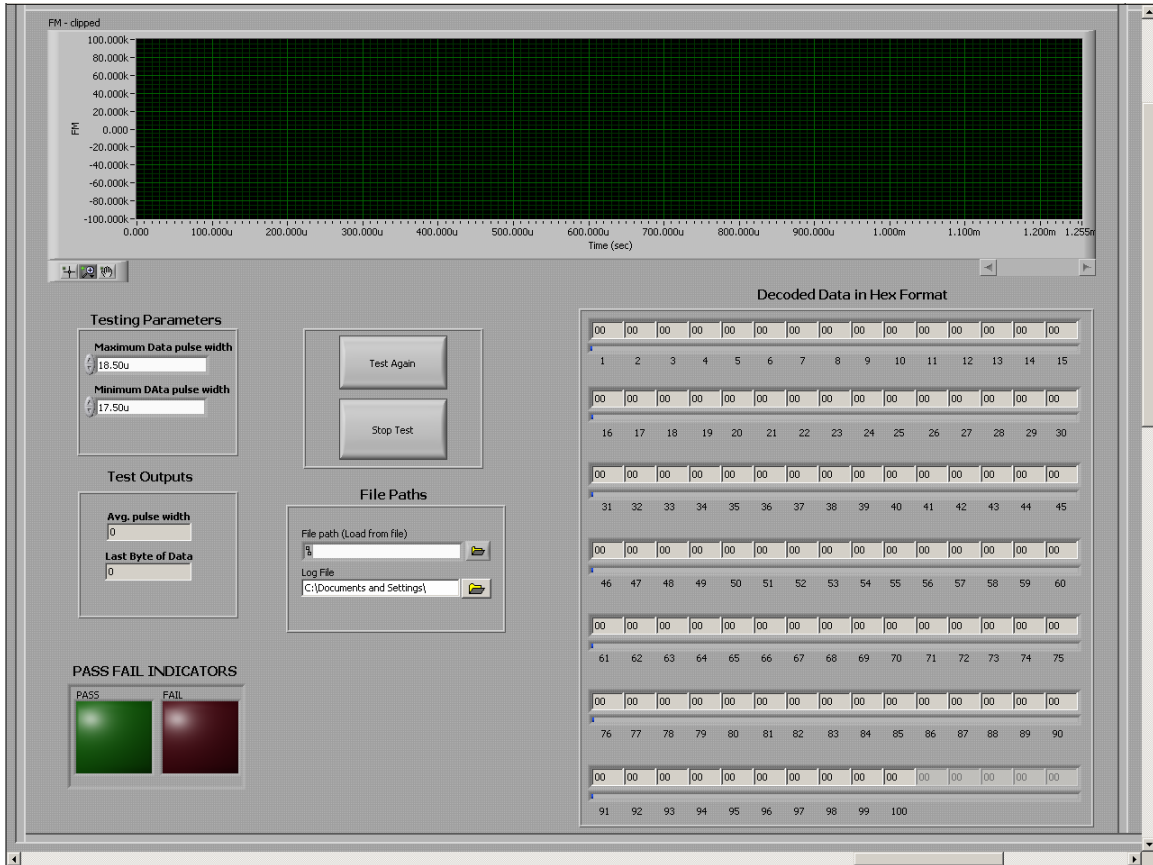


Figure 15.15 Front panel of Data Transmitted Test VI – Load and Test

Inputs

1. Deviation limits of data pulse duration – Maximum pulse duration and minimum pulse duration
2. File Path – specifies where the signal for testing should be loaded from

Outputs

1. Average pulse width
2. Decoded Data in Hex
3. Index of last byte of data
4. Pass or Fail

Procedure to use the VI

1. User will input all the required parameters for testing mentioned above.
2. User will start executing VI.
3. User selects the signal to be loaded and tested.
4. Test is performed.
5. Pass or fail indicators indicate if the test result is pass or fail.

Algorithm of test

1. **Resampling the signal to 10ns:** The signal loaded from memory is resampled at 10ns intervals. All logical analysis is done at 10ns intervals as the standard specifies.
2. The resampled signal is filtered by a median filter to eliminate pulse distortion caused by resampling.
3. **Check for glitch and remove it box:** Then the signal is checked for a glitch. There is a glitch if there is a set of continuous points exceeding the maximum deviation value as specified by the user.
4. If there is a glitch, and its width is within the specified value as set by the user, it is eliminated.
5. **Calculate what is 1st bit and low sync length box:** This box calculates 1st bit value and low sync length. From the start of the clipped signal, when the signal deviation value goes from positive to negative, it is start of low sync pulse. In the for-loop, 'i' value is initialized to 0 and maximum iterations of the loop is set to 2500 ($i < 2500$). This implies that the start of low sync will be found before 25us ($2500 \times 10\text{ns}$) from the start of the

clipped signal sent for testing. From the start of the low sync pulse, it is expected to find the low sync pulse end before 7351 samples, i.e., before 73.51us. If length of low sync pulse is within 55us, first bit is 0, else first bit is 1.

6. **Calculate start of Data box:** This box gives the sample value where data starts. If the first data bit is 0, data starts after low sync pulse ends. Otherwise it starts 17.95us before end of low sync pulse.
7. **Calculate Avg, min and max data pulse width box:** From the start of data to end of the data, the pulse width of each high and low pulse is calculated.
8. Each pulse is measured for tolerance values.
9. Average pulse width is calculated by adding width of each pulses and dividing by the number of pulses.
10. **Decode data into Binary Box:** From start of data to end of data, the amplitude at center of each pulse is measured.
11. If amplitude is greater than zero, it is decoded as '1', and if it is less than zero, it is decoded as '0'.
12. **Changes binary to Manchester Coding:** If '1' is followed by a '0', it is decoded as bit '0' and if '0' is followed by a '1', it is decoded as bit '1'.
13. **Check stop Bit:** Every 9th bit in the binary data is checked to be a stop bit (zero bit).
14. **Change Manchester to Hex:** Data decoded in binary is converted into hexadecimal format and displayed.
15. Pass or fail indicators indicate if the signal passed the test or not.

15.11 DECODE DATA - LOAD AND TEST

Test Objective

The objective of this test is to verify that the signal transmission employs Manchester encoding, the data format is correct and the data transmitted are correct.

This test is done off-line without using the signal generator or the signal analyzer. The command captured by RFSA is stored in memory can be loaded into Labview and tested. The VI reads data from a file and compares it with the decoded data from the signal.

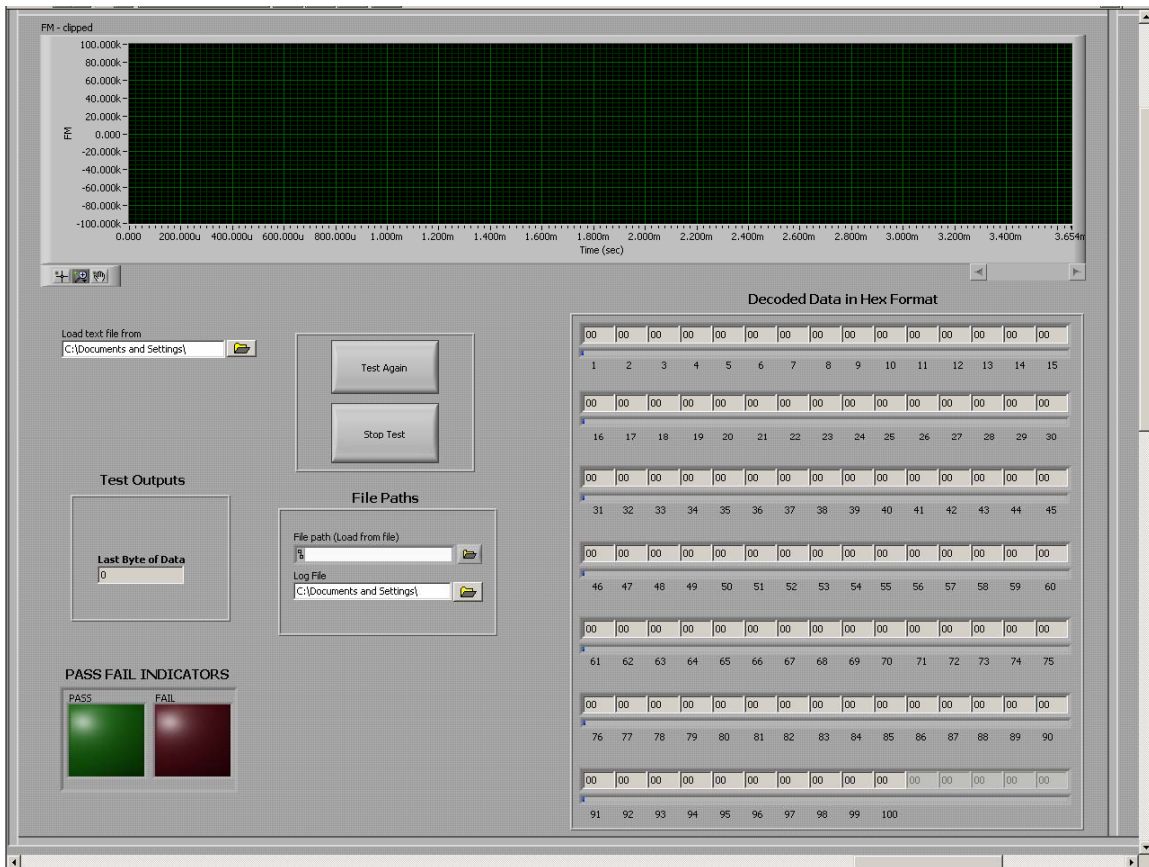


Figure 15.16: Front panel of Decode Data - Reader to tag - Load and Test

Inputs

1. Load text file from - specifies from where the text file with description and data of the signal for testing should be loaded. Text file should be in the format:

$$\langle \text{Description of command} \rangle : 0x \langle \text{Data in Hex} \rangle ;$$
2. File Path – specifies where the signal for testing should be loaded from

Outputs

1. Decoded Data in Hex
2. Index of last byte of data
3. Pass or Fail

Procedure to use the VI

1. User will input all the required parameters for testing mentioned above.
2. User will start executing VI.
3. User selects the signal to be loaded and tested.
4. User selects the text file to be read.
5. Test is performed.
6. Pass or fail indicators indicate if the test result is pass or fail.

Algorithm of test

1. Read text file and store data in an array.
2. Load signal from path specified.
3. **Resampling the signal to 10ns:** The signal loaded from memory is resampled at 10ns intervals. All logical analysis is done at 10ns intervals as the standard specifies.
4. The resampled signal is filtered by a median filter to eliminate pulse distortion caused by resampling.
5. **Check for glitch and remove it box:** Here the signal is checked for a glitch. There is a glitch, if there is a set of continuous points exceeding maximum deviation value as specified by the user.
6. If there is a glitch and the width of the glitch is within specified value as set by the user, it is eliminated.
7. **Calculate what is 1st bit and low sync length box:** This box calculates 1st bit value and low sync length. From the start of the clipped signal, when the signal deviation value goes from positive to negative, it is the start of low sync pulse. In the for-loop, 'i' value is initialized to 0 and maximum iterations of the loop is set to 2500 ($i < 2500$). This

implies that the start of the low sync will be found before 25us (2500 x 10ns) from start of clipped signal sent for testing. From the start of the low sync pulse, it is expected to find the low sync pulse end before 7351 samples, i.e., before 73.51us. If the length of low sync pulse is within 55us, the first bit is 0, else first bit is 1.

8. **Calculate start of Data box:** This box gives the sample value where data starts. If first data bit is 0, data starts after low sync pulse ends. Otherwise it starts 17.95us before end of low sync pulse.
9. **Decode data into Binary Box:** From start of data to end of data, the amplitude at the center of each pulse is measured.
10. If amplitude is greater than zero, it is decoded as '1', and if it is less than zero it is decoded as '0'.
11. **Changes binary to Manchester Coding:** If '1' is followed by a '0', it is decoded as bit '0' and if '0' is followed by a '1', it is decoded as bit '1'.
12. **Check stop Bit:** Every 9th bit in the binary data is checked to be a stop bit.
13. **Change Manchester to Hex:** Data decoded in binary is converted into hex and displayed.
14. Pass or fail indicators indicate if the signal passed the test or not.

15.12 LOAD AND ANALYZE A LONG CAPTURE VI

VI Objective

The objective of this VI is to analyze a long capture of data transmission between wireless devices.

This analysis is done off-line without using signal generator or the signal analyzer. The long capture by RFSA that is stored in memory can be loaded into Labview and analyzed. The capture can be made using "ni5660 Get FM – streaming.vi".

Inputs

1. Load capture from file - specifies from where the long capture should be loaded from.
2. Command captured – specifies which command (out of many in the capture) to be shown on the window.
3. File Path – path where the command should be saved

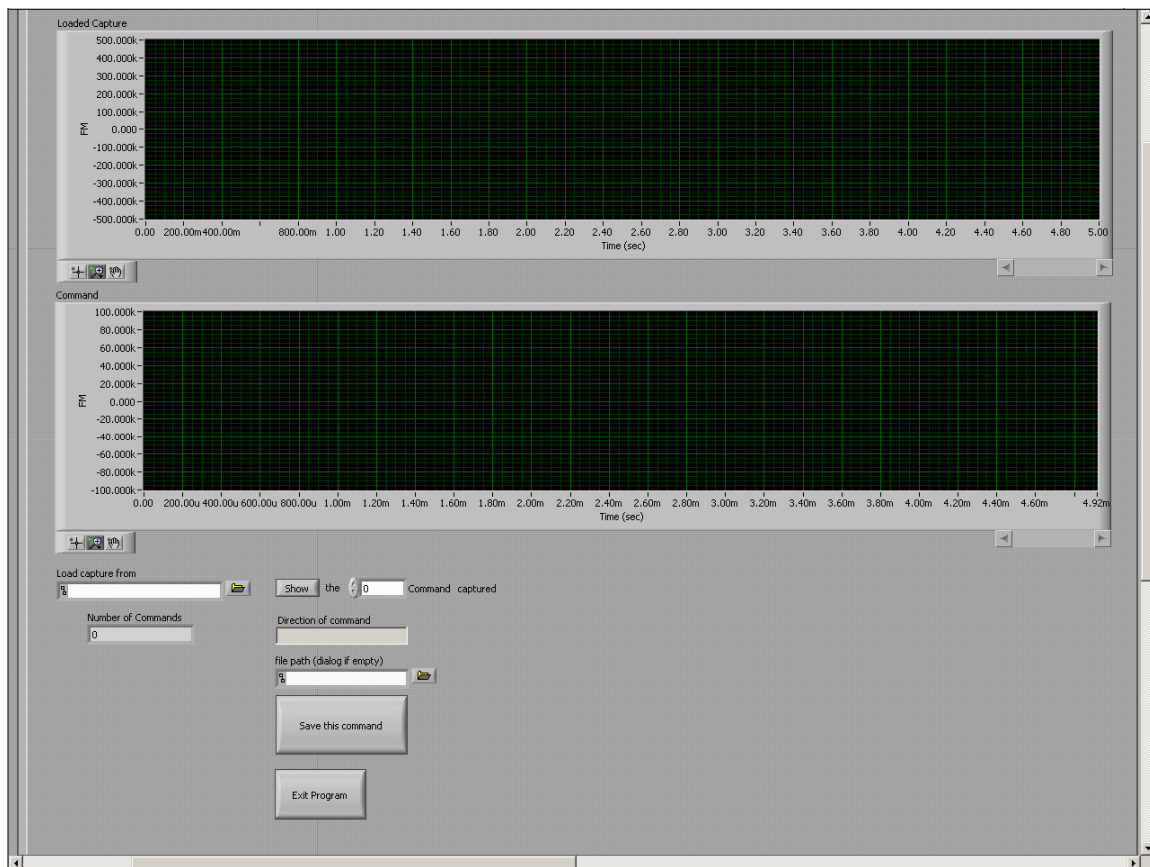


Figure 15.17: Front panel of Load and analyze a long capture VI

Outputs

1. Number of commands
2. Direction of each command

Procedure to use the VI

1. User will input all the required parameters for test mentioned above.
2. User will start executing the VI.
3. After the “number of commands” is displayed (indicates that analysis is done) user selects which command to display.
4. User can save the command in the Labview format by pressing *save this command* button.

15.13 RECEIVER BANDWIDTH TEST VI

Test Objective

The objective of this test is to verify that the interrogator has a minimum -3dB receiver BW conforming to standard ISO 18000-7.

Inputs

1. Resource Name
2. Waveform to transmit
3. Carrier Frequency
4. Reference level
5. RSSI tolerance
6. Delay time
7. Calibration Table (see procedure for more details)
8. Apply
9. Exit

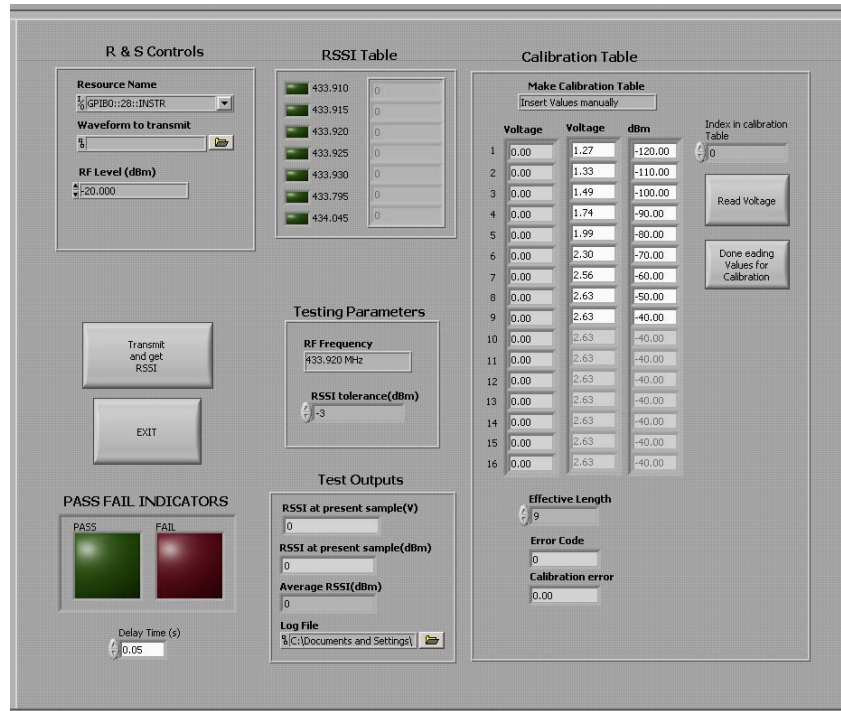


Figure 15.18: Receiver Bandwidth Test VI

Procedure to use the VI

1. User will start executing VI.
2. User will specify if the calibration values are to set manually or are to be acquired from the oscilloscope.
3. If they are set manually then proceed to step 7.
4. If the values are to be read from the oscilloscope, set the index in calibration table to the required value and press *read voltage*.
5. This will update the calibration table voltage values.
6. After the system is done with reading values press *done reading voltages for calibration*.
7. User will then input the remaining inputs required for testing.
8. User will select RF frequency from a drop down menu.

9. User will press *apply* to transmit signal at the chosen frequency and obtain the RSSI at that frequency.
10. LEDs display at what frequencies the RSSI value is read.
11. Pass or fail indicators indicate if the test result is pass or fail.

Algorithm of test:

1. From the inputs given the VI configures the R&S SMJ100A RFSA to generate a signal at that frequency.
2. When the *apply* button is pressed the signal is transmitted and also the VI communicates with the HP oscilloscope and obtains the RSSI value as a function of DC voltage.
3. The RSSI value is stored in an array, and the LED corresponding to that frequency glows indicating the sample is taken.
4. After all samples are taken, the average value of RSSI is calculated.
5. The result is compared to RSSI at -3dB frequencies based on RSSI tolerance value.
6. Test is passed if RSSI at each -3dB frequency is within the tolerance value specified from the average RSSI calculated from the other five frequencies.

16.0 RESULTS

This section discusses and explains the results of testing different signals if they conform to the ISO 18000-7 standard using various VIs developed for different tests. The signals have been generated by the RFSG which is programmed to emulate the interrogator.

Figure 16.1 illustrates a command that is tested using FSK Deviation test VI, and it conforms to the standard. The green light is active indicating that the command has passed the test.

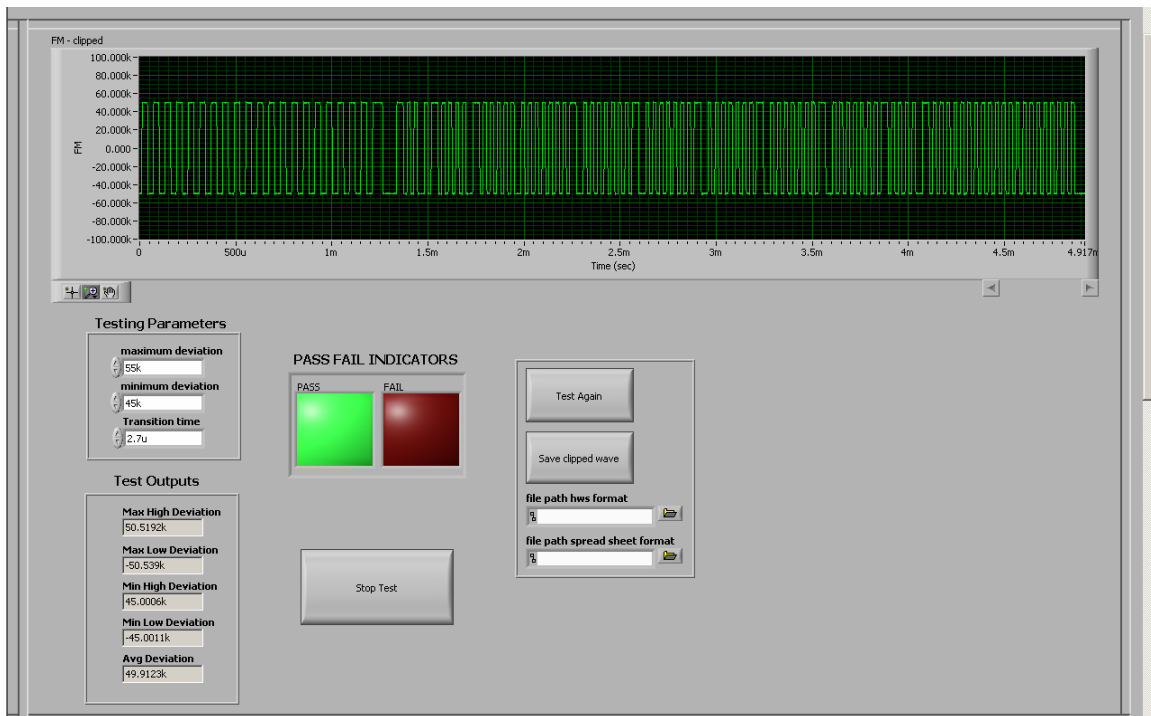


Figure 16.1: A signal that passes the FSK Deviation Test

The same FSK Deviation test has been performed on another command as shown in figure 16.2. This command does not pass the test. The frequency deviates above and below the boundary values specified by the standard. Hence, the red light is active indicating that the command failed the test.

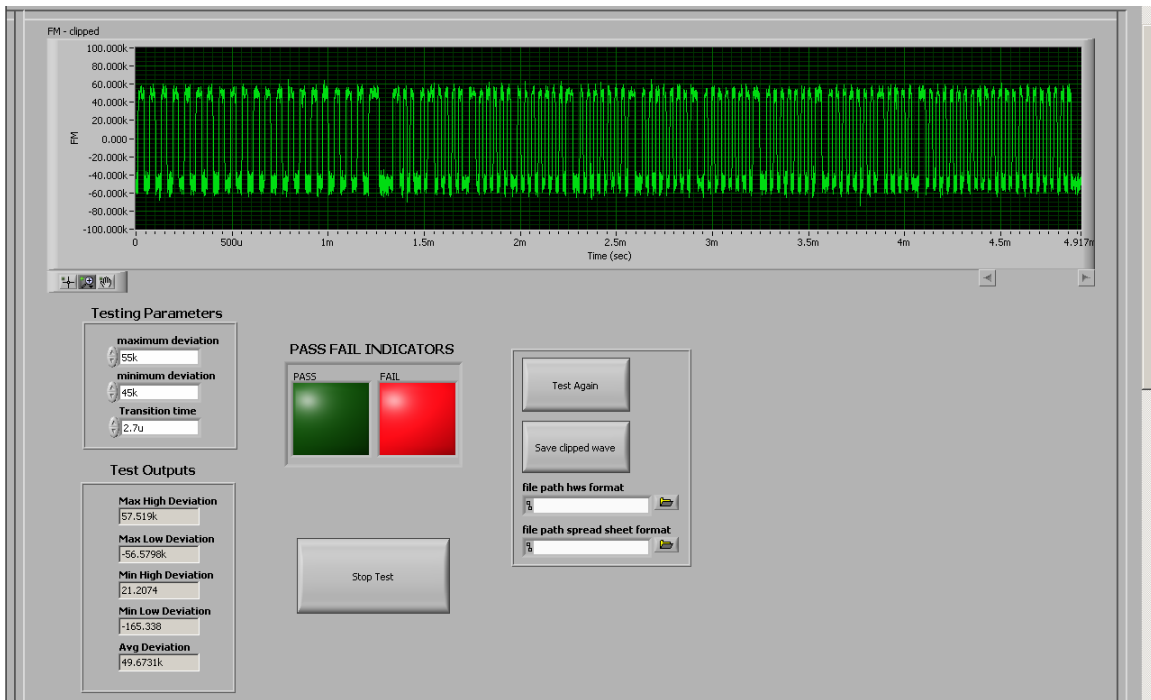


Figure 16.2: A signal that fails the FSK Deviation Test

Figure 16.3 illustrates a command tested that its carrier frequency is within operating limits using Carrier Frequency test VI. The command passed the test.

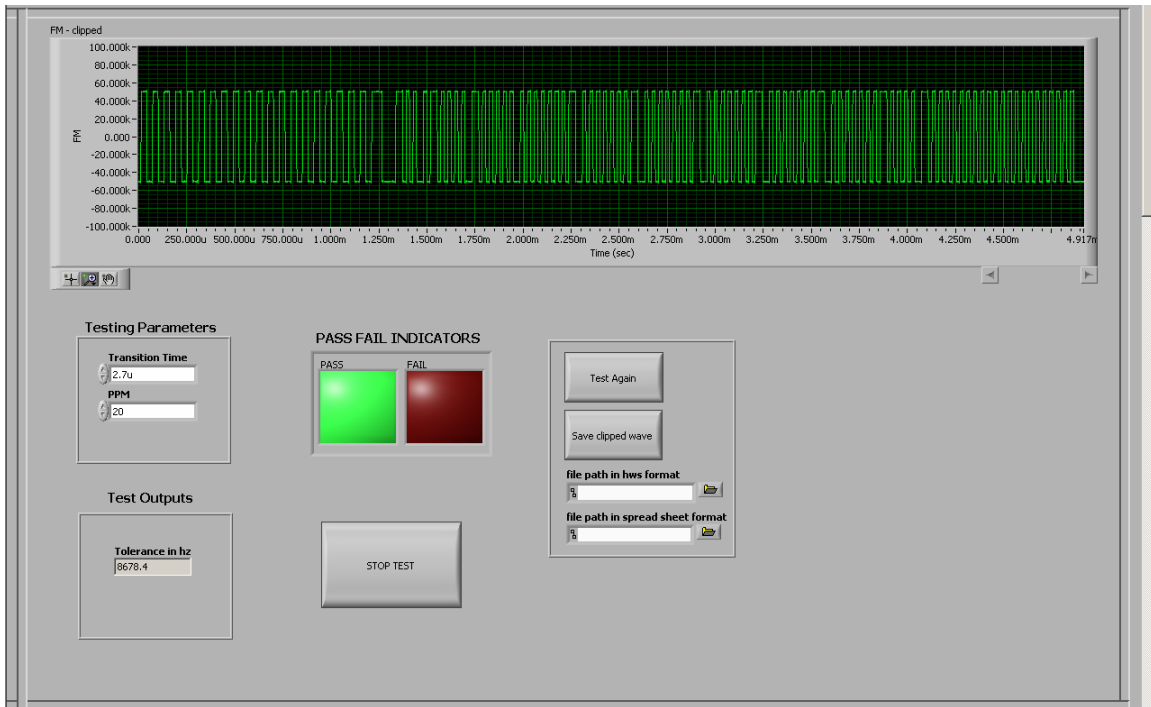


Figure 16.3: A signal that passes the Carrier Frequency Test

The same carrier frequency test is performed on the command illustrated in Figure 16.4 below. This command fails the test. The carrier frequency has deviated more than the permissible range.

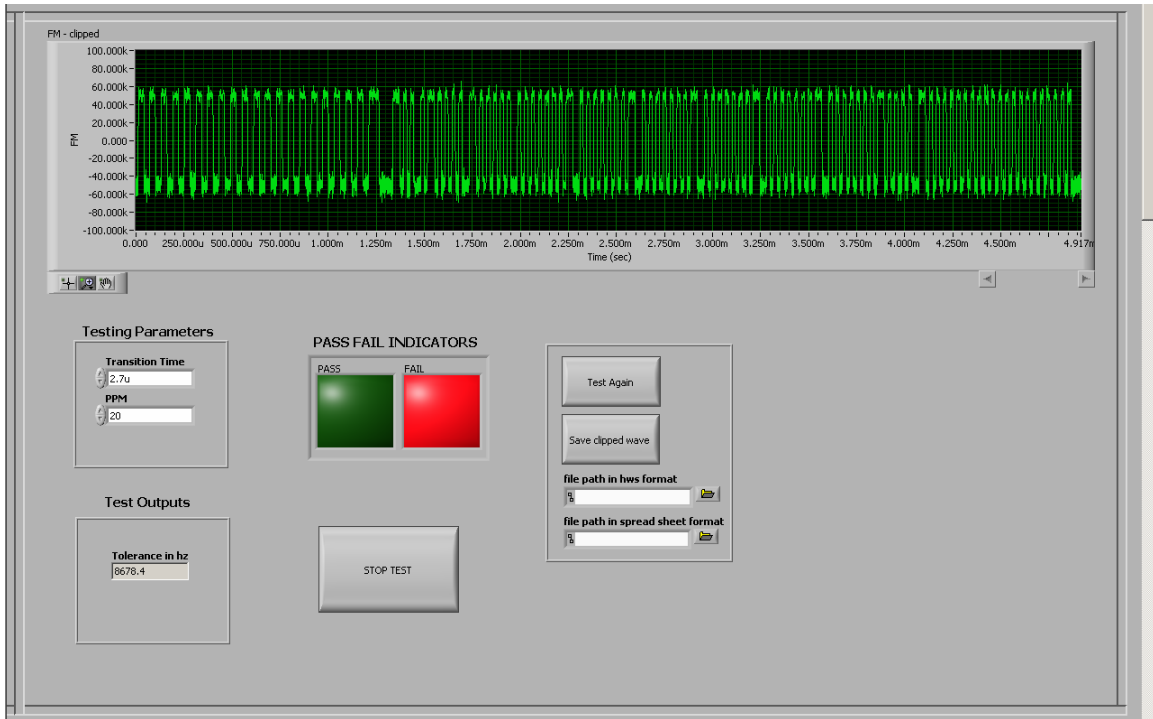


Figure 16.4: A signal failing Carrier Frequency Test

Figure 16.5 illustrates results of preamble test performed on a command that passes. The VI used to test the command is Preamble test. The command passes the test.

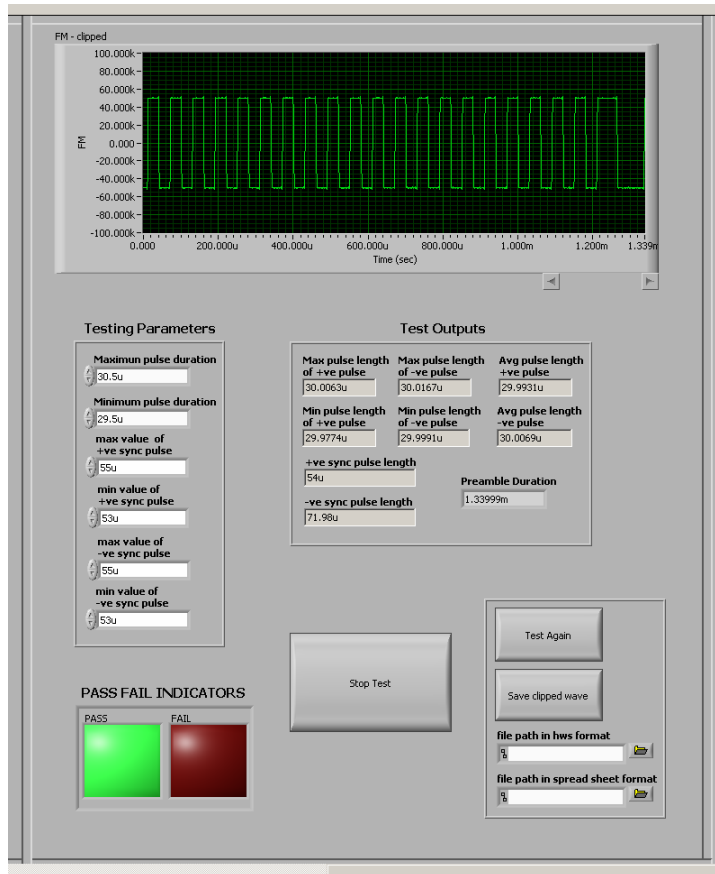


Figure 16.5: A signal passing Preamble Test

Figure 16.6 shows a command which fails the preamble test because it is missing one entire high pulse and an entire low pulse. Also the fourth low pulse width is less than the minimum pulse width specified by the standard.

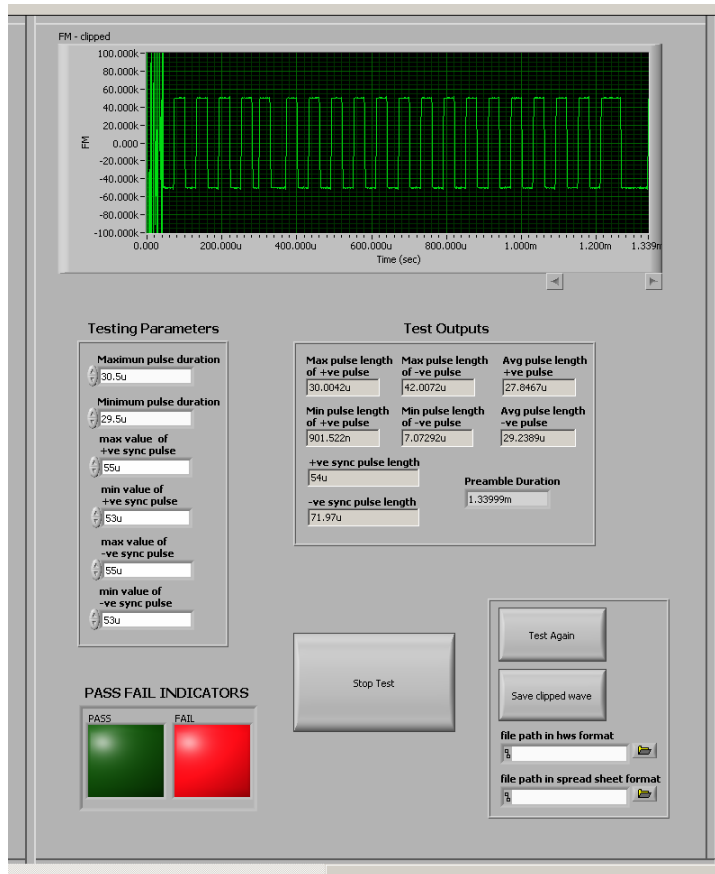


Figure 16.6: A signal that fails Preamble Test

Figure 16.7 illustrates the results of a test where the data transmitted by the command are in the correct format with the data pulse widths all within boundary conditions. The decoded data are also shown.

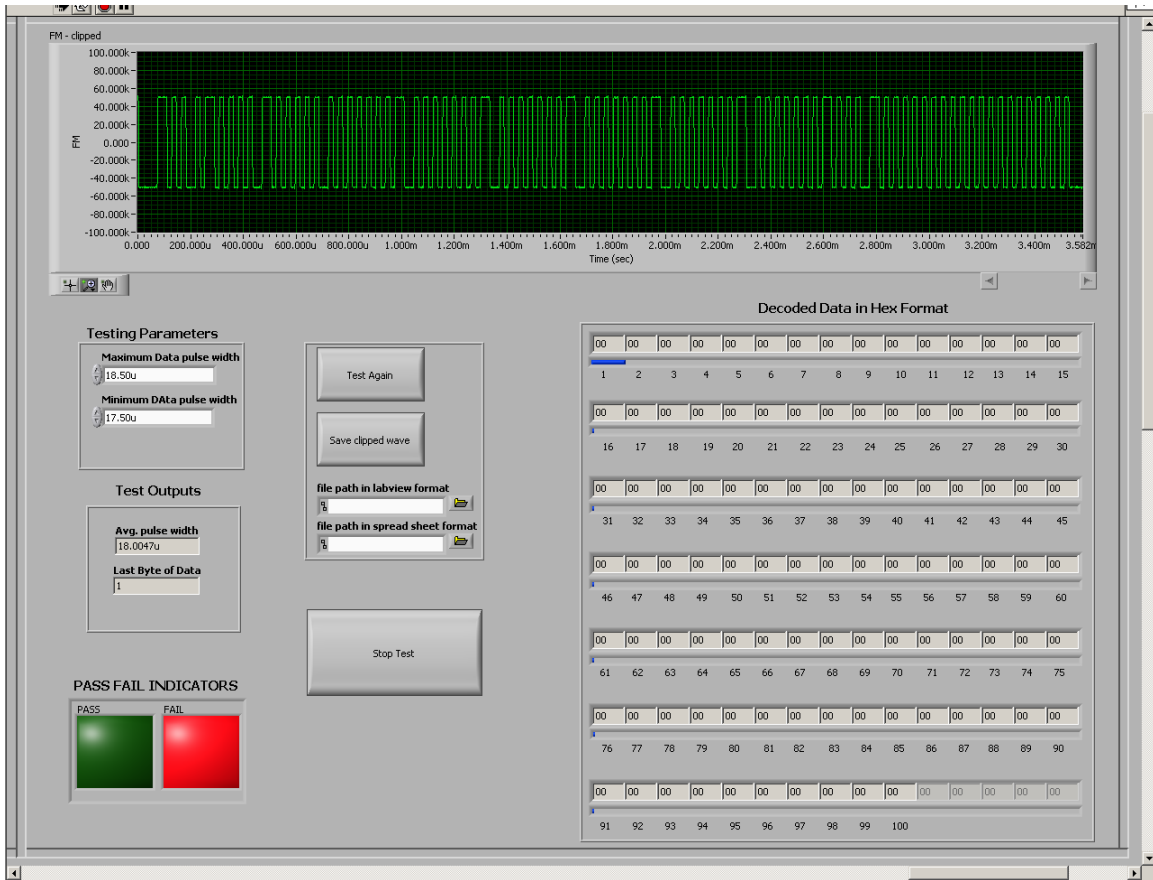


Figure 16.8: Signal with improper data format that fails the test

The test VIs also allow the commands captured to be saved and retested. Once the command is captured, it can be tested many times using different test parameters each time. The VIs also output a log file describing the test parameters of each test and the test results. The log file of Data Test VI is shown below.

Table 16.1: Example Log File

```

*** Data Test VI ***

Test Started at : 8:20:21.718 PM 6/23/2007

***NI-5660 Settings***
Down Converter Device Number : 2

```


Table 16.1 (Continued)

Test Results Test #3
Sync Pulse found
Average Data pulse width : 18.004495u
Test Result : PASS
Test performed at 8:21:10.890 PM 6/23/2007

Decoded Data in Hex format :
Byte #1 = 31
Byte #2 = 04
Byte #3 = FF
Byte #4 = FF
Byte #5 = FF
Byte #6 = FF
Byte #7 = FF
Byte #8 = FF
Byte #9 = 10
Byte #10 = 00
Byte #11 = 00

As is described in the log file, the command fails for the first time when the data pulse boundary conditions are 17.5us and 18.5us. The signal also fails when the boundary conditions are 17.5us and 19us. However the signal passes when boundary conditions are increased to 16us and 20us. Figure 16.9 below shows the signal that passes when pulse width boundary conditions are increased to 16us and 20us.

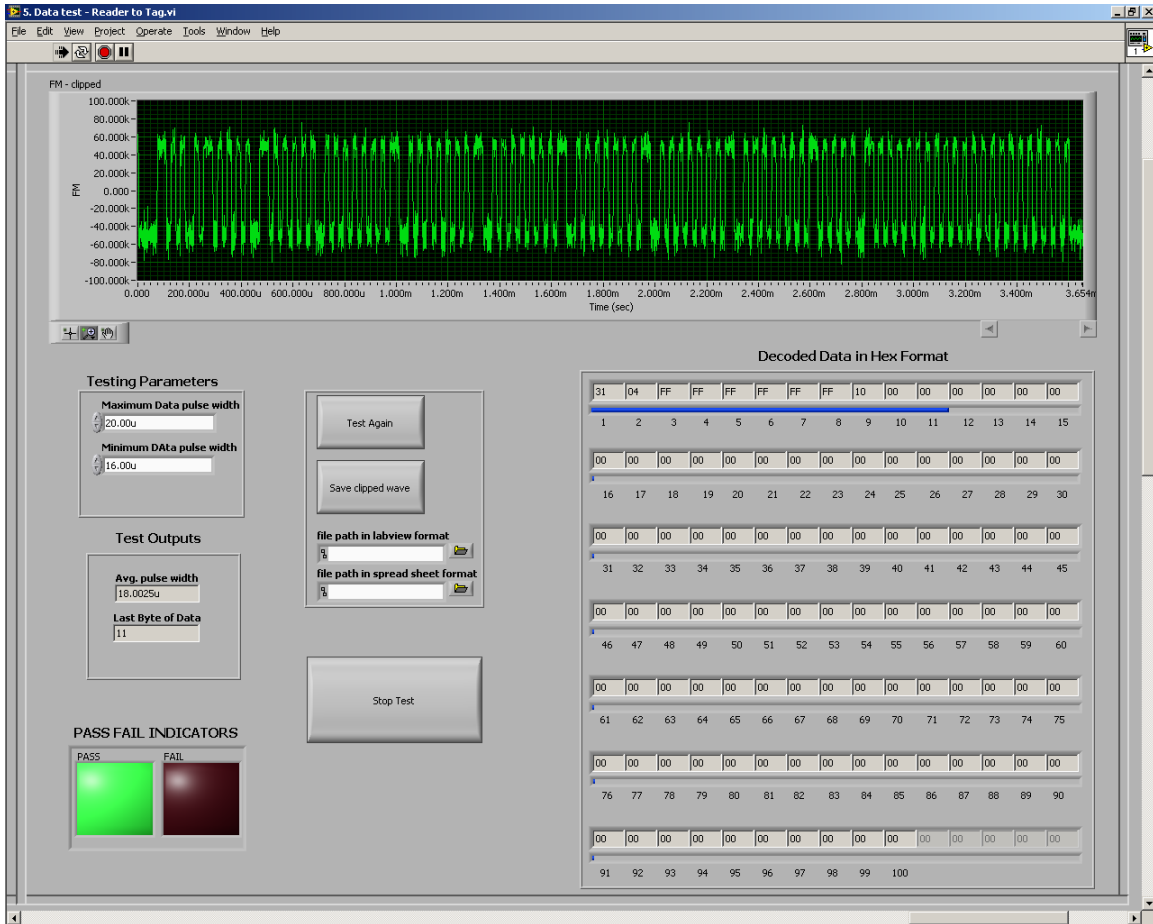


Figure 16.9: Command that passes data test

Figure 16.10 illustrates the VI that analyzes a long capture and gives the number of commands, their direction and their start and end times within the capture. The signal loaded is a data transmission between two wireless devices. Commands can be distinguished from noise in the capture by searching the capture for the sync pulse in each command. There are as many commands in the capture as there are sync pulses. The second command out of total nine commands in the capture is shown in figure 16.10.

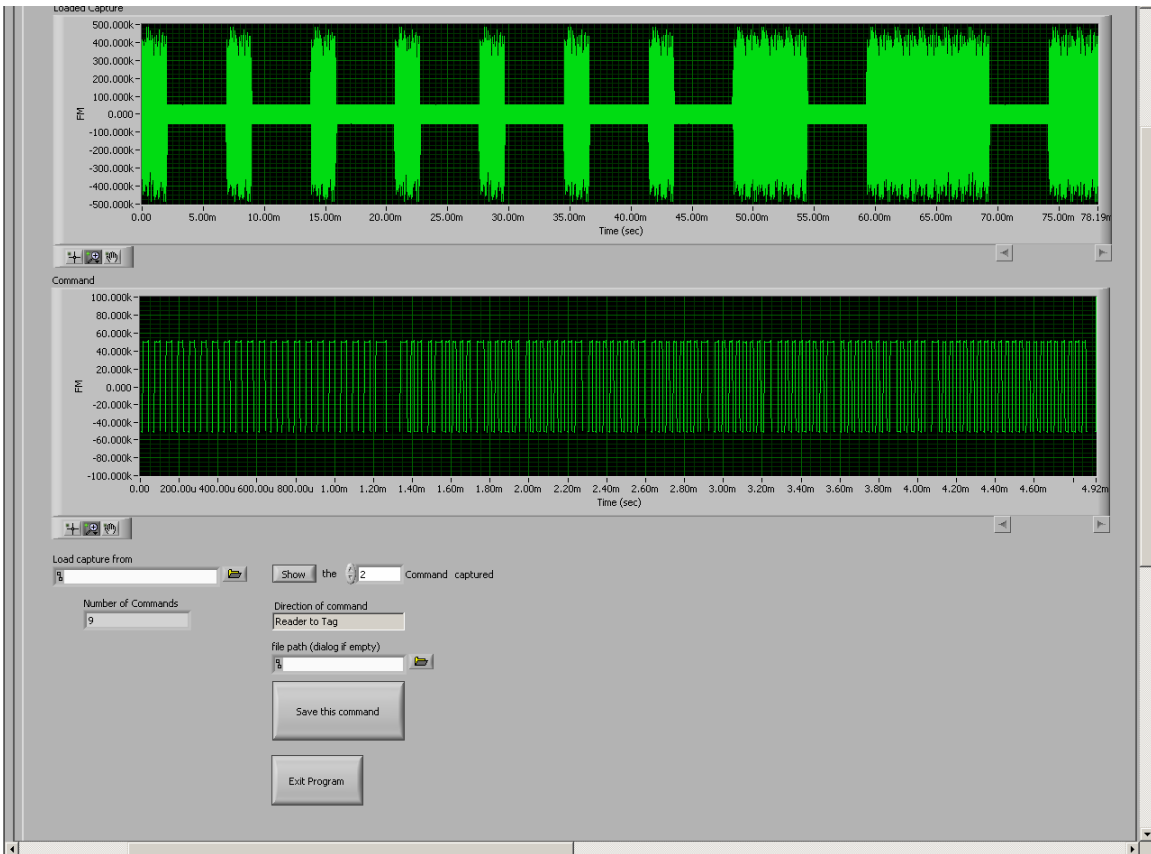


Figure 16.10: Analyzing a long capture

Test timings

Table 16.2 below shows the time taken by each VI to test the same command. The PC used had a 3GB RAM and two 1.5GHz processors.

Table 16.2: Test Timings

S.No	Test Name	Direction	Trial 1	Trial 2	Trial 3
1	FSK Deviation Test	R-T	13.2sec	13.7sec	13.3sec
2	FSK Deviation Test	T-R	14.1sec	13.7sec	13.4sec
3	Carrier Freq Test	R-T	13.4sec	13.4sec	13.8sec
4	Carrier Freq Test	T-R	13.5sec	13.7sec	13.4sec
5	Preamble Test	R-T	10.2sec	8.3sec	10.4sec
6	Preamble Test	T-R	10.3sec	10.4sec	10.2sec
7	Data Test	R-T	11.4sec	11.2sec	10.9sec
8	Data Test	T-R	10.2sec	11.5sec	11.2sec
9	Wakeup Capture		6.02min	5.25min	6.01min
10	Wakeup Test		57.8min	56.9min	59.1min

Test Memory Requirements

Table 16.3 illustrates the memory required by the VI including all sub VIs it calls. It is important to understand the memory requirement of the tests because the maximum size of an array that can be handled by Labview 8.2 is 800MB.

Table 16.3: Test Memory Requirements

S.No	Test Name	Direction	Average Bytes	Min Bytes	Max Bytes
1	FSK Deviation Test	R-T	34868.34k	34868.34k	34868.34k
2	FSK Deviation Test	T-R	34868.27k	34868.27k	34868.27k
3	Carrier Freq Test	R-T	92061.99k	92061.99k	92061.99k
4	Carrier Freq Test	T-R	92058.99k	92058.99k	92058.99k
5	Preamble Test	R-T	13030.01k	13030.01k	13030.01k
6	Preamble Test	T-R	13022.01k	13022.01k	13022.01k
7	Data Test	R-T	10131.19k	10131.19k	10131.19k
8	Data Test	T-R	10130.19k	10130.19k	10130.19k
9	Wakeup Capture & Test		672034.72K	672034.72K	672034.72K

17.0 CONCLUSION

In my research, to develop fully automated test procedures for wireless link standards, complex and modern RF equipment were utilized (NI RF Signal Analyzer and R&S RF Signal Generator). Procedures were developed by me to configure the test equipment and setup the entire test environment using a host computer through remote commands.

I have partitioned the wireless commands based on their characteristics for analysis. This led to development of various procedures for testing parameters of the commands and their response transmitted through air.

Test procedures (VIs) were developed in Labview 8.2 for verifying that commands and their responses conform to ISO 18000 – Part 7 standard. All the tests are packaged into simple scripted testing procedures. The output of each test is given as a simple pass or fail, indicated by LEDs, along with a text file recorded with the input parameters and explaining the entire procedure of the test.

The entire data transfer can be captured and saved as waveforms. The transmitted signals can be demodulated and decoded. The results of each test can be saved for analysis.

A manual was written explaining in detail the procedure to use each VI. The steps include configuring the test equipment, inputting test parameters and analyzing outputs.

I have translated the ISO 18000 - Part 7 standard conformance requirements into the procedures as an example application. This research is the first fully automated conformance test procedures developed for testing a RFID system.

18.0 FUTURE DEVELOPMENTS

The automated test procedures developed for the ISO 18000 – 7 standard can be modified to develop automated test procedures for standards such as the ISO 18185 and ISO 18000 – 6c. The present test procedures were developed such that it is possible to modify the test algorithms if the standard (ISO 18000 -7) itself is updated.

The portion of the program (VI) where the RFSG is programmed to send out an arbitrary waveform can be reused for developing testing procedures for any standard without any modifications. The same VIs can be used to transmit waveforms generated using WinIQSim. WinIQSim supports many modulation schemes as mentioned in Table 8.3. Upon generation of appropriate “.wv” file in WinIQSim (with required modulation, symbol rate etc.), it can be loaded and transmitted by RFSG using the VIs already developed.

Parts of the program where the RFSA is programmed to acquire the RF signal can be reused to capture signals irrespective of the modulation scheme used by the transmitter. As mentioned in Section 9.0 , the captured signal is sent to the computer in the form of IQ. This IQ, when passed through the corresponding demodulator VI, will give us the desired waveform demodulated from IQ. Demodulator VIs are available in Labview Modulation package.

After acquiring the demodulated waveform, it has to be searched for the command. As explained in Section 14.2, every command has some unique sequence of pulses such as the sync pulse or the preamble that can be distinguished from all other pulses. A block named “finding sync pulse box” is a program written in ‘C’ that can be adapted to find sync pulse of any pulse widths. This algorithm can be used to find the sync pulse, start and end of any command within

the demodulated waveform. Figure 18.1 shows the clip command sub VI implemented for the ISO 18000 – 7 standard.



Figure 18.1: Clip command SubVI for ISO 18000-7

Once the signal is clipped from the demodulated waveform, it is in the form of a structure containing start time of signal, “dt” and an array of numbers. Most of the conformance tests include comparative operations. User defined functions can be written in ‘C’ and also Matlab to analyze and process array values. Therefore, it is easy to develop new testing procedures for different standards based on the present setup. Where it is required to measure pulse width, duty cycle and time period of periodic waveforms, we can use “Pulse Measurements” VI to do calculations.

Data can be acquired continuously for a long time (10sec) using the VI named “ni5660 Get FM – streaming”. This VI can be modified to capture signal of any modulation. This makes it possible to record the entire data transfers (commands and their response) between two wireless devices. Then the VI “How many commands” can be used to analyze the capture and calculate the start and end of each command in the entire capture. The commands can be clipped and saved onto the hard disk. The saved commands can be loaded and tested. Figure 18.2 shows “How many commands.vi”, which analyzes a captured waveform for a long period and finds all commands conforming to the ISO 18000-7 standard.

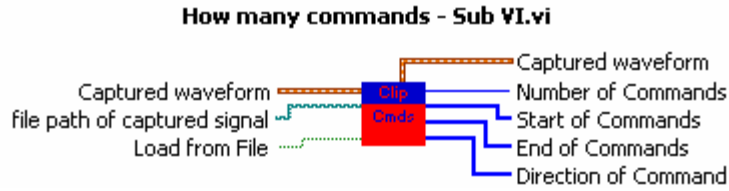


Figure 18.2: How many commands - SubVI

As mentioned before, the output of most RFSAs is IQ data which is saved to disk as I and Q data separately (but not as IQ). VIs are developed that accept I and Q data as input saved in a text file and convert them into IQ data. This IQ data can be demodulated using the demodulator VIs in Labview and the resulting waveform can be searched for commands which can be clipped and saved to memory. The saved commands can be loaded and tested later. This gives a platform for developing testing procedures for different standards using different hardware equipment. Figure 18.3 show the different VIs for converting I and Q data into waveform.

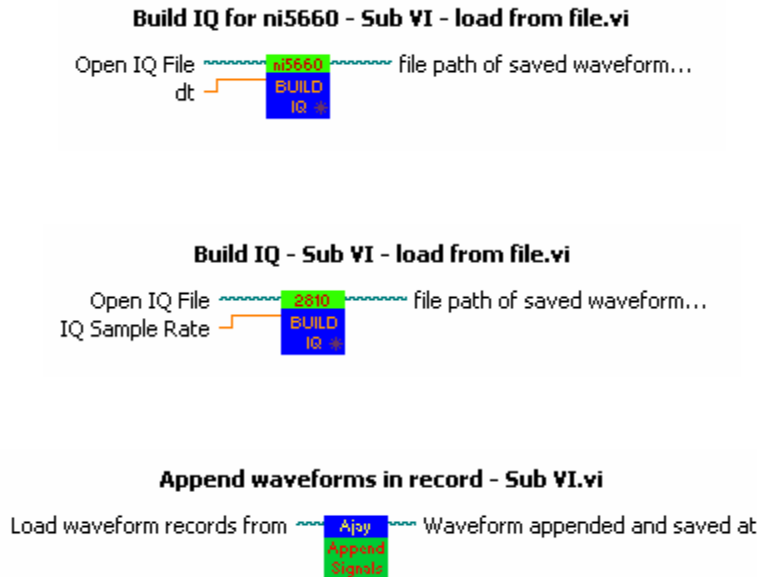


Figure 18.3: VIs that convert I and Q data into waveform

The only difference between Build IQ VIs built for the NI5660 RFSA and the Keithley 2810 RFSA is that NI5660 RFSA gives “dt” along with the I and Q. Whereas the Keithley 2810 RFSA gives IQ sample rate along with it’s I and Q. Therefore, build IQ for 2810 takes the IQ sample rate and calculates its inverse inside the VI to get “dt” and uses this value to generate the waveform. By making such minor modifications all of the VIs can be reused for developing different test procedures for other wireless standards.

18.1 ADAPTING TO THE ISO 18000-6C STANDARD

The test setup developed for the ISO 18000 – 7 standard handles the ISO 18000 – 6c standard with few modifications. The most fundamental change is that the ISO 18000 – 6c standard uses ASK modulation for communication. This can be easily emended by replacing the FM demodulator VI with AM demodulator VI. This allows testing of RFID Readers and Tags by transmitting GOLD standard commands that are ASK modulated.

An elementary step in automated testing for conformance is to clip the command or the response from the RF data captured by the RFSA. As the ISO 18000 – 6c standard does not include a sync pulse in the signals transmitted; the test procedures can be programmed to search for the preamble pulse in the capture.

After the command or response is clipped, the entire potential of Labview can be used to analyze the signal.

18.2 ADAPTING TO THE ISO 18185 – 7 TYPE A STANDARD

The test setup developed for the ISO 18000 – 7 standard also handles the ISO 18185 – 7 Type A standard. The ISO 18185 – 7 Type A standard used On-Off Keying (OOK Modulation). The ASK demodulation VI can be used to demodulate OOK modulated signal.

Similar to the ISO 18000 – 6c standard, the test procedures for the ISO 18185 – 7 Type A standard can be programmed to search for the preamble pulse in the capture as there is no sync pulse.

Currently research is being done to develop test procedures for the ISO 18185 – 7 Type A standard based on test procedures developed for the ISO 18000 – 7 standard.

BIBLIOGRAPHY

- [1] John D. Bowne, Standards Based Wireless Device Testing, Rohde&Schwarz, 2003.
- [2] David Murfett, The Challenge of Testing RFID Integrated Circuits, TAGSYS Australia Pty Ltd, 2004.
- [3] Matt Ward, Rob van Kranenburg, RFID: Frequency, standards, adoption and innovation, JISC Technology and Standards Watch, May 2006.
- [4] Elaine May, RFID Testing Challenges, December 2006.
- [5] Ian J Shelley & Rachel Ho, Applications of Radio Frequency for the Construction Industry, Intrare Consulting Pte Ltd, Singapore, January 2003.
- [6] Dean Miles, RTSA aids RFID testing, Spectrum Analyzers Tektronix Inc.
- [7] Edmund W. Schuster, FDA, Counterfeit, and RFID Technology.
- [8] David Hanny, Manuel A. Pachano, Les G. Thompson, RFID Applied, 2007
- [9] www.whatis.techtarget.com
- [10] www.ni.com
- [11] www.rhodes-schwarz.com
- [12] www.wikipedia.org
- [13] The ISO/ICE 18000 Part 7 Standard: Test methods for active air interface communications at 433MHz.
- [14] Class-1 Generation-2 UHF Air Interface Protocol Standard Version 1.0.9 from EPCglobal Inc.
- [15] The ISO/FDIS 18185 – 7 Type A standard.
- [16] Alan V. Oppenheim, Ronald W. Schafer and John R. Buck, Discrete – Time Signal Processing, 2nd Edition