

## Supplement to the R&S UPL Manual

The R&S UPL software version 3.02/3.03 has been upgraded to 3.04/3.05. The modifications in 3.04 and 3.05 are described below.

### Brief Overview

#### GENERATOR panel:

Previously ...	Now ...
... frequency sweeps were only possible with an activated <b>equalizer</b> when you selected at least three sweep points. With only two sweep points, a wrong output level was set.	... this error is eliminated.  For manual control, see 2.5.4.1.3 Equalization of SINE, SINE BURST, DFD, MULTISINE, RANDOM For IEC/IEEEbus command, see 3.10.1.4.2 Sweep Settings for Gen. SIN, STER, BURS, S2P, MDIS,DFD, DC
... in the <b>RANDOM Domain TIME</b> function, acoustically interfering periodic artifacts were audible but irrelevant to measurements.	... this noise generator has been improved in such a way that there is no periodicity anymore.  For manual control, see 2.5.4.9 RANDOM For IEC/IEEEbus command, see 3.10.1.5.7 RANDOM
... negative phase parameters were allowed in the <b>MULTISINUS</b> function but were internally limited to 0°. Thus, you had to indicate phases in the range between 0° and 360° (e.g. 330° instead of -30°).	... this error is eliminated.  For manual control, see 2.5.4.4 MULTISINE For IEC/IEEEbus command, see 3.10.1.5.2 MULTISINE
	... you can load FTF data of the UPL successor <b>UPV</b> in the <b>RANDOM, Domain FREQUENCY</b> function without any restrictions. During this process, you ignore the spacing indication in the file and use, as usual, the spacing selected in the panel.  For manual control, see 2.5.4.9 RANDOM For IEC/IEEEbus command', see 3.10.1.5.7 RANDOM

Previously ...	Now ...
... the <b>Reclock</b> function on the reference output was only effective when you used the jitter <i>analyzer</i> .	... this restriction is eliminated. For manual control, see 2.5.3 Configuration of the Digital Generator For IEC/IEEEbus command, see 3.10.1.3 Configuration of Digital Generators

**ANALYZER panel:**

Previously ...	Now ...
... the (frequency-dependent) measurement result of the <b>quasi-peak measurement function</b> was scaled in such a way that the deviation from the rms measurement value over the entire frequency range was as low as possible. Thus, the measurement value at 1 kHz was approx. 0.5% too high.	... the quasi-peak measurement result is scaled in such a way that at 1 kHz it corresponds to the rms value.  For manual control, see 2.6.5.4 PEAK, Q-PEAK For IEC/IEEEbus command see 3.10.2.5.4 Peak and Quasi-peak Measurement incl. S/N

**FILTER panel (version 3.05 only):**

Previously ...	Now ...
... the A-weighting filter in analyzer ANLG 22 kHz and at sample rates below 96 kHz satisfied only accuracy class 1 in frequency range above 12 kHz.	... the A-weighting filter satisfies class 0 accuracy in both analog analyzers and at all sample rates of digital analyzer  For manual control, see 2.7.1 Weighting filters

Remotecontrol / universal sequence controller:

Previously ...	Now ...
<p>... you could transfer <i>measurement data and curves</i> via the trace buffer from the host to the UPL (download) as well as from the UPL to the host (upload).</p> <p>Furthermore, you could transfer (download) <i>any</i> files from the host to the UPL via the remote-control interfaces. For the opposite direction (upload of any files) you had to transfer these files by means of a special BASIC application via a COM interface.</p>	<p>... an IEC/IEEEbus command is also available for the <b>upload of any files</b> (from the UPL to the host).</p> <p>It is the query form of the already existing download command:</p> <p style="padding-left: 40px;">MMEMory:DATA? 'file name'.</p> <p>The data set transferred contains the header usual for binary data</p> <p style="padding-left: 40px;">#&lt;le&gt;&lt;le&gt;</p> <p>with the "length of the length" and "length of the binary data" information and is terminated</p> <p style="padding-left: 40px;">with the delimiter &lt;NL&gt;</p> <p>(hexcode 0A) that is usual for all data.</p> <p>For IEC/IEEEbus command, see 3.15.20 Binary Data via IEC/IEEE-Bus Interface</p>
<p>... you had four possibilities to start a <b>BASIC macro</b></p> <ol style="list-style-type: none"> <li>1. Calling it from the UPL user interface (entering or selecting the BASIC file name under Exec macro in the OPTIONS panel.</li> <li>2. Calling it from the BASIC user interface (entering the BASIC commands LOAD 'file name' and RUN).</li> <li>3. Calling it via a remote-control program (performing the IEC/IEEEbus command SYST:PROG:EXEC "file name").</li> <li>4. Autostart when switching on (using the call parameter -bn&lt;file name&gt; when starting the UPL).</li> </ol>	<p>... you have a further and extremely quick possibility to start a specific BASIC macro called USERMAC.BAS (provided you have installed the UPL-B10 option).</p> <p>To do this, simply press the Contrast key (Cntrl F10), (which is no longer necessary for color UPL).</p> <p>If available, the macro C:\UPL\USER\USERMAC.BAS is loaded and started.</p> <p>For manual control, see 2.16 Macro <b>Operation</b> IEC-Bus-Befehl siehe 3.15.18 Call BASIC-Macro</p>

Previously ...	Now ...
<p>... when starting a sweep cycle via <i>remotecontrol</i>, the previous curves were deleted. Thus, in multi-scan mode it was not possible to display several curves of a trace in a common graphic by using single triggering.</p> <p>In the universal sequence controller, this defect did not occur; the curves were only deleted when you started a continuous sweep (init:cont on).</p>	<p>... the recorded sweep curves of a multi-scan during single triggering are no longer deleted, even under remotecontrol.</p> <p>To delete the sweep curves you must transmit init:cont on exactly in the same way as under universal sequence controller.</p>
	<p>... Two new BASIC macros are provided under C:\UPL\USER .</p> <ol style="list-style-type: none"> <li>1. USERMAC.BAS as an example program for macro start by pressing the Contrast key.</li> <li>2. SELFTEST.BAS to perform a short selftest.</li> </ol> <p>The macros can only be carried out when you have installed the UPL-B10 option.</p>

## **Detailed Description of Manual Control**

In the following, the new characteristics and functions are described in detail. These pages can be inserted into the manual.

## 2.16 Macro-Operation

With the UPL, setting and measurement sequences can be written as BASIC programs or else recorded via the built-in program generator (see 3.16.4.3 Logging Mode - UPL-B10). Option UPL-B10 (Automatic Sequence Control) is required to do this. The generated BASIC programs can be stored (preferred file extension: .BAS) and called and used in various ways:

### 1. Call from BASIC user interface:

Once the BASIC user interface has been activated by pressing F3 (on the external keyboard) or BACKSP (on the UPL keyboard), the program can be loaded with LOAD (softkey or F11) and then started with RUN (softkey or F6). The program name needs to be typed during loading. After the program has been quit, the UPL user interface is reactivated with F3 or the LOCAL key.

### 2. Automatic start of a BASIC program on switch-on:

The UPL can be configured to load and execute (once) a particular program at switch-on. After the program has been quit, the UPL user interface is reactivated with F3 or the LOCAL key. The BASIC program can be either of the following:

- the program INIT.BAS under C:\UPL\USER\, if the power-up mode has been set to 2 with the aid of utility programs BASSET or UPLSET.
- any program whose name is entered with "-bn<filename>" as a call parameter when UPL is started.

### 3. Call from UPL user interface:

Via the menu item "Exec Macro" on the OPTIONS panel, a BASIC file name can be selected with the aid of the standard file box. In the file box, all files with the extension .BAS are listed as standard. The selected BASIC program is loaded and started automatically. After the program has been quit, the UPL user interface is automatically reactivated.

Advantages of this method:

- All BAS files (macros) available in the selected directory are displayed in the file box.
- Faster and more convenient file handling: the file can be selected directly from the UPL file box and does not need to be typed (possibly plus path) after activating the BASIC user interface.
- As no file name needs to be typed, a macro of this type can also be started without an external keyboard (which is required for typing text into a BASIC user interface).

#### Exec Macro

The file bearing the name shown here is loaded and executed as a BASIC file. It is essential this file was generated with the Rohde&Schwarz BASIC, eg in the autocontrol option of either UPD or UPL (UPD-K1 or UPL-B10).

Recommended file extension: .BAS

If the file cannot be opened or if it contains invalid BASIC lines, a BASIC error message is issued.

For infor about entering file names, see 2.3.2.5

#### 4. Calling by pressing the Contrast key:

By pressing the Contrast key (Ctrl F10) – which is not required in color UPLs – the USERMAC.BAS macro in the C:\UPL\USER\ directory is automatically loaded and started. An appropriate example file will be installed there as of software version 3.04. If the example file does not exist, an error message will appear.

When the macro is terminated, you will automatically return to the UPL interface.

Advantages of this procedure:

- One keystroke is sufficient to load and start the macro.
- Since it is not necessary to type the file name, you can also start this macro without an external keyboard.

#### 5. Calling from an external controller program via IEC/IEEEbus interface:

You can load and start any BASIC program via the SYST:PROG:EXEC IEC/IEEEbus command. After you terminate the program, a 1-> 0 transition is generated in the RUN bit (#14) of the operation register. The controller obtains this information via SRQ or serial poll and can fetch the measurement data. The data exchange between the external controller program and BASIC program can be performed via the measurement value displays and the measurement value buffers.

Advantages of this procedure:

- Modularity of the measurement tasks; the controller does not have to take care of *how* the measurement in the UPL is performed.
- Easier for the controller; after starting the macro, the controller can handle other tasks.
- The measurement algorithms implemented in the BASIC programs of the UPL-B10 are also available for manual control at the press of a button (see above).

## New or modified IEC/IEEE-bus commands

### 3.15.20 Binary data via IEC/IEEEbus interface

With the IEC/IEEEbus command **MMEMory:DATA 'file name'** you can transfer any binary data and therefore files between controller and UPL.

The setting command format

**MMEMory:DATA 'file name' #<lele><le><binary data>**

transfers any binary data from the controller to the data carrier (hard disk, floppy disk drive) of the UPL. The data set transferred is stored in the UPL in the file indicated as 'file name'.

The controller reads the binary data set, for example, from a file of its local data carrier, determines the length of the data set <le>, as well as the length of the length <lele> and generates the header

**#<lele><le>**

and inserts it in front of the binary data set.

The query form

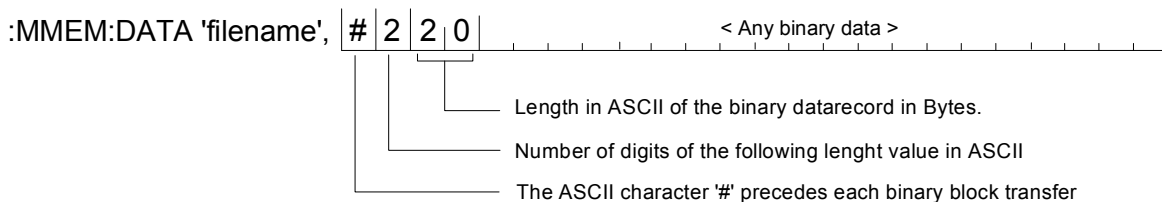
**MMEMory:DATA? 'file name'**

transfers any binary data from the data carrier of the UPL (hard disk, disk drive) to the controller. The file indicated as 'file name' on the UPL is read as a binary data set and is transferred together with the header

**#<lele><le>**

and the delimiter <NL> (hexcode 0A) to the controller.

By the length specification of the header, the controller knows how many bytes are transferred and writes this binary data (without the appended <NL>) in a file of its local data carrier, for example.



Using the IEC bus command **MMEMory:CHECK? 'filename'**, a digital signature can be calculated on the transferred binary data record in order to check whether an error occurred.

To ease the file transfer from host to the UPL not only experienced C- and IEC bus programmers the DOS programs IEC\_BT.EXE, RS232\_BT.EXE and UPMD5.EXE are provided from UPL version 2.0. These programs cover the most frequently used applications, i.e. file transfer from the controller to the UPL.



**First Steps**

After the installation of a new UPL software version 2.0 or higher, the following two EXE files are stored in the **C:\UPL\IEC\_EXAM** directory (the EXE files can be run under DOS):

IEC_BT.EXE	(copied by a process controller from a file and transferred to UPL via the IEC/IEEE bus)
RS232_BT.EXE	(copied by a process controller from a file and transferred to UPL RS232-Interface)
UPMD5.EXE	(generates the unique signature of a file)

The associated source code

IEC\_BT.C  
RS232\_BT.BAS  
is stored too.

To be able to transfer a file from a process controller to the UPL via the IEC/IEEE-bus interface or RS232 control, the files IEC\_BT.EXE, RS232\_BT.EXE and UPMD5.EXE must be copied to a floppy disk and transferred from the disk to the process controller. The disk can be copied either on the UPL under DOS or, with the UPL measurement software running, from the FILE Panel using the commands 'Copy' and 'To'.

Important: The program UPMD5.EXE must be in the directory from which IEC\_BT.EXE or RS232\_BT.BAS is started, or in a directory specified under PATH.

## Transfer of File to UPL via IEC/IEEE-Bus Interface

The program IEC\_BT.EXE allow the transfer of any file to the UPL **via the IEC/IEEE-bus interface**.

This is done by transmitting then IEC bus command MMEM:DATA followed by then contents of the file to the UPL. The filename is specified interactively.

The source file IEC\_BT.C is written in the programming language C. The source file provides information on the transfer procedure and the call-up of the MD5 signature method. IEC\_BT can be adapted to user's requirements.

For the signature method to be executed, an IEC/IEEE-bus driver from National Instruments must be installed in the process controller, and the setting Remote via IEC BUS selected in the OPTIONS Panel of the UPL.

After IEC\_BT is called up, the program tries to find

- an UPL connected to the IEC/IEEE bus. When an UPL has been found, measurements on UPL are stopped to obtain maximum transfer speed.
- The path and name of the file to be transferred are requested, as well as the path and name of the target file to be generated in the UPL.
- The program generates a temporary file with the name TEMP.OUT, which contains the IEC/IEEE-bus command required for the UPL as well as the data record to be transferred.
- TEMP.OUT is transferred to the UPL, followed by a final NL. The transfer rate between 30 and 100 Kbytes per second depends on the process controller and the UPL configuration. The file TEMP.OUT remains stored in the process controller after it is transferred so that it can be read if necessary by means of an editor capable of handling binary characters.
- At the end of the transfer, a signature is generated both of the original file stored in the process controller and the file generated in the UPL (see MD5 signature method). If the signatures agree, it can be assumed that the two files are completely identical and no transmission errors have occurred.

## Transmitting a File to UPL via RS232 Interface

see 3.17.5 Binary Data via RS232 Interface

## MD5 Signature Method

MD5 stands for "Message Digest 5" (coding and processing rule), an algorithm recognised world-wide which is used for generating a 128-bit checksum (Signature) of a data record.

To check whether a file was transferred error-free from the process controller to the UPL, the MD5 signature method can be used to generate the digital signature of the file on the process controller prior to the transfer. After the transfer of the file to the UPL via the IEC/IEEE bus, a digital signature is generated on the UPL. If the two signatures agree, it can be assumed that the contents of the files are identical and the transfer was therefore error-free. Moreover, it can be determined in this way whether any subsequent modifications have been made to the file.

The program UPMD5.EXE generates a 16-byte hexadecimal signature of any file. The signature is displayed as a 32-digit ASCII character string on the screen.

Example:

The signature of the file IEC\_BT.C is to be generated:

UPMD5 IEC\_BT.C

ASCII character string displayed on the screen:

0d45494a3e3e262609e20050b5274f58

If the signature of a file is needed for further processing in a program, the signature can be written to a file instead of being displayed on the screen:

Example:

```
UPMD5 IEC_BT.C > IEC_BT.CHK
```

UPMD5.EXE can conveniently be called up as "child process" from its IEC/IEEE-bus or RS232 control program in order to evaluate the signature.

As an example in programming language C, here are the lines of source code IEC\_BT.C:

```
// Calculate MD5 checksum of host file. Pipe result to chkfile
sprintf (syststr,"UPMD5.EXE %s > %s",hostfile, chkfile);
// UPMD5.EXE child process prints checksum to chkfile
err = system (syststr); // Call MD5 data security child process
```

Example in QuickBASIC see source code RS232\_BT.BAS lines 53 and 54

```
shellcmd$ = "UPMD5.EXE " + hostfina$ + "> " + hostcheckfina$
```

```
SHELL (shellcmd$)
```

To fetch the signature of a file stored on the UPL via the IEC/IEEE bus or under RS232 control, the command

```
"MMEMory:CHECK? 'filename'"
```

is to be sent to the UPL.

The 32-digit signature of the requested file will be sent in return.

If 'filename' is specified without giving the path, the file will be searched in the current working directory of the UPL. The current working directory of the UPL is the directory specified under Work Dir in the FILE Panel of the UPL. The source code for the MD5 signature method is available on the Internet under [www.faqs.org/rfcs/rfc1321.html](http://www.faqs.org/rfcs/rfc1321.html)