Products: AMIQ / WinIQSIM, SMIQ

Creating Test Signals for Bluetooth with AMIQ / WinIQSIM and SMIQ

Bluetooth is a universal radio interface using the license-free 2.45 GHz frequency band. It enables electronic devices to connect and communicate without connecting cables in short-range, ad hoc networks. As the SMIQ Vector Signal Generator in combination with I/Q Modulation Generator AMIQ is able to generate nearly any kind of digitally modulated signal, this combination is also an ideal signal source for Bluetooth test signals.

ROHDE&SCHWARZ

Subject to change – Dr. René Desquiotz 11/99 - Application Note 1GP38_0E

Contents

1 Overview

The combination SMIQ / AMIQ can generate Bluetooth signals, including the slot structure of the physical layer and frequency hopping. This application note describes the procedures for generating signals, especially how to create Bluetooth data structures in the physical layer with the WinIQSIM software, with frequency hopping using the list mode of the SMIQ vector signal generator in cooperation with AMIQ also covered in detail.

2 Introduction to Bluetooth

Bluetooth is a universal radio interface using the license-free 2.45 GHz frequency band. It enables electronic devices to connect and communicate without connecting cables in short-range, ad hoc networks. Each device can communicate with up to seven other devices per piconet. Units can simultaneously belong to several piconets.

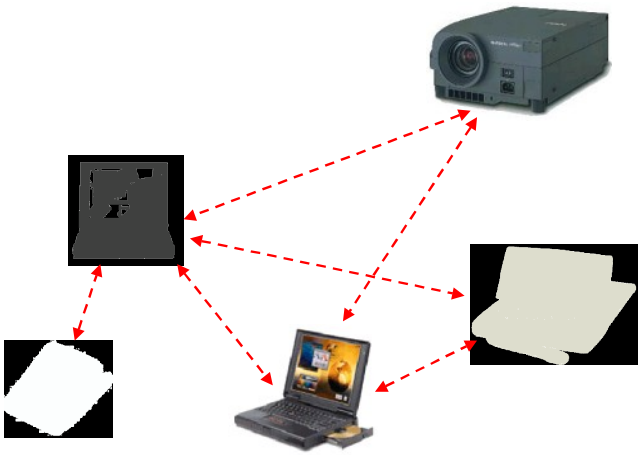


Fig.: 2-1 Typical members of a Bluetooth piconet

The potential applications for Bluetooth cover a wide range, from wireless connection between a PC and a printer, to data exchange between mobile phones, digital cameras, PCs and organizers.

The basic parameters of the Bluetooth system are shown in Fig.: 2-2. The radio frequency lies in the 2.4 GHz ISM band. Bluetooth uses frequency hop spread-spectrum technology, dividing the frequency band into several hop channels. During a connection, radio transceivers hop from one channel to another in a pseudo-random fashion. The channel spacing is 1 MHz, the whole frequency band covers 79 MHz (23 MHz in some countries). As Bluetooth is a TDMA system, each channel is divided into 625 µs intervals in the time domain. A communication channel uses a different hop frequency for each slot, leading to a nominal hop rate of 1600 hops/s. One packet can be transmitted per slot. Subsequent slots are alternately used for transmitting and receiving. The modulation is of GFSK type, with a symbol rate of 1 MSym/s.

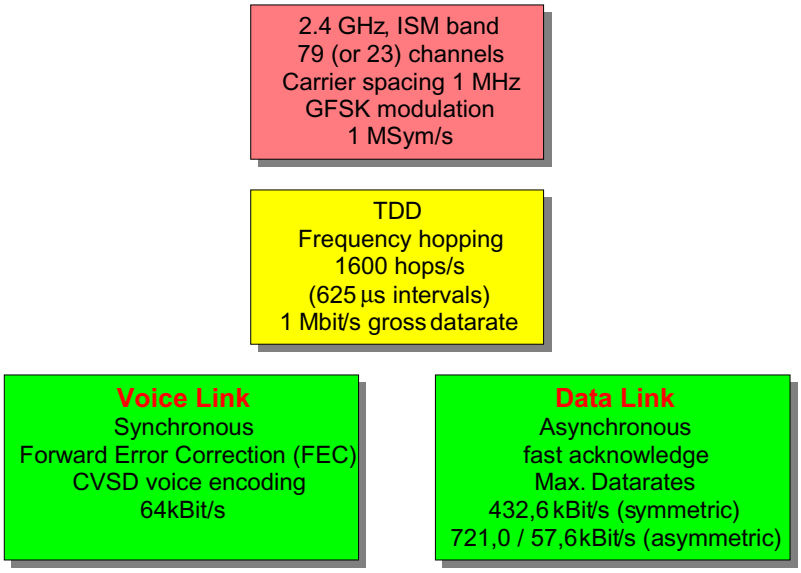


Fig.: 2-2 Basic connection parameters for Bluetooth

The most important parameters of the modulation are given in the table below.

modulation type	2FSK
symbol rate	1 MHz
modulation index	0.28 – 0.35
max. frequency deviation	140 – 175 kHz
baseband filter	Gauss, B*T = 0.5

These parameters can be realized either with the SMIQ modulation coder or AMIQ and WinIQSIM. The main difference is that the SMIQ modulation coder generates real-time signals while the AMIQ plays waveforms precalculated by WinIQSIM.

The methods of generation are described in the next sections.

3 General Setups

The options required for using SMIQ as a stand-alone Bluetooth signal source are: SMIQB10 or B20 Modulation Coder, and the SMIQB11 Data Generator if customized data is to be used.

Used in conjunction with AMIQ / WinIQSIM, see the setup in **Fehler! Verweisquelle konnte nicht gefunden werden.**. The AMIQ I and Q outputs are connected to the I and Q inputs of the SMIQ signal generator. SMIQ is operated in mode VECTOR MOD and needs no options in this case.

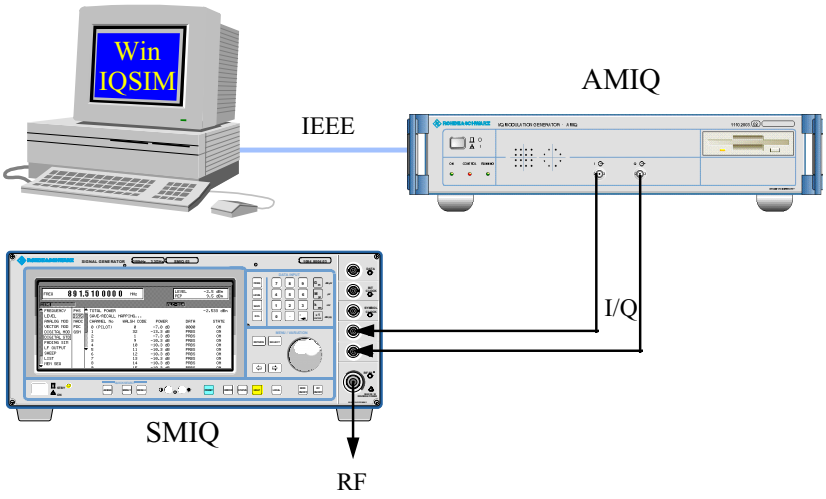


Fig.: 3-1 *Setup for calculating waveforms and controlling AMIQ via WinIQSIM.*

The parameter setting and signal calculation is done in WinIQSIM running on a PC. The calculated waveform is then transmitted to the AMIQ RAM or hard disk via the GPIB connection. The waveform can also be stored on the PC hard disk and transmitted later.

The AMIQ hardware is also controlled by WinIQSIM via the GPIB connection – including selecting waveforms pre-stored on the AMIQ harddisk. Alternatively, SMIQ can provide hardware control via GPIB instead of WinIQSIM on the PC. In this case the PC is only required for waveform calculation.

4 Creating Bluetooth Signals

Creating simple continuous signals with the SMIQ Modulation Coder

For a sophisticated signal generator like SMIQ, creating a continuous Bluetooth signal is a straightforward task.

The modulation parameters are:

modulation type	2FSK
max. frequency deviation	140 kHz
symbol rate	1 MHz
baseband filter	Gauss, B*T = 0.5

The following example settings generate a simple (i.e not structured) Bluetooth signal on 2.4 GHz RF frequency. The data bits are of pseudo-random type, i.e. a PRBS sequence.

Using the SMIQ modulation coder the SMIQ settings to be made are:

PRESET

FREQUENCY 2.4 GHz

LEVEL 0 dBm

DIGITAL MOD:

STATE ON

SOURCE... PRBS

PRBS LENGTH 9

MODULATION... TYPE 2FSK

FSK DEV 140 kHz

SYMBOL RATE 1 MHz

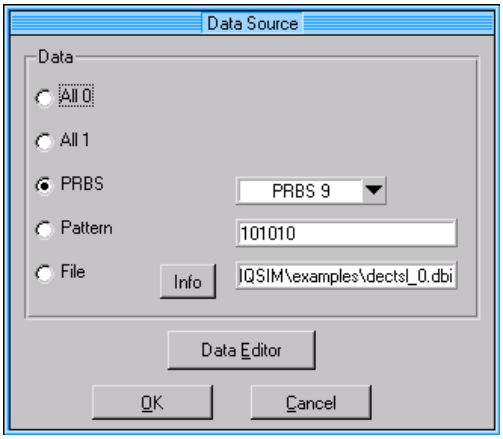
FILTER... TYPE GAUSS

FILTER PARAMETER 0.5

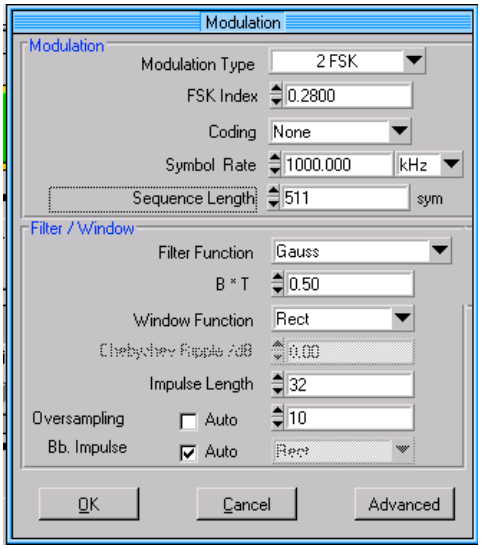
Creating simple continuous signals with WinIQSIM / AMIQ

The settings in WinIQSIM for a simple continuous signal are the following.

Start the WinIQSIM software and select **`File --> new --> Single Carrier`**. Open the **`Data Source`** panel and choose **PRBS 9** as data.



Confirm the setting by clicking on the **OK** button. Then open the **`Modulation Settings`** panel and set the parameters as shown below.



The sequence length of 511 symbols (= 511 bits for 2FSK modulation) creates one complete PRBS 9 sequence. The AMIQ will repeat this calculated sequence continuously, so the resulting signal is equivalent to a real-time PRBS signal.

Again confirm with the **OK** button. Select **`Graphics --> Show Graphic...`** to calculate the signal.

Save your WinIQSIM settings with **`File --> Save settings as...`** for further use.

To transmit the calculated waveform to AMIQ, select **`AMIQ --> Transmission...`**. Choose **Internal (WinIQSIM)** as the source and **AMIQ HD** or **AMIQ RAM** as the destination. If you want to store the waveform on your PCs harddisk, choose **File** as the destination and set the appropriate path.

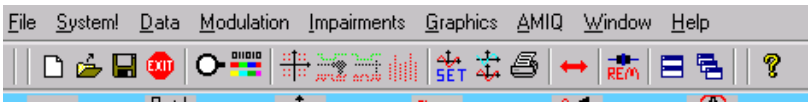
Creating packet structures with the WinIQSIM data editor

WinIQSIM contains a powerful data editor for creating user-defined TDMA data structures. In the following examples we use the data editor to build a typical Bluetooth frame. Although the procedure described below starts from a zero point, you can also modify existing data editor settings.

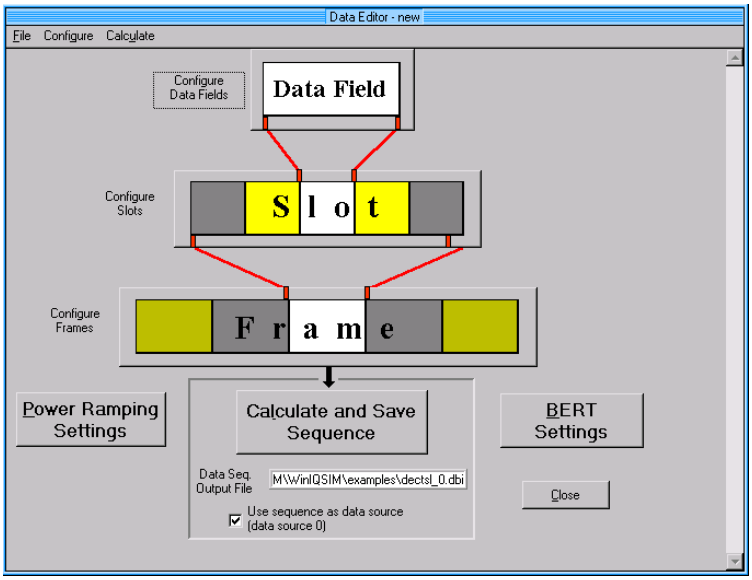
Note: *An example file for the data editor containing several Bluetooth data structures is provided with this Application Note. The file is called Bluetooth_data_structures.ded.*

Creating a Bluetooth data field library

Open the **`Data Editor`** by clicking on the small icon  in the icon bar.

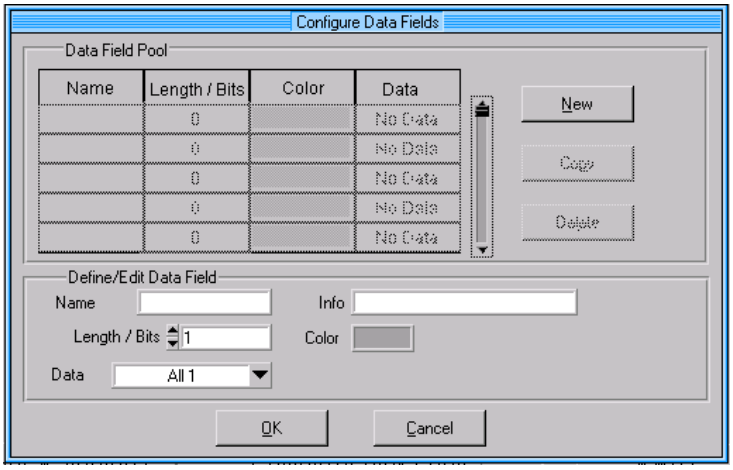


The **`Data Editor`** panel appears. Select **`File --> New...`** so that the panel looks like this.

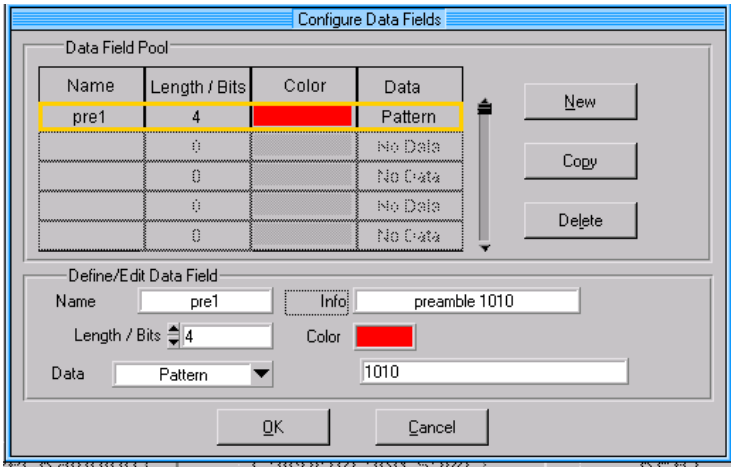


The **`Data Editor`** has three levels: data fields, slots and frames. First create the necessary data fields.

Click on the white rectangle named **Data Field**. The following panel appears.



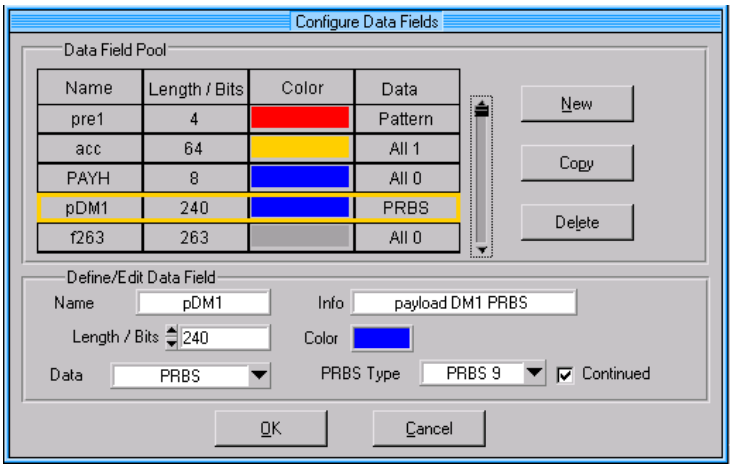
Click on the **New** button to create a new data field. Edit the field with the **Define/Edit** functions in the lower half of the panel.



Create all the necessary data fields in the same way. For a typical Bluetooth signal we need:

- A 4-bit preamble field with 0101.
- A 64-bit access field (here filled with ones).
- An 8-bit payload header (here filled with zeroes).
- A 240-bit payload field with user data (here PRBS data).
- A 54-bit header field (here filled with PRBS data).
- Two dummy fields to fill the slots up to the 625 bits defined in the specifications. One field is 263 bits long, the other 625 bits.

The entire data field pool is shown in the figure below.

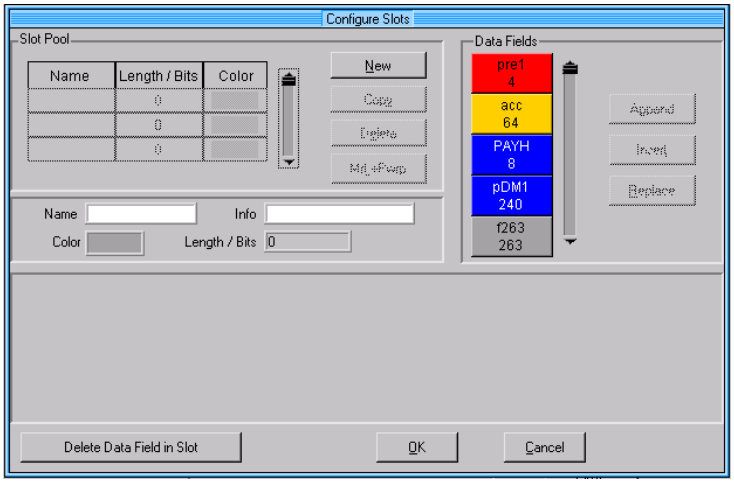


The pDM1 field (containing the user data) is filled with continuous PRBS, the pseudo random sequence is continued in successive data fields of the type pDM1.

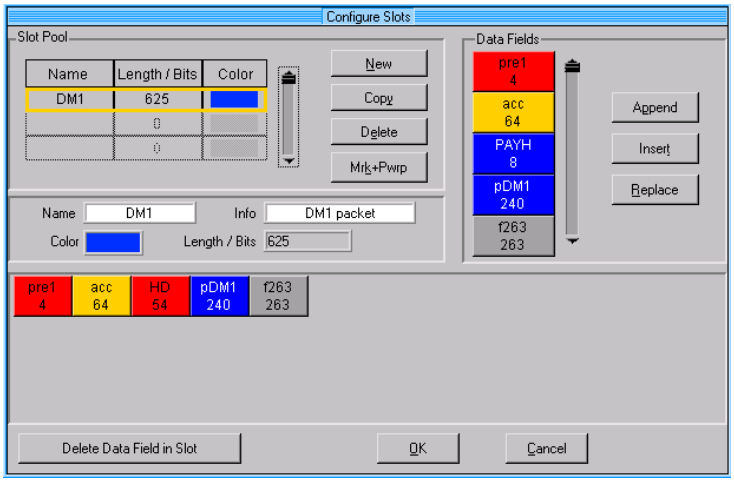
After creating all the necessary data fields, confirm (and close the panel) with the **OK** button.

Combining data fields in Bluetooth time slots

Next step is combining the data fields in timeslots. Click on the **Slot** field in the **`Data Editor`** panel to open the **`Slot`** menu.



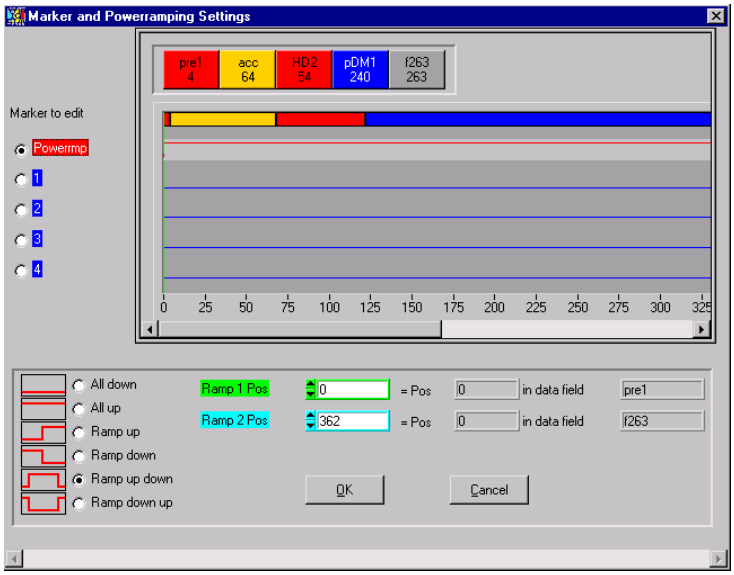
Click on the **New** button to call and edit a new slot. The list of available data fields is shown in the upper right part of the panel. These are the data fields you created in the previous step. Build a standard DM1 packet as shown below.



The packet consists of preamble, access field, header and user data. The rest up to the required 625 bits is filled with the dummy data field f263.

Configure powerramping

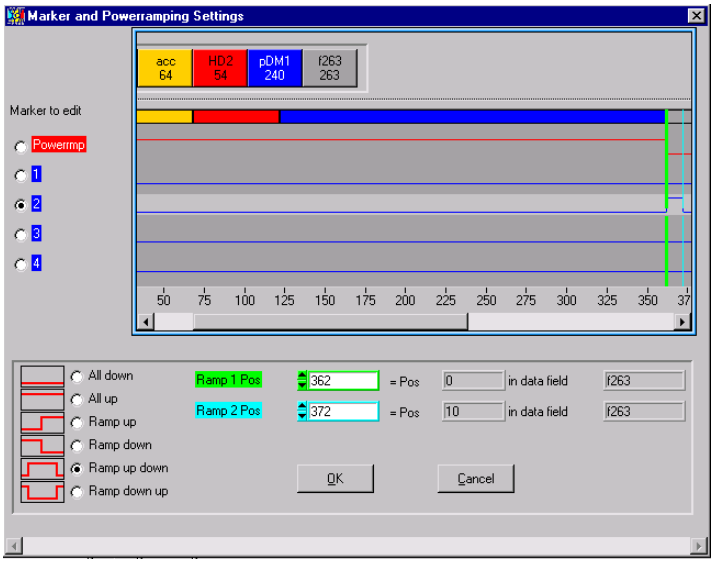
To configure the powerramping for the slot, click on the **Mrk + Pwp** (Marker and Powerramping) button.



The powerramping is programmed with an **up-down ramp**, starting at the first bit of the slot (bit 0 of the preamble) and ending at bit 362 of the slot (bit 0 of the dummy data field).

Programming a marker with a trigger signal

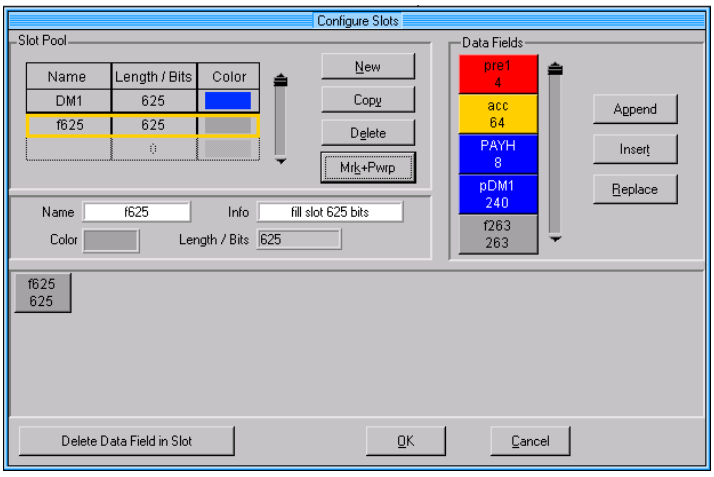
In addition to the powerramping we will program one of the marker outputs (Marker 2) to generate a trigger signal at the beginning of the slot. This trigger signal will be used later in section 5 for frequency hopping.



Click on **Marker 2** to edit the second marker channel. Program a **ramp up down** sequence with ramp up at bit 0 of the f263 dummy field (position 362) and ramp down 10 bits later (position 372). This generates a trigger signal of 10 µs length just after the pDM1 data packet, as one bit takes 1 µs with a bit rate of 1 Mbit/s.

Confirm with the **OK** button, this returns you to the slot panel.

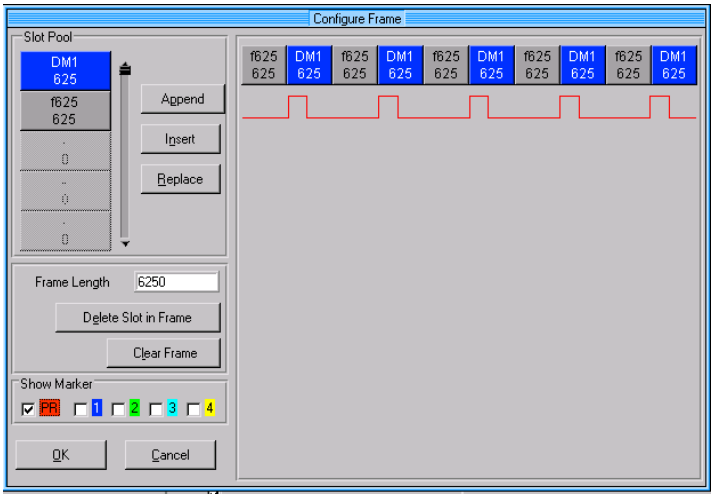
Build a second slot consisting of only the 625-bit dummy data field. This dummy slot will be used for the „power down“ parts of the signal. Therefore the powerramping setting for the dummy slot must be **all down**.



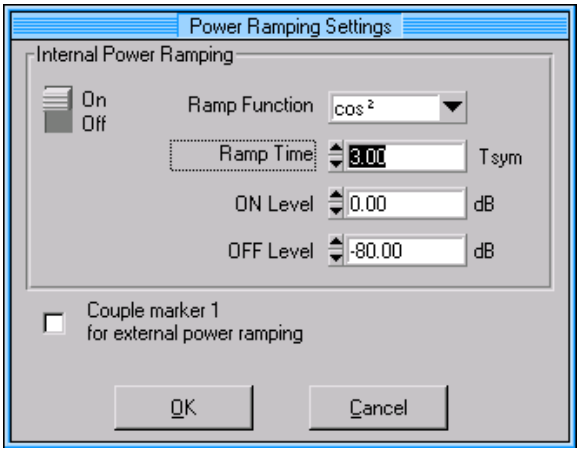
Creating a structured Bluetooth signal

Combine the two slots as an entire signal. Confirm the slot settings with the **OK** button to close the `Slot` panel and return to the main menu of the `Data Editor`.

Click on the **Frame** field to open the frame configuration panel. As every Bluetooth device only sends in either even or odd timeslots, we create a sequence of alternating DM packages and dummy slots as shown in the figure below.

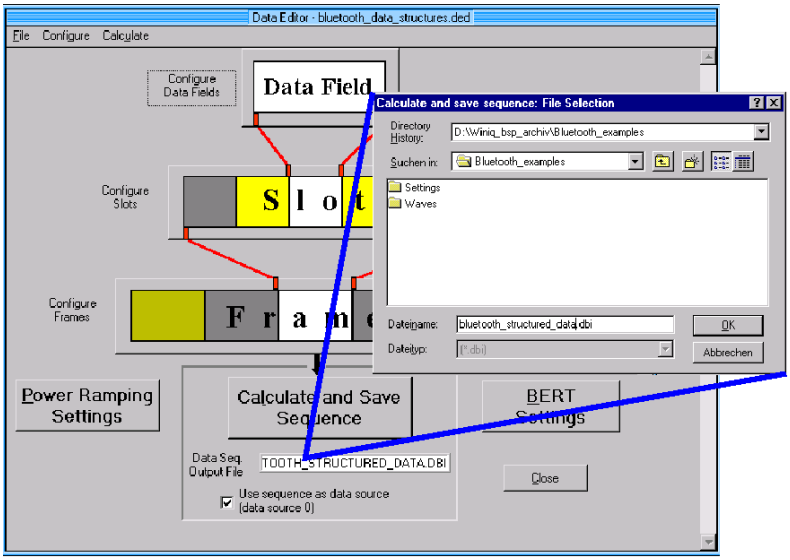


Confirm with the **OK** button. Back in the main panel of the **`Data Editor`**, open the **powerramping** settings to configure the shape of the powerramping.

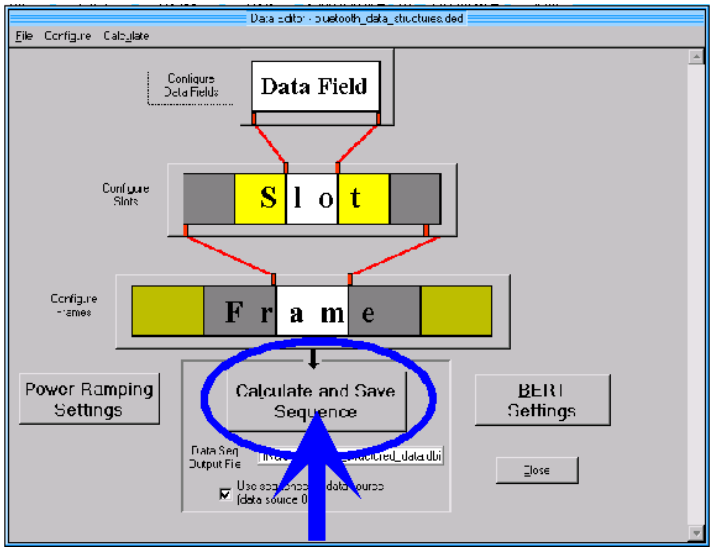


Set the **Ramp Time** to 3 Tsym. This is slow enough for an „analog-like“ ramp which avoids spurious in the frequency spectrum due to steps in the time signal. On the other hand it is fast enough to perform the entire ramp process during the preamble field. Set the **ON Level** to 0 dB and the **OFF Level** to –80 dB, leading to full range powerramping. Confirm with **OK** to return to the main panel of the **`Data Editor`**.

Finally, calculate a bit sequence with this frame structure and save the sequence in a file. Click on the white panel below **Calculate and save Sequence**. This opens a save file window, specify the file name and path of the sequence to be saved. WinIQSIM saves the sequences created with the **`Data Editor`** in a format called .dbi which is an ASCII file with a special header.

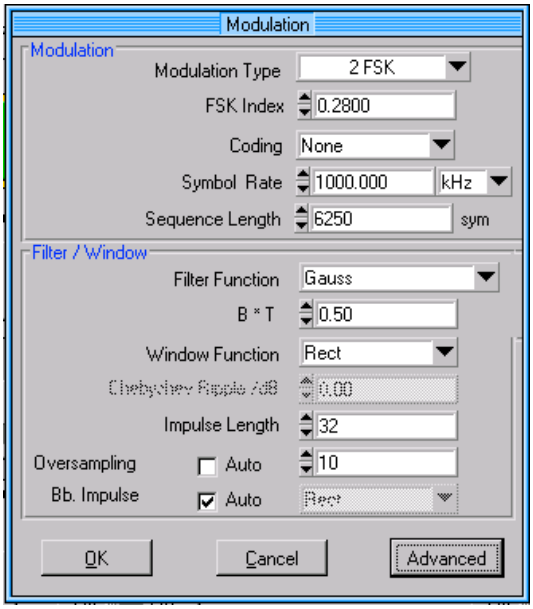


Name your sequence, for example, *bluetooth_structured_data.dbi* and confirm the settings with **OK**. Then click on the **Calculate and save Sequence** button. The sequence is calculated and stored in the file. At the same time WinIQSIM is told to use this .dbi file as **Data Source**. (You can control this in the `Data Source` panel.) The **Sequence Length** is set to the length of the data sequence (here: 6250 bits = 10 slots).

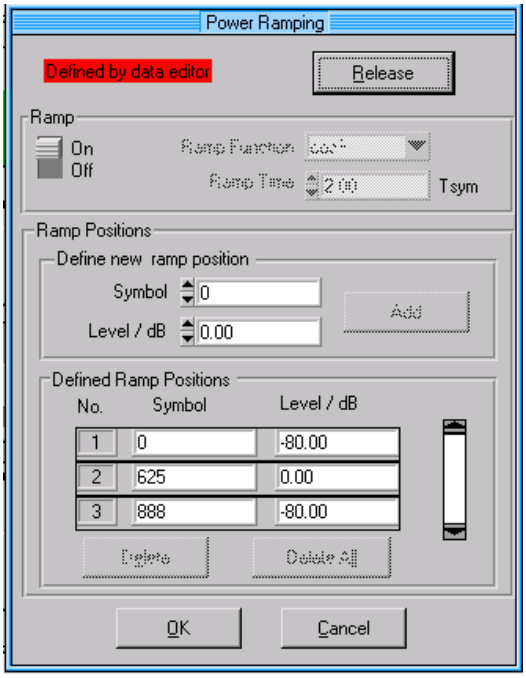


Hint: *There is one trap you can fall in at this point: always check the filename and path before saving a sequence. Otherwise you might overwrite an existing .dbi file by mistake.*

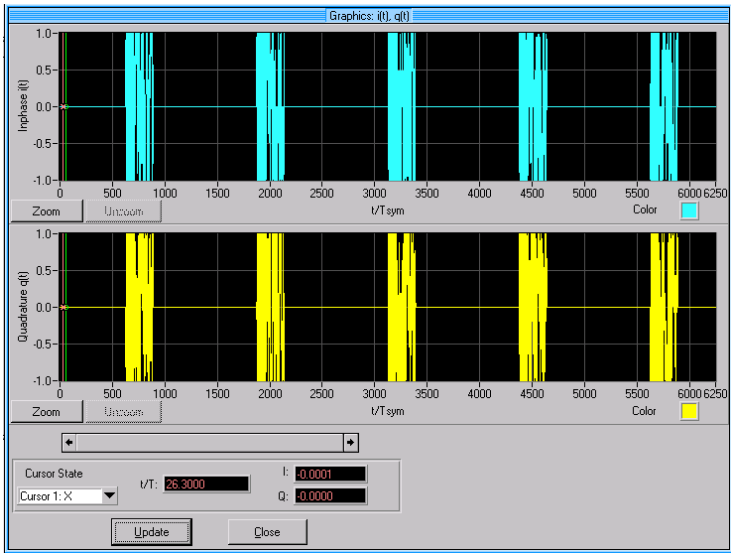
The **Modulation Settings** are the same as in the simple case (except for the sequence length).



Make sure that the **`Powerramping`** function is activated (**`Powerramping`** panel in the block diagram is green) and that the powerramping is defined by the Data Editor.



Calculate the signal with the **`Graphics`** function – for example, click on the **Graphics** icon  in the iconbar. The signal in time domain (amplitude and frequency) should now be as shown below:



Save your WinIQSIM settings with **`File --> Save settings as...`** for further use.

To transmit the calculated waveform to AMIQ, select **`AMIQ --> Transmission...`**. Choose **Internal (WinIQSIM)** as source **and AMIQ HD** or **AMIQ RAM** as destination. If you want to store the waveform on the PCs hard disk, choose **File** as destination and set the appropriate path.

Make sure to activate the AMIQ 2.5 MHz low pass filter to suppress the aliasing products. Select **`AMIQ --> remote control and BERT --> Hardware Setting`** and set **Filter** to 2.5 MHz.

5 Frequency Hopping

Frequency hopping with AMIQ / WinIQSIM

As AMIQ is also a sophisticated multi carrier source, one can in principle generate a signal simulating frequency hopping. The procedure to create such a signal is described in the WinIQSIM Application Manual section 4.7. However, the Bluetooth frequency band covers 79 MHz in most countries. As AMIQ can cover a total signal bandwidth of about 40 MHz, it is more suitable to use the SMIQ as hopping device.

Frequency hopping with SMIQ

SMIQ Vector Signal Generator in LIST MODE can be used for frequency hopping by programming a list of hop frequencies. The AMIQ marker outputs can be used to trigger SMIQ. Every trigger signal makes SMIQ step to the next frequency in the list.

The main characteristic of this method is that all hop frequencies have to be programmed in advance, it is not possible to send the next desired frequency during the transmission.

A different way is to use the ‚Fast Restore' mode [1]. All possible frequencies are stored in system states. Then it is possible to address these states (and therefore the hop frequencies) by a GPIB command.

For both methods, the hops are triggered directly after an active slot. While power is down SMIQ hops to the next desired frequency. As every Bluetooth device only sends in either odd or even timeslots, there is always enough dwell time between two transmission packets.

Example: hopping between two frequencies

The following simple example demonstrates the frequency hopping capabilities of SMIQ and how AMIQ and SMIQ work together.

In section 4 we generated an AMIQ waveform which is already prepared for use with frequency hopping. We programmed a trigger event on marker channel 4 at the beginning of each DM1 slot. Now we will use these trigger events to control the SMIQ.

Add an additional connection to the setup from Fig.: 3-1, between AMIQ marker 2 output and the TRIGGER input at SMIQ's rear panel.

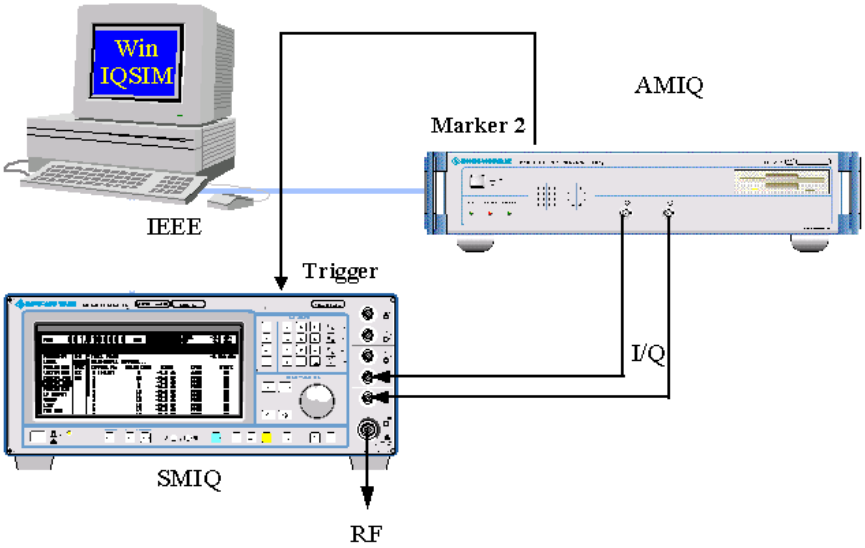


Fig.: 5-1 Setup for a Bluetooth signal including frequency hopping.

Preparing AMIQ

If the waveform from section 4 is not loaded in AMIQ RAM anymore, open the **`AMIQ --> remote control and BERT...`** menu in WinIQSIM and choose **Load HD file**. Select the appropriate file and make sure that the waveform output is started, indicated by the green **Running** light.

Activate the **Marker Output 4** with the trigger signal.

[1] This mode is only possible with the fast processor unit.

Preparing SMIQ

Now program SMIQ with the following settings:

PRESET

FREQUENCY 2.4 GHz

LEVEL 0 dBm

VECTOR MOD

STATE ON

This sets SMIQ to vector modulation, so that AMIQ provides the I/Q baseband signal.

LIST MODE

Program a list consisting of two frequencies, 2.4 GHz and 2.45 GHz:

SELECT LIST... CREATE NEW LIST

FUNCTION EDIT/VIEW

Set the list as follows

Index	Frequency	Level
00000001	2.4 GHz	0.0 dBm
00000002	2.45 GHz	0.0 dBm

LEARN

SMIQ learns the list you programmed.

MODE EXT-STEP

This command starts the list mode. Every signal from the AMIQ marker 2 triggers a hop from one frequency position to the next. As a result, the SMIQ changes frequency right after every active timeslot of the signal. After every trigger signal, SMIQ needs a certain settling time for the frequency hop (as indicated by the hatched fields in the bottom line of the figure below). But as this settling time always occurs when the signal power is down, it does not affect the signal.

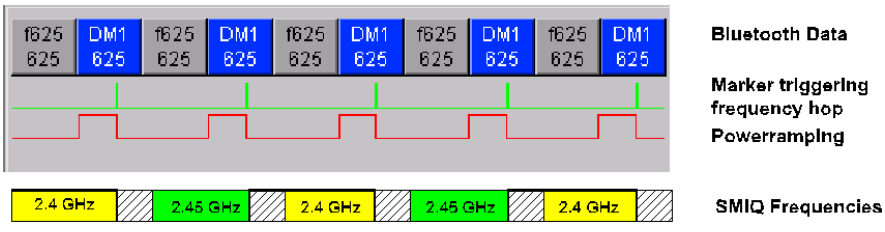


Fig.: 5-2 Bluetooth signal and related frequencies output by SMIQ

As the AMIQ waveform (and also the trigger signal) is continuosly repeated, SMIQ hops between the two frequencies until the list mode is stopped (command MODE OFF in LIST menu).

This example can easily be extended to more frequency (and level) values. Another possible method for frequency hopping would to use the „fast

restore“ mode of the SMIQ. Store all desired frequency values as „fast restore settings“ which can be adressed by very fast GPIB commands. The frequency hops then are triggered via GPIB. (See „fast restore mode“ in the SMIQ user manual for more details.)

6 Signals for Bit Error Rate Measurements

As for all TDMA systems, the BER measurement capabilities of AMIQ / WinIQSIM can be used for Bluetooth. The signals described in section 4 can be modified for use in BER measurements. The BER measurement capabilities of AMIQ / WinIQSIM are described in detail in Application Note 1GP36_0E (see references). The methods can easily be adapted to the Bluetooth signals described in section 4.

7 References

J. Haartsen: *BLUETOOTH - The universal radio interface for ad hoc, wireless connectivity*, Ericsson Review **3**, 110 (1998)

D. Mahnken: *Bluetooth – a global standard for wireless connectivity*, Rohde & Schwarz (1999)

R. Desquiotz: *Bit Error Rate Measurements with AMIQ and WinIQSIM*, Application Note 1GP36_0E, Rohde & Schwarz (1998)

Vector Signal Generator SMIQ, Operating Manual, Rohde & Schwarz (1999)

I/Q Modulation Generator AMIQ, Operating Manual, Rohde & Schwarz (1999)

Software WinIQSIM for Calculating I/Q Signals for I/Q Modulation Generator AMIQ, Software Manual, Rohde & Schwarz (1999)

Software WinIQSIM for Calculating I/Q Signals for I/Q Modulation Generator AMIQ, Application Manual, Rohde & Schwarz (1999)

8 Ordering Information

I/Q Modulation Generator		
AMIQ / WinIQSIM		1110.2003.02
Vector Signal Generator:		
SMIQ02B	300 kHz to 2.2 GHz	1125.5555.02
SMIQ03B	300 kHz to 3.3 GHz	1125.5555.03
SMIQ04B	300 kHz to 4.4 GHz	1125.5555.04
SMIQ06B	300 kHz to 6.4 GHz	1125.5555.06
Options:		
SMIQB11	Data Generator	1085.4502.04
SMIQB12	Memory Extension	1085.2800.04
SMIQB20	Modulation Coder	1125.5190.02

ROHDE & SCHWARZ GmbH & Co. KG · Mühldorfstraße 15 · D-81671 München
P.O.B 80 14 69 · D-81614 München · Telephone +49 89 4129 -0 · Fax +49 89 4129 - 3777 · Internet: http://www.rsd.de