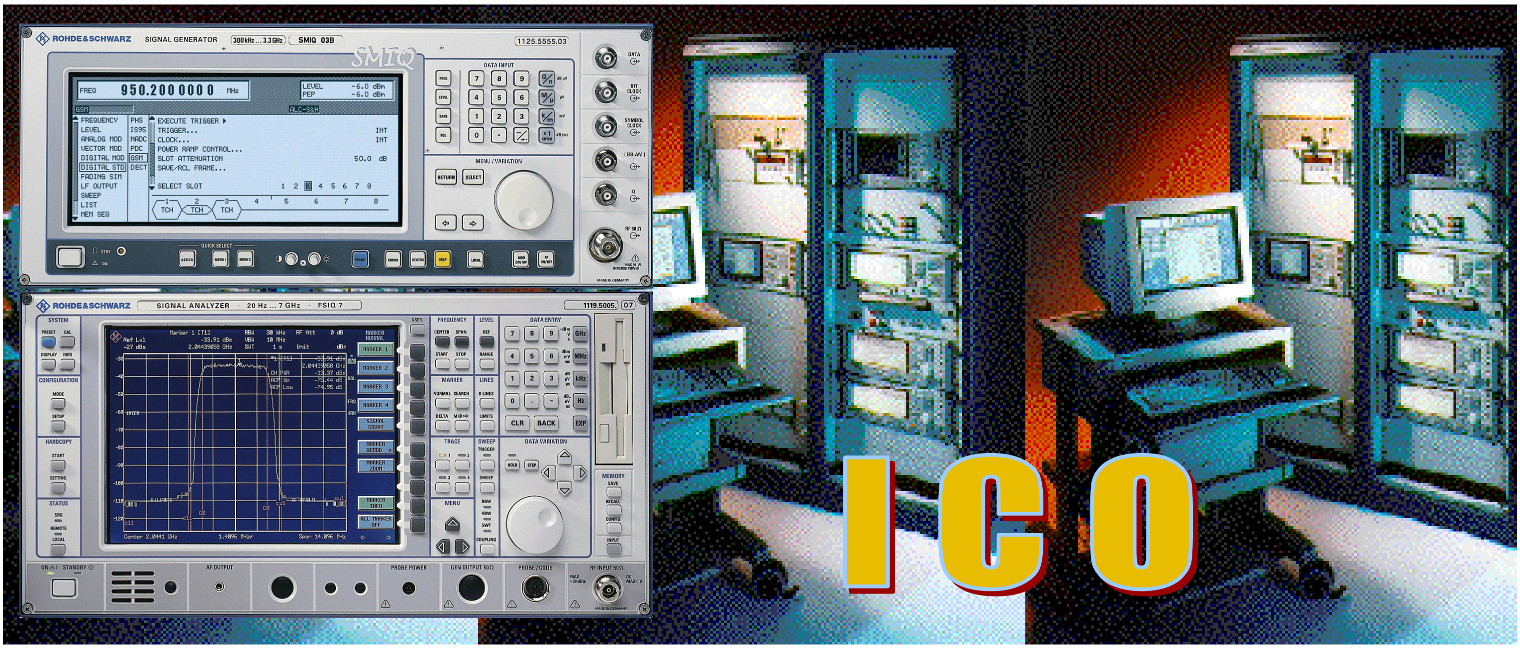


Products: SMIQ, FSE, FSIQ

Production Tests on ICO User Terminals with SMIQ and FSE/FSIQ

The combination of an SMIQ Signal Generator and an FSE / FSIQ Analyzer enables the most important transmitter and receiver tests on ICO user terminals in quality management and development. This application note presents a series of quick tests in unsynchronized, non-signalling mode.

To facilitate integration of SMIQ and FSE / FSIQ into existing test systems, complex instrument settings are supplied as sets of IEC/IEEE-bus commands. Sample files with burst data for uplink and downlink signals allow the configuration, execution and optimization of the tests described, even without a DUT.

ROHDE&SCHWARZ

Contents

1 Overview

The combination of an SMIQ Signal Generator and an FSE / FSIQ Analyzer enables the most important transmitter and receiver tests on ICO user terminals (**UT**s) in production, quality management and development. This application note presents fast transmitter and receiver measurements in unsynchronized, non-signalling mode.

Transmitter measurements:

- peak and RMS power, power versus time
- phase and frequency error
- spectral measurements

Receiver measurements:

- Adjustment of automatic level control
- Bit error rate measurement
- IQ impairments

To facilitate integration of SMIQ and FSE / FSIQ into existing test systems, complex instrument settings as supplied with the units as sets of **IEC/IEEE-bus commands**. Sample files with burst data for uplink and downlink signals allow the configuration, execution and optimization of the tests described **even without DUT**.

All analyzer measurements can be performed with either an **FSIQ** or an **FSE** fitted with option **FSE-B7**. Operation is the same for either type of instrument. For clarity's sake, only Analyzer FSIQ is mentioned hereinafter.

The generator SMIQ has to be equipped with options SMIQB11, and SMIQB10 or SMIQB20.

2 Principle of Operation

Minimum Equipment Configuration

For production tests on user terminals (UTs), a range of equipment is used.

In the first place, measuring instruments like generators, analyzers, voltmeters and power meters are employed, and in addition power supplies, couplers, switches, adapters and interfaces.

These components are remote-controlled by the computer running the production line. A wide range of test and management software is also required.

The description below focuses on the minimum equipment configuration of a **generator** and an **analyzer.** It will be explained how these instruments can be integrated into an existing test system and optimally controlled by the production line computer which is represented by the laptop in figure 1.

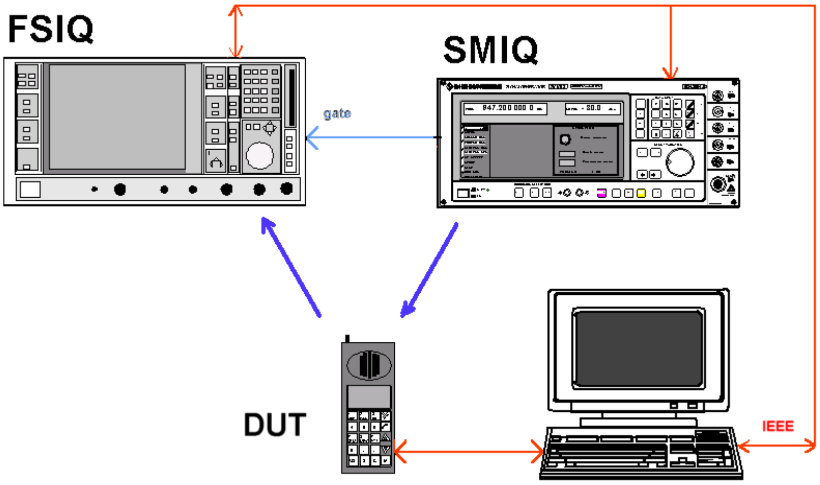


Fig. 1: Minimum equipment configuration

Signal Paths and Test Procedures

The SMIQ Signal Generator supplies a **downlink signal** to the DUT, the user terminal (UT). Depending on requirements, the downlink signal consists of QPSK or BPSK bursts with the frequencies, symbol rates and power ramps in conformance with the ICO standard. To test **multimode** UTs, SMIQ generates downlink signals in conformance with **other mobile-radio standards** (eg GSM, PDC, PHS etc).

SMIQ can further supply TTL output signals programmable to bit-accuracy as **trigger** and **gate inputs** for FSIQ or the test system.

The FSIQ Spectrum Analyzer performs all **RF measurements** on the **uplink signal** from the user terminal. For this purpose, the UT sends uplink bursts with data of its own, or provides a loopback between the downlink and the uplink.

In Vector Analyzer mode, FSIQ alone measures the **burst ramp** and calculates the **burst power** as well as all error parameters, eg **phase** and **frequency error**.

All components are controlled by the computer running the production line. The **test procedure** generally takes place in the following steps:

1. Start the SMIQ generator.
2. Switch the UT to the required test mode, for example so that the UT is able to receive the downlink signal without synchronizing and signalling channels.
3. Read data from the FSIQ Analyzer.
4. Align the UT.
5. For RF measurements, FSIQ delivers the final results. For bit error rate measurements and ALC / RSSI adjustment, results have to be determined by the UT itself or the computer of the production line.

Auxiliary Programs for Putting Instruments into Operation and for System Integration

This application note comes with PC programs SMIQ-K1 and Winbatch as well as several sample files as an aid for putting the instruments into operation and for their system integration.

Program **SMIQ-K1** generates data streams with modulation and control data on the PC and transfers the streams to the memory of SMIQ. These data streams, which are referred to as **sequences**, can very easily be edited, duplicated, etc, so that **user-defined bursts, frames** or **multiframes** are obtained (not only for ICO).

For various bursts, sample sequences (**siq files)** are supplied with this application note.

Winbatch is intended for the control of IEC/IEEE-bus devices involving a minimum of programming effort. It comes in the form of a small text editor by which the desired IEC/IEEE-bus commands can be written line by line. These **scripts** can be executed or stored as ASCII files, or such files can be loaded and the scripts executed.

For the ICO tests described here, script files (**iee files**) are supplied, which are to be used as examples.

Winbatch in addition offers simple commands for loops, branches, interrupts, and single-step operation, allowing the fast generation and optimization of **user-defined test sequences**.

SMIQ-K1 and Winbatch are very easy to operate. For a detailed description and instructions for installation please refer to the Appendix.

The two programs are used in all of the following sections:

- Install SMIQ-K1 and Winbatch now, as described in section 6.
- Make yourself familiar with SMIQ-K1 and Winbatch.

3 Generation of Downlink and Uplink Bursts

The SMIQ Signal Generator supplies a downlink signal for the RF tests carried out in production (see Fig. 1). Two sample files with burst data are supplied with this application note:

- ico_nbd.siq for the generation of nominal penetration downlink bursts
- ico_mbd.siq for the generation of medium penetration downlink bursts

In addition, two sample files with burst data for **uplink signals** are enclosed:

- ico_nbu.siq for the generation of nominal penetration uplink bursts
- ico_iab.siq for the generation of initial access bursts.

These files allow you to configure, run, and optimize the transmitter tests described in the following sections, even if **no DUT** is available for the time being.

For the **burst generation**, load the above-named burst files into the SMIQ memory using software SMIQ-K1. Proceed as follows:

1. Start the software SMIQ-K1.
2. Load the desired data sequence (for example ico_nbu.siq) with this software.
3. If necessary, modify the data as required for your application.
4. Store the modified sequence as a separate siq file on your controller.
5. Transfer the bit sequence to the SMIQ memory.
6. Close the software SMIQ-K1.

The sequence loaded can now be called by name (for example ico_nbu) on SMIQ.

The generator is configured by means of a Winbatch script:

1. Start the **Winbatch** program.
2. Load the Winbatch setting file required for generating the desired burst (for example gen_nbu.iee for ico_nbu.siq).
3. IIf necessary, modify the entries for the following parameters in the iee file as required for your application (do not change the other lines):
   - 10 MHz reference (external or internal)
     ```
     SMIQ: :rosc:sour ...
     ```
   - Frequency
     ```
     SMIQ: :freq:cw ...
     ```
     Downlink: f = 2170 MHz + N * 0.025 MHz f for channel N

     Uplink: f = 1985 MHz + N * 0.025 MHz for channel N
   - Level
     ```
     SMIQ: :pow:imm ...
     ```
4. Store the modified script as a separate iee file on your controller.

5. Run your **Winbatch script**.
   The desired signal is now available.
6. Close Winbatch.

If the bit sequence needed is already stored in the SMIQ memory, to configure SMIQ it is sufficient to run the corresponding Winbatch script file.

4 Transmitter Tests

Test Setup for Transmitter Tests

For TX measurements for board test, the DUT (the UT) is in general connected by wire. The **insertion loss a_{TX}** between the RF connector of the UT and the spectrum analyzer must be known exactly.

At the beginning of the test, the production line computer switches the UT to a **test mode** in which the UT generates uplink bursts with a predefined frequency and adjustable level. The bursts must contain a fixed bit pattern which the spectrum analyzer uses as a **SYNC WORD** to generate a bit-accurate power template.

Results are read from the FSIQ Spectrum Analyzer by the production line computer. Fig. 2 shows the signal paths active in this case.

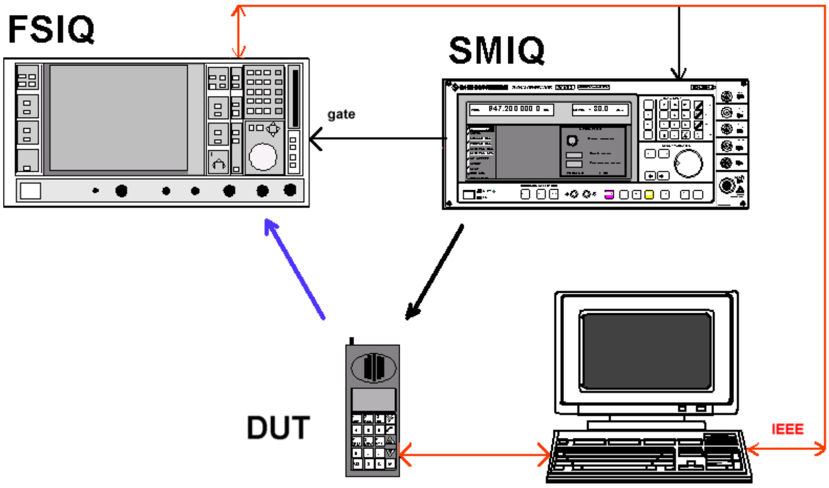


Fig. 2: Signal paths for TX measurements

For this test procedure **SMIQ is not needed**.

For some TX measurements, however, a suitable **downlink signal** may be needed from SMIQ; for example if the signal is to be returned by the UT in **RF loopback** mode. Or SMIQ is to generate a pattern to which the UT can **synchronize**. In either case, the generator is remote-controlled by the production line computer (signal selection, frequency and level settings, etc.).

To generate a downlink signal suitable for your application see section 3.

Peak- und RMS-Power, Power versus Time Measurements

If no DUT is available, use SMIQ to simulate the uplink signal to be subsequently measured. Generate a nominal uplink burst with the data sequence ico_nbu.siq, for example, as described in section 3. Connect the RF output of SMIQ to the RF input of the analyzer.

Setup of Spectrum Analyzer for Power Measurements

1. Start the **Winbatch** program on your test computer.
    - In the Winbatch program, load:
    - script file nbu_pow.iee to measure the nominal uplink burst,
    - script file iab_pow.iee to measure the initial access bursts.
2. Modify in the iee file the entries for the following parameters as required for your application (do not change the other lines):
    - 10 MHz **reference** (external or internal)

      ```
      FSIQ: sens:rosc:sour ...
      ```

    - Center **frequency** (for channel N: f = 1985 MHz + N * 0.025 MHz)

      ```
      FSIQ: freq:cent ...
      ```

    - Reference **level** (nominal level at UT port)

      ```
      FSIQ: display:trace1:y:rlevel ...
      ```

    Attention! *The reference level must in any case be programmed <u>before</u> the reference level offset!!*

    - Reference **level offset** (insertion loss $\mathbf{a}_{\mathbf{TX}}$ between UT and FSIQ)

    Attention! *The reference level must in any case be (re)programmed <u>before</u> the reference level offset!*

      ```
      FSIQ: display:trace1:y:rlevel:offset ...
      ```

    - **Sync pattern** (reference pattern for positioning in X direction)

      Currently the bit sequence 0000 1110 1110 1101 is selected as sync pattern.

      ```
      FSIQ: ddem:sear:sync:patt ...
      ```

    - **Sync offset** (position of sync pattern relative to lefthand display edge in symbol lengths)

      ```
      FSIQ: ddem:sear:sync:offset ...
      ```

Store the modified script as a separate iee file on your computer.

The resulting analyzer setup is:

- Operation as vector analyzer
- Standard settings same as for GSM, except for symbol rate: 36 ksym/s for NBU or 18 ksym/s for IAB
- Display of power ramp
- Video trigger (at 80 %)
- The section displayed in X direction is 258 symbols long (allowing the rising and the falling edge of the 240-symbol burst to be measured, too).
- The peak and the RMS power are calculated between time line 1 (TS1) and time line 2 (TS2).
- Time line 1 is positioned on symbol 2 of the useful part of the burst (corresponding to symbol 11 on the display).
- Time line 2 is positioned on symbol 237 of the useful part of the burst (corresponding to symbol 247 on the display).
- 4 tolerance lines (limit lines) are specified for measuring the burst shape.
- The frequency, level and synchronization are user-defined.

These settings have to be programmed **only once**. They can also be stored as a **memory setup** on the analyzer hard disk and loaded as required.

Starting the Power Measurements

1. Switch the UT to the **test mode** in which the desired uplink bursts are generated (with the sync pattern set on the analyzer).

   Note: *If no UT is available, simulate a nominal uplink burst on SMIQ, for example with the data sequence ico_nbu.siq.*

2. Run your Winbatch script.

Measurement is started when the spectrum analyzer detects the first burst.

Fig. 3 shows the display of results on the analyzer.

The power template for the burst ramp is defined by limit lines L01 to L04.

Note: *The limit lines are not visible in remote control mode. Switch the analyzer to LOCAL to make the lines appear.*

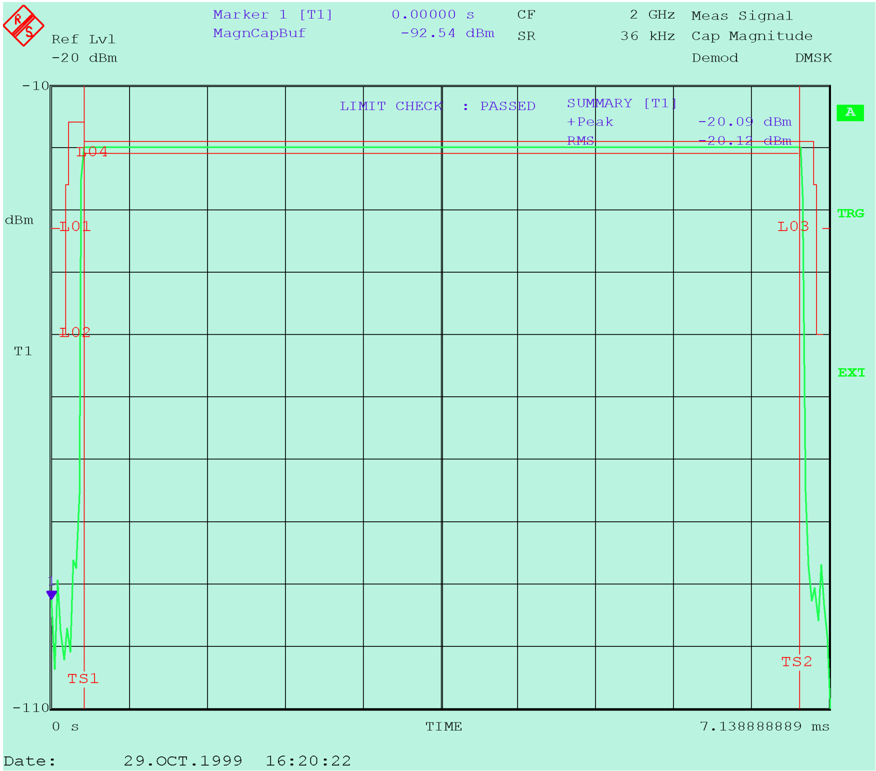


Fig. 3: Peak and RMS power, power versus time for NBU

The message **LIMIT CHECK : PASSED** indicates that the curve complies with specifications. With this test performed in the analyzer, it is not necessary to read the curve point by point into the test computer. It is sufficient to query the four limit results.

L01 and L03 are absolute limit lines for the time outside the active time slot. L02 is the upper limit line, L04 the lower limit line for the active time slot.

L02 and L04 are relative limit lines, relative to the reference value for the y-axis.

To define or check tolerance limits different from those above, specify **limit lines** of your own. Use the corresponding program lines in nbu_pow.iee or iab_pow.iee as a basis.

By entering the cable loss $\mathbf{a}_{\mathbf{TX}}$ as a reference level offset, this loss is compensated by the program. Therefore, the display shows exactly the level values at the RF port of the UT.

The **peak** and **RMS values** are shown as the results of a **summary marker** that covers the range between time lines TS1 and TS2. TS1 is positioned on symbol 2, TS2 on symbol 237 of the active part of the burst.

Note: *The summary marker can calculate further parameters, for example +peak and -peak, +/-peak / 2 and MEAN.*

Measurement of Phase and Frequency Error

If no DUT is available, simulate with SMIQ the uplink signal to be subsequently measured. To this end, generate a nominal uplink burst with the data sequence ico_nbu.siq, for example, as described in section 3. Connect the RF output of SMIQ with the RF input of the analyzer.

Configuration for Phase and Frequency Error Measurements

1. Start the **Winbatch** program on your test computer.
2. Load script file **nbu_pfe.iee** to measure the nominal uplink burst.
3. Modify in the script the entries for the following parameters as required for your application (do not change the other lines):

   - 10 MHz reference (external or internal)

     ```
     FSIQ: sens:rosc:sour ...
     ```

   - Center frequency (for channel N: f = 1985 MHz + N * 0.025 MHz)

     ```
     FSIQ: freq:cent ...
     ```

   - Reference level (nominal level at UT port)

     ```
     FSIQ: display:trace1:y:rlevel ...
     ```

   Attention! *The reference level must in any case be programmed before the reference level offset!*

   - Reference level offset (insertion loss $\mathbf{a}_{\mathbf{TX}}$ between UT and FSIQ)

     ```
     FSIQ: display:trace1:y:rlevel:offset ...
     ```

   Attention! *The reference level must in any case be (re)programmed before the reference level offset!*

   - Sync pattern (reference pattern for selecting the data stream section to be displayed)

     Currently the bit sequence 0000 1110 1110 1101 is selected as sync pattern.

     ```
     FSIQ: ddem:sear:sync:patt ...
     ```

   -

   - Sync offset (position of sync pattern relative to lefthand display edge, in symbol lengths)

     ```
     FSIQ: ddem:sear:sync:offset ...
     ```

   - No continuous measurement

     ```
     FSIQ: init:cont off
     ```

   - Averaging over 10 bursts (sweep counter setting)

     ```
     FSIQ: swe:coun 10
     ```

4. Store the modified script as a separate iee file on your test computer.

The resulting analyzer setup:

- Operation as vector analyzer
- Standard settings same as for GSM, except for: Symbol rate: 36 ksym/sec
- Display of error table and demodulated symbols
- Video trigger (at 80 %)
- Display of 258 symbols
- Error values (eg phase and frequency error) are calculated from 236 samples from the active part of the burst.
- The frequency, level and synchronization are user-defined.

These settings have to be programmed **only once**. They can also be stored as a **memory setup** on the analyzer hard disk and loaded when required.

Starting the Phase and Frequency Error Measurements

1. Switch the UT to the **test mode** in which the desired uplink bursts are generated (with the sync pattern set on the analyzer).

   Note: *If no UT is available, simulate a nominal uplink burst on SMIQ, for example with the data sequence ico_nbu.siq, as described in section 3.*

2. Run your script from Winbatch

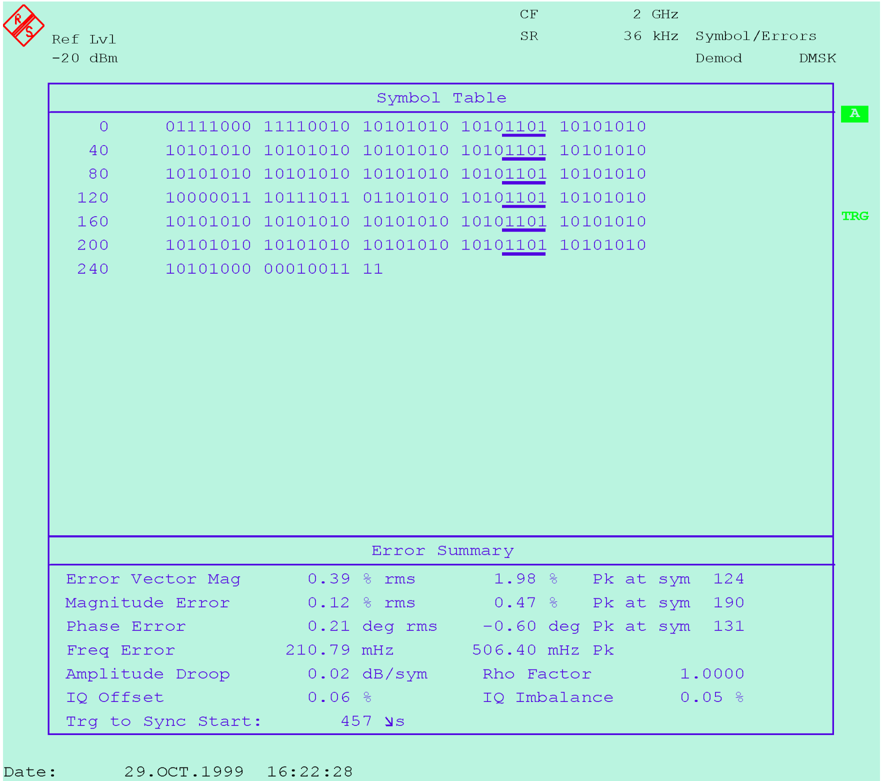
Measurement is started when the spectrum analyzer detects the first burst and stopped when the sweep counter has run to finish.

Fig. 4 shows a typical result display on the analyzer.

The upper half of the display (trace 1) shows the **symbol table** with the decoded bits of the current burst. (Each sweep covers one burst.)

The **sync pattern** is underlined. (Sync patterns with “don't care” digits can be entered from FSIQ firmware version 2.05 onwards)

The error summary in the lower half (trace 3) gives an overview of the deviations of the individual signal parameters from their ideal values. While for each burst, more than 240 bits are recorded and displayed, only the values of bit positions 2 to 237 of the useful part of the burst are used for error calculation.



```
                                         CF           2 GHz
  Ref Lvl                                SR          36 kHz  Symbol/Errors
  -20 dBm                                                    Demod       DMSK

                            Symbol Table                                      A
     0     01111000 11110010 10101010 10101101 10101010
    40     10101010 10101010 10101010 10101101 10101010
    80     10101010 10101010 10101010 10101101 10101010
   120     10000011 10111011 01101010 10101101 10101010
   160     10101010 10101010 10101010 10101101 10101010                       TRG
   200     10101010 10101010 10101010 10101101 10101010
   240     10101000 00010011 11

                            Error Summary
 Error Vector Mag        0.39 % rms        1.98 %   Pk at sym  124
 Magnitude Error         0.12 % rms        0.47 %   Pk at sym  190
 Phase Error             0.21 deg rms     -0.60 deg Pk at sym  131
 Freq Error            210.79 mHz        506.40 mHz Pk
 Amplitude Droop         0.02 dB/sym      Rho Factor         1.0000
 IQ Offset               0.06 %           IQ Imbalance       0.05 %
 Trg to Sync Start:        457 ↘s

Date:      29.OCT.1999  16:22:28
```

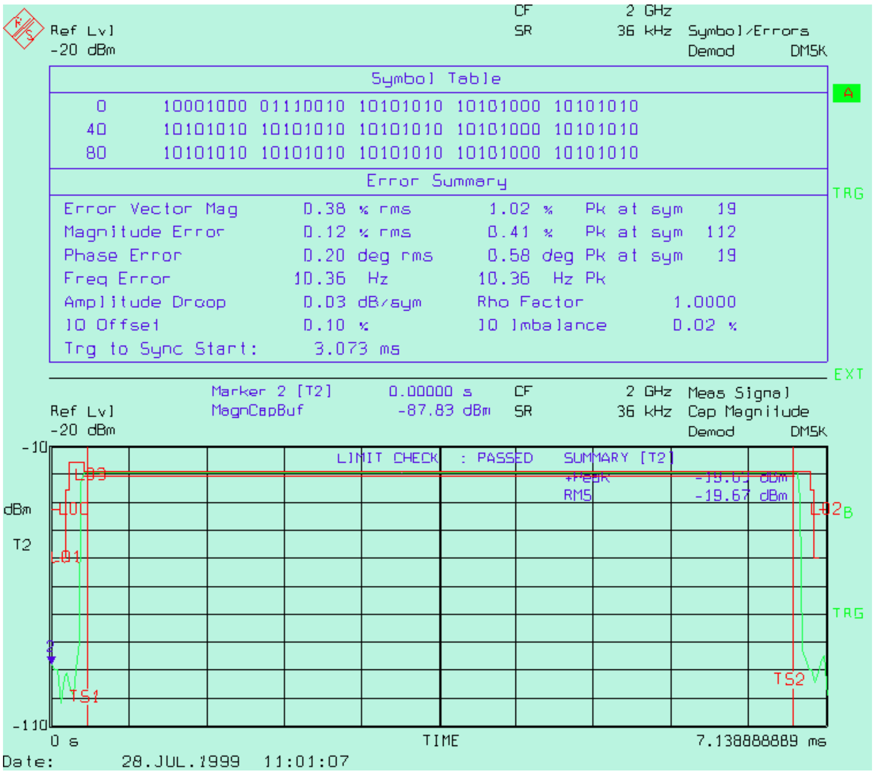
Fig. 4: Phase / frequency error for NBU

Note: *In peak value calculation, the averaging function does not yield a mean value but increases the number of bursts taken into account. The highest value of the 10 bursts evaluated is displayed.*

Combined Measurements of Uplink Signal

It is advisable to perform as many measurements as possible at the same time and analyze the maximum possible number of parameters during each recording (sweep), so as to obtain all the results needed in one go.

The following example shows how all the measurements on a UT transmitter as dealt with in the previous section can be carried out at the same time.



```
                                                   CF            2 GHz
Ref Lvl                                            SR           36 kHz  Symbol/Errors
-20 dBm                                                                 Demod       DMSK
                              Symbol Table
     0      10001000 01110010 10101010 10101000 10101010
    40      10101010 10101010 10101010 10101000 10101010
    80      10101010 10101010 10101010 10101000 10101010
                              Error Summary
 Error Vector Mag          0.38 % rms         1.02 %    Pk at sym    19
 Magnitude Error           0.12 % rms         0.41 %    Pk at sym   112
 Phase Error               0.20 deg rms       0.58 deg Pk at sym    19
 Freq Error               10.36  Hz          10.36  Hz Pk
 Amplitude Droop           0.03 dB/sym       Rho Factor          1.0000
 IQ Offset                 0.10 %            IQ Imbalance        0.02 %
 Trg to Sync Start:         3.073 ms

                 Marker 2 [T2]        0.00000 s    CF            2 GHz  Meas Signal
Ref Lvl          MagnCapBuf            -87.83 dBm  SR           36 kHz  Cap Magnitude
-20 dBm                                                                 Demod       DMSK
                         LIMIT CHECK   : PASSED    SUMMARY [T2]
                                                   +Peak              -19.63 dBm
                                                   RMS                -19.67 dBm

Date:       28.JUL.1999   11:01:07
```

Fig. 5: Combined measurement of power versus time and error summary

Settings are the same as in the previous section. For this measurement, the (rather complex) Winbatch script file **tx_all.iee** is needed.

If you require different parameter settings, modify the relevant lines of the remote control script as described in the sections above.

Note: *Measurement with the spectrum analyzer will of course be* ***quicker*** *the* ***fewer the results*** *to be calculated. If, for example, the results of the summary marker are not needed in a power-versus-time measurement, switch off the summary marker.*

Spectral Measurements

Spectral measurements in production are usually carried out with **other parameters** than those specified by the ICO standard for **conformance testing**. Measurements would otherwise take too long. For example, a minimum of 37 minutes is needed for just one of the 12 measurement series for spectrum due to modulation as specified by ICO 11.10 section 13.4.

The user should decide **prior to measurements** what limit values and test parameters (frequency band, measurement bandwidth, averaging) are suitable for the UT model under test. The Winbatch program and the remote control scripts supplied are ideal for finding out the best parameter values.

Depending on the origin of spectral components a differentiation is made between **spectrum due to modulation (dtm)** and **spectrum due to switching (dts)**. This application note comes with remote control scripts for both measurements:

- The two files dtm.iee and dts.iee contain basic structures for spectral measurements in the **zero span** mode as defined by the **ICO standard**. Results are output in the form of a list of measured values at the discrete frequencies.
- In addition, the dtm_limit.iee file is for spectral measurement with **continuous sweep** and **limit lines**. The result is in the form of a message stating whether measured values are within a predefined limits range.

Deriving a Gate Signal

To blank the spectral components originating from the pulse edges of the transmit signal, the **spectrum due to modulation** is recorded only in a small **measurement window** (in the active part of the burst). The results of the individual measurements are averaged.

To open the measurement window, the analyzer needs a **gate signal**.

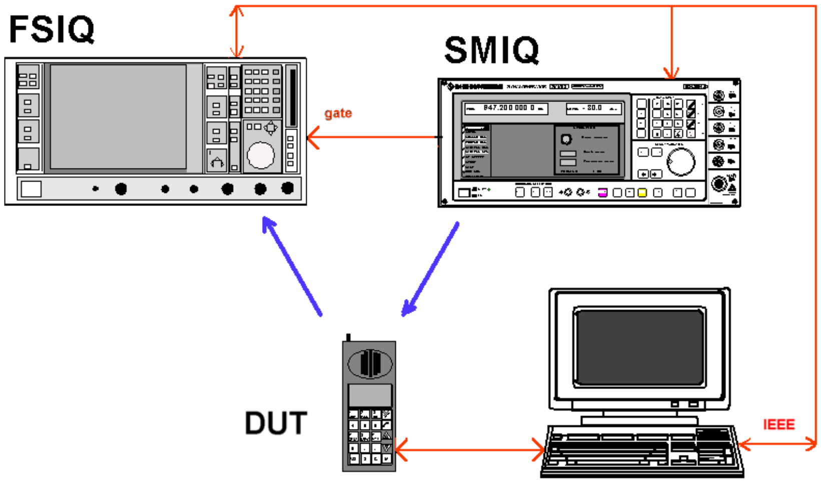


Fig. 6: Signal paths for spectral measurements

The gate signal can be generated with **SMIQ** if it supplies a downlink signal, as shown in Fig. 6. SMIQ offers an editable **control list** via which the burst envelope is controlled, for example. With this list, the level at the generator's PAR DATA TTL outputs can be controlled with bit-level accuracy (see the section "Working with SMIQ-K1" in the Appendix).

Alternatively, the output voltage of an **RF detector diode** coupled into the uplink path can for example be used as gate signal.

The position of the measurement window relative to the gate signal can be widely varied on with FSIQ.

Note: ***No gate signal*** *is required for* ***spectrum due to switching*** *measurements. A* ***continuous*** *measurement is made over several frames.*

Measurement of Spectrum Due to Modulation

The **dtm.iee** script file simulates typical conformance measurement (similar to **dts.iee**).

To **reduce measurement time** in comparison with type approval testing, use a **coarse frequency grid** (and a larger measurement bandwidth) and carry out averaging over **fewer bursts**. Results are output as **numerical values** for each frequency.

In the progam example, the numerical values are written to the **statis.dat** log file. To access this file, Winbatch must first be exited. The file can then be accessed in the current directory.

In production tests, the numerical values from the log file have to be compared with predefined **limit values**.

If no DUT is available, simulate with SMIQ the uplink signal to be subsequently measured. To this end, generate a nominal uplink burst with the data sequence ico_nbu.siq as described in section 3. Connect the RF output of SMIQ with the RF input of the analyzer. ico_nbu.siq generates a gate signal precisely in the middle of the burst. Connect pin 11 of the PAR DATA output of SMIQ (TRIGOUT 1 signal) with the EXT TRIG GATE input of FSIQ.

Configuration for Spectrum Due to Modulation

To measure the spectrum due to modulation,

1. load the **dtm.iee** file in the Winbatch program.

The settings contained in this file are default settings and have to be modified as required for your application. Default settings are as follows:

- The AVERAGE detector mode is selected. For each frequency, averaging is carried out over the selected number of bursts.
- Measurements are performed in the range ±30 kHz about a transmit frequency of 2 GHz (as example values only).
- The resolution bandwidth, the video bandwidth and the frequency steps are 3 kHz.
- A rising TTL edge in the middle of the burst is expected as a gate signal.
- Recording takes place between 50 % and 90 % of the burst.

Note: *The file dtm.iee comprises a **configuration section** with setting commands which are sent **only once**, and a looped **measurement section** which repeats to read the individual results.*

*After running your script, you can store the complete FSIQ **configuration**, which was made in the configuration section, as a **memory setup** on the hard disk of the analyzer and load it when required.*

2. Modify in the iee file the entries for the following parameters as required for your application (do not change the other lines)

- 10 MHz reference (external or internal)

```
FSIQ: sens:rosc:sour ...
```

- Start frequency

```
FSIQ: freq:cent 19...
```

- Frequency grid (step increment)

```
FSIQ: freq:cent:step ...
```

- Reference level (nominal level at UT port)

```
FSIQ: display:trace1:y:rlevel ...
```

Attention! *The reference level must in any case be programmed before the reference level offset!*

- Reference Level Offset (Dämpfung $\mathbf{a}_{TX}$ zwischen UT und FSIQ)

```
FSIQ: display:trace1:y:rlevel:offset ...
```

Attention! *The reference level must in any case be (re)programmed before the reference level offset!*

- Resolution bandwidth and video bandwidth

```
FSIQ: bwid:res ...
FSIQ: bwid:vid ...
```

- Number of bursts (sweeps) to be averaged (start with a **small number** to assess the measurement time)

```
FSIQ: swe:coun ...
```

- Position of time lines TS1 and TS2 (recording window) relative to your trigger signal

```
FSIQ: calc:tlin1 ...
FSIQ: calc:tlin2 ...
```

- Number of test runs, possibly extension by further test loops with other parameters

```
REPEAT ...
```

- **IEC/IEEE-bus timeout** of Winbatch program (Remote... menu item)

3. Store the modified script as a separate iee file on your test computer.

Starting the Spectrum Due To Modulation Measurement

1. Switch the UT to the **test mode** in which the desired uplink bursts are generated.

Note: *If no UT is available, simulate a nominal uplink burst on SMIQ, for example with the data sequence ico_nbu.siq, as described in section 3.*

2. Check whether the **gate signal** is **applied to** FSIQ at the right time. If necessary, correct the positions of time lines TS1 and TS2.
3. Start the script from the Winbatch program.
4. Close **Winbatch** after running the measurement and check results in the **statis.dat** file.

Fig. 7 shows a typical list of results for the transmit frequency 2 GHz. The lefthand column states the test frequencies in Hz, the righthand column the results in dBc.

```
1999970000,0000   -45,6473
1999973000,0000   -47,5992
1999976000,0000   -40,9662
1999979000,0000   -31,4058
1999982000,0000   -25,1870
1999985000,0000   -19,5662
1999988000,0000   -15,2833
1999991000,0000   -14,9358
1999994000,0000   -11,3755
1999997000,0000    -9,2431
2000000000,0000    -8,8395
2000003000,0000    -9,2554
2000006000,0000   -12,0885
2000009000,0000   -15,1253
2000012000,0000   -16,4845
2000015000,0000   -19,5386
2000018000,0000   -25,5354
2000021000,0000   -31,3677
2000024000,0000   -40,7593
2000027000,0000   -47,5561
2000030000,0000   -46,6644
```

Fig. 7: Results obtained for spectrum due to modulation

A comparison with the limit values shows that the above results are clearly within the ICO standard.

Measurement of Spectrum Due to Switching

The **dts.iee** script file simulates typical conformance measurement (similar to **dtm.iee**).

To **reduce measurement time** in comparison with conformance testing, use a **coarse frequency grid** (and a larger measurement bandwidth) and carry out averaging over **fewer bursts**. Results are output as **numerical values** for each frequency.

The script writes the numerical values to the log file **statis.dat**. To access this file, Winbatch must be exited first. The file can then be accessed in the current directory.

Note: *In production tests, the numerical values have to be compared with predefined **limit values**.*

If no DUT is available, simulate with SMIQ the uplink signal to be measured subsequently. To this end, generate a nominal uplink burst with the data sequence ico_nbu.siq as described in section 3. Connect the RF output of SMIQ with the RF input of the analyzer.

Configuration for Spectrum Due To Switching

1. To measure the spectrum due to switching, **load** the **dts.iee** file in the Winbatch program.

The settings contained in this file are default settings and have to be modified as required for your application. Default settings are as follows:

- Measurements are made in the analyzer mode.
- The MAX PEAK detector mode is selected. For each frequency, measurement is carried out over 10 frames (400 ms).
- Measurements are made (only) at a spacing of 98 kHz to 50 kHz from the TX frequency.
- The resolution bandwidth and the frequency steps are 3 kHz.
- The video bandwidth is 10 kHz.

Note: *The dts.iee file comprises a **configuration section** with setting commands that are sent **only once**, and a looped **measurement section** which repeats to read the individual results.*

*After running your script, you can store the complete FSIQ **configuration**, which was made in the configuration section, as a **memory setup** on the hard disk of the analyzer and load the configuration when required.*

2. Modify in the iee file the entries for the following parameters as required for your application (do not change the other lines):

   - 10 MHz reference (external or internal)

     ```
     FSIQ: sens:rosc:sour ...
     ```

   - Start frequency

     ```
     FSIQ: freq:cent 19...
     ```

   - Frequency grid (step increment)

     ```
     FSIQ: freq:cent:step ...
     ```

- Reference level (nominal level at UT port)

```
FSIQ: display:trace1:y:rlevel ...
```

Attention! *The reference level must in any case be programmed before the reference level offset*

- Reference Level Offset (insertion loss $\mathbf{a}_{TX}$ between UT and FSIQ)

```
FSIQ: display:trace1:y:rlevel:offset ...
```

Attention! *The reference level must in any case be (re)programmed before the reference level offset!*

- Resolution bandwidth and video bandwidth

```
FSIQ: bwid:res ...
FSIQ: bwid:vid ...
```

- Number of frames over which analysis is to be carried out

  SWEEP TIME = 40 ms * number of frames

```
FSIQ: swe:time ...
```

- Number of test runs, possibly extension by further test loops with other parameters

```
REPEAT ...
```

- **IEC-Bus-Timeout** of Winbatch program (Remote... menu item)

3. Store the modified script as a separate iee file on your test computer.

Starting the Spectrum Due to Switching Measurement

1. Switch the UT to the **test mode** in which the desired uplink bursts are generated.

Note: *If no UT is available, simulate a nominal uplink burst on SMIQ, for example with the data sequence ico_nbu.siq, as described in section 3.*

2. Start your script from the Winbatch program.
3. Close **Winbatch** after running the measurement and check the results in the file **statis.dat**.

Fig. 8 shows a typical list of results in the vicinity of the TX frequency (2 GHz). The lefthand column states the test frequencies in Hz, the righthand column the results in dBm.

```
1999920000,0000    -70,7174
1999923000,0000    -69,2165
1999926000,0000    -69,7799
1999929000,0000    -68,8809
1999932000,0000    -67,9271
1999935000,0000    -67,0018
1999938000,0000    -67,7644
1999941000,0000    -68,3522
1999944000,0000    -67,5732
1999947000,0000    -64,5796
1999950000,0000    -63,1620

2000050000,0000    -65,2629
2000053000,0000    -64,8053
2000056000,0000    -66,3632
2000059000,0000    -67,5143
2000062000,0000    -68,1325
2000065000,0000    -66,6886
2000068000,0000    -68,2017
2000071000,0000    -68,4274
2000074000,0000    -69,9304
2000077000,0000    -69,8348
2000080000,0000    -69,7717
```

Fig. 8: Results obtained for spectrum due to switching

Results are just within the permissible limit values.

Measurements with Continuous Sweep and Limit Lines

The dtm_limit.iee file is for spectral measurement with **continuous sweep** and **limit lines**. The result is in the form of a message stating whether measured values for Frequency are within a predefined limits range.

If no DUT is available,

1. simulate with SMIQ the uplink signal to be measured subsequently. To this end, generate a nominal uplink burst with the data sequence ico_nbu.siq as described in section 3.
2. Connect the RF output of SMIQ with the RF input of the analyzer. ico_nbu.siq generates a gate signal precisely in the middle of the burst.
3. Connect pin 11 of the PAR DATA output of SMIQ (TRIGOUT 1 signal) with the EXT TRIG GATE input of FSIQ.

Configuration for Measurement with Continuous Sweep

To measure the spectrum due to modulation:

1. load the **dtm_limit.iee** file in the Winbatch program.

In the measurement described here, the **limit values for spectrum due to modulation** as defined by ICO are used, and the signal is recorded in a **measurement window** of 50 % to 90 % of the burst, likewise in accordance with ICO. Averaging is in this case not over time but over several sweeps.

The following configuration is programmed for this measurement:

- AVERAGE is selected; averaging is over 10 sweeps.
- The measurement is performed in the range ±150 kHz about a TX frequency of 2 GHz.
- The resolution and the video bandwidth are 3 kHz.
- The analyzer is operated in gate mode.
- A rising TTL edge in the middle of the burst is used as a gate signal.
- Recording is between 50 % and 90 % of the burst.

Note: *The dtm_limit.iee file comprises a **configuration section** with setting commands that are sent **only once**, and a looped **measurement section** which repeats to read the individual results.*

*After running your script, you can store the complete FSIQ **configuration**, which was made in the configuration section, as a **memory setup** on the hard disk of the analyzer and load the configuration when required.*

2. Modify in the iee file the entries for the following parameters as required for your application (do not change the other lines):

   - 10 MHz reference (external or internal)

     ```
     FSIQ: sens:rosc:sour ...
     ```

   - Center frequency

     ```
     FSIQ: freq:cent 19...
     ```

   - Reference level (nominal level at UT port)

     ```
     FSIQ: display:trace1:y:rlevel ...
     ```

Attention! *The reference level must in any case be programmed before the reference level offset!*

- Reference level offset (insertion loss $\mathbf{a}_{\mathbf{TX}}$ between UT and FSIQ)

  ```
  FSIQ: display:trace1:y:rlevel:offset ...
  ```

Attention! *The reference level must in any case be (re)programmed before the reference level offset!*

- Frequency span

  ```
  FSIQ: freq:span ...
  ```

- Resolution bandwidth and video bandwidth

  ```
  FSIQ: bwid:res ...
  FSIQ: bwid:vid ...
  ```

- Gate length

  ```
  FSIQ: swe:egat:length
  ```

- Number of sweeps to be averaged (start with a **small number** to assess the measurement time)

  ```
  FSIQ: swe:coun ...
  ```

- Number of test runs, possibly extension by further test loops with other parameters

  ```
  REPEAT ...
  ```

3. Store the modified script as a separate iee file on your test computer.

Starting the Measurement with Continuous Sweep

1. Switch the UT to the **test mode** in which the desired uplink bursts are generated.

Note: *If no UT is available,* ***simulate*** *a nominal uplink burst with SMIQ, for example with the data sequence ico_nbu.siq, as described in section 3. As modulation is the same for every burst, a rather uniform spectrum will be obtained.*

2. Check whether the **gate signal** is **applied to** FSIQ at the right time. If necessary, correct the gate delay and the gate length.

3. Start your script from the Winbatch program.

Fig. 9 shows a typical result display.

Note: The limit lines are not visible in remote control mode. Switch the analyzer to LOCAL to make the lines appear.

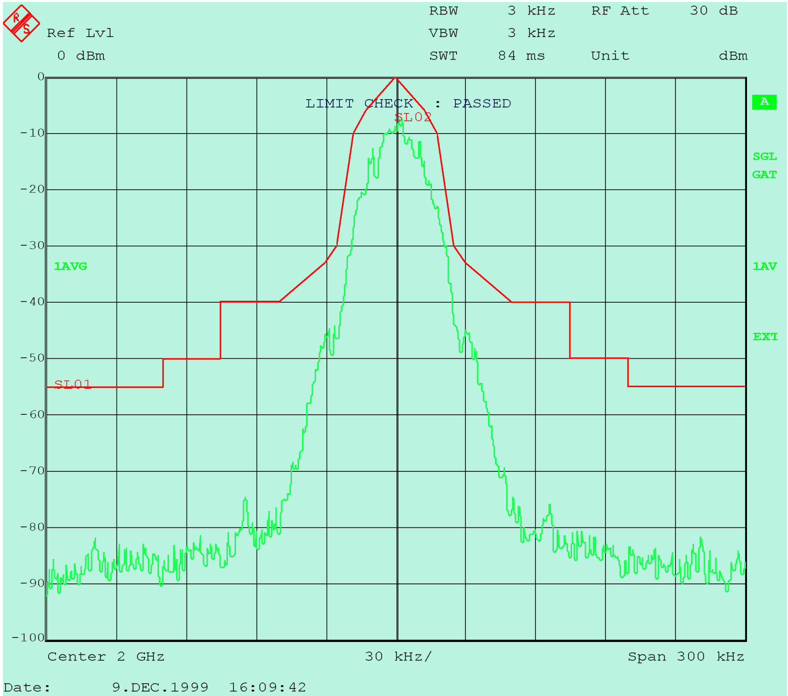


Fig. 9: Measurement result with continuous sweep and limit lines

The message **LIMIT CHECK : PASSED** indicates that the curve complies with the specifications. With this test performed in the analyzer, it is not necessary to read the curve point by point into the test computer. It is sufficient to query the two limit results.

5 Receiver Tests

Adjustment of UT Automatic Level Control

Adjustment of automatic level control (**ALC**), or received-signal strength indication (**RSSI**), is effected by means of a program loop run by the production line computer. Controlled by this loop, a generator applies a standard-conforming downlink signal with a precisely defined RF level to the RX input of the UT. The ALC voltage is measured by a tester and the UT characteristic corrected.

SMIQ supplies a highly accurate test signal for this measurement. To compensate the loss of the UT RX path:

1. determine the insertion loss of the downlink path from SMIQ to the RF port of the UT
2. enter this value later as **SMIQ level offset**.

Sample Program

The mechanical attenuator of SMIQ is subject to wear in continuous operation. For applications in production, which involve frequent level changes to SMIQ, **electronic level setting** should therefore be used.

Electronic level setting operates in a range of about 90 dB without switching the mechanical attenuator.

The **smiq_lev.iee** file contains a sample programming loop for level setting. Later in production testing, the trimming routines of the production tester for the respective levels are executed instead of the 'WAITKEY' Winbatch command.

For executing the Winbatch script, a bit sequence with the name ico_nbd.siq must be stored in the SMIQ memory. If this is not the case, proceed as follows:

1. Load the bit sequence **ico_nbd.siq** with software SMIQ-K1.
2. If necessary, modify the data as required for your application.
3. Store the modified sequence as a separate siq file on your controller.
4. Transfer the bit sequence to the SMIQ memory.
5. Load the **smiq_lev.iee** script file in the Winbatch program (File, Open).
6. Modify by means of the editor the entries for the following parameters as required for your application (do not change the other lines):

   - 10 MHz reference (external or internal)

     ```
     SMIQ: :rosc:sour ...
     ```

   - Frequency (for channel N: f = 2170 MHz + N * 0.025 MHz)

     ```
     SMIQ: :freq:cw ...
     ```

   - Level offset (insertion loss $\mathbf{a}_{RX}$ between SMIQ and UT.
     Enter this value as a **negative** number on SMIQ.)

     ```
     SMIQ: :sour:pow:offs ...
     ```

   - Level steps

     ```
     SMIQ: :sour:pow:step:incr ...
     ```

   - Level (nominal level at UT port)

```
SMIQ: :pow:imm ...
```

7. Store the modified script as a separate iee file on your controller.
8. Start the **Winbatch** program.

Notes: *In the smiq_lev.iee program, the SMIQ level control loop is briefly closed prior to each run of the setting loop (command :sour:pow:alc:sear once). This increase level accuracy while taking only a very short time (approx. 20 ms).*

Program the level after level offset programming.

The Winbatch program does not use any nested loops.

Measurement of Bit Error Rate

For bit error measurements, SMIQ generates a downlink signal with a **highly accurate level** and modulated with a **suitable data pattern**.

The test procedure employed for bit error rate measurement depends on the complexity of the measurement and on the **test modes** offered by the UT. The test mode determines the test method, and also the configuration of the measurement loop.

Generally, two types of measurement loop are possible:

- The UT returns the **decoded bits** to the production line computer. The computer evaluates the bits. Fig. 10 shows the active signal paths for this method.
- The UT performs an internal **loopback** and returns the decoded bits as an **uplink signal**. The bits are **demodulated by FSIQ**, and read and evaluated by the production line computer. Fig. 11 shows the active signal paths for this method.

Both methods are discussed in the following sections.

UT Returns the Decoded Bits

Fig. 10 shows the signal paths active with the UT returning the **decoded bits** to the production line computer.

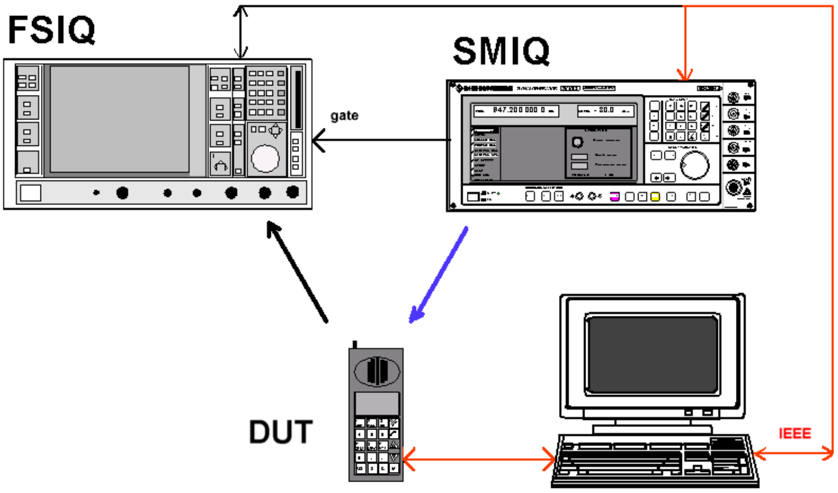


Fig. 10: Signal paths for bit error measurement with decoded bits returned

In the above test method, SMIQ performs a task similar to that in ALC adjustment of the UT. For this reason, no separate sample script is included here.

For bit error rate measurements the insertion loss of the UT RX path has to be compensated very accurately.

1. Determine the **insertion loss a_{RX}** of the downlink path from SMIQ to the RF port of the UT.

2. Use this value later as **SMIQ level offset**.

It is best to modulate the downlink signal with a **constant pattern**, which is stored as a **data list** in the SMIQ memory. This pattern is also included in the test program of the production line computer. The computer compares the pattern with the incoming data and determines the error rate.

Use the software **SMIQ-K1** to generate your bit pattern.

Some UTs may offer a **test mode** in which the UT performs bit error rate measurement **internally and independently** for a defined expected input pattern. In this case, only the **result is returned** to the production line computer. This is the simplest and fastest method.

UT Returns Bits in RF Loopback Mode

Fig. 11 shows the signal paths active with the UT returning bits in the RF loopback mode.

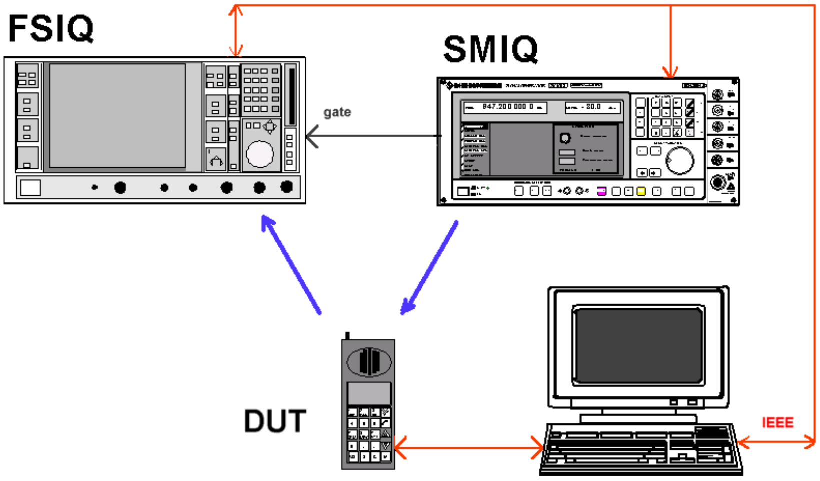


Fig. 11: Signal paths for bit error measurement with RF loopback

For bit error rate measurements the insertion loss of the UT RX path has to be compensated very accurately.

1. Determine the **insertion loss $\mathbf{a_{RX}}$** of the downlink path from SMIQ to the RF port of the UT.
2. Use this value later as **SMIQ level offset**.

<u>Sample program:</u>

The **berr.iee** script includes the programming elements needed for bit error rate measurement with RF loopback.

The command lines of this script are default lines only and have to be modified and extended as required for your application.

The commands cover **control of the UT** by means of a periodic downlink signal from SMIQ, as well as the complete **configuration** of FSIQ for the **recording, demodulation** and **output** of the demodulated uplink bits.

Since Winbatch is not capable of realtime computing, **neither comparison with the original bits** nor error rate calculation are carried out here.

If <u>no</u> DUT is available, use SMIQ to simulate the uplink signal to be subsequently analyzed. To this end, generate a nominal uplink burst with the data sequence ico_nbu.siq as described in section 3. Connect the RF output of SMIQ with the RF input of the analyzer.

If a DUT is available, proceed as follows:

1. Load the bit sequence **ico_nbd.siq** with software SMIQ-K1.
2. If necessary, modify the data as required for your application.
3. Store the modified sequence as a separate .siq file on your controller.
4. Transfer the bit sequence to the SMIQ memory.
5. Load the **berr.iee** file intp the Winbatch program.

The following configuration is preprogrammed:

- The generator modulates a constant pattern on each burst.
- The generator uses electronic level setting (as in ALC adjustment).
- The downlink level is set starting from 0 dBm.
- The meas_loop program loop is run ten times. For each run, one burst is recorded. The burst bits are demodulated by FSIQ and transferred to the Winbatch program, and the SMIQ level is lowered by 1 dB for the next loop sequence.

6. Modify in the iee file the entries for the following parameters as required for your application (do not change the other lines):

   - 10 MHz reference (external or internal)

     ```
     SMIQ: :rosc:sour ...
     FSIQ: sens:rosc:sour ...
     ```

   - Downlink frequency (for channel N: f = 2170 MHz + N * 0.025 MHz)

     ```
     SMIQ: :freq:cw ...
     ```

   - Uplink frequency (for channel N: f = 1985 MHz + N * 0.025 MHz)

     ```
     FSIQ: freq:cent
     ```

   - Level offset (insertion loss **a_{RX}** between SMIQ and UT.
     Enter this value as a **negative** number on SMIQ.)

     ```
     SMIQ: :sour:pow:offs -...
     ```

   - Number and step size of level steps, etc

     ```
     SMIQ: :sour:pow:step:incr
     ```

   - Level (nominal level at UT port) at which the test is to start

     ```
     SMIQ: :pow:imm ...
     ```

   If you use SMIQ and the ico_nbu.siq data sequence for simulating the UT signal, enter the pattern name ico_nbu and change the following lines:

   ```
   SMIQ: :dm:dlis:sel "ico_nbu"
   SMIQ: :dm:clis:sel "ico_nbu"
   SMIQ: :dm:stan icog
   ```

   Add the following lines:

   ```
   SMIQ: :dm:filt:par 0.3
   SMIQ: :dm:cod gsm
   SMIQ: :dm:srat 36000
   ```

7. If necessary, modify the **loop parameters.**
8. Store the modified script as a separate iee file on your controller.
9. Switch the UT to the **test mode** in which RF loop is closed.
10. Start the **Winbatch** program.

Notes: *In the berr.iee sample program, the demodulated bits are transferred from FSIQ to the PC in **ASCII format** since only this format can be read by Winbatch. For speedier transfer later in the* *actual application, the bits should be read out in the **UINT** (unsigned integer) format.*

*The errors displayed in the **error summary** window are **not valid** if the analyzer is configured with berr.iee. This is because – if not explicitly programmed – the **complete contents of the** (analyzer) **display window** are used as the basis for calculation. For the result length defined in this case, this includes the sections ahead of and after the useful part of the burst, in which there is no RF signal.*

*To obtain a valid frequency and phase error display at the same time, it is necessary to define by means of **time lines** a recording window (as described in section 4) which contains only valid bits.*

*In the berr.iee sample program, the SMIQ **level control loop** is briefly closed prior to each run of the measurement loop (command :sour:pow:alc:sear once). This increases level accuracy while taking only a very short time (approx. 20 ms).*

*Program the **level** on SMIQ after **level offset** programming.*

*Although no curve display on FSIQ is selected in this sample program, **video triggering** is still possible. FSIQ internally evaluates the amplitude information even if purely numerical representation is selected.*

*The Winbatch program does not use any **nested loops**.*

Simulation of IQ Impairments in the Downlink Signal

Defined IQ impairments can be simulated in the downlink signal by means of the signal generator.

The following parameters can be set:

- **leakage** (amplitude of residual carrier)
  `SMIQ: :dm:leak ...`
- **imbalance** (different gain of I and Q stages)
  `SMIQ: :dm:iqr ...`
- **quadrature offset**
  `SMIQ: :dm:quad:angl ...`

➢ Activate the IQ impairments with:
`SMIQ: :dm:iq:imp on`

With local control of SMIQ, these parameters can be set under the VECTOR MODULATION menu item. Please note that these settings are valid for both the analog and digital modulation modes.

Note: *Quadrature offset is given in degrees with SMIQ, and in percent with FSIQ. Please take into account that different units are used.*

6 Appendix

Working with Winbatch

The **Winbatch** program is for the control of IEC/IEEE-bus devices and is very easy to program. It comes in the form of a small text editor by which the desired IEC/IEEE-bus commands can be written line by line. The scripts thus obtained can be run or stored as ASCII files, or such files can be loaded and the stored scripts run.

Requirements:

To run Winbatch, GPIB-bus card PS-B4 from Rohde & Schwarz or an equivalent GPIB-bus card from National Instruments is required. The appropriate driver software should be installed on your computer.

Installing the Winbatch Software

Winbatch is attached to this application note as a self-expanding file (wbatch.exe). It is suitable for Windows 95, Windows 98 or Windows NT. Install the software as follows:

1. Copy **wbatch.exe** to a directory of your choice.
2. Start **wbatch.exe**.

Among the files expanded is the setup.exe installation program.

3. Start setup.exe and proceed in accordance with the dialog

First Steps with Winbatch

Start Winbatch from your Windows user interface.

The Winbatch menu bar is displayed with an empty workspace.

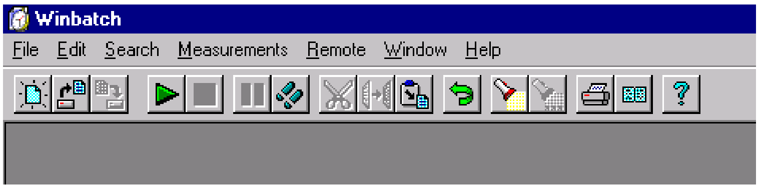


Fig. 12: Winbatch menu bar

Winbatch is **file-oriented** like a standard editor. The pull-down menus under 'File', 'Edit' and 'Search' therefore resemble the pull-down menus of other editors.

In addition, you find under 'Measurements' the items **'Start'**, **'Stop'**, **'Pause'** and **'Step'**, which allow **line-by-line** processing of loaded or newly configured files.

Moreover, there is the 'Remote' menu item for the configuration of remote-control parameters.

The main control functions can be directly accessed via icons in the **toolbar**. If the mouse pointer dwells on an icon, a help text describing the function of the icon is inserted.

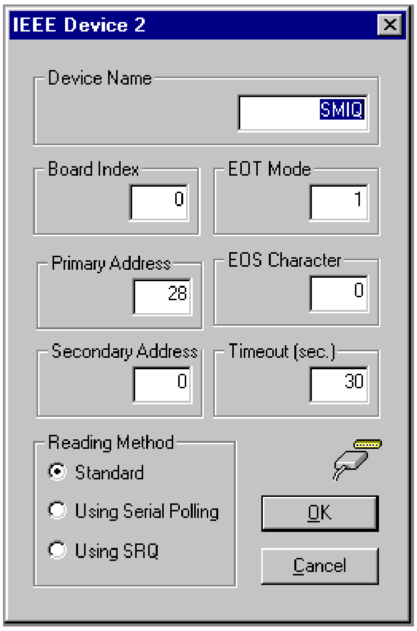
Under the ’**Help**’ menu item, you find explanations for **all control elements and commands**.

Setting Remote-Control Parameters in Winbatch

Before running Winbatch for **the first time**, the **remote-control parameters** for all devices to be controlled by Winbatch have to be specified.

1. Click on ’Remote’ in the menu bar
2. Click on an IEEE 488 device.

The parameter window appears:



IEEE Device 2

Device Name: SMIQ

Board Index: 0

EOT Mode: 1

Primary Address: 28

EOS Character: 0

Secondary Address: 0

Timeout (sec.): 30

Reading Method:
● Standard
○ Using Serial Polling
○ Using SRQ

Fig. 13: Winbatch window for remote-control parameters

The **device name** to be entered here is the logical name under which the device will be addressed later.

If you work with the supplied **sample programs**, enter the device name ’**SMIQ**’ for the generator and ’**FSIQ**’ for the analyzer.

11.Specify all parameters in the parameter window.

12.Close the parameter window with ’OK’.

13.Repeat these steps for each device connected.

Note: *Analyzer measurements involving averaging over many bursts take a relatively long time. This should be taken into account when defining the timeout parameter.*

Structure of Winbatch Files

A Winbatch file contains lines with ASCII text. The types of line are:

- blank lines
- comment lines
- command lines.

Winbatch files have the extension ’**iee**’.

Comment lines start with ’;’ or ’REM’.

Note: *Do not write comments in command lines. The character ’;’ is a valid element of the IEC/IEEE-bus syntax, so the comments would be sent too.*

Command lines contain:

- IEC/IEEE-bus commands
- labels
- execution commands.

IEC/IEEE-bus commands have the following general form:

```
Device name: <repetition factor> <command>
```

or

```
Device name: <Command>
```

Example:

```
SMIQ: *RST
SMIQ: :POW:IMM 0.5dbm
```

A **question mark** at the end of the command causes the computer to read a reply after the command is sent, for example:

```
SMIQ: *IDN?
```

Labels as branch destinations are placed in brackets, for example:

```
[label_0]
```

Execution commands are, for example:

```
"GOTO","JUMP","RETURN","REPEAT","PAUSE","WAITKEY"
```

possibly with **conditions** like `"IF","WHILE".`

In addition, there are **reports** `"REPORT ON / OFF","FPRINT", "APPENDDATA"`

and much more.

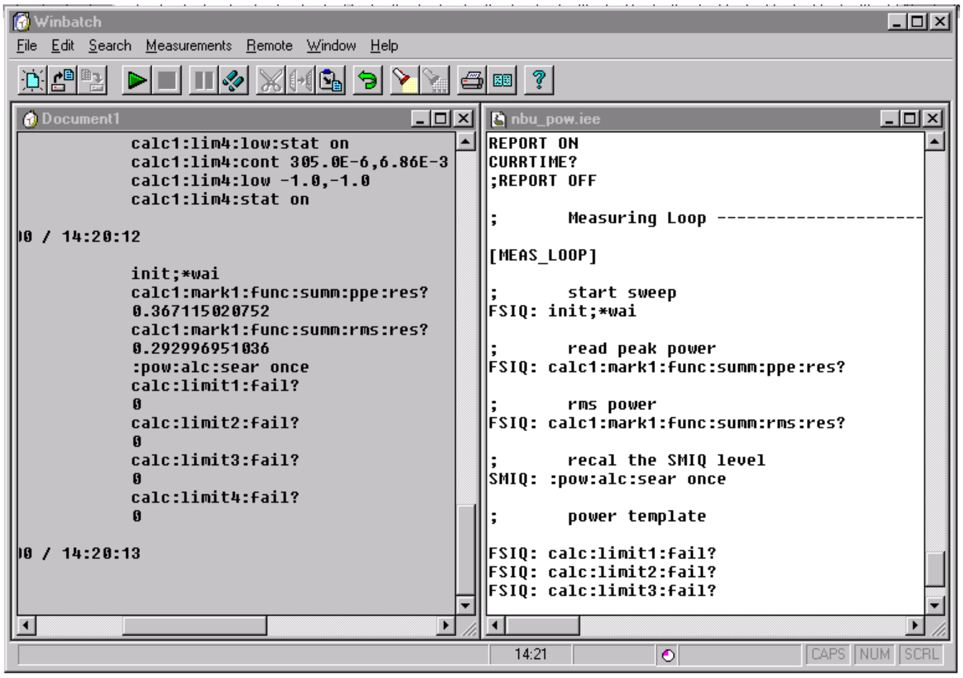
An overview of **all language elements** is given under the ’Help’, ’Contents’, ’Remote Scripts’ menu items.

Running Winbatch Files

1. Load one of the supplied sample files or write a few command lines.
2. Connect your measuring instruments to the IEC/IEEE bus of your computer and switch the equipment on.
3. Start program execution with 'Measurement', 'Start' or by clicking on the icon ▶ in the toolbar.

Winbatch opens a new window which documents the traffic on the IEC/IEEE bus. You can **follow** the line-by-line processing in the program window, too. You can **interrupt**, **resume**, and **stop** program execution or advance in the **single-step** mode.

Fig. 14 shows a section of a small test program:



```
          calc1:lim4:low:stat on
          calc1:lim4:cont 305.0E-6,6.86E-3
          calc1:lim4:low -1.0,-1.0
          calc1:lim4:stat on

)0 / 14:20:12

          init;*wai
          calc1:mark1:func:summ:ppe:res?
          0.367115020752
          calc1:mark1:func:summ:rms:res?
          0.292996951036
          :pow:alc:sear once
          calc:limit1:fail?
          0
          calc:limit2:fail?
          0
          calc:limit3:fail?
          0
          calc:limit4:fail?
          0

)0 / 14:20:13
```

```
REPORT ON
CURRTIME?
;REPORT OFF

;       Measuring Loop --------------------

[MEAS_LOOP]

;       start sweep
FSIQ: init;*wai

;       read peak power
FSIQ: calc1:mark1:func:summ:ppe:res?

;       rms power
FSIQ: calc1:mark1:func:summ:rms:res?

;       recal the SMIQ level
SMIQ: :pow:alc:sear once

;       power template

FSIQ: calc:limit1:fail?
FSIQ: calc:limit2:fail?
FSIQ: calc:limit3:fail?
```

Fig. 14: Winbatch screen

The program extract shown is part of a measurement loop. On the right, the text of the command file is displayed, on the left, the log of the IEC/IEEE-bus traffic, as well as the measurement results read from the analyzer.

Special Winbatch features:

Winbatch can generate logs including the **time of day**. To this effect, enter the command `'CURRTIME?'`.

To increase **running speed**, logging in the documentation window can be suppressed by means of the command `'REPORT OFF'`**.**

In this way, measurement sequences can be **time-optimized**.

Notes: *Winbatch scripts* ***must not include any nested loops****. Instead, use 'COUNTER' and conditional jumps, and make use of the repetition factor and the 'REPEAT' command.*
Results are written to the ***statis.dat*** *file by means of the command* *`'APPENDDATA'`**. This file can be accessed only upon*

exiting Winbatch. The file name 'statis.dat' cannot be changed by the user.

Winbatch Sample Files

This application note comes with the following Winbatch script files:

gen_nbd.iee For generating nominal penetration downlink bursts. For this, the ico_nbd.siq sequence must be stored in the SMIQ memory.

gen_mdb.iee Generates Medium Penetration Downlink-Bursts. The sequence ico_mbd.siq must be loaded in SMIQ memory

gen_nbu.iee For generating medium penetration downlink bursts. For this, the ico_mbd.siq sequence must be stored in the SMIQ memory.

gen_iab.iee For generating initial access bursts. For this, the ico_iab.siq sequence must be stored in the SMIQ memory.

nbu_pow.iee For measuring peak and RMS power, power versus time of ICO nominal penetration uplink bursts.

iab_pow.iee For measuring peak and RMS power, power versus time of ICO initial access bursts.

nbu_pfe.iee For measuring phase and frequency errors of ICO nominal penetration uplink bursts.

tx_all.iee For combined measurements of ICO nominal penetration uplink bursts.

dtm.iee For measuring spectrum due to modulation (zero span).

dts.iee For measuring spectrum due to switching (zero span).

dtm_limit.iee For spectral measurements with continuous sweep and limit lines.

smiq_lev.iee For level setting of SMIQ.

berr.iee Sample script for bit error measurements.

Working with SMIQ-K1

With the SMIQ-K1 software package, modulation and control data for the generators SME and SMIQ can be generated on a PC. Existing data records can be loaded, edited, stored and transferred to the SMIQ memory. Conversely, data can be read from the generator and stored in files.

This makes it possible to generate **user-defined bursts, frames** and **multiframes,** not only to the ICO standard.

Requirements:

To run SMIQ-K1, GPIB-bus card PS-B4 from Rohde & Schwarz or an equivalent GPIB-bus cart from National Instruments is required. The appropriate driver software must be installed on your computer.

Note: *If working under 32-bit Windows:*

Once installation is complete, you must activate the 16-bit DOS driver :

Windows ’95, Windows ’98

1. Open the *Device Manager* (Control Panel > System > Device Manager)
2. Open "National Instruments GPIB Interfaces"
3. Click on "*Properties*"
4. Activate "Enable Support for DOS NI-488.2 Applications"

Windows NT

Edit "config.nt"; remove the comment syntax rem from the entry:

```
rem device=D:\APP\NATION~1\NI-488.2\DosWin16\Gpib-nt.com
```

Remove any remaining entry for the 16-bit driver "gpib.com" from the system file "config.sys"

Installing SMIQ-K1 Software

SMIQ-K1 is attached to this application note as a self-expanding file (smiq240.exe). SMIQ-K1 is actually a DOS program. It therefore operates with 16-bit and 32-bit Windows versions. Install the software as follows:

1. Copy smiq240.exe to a directory of your choice.
2. Start **smiq240.exe** and proceed in accordance with the dialog.

Among the files expanded is the smiq_k1.pdf **manual**.

User Interface of SMIQ-K1 Software

After starting SMIQ-K1, the software first attempts to address the generator via the IEC/IEEE bus.

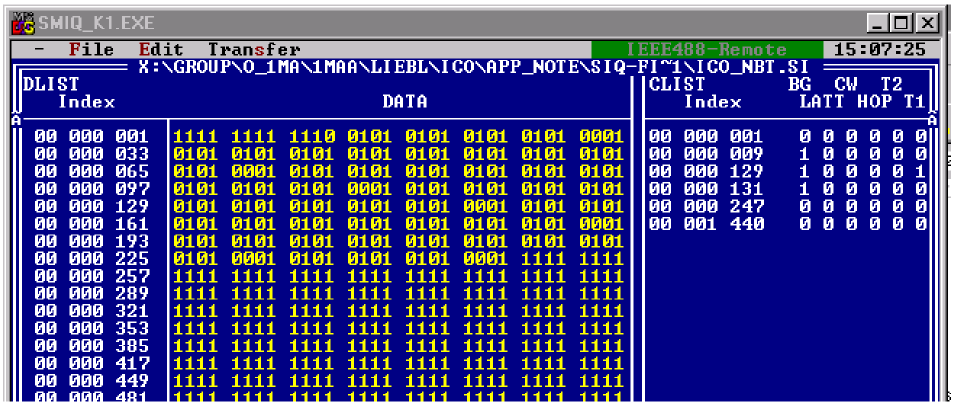
The **device address 28** is used as a default address after installation. Even if communication does not function properly, for example because a different address is set on SMIQ, the SMIQ-K1 user interface will appear, and data files (sequences) can be loaded or new files generated.

The **device address** can be **redefined later** when data is transferred to SMIQ. (The 'Transmit' menu item under 'Transfer' is accessible only if data has been loaded or edited.) The new address is used as the default address the next time SMIQ-K1 is started.

SMIQ-K1 files for SMIQ have the **extension '.siq'** (**s**equence for sm**iq**).

Load one of the supplied burst files with 'File', 'Load Sequence', or generate a new file with 'File', New Sequence'.

After loading a stored file or generating a new sequence file, two windows are displayed on the workspace (Fig. 15):



```
X:\GROUP\O_1MA\1MAA\LIEBL\ICO\APP_NOTE\SIQ-FI~1\ICO_NBT.SI
DLIST                                                          CLIST         BG   CW   T2
   Index                       DATA                               Index      LATT HOP T1

  00 000 001  1111 1111 1110 0101 0101 0101 0101 0001   00 000 001    0 0 0 0 0 0
  00 000 033  0101 0101 0101 0101 0101 0101 0101 0101   00 000 009    1 0 0 0 0 0
  00 000 065  0101 0001 0101 0101 0101 0101 0101 0101   00 000 129    1 0 0 0 0 1
  00 000 097  0101 0101 0101 0001 0101 0101 0101 0101   00 000 131    1 0 0 0 0 0
  00 000 129  0101 0101 0101 0101 0101 0001 0101 0101   00 000 247    0 0 0 0 0 0
  00 000 161  0101 0101 0101 0101 0101 0101 0101 0001   00 001 440    0 0 0 0 0 0
  00 000 193  0101 0101 0101 0101 0101 0101 0101 0101
  00 000 225  0101 0001 0101 0101 0101 0001 1111 1111
  00 000 257  1111 1111 1111 1111 1111 1111 1111 1111
  00 000 289  1111 1111 1111 1111 1111 1111 1111 1111
  00 000 321  1111 1111 1111 1111 1111 1111 1111 1111
  00 000 353  1111 1111 1111 1111 1111 1111 1111 1111
  00 000 385  1111 1111 1111 1111 1111 1111 1111 1111
  00 000 417  1111 1111 1111 1111 1111 1111 1111 1111
  00 000 449  1111 1111 1111 1111 1111 1111 1111 1111
  00 000 481  1111 1111 1111 1111 1111 1111 1111 1111
```

Fig. 15: SMIQ-K1 workspace

The **DLIST**-window contains data bits, the **CLIST**-window control bits.

The DLIST is **organized in bits**. The index counts the bit positions.

The CLIST is **organized in symbols** (the index counts symbol positions). The CLIST contains status **changes**.

The **BG** (burst gate) and **LATT** (level attenuation) input fields of the CLIST are for **envelope control**. These input fields can be used for generating the **burst ramp**.

The **T1** and **T2** input fields of the CLIST are for level control at the **TRIG1** and **TRIG2** pins (on the PAR DATA connector of SMIQ). These input fields can be used for generating a **gate** or **trigger signal** for **spectral measurements**.

1. Edit the DLIST and the CLIST.
2. Store the resulting sequence as a separate .siq file.
3. Transfer the data to SMIQ.

Notes: *When a new sequence is generated, the DLIST is built up first. In the CLIST window, only the text 'no list' is displayed. To generate the first entry, click into the CLIST window, then select 'Edit' and 'Append...'.*

If 'DATA LIST' is selected on SMIQ as modulation data source, the control signals must be output from the 'CONTROL LIST'. Envelope control by external signals is in this case not possible.

SMIQ-K1 Sample Files

This application note comes with the following four sample files with burst data:

- ico_nbd.siq For generating nominal penetration downlink bursts.
- ico_mbd.siq For generating medium penetration downlink bursts.
- ico_nbu.siq For generating nominal penetration uplink bursts.
- ico_iab.siq For generating initial access bursts

1. Load the desired file with the SMIQ-K1 software,
2. modify the data if necessary and
3. transfer the sequence to SMIQ.

The Winbatch gen_nbd.iee, gen_mbd.iee, etc files activate the corresponding file in SMIQ.

7 Ordering Information

Vector Signal Generator SMIQ:

SMIQ02B	300 kHz to 2.2 GHz	1125.5555.02
SMIQ03B	300 kHz to 3.3 GHz	1125.5555.03
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

Spectrum Analyzer FSE:

FSEA30	20 Hz to 3.5 GHz	1065.6000.35
FSEB30	20 Hz to 7 GHz	1066.3010.35
FSEM30	20 Hz to 26.5 GHz	1079.8500.35

Options:

FSE-B7	Vector Signal Analyzer	1066.4317.02

Spectrum Analyzer FSIQ:

FSIQ3	20 Hz to 3.5 GHz	1119.5005.03
FSIQ7	20 Hz to 7 GHz	1119.5005.07
FSIQ26	20 Hz to 26 GHz	1119.6001.26

IEC/IEEE-Bus Card

PS-B4	IEEE 488.2-Bus Interface	1006.6207.04

ROHDE & SCHWARZ

ROHDE & SCHWARZ GmbH & Co. KG · Mühldorfstraße 15 · D-81671 München
P.O.B 80 14 69 · D-81614 München · Telephone +41 89 4129 -0 · Fax +41 89 4129 - 3777 · Internet: www.rohde-schwarz.com