3 Remote Control

3.1 Introduction

- UPL may be equipped with option UPL-B4 permitting remote control via an IEC/IEEE-bus interface to IEC 625.1/IEEE 488.2 standard and an
- RS-232 interface at the COM2 port.

In addition, UPL may be equipped with option UPL-B10 permitting sequence control for automatic measurements.

Note:

The sections below mainly describe remote control via the IEC/IEEE-bus interface but most of them also applies to ***remote control via the RS-232 interface*** *and the* ***Universal Sequence Controller UPL-B10.*** *If this is not the case reference will be made to the applicable section.*

Note:

If the remote-control option UPL-B4 was not originally ordered with the UPL, it may be obtained via the local sales engineer (for Order No. see data sheet). After entering an enable code as described in the supplied instructions, the IEC/IEEE-bus option and the RS-232 interface are ready for use without any additional hardware installations being required.

Note:

If the remote control option UPL-B4 is not enabled, UPL should not be connected to other instruments via the IEC/IEEE bus as in this case problem-free operation cannot be guaranteed.

The connectors for the IEC/IEEE-bus and the RS-232 interface (COM2) are located at the rear of the instrument. They allow a controller to be connected for remote control. The instrument supports SCPI version 1993.0 (Standard Commands for Programmable Instruments). The SCPI standard is based on standard IEEE 488.2 and aims at the standardization of device-specific commands, error handling and status registers (see section 3.5.1 SCPI Introduction).

Note:

In the IEC/IEEE-bus command lists given below, all commands confirmed or approved by the SCPI committee are written in normal characters. Other innovative commands or commands required for general measurements are in italics and in the form and style as specified by SCPI (not part of SCPI definition).

This section assumes a basic knowledge of IEC/IEEE-bus programming and controller operation. A description of the interface commands is to be obtained from the relevant manuals.
The requirements of the SCPI standard placed on command syntax, error handling and configuration of status registers are explained in detail in the respective sections. Tables provide a fast overview of the bit assignment in the status registers. The tables are supplemented by a comprehensive description of the status registers. For detailed program examples of the main functions see **3.15 Examples of IEC/IEEE-Bus Programming**.

All program examples for IEC/IEEE-bus control are written in R&S BASIC, likewise all program examples in the UPL software in the paths

C:\UPL\IEC EXAM\EXAM1.BAS ff and
C:\UPL\B10 EXAM\EXAM1.BAS ff.

Examples C:\UPL\IEC EXAM\EXAM1.BAS ff can be directly run on a controller using a suitable IEC/IEEE-bus card and R&S BASIC. Files with the extension .SAC are setup files required by the program examples for setting the UPL. Files with the extension .TXT provide the program code of the examples as an ASCII file and can be accessed by any editor.

Examples C:\UPL\B10 EXAM\EXAM1.BAS ff can be run on an UPL using the Universal Sequence Controller UPL-B10. Meaning of files with the extensions .SAC and .TXT as described above.

3.2 First Steps (Read-in of Measured Values)

The short and simple operating sequence below permits rapid setup of the instrument and of its basic functions.
A condition is, however, that the **IEC/IEEE-bus address**, which is factory-set to **20**, **is not changed**.

Program example:
Triggering 10 measurement results and display on the screen.

1. Interconnect instrument and controller using the IEC/IEEE-bus cable.

2. Write the following program on the controller and start it with RUN (F2):

```
10 IEC TERM 10: ’     Controller expects LF as a delimiter of an UPL reply
20 IEC TIME 5000:              ’Controller waits max. 5 s for a reply from
30 ’                            UPL before sending an IEC/IEEE-bus timeout
40 IEC OUT 20,"*RST": ’                                  UPL default setup
45 IEC OUT 20,"INP:TYPE GEN2": ’Internal connection to generator channel 2
50 IEC OUT 20,"*CLS": ’                Resets IEC/IEEE-bus status register
60 FOR I = 1 TO 10
70 IEC OUT 20,"INIT:CONT OFF;*WAI": ’        Triggers a single measurement
80 IEC OUT 20,"SENS:DATA?": ’                Requests a measurement result
90 IEC IN 20, M$:                           ’Read-in of measurement result
100 PRINT M$:                                ’Output of measurement result
110 NEXT I
120 END
```

As a result of the default setup with *RST (see annex A UPL Default Setup in the UPL manual), the UPL generator produces a 1-kHz sinewave signal with a level of 0.5 V.
Command `INP:TYPE GEN2` establishes an internal connection between generator channel 2 and analyzer channel 1 so that no cabling of outputs and inputs is required for this first test. The UPL analyzer carries out 10 RMS measurement, indicates the measurement results in the display and outputs them continuously on the controller display.

Note:

A measurement result must be triggered before it can be displayed on the screen (line 70). The settled result available after triggering can be requested (line 80), read-in (line 90) and output to the controller display (line 100).

3. Press the [LOCAL] key on the front panel to return to manual operation.

Universal Sequence Controller (UPL-B10)	**RS-232 interface**
See **3.16.4.2 First Steps (Readout** of Measurement Results)	See **3.17.4 First Steps (Readout** of Measurement Results)

3.3 Switchover to Remote Control

After power-on, the instrument is always in the manual operating state (LOCAL state) and can be operated from the front panel or the keyboard. It is switched to remote control (REMOTE state shown by REMOTE caption in the screen center) as soon as it receives an addressed command from a controller.

If the UPL has been set to the manual status from REMOTE with the LOCAL key, then each command via

R&S BASIC switches the UPL to the REMOTE status, because R&S BASIC sends each command in addressed form to the UPL. If the

NI GPIB device driver is used, the first IEC/IEEE command should be send addressed. This addressing for board-specific programming is for example the command ibcmd (`board address, 20, 1`) or for device-specific programming the **repeat addressing** setting in the NI GPIB device driver – or `ibloc(0)` before the first IEC/IEEE command. If the UPL in the LOCAL state is sent a non-addressed command, the latter will be carried out but the screen will show the old manual settings and not the expected empty screen with the REMOTE caption.

During remote control, operation from the front panel is disabled. The instrument remains in the remote state until it is reset to manual operation via the front panel or the IEC/IEEE bus (see section 3.3.3 Return to Manual Operation). Switching from manual operation to remote control and vice versa has no effect on the instrument setup.

Universal Sequence Controller (UPL-B10)	RS-232 interface
See **3.16.4.1 Connection Basic-UPL**	**3.17.2 Switchover to Remote** Control

3.3.1 Setting the Device Address

The IEC/IEEE-bus address of the instrument is factory-set to 20. It can be changed manually under UPL IECadr in the OPTIONS panel or via the IEC/IEEE bus.

Manually: Enter the required address under **UPL IECadr** in the OPTIONS panel.

Via IEC/IEEE bus:

```
IECOUT 20,"SYST:COMM:GPIB:ADDR 30":'                    Set new address 30
HOLD 500:'          Wait approx. 500 ms before new commands are effected
```

Universal Sequence Controller (UPL-B10)	RS-232 interface
Not applicable	Not applicable

3.3.2 Indications During Remote Control

- The remote-control status is signalled by the word REMOTE in the display center and by the REM LED on the UPL front panel.
- Measurement results (active) are indicated in the upper part of the display.
- Error messages occurring during IEC/IEEE-bus control are displayed in plain text in the lower part of the UPL display (see also section **3.14 List of Error Messages**).

3.3.3 Return to Manual Operation

Return to manual control can be made from the front panel or via the IEC/IEEE bus.

Manual: Press the [LOCAL] key, the REM LED goes out.

Prior to the switchover, command processing must be completed as otherwise remote control is immediately switched on again.

Note: LLO state (Local Lockout):

The [LOCAL] key can be locked by the universal command LLO in order to prevent inadvertent switchover:

R&S-BASIC:
R&S BASIC command `IECLLO`

NI-GPIB-Treiber:
NI command `SendLLO(0)`

This state can be cancelled only by switching the "REN" line of the IEC/IEEE bus from 0 to 1:

R&S-BASIC:
R&S BASIC commands IECNREN, followed by IECREN.

NI-GPIB-Treiber:
NI commands ibsre(0,0); followed by ibsre(0,1);.

Via IEC/IEEE bus: **R&S-BASIC:**

```
   :
IECLAD 20:'                Sets device to manual operation
IECGTL
   :
```

NI-GPIB-Treiber:

```
ibloc (0);
```

To switch UPL from LOCAL state to REMOTE state, see 3.3 Switchover to Remote Control

Universal Sequence Controller (UPL-B10)	RS-232 interface
See **3.16.4.1 Connection Basic-UPL**	**3.17.3 Return to Manual** Operation

3.4 IEC/IEEE-Bus Messages

The messages transmitted on the data lines of the IEC/IEEE bus can be subdivided into two groups:
- **interface messages** and
- **device-dependent messages**

3.4.1 Interface Messages

Interface messages are transmitted on the data lines of the IEC/IEEE bus, with the control line "ATN" being active. They are used for communication between the controller and the instrument and can only be sent by a controller with controller function on the IEC/IEEE bus.
There are two groups of interface messages:
- **common commands** and
- **addressed commands**

See 3.13.1, Universal Commands and 3.13.2, Addressed Commands

Common commands affect all devices connected to the IEC/IEEE bus without any addressing being required, whereas addressed commands only affect devices addressed as a listener. The relevant interface messages for the UPL are listed in annex A.

Universal Sequence Controller (UPL-B10)	RS-232 interface
Not applicable	Not applicable

3.4.2 Device-Dependent Messages (Commands and Responses)

The device-dependent messages are transmitted on the data lines of the IEC/IEEE bus, with the control line ATN being not active. The ASCII code is used for data transmission. Device-dependent messages are differentiated according to the direction in which they are sent via the IEC/IEEE bus:

- **Commands** are messages sent by the controller to the UPL. They control the device functions and request information.
  The commands are subdivided according to two criteria:
  1. According to the effect they have on the device:
  Setting commands cause device settings to be made, e.g. the resetting of the UPL or the setting of the output level to 1 Volt.
  Queries cause data to be provided for output via the IEC/IEEE bus, eg for device identification or query of the active input.
  2. According to their definition in the IEEE 488.2 standard:
  Common commands are precisely defined in their function and notation in the IEEE 488.2 standard. They refer to functions as for instance the management of the standardized status registers, resetting and selftest
  Device-specific commands refer to functions that depend on the device characteristics, such as frequency setting. A large number of these commands has also been standardized by the SCPI Consortium (see section 3.5.1 SCPI Introduction)
- **Responses** are messages sent by the UPL to the controller following a query. They may contain results, device settings or information on the device status (see section 3.5.4 Responses to Queries).

The section below describes structure and syntax of device-dependent messages. As from section **3.10 IEC-bus Commands** onwards, the commands are listed independent of their function, described in brief and listed in alphabetical order.

3.5 Structure and Syntax of Device-Dependent Messages

3.5.1 SCPI Introduction

SCPI (Standard Commands for Programmable Instruments) describes a standardized command set for the programming of instruments regardless of the type of instrument or manufacturer. The goal of the SCPI Consortium is to standardize device-specific commands to a large extent. For this purpose an instrument model has been developed which defines identical functions within an instrument or of different instruments. Command systems have been generated and assigned to these functions so that it is possible to address identical functions by the same commands. The command systems have a hierarchical structure. Fig. 3-1 shows this tree structure, using a detail from the SOURce command system for controlling the signal sources of the instrument. The other examples of syntax and structure of the commands are taken from this command system.
SCPI is based on the IEEE 488.2 standard, ie it uses the same syntax elements as well as the "common commands" defined therein. The syntax of the responses is partly subjected to stricter rules than laid down in the IEEE 488.2 standard).

3.5.2 Command Structure

The commands consist of a so-called header and usually one or several parameters. Header and parameters are separated by a "white space" (ASCII code 0 to 9, 11 to 32 decimal, eg space). The headers may be composed of several keywords. The query form is generated by appending a question mark directly to the header.

Common Commands

Common commands consist of a header preceded by an asterisk "*" and one or several parameters.

Examples		
	`"*RST"`	RESET, resets the UPL
	`"*ESE 253"`	EVENT STATUS ENABLE, sets the bits of the Event Status Enable Registers
	`"*ESR?"`	EVENT STATUS QUERY, queries the contents of the Event Status Register.

Universal Sequence Controller (UPL-B10)	RS-232 interface
Common commands to be used can be looked up in Table **3.9 Common Commands**.	

Device-specific commands

Hierarchy

Device-specific commands have a hierarchical structure (see Fig. 3-1). The various levels are represented by compound headers. Headers of the highest level (root level) have one keyword only. This keyword stands fro a whole command system.

Example: `SENSe` This keyword denotes the command system SENSe (see section 3.5.7 Programming Model of UPL Generator)

For lower-level commands the full path has to be specified, starting with the highest level in the left-most position. The individual keywords are separated by a colon ":".

Example: "SENSe:FUNCtion:SETTling:MODE EXPonential"
(exponential settling of function test results)

This command is at the fourth level of the SENSe system. It selects the function settling mode.

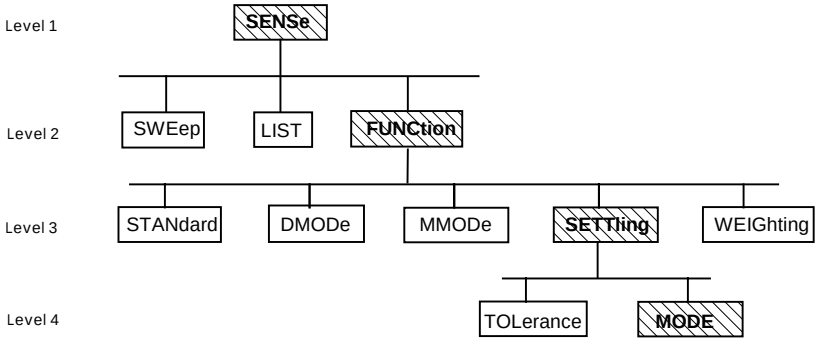


Fig. 3-1 Tree structure of SCPI command system, SENSe being shown as an example

Optional keywords: Some command systems permit certain keywords to be optionally inserted into the header or omitted. In the manual these keywords are marked by square brackets. For reasons of compatibility with the SCPI standard, the instrument must be able to recognize the full command length. Some of the commands become considerably shorter when the optional keywords are omitted.

Example: "DISPlay[:WINDow]:TEXT[:DATA] 'String'"
The following command has the same effect:
"DISPlay:TEXT 'String'" (text in graphic display)

Long and short form: The keywords have a long and a short form. The short form or the long form can be entered, other abbreviations are not permissible

Example: "STATus:QUEStionable:ENABle 1"
The following command has the same effect:
"STAT:QUES:ENAB 1" (0 bit of Status Questionable Register enabled)

Note:

The short form uses uppercase characters, the long form gives the whole keyword. Uppercase and lowercase letters are used for identification in the manual, the UPL itself does not differentiate between uppercase and lowercase characters.

Parameter: The parameter must be separated from the header by a "white space". If a command contains several parameters, they have to be separated by a comma ",". Some of the queries permit the parameters MINimum and MAXimum to be entered. For a description of the various types of parameter see section 3.5.5 Types of Parameters.

Example: "SENSe:FREQuency:STARt? **MAX**imum"
Response: 21641.8
(query requesting the maximum value for starting a frequency sweep)

Numeric suffix: If a device has several identical functions or features, eg inputs, the desired function can be selected by a suffix added to the command.

Example: "SENSe**2**:Voltage:REFerence 1V"
(reference value for a relative INPUT peak measurement)

Note: *Entries without suffix are interpreted like entries with suffix 1*

3.5.3 Structure of a Command Line

The UPL can process IEC/IEEE-bus commands containing up to 240 characters. If this length is exceeded, an error message is output.
A command line may take up more than one line on the controller display. The end of a line is determined by a delimiter.

Delimiters for a command set to the UPL:
A command line to the UPL may contain one or several commands. It is terminated by a <New Line> (ASCII code, 10 decimal) or <EOI> (EOI line active) together with the last useful character of the command line or by <New Line>. Since a <Carriage Return> character (ASCII code, 13 decimal) as a filler before the delimiter has no effect, the combination <Carriage Return><New Line> is permissible.

Example for generating the various delimiter combinations:

<**Carriage Return**>+<**New Line**>+<**EOI**> (default setting of IEC/IEEE-bus controller):

```
10 IECEOI
20 IECOUT 20,"XYZ"
```

If no **';'** is sent at the line end, <CR><NL> is added.

<**New Line**>+<**EOI**>**:**

```
10 IECEOI
20 IECOUT 20,"XYZ"+CHR$(10);
```

';' ensures that no <CR><NL> is added.

<**EOI**>**:**

```
10 IECEOI
20 IECOUT 20,"XYZ";
```

';' ensures that no <CR><NL> is added.

<**New Line**>**:**

```
10 IECNEOI
20 IECOUT 20,"XYZ"+CHR$(10);
```

';' ensures that no <CR><NL> is added.

<**Carriage Return**>+<**New Line**>**:**

```
10 IECNEOI
20 IECOUT 20,"XYZ"
```

If no **';'** is sent at the line end, <CR><NL> is added.

Universal Sequence Controller (UPL-B10)	RS-232 interface
User cannot influence the delimiter.	See **3.17.6 Differences to Remote** Control via IEC/IEEE Bus

Sequencing of commands:

Commands may be lined up unless they exceed 240 characters in length.

Several commands in a command line are separated by a semicolon **";"**. If the next command belongs to a different system, the semicolon is followed by a colon **":"**

Example:

```
IECOUT 20,"SOUR:FREQ:STAR 20Hz;:SYST:BEEP:STAT ON"
```

(switches on beeper for error messages)

This command line contains two commands. The first command belongs to the SOURce system and is used to set the start frequency for a generator sweep. The second command is part of SYSTem and switches on an acoustic alarm in the case of errors.

If the successive commands belong to the same system and therefore have one or several common levels, the command line may be shortened. The second command following the semicolon then starts at the level that is below the common levels (see also Fig. 3-1). The colon after the semicolon has to be omitted.

Example:

```
IECOUT 20,"SOUR:FREQ:STAR 20Hz;:SYST:BEEP:STAT ON"
```

(Start- and stop command for sweep)

This command line is shown in full length and contains two commands separated by a semicolon. Both commands belong to the SOURce command system, FREQency subsystem, ie they have two common levels.
In the shortened command line the second command starts at the level below SOURce:FREQ. The colon after the semicolon has to be omitted.

The abbreviated form of the command line is:

```
IECOUT 20,"SOURce:FREQ:STAR;STOP 15kHz"
```

A new command line always begins with the complete path.

Example:

```
IECOUT 20,"SOUR:FREQ:START 20Hz"
IECOUT 20,"SOUR:FREQ:STOP 15 kHz
```

(Start- and stop command for sweep)

When common commands starting with an asterisk '*' are linked in a command string, they are preceded by a semicolon ';' and not by ';:' as is the case with linked UPL commands.

Example:

```
IECOUT 20,"INIT;*WAI;:SENS:DATA1?"
                  ↑
```

(triggers measurement, selects result of function measurement on channel 1 and sends next command only after completion of the measurement!)

3.5.4 Responses to Queries

Delimiters of responses sent by the UPL:

All responses sent by the UPL to the IEC/IEEE-bus controller are terminated by <New Line> (ASCII code, 10 decimal) and <EOI> (EOI line active). With the aid of the `IECTERM` command, the IEC/IEEE-bus controller can be set to various delimiters, but only the setting

`IECTERM 10` is useful.

- `IECTERM 10` causes the IEC/IEEE-bus controller to respond to the delimiter <New Line>. With
- `IECTERM 1` (EOI only)), the response string from the UPL displayed on the monitor of the IEC/IEEE-bus controller is followed by an empty line, as <New Line> before <EOI> is interpreted as part of the response string.
- `IECTERM 0` (<Carriage Return>+<New Line>), the response string from the UPL displayed on the monitor of the IEC/IEEE-bus controller will again be followed by an empty line, as the <Carriage Return> is interpreted as part of the response string and changed into a line feed command by the IEC/IEEE-bus controller.

All other settings cause a TIMEOUT.

Universal Sequence Controller (UPL-B10)	RS-232 interface
User cannot influence the delimiter.	See **3.17.6 Differences to Remote** Control via IEC/IEEE Bus

Possible queries

Unless explicitly specified otherwise, a query is defined for each setting command. The query is generated by appending a question mark to the associated setting command. Some of the SCPI rules imposed on the query responses are stricter than those of the IEEE 488.2 standard:

1. The requested parameter is sent without header.
   Example: **Setting:** `"INPut:TYPE BAL"`
   Query: `"INPut:TYPE?"` **Response:** BAL
   (input BAL for analog analyzer)

2. Maximum and minimum values requested by the character data MAXimum and MINimum are returned as numeric values
   Example: **Setting:** `"SENSe:FREQuency MAX|MIN|`any value`"`
   Query: `"SENSe:FREQuency? MAX"` **Response:** `21641.8`
   Query: `"SENSe:FREQuency? MIN"` **Response:** `2.0`
   (maximum or minimum value depend on current setting)

3. Numeric values with floating decimal point are output in the same unit as entered.
   Example: **Setting:** `"SENSe:FREQuency:STARt 20kHz?"`
   Query: `"SENSe:FREQuency:STARt?"` **Response:** 20.0 for 20 kHz
   (sweep start frequency)

4. Integral values are returned as such.
   Example: **Setting:** `"INST:NSEL 1"`
   Query: `"INST:NSEL?"` **Response:** `1`
   (25-kHz analog generator)

5. Text (character data) are returned in short form (see sections 3.5.1 and 3.5.4).
   Example: **Setting:** `"OUTPut:TYPE BALanced"`
   Query: `"OUTPut:TYPE?"` **Response:** BAL
   (generator output channel 1 XLR balanced)

6. Character strings are output in the same way as they are input, ie the simple and double quotation marks are also output (see section 3.5.6 Overview of Syntax Elements).

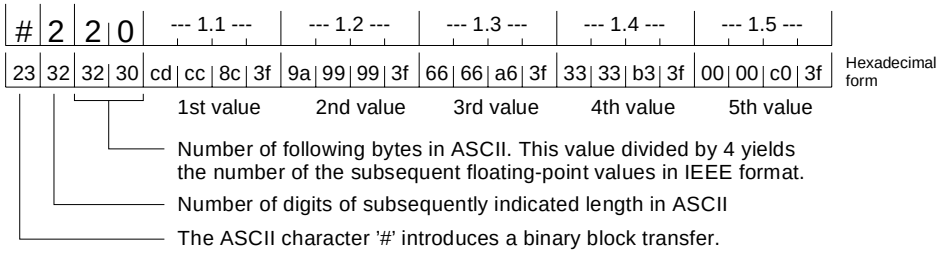
Example: **Setting:** `"MMEMory:STORe:STATe 2,'LASTSAVE.SCO'"`
Query: `"MMEMory:STORe:STATe? 2"` **Response:** `'LASTSAVE.SCO'`
(storage of UPL setting under 'LASTSAVE.SCO'

7. Data blocks are output as ASCII characters or in binary form depending on the IEC/IEEE-bus commands FORM ASC and FORM REAL.
Example:
Five floating-point values in the form of a trace are available after a sweep.

Setting: `"FORM ASC"`
Query: `"TRACe? TRACe"` **Response:** `1.1,1.2,1.3,1.4,1.5`
(load Y values of curve A into the UPL)

Setting: `"FORM REAL"`
Query: `"TRACe? TRACe"`

The **response** is sent as a binary data stream. Floating-point values are 4 bytes long and output in the IEEE format (LSB first):



To receive the binary data stream in its full length, set **EOI** (in R&S-BASIC: **IEC TERM 1**) in the controller program and not LF (0Ah = 10d) (in R&S-BASIC: IEC TERM 10) as would normally be set for the reception of ASCII strings. This is to avoid the binary data stream being interrupted if the bit combination 0Ah occurs.

Universal Sequence Controller (UPL-B10)	RS-232 interface
See **3.16.4.4 Differences from the** IEC-bus Syntax - UPL-B10	See **3.17.6 Differences to Remote** Control via IEC/IEEE Bus

3.5.5 Types of Parameters

Most commands require the specification of a parameter. The parameters must be separated from the header by a "white space" (ASCII code, 0 to 9, 11 to 32 decimal, generally a blank). Parameters may be specified as numeric values, Boolean parameters, character data, character strings and block data. The type of parameter required for the specific command as well as the permitted range of values are described together with the commands (see section **3.10 IEC-bus Commands**).

Numeric values Numeric values may be entered in any customary form, ie with sign, decimal point and exponent. If the values exceed the resolution of the UPL, they will be rounded off. The number including the exponent may comprise up to 20 characters. The number of digits of mantissa and exponent is only limited by this condition. The exponent is denoted by an "E" or "e". The exponent alone must not be used. Physical quantities may be stated with the unit. Permissible prefixes for the unit (also called engineering) are M (Mega), K (Kilo), m (milli) and u (micro). The units for which these prefixes may be used can be seen from the tables in section **2.4 Units** in the UPL manual. With no unit stated, the basic unit is assumed (see section **3.10 IEC-bus Commands** column Basic unit of UPL manual).

Examples:

"SOURce:FREQuency	1.5 kHz"	1500 Hz
	1.5E3	1.5E+3
	1.5E 3	1.5E 03
	+1.5E3	001.5E3

Special numeric values The parameters MINimum, MAXimum, UP and DOWN are interpreted as special numeric values.

MIN/MAX MINimum and MAXimum denote the minimum and maximum value.
Upon a query the numeric value will be returned.

Example: Setting command: `"SOURce:VOLTage MAXimum"`
Query: `"SOURce:VOLTage?"` **Response:** `24`
(maximum generator level)

UP/DOWN UP increments, DOWN decrements the current numeric value by 0.001%.

NAN **N**ot **A** **N**umber represents the value 9,91E37, is only sent as a device response and denotes missing or illegal values.

Boolean parameter Boolean parameters represent two states. The on state (true condition) is represented by ON, the off state (false condition) by OFF.

Example: Setting command: `"SENS:FUNC:WEIG ON"`
Query: `"SENS:FUNC:WEIG?"` **Response:** `ON`
(W&F weighting filter on)

Character data Character data follow the syntax rules for keywords, ie they also have a short and a long form. Like any other parameter, they must be separated from the header by a 'white space'. A query returns the short form of the character data.

Example: Setting command: `"DISPlay:MODE COLB"`
Query: `"DISPlay:MODE?"` **Response:** `COLB`
(Coloured representation on internal and external display)

Character string According to SCPI strings must always be given in single or double quotation marks. Normally single quotes are used as the double quotes are used for identifying the total IEC/IEEE-bus command as a string.

Example: `"MMEMory:DELete 'C:\UPL\USER\MYSETUP.SCO'"`
(clearing of specified file)

Data strings Numeric values are to be separated by commas, decimal digits by a '**.**'.

Example: `"TRAC LIST1, 100.0,1000.0,5000.0,15000.0"`
(setting the X values in a graphics display)
`"SOUR:LIST:FREQ 1.1,1.2,1.3,1.4,1.5"`
(setting the frequencies for a frequency sweep)

Whether the data strings start with a comma or a blank depends on the command and is defined by SCPI.

3.5.6 Overview of Syntax Elements

The following list provides an overview of the syntax elements..

: The colon separates the keywords of a command.
In a command line, the colon following a semicolon identifies the highest command level.

; The semicolon separates two commands in a command line. It does not change the path.

, The comma separates several parameters of a command.

? The question mark forms a query.

***** The asterisk identifies a common command.

" Quotation marks denote the beginning of a character string and terminate it.

The double cross denotes the beginning of block data.

␣ A "white space" (ASCII code 0 to 9, 11 to 32 decimal, eg space) separates header and parameter.

. Decimal point of numeric values.

3.5.7 Programming Model of UPL Generator

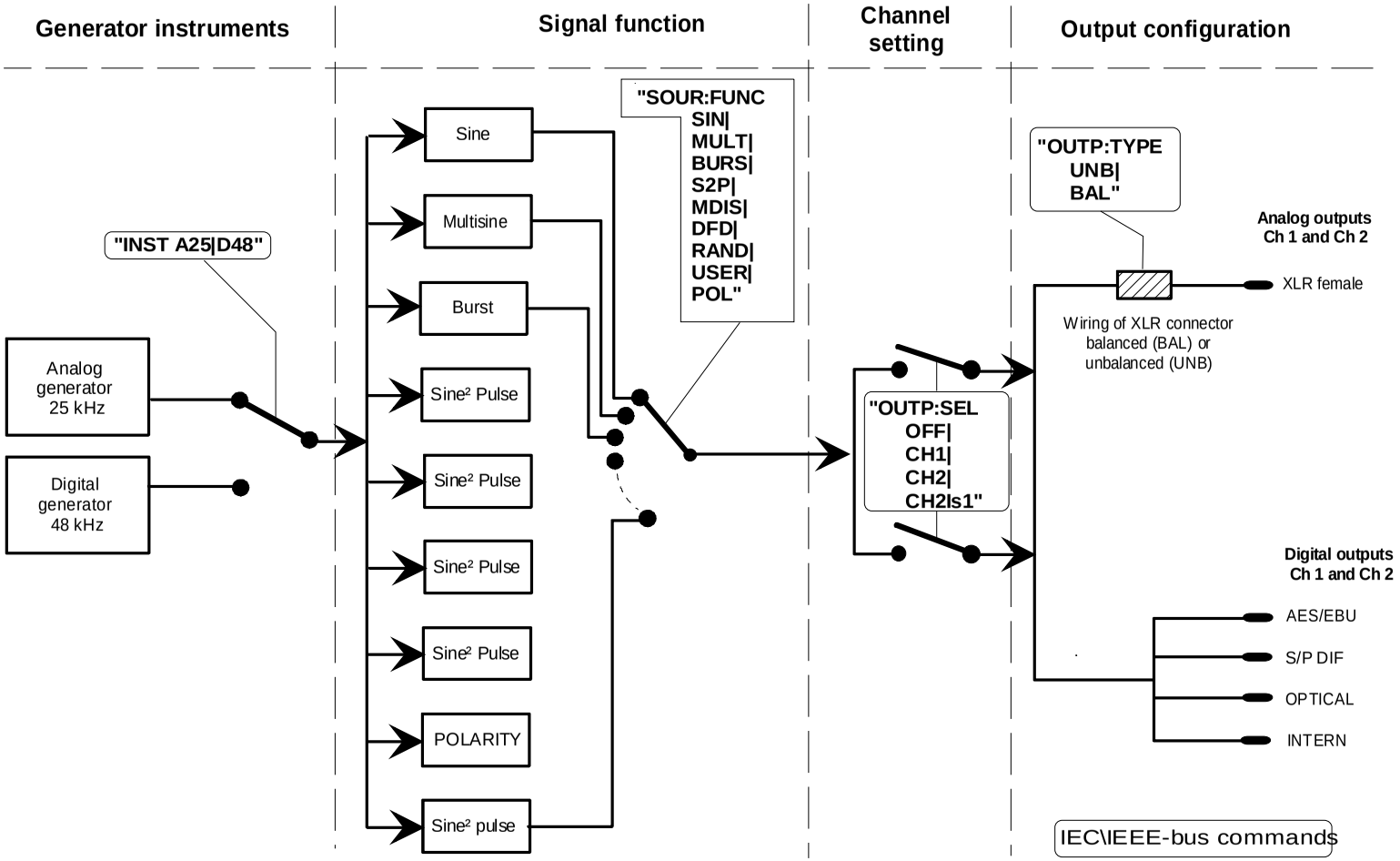


Fig. 3-2 Instruments and signal functions of UPL generator

3.5.8 Programming Model of UPL Analyzer

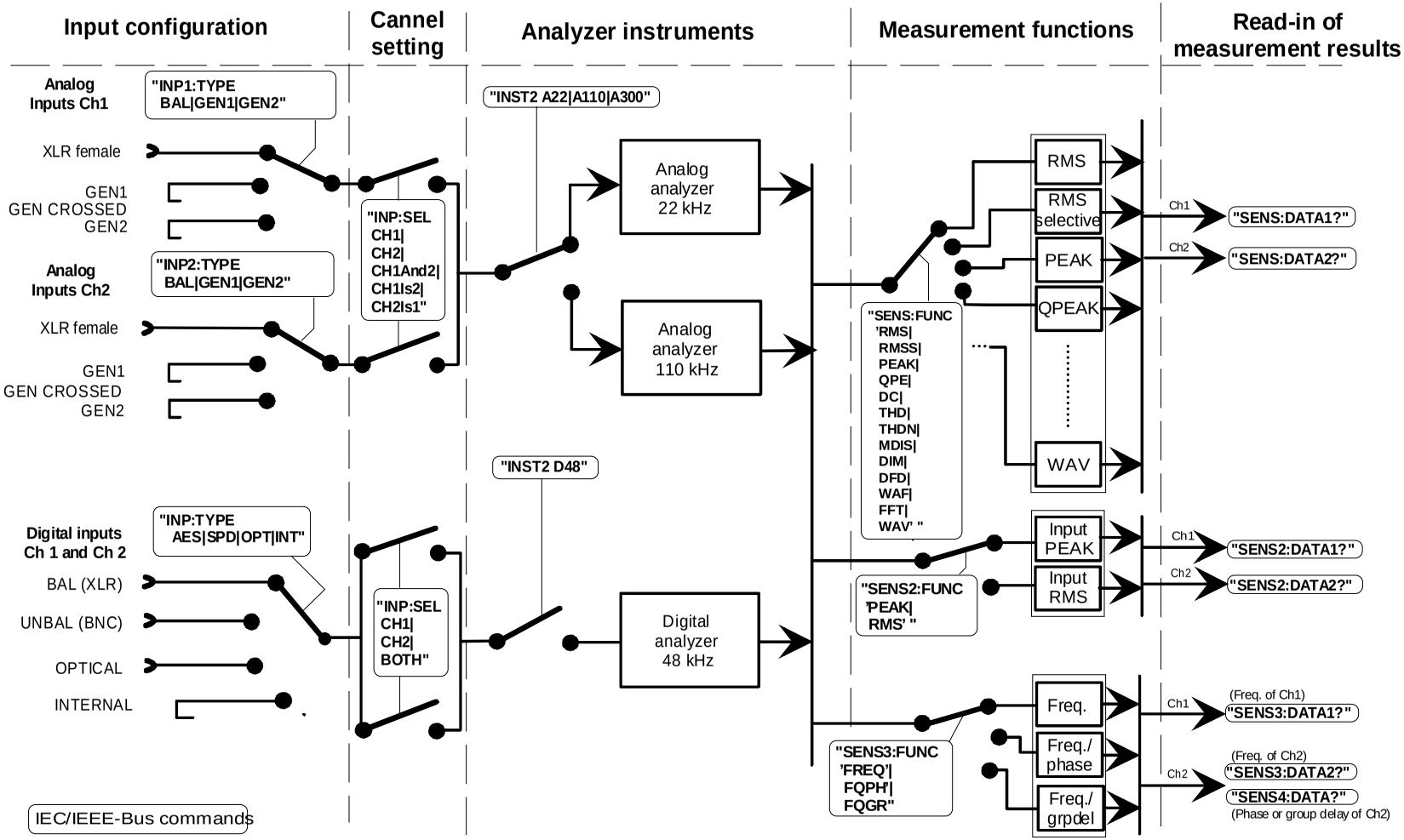


Fig. 3-3 Instruments and measurement functions of UPL analyzer

3.6 Instrument Model and Command Processing

The instrument model shown in the following figure has been configured under the aspect of processing IEC/IEEE-bus commands. The individual components operate independently of each other and simultaneously. They communicate with each other by means of so-called messages.

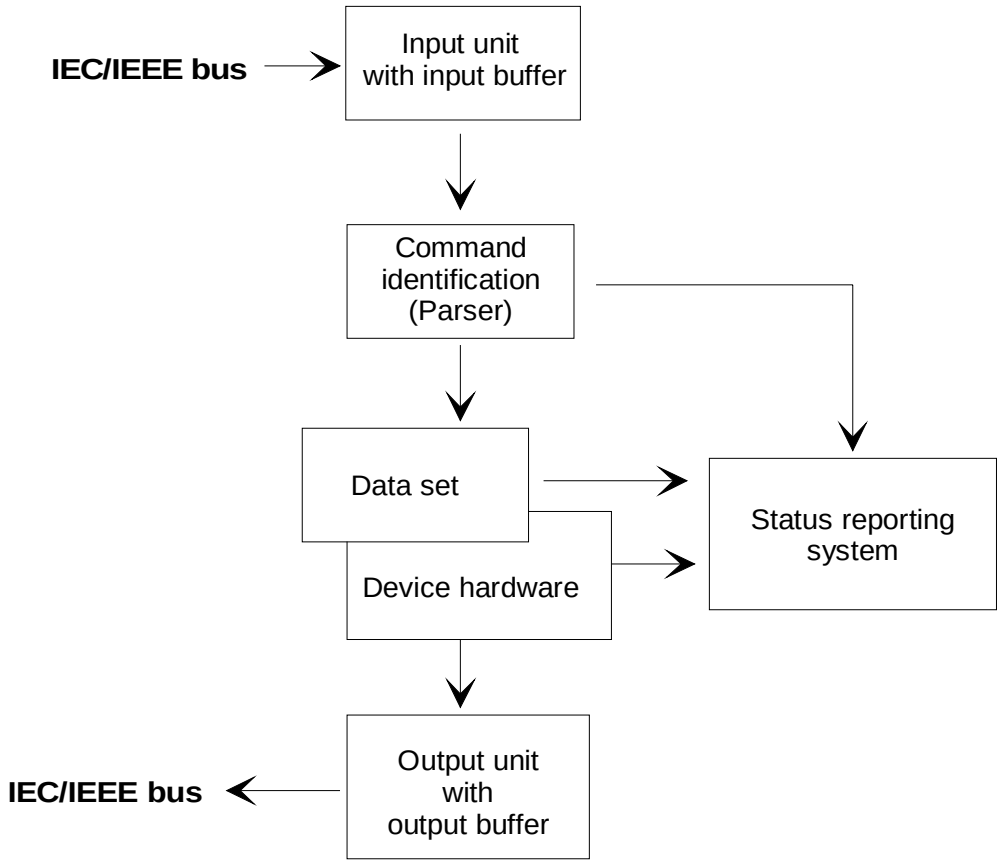


Fig. 3-4 Instrument model with remote control via IEC/IEEE-bus

3.6.1 Input Unit

The input unit receives the commands in the form of characters from the IEC/IEEE bus and collects them in the input buffer. The input buffer has a capacity of 1024 characters. As soon as the input buffer is full or receives the interface message DCL, transfer on the IEC/IEEE bus is stopped and the received data are sent to the parser where the commands are checked for syntax and semantic errors and the hardware is set in the sequence in which the commands arrived. Data transfer on the IEC/IEEE bus is then continued. However, if the buffer is not yet full when a terminator is received, the input unit can receive the next command while the previous command is identified and executed. Reception of a DCL clears the input buffer.

3.6.2 Parser

The parser (to parse = grammatically analyze) analyzes the data received from the input unit proceeding in the order in which the data are received. A GET (Group Execute Trigger), for example, is only processed when the previously received commands are executed. A DCL command will be given priority. Each command identified as correct will immediately be executed and causes a hardware setting.
Syntax and semantic errors are recognized and passed on to the status reporting system. The remaining part of a command line after the syntax error will be further analyzed as far as possible and processed.

While the parser is setting the hardware, the input unit can collect new commands in the input buffer. This means that further commands can already be processed while settings are being made in the hardware ("overlapping execution").

3.6.3 Setting the Device Hardware

The term "device hardware" refers to that part of the UPL which performs the instrument function proper: signal generation, measurements, etc. This does not include the controller.

An IEC/IEEE-bus command line which may contain several setting commands is sent to the parser, which checks the IEC/IEEE-bus commands for syntax and semantic errors. If a command is identified as correct, a check is made with the aid of the current settings whether the command is permissible. Permissible commands are taken over into the UPL data set (corresponds to hardware settings) and the device hardware is set. If it turns out that the command is illegal because of the existing settings, as for example a generator level of 20 V on the unbalanced outputs, an "execution error" message is sent to the status reporting system and no new settings will be made in the device hardware for this command. Subsequent commands which prove to be permissible are executed as described in the following section.
This strictly hierarchical sequence ensures that at no time illegal device states will be set.

3.6.4 Why is a Specific Operating Sequence Sometimes Required?

- Each IEC/IEEE-bus command received by the UPL is immediately checked whether it is permissible. This check can only be carried out if the UPL knows which instrument or function the command is intended for, ie if the addressed instrument or command has been activated **before.**

  Example1:
  Selecting an input impedance of 600 Ohm is permissible for the balanced input, not for the unbalanced one.

  If this check and the respective error messages were to be omitted, measurements would be carried out with incorrect instrument settings or would not be possible at all and the user would not even know about it.

  Example2:
  Setting: generator, unbalanced outputs.
  Remote programming of 20 V generator level.
  This is not possible, however, as the balanced output allows only 12 V to be set.

  – Should the UPL accept this incorrect setting without signalling an error because the user might afterwards select unbalanced outputs which can handle 20 V.
  – What will happen when this subsequent selection is not made?
  – Should the incorrect voltage be output?
  – Or, should an error message be output when the switchover is not performed?
  – And **when** should this error be signalled?

These conflicts can be avoided by observing the required operating sequence!

- Since similar menu items of different instruments or functions are addressed with the same command in remote control, they can only be distinguished by a reference to the currently used instrument or function.

  Example:
  Command `"INPut:SELect CH1"` may be used for all 3 analyzer units.

  To provide the possibility for presetting also parameters of non-active functions, information on the instrument or function to be addressed would have to be added to each IEC/IEEE-bus command. Thus the internal address management of UPL commands would have to be carried out by the IEC/IEEE-bus programmer. This would make programming unnecessarily difficult and the number of commands would increase by many times. The UPL uses approx. 6000 menu items which would then have to be "called up" separately by the user.

- An important operational feature of the UPL is its capability to note the settings for the individual generator and analyzer functions and to restore all required parameters when one of these functions is selected.

  This eliminates the need to reset each parameter when a new function is selected.

  Example:
  Settings: Generator, MOD DIST function with frequencies 400 Hz and 7 kHz.
  Switchover to DFD function
  The frequency pair 11 and 12 kHz, which was set when the DFD function was used last, is automatically reset.

  This function of the UPL is also available for operation via IEC/IEEE bus, ie even in the case of remote control a function switchover triggers an **automatic** reset of all pertaining parameters. This is a much quicker and more reliable than a reset of each parameter every time by the user via IEC/IEEE bus.
  If settings for non-active instruments would be permissible as well, this automatic function could no longer be used, as in this case already set parameters would also be overwritten.

- As the same internal data sets are used for manual and remote control (combined manual and remote control should be possible), the same method of internal parameter processing can only be allowed for the two modes. Since automatic parameter restoration is indispensable for manual control, it must also be used in the case of remote operation.

To relieve the user from having to pay attention to such not so obvious conditions, the following applies for manual and remote control of the UPL:

Any setting that is not permissible cannot become effective in the UPL

This has sometimes the disadvantage that a certain sequence has to be observed even in the case of remote-control. However, this is more than compensated for by the fact that setting errors are immediately signalled.

Note*:*

The command logging capability offered by the Universal Sequence Controller for UPL, option UPL-B10, minimized the risk of incorrect settings (see 3.15.3 Command Logging - Converting B10 into IEC/IEEE-Bus Commands*)*

3.6.5 Status Reporting System

The status reporting system collects information on the device status and makes it available to the output unit on request. Structure and function are described in detail in section **3.7 Status Reporting System**.

3.6.6 Output Unit

The output unit collects the information requested by the controller. It processes such information in line with the SCPI rules and makes it available in the 1024-character output buffer. If the requested information is longer it will be made available in blocks in a way that is not noticeable to the controller.
If the UPL is addressed as a talker and the output buffer does not contain any data or expect data from the data set management, the output unit sends the error message "Query UNTERMINATED" to the status reporting system. No data will be sent on the IEC/IEEE bus and the controller waits for the set time limit. This procedure is prescribed by SCPI.

3.6.7 Triggering a Measurement/Sweep

A measurement or a sweep can be triggered in three different ways:

- With the SCPI-specific command `IEC OUT 20,"INIT"`
- With the Common Command `IEC OUT 20,"*TRG"`
- With the addressed command "**G**roup **E**xecute **T**rigger" `IEC LAD 20: IEC GET`

The three trigger modes take about the same execution time.

Universal Sequence Controller (UPL-B10)	RS-232 interface
See **3.16.4.4 Differences from the** IEC-bus Syntax - UPL-B10 The addressed GET command is illegal.	See **3.17.6 Differences to Remote** Control via IEC/IEEE Bus The addressed GET command is illegal.

Another three commands are available for waiting for a triggered measurement result. They are described in section 3.6.8.3 Comparison of Synchronization Capabilities. The synchronization command **"*WAI"** is used in the program examples below and in 3.15.15.1 Command Synchronization with *WAI for demonstrating the three trigger modes for a single measurement.

3.6.8 Command Synchronization

There are two events in the UPL, which have to be waited for before the next command can be executed to make sure that subsequent commands meet clear conditions:

- End of a calibration
- End of a measurement (measurement result ready to be fetched)

Through programming, the controller can be forced to wait for the end of a calibration or measurement (see table 3-2 Comparison of synchronization capabilities). Commands `"*OPC"`, `"*OPC?"` or `"*WAI"` are used to ensure that a calibration or measurement is completed before a new command is sent (see section 3.6.8.3 Comparison of Synchronization Capabilities).
In the examples below synchronization by means of the *WAI command is described.

3.6.8.1 Wait for End of Calibration

Wait for the calibration to be completed before sending the next setting command:

When analog analyzer instruments are called up with commands
```
IECOUT 20,"INSTrument2 A22;*WAI"
IECOUT 20,"INSTrument2 A110;*WAI"
```

with calibration switched on in the current setup ("`CALibrate:ZERO:AUTO ON`")
or upon loading a setup with commands

```
IECOUT 20,"MMEMory:LOAD:STATe 0, 'filename';*WAI"
          (current setup)
IECOUT 20,"MMEMory:LOAD:STATe 2, 'filename';*WAI"
          (complete setup)
IECOUT 20,"*RST;*WAI"
          (default setup)
```

with calibration switched on in the setup to be loaded ("`CALibrate:ZERO:AUTO ON`")
or after calling up a calibration with commands

```
IECOUT 20,"CALibrate:LDG:AUTO ONCE;*WAI"  or "CALibrate LDG;*WAI"
IECOUT 20,"CALibrate:ZERO:AUTO ON;*WAI"   or "CALibrate AUTO;*WAI"
IECOUT 20,"CALibrate:ZERO:AUTO ONCE;*WAI" or "CALibrate DCC;*WAI"
```

Without this forced synchronization, a subsequent generator or analyzer setting command would trigger a new calibration procedure.

3.6.8.2 Wait for End of Measurement/Sweep

Note:

In the explanations below the term measurement result denotes either a single measurement result or a sequence of results obtained in a single sweep.

A measurement or a sweep can be triggered by means of INIT , *TRG or GET (see section 3.6.8.2 Wait for End of Measurement/Sweep).
INIT will be used in the examples below.

With IEC/IEEE-bus commands assuming a settled measurement result, synchronization with *WAI, *OPC? or *OPC must be effected by means of an SRQ. The use of the three synchronization methods is described in section 3.6.8.3 Comparison of Synchronization Capabilities.
*WAI will be used in the examples below.

The effect of the various combinations of measurement or sweep trigger with a synchronization command is described in the table below.

Table 3-1 Trigger command with/without synchronization

Trigger command with/without synchronization	Description
`"INIT:CONT OFF;*WAI"`	A single measurement or sweep is triggered. Subsequent commands are processed after the measurement or sweep is completed. Generator and analyzer setting commands **do not** trigger a new measurement. A new measurement has to be triggered with `IECOUT 20,"INIT;*WAI"`. **This command is the simplest to use and should preferably be used for triggering a measurement**.
`"INIT:CONT OFF"` **without** `*WAI`	When a generator or analyzer setting command is sent **during a single measurement**, ie before the measurement is completed, the measurement is restarted to avoid incorrect results (incorrect results could be obtained, for instance, when the generator voltage is varied during the measurement). Generator or analyzer setting commands sent **after the single measurement has been terminated** will **not** trigger a new measurement. A new measurement has to be triggered with `IECOUT 20,"INIT"` .
`"INIT:CONT ON;*WAI"`	Generator and analyzer setting commands will only be processed when the measurement is terminated. They will **not** trigger a new measurements. A new measurement has to be triggered with `IECOUT 20,"INIT;*WAI" .`
`"INIT:CONT ON"` **without** `*WAI`	Each generator or analyzer setting command restarts a measurement to avoid incorrect measurement results (incorrect results could be obtained, for instance, when the generator voltage is varied during the measurement).

Note:

A synchronization of commands other than analyzer, generator, INIT, status loading or calibration commands is ineffective and should be avoided considering that synchronization commands are .

3.6.8.3 Comparison of Synchronization Capabilities

Table 3-2 Comparison of synchronization capabilities

Command	Action after hardware settling	Controller programming

`*OPC?`	Upon completion of the calibration or after a measurement result has been obtained, command `"*OPC?"` causes "1" to be entered in the output buffer. The "1" is irrelevant - the method is based on the fact that command IECIN 20,A$ stops the program run until the buffer has a "1".
	This method is therefore not suitable to wait for a "1" in the program loop. This is possible with *OPC.
		`IECOUT 20,"INST2 A22"`
		`IECOUT 20,"*OPC?`
		`IECIN 20,A$`

`*WAI`	Upon completion of the calibration or after a measurement result has been obtained, the next command is executed.	`IECOUT 20,"INST2 A22;*WAI"`
`*OPC`	Upon completion of the calibration or after a measurement result has been obtained, the Operation Complete bit in the Event Status Register (ESR) is set which triggers an SRQ if bit 5 in the Status Enable Register is set.	- Set bit 0 in the ESE - Set bit 5 in the SRE - Wait for Service Request (SRQ) **Advantage over "*OPC?" and "*WAI"**: While waiting for SRQ with the Operation Complete bit in the Event Status Register set, the program may carry out other tasks.

Universal Sequence Controller (UPL-B10)	RS-232 interface
See **3.16.4.4 Differences from the** IEC-bus Syntax - UPL-B10 *OPC cannot be used.	See **3.17.6 Differences to Remote** Control via IEC/IEEE Bus *OPC cannot be used.

Detailed examples for command synchronization are given in section 3.15.15 Command Synchronization

3.7 Status Reporting System

The status reporting system (see Fig. 3-6, Overview of Status Register) stores all information on the current operating status of the UPL, eg AUTORANGE being performed, and on errors. Such information is stored in the status registers and in the error queue. The contents of the status registers and of the error queue can be queried via the IEC/IEEE bus.

The information is hierarchically structured. The topmost level is formed by the Status Byte Register (STB) defined by IEEE 488.2 and the associated mask register Service Request Enable (SRE). The STB receives its information from the Standard Event Status Register (ESR) also defined in IEEE 488.2 and the associated mask register Standard Event Status Enable (ESE) as well as from the SCPI-defined STATus:OPERation and STATus:QUEStionable registers and the UPL-specific register STATus:XQUEstionable, which contain detailed information on the UPL.

The status reporting system also includes the IST flag (Individual STatus) and the Parallel Poll Enable Register (PPE) assigned to it. The IST flag, just as SRQ, combines the complete device status in a single bit. The PPE for the IST flag has an analog function like the SRE has for the Service Request.

3.7.1 Structure of SCPI Status Register

The STATus:OPERation Register and the STATus:QUEStionable Register (see section 3.7.3.4 STATus:OPERation Register and) consists of five registers of 16 bits each with different functions (See Fig. 3-5). The individual bits are independent of each other, ie each hardware status is assigned a bit number which is the same for all five registers. Bit 5 of the STATus OPERation Registers, for instance, is assigned in all five registers to the hardware status "wait for trigger". Bit 15 (the most significant bit) is set to zero in all registers. Thus the contents of the registers can be processed by the controller as a positive integer.

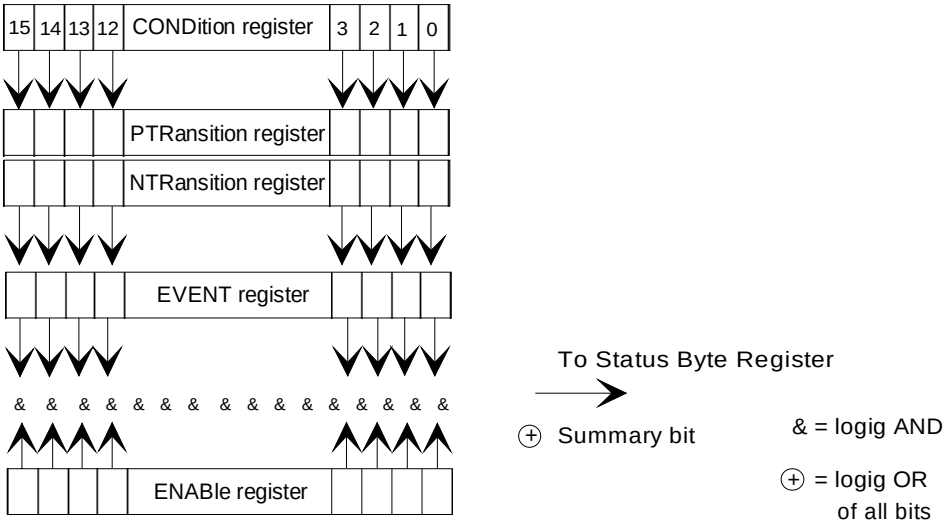


Fig 3-5 Status register model

CONDition register The CONDition register is directly written to by the hardware or the summary bit of the next lower register. Its contents reflects the current device status. This register can only be read, but neither written to nor cleared. After an SRQ has been caused by an entry in one of the status registers, reading the CONDition register does **not** trigger another SRQ. Further SRQs are only possible when the EVENT register of the status register is read.

Reading the register does not change its contents.

PTRansition register The Positive TRansition register acts as a transition filter. Upon transition of a bit of the CONDition register from 0 to 1, the associated PTR bit decides whether the EVENt bit will be set to 1
PTR bit = 1: the EVENt bit is set.
PTR bit = 0: the EVENt bit is not set.
This register can be written and read.
Reading the register does not change its contents.

NTRansition register

The Negative TRansition register also acts as a transition filter. Upon transition of a bit of the CONDition register from 1 to 0, the associated NTR bit decides whether the EVENt bit is set to 1.
NTR bit = 1: the EVENt bit is set.
NTR bit = 0: the EVENt bit is not set.
This register can be written and read.
Reading the register does not change its contents.

With the aid of these two transition registers the user can define the status change of the CONDition register (none, 0 to 1, 1 to 0 or both) that is to be stored in the EVENt register.

EVENt register

The EVENt register reports whether an event has occurred since its last reading, it is the memory of the CONDition register. It registers events that have been reported by the transition registers. The EVENt register is continuously updated by the instrument. It can only be read by the user. **Reading this register clears its contents.** After an SRQ has been caused by an entry in a status register, another SRQ is only possible when the EVENt register of the status register is read. Reading the CONDition register does **not** cause another SRQ.

This register is frequently referred to as the overall register.

ENABle register

The ENABle register determines whether the EVENt bit affects the summary bit (see below). Each bit of the EVENt register is ANDed (symbol '&') with the associated ENABle bit. The events of all logical operations of this register are ORed (symbol '+') and passed on to the summary bit.
ENAB bit = 0: the associated EVENt bit does not affect the summary bit.
ENAB bit = 1: if the associated EVENt is "1", the summary bit is also set to "1".
This register can be written and read by the user.
Reading the register does not change its contents.

Summary bit

As stated above, the summary bit for each register is derived from the EVENt and the ENABle registers. The result is entered into a bit of the CONDition register of the next higher register.
The instrument automatically generates the summary bit for each register. An event, eg a non-locking PLL, may thus cause a service request through all hierarchical levels.

Note:

The Service Request Enable Register SRE defined in IEEE 488.2 may be considered as the ENABle register of the STB provided that the STB is configured in conformance with SCPI. Accordingly, the ESE may be considered as the ENABLe register of the ESR.

3.7.2 Overview of Status Register

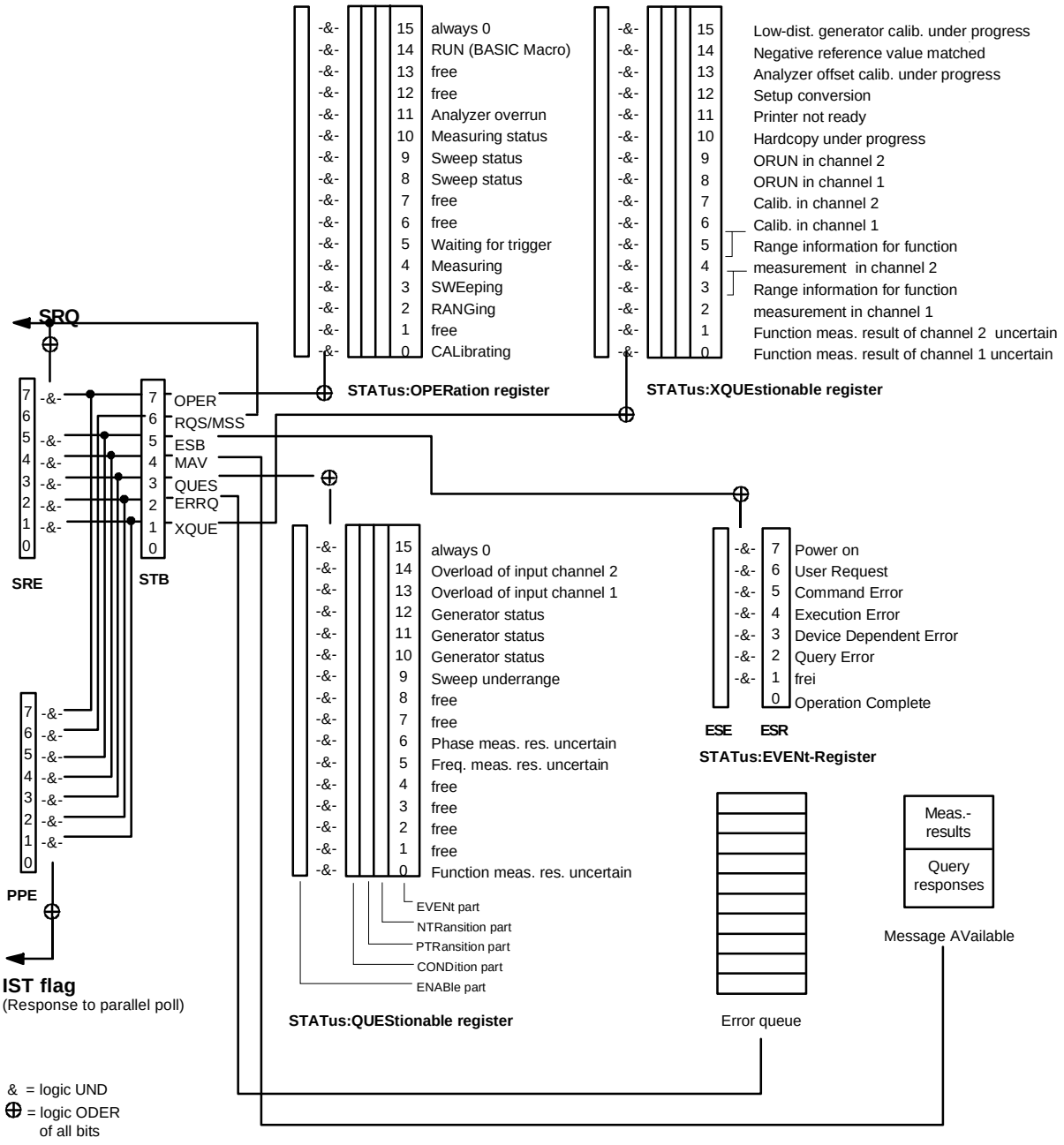


Fig. 3-6 Overview of Status Register

Universal Sequence Controller (UPL-B10)	RS-232 interface
See **3.16.4.4 Differences from the** IEC-bus Syntax - UPL-B10 List of usable commands of the status reporting system.	See **3.17.6 Differences to Remote** Control via IEC/IEEE Bus List of usable commands of the status reporting system.

3.7.3 Description of Status Registers

3.7.3.1 Status Byte (STB) and Service Request Enable Register (SRE)

The STB is already defined in IEEE 488.2. It provides a rough overview of the UPL status, collecting information from the lower-level registers. It is comparable with the CONDition register of a SCPI-defined register and is at the highest level of the SCPI hierarchy. Its special feature is that bit 6 acts as the summary bit of all other bits of the Status Byte Register.
The Status Byte Register is read out by the query *STB? or a Serial Poll .

The SRE is associated with the STB. The function of the SRE corresponds to that of the ENABle register of the SCPI registers. Each bit of the STB is assigned a bit in the SRE. Bit 6 of the SRE is ignored. If a bit is set in the SRE and the associated bit in the STB changes from 0 to 1, a Service Request (SRQ) will be generated on the IEC/IEEE bus, which triggers an interrupt in the controller configured for this purpose, and can be further processed by the controller.
The SRE can be set by the command *SRE and read out by the query *SRE?.
Its contents are <u>not</u> set to zero during reading.
If the EVENt part of a status register is read, the associated bit in the status byte register is deleted (eg STAT:OPER:EVEN? deletes the OPER bit (d7) in the OPERation register).

Table 3-3 Definition of bits used in the Status Byte Register

Bit No.	Definition
0	**Free**
1	**XQUEstionable Status summary bit** This bit is set if in the XQUEstionable Status Register an EVENt bit is set and the associated ENABle bit is set to 1. **The query command of the XQUEstionable Status Registers is "STATus:XQUEstionable:CONDition?" or "STATus:XQUEstionable[:**
2	**Error Queue** EVENt]?" The Error Queue contains an entry to be read with command "SYSTem:ERRor?" followed by IECIN.
3	**QUEStionable Status summary bit** This bit is set if in the QUEStionable Status Register an EVENt bit is set and the associated ENABle bit is set to 1. A set bit denotes a questionable device status which can be specified in greater detail by querying the QUEStionable Status Registers with "STATus:QUEStionable:CONDition?" or "STATus:QUEStionable[:EVENt]?"
4	**MAV** A measurement result, response to a query (IEC/IEEE-bus command with '?') or Error Queue entry is available and can be read with an IECIN command.
5	**ESB bit** Summary bit of the Event Status Register. This bit is set if one of the bits in the Event Status Register is set and enabled in the Event Status Enable Register. Setting of this bit denotes a serious error which can be specified in greater detail by querying the Event Status Registers with "*ESR?".
6	**MSS bit** (master status summary bit) This bit is set if the UPL triggers a service request. This is the case if one of the other bits of this register is set together with its mask bit in the Service Request Enable Register SRE.
7	**OPERation Status Register summary bit** This bit is set if an EVENt bit is set in the OPERation Status Register and the associated ENABle bit is set to 1. A set bit denotes that an action is just being performed by the UPL. Information on the type of the action can be obtained by querying the OPERation Status Register with "STATus:OPERation:CONDition?" or "STATus:OPERation[:EVENt]?".

Universal Sequence Controller (UPL-B10)	RS-232 interface
See **3.16.4.4 Differences from the** IEC-bus Syntax - UPL-B10 Cannot be used	See **3.17.6 Differences to Remote** Control via IEC/IEEE Bus Cannot be used

3.7.3.2 IST Flag and Parallel Poll Enable Register (PPE)

Similar to the SRQ, the IST flag (**I**ndividual **St**atus Flag) combines the complete status information in a single bit. It can be queried by a parallel poll (see section 3.7.4.3 Parallel Poll, and 3.15.16.1 SRQ Interrupt Routine with Serial Poll or with "*IST?".

The Parallel Poll Enable Register (PPE) determines which bits of the STB affect the IST flag. The bits of the STB are ANDed with the corresponding bits of the PPE, bit 6 - in contrast to the SRE - being used too. The IST flag is obtained by ORing all results together. The PPE can be set by the command *PRE and read by the query *PRE?.

Universal Sequence Controller (UPL-B10)	RS-232 interface
See **3.16.4.4 Differences from the** IEC-bus Syntax - UPL-B10 Cannot be used	See **3.17.6 Differences to Remote** Control via IEC/IEEE Bus Cannot be used

3.7.3.3 Definition of bits used in the Event Status Register

The ESR is already defined in the IEEE 488.2 standard. It is comparable to the EVENt register of an SCPI register. The Event Status Register can be read out by the query *ESR?.
The ESE forms the associated ENABle register. It can be set by the command *ESE and read out by the query *ESE?.

Table 3-4 Definition of bits used in the Event Status Register

Bit No.	Definition
0	**Operation Complete** Upon reception of the "*OPC" command this bit is set exactly when all previous commands have been executed.
1	**Free**
2	**Query Error** This bit is set if the controller wants to read data from the instrument but has not sent a data request command, or if the controller does not fetch the requested data but sends instead a new command to the instrument. A frequent cause is a faulty query which cannot be executed.
3	**Device-dependent Error** This bit is set if a device-dependent error occurs. An error message with a positive number denoting the error in greater detail in plain text (see **3.14 List of Error Messages**) will be entered into the Error Queue.
4	**Execution Error** This bit is set if the syntax of the command received is correct but the command cannot be executed due to various marginal conditions. An error message with a number between -200 and -300 describing the error in greater detail (see **3.14 List of Error Messages**) will be entered into the Error Queue.
5	**Command Error** This bit is set if an undefined command or a command with incorrect syntax is received. An error message with a number between -100 and -200 describing the error in greater detail (see **3.14 List of Error Messages**) will be entered into the Error Queue.
6	**User Request** This bit is set upon pressing the [LOCAL] key, ie when the instrument is switched to manual control.
7	**Power On** This bit is set upon power on of the instrument.

3.7.3.4 STATus:OPERation Register

The CONDition part of this register contains information on the operations currently performed by the UPL and the EVENt part on the operations performed by the UPL since the last readout of the register. The register can be read by the commands
`"STATus:OPERation:CONDition?"` or
`"STATus:OPERation[:EVENt]?"`.

Table 3-5 Definition of bits used in the STATus:OPERation register

Bit No.	Definition
0	**CALibrating** This bit is set as long as a calibration is performed by the UPL.
1	**Free**
2	**RANGing** This bit is set as long as the UPL performs a range change (eg. autorange).
3	**SWEeping** This bit is set while the UPL is performing a sweep. Bits 8 and 9 inform on the current sweep state.
4	**MEASuring** This bit is set while the UPL is performing a measurement. Bit 10 informs on the current measurement state.
5	**WAIT for TRIGGER** This bit is set while the UPL waits for a trigger event.
6 - 7	**Free**
8 and 9	**Sweep Status** Bit 3, 9, 8 0 0 0 = Sweep OFF — No sweep performed 0 0 1 = Sweep TERMINATED — Single sweep completed 0 1 0 = Sweep STOPPED — Sweep has been stopped and may be continued 0 1 1 = Sweep INVALID — Sweep invalid as not yet started 1 0 0 = Sweep MANU RUNNING — Manual sweep running 1 0 1 = Sweep SNGL RUNNING — Single sweep running 1 1 0 = Sweep CONT RUNNING — Continuous sweep running 1 1 1 = Sweep OFF
10	**Measuring Status** Bit 4, 10 0 0 = Measuring TERM — Single measurement completed 0 1 = Measuring STOP — Measurement stopped 1 0 = Measuring SNGL — Single measurement running 1 1 = Measuring CONT
11	**Analyzer Overrun** The sampling rate applied to the external input is too high for the digital meter selected. Remedy: - Select a lower external sampling rate and set function again - Select a fast digital meter.
12,13	**Free**
14	**RUN (BASIC-Macro)** If a BASIC macro (see **2.16 Macro Operation**) is started with the command SYST:PROG:EXEC 'filename.bas', this bit is set to 1. This bit is set to 0 when the program has been quit. This 1→0 transition can be queried via serial poll or initiate an SRQ, eg to fetch measurement data (for a detailed example see **3.15.18 Call BASIC-Macro**
15	**Free**

3.7.3.5 STATus:QUEStionable Register

This register contains information on questionable device states. These may for instance occur if the UPL is operated out of specifications. The register can be read by the commands "STATus:QUEStionable:CONDition?" or "STATus:QUEStionable[:EVENt]?".

Table 3-6 Definition of bits used in the STATus:QUEStionable Register

Bit No.	Definition
0	**Function** This bit is set when the result of a function measurement is questionable.
1 - 4	**Free**
5	**FREQuency** This bit is set when the result of a frequency measurement is questionable.
6	**PHASe** This bit is set when the result of a phase measurement is questionable.
7 - 8	**Free**
9	**Sweep underrange** During the sweep, valid but inaccurate measurement results were obtained because of underranging.
10 - 12	**Generator status** Bit 12, 11, 10 0 0 0 = Generator OFF — Both generator channels are switched off 0 0 1 = Generator OFF 0 1 0 = Generator RUNNING — Generator outputs a signal 0 1 1 = Generator BUSY — Generator DSP computes a waveform 1 0 0 = Generator HALTED — No generator output signal due to incomplete or invalid setting 1 0 1 = Generator OVERRUN — The sampling rate applied to the external input is too high for the digital generator selected Remedy: - Select lower sampling rate, call up function again. - Select faster digital generator 1 1 0 = Generator OFF 1 1 1 = Generator OFF
13 -14	**Analyzer Status** Bit 14, 13 0 0 = — Normal working order (no overload at any channel) 0 1 = ANL 1: OVLD — Overload at input channel 1 1 0 = ANL 2: OVLD — Overload at input channel 2 1 1 = ANL 1: OVLD 2: OVLD — Overload at both input channels With input impedance of 300Ω or 600Ω one or both input channels are overloaded! To protect the analyzer input against damages the input impedance temporarily is switched to 200 kΩ and the generator output is turned off. Measurement will be continued. **Ways to recover from overload:** - remove overload (input voltage at 300Ω or 600Ω must not exceed 25 V) or - set input impedance to 200kΩ (INP:IMP R200K). Reactivate generator output by sending command "OUTPut ON". C.f. **2.13 Fast Switch-off of Outputs**. The overload protection of the analyzer inputs is valid for analog board versions ≥ 4.00 and software version ≥ 1.0.
15	**Free**

3.7.3.6 STATus XQUEStionable Register

This register contains additional information for the Status Operation Register and information on rarely occurring states. It can be read by the queries "`STATus:XQUEStionable :CONDition?`" or "`STATus:XQUEStionable[:EVENt]?`".

Table 3-7 Definition of bits used in the STATus:XQUEstionable Register

Bit No.	Definition
0	Bit set: Information of MEASuring bit (d4) and Measuring Status bit (d10) of Status Operation Register refers to **channel 1**.  MEASuring bit (d4) Measuring status bit (d10) 0 0 = Measuring TERM — Single measurement terminated on 0 1 = Measuring STOP — Measurement stopped on 1 0 = Measuring SNGL — Single measurement in progress on 1 1 = Measuring CONT — Continuous measurement in progress on
1	Bit set: Information of MEASuring bit (d4) and Measuring Status bit (d10) of Status Operation Register refers to **channel 2**. Analogous to bit No. 0.
2 - 3	Additional range information for **channel 1** d3 d2 0 0 = Measurement result valid 0 1 = Ranging in channel 1. Bit d2 (RANGing) of STATus OPERation Register set simultaneously. 1 0 = Underrange (measurement result inaccurate) in channel 1 1 1 = Overrange (measurement result invalid) in channel 1 Function bit d0 (function measurement result uncertain) of the Status Questionable Register set simultaneously with underrange and overrange bits.
4 - 5	Additional range information for **channel 2** d3 d2 0 0 = Measurement result valid 0 1 = Ranging in channel 2. Bit d2 (RANGing) of STATus OPERation Register set simultaneously. 1 0 = Underrange (measurement result inaccurate) in channel 2 1 1 = Overrange (measurement result invalid) in channel 2 Function bit d0 (function measurement result uncertain) of the Status Questionable Register set together with the underrange and overrange bits.
6	Bit set: The calibration denoted in bit 13 or 15 is in progress in **channel 1.** No measurements are performed during this period.
7	Bit set: The calibration denoted in bit 13 or 15 is in progress in **channel 2.** No measurements are performed during this period.
8	Bit set: The sampling rate at the external input of **channel 1** is too high for the digital instrument. Remedy: Set lower rate and recall measurement function.
9	Bit set: Ditto for **channel 2**
10	Bit set: Read-out of screen content to a printer or a file. Operation of UPL is disabled for a few seconds.
11	Bit set: Printer not ready.
12	Bit set: A previous SETUP is converted to the latest status.
13	Bit set: Calibration of analyzer offset in progress. No measurement results are output during this period.
14	Bit set: A negative reference value is limited to $+10^{-10}$ for the dBr calculation.
15	Bit set: Calibration of low-distortion generator in progress. No measurement results are output during this period.

3.7.4 Use of Status Reporting System

For an efficient use of the status reporting system, the information contained therein has to be transferred to the controller and further processed. There are various methods which are described in the following. Detailed program examples are given in **3.15 Examples of IEC/IEEE-Bus Programming**.

3.7.4.1 Service Request, Use of Hierarchical Structure

Under certain conditions, the UPL may send a service request (SRQ) to the controller. This service request usually causes an interrupt at the controller to which the controller program can respond by a suitable action. As shown in Fig. 3-6 (Section 3.7.2 Overview of Status Register), a SRQ will always be triggered if one or several of the bits 3, 5 or 7 have been set in the Status Byte Register and enabled in the SRE. Each of these bits combines the information from a further status register. By setting the ENABle registers of the status registers accordingly, any bit in any status register will be able to trigger a SRQ. To utilize the possibilities of the service request, all bits in the enable registers SRE and ESE should be set to "1".

Examples:
(see also Fig 3-6 , Section 3.7.2 Overview of Status Register):

Use command `"*OPC"` for generating an SRQ. While waiting for the SRQ, the program may perform other tasks.

- Setting bit 0 (Operation Complete) in the ESE
- Setting bit 5 (ESB) in the SRE

Upon completion of the settings, the UPL generates an SRQ.

Indicating the end of a sweep by an SRQ via bit 3 in the STATus OPERation Register. While waiting for the SRQ the program may perform other tasks.

- Setting bit 7 (summary bit of STATus:OPERation Register) in the SRE
- Setting bit 3 (Sweep Terminated) in the STATus:OPERation:ENABle Register.
- Setting bit 3 in the STATus:OPERation:NTRansition to ensure that the transition of sweeping bit 3 from 1 to 0 (Sweep-Terminated) is also stored in the EVENt register. Calling up the *CLS command causes all bits of the NTRansition and PTRansition to be set to 1 so that any bit change is recorded. Enabling the desired enable bit, in this case bit 3, will normally be sufficient.

After having completed the sweep, the UPL generates an SRQ.

The SRQ is thus the only way for the UPL to become active of its own. A controller program should set the UPL so that a service request will be generated in case of malfunctions. The program should suitably respond to the service request. A detailed example of a service request routine is given in section 3.15.15 Command Synchronization.

Universal Sequence Controller (UPL-B10)	RS-232 interface
See **3.16.4.4 Differences from the** IEC-bus Syntax - UPL-B10 Cannot be used	See **3.17.6 Differences to Remote** Control via IEC/IEEE Bus Cannot be used

3.7.4.2 Serial Poll

Serial Poll is mainly used for obtaining a quick overview of the status of several devices connected to the IEC/IEEE bus
In the case of a Serial Poll, the status byte of a device can be queried with

```
IECOUT 20,"*STB?"
IEC IN 20,A%
```

However, querying the status byte is normally implemented by interface messages (see annex A, Interface Messages) which means that a single byte is set to the hardware.

The R&S BASIC command for the execution of a Serial Poll is

```
IEC SPL 20,A%
```

It is much quicker than the Common Command "*STB?".

The serial poll method has already been defined in the IEEE 488.1 standard and used to be the only standard method for querying the status byte of several devices. This method also works with instruments which neither conform to SCPI nor to IEEE 488.2.

Universal Sequence Controller (UPL-B10)	RS-232 interface
See **3.16.4.4 Differences from the** IEC-bus Syntax - UPL-B10 Cannot be used	See **3.17.6 Differences to Remote** Control via IEC/IEEE Bus Cannot be used

3.7.4.3 Parallel Poll

In the parallel poll mode up to eight devices are simultaneously requested by a command from the controller to transmit 1 bit of information on the assigned data line, ie to pull the assigned data line to logic 0 or 1. Similar to the SRE register which defines the conditions under which an SRQ will be generated, there is a Parallel Poll Enable Register (PPE), which is also ANDed bit by bit with the STB – taking into account bit 6. The result is ORed and is then returned (may be inverted) as a reply to a parallel poll of the controller. The result can also be read out without parallel poll by the query `*IST`.

The UPL must first be set for the parallel poll using the R&S BASIC command `IEC PCON`.
Example:

`IECPCON 20,1,6`: UPL identifies itself by a 1 on line 6.

This command assigns a data line to the device on which it sends an SRQ. The parallel poll itself is made by `IEC PPL Pp%`.
The parallel poll mode is mainly used to find out quickly which of the devices connected to the IEC/IEEE bus has caused an SRQ. For this purpose SRE and PPE must be set to the same value. A detailed example of parallel poll is given in section 3.15.16.1.2 Serial Poll SRQ Routine.

Universal Sequence Controller (UPL-B10)	RS-232 interface
See **3.16.4.4 Differences from the** IEC-bus Syntax - UPL-B10 Cannot be used	See **3.17.6 Differences to Remote** Control via IEC/IEEE Bus Cannot be used

3.7.4.4 Queries

Each individual register of a status register can be read out by queries. The individual queries are given in the detailed description of the registers in Section 3.7.3 Description of Status Registers. The queries always return a number representing the bit pattern of the queried register. This number is evaluated by the controller program.

Queries are mainly used after a SRQ to obtain detailed information on the cause for the SRQ.

3.7.4.5 Error Queue Query

- Each error condition in the instrument causes an entry in the error queue. The entries in the error queue are detailed error messages in plain text which can be read out via IEC/IEEE bus by the query `SYSTem:ERRor?`. Each query `SYSTem:ERRor?` returns an entry from the error queue. If there are no more error messages in the error queue, **0 = "No error"** is returned by the instrument
- The error queue should be queried in the controller program after each SRQ since the queue entries provide a more precise description of the error cause than the status registers. In particular in the test phase of a controller program the error queue should be queried at regular intervals since it also registers faulty commands from the controller to the UPL.

Note:

In addition to an entry in the Error Queue, each error causes a plain-text message to be output on the UPL display so that the IEC/IEEE-bus control program can be checked in the REMOTE control mode without reading out the Error Queue.

3.7.5 Resetting the Status Reporting Systems

Table 3-8 contains the various commands and events causing a reset of the status reporting system. None of the commands, with the exception of *RST affects the functional device setting. In particular DCL does not clear the device settings.

Table 3-8 Resetting of device functions

Event	Effect			
	DCL, SDC (Device Clear, Selected Device Clear)	***RST or SYSTem:PRESet**	**STATus:PRESet**	***CLS**
Clears STB, ESR	—	—	—	yes
Clears SRE, ESE	—	—	—	—
Clears PPE	—	—	—	—
Clears EVENt registers	—	—	—	yes
Clears ENABle registers of OPERation, QUESTionable and XQUEStionable Register, fills ENABle registers of all other registers with "1".	—	—	yes	—
Fills PTRansition registers with "1", clears NTRansition registers	—	—	yes	—
Clears Error Queue	—	—	—	yes
Clears output buffer	yes	1)	1)	1)
Clears command processing and input buffer	yes	—	—	—

1) Any command that is the first one in a command line clears the output buffer.

3.8 Notation of Command Table

All commands implemented in the UPL according to the command system are tabulated and described in detail in section **3.10 IEC-bus Commands**. The notation is in line with the SCPI standard provided the committee has defined a command for the required function.

Command table in section 3.10 IEC-bus Commands

Command: In this column the complete command without parameters is listed.

Parameter: Here the required parameters and their range of values are stated. If the command is only available in the form of a query, 'Query only' is marked in this column.

Basic unit: Basic unit of physical parameter.

Meaning: Brief description of command.

Section: Reference to the detailed function description in the case of manual operation, mainly section 2 of the UPL manual.

Upper/lower case Upper/lowercase characters are used to differentiate between the long form and the short form of the keywords of a command. The UPL itself does not distinguish between uppercase and lowercase letters.

Special character | For some commands there is a choice of keywords having the same effect. These keywords are stated in the same line and separated by a vertical bar. Only one of these keywords need to be stated in the header of the command. The effect of the command is independent of the keyword selected.

Example: `"SOURce:FREQuency:CW|:FIXed"`
(setting generator to a constant frequency of 1 kHz)

The two commands below have an identical effect
`"SOURce:FREQuency:CW 1E3"≡ "SOURce:FREQuency:FIXed 1E3"`

A vertical bar in the notation of the parameters is used to separate alternative options and is to be seen as "or". The effect of the command differs according to the parameter stated.

Example: Selection of parameters for the command
`"SENSe:VOLTage:UNIT V|DBV|DBU"`
(measurement result may be displayed in Volt, dBV or dBu)

Special characters [ ] Keywords in square brackets may be omitted in compound headers (see section 3.5.3 Structure of a Command Line). For reasons of compatibility with the SCPI standard, the instrument must be able to accept the full length of the command.
Parameters in square brackets may also be optionally inserted in the command or omitted.

Example: `"SENSe[1][:VOLTage|POWer]:REFerence:MODE ..."`
has the same effect as
`"SENSe:REFerence:MODE ..."`
(selecting the method for generating a level reference value)

Parameters in square brackets
may also be optionally inserted in the command or omitted.

Example: TRACe[1] stands for TRACe and TRACe1

TRACe**[1|2]** denotes that either TRACe1 or TRACe2 can be selected, causing **different settings.**
`DISPlay:TRACe[1|2]:MARKer MODE ...`
(markers for FFT spectrum display may be different for TRACe1 and TRACe2)

TRACe**[]** denotes that the command can be used for TRACe1 and TRACe2, causing the **same settings** in both cases.
`DISPlay:TRACe[]:CURSor[1]:MODE ...`
(selected cursor function apply to both traces!)

Special characters { } Parameters in curly brackets may be included in the command as often as required.

Example: `SENSe[1]:LIST:FREQuency <n>{,<n>}`
(frequencies of an RMS-selective sweep)

3.9 Common Commands

The common commands are based on the IEEE 488.2 (IEC 625.2) standard. A specific command has the same effect in different instruments. The headers of these commands consist of an asterisk "*" followed by three letters. Many common commands refer to the status reporting system described in detail in section **3.15 Examples of IEC/IEEE-Bus Programming**.

Table 3-9 Common Commands

Command	Brief description	Parameter / Notes	Universal Sequence Controller UPL-B10	RS-232 interface
***CLS**	Resets status byte	no query	Not usable	Not usable
***ESE**	Sets Event Status Enable Register	0 to 255	Not usable	Not usable
***ESR?**	Readout of content of Event Status Register	query only	Usable	Usable
***IDN?**	Identification query	ROHDE&SCHWARZ, UPL, 0, 2.xx	Usable	Usable
***IST?**	Query for content of IST flag	query only	Not usable	Not usable
***OPC**	Synchronization command		Usable	Usable
***PCB**	Address for passing back the IEC/IEEE-bus control	0 to 30, no query	Not usable	Not usable
***PRE**	Sets Parallel Poll Enable Register	0 to 255	Not usable	Not usable
***RST**	Resets the device to a defined default state. The parameter link (see **2.15.8 Transfer of Parameters** (Parameter Link Function)) is disabled to maintain the default setting described in Appendix **A UPL Default Setup**	no query	Usable	Usable
***SRE**	Sets Service Request Enable Register	0 to 255	Not usable	Not usable
***STB?**	Query for content of Status Byte	query only	Not usable	Not usable
***TRG**	Triggers measurement	no query	Usable	Usable
***TST?**	UPL selftest	query only	Usable	Usable
***WAI**	Synchronization command		Usable	Usable

***CLS**

CLEAR STATUS sets the status byte (STB), the Standard Event Register (ESR) and the EVENt part of the QUEStionable, OPERation and of the XQUEstionable Register to zero. The command has no effect on the mask and transition parts of the register. The output buffer is cleared.

***ESE 0 to 255**

EVENT STATUS ENABLE sets the Event Status Enable Register to the defined value. The query *ESE? returns the content of the Event Status Enable Registers in decimal form.

***ESR?**

EVENT STATUS ENABLE returns the contents of the Event Status Enable Register in decimal form (0 to 255) and clears the register.

***IDN?**

IDENTIFICATION QUERY for identification of the instrument.
The response is for example: "Rohde&Schwarz, UPL, 0, 2.xx"
UPL = instrument designation: 0 = serial number, 2.xx = firmware version

***IST?**

INDIVIDUAL STATUS QUERY returns the contents of the IST flag in decimal form (0 | 1).
The IST flag is the status bit sent during a Parallel Poll (see section 3.7.3.2 IST Flag and Parallel Poll Enable Register (PPE)).

***OPC**

OPERATION COMPLETE sets bit 0 in the Event Status Register if all preceding commands have been executed. This bit may be used to assert a Service Request (see section 3.6.8.3 Comparison of Synchronization Capabilities, 3.7.3.3 Event Status Reg. (ESR), Event Status Enable Reg. (ESE), and section 3.15.15.3 Command Synchronization with *OPC and SRQ

***OPC?**

OPERATION COMPLETE QUERY places an ASCII character "1" in the output buffer as soon as all preceding commands have been executed (see 3.6.8.3 Comparison of Synchronization Capabilities and section 3.15.15.3 Command Synchronization with *OPC and SRQ.

***PCB 0 to 30**

PASS CONTROL BACK notifies the address of the controller to which the IEC/IEEE-bus control is to returned.

***PRE 0 to 255**

PARALLEL POLL REGISTER ENABLE sets the Parallel Poll Enable Register to the defined value. The query *PRE? returns the contents of the Parallel Poll Enable Registers in decimal form

***RST**

RESET sets the UPL to a defined default state. The parameter-Link (see 2.15.8 Transfer of Parameters (Parameter Link Function)) is switched off to ensure that the default state as described in Annex A UPL Default Setup is maintained after a change of instrument or function.

***SRE 0...255**

SERVICE REQUEST ENABLE sets the Service Request Enable Register to the defined value. Bit 6 (MSS mask bit) remains 0. This command determines the conditions under which a Service Request will be asserted. The query *SRE? outputs the contents of the Service Request Enable Registers in decimal form.
Bit 6 is always 0.

***STB?**

READ STATUS BYTE QUERY outputs the contents of the status byte in decimal form.
Its contents are not set to zero during reading.
If the EVENt part of a status register is read, the associated bit in the status byte register is deleted (eg STAT:OPER:EVEN? deletes the OPER bit (d7) in the OPERation register).

***TRG**

TRIGGER starts all actions waiting for a trigger event.
See sections 3.6.7 Triggering a Measurement/Sweep; 3.6.8.2 Wait for End of Measurement/Sweep and section 3.15.8.1 Readout of Triggered Measurements.

***TST?**

SELF TEST QUERY causes a brief selftest of the UPL and outputs an error code in decimal form ('0' for ok., '1' for error)

***WAI**

WAIT-to-CONTINUE allows processing of commands only after all preceding commands have been executed, all signals settled and current measurements are terminated (see sections 3.6.8.2 Wait for End of Measurement/Sweep, 3.6.8.3 Comparison of Synchronization Capabilities, and section 3.15.15.1 Command Synchronization with *WAI.