Transmit and Receive Testing of Digital Waveforms using R&S®CMA180

Application Note

The compact R&S®CMA180 Radio Test Set allows straighforward tests on devices that use analog or digital modulation and demodulation formats. This application note focusses on the use of the digital radio capabilities for measurements on radio transceiving devices in terms of their functionality and parameter characterization. Also shown is how to use the instrument with the various software tools available from Rohde & Schwarz for creating and evaluating digitally modulated signals. Various software examples are provided as free download along this paper.

This application note addresses readers who develop, produce or optimize digital radio transmitting and receiving devices or maintain analog radio communication equipment alongside those with digital modulation.

Note:

Visit our homepage for the most recent version of this application note

(http://www.rohde-schwarz.com/appnote/1MA277).

CMA180 Digital Modulation ─ 1MA277_0e

Dieter Bues

Application Note

ROHDE&SCHWARZ

Contents

1 Introduction

IQ modulation can be considered as an abstraction of classic modulation methods like AM, FM and Phim. This way the radio test Set R&S®CMA180 along with it is integrated IQ-modulator/recorder is not only a successor of the Radio Communication Tester R&S®CMS, but also an instrument for tests on digital modulation formats such as eg. FSK, QAM and OOK. The application note shows how to use the R&S®CMA180 for generation and evaluation of these digital modulation schemes in comm's equipment or digital remote control devices.

The presentation focuses on practical examples that have been carefully implemented and tested and are easy reproducible. As far as program code was used either in C++, MATLAB or Python the appropriate source code is provided in the download area of the application note.

The configuration described in chapter 2.1 was used for all measurements presented. For the OOK example, I/Q data was sampled by the standard "FFT spectrum" function of R&S®CMA180. The pulse evaluation was done using the freely available R&S Forum software environment, which the reader is encouraged to download from reference [6].

The following abbreviations are used in this application note:

R&S®CMA180 Radio Test Set:	CMA180
R&S®CMS Radiocommunication Service Monitor:	CMS
R&S®SMW200A Vector Signal Generator:	SMW
R&S®VSE Vector Signal Explorer Software:	VSE
R&S®WinIQSIM2™ Simulation Software:	WinIQSIM2
R&S®ARB Toolbox Plus:	ARB Toolbox

2 Tools for Creating and Evaluating IQ-Data

This chapter provides an overview of the tools needed to create and evaluate IQ modulated signals along with CMA180. The first part describes the required CMA180 options, the second the available software tools.

2.1 Options Needed on CMA180

Baseband Generator

The arbitrary waveform generator (AWG) option CMA-B110B works as a modulation source within the CMA180 [1]. The option can be retrofitted in R&S service. It provides all possible modulations up to 20 MHz bandwidth because it is internally connected to the CMA180 I/Q modulator. The resolution is 16 bit for I and Q signals each. With its memory size of 1 GByte, it provides sufficient memory space to store and to play even complex IQ-sequences. The AWG is required for all applications mentioned here, because it is a prerequisite for the IQ-recorder. Various waveform creation methods for the AWG are described in the application note.

IQ Recorder

The R&S®CMA-K220 I/Q recorder option can sample and store I/Q data directly from CMA180's internal IQ-demodulator. With a sampling rate up to 100 MHz, depending on the filter, it can store up to 4 MSamples of I/Q data. The option is retrofitable at the user's premises. Along with the required AWG, it provides good performance as shown in the upcoming examples.

2.2 Software Tools

ARB Toolbox

This free R&S tool is used to transfer I/Q data streams from one format to another. The software also offers additional features. It is available under [3]. The software can also be used for creating ARB waveform files from custom I/Q data, import data from MATLAB .mat files, evaluate waveform content graphically, resample or filter waveforms, create analog modulated signals as ARB files and edit marker data in waveform files. ARB Toolbox is available for Windows 7/8/10, Mac OS X and Linux.

RSCommander

RSCommander [8] is a versatile software tool for a wide range of Rohde & Schwarz instruments. It makes it possible to take screenshots for documentation purposes, read trace data for further processing and upload user correction data to signal generators for linearization. The software also allows for file management and interactive control of the instrument using its remote control commands. RSCommander can automatically detect all Rohde & Schwarz instruments connected via GPIB and LAN interfaces.

WinIQSIM2

WinIQSIM2 is a software freely available on the R&S homepage [4]. The interactive GUI-controlled software is closely related to the GUIs of the R&S signal generators, e.g. SMW. The easy to operate software includes all main digital modulation standards, e.g. GSM, WLAN, NFC, avionics & navigation. Even though the software is free, some standards are bound to specific license options which are needed to use these standards on the instruments. WinIQSIM2 will be used to do waveforms directly by entering binary data i.e. for creation of the 16QAM signal used in this paper. The software also has features to visualize baseband signals, which are useful to verify digital modulation before transmission.

Vector Signal Explorer VSE

Similar as WinIQSIM2 is based on the R&S signal generator firmware, there is the Vector Signal Explorer Software (VSE) software [5] which is based on the R&S spectrum analyzer series firmware. Users who are familiar with R&S spectrum analyzer operation, do easy with the VSE software. The VSE software can also be installed and run on the target instrument, e.g on a CMA180. In this case, the software can be remote controlled via LAN by using the local host address of CMA180. The VSE software for digital modulated signal evaluation is also suitable for users who want to avoid programming of signal evaluation and graphic display. The VSE provides predefined signal evaluation procedures where the results are presented numerically or graphically. The VSE software with all its CMA180 related features is tailored to the options CMA180-K300, /-K305 and /-K310. Fig. 2-1 shows the relationship between the various software options. The base software CMA-K300 is always needed. On top the digital evaluation software K305 and/or the analog evaluation software can be used.

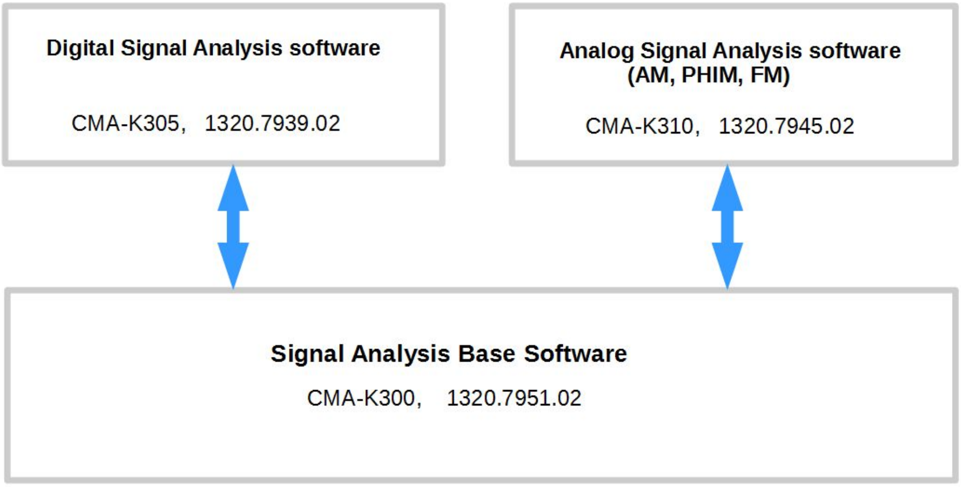
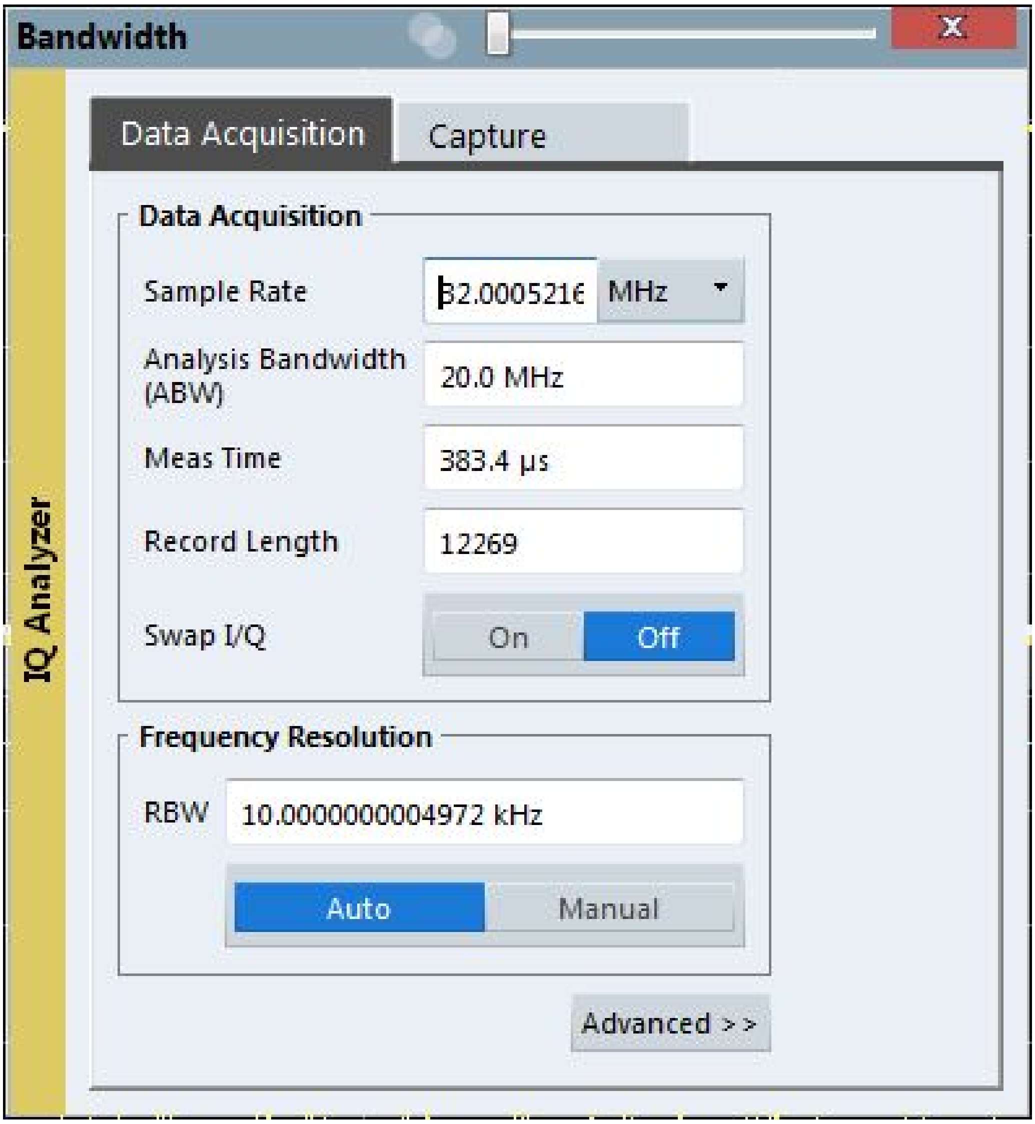

Digital Signal Analysis software
CMA-K305, 1320.7939.02
Analog Signal Analysis software
(AM, PHIM, FM)
CMA-K310, 1320.7945.02
Signal Analysis Base Software
CMA-K300, 1320.7951.02

Figure 2-1: VSE derivatives tailored to CMA180

The VSE software can also be used to remote control the CMA180. In this case the features according to Fig. 2-1 are available on a remote controlled basis. The powerful IQ-Analyzer functions of VSE are available along with the basic software K300. Analog

Demodulation features of VSE can be used along with the K310 option. Vector Signal Analysis functions of VSE are available with the K305 option. For these applications the RF COM connector of CMA180 is used as input and the IQ-data is aquired from the standard IQ demodulator of CMA, i.e. the R&S®CMA-K220 I/Q recorder option is not needed in this case. Fig. 2-2 shows the CMA180 data aquisition settings done in VSE. All data entries, e.g. the bandwidth, are checked against the possible values of CMA180.



Data Acquisition

Sample Rate: 32.0005216 MHz

Analysis Bandwidth (ABW): 20.0 MHz

Meas Time: 383.4 µs

Record Length: 12269

Swap I/Q: Off

Frequency Resolution

RBW: 10.0000000004972 kHz

Auto

Figure 2-2: CMA180 data aquisition parameter settings using VSE

MATLAB

MATLAB is a software development platform commonly used in R&D. The MATLAB script language, stored in so-called m-files, which are suitable in the early phase of development projects to explore the behavior of a new system. The Mathworks MATLAB script language can be run from that environment, if stored as "*.m"-files, or can be exported into VHDL C++ code or executables to cover other requirements. In this application note MATLAB use is shown in examples of *.m format to calculate waveforms for the CMA180 arbitrary waveform generator, and to evaluate and visualize data extracted from the CMA180 IQ recorder.

Summary of software tools

Figure 2-3 shows the software tools available for generation and evaluation of digital signals on the CMA180. The figure can be subdivided horizontally and vertically in two basic sections each. The left part shows the tools available for of IQ baseband signal generation. On the right, there are tools for evaluation of recorded I/Q data. In the upper part is the MATLAB path, which is always applicable for generation and for evaluation. In the lower half are the R&S solutions, WinIQSIM2 for generation and VSE for evaluation. The R&S solution needs no programming while the MATLAB solution always requires a certain number of programming efforts.

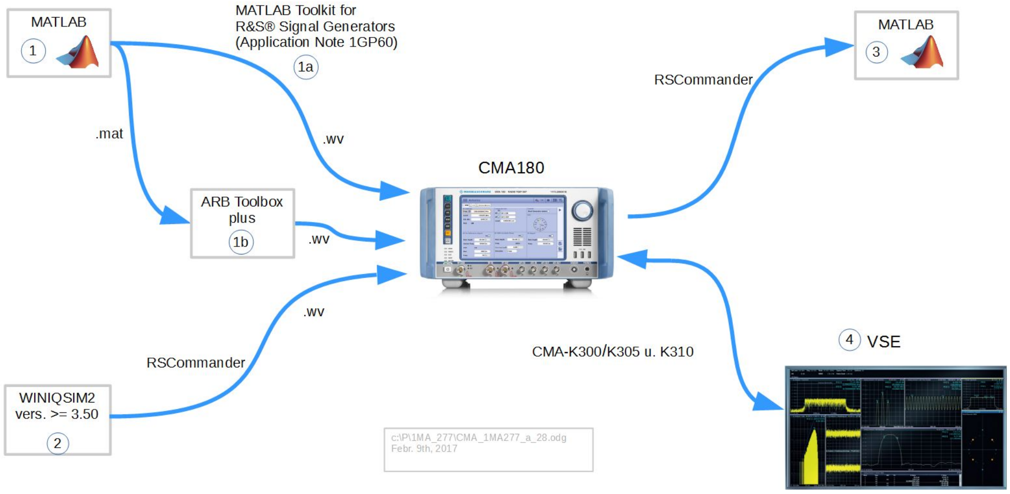


Figure 2-3: Summary of software tools for digital modulation with CMA180

3 Applications for CMA180 Digital Baseband

Chapter overview

This chapter provides some practical examples according to Fig. 3-1. The Tomahawk example, indicated by green lines, is intended as a simple means to test the workflow of creating and evaluating digital signals along with CMA180. The Tomahawk example using WinIQSIM2 is described in chapter 3.1.1 as shown in the lower left of Fig. 3-1. The appropriate evaluation is described in chapter 3.2.1. as indicated in the upper right.

A more complex example is given by the "chirped pulse" signal creation and evaluation as indicated by the blue lines in Fig. 3-1. The chirped pulse signal is created by MATLAB. The evaluation is demonstrated in chapter 3.2.3 using VSE also as indicated on the right.

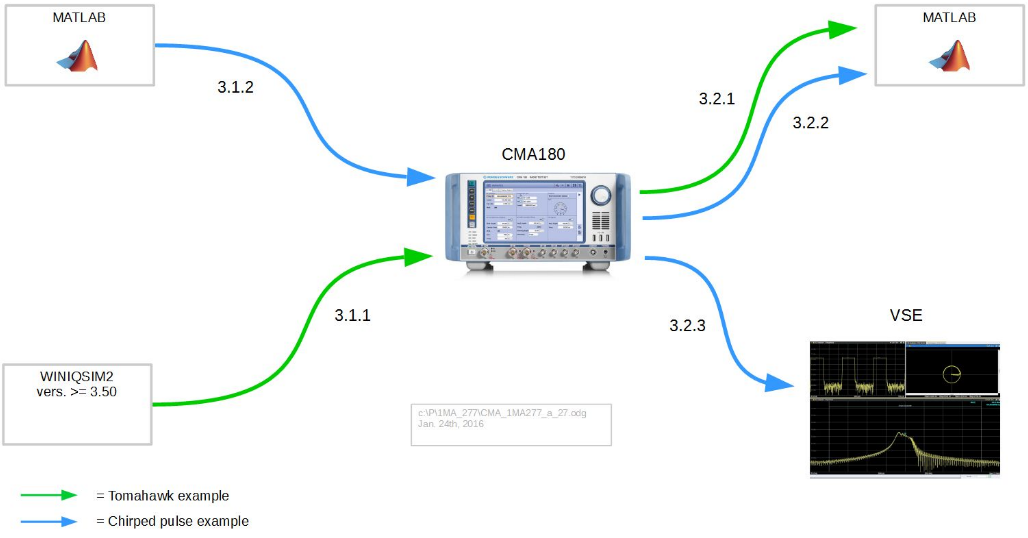


Figure 3-1: Chapter overview

3.1 Creating digital signals

This chapter provides two examples how to create digital signals along with CMA180. In the first example, the digital data is entered by hand. The second example shows, how to create a more complex signal using MATLAB.

3.1.1 WinIQSIM2 using the tomahawk shape

The "Tomahawk" example has been created for an easy start with the CMA180 along with digital modulation. Like "Hello world" for programmers, this application simply tests the development environment. The appropriate shape as shown in Fig. 3-1 has been defined as a simple bit sequence of a QAM16 signal. The appropriate hex-Code sequence "0F4C" can be easily manually entered into the pattern entry field of WinIQSIM2 [4]. Fig. 3-2 shows the IQ constellation diagram of the symbol for 16QAM modulation. The tomahawk shape and the hex-code-sequence can be verified with this diagram.

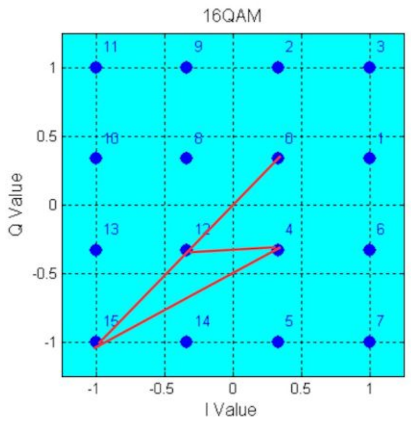


Figure 3-2: The tomahawk symbol and its hex-code

It first has to be entered in the WinIQSIM2-software. Fig. 3-3 shows the appropriate workflow steps 1 and 2a as described in this chapter. The dimmed steps 2b and 3 are described in chapter 3.2.1.

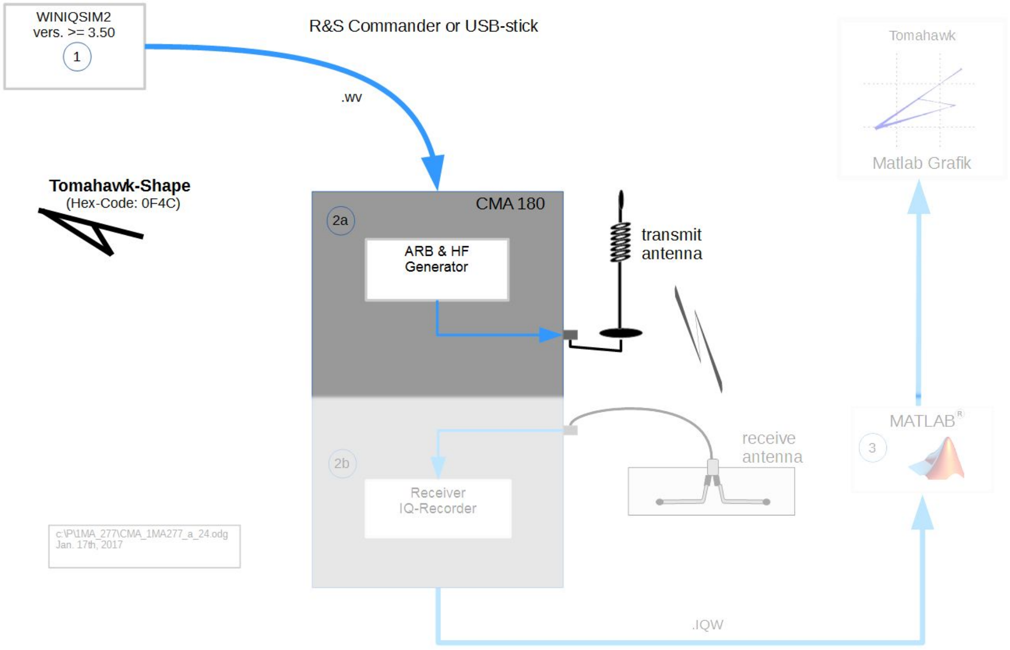


Figure 3-3: Transmit workflow of the "Tomahawk" application

(1) creating the wave file using WinIQSIM2

Now a wave file is created with WinIQSIM2 [4]. When the WinIQSIM2 software is started, the baseband settings have to be done by pressing the baseband button as shown in Fig. 3-4.

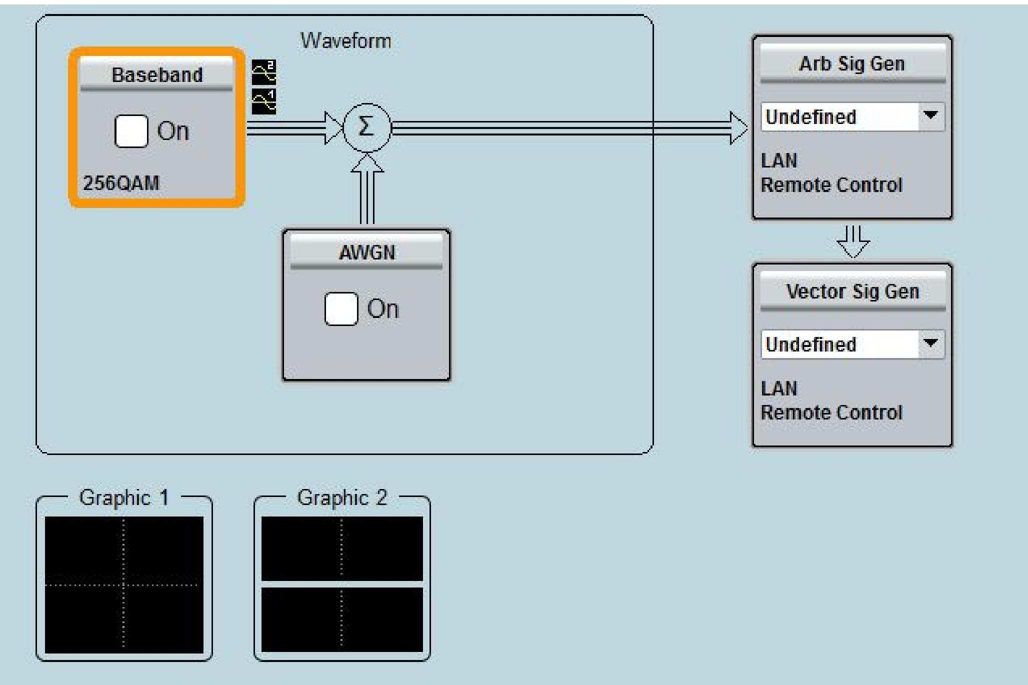
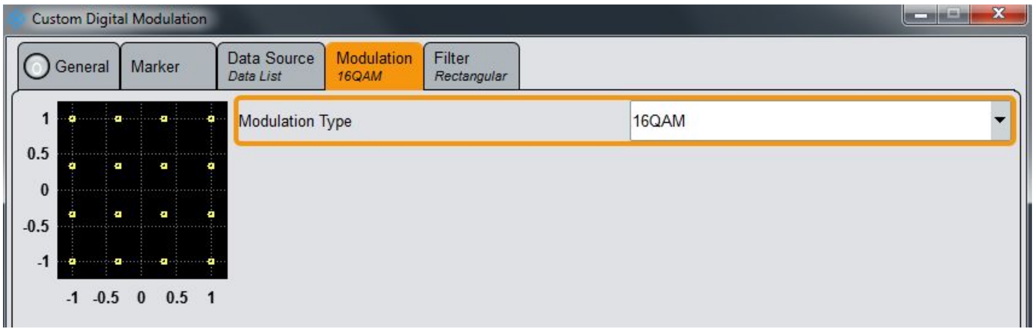


Figure 3-4: WinIQSIM2 starting screen

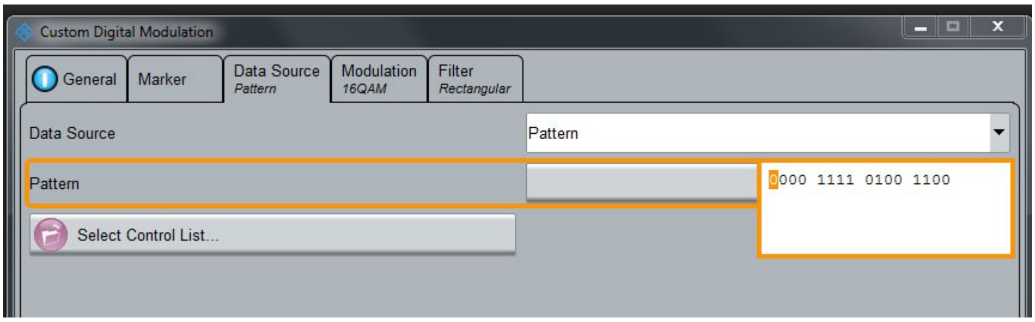
A selection list now appears where 'Custom Digital Modulation' form the 'Misc' group has to be chosen. Then the modulation hast to be set to QAM16 as shown in Fig. 3-5.



Modulation Type: 16QAM

Figure 3-5: Setting the Modulation Type

Now the tab 'Data Source' has to be selected and the pattern of the tomahawk symbol is entered as shown in Fig. 3-6.

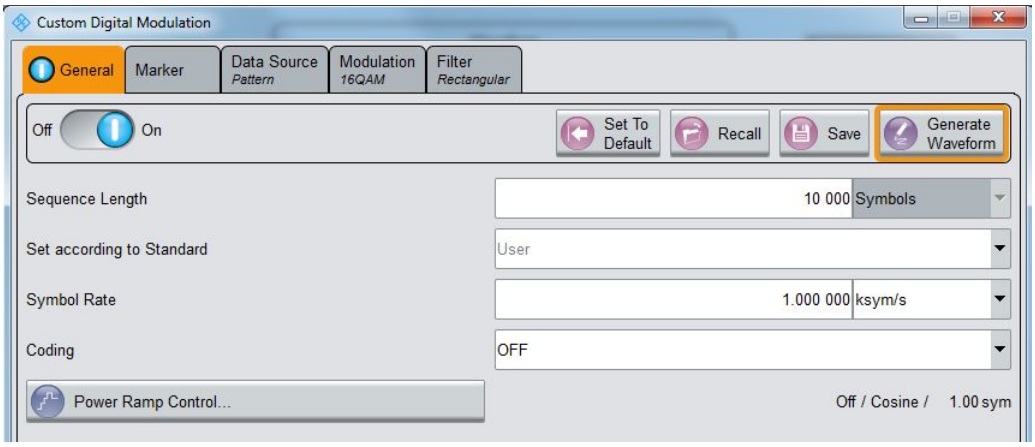


Data Source: Pattern

Pattern: 0000 1111 0100 1100

Figure 3-6: WinIQSIM data entry according to the "Tomahawk" shape (0x0F4C)

Finally some basic settings are made according to Fig. 3-7. When the blue Off/On button in the upper left is set to 'On' the wave can be created and saved on the harddisk of the PC by pressing the 'Generate Waveform' button in the upper right.



Sequence Length: 10 000 Symbols

Set according to Standard: User

Symbol Rate: 1.000 000 ksym/s

Coding: OFF

Off / Cosine / 1.00 sym

Figure 3-7: Setting basic parameters

When closing the 'Custom Digital Modulation' window the main window of WinIQSIM2 finally shows the tomahawk shape as shown in Fig. 3-8.

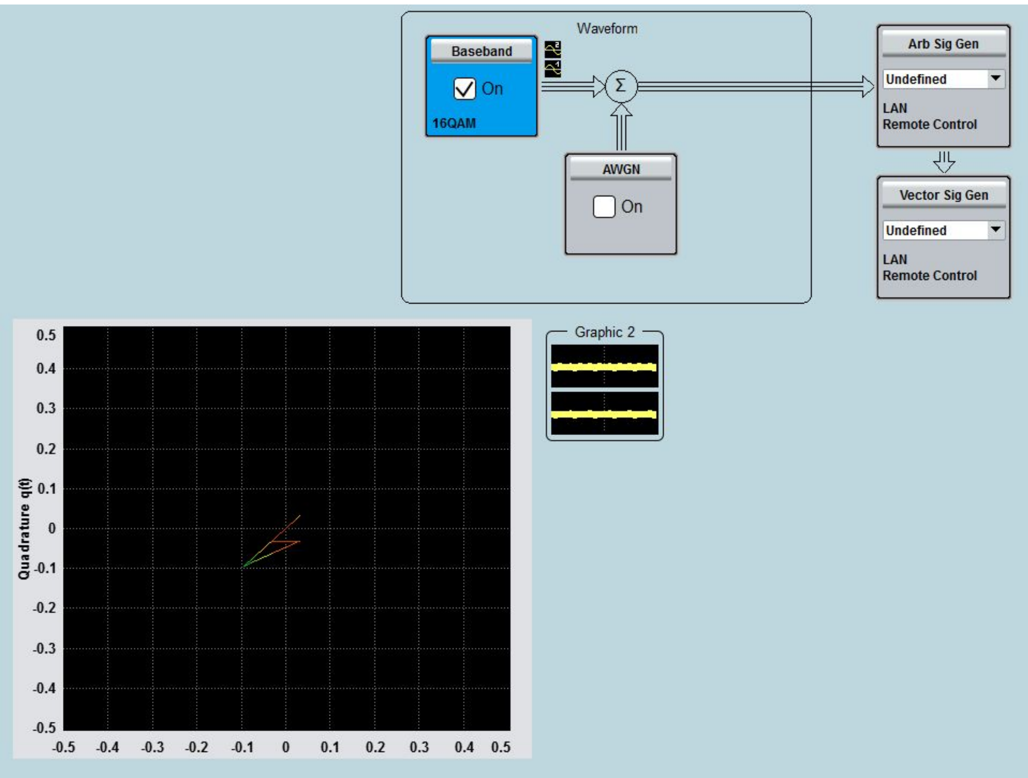
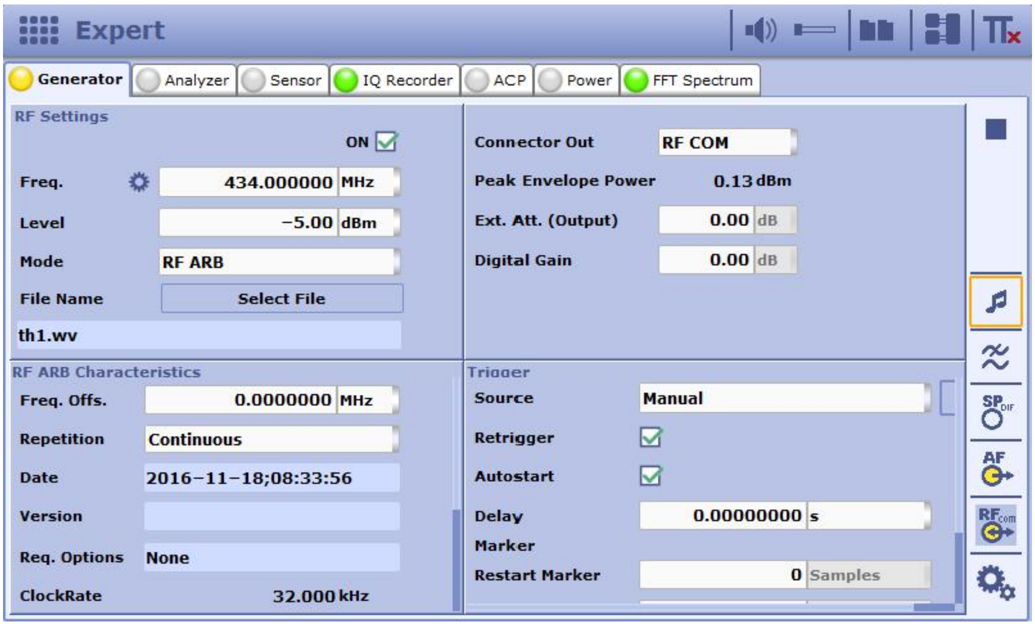


Figure 3-8: Verification of the manually entered vector data

(2a) preparing the CMA180 generator

Before transmitting the signal to the air, the CMA180 generator must be prepared according to Fig. 3-9. We are working in the ISM band at 434 MHz. The transmit power must be set to -5dBm because of the path attenuation between the transmitting and receiving antenna. Setting the output power is recommended to be done by approaching from lower values, e.g. from -30dBm, to avoid over-ranging the receiver later on.



RF Settings

ON ☑

Freq. 434.000000 MHz

Level −5.00 dBm

Mode RF ARB

File Name

th1.wv

Connector Out RF COM

Peak Envelope Power 0.13 dBm

Ext. Att. (Output) 0.00 dB

Digital Gain 0.00 dB

RF ARB Characteristics

Freq. Offs. 0.0000000 MHz

Repetition Continuous

Date 2016−11−18;08:33:56

Version

Req. Options None

ClockRate 32.000 kHz

Trigger

Source Manual

Retrigger ☑

Autostart ☑

Delay 0.00000000 s

Marker

Restart Marker 0 Samples

Figure 3-9: CMA180 generator settings for the "Tomahawk" application

When all settings are done the CMA180 transmitter can be started by pressing the triangle in the upper right. The transmit antenna is connected to the RF COM connector of CMA180. When the transmitter is started, the signal with the tomahawk shape is on the air. The signal now can be received and evaluated which is described in chapter 3.2.1.

3.1.2 MATLAB using the chirped pulse example

This example shows how to create an arbitrary waveform with MATLAB, play the output waveform file on the CMA180 arb generator, record and display the resulting I/Q data. This example is mainly addressing MATLAB users who want to create and to evaluate IQ modulated signals. Fig. 3-10 shows the setup used.

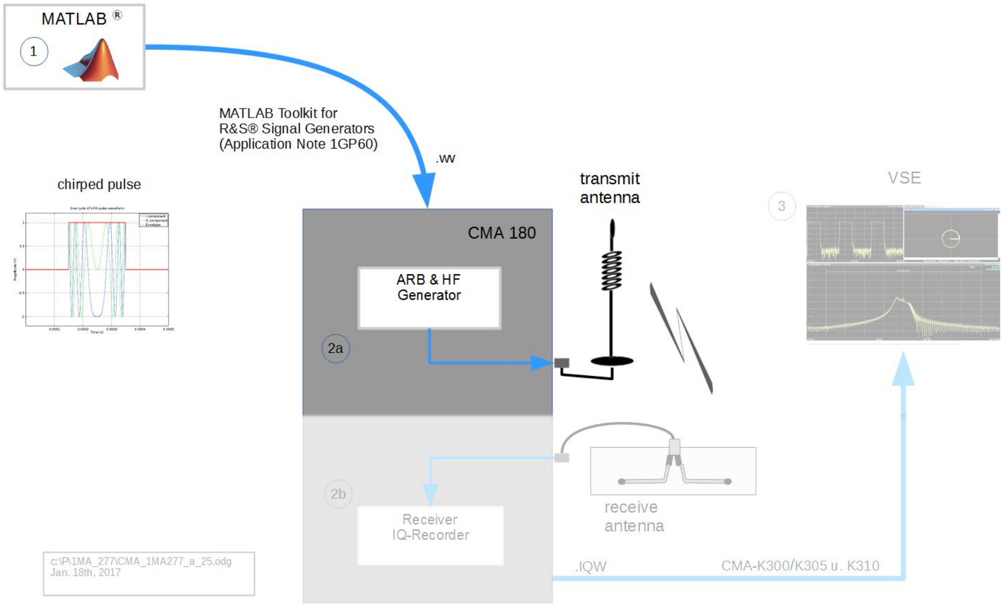


Figure 3-10: Transmit workflow of the "chirped pulse" application

The chirped pulse signal has been chosen, because the appropriate MATLAB program shows how to create waveforms in terms of amplitude and of phase and frequency variation. Other arbitrary waveforms can be easily derived from this example. When the IQ-modulated signal has been transmitted and received the resulting IQ-date can be easily displayed and compared to the original signal. Thus verifying the whole transmit-receive-channel is working properly. The MATLAB program to create the waveform is listed in Fig. 3-11 anf Fig. 3-12. The source file is attached to the application note for download.

(1) calculating the baseband signal

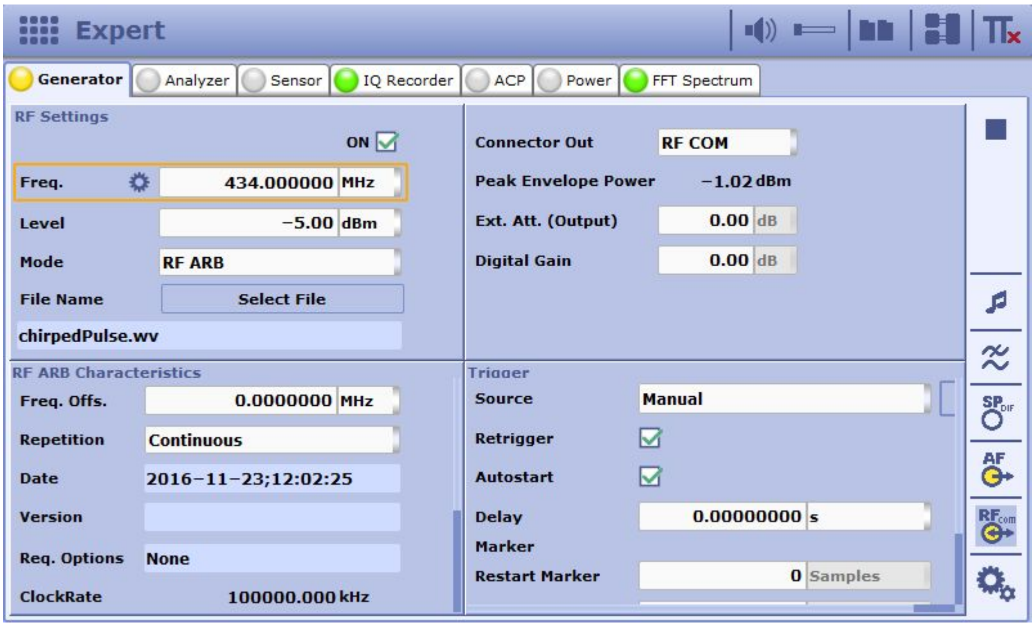
The baseband signal, e.g. the pulsed chirp, is calculated with the short MATLAB program listed in Appendix 4.1.1. Some basic settings are made in the lines 6 ... 11. The parameters can be changed in the program. In lines 13 to 16 the complex baseband signals are calculated. The chirp-function is used from the MATLAB Signal Processing Toolbox, which is needed for running in the MATLAB environment.

The result file is stored locally on the PC, refer to line 38, additionally it is converted from .mat to the .wv file format, which is finally transferred to CMA180.

Using the R&S MATLAB Toolkit [9] the calulated waveform will be transferred into a wv file and directly downloaded into the CMA180. The appropriate m-code is listed in Appendix 4.1.2.

(2a) prepare the CMA180 for transmission

Now configure the CMA180 generator according to Fig. 3-11. The operating frequency is the 434 MHz ISM band to transmit through the air. A relative high output power of -5 dBm is necessary due to the path attenuation from the transmit- to the receive-antenna. Next specify the *.wv file created above, here "chirpedPulse.wv". The transmit antenna is connected to the RF COM output. The generator has been already started, which indicated by the stop symbol at the top right and the illuminated LED beside the "Generator" button.



RF Settings

ON ☑

Freq. 434.000000 MHz

Level −5.00 dBm

Mode RF ARB

File Name

chirpedPulse.wv

Connector Out RF COM

Peak Envelope Power −1.02 dBm

Ext. Att. (Output) 0.00 dB

Digital Gain 0.00 dB

RF ARB Characteristics

Freq. Offs. 0.0000000 MHz

Repetition Continuous

Date 2016−11−23;12:02:25

Version

Req. Options None

ClockRate 100000.000 kHz

Trigger

Source Manual

Retrigger ☑

Autostart ☑

Delay 0.00000000 s

Marker

Restart Marker 0 Samples

Figure 3-11: CMA180 generator settings for the "Chirped Pulse" application

3.2 Evaluating digital signals

3.2.1 Workflow steps for evaluation the example signal

Fig. 3-12 shows the receive workflow of this special test, where the dimmed transmit part was described in chapter 3.1.1.The receiving antenna is connected to the RF IN port. The I/Q data is received over the air (OTA) and recorded with the CMA180 IQ-recorder. The resulting .IQW file is then transferred to MATLAB where the final evaluation takes place.

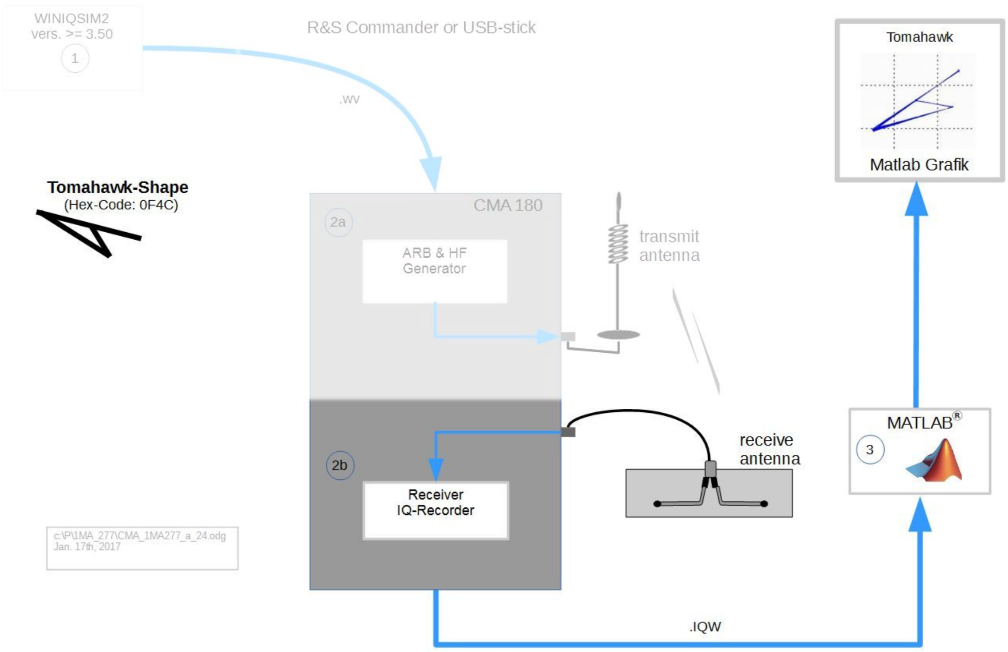
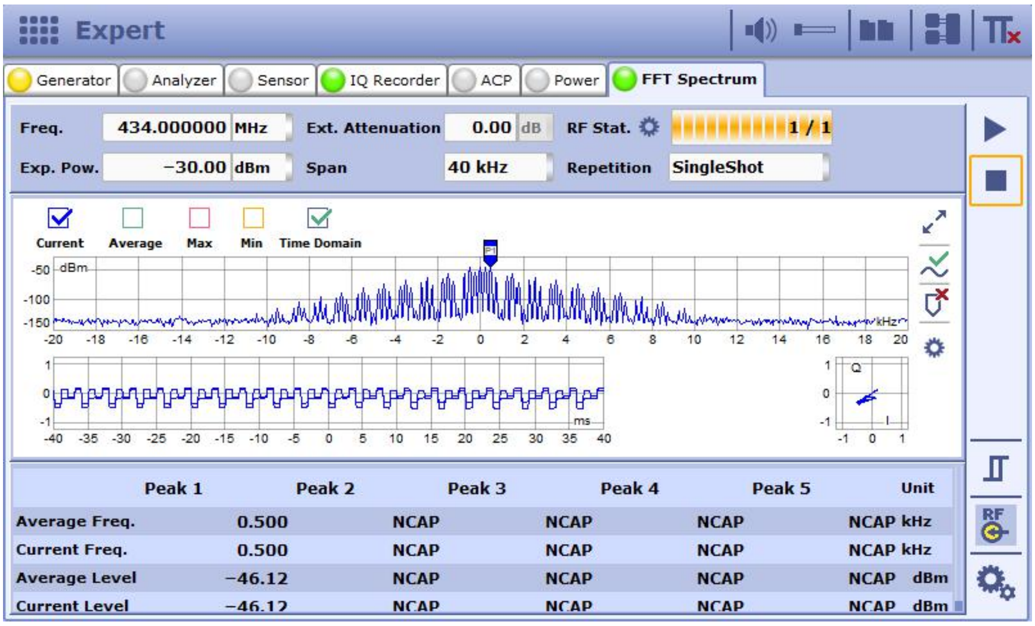


Figure 3-12: Receive workflow of the "Tomahawk" application

(2b) preparing the CMA180 receiver

The CMA180 receiver must be prepared to decode and record the I/Q-data. Receiver setup is performed according to Fig. 3-13



Freq. 434.000000 MHz

Exp. Pow. −30.00 dBm

Ext. Attenuation 0.00 dB

Span 40 kHz

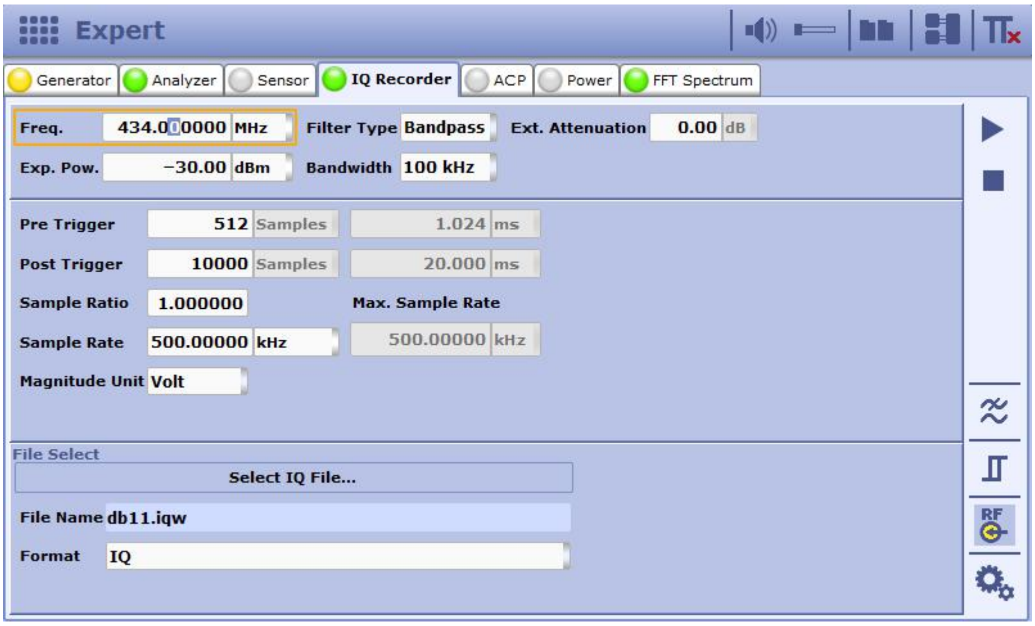
RF Stat. 1 / 1

Repetition SingleShot

	Peak 1	Peak 2	Peak 3	Peak 4	Peak 5	Unit
Average Freq.	0.500	NCAP	NCAP	NCAP	NCAP	kHz
Current Freq.	0.500	NCAP	NCAP	NCAP	NCAP	kHz
Average Level	−46.12	NCAP	NCAP	NCAP	NCAP	dBm
Current Level	−46.12	NCAP	NCAP	NCAP	NCAP	dBm

Figure 3-13: CMA180 "FFT spectrum" display of the received "Tomahawk" symbol

The frequency is identical to the transmit frequency of 434 MHz, ISM-band. With -30 dBm, the "expected power" is at its minimum. Power setting is recommended to be approached from higher values, e.g. 0 dBm to avoid over-range. When the setup is completed, start the receiver with the handles in the upper right area. Use the "FFT Spectrum" measurement to verify the received signal as shown in Fig. 3-13. The large diagram at the top shows spectrum vs. span as defined before. In the lower left, the receiver I/Q-data vs. time is displayed. In the lower right is the I/Q-constellation diagram which displays the received tomahawk shape. The IQ-recorder is used from now to prepare a detailed analysis by MATLAB. The IQ-recorder is setup according to Fig. 3-14. Pressing the "gear wheel button" opens another setup window with additional settings not shown. The trigger conditions can be set here. In the beginning, the "Free-Run" trigger is used to receive a signal at all. Later on the settings can be enhanced by using the IQ-power- or external trigger, where the trigger is derived from the marker function as always available in the R&S wv-format. The concept of this application would be spoiled a bit by an external trigger since we want to work wireless from source to sink.



Freq. 434.000000 MHz
Filter Type Bandpass
Ext. Attenuation 0.00 dB
Exp. Pow. −30.00 dBm
Bandwidth 100 kHz
Pre Trigger 512 Samples 1.024 ms
Post Trigger 10000 Samples 20.000 ms
Sample Ratio 1.000000
Max. Sample Rate
Sample Rate 500.00000 kHz 500.00000 kHz
Magnitude Unit Volt
File Select
Select IQ File...
File Name db11.iqw
Format IQ

Figure 3-14: CMA180 settings of IQ recorder

When the IQ-recorder is finished, press the "run-button" in the upper right. The IQW file is created automatically after the recording and can now be transferred to a PC for postprocessing with MATLAB. The IQW file is transferred from CMA180 to the PC via R&S Commander. An USB stick can also be used for file transfer.

(3) postprocessing with MATLAB

The four diagrams resulting from the MATLAB program are shown in Fig. 3-15. In the upper left, we can see the tomahawk symbol again within the IQ-constellation diagram. In the upper right, we find the time domain plot of the appropriate I/Q data. In the lower left is the time domain of the received pulse power and finally in the lower right we find the normalized pulse spectrum. The MATLAB program is listed with syntax coloring in appendix 1. The complete source code is also available in the download area of this application note.

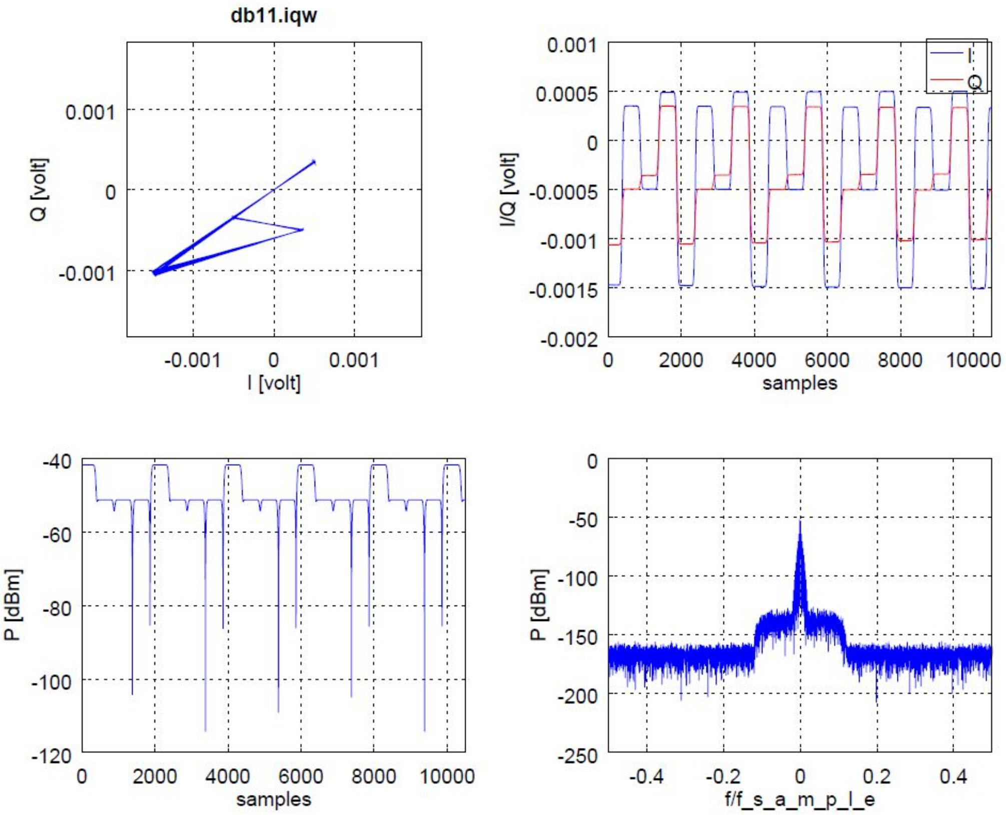


Figure 3-15: MATLAB evaluation of the IQ recorder data output

Summary of the "Tomahawk" Example

The tomahawk example introduced a complete transmission and reception workflow of an IQ-modulated signal using a CMA180. For the sake of simplicity, the data was entered manually. This application can also be used as a base for further enhancements with more complex IQ-sequences and higher sophisticated postprocessing programs on the receiving side. Even with higher complex IQ sequences, the workflow always remains the same as described here.

3.2.2 Evaluating the Chirped Pulse using MATLAB

The MATLAB program for visualizing the recorded data is the same used in the tomahawk example. It is listed in appendix 1 and is available in download area of this application note. The graphics output of the program is shown in Fig. 3-16. In the upper left is the IQ-constellation diagram, which is a circle, because the sweep function of MATLAB works with sinusoidal signals swept along the chirp frequencies. The lines to the center of the diagram are stemming from the pulses detected within the sampling time. In the upper right is the time domain diagram of the sequence of chirped pulses. With a sample rate of 10 MHz and time interval of 0.1 usec you can easily verify the pulse width and pulse repitition rate : 200 * 0.1 usec --> 20 usec for the first, and 500 * 0.1

usec --> 50 usec for the latter, equating the values specified in the MATLAB program,Appendix 4.1.1, line 7 and 8.

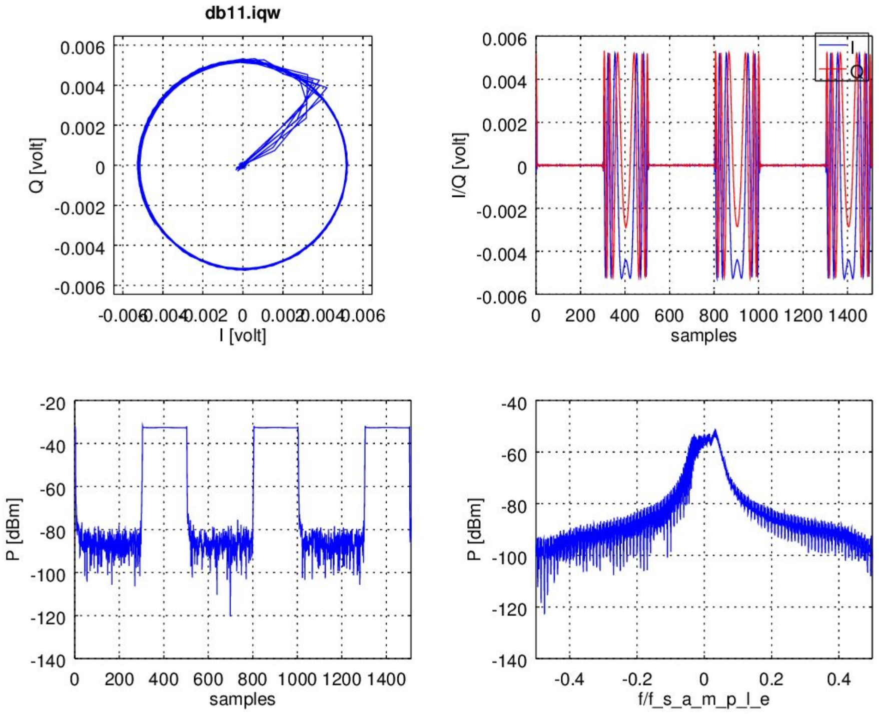


Figure 3-16: MATLAB evaluation of "Chirped Pulse"

3.2.3 Evaluating the Chirped Pulse using VSE

The VSE [5] provides full featured graphical evaluation without the need to write any software. Fig. 3-17 shows the appropriate block diagram.

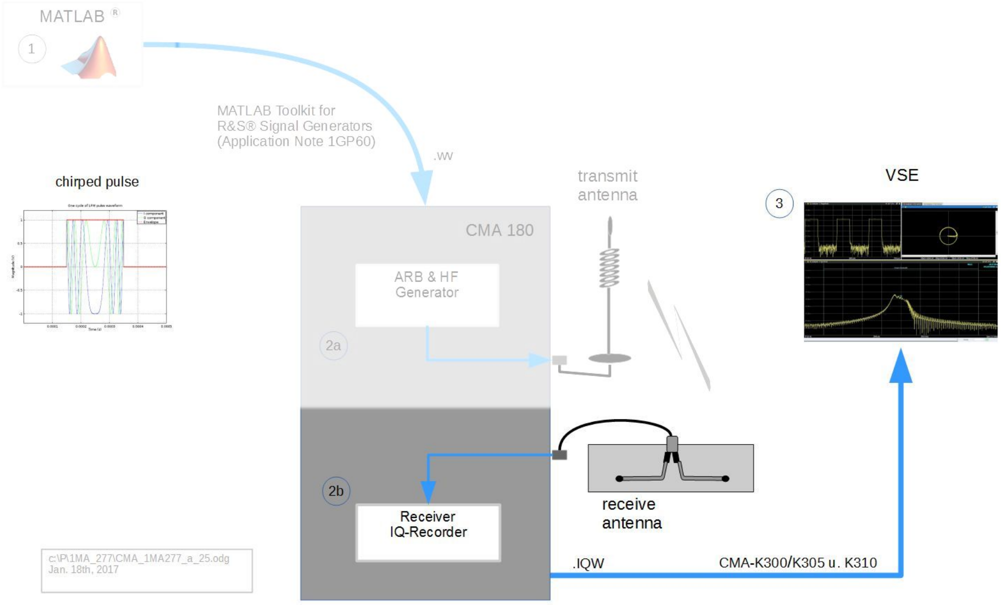


Figure 3-17: Workflow of "Chirped Pulse" application using VSE for postprocessing

Workflow steps 1 and 2 are the same as described in the last chapter. In the postprocessing step 3, VSE is used for postprocessing the recoreded data. The IQW result file from the CMA IQ recorder can be directly imported via the VSE menu item "Recall I/Q Recording" as shown in Fig. 3-18.

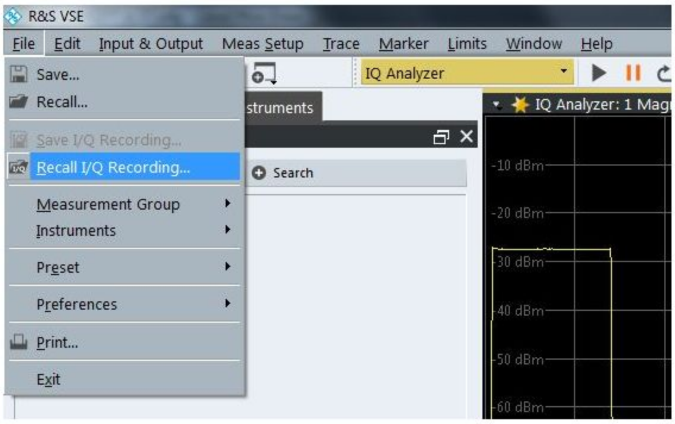


Figure 3-18: Importing I/Q-data into VSE

3.3 Recording and Evaluating OOK Signals

Remote garage door keys are based on On-Off-Key (OOK) modulation. It is a low-cost method especially for user groups where safety requirements are not so strict. The current example uses a remote key for opening a garage. The key signal is recorded with the CMA180 IQ recorder in a laboratory away from the garage door. The I/Q data representing the modulation signal is stored on the internal hard disk of the CMA180. A live test with the garage door can be performed with a CMA180 equipped with the battery option CMA-B060A. Even though the CMA180 is a bit too expensive to be permanently used as garage door opener, this example demonstrates in a comprehensive way how to "record and play" radio signals. This method can be applied to any other radio signal as long as the requirements regarding bandwidth, bit resolution and memory depth are met. It thus suggests various additional applications based upon the herein described method. Fig. 3-19 shows the workflow for recording and evaluation of a remote key signal.

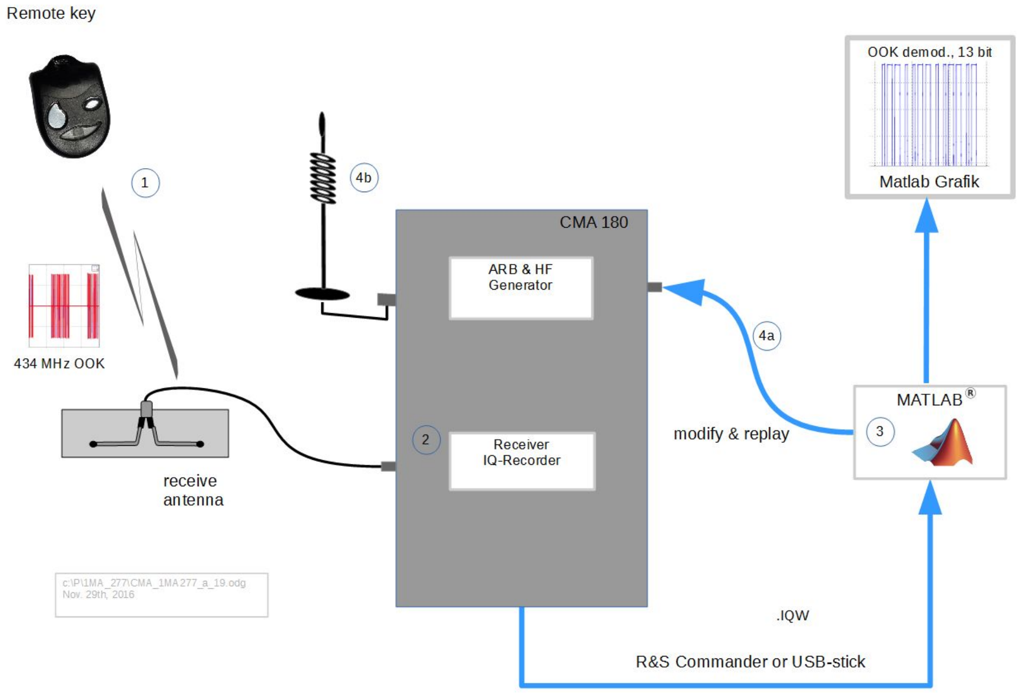


Figure 3-19: Workflow of IQ recording of an OOK modulated garage key

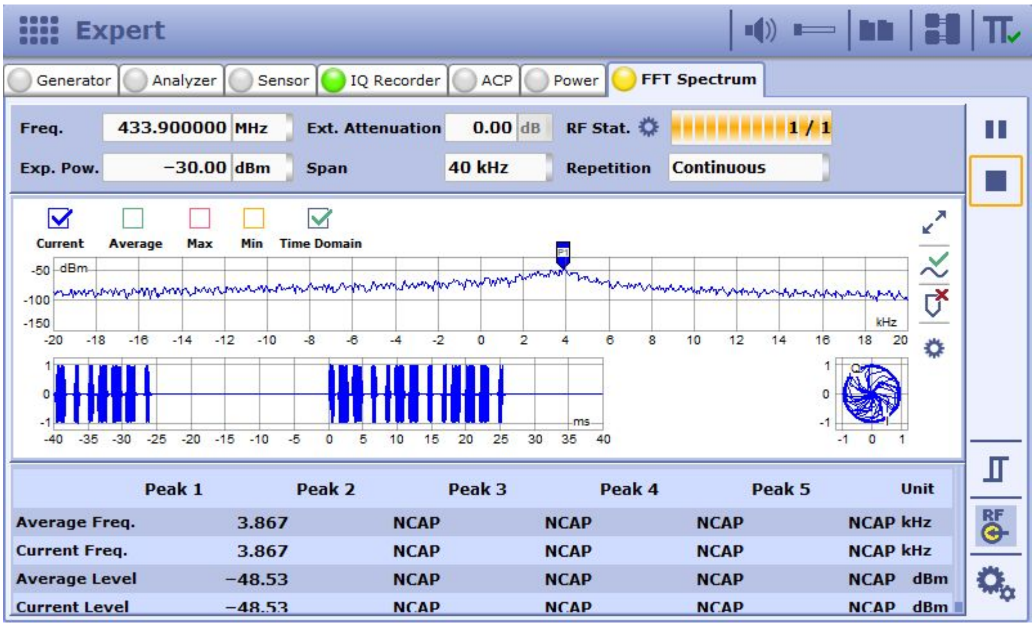
3.3.1 Workflow Steps

(1) transmitting the key code

In this example, we use a standard remote key working at approximately 434 MHz with 13-bit OOK modulation. The distance to the receive antenna is at least 1 meter. The remote key has two buttons. Both buttons initiate the same RF signal but with a different bit sequence. The very start is done with "Free run" trigger in the CMA180. When the basic RF parameters such as frequency and level / attenuation are set, change to "IF Power" trigger. In this case, start the CMA180 which then signals its ready state with a yellow LED. The CMA can be triggered by pressing one of the two remote key buttons. After recording the LED changes from yellow to green. The LED change appears to be immediately after pressing the remote key button.

(2) receiving and recording

Fig. 3-20 shows a CMA180 screenshot in the "FFT Spectrum" mode. It is useful to perform the "FFT Spectrum" measurement before the IQ-recording is done. Power and demodulation parameter setup can be verified and optimized this way. The 13 bits from the OOK signal can be roughly recognized. Now change to the "IQ Recorder Mode" by pressing the appropriate tab.

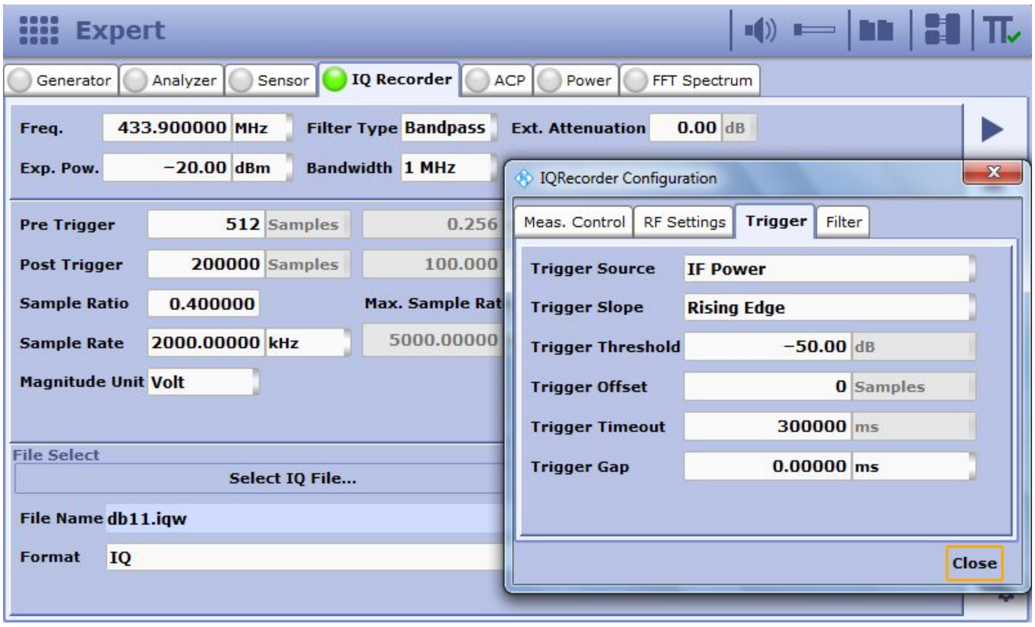


Freq. 433.900000 MHz
Ext. Attenuation 0.00 dB
RF Stat. 1 / 1
Exp. Pow. −30.00 dBm
Span 40 kHz
Repetition Continuous

	Peak 1	Peak 2	Peak 3	Peak 4	Peak 5	Unit
Average Freq.	3.867	NCAP	NCAP	NCAP	NCAP	kHz
Current Freq.	3.867	NCAP	NCAP	NCAP	NCAP	kHz
Average Level	−48.53	NCAP	NCAP	NCAP	NCAP	dBm
Current Level	−48.53	NCAP	NCAP	NCAP	NCAP	dBm

Figure 3-20: CMA180 "FFT spectrum" display of the received OOK signal

Fig. 3-21 shows the IQ recorder along with the trigger setup window which can be opened by pressing the gear wheel in the lower right of the CMA180 display. The recorder is started by pressing the "Run" symbol in the top right. The LED in the recorder tab then changes to yellow until the recorder is triggered. This step is done by pressing the remote key button located close to the receiving antenna. When the IQ-recorder LED changes to green again, recording is done and the MATLAB postprocessing can take place.



Freq. 433.900000 MHz

Filter Type Bandpass

Ext. Attenuation 0.00 dB

Exp. Pow. −20.00 dBm

Bandwidth 1 MHz

Pre Trigger 512 Samples 0.256

Post Trigger 200000 Samples 100.000

Sample Ratio 0.400000

Max. Sample Rat 5000.00000

Sample Rate 2000.00000 kHz

Magnitude Unit Volt

File Select

Select IQ File...

File Name db11.iqw

Format IQ

IQRecorder Configuration

Trigger Source IF Power

Trigger Slope Rising Edge

Trigger Threshold −50.00 dB

Trigger Offset 0 Samples

Trigger Timeout 300000 ms

Trigger Gap 0.00000 ms

Figure 3-21: CMA180 IQ recorder setup for OOK evaluation

(3) postprocessing

Similar as in the examples above transfer the *.IQW result file to the PC and evaluate the result with the same software used before, refer to appendix 1 for details. The result is shown in Fig. 3-22. Again there are four diagrams, IQ-constellation diagram in the upper left, time domain plot of the appropriate I/Q data in the upper right. In the lower left is the time domain of the received pulse power in the lower the normalized pulse spectrum.

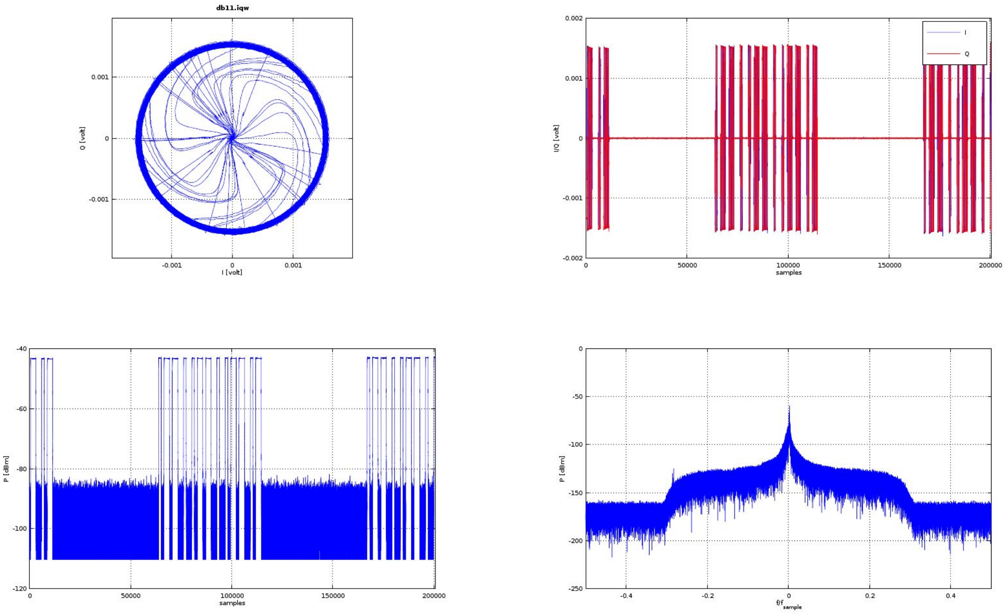


Figure 3-22: MATLAB evaluation of OOK signal recordings

(4) modify and play

With MATLAB can now we now decode the signal down to bit level. Fig. 3-23 shows the OOK pattern for the left and the right button. The pulse width represents the logical value 0 or 1. With this information and the physical parameters measured before with the CMA180, the signal can be modified and resent to the CMA180 for transmission. Following useful tests can be performed now:

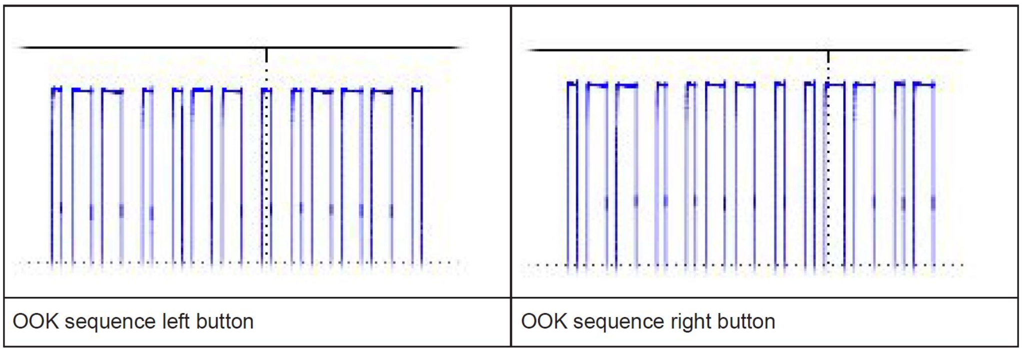
* Retransmission with a modified bit pattern

* Retransmission with variation in frequency or modulation depth

* Retransmission with various levels of noise interferers

* Determining the Bit Error Rate with MATLAB

* Evaluating the bit pattern by cross correlation and controlling external equipment, e.g. via CMA CONTROL port.



OOK sequence left button

OOK sequence right button

Figure 3-23: Bit pattern evaluation using MATLAB

3.4 Pulse Evaluation Using the Base R&S®CMA180

This example shows the evaluation of a pulsed modulated signal which can be performed even with the base CMA180, i.e. without CMA-B110 baseband-generator and without R&S®CMA-K220 IQ Recorder. The I/Q data being evaluated is not taken from the IQ-recorder but rather from CMA180 FFT Spectrum application. The software for this demo was written in the programming language Python, integrated into the FORUM development environment and is freely available from the R&S homepage [6]. Figure 3-26 shows the appropriate block diagram of this application. The test transmitter, which is an emulation of a garage remote key, shown on the left, transmits a pulse modulated ISM signal. Similar to the garage remote key the pulse sequence can be affected by two push buttons available on the test transmitter. The CMA180 receives and evaluates the signals. According to the pulse sequence, a USB controlled rotary device is turning either left or right according to the push button pressed at the test transmitter. To simplify things, the transmitted pulses are square wave shaped with two different frequencies.

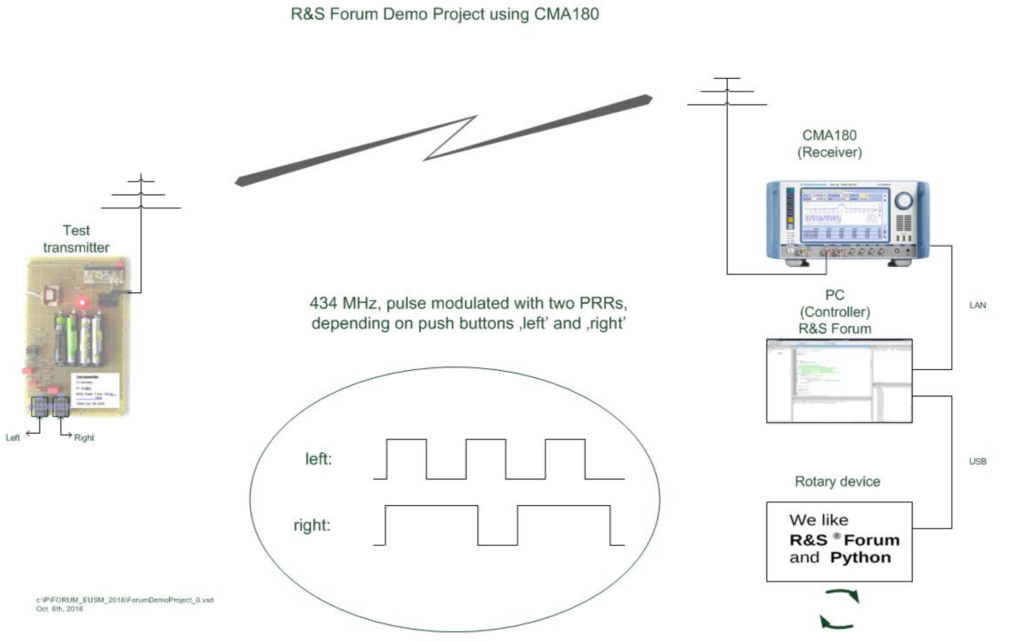
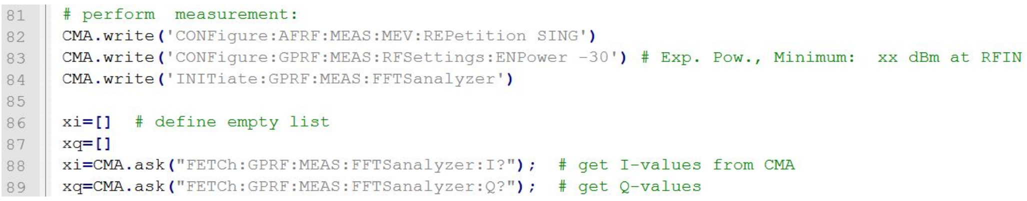

R&S Forum Demo Project using CMA180
Test transmitter
Left
Right
CMA180 (Receiver)
434 MHz, pulse modulated with two PRRs, depending on push buttons ‚left' and ‚right'
left:
right:
PC (Controller) R&S Forum
LAN
USB
Rotary device
We like R&S ® Forum and Python
c:\P\FORUM_EUSM_2016\ForumDemoProject_0.vsd
Oct. 6th, 2016

Figure 3-24: CMA180 direct IQ evaluation

The entire Python program is provided along with the application note in the R&S homepage [6] . The Python program excerpt in Fig. 3-25 shows the CMA180 programming. Lines 82 and 83 include some basic setups. The trigger was set manually on "Free Run" before. The code in line 84 finally initiates the measurement. With the commands in lines 88 and 89, the IQ data is read and stored in Python lists, which then are evaluated regarding the two pulse frequencies.



```
81  # perform  measurement:
82  CMA.write('CONFigure:AFRF:MEAS:MEV:REPetition SING')
83  CMA.write('CONFigure:GPRF:MEAS:RFSettings:ENPower -30') # Exp. Pow., Minimum:  xx dBm at RFIN
84  CMA.write('INITiate:GPRF:MEAS:FFTSanalyzer')
85
86  xi=[]  # define empty list
87  xq=[]
88  xi=CMA.ask("FETCh:GPRF:MEAS:FFTSanalyzer:I?");  # get I-values from CMA
89  xq=CMA.ask("FETCh:GPRF:MEAS:FFTSanalyzer:Q?");  # get Q-values
```

Figure 3-25: Forum/Python program excerpt for CMA180 direct IQ evaluation

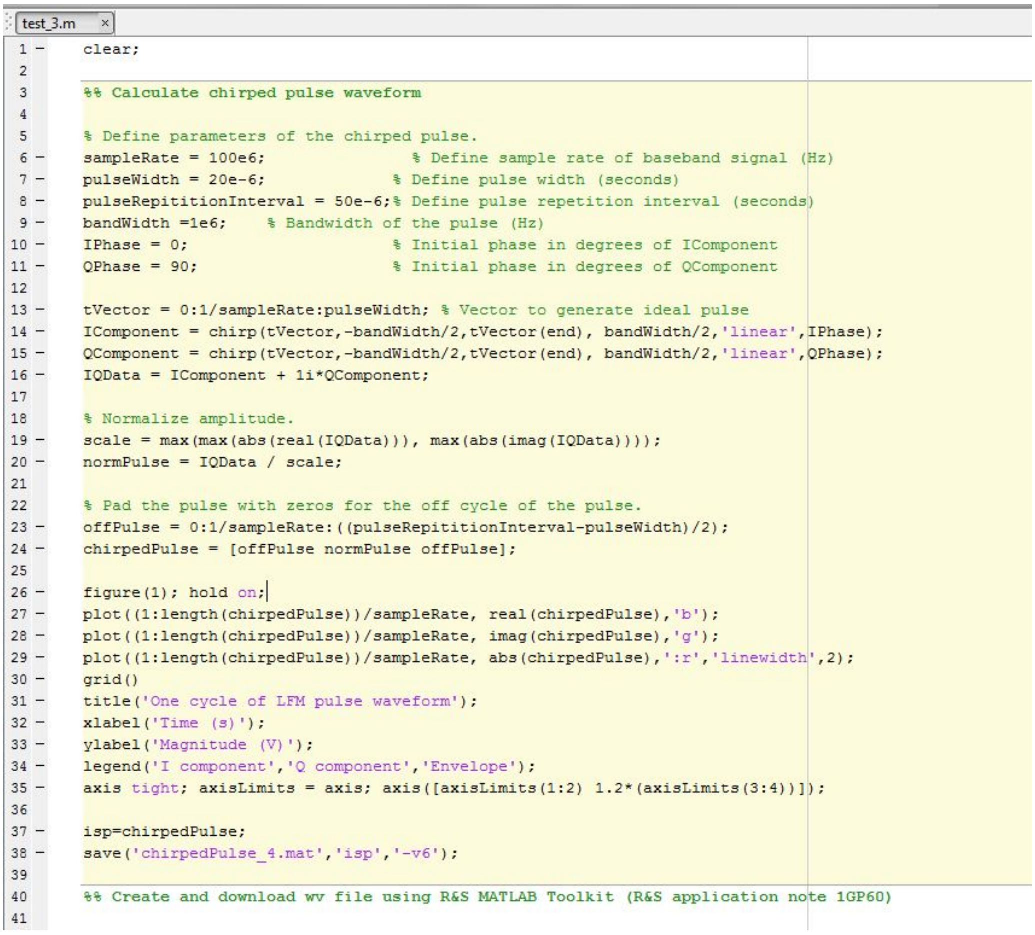
This example shows I/Q data evaluation with the CMA180 in its simplest form. It can be used to evaluate the CMA180 IQ demodulation features before the IQ-recorder option is purchased.

4 Appendix

4.1 MATLAB Program Listings "Create and download chirped pulse waveform"

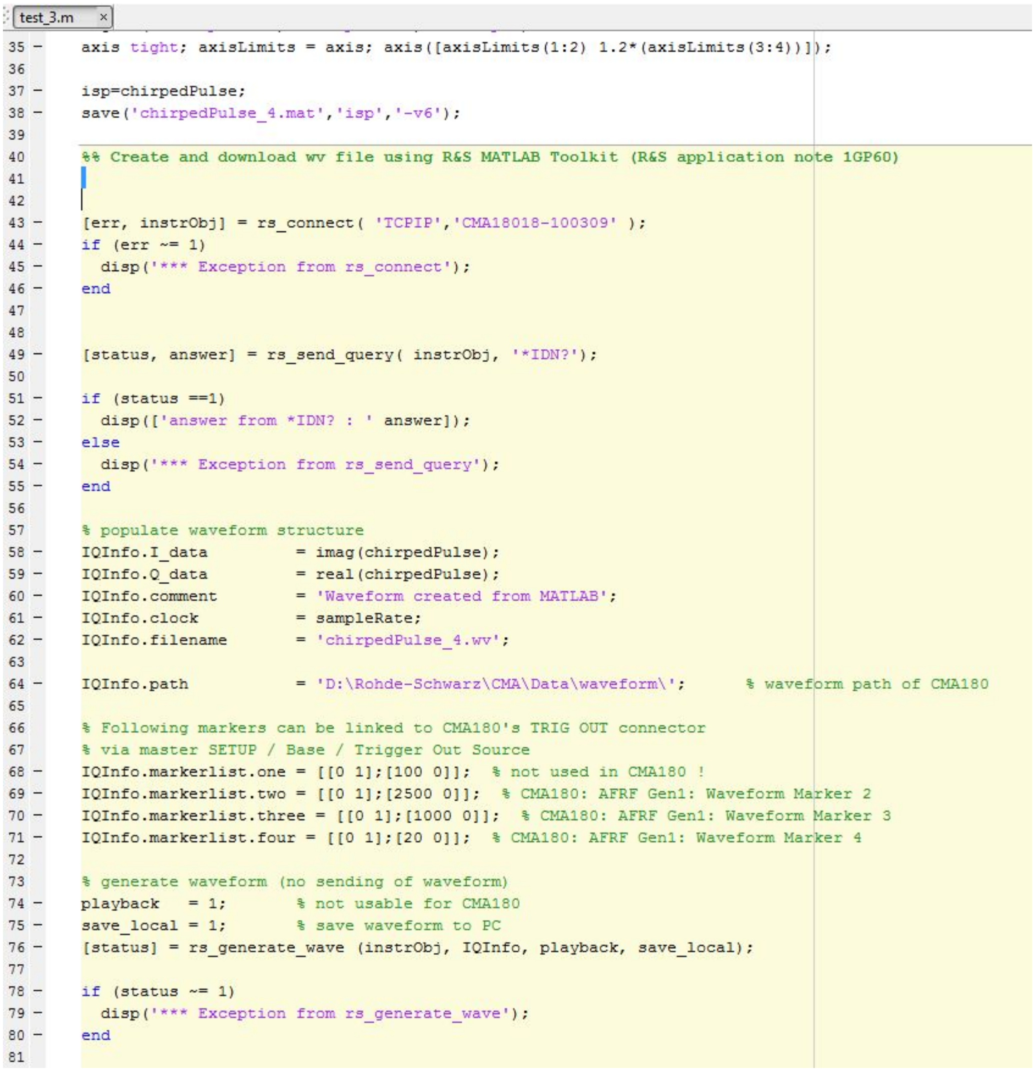
The MATLAB program to create and to download the chirped pulse waveform to CMA180 is listed in this appendix. For the sake of clarity line numbers and syntax coloring are given. To use the program directly in MATLAB it is also available in the download area of the application note.

4.1.1 Calculate chirped pulse waveform



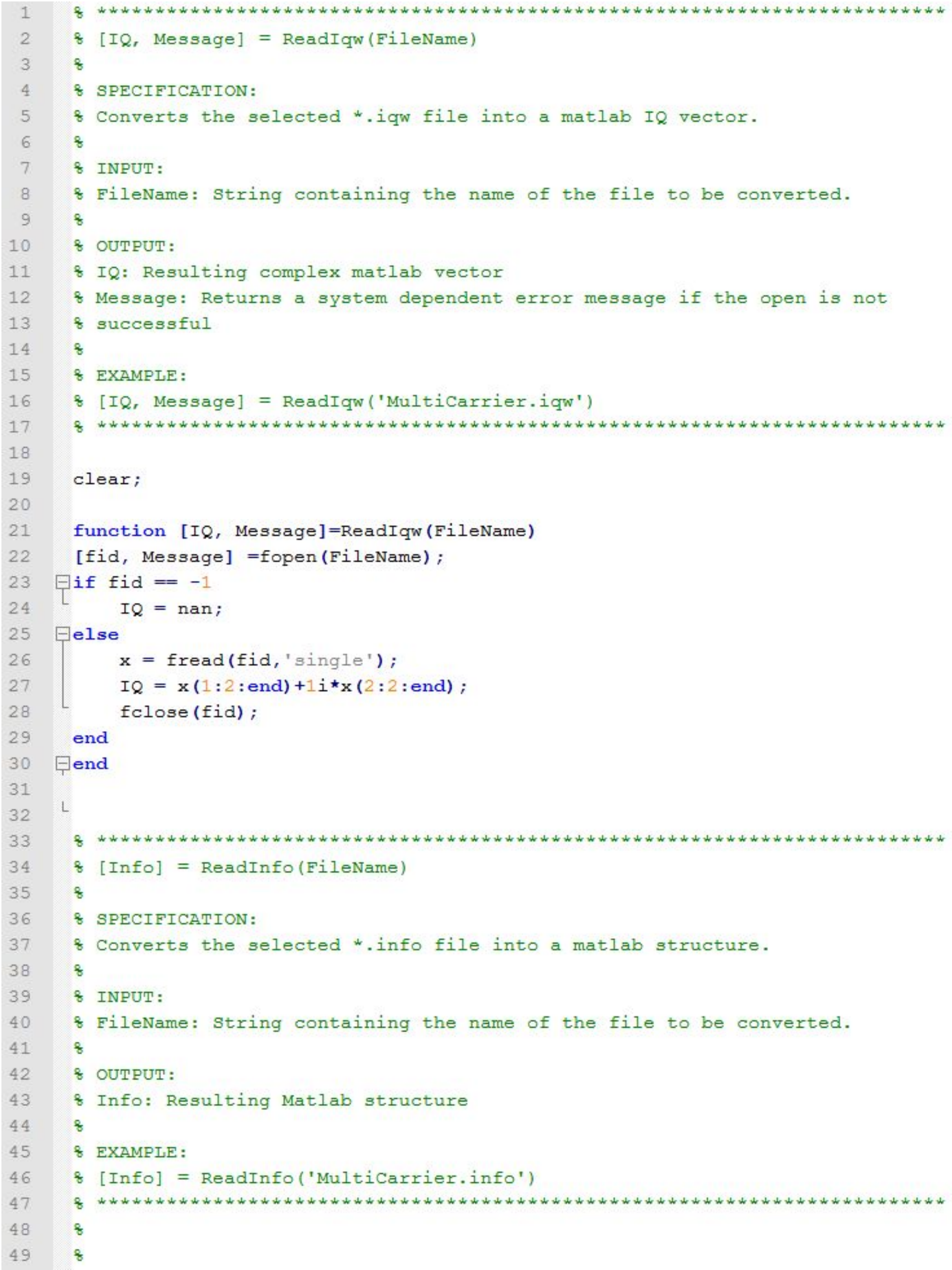
```
clear;

%% Calculate chirped pulse waveform

% Define parameters of the chirped pulse.
sampleRate = 100e6;                % Define sample rate of baseband signal (Hz)
pulseWidth = 20e-6;              % Define pulse width (seconds)
pulseRepitionInterval = 50e-6;% Define pulse repetition interval (seconds)
bandWidth =1e6;     % Bandwidth of the pulse (Hz)
IPhase = 0;                      % Initial phase in degrees of IComponent
QPhase = 90;                     % Initial phase in degrees of QComponent

tVector = 0:1/sampleRate:pulseWidth; % Vector to generate ideal pulse
IComponent = chirp(tVector,-bandWidth/2,tVector(end), bandWidth/2,'linear',IPhase);
QComponent = chirp(tVector,-bandWidth/2,tVector(end), bandWidth/2,'linear',QPhase);
IQData = IComponent + 1i*QComponent;

% Normalize amplitude.
scale = max(max(abs(real(IQData))), max(abs(imag(IQData))));
normPulse = IQData / scale;

% Pad the pulse with zeros for the off cycle of the pulse.
offPulse = 0:1/sampleRate:((pulseRepitionInterval-pulseWidth)/2);
chirpedPulse = [offPulse normPulse offPulse];

figure(1); hold on;
plot((1:length(chirpedPulse))/sampleRate, real(chirpedPulse),'b');
plot((1:length(chirpedPulse))/sampleRate, imag(chirpedPulse),'g');
plot((1:length(chirpedPulse))/sampleRate, abs(chirpedPulse),':r','linewidth',2);
grid()
title('One cycle of LFM pulse waveform');
xlabel('Time (s)');
ylabel('Magnitude (V)');
legend('I component','Q component','Envelope');
axis tight; axisLimits = axis; axis([axisLimits(1:2) 1.2*(axisLimits(3:4))]);

isp=chirpedPulse;
save('chirpedPulse_4.mat','isp','-v6');

%% Create and download wv file using R&S MATLAB Toolkit (R&S application note 1GP60)

```

4.1.2 Create and download wv file using R&S MATLAB toolkit

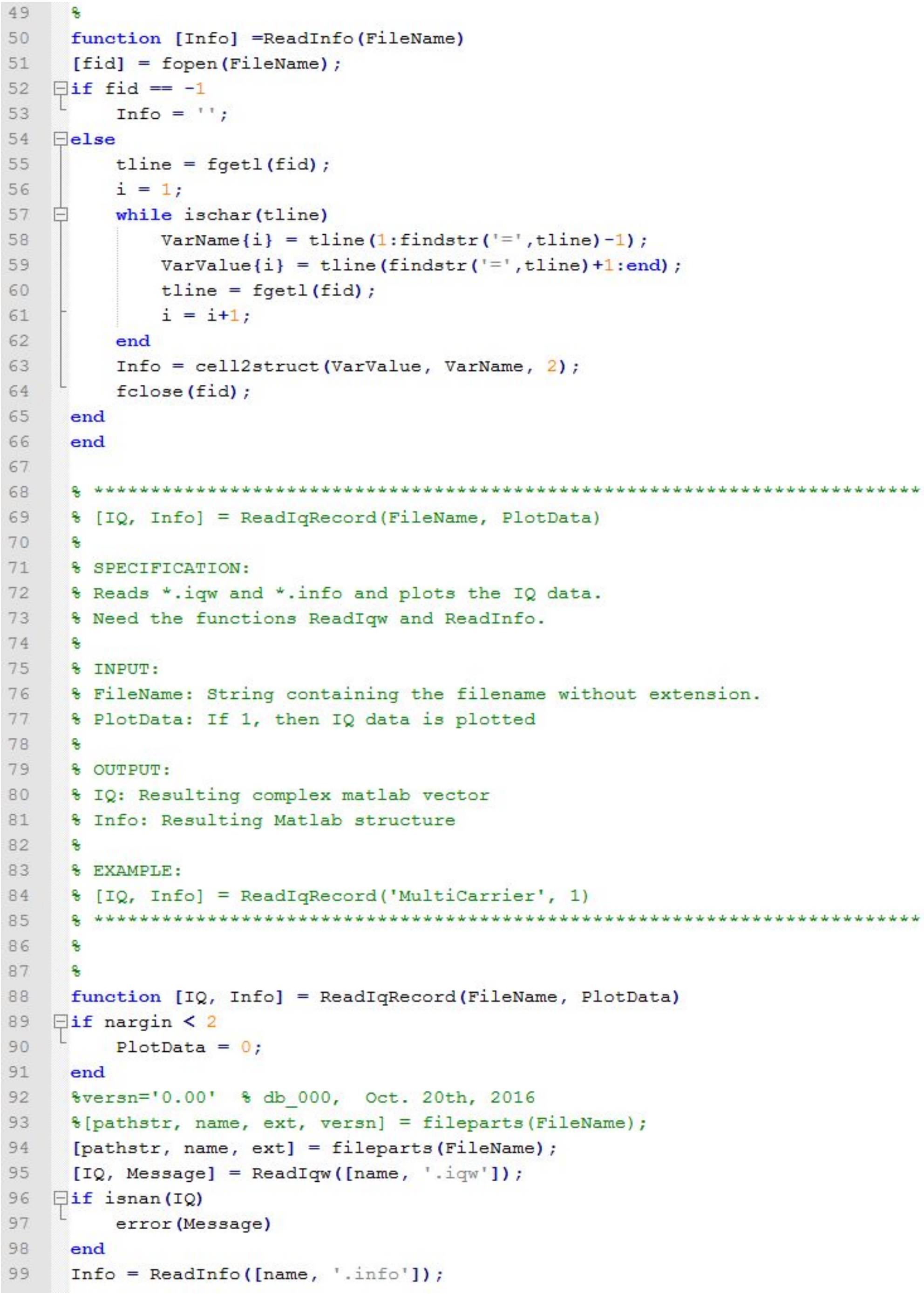


```
axis tight; axisLimits = axis; axis([axisLimits(1:2) 1.2*(axisLimits(3:4))]);

isp=chirpedPulse;
save('chirpedPulse_4.mat','isp','-v6');

%% Create and download wv file using R&S MATLAB Toolkit (R&S application note 1GP60)


[err, instrObj] = rs_connect( 'TCPIP','CMA18018-100309' );
if (err ~= 1)
  disp('*** Exception from rs_connect');
end


[status, answer] = rs_send_query( instrObj, '*IDN?');

if (status ==1)
  disp(['answer from *IDN? : ' answer]);
else
  disp('*** Exception from rs_send_query');
end

% populate waveform structure
IQInfo.I_data          = imag(chirpedPulse);
IQInfo.Q_data          = real(chirpedPulse);
IQInfo.comment         = 'Waveform created from MATLAB';
IQInfo.clock           = sampleRate;
IQInfo.filename        = 'chirpedPulse_4.wv';

IQInfo.path            = 'D:\Rohde-Schwarz\CMA\Data\waveform\';      % waveform path of CMA180

% Following markers can be linked to CMA180's TRIG OUT connector
% via master SETUP / Base / Trigger Out Source
IQInfo.markerlist.one = [[0 1];[100 0]];  % not used in CMA180 !
IQInfo.markerlist.two = [[0 1];[2500 0]];  % CMA180: AFRF Gen1: Waveform Marker 2
IQInfo.markerlist.three = [[0 1];[1000 0]];  % CMA180: AFRF Gen1: Waveform Marker 3
IQInfo.markerlist.four = [[0 1];[20 0]];  % CMA180: AFRF Gen1: Waveform Marker 4

% generate waveform (no sending of waveform)
playback   = 1;       % not usable for CMA180
save_local = 1;       % save waveform to PC
[status] = rs_generate_wave (instrObj, IQInfo, playback, save_local);

if (status ~= 1)
  disp('*** Exception from rs_generate_wave');
end

```

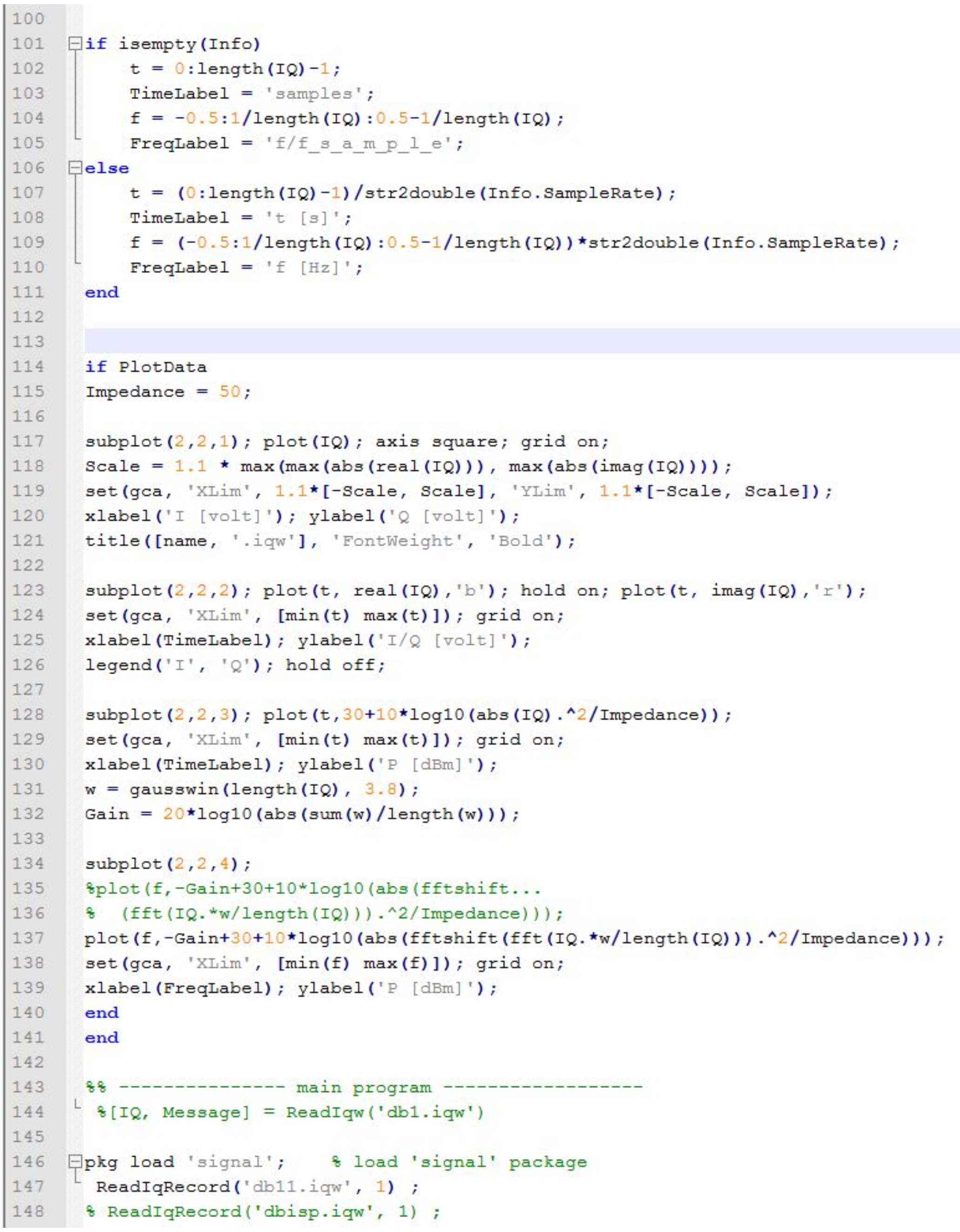
4.2 MATLAB Program Listing "Recorded IQ data evaluation"



```
% **************************************************************************
% [IQ, Message] = ReadIqw(FileName)
%
% SPECIFICATION:
% Converts the selected *.iqw file into a matlab IQ vector.
%
% INPUT:
% FileName: String containing the name of the file to be converted.
%
% OUTPUT:
% IQ: Resulting complex matlab vector
% Message: Returns a system dependent error message if the open is not
% successful
%
% EXAMPLE:
% [IQ, Message] = ReadIqw('MultiCarrier.iqw')
% **************************************************************************

clear;

function [IQ, Message]=ReadIqw(FileName)
[fid, Message] =fopen(FileName);
if fid == -1
    IQ = nan;
else
    x = fread(fid,'single');
    IQ = x(1:2:end)+1i*x(2:2:end);
    fclose(fid);
end
end


% **************************************************************************
% [Info] = ReadInfo(FileName)
%
% SPECIFICATION:
% Converts the selected *.info file into a matlab structure.
%
% INPUT:
% FileName: String containing the name of the file to be converted.
%
% OUTPUT:
% Info: Resulting Matlab structure
%
% EXAMPLE:
% [Info] = ReadInfo('MultiCarrier.info')
% **************************************************************************
%
%
```



```
%
function [Info] =ReadInfo(FileName)
[fid] = fopen(FileName);
if fid == -1
    Info = '';
else
    tline = fgetl(fid);
    i = 1;
    while ischar(tline)
        VarName{i} = tline(1:findstr('=',tline)-1);
        VarValue{i} = tline(findstr('=',tline)+1:end);
        tline = fgetl(fid);
        i = i+1;
    end
    Info = cell2struct(VarValue, VarName, 2);
    fclose(fid);
end
end

% *************************************************************************
% [IQ, Info] = ReadIqRecord(FileName, PlotData)
%
% SPECIFICATION:
% Reads *.iqw and *.info and plots the IQ data.
% Need the functions ReadIqw and ReadInfo.
%
% INPUT:
% FileName: String containing the filename without extension.
% PlotData: If 1, then IQ data is plotted
%
% OUTPUT:
% IQ: Resulting complex matlab vector
% Info: Resulting Matlab structure
%
% EXAMPLE:
% [IQ, Info] = ReadIqRecord('MultiCarrier', 1)
% *************************************************************************
%
%
function [IQ, Info] = ReadIqRecord(FileName, PlotData)
if nargin < 2
    PlotData = 0;
end
%versn='0.00'  % db_000,  Oct. 20th, 2016
%[pathstr, name, ext, versn] = fileparts(FileName);
[pathstr, name, ext] = fileparts(FileName);
[IQ, Message] = ReadIqw([name, '.iqw']);
if isnan(IQ)
    error(Message)
end
Info = ReadInfo([name, '.info']);
```



```
if isempty(Info)
    t = 0:length(IQ)-1;
    TimeLabel = 'samples';
    f = -0.5:1/length(IQ):0.5-1/length(IQ);
    FreqLabel = 'f/f_s_a_m_p_l_e';
else
    t = (0:length(IQ)-1)/str2double(Info.SampleRate);
    TimeLabel = 't [s]';
    f = (-0.5:1/length(IQ):0.5-1/length(IQ))*str2double(Info.SampleRate);
    FreqLabel = 'f [Hz]';
end


if PlotData
Impedance = 50;

subplot(2,2,1); plot(IQ); axis square; grid on;
Scale = 1.1 * max(max(abs(real(IQ))), max(abs(imag(IQ))));
set(gca, 'XLim', 1.1*[-Scale, Scale], 'YLim', 1.1*[-Scale, Scale]);
xlabel('I [volt]'); ylabel('Q [volt]');
title([name, '.iqw'], 'FontWeight', 'Bold');

subplot(2,2,2); plot(t, real(IQ),'b'); hold on; plot(t, imag(IQ),'r');
set(gca, 'XLim', [min(t) max(t)]); grid on;
xlabel(TimeLabel); ylabel('I/Q [volt]');
legend('I', 'Q'); hold off;

subplot(2,2,3); plot(t,30+10*log10(abs(IQ).^2/Impedance));
set(gca, 'XLim', [min(t) max(t)]); grid on;
xlabel(TimeLabel); ylabel('P [dBm]');
w = gausswin(length(IQ), 3.8);
Gain = 20*log10(abs(sum(w)/length(w)));

subplot(2,2,4);
%plot(f,-Gain+30+10*log10(abs(fftshift...
%   (fft(IQ.*w/length(IQ))).^2/Impedance)));
plot(f,-Gain+30+10*log10(abs(fftshift(fft(IQ.*w/length(IQ))).^2/Impedance)));
set(gca, 'XLim', [min(f) max(f)]); grid on;
xlabel(FreqLabel); ylabel('P [dBm]');
end
end

%% --------------- main program ------------------
 %[IQ, Message] = ReadIqw('db1.iqw')

pkg load 'signal';    % load 'signal' package
 ReadIqRecord('db11.iqw', 1) ;
% ReadIqRecord('dbisp.iqw', 1) ;
```

4.3 References

[1] R&S®CMA180, June 2014, Radio Test Set, Specifications 06.00 - Oct 27, 2016-PD 3606.9404.22, available from http://www.rohde-schwarz.com, search tag "CMA180", Brochures and Data Sheets

[2] R&S®SMW200A Vector Signal Generator, https://www.rohde-schwarz.com/product/smw200a-productstartpage_63493-38656.html, search tag "SMW", Brochures and Data Sheets

[3] R&S ARB Toolbox Application Note 1GP88, available from http://www.rohde-schwarz.com/appnote/1GP88, retrieved Dec. 7th, 2016

[4] R&S®WinIQSIM2™ Simulation Software , available from https://www.rohde-schwarz.com/product/winiqsim2-productstartpage_63493-7614.html, retrieved Dec. 7th, 2016

[5] R&S®VSE Vector Signal Explorer Software, available from https://www.rohde-schwarz.com/product/vse-productstartpage_63493-124109.html, retrieved Dec. 7th, 2016

[6] Using R&S®Forum Application for Instrument Remote Control 1MA196, available from http://www.rohde-schwarz.com/appnote/1MA196, retrieved Dec. 14th, 2016

[7] MathWorks homepagehttp://www.mathworks.com/, retrieved Dec. 7th, 2016

[8] Application Note 1MA074: RSCommander - Versatile Software Tool for Rohde & Schwarz Instruments, available on http://www.rohde-schwarz.com/appnote/1MA74, retrieved Jan. 24th, 2017

[9] Application Note 1GP60: R&S MATLAB® Toolkit for Signal Generators, available on http://www.rohde-schwarz.com/appnote/1GP60, retrieved Jan. 27th, 2017

5 Rohde & Schwarz

Der Elektronikkonzern Rohde & Schwarz bietet innovative Lösungen auf den Arbeitsgebieten Messtechnik, Rundfunk- und Medientechnik, sichere Kommunikation, Cyber-Sicherheit sowie Funküberwachungs- und -ortungstechnik. Vor mehr als 80 Jahren gegründet, ist das selbstständige Unternehmen in über 70 Ländern mit einem engmaschigen Vertriebs- und Servicenetz präsent.

In seinen angestammten Arbeitsgebieten zählt der Elektronikkonzern zu den führenden Anbietern weltweit. Der Firmensitz ist in München, Deutschland. Ergänzend steuern je ein regionales Headquarter in Singapur sowie in Columbia, USA, die Geschäfte.

Nachhaltige Produktgestaltung

- Umweltverträglichkeit und ökologischer Fußabdruck
- Energie-Effizienz und geringe Emissionen
- Langlebigkeit und optimierte Gesamtbetriebskosten

Certified Quality Management
ISO 9001

Certified Environmental Management
ISO 14001

Kontakt

- Europa, Afrika, Mittlerer Osten - customersupport@rohde-schwarz.com
  +49 89 4129 12345
- Nordamerika - customer.support@rsa.rohde-schwarz.com
  1-888-TEST-RSA (1-888-837-8772)
- Lateinamerika - customersupport.la@rohde-schwarz.com
  +1-410-910-7988
- Asien/Pazifik - customersupport.asia@rohde-schwarz.com
  +65 65 13 04 88
- China - customersupport.china@rohde-schwarz.com
  +86-800-810-8228 / +86-400-650-5896

Rohde & Schwarz GmbH & Co. KG

Mühldorfstraße 15 | D - 81671 München

+ 49 89 4129 - 0 | Fax + 49 89 4129 – 13777

www.rohde-schwarz.com

Die Nutzung dieser Application Note und der zugehörigen Programme unterliegt den Nutzungsbedingungen, wie sie auf der Download Area der Rohde & Schwarz Webseite festgelegt sind.