3.11 Alphabetical List of IEC/IEEE-Bus Commands

Command	Parameter	Section
ABORt		**2.11** **Taste STOP/CONT**
ARM:**FREQ**uency:**STAR**t **ARM**:**FREQ**uency:**STOP**	**<nu>** Value range determined by instrument or function	**2.6.4** **ANLR-Panel** Start | Stop
ARM:**LEV**el:**MIN**	**<nu>** Analog instruments 10 mV to 1000 V Digital instrument 1µFS to 1.0 FS	**2.6.4** **ANLR-Panel** Min VOLT
ARM:**VOLT**age:**STAR**t **ARM**:**VOLT**age:**STOP**	**<nu>** Analog instruments 10 mV to 1000 V Digital instrument 1 mFS to 1.0 FS	**2.6.4** **ANLR-Panel** Start | Stop
CALCulate:**EQU**alize:**FEED**	**TRAC**e**1** **TRAC**e**2**	**2.9.1.2** **FILE-Panel** Volt Source → TRACE A → TRACE B
CALCulate:**EQU**alize:**INV**ert	**ON** **OFF**	**2.9.1.2** **FILE-Panel** Invert 1/n → ON → OFF
CALCulate:**EQU**alize:**NORM**freq	**<nu>** f_{min} to f_{max}	**2.9.1.2** **FILE-Panel** Norm Freq
CALCulate:**LIM**it:**FAIL?**	**<n>** Query only	**2.10.7** **keine Handbedienung**
CALCulate:**LIM**it:**LOW**er:**CONT**rol:**POIN**ts**?**	**<n>** Query only	**2.9.1.3** **No manual control**
CALCulate:**LIM**it:**LOW**er:**CONT**rol[:DATA]	**<n>**{**,<n>**}	**2.9.1.3** **No manual control**
CALCulate:**LIM**it:**LOW**er:**POIN**ts**?**	**<n>** 0 to 1023 Query only	**2.9.1.3** **No manual control**
CALCulate:**LIM**it:**LOW**er:**STAT**e	**ON** **OFF**	**2.10.7** **DISP-Panel** LIMIT CHECK Mode → LIM LOWER → OFF

Command	Parameter	Section
CALCulate:**LIM**it:**LOW**er:**TRAC**e	**<n>**{,**<n>**}	2.9.1.3 **No manual control**
CALCulate:**LIM**it:**LOW**er:**VAL**ue	**<nu>**	**2.10.7** **DISP-Panel** Lim Lower → VALUE:
CALCulate:**LIM**it:**LOW**er[:DATA]	**<n>**{,**<n>**}	**2.9.1.3** **No manual control**
CALCulate:**LIM**it:**ON**	**TRAC**e**1** **TRAC**e**2** **TR1A**nd2	**2.10.7** **DISP-Panel** Check → TRACE A → TRACE B → TRACE A+B
CALCulate:**LIM**it:**REP**ort:**POIN**ts**?**	**<n>** 0 to 1023 Query only	**2.9.1.3** **No manual control**
CALCulate:**LIM**it:**REP**ort[:DATA]**?**	**<n>**{,**<n>**} Query only	**2.9.1.3** **No manual control**
CALCulate:**LIM**it:**UPP**er:**CONT**rol:**POIN**ts**?**	**<n>** 0 to 1023 Query only	**2.9.1.3** **No manual control**
CALCulate:**LIM**it:**UPP**er:**CONT**rol[:DATA]	**<n>**{,**<n>**}	**2.9.1.3** **No manual control**
CALCulate:**LIM**it:**UPP**er:**POIN**ts**?**	**<n>** 0 to 1023 Query only	**2.9.1.3** **No manual control**
CALCulate:**LIM**it:**UPP**er:**STAT**e	**ON** **OFF**	**2.10.7** **DISP-Panel** LIMIT CHECK Mode → LIM UPPER Mode → OFF
CALCulate:**LIM**it:**UPP**er:**TRAC**e	**<n>**{,**<n>**}	2.9.1.3 **No manual control**
CALCulate:**LIM**it:**UPP**er:**VAL**ue	**<nu>**	**2.10.7** **DISP-Panel** Lim Upper → VALUE:
CALCulate:**LIM**it:**UPP**er[:DATA]	**<n>**{,**<n>**}	**2.9.1.3** **No manual control**

Command	Parameter	Section
CALCulate:**TRAN**sform:**FREQ**uency:**AVER**age	**<n>** 1 to 256	**2.6.5.12** **ANLR-Panel** Average
CALCulate:**TRAN**sform:**FREQ**uency:**AVER**age:**TCON**trol	**NORM**al **EXP**onential	**2.6.5.12** **ANLR-Panel** Avg Mode → NORMAL → EXPONENTIAL
CALCulate:**TRAN**sform:**FREQ**uency:**CENT**er	**<nu>** Value range determined by instrument or function	**2.6.5.12** **ANLR-Panel** Center
CALCulate:**TRAN**sform:**FREQ**uency:**FFT**	**S256** **S512** **S1K** **S2K** **S4K** **S8K**	**2.6.5.12** **ANLR-Panel** FFT Size → 256 → 512 → 1024 → 2048 → 4096 → 8192
CALCulate:**TRAN**sform:**FREQ**uency:**MTIM**e**?**	**<nu>** Query only	**2.6.5.12** **ANLR-Panel** Meas Time
CALCulate:**TRAN**sform:**FREQ**uency:**RES**olution**?**	**<nu>** Query only	**2.6.5.12** **ANLR-Panel** Resolution
CALCulate:**TRAN**sform:**FREQ**uency:**SPAN?**	**<nu>** Query only	**2.6.5.12** **ANLR-Panel** Span
CALCulate:**TRAN**sform:**FREQ**uency:**STAR**t**?** **CALC**ulate:**TRAN**sform:**FREQ**uency:**STOP?**	**<nu>** Query only	**2.6.5.12** **ANLR-Panel** Start / Stop
CALCulate:**TRAN**sform:**FREQ**uency:**STAT**e	**OFF** **ON**	**2.6.5.1** **ANLR-Panel** POST FFT → OFF → ON
CALCulate:**TRAN**sform:**FREQ**uency:**WIND**ow	**RECT**angular **HANN**ing **BLAC**kman_harris **RIF1** **RIF2** **RIF3** **HAMM**ing **FLAT**top **KAIS**er	**2.6.5.12** **ANLR-Panel** Window → RECTANG... → HANN → BLACKMAN H → RIFE VINC 1 → RIFE VINC 2 → RIFE VINC 3 → HAMMING → FLAT TOP → KAISER

Command	Parameter	Section
CALCulate:**TRAN**sform:**FREQ**uency:**WIND**ow:**BETA**factor	**<n>** = 1 to 20	**2.6.5.12** **ANLR-Panel** ß-Factor
CALCulate:**TRAN**sform:**FREQ**uency:**ZOOM**	**<n>** 1 to 128 for iInstrument A22 u. D48: n = 1,2,4,8,16,32,64,128 A110: n = 1,2,4,8,16 n **= 1**: **Zooming aus**	**2.6.5.12** **ANLR-Panel** Zoom-FFT
CALibrate	**OFF** Equivalent to CAL:ZERO:AUTO OFF **AUTO** Equivalent to CAL:ZERO:AUTO ON **DCC** Equivalent to CAL:ZERO:AUTO ONCE **LDG** Equivalent to CAL LDG:AUTO ONCE	2.15.6
CALibrate:**JITT**er:**AUTO**	**OFF** **ONCE**	**2.15.6** **OPTIONS-Panel** CALIBR. DIG PhaseToRef → OFF → ONCE
CALibrate:**LDG**:**AUTO**	**OFF** **ONCE**	**2.15.6** **OPTIONS-Panel** CALIBR. GEN Low Dist → OFF → ONCE
CALibrate:**ZERO**:**AUTO**	**OFF** **ON** **ONCE**	**2.15.6** **OPTIONS-Panel** CALIBR. ANL Zero Auto → OFF → ON → ONCE
CONFigure**:DAI**	**BRM** **HRM**	**2.15.9** **OPTIONS-Panel** Sampl Mode → BASE RATE → HIGH RATE
DISPlay:**ACT**ualize	**ON** **OFF**	**No manual control**

Command	Parameter	Section
DISPlay:**ANN**otation[:ALL]	**ON** **OFF**	**2.15.5** **OPTIONS-Panel** Meas Disp → ON → OFF Ext. Keyboard: CTRL D
DISPlay:**CONF**iguration	**P** **SP** **AP** **GP** **FP** **DP** **OP** **GAT** **GAO** **GAD** **FAT** **FAO** **FAD** **SHON** **SHOF**f	**2.3.1** Keys — Ext. UPL keyboard GEN — ALT+G ANLR — ALT+A FILT — ALT+T FILE — ALT+F DISP — ALT+D GRAPH — ALT+R ZOOM — ALT+Z SHOW I/O — ALT+I OPTIONS — ALT+O
DISPlay:**MODE**	**INT**ern **COLB**oth **BWB**oth **AUTO**	**2.15.5** **OPTIONS-Panel** Extrn Disp → INTERN ONLY → BOTH COLOR → BOTH B/W → BOTH AUTO
DISPlay:**PROT**ocol:**CHST**atus**?**	Query only Response:: **NO** **LTC** **YES**	**2.10.8** **GRAPH-Panel** **Display**
DISPlay:**PROT**ocol:**ERR**or:**GEN**eral**?**	Query only Response:: **UBB** **SQB** **NSYN** **PRMB** **SQLR** **RERR** **NONE**	**2.10.8** **GRAPH-Panel** **Display**
DISPlay:**PROT**ocol:**ERR**or:**LCRC?**	**<n>** Query only	**2.10.8** **GRAPH-Panel** **Display**
DISPlay:**PROT**ocol:**ERR**or:**PAR**ity**?**	**<n>** Query only	**2.10.8** **GRAPH-Panel** **Display**
DISPlay:**PROT**ocol:**ERR**or:**RCRC?**	**<n>** Query only	**2.10.8** **GRAPH-Panel** **Display**

Command	Parameter	Section
DISPlay:**PROT**ocol:**FORM**at	**BIN**ary **HEX**adecimal **ASCii** **FILE**	**2.10.8** **DISP-Panel** Format → BIN → HEX → ASCII → FILE DEF
DISPlay:**PROTocol**:**LR?**	Query only Response: **EQUAL** **DIFF'**	**2.10.8** **GRAPH-Panel** **Display**
DISPlay:**PROT**ocol:**LVAL**bit**?**	Query only Response: **Y0** **N1**	**2.10.8** **GRAPH-Panel** **Display**
DISPlay:**PROT**ocol:**RVAL**bit**?**	Query only Response:: **Y0** **N1**	**2.10.8** **GRAPH-Panel** **Display**
DISPlay:**PROT**ocol:**SEL**ect	**LCH**annelstatus **RCH**annelstatus **LUS**erdata **RUS**erdata	**2.10.8** **DISP-Panel** Source → CHAN STAT L → CHAN STAT R → USER DATA L → USER DATA R
DISPlay[:WINDow]:**TEXT**:**LOC**ate	<**ny**>[,<**nx**>]	**2.10.1** **DISP-Panel** X Pos, Y Pos
DISPlay[:WINDow]:**TEXT**[:DATA]	**'string'**	**2.10.1** **DISP-Panel** COMMENT
DISPlay[:WINDow]:**TRAC**e:**X**:**LAB**el	**'string'**	**2.10.2** **DISP-Panel** Unit/Label
DISPlay[:WINDow]:**TRAC**e[]:**AUT**oscale alias **AUTO**scale		**2.10.2** **Softkey** F7 (AUTOSCALE) → F6 (ALL)
DISPlay[:WINDow]:**TRAC**e[]:**COUN**t	**<n>** recorded: 1 to 100,000 stored: 17 traces max.	**2.10** **DISP-Panel** Scan Count
DISPlay[:WINDow]:**TRAC**e[]:**CURS**or[]:**DATA1?** **DISP**lay[:WINDow]:**TRAC**e[]:**CURS**or[]:**DATA2?** **DISP**lay[:WINDow]:**TRAC**e[]:**CURS**or[]:**DATA3?**	Query only	**2.10.2** **Display in graphics window**

Command	Parameter	Section
DISPlay[:WINDow]:**TRAC**e[]:**CURS**or[1]:**MODE**	**N12** **D12** **OFF**	**2.10.2** **Softkey** F8: sel. O-CURS. F9: (O-CURSOR) → F6 (A,B) → F7 (A-B) → F11 (ON/OFF)
DISPlay[:WINDow]:**TRAC**e[]:**CURS**or[1|**2**]	**ACT**ive	**2.10.2** **DISP-Panel** Softkey level 1 F8
DISPlay[:WINDow]:**TRAC**e[]:**CURS**or[1|**2**]:**POS**ition	**<nu>**	**2.10.2** nicht über Softkey bedienbar
DISPlay[:WINDow]:**TRAC**e[]:**CURS**or[1|**2**]:**POS**ition:**MODE**	**MIN1** **MIN2** **I MAX1** **MAX1** **I MAX2** **MAX2** **MARK**er1 **NEXT**marker **VAL**ue	**2.10.2** **Softkey** F8 sel. O-CURS or *****-CURS. → F10 (SET TO) → ---- → ---- → F6 (I MAX A) → F7 (MAX A) → F8 (I MAX B) → F9 (MAX B) → F10 (MARKER) → F11 (NXT HARM)
DISPlay[:WINDow]:**TRAC**e[]:**CURS**or**2**:**MODE**	**N12** **D12** **C12** **HL1** **HL2** **HLD1** **HLD2** **OFF**	**2.10.2** **Softkey** F8 sel. O-CURSOR F9 sel. *****-CURSOR → F6 (A,B) → F7 (A-B) → F8 (* - O) → F9 (HLINE) → A → F9 (HLINE) → B → F9 (HLINE) → ΔA → F9 (HLINE) → ΔB → F11 (ON/OFF)
DISPlay[:WINDow]:**TRAC**e[]:**IND**ex	**<n>** 1 to 17	**2.9.3.3** **Tasten** PAGE UP / PAGE DOWN
DISPlay[:WINDow]:**TRAC**e[]:**LAB**el	**ON** **OFF**	**2.10.2** **DISP-Panel** User Label → ON → OFF

Command	Parameter	Section
DISPlay[:WINDow]:**TRAC**e[]:**MODE**	**DELete_bef_wr** **WAT**erfall|**CASC**ade **MAXH**old	**2.10** **DISP-Panel** Mode → DEL BEF WR → WATERFALL → MAX HOLD
DISPlay[:WINDow]:**TRAC**e[]:**OPER**ation	**CURV**eplot **LIST** alias **TLIS**t **ERR**ors **BARG**raph **SPEC**trum **FFTL**ist **FFTE**rrors **PROT**ocol **AUT**oprotocol alias **AUTO**protocol	**2.10** **DISP-Panel** OPERATION → CURVE PLOT **2.10.2** → SWEEP LIST **2.10.4** → SWP LIM REP **2.10.4** → BARGRAPH **2.10.2** → SPECT LIST **2.10.8** → SPC LIM REP **2.10.6** PROTOCOL **2.10** **2.10.8** → PROTO AUTO
DISPlay[:WINDow]:**TRAC**e[]:**X**:**SPAC**ing	**LIN**ear **LOG**arithmic	**2.10.1** **DISP-Panel** Spacing → LIN → LOG
DISPlay[:WINDow]:**TRAC**e[]:**X**[:SCALe]:**AUTO**	**ON** **OFF**	**2.10.1** **DISP-Panel** Scale → AUTO → MANUAL or Softkey F7 (AUTOSCALE) → F9 (X)
DISPlay[:WINDow]:**TRAC**e[]:**X**[:SCALe]:**LEFT** **DISP**lay[:WINDow]:**TRAC**e[]:**X**[:SCALe]:**RIGHT**	**<nu>**	**2.10.1** **DISP-Panel** **Left** Right
DISPlay[:WINDow]:**TRAC**e[]:**X**[:SCALe]:**RLEV**el	**<nu>**	**2.10.1** **DISP-Panel** Reference → VAQLUE
DISPlay[:WINDow]:**TRAC**e[]:**X**[:SCALe]:**UNIT**	**V** **Hz** **s** and so on see **3.10.4 Units** for IEC Measurement Results	**2.10.1** **DISP-Panel** Unit

Command	Parameter	Section
DISPlay[:WINDow]:**TRAC**e[]:**Y**:**AUTO**	**ONCE** **OFF**	**2.10.2** **DISP-Panel** Scale → AUTO ONCE → MANUAL ONCE via Softkey F7 (AUTOSCALE) → F7 (A) → F8 (B)
DISPlay[:WINDow]:**TRAC**e[]:**Y**[:SCALe]:**RLEV**el	**<nu>**	**2.10.1** **DISP-Panel** Reference → VALUE
DISPlay[:WINDow]:**TRAC**e[]:**Y**[:SCALe]:**UNIT**	<**u**> see 3.10.4 Units for IEC Measurement Results	**2.10.1** **DISP-Panel** Unit
DISPlay[:WINDow]:**TRAC**e[]:**ZOOM**	**<n>** **0** **1** **–1** **2** **3** **4**	**2.10.2** **Softkey** F10 (ZOOM) → F10 (UNZOOM) → F6 (AT o UP) → F7 (AT o DOWN) → F8 (CEN TO o) → F9 (o TO *) → F11 (UNDO)
DISPlay[:WINDow]:**TRAC**e[1]:**Y**:**LAB**el	**'string'**	**2.10.2** **DISP-Panel** Unit/Label
DISPlay[:WINDow]:**TRAC**e[1|**2**]:**CURV**e	**OFF** **ON**	**2.10.2** **Softkey** → F6 (CURVE) → F6 (A ON/OFF) → F7 (B ON/OFF)
DISPlay[:WINDow]:**TRAC**e[1|**2**]:**FEED**	**'SENS**e1:**DATA**1**'** **'SENS**e1:**DATA2'** **'SENS**e**2**:**DATA**1**'** **'SENS**e**2**:**DATA2'** **'SENS**e**3**:**DATA**1**'** **'SENS**e**3**:**DATA2'** '**HOLD'** **'FILE'** **'DFIL**e' **'OFF'**	**2.10.1** **DISP-Panel** TRACE A/B → FUNC CH1 → FUNC CH2 → INP RMS CH1 → INP RMS CH2 → FREQ CH1 → FREQ CH2 → PHASE → GROUP DEL → HOLD → FILE → DUAL FILE → OFF

Command	Parameter	Section
DISPlay[:WINDow]:**TRAC**e[1|**2**]:**MARK**er:**HARM**onics	**ON** **OFF**	**2.10.2** **Softkey** F11 (MARKER) F6 (TRACE A) or F7 (TRACE B) selects → F10 (HARM) on/off
DISPlay[:WINDow]:**TRAC**e[1|**2**]:**MARK**er:**MODE**	**MAX**imum **CURS**or **OFF**	**2.10.2** **Softkey** F11 (MARKER) F6 (TRACE A) or F7 (TRACE B) selects → MAX → CURSOR → VIEW OFF
DISPlay[:WINDow]:**TRAC**e[1|**2**]:**Y**:**SPAC**ing	**LIN**ear **LOG**arithmic	**2.10.1** **DISP-Panel** Spacing → LIN → LOG
DISPlay[:WINDow]:**TRAC**e[1|**2**]:**Y**[:SCALe]:**AUTO**	**ONCE** **OFF**	**2.10.1** **DISP-Panel** Scale → AUTO ONCE → MANUAL
DISPlay[:WINDow]:**TRAC**e[1|**2**]:**Y**[:SCALe]:**BOTT**om	**<nu>**	**2.10.1** **DISP-Panel** Bottom
DISPlay[:WINDow]:**TRAC**e[1|**2**]:**Y**[:SCALe]:**NORM**aliz e	**<nu>** 10^{-12} to 10^{6} or –200 dB to 120 dB	**2.10.1** **DISP-Panel** Normalize
DISPlay[:WINDow]:**TRAC**e[1|**2**]:**Y**[:SCALe]:**NORM**aliz e:**MODE**	**CURS**or[1] (**o**-Cursor) **CURS**or**2** (*****-Cursor) **VAL**ue	**2.10.1** **DISP-Panel** Normalize → **o**-Cursor → *****-Cursor → VALue

Command	Parameter	Section
DISPlay[:WINDow]:**TRAC**e[1|**2**]:**Y**[:SCALe]:**RLEV**el:**MODE**	**VAL**ue **MAX**imum **CURS**or[1] **CURS**or 2 **FILE** **HOLD** **OTRA**ce **CH1M**eas **CH2M**eas **GENT**rack **IFIL**e **REF9**97 **REF1**000	**2.10.1** **DISP-Panel** Reference → VALUE → MAX → oCURSOR → *CURSOR → FILE → HOLD → OTHER TRACE → MEAS CH1 → MEAS CH2 → GEN TRACK → FILE INTERN → REF 997 Hz → REF 1000 Hz
DISPlay[:WINDow]:**TRAC**e[1|**2**]:**Y**[:SCALe]:**TOP**	**<nu>**	**2.10.1** **DISP-Panel** Top
DISPlay[:WINDow]:**TRAC**e**2**:**Y**:**LAB**el	**'string'**	**2.10.2** **DISP-Panel** Unit/Label
DISPlay[:WINDow]:**TRAC**e**2**:**Y**[:SCALe]:**EQU**al	**ON** **OFF**	**2.10.1** **DISP-Panel** Scale B → EQUAL A → NOT EQUAL A
FORMat[:DATA]	**ASCI**i **REAL**	**No manual control**
HCOPy:**ABOR**t		**2.14** **Taste** H COPY or CTRL F8

Command	Parameter	Section
HCOPy:**DEST**ination	PRSPc alias PRINter PLHPgl alias PLOTter PRPS PRHPgl FIPCx, 'filename.PCX' alias PCXFile, 'name.PCX' FIHPgl, 'filename.GL' alias HPGLfile, 'name.GL' FIPS, 'filename.PS' FIEPs, 'filename.EPS' Query response: PRIN PLOT PRHP PRPS PCXF HPGL FIPS FIEP	**2.14** **OPTIONS-Panel** Destin (**Ziel/Format**) → PRINTR/SPC → PLOTTR/HPGL → PRINTR/HPGL → PRINTR/PS → FILE/PCX → FILE/HPGL → FILE/PS → FILE/EPS
HCOPy:**DEV**ice:**COL**or	**ON** **OFF**	**2.14** **OPTIONS-Panel** COLOR → ON → OFF
HCOPy:**DEV**ice:**PRIN**ter	**<n>**	**2.14** **OPTIONS-Panel** Printname
HCOPy:**DEV**ice:**RES**olution	**HIGH** **MED**ium **LOW**	**2.14** **OPTIONS-Panel** Prn Resol → HIGH → MEDIUM → LOW
HCOPy:**ITEM**	**ALL** **GRAT**icule **TRAC**e	**2.14** **OPTIONS-Panel** Copy → SCREEN → CURVE/GRID → CURVE
HCOPy:**ITEM**:**FRAM**e	**WHITe** **FDEF**ined	**2.14** **OPTIONS-Panel** Frame → WHITE → FILE DEF

Command	Parameter	Section
HCOPy:**ITEM**:**LAB**el:**STAT**e	**ON** **OFF**	**2.14** **Taste** H COPY or Ctrl F8
HCOPy:**PAGE**:**LENG**th**?**	**<n>** Query only	**2.14** **OPTIONS-Panel** Prn Height
HCOPy:**PAGE**:**LMAR**gin	**<n>** 0 to 80	**2.14** **OPTIONS-Panel** LEFT MRGN
HCOPy:**PAGE**:**ORI**entation	**LAND**scape **PORT**trait	**2.14** **OPTIONS-Panel** ORIENTATION → LANDSCAPE → PORTRAIT
HCOPy:**PAGE**:**SCAL**e:**X**	**<n>** 0.1 to 10	**2.14** **OPTIONS-Panel** X-SCALING
HCOPy:**PAGE**:**SCAL**e:**Y**	**<n>** 0.1 to 10	**2.14** **OPTIONS-Panel** Y-SCALING
HCOPy:**PAGE**:**WIDT**h**?**	**<n>** Query only	**2.14** **OPTIONS-Panel** Prn Width,
HCOPy:**PLAD**dress	**<n>** 0 to 31	**2.14** **OPTIONS-Panel** → IEC Adr
HCOPy:**PLOT**s	<n> 1 to 6	**2.14** **OPTIONS-Panel** Plots/Page
HCOPy:**PLP**ort	**COM2** **LPT1** **IEC**	**2.14** **OPTIONS-Panel** Plot on → COM 2 → LPT 1 → IEC BUS
HCOPy:**SIZE**	**A4** **LETT**er	**2.14** **OPTIONS-Panel** Paper Size → A4 → LETTER
HCOPy:**WAIT**		**2.14** **No manual control**
HCOPy[:IMMediate]	**CNF** **CF** **NCNF** **NCF** **CONF**ig	**2.14** **Taste** H COPY or CTRL F8

Command	Parameter	Section
INITiate:**CONT**inuous	**ON** **OFF**	**2.11** **Taste START** **Taste SINGLE**
INITiate:**FORC**e	**STAR**t **SING**le **STOP** **CONT**inuous	**2.11** → **Taste START** → **Taste SINGLE** → STOP function of toggel key **STOP/CONT** → CONT function of toggel key **STOP/CONT**
INITiate:**NEXT**	**<n>**	**2.11** **Drehrad**
INITiate[:IMMediate]		**2.11** **Taste START** **Taste SINGEL**
INPut:**FILT**er[:LPASs]:**FREQ**uency	**<n>** Query only **10** Hz | **20** Hz	**2.6.1** **ANLR-Panel** Min Freq
INPut[]:**AUD**iobits	**<n>** Value range see 2.6.3 Configuration of the Digital Analyzer	**2.6.3** **ANLR-Panel** Audio Bits
INPut[]:**SAMP**le:**FREQ**uency	**<nu>** Opt. UPL-B2 (Digital Audio I/O) 27 kHz to 55 kHz Opt. UPL-B29 im Base Rate Mode 40 kHz to 55 kHz Opt. UPL-B29 im High Rate Mode 40 kHz to 106 kHz	**2.6.3** **ANLR-Panel** Sample Frq → VALUE:
INPut[]:**SAMP**le:**FREQ**uency:**MODE**	**F32** **F44** **F48** **F88** **F96** **VAL**ue **AUTO** **CHST**atus	**2.6.3** **ANLR-Panel** Sample Frq → 32 kHz → 44.1 kHz → 48 kHz → 88.2 kHz → 96.0 kHz → VALUE: → AUTO → CHAN STATUS
INPut[]:**SEL**ect	**CH1** **CH2** **CH1A**nd2 **CH1I**s2 **CH2I**s1 **BOTH**	**2.6.2** **2.6.3** **ANLR-Panel** CHANNEL(s) → 1 → 2 → 1 & 2 → 1 ≡ 2 → 2 ≡ 1 → BOTH

Command	Parameter	Section
INPut[1|**2**]:**COUP**ling	**AC** **DC**	**2.6.2** **ANLR-Panel** CH1 Coupl CH2 Coupl → AC → DC
INPut[1|**2**]:**IMP**edance	**R300** **R600** **R200K**	**2.6.2** **ANLR-Panel** Imped → 300 Ω → 600 Ω → 200 kΩ
INPut[1|**2**]:**LOW**	**FLO**at **GRO**und	**2.6.2** **ANLR-Panel** Common → FLOAT → GROUND
INPut[1|**2**]:**TYPE**	**BAL**anced **GEN1** **GEN2** **AES**ebu **SPD**if **OPT**ical **INT**ern	**2.6.2** **2.6.3** **ANLR-Panel** Input → BAL XLR → GEN1 → GEN2 → GEN CROSSED → BAL (XLR) → UNBAL (XLR) → OPTICAL → INTERN
INSTrument[1]:**NSEL**ect	**1** **3**	**2.5.1** **GEN-Panel** INSTRUMENT → ANALOG → DIGITAL
INSTrument[1][:SELect]	**A25** **D48**	**2.5.1** **GEN-Panel** INSTRUMENT → ANALOG → DIGITAL
INSTrument**2**:**NSEL**ect	**1** **2** **4**	**2.6.1** **ANLR-Panel** INSTRUMENT → ANLG 22 kHz → ANLG 110 kHz → DIGITAL
INSTrument**2**[:SELect]	**A22** **A110** **D48**	**2.6.1** **ANLR-Panel** INSTRUMENT → ANLG 22 kHz → ANLG 110 kHz → DIGITAL
MMEMory:**CDIR**ectory	**'pathname'**	**2.9.2** **FILE-Panel** Work Dir

Command	Parameter	Section
MMEMory**:CHEC**k**?**	**'filename'**	**3.10.13** **3.15.20** **3.17.5** keine Handbedienung
MMEMory:**COPY**	**'filename1','filename2'**	**2.9.2** **FILE-Panel** Copy + To
MMEMory:**DATA**	**'filename'**	**3.10.13** **3.15.20** **3.17.5** keine Handbedienung
MMEMory:**DEL**ete	**'filename'**	**2.9.2** **FILE-Panel** Delete
MMEMory:**LOAD**:**LIST**	**LIMU**pper**, 'filename'** Query-Form MMEM:LOAD:LIST? LIMU	**2.10.7** **DISP-Panel** Lim Upper → FILE + filename
MMEMory:**LOAD**:**LIST**	**LIML**ower, **'filename'** Query-Form MMEM:LOAD:LIST? LIML	**2.10.7** **DISP-Panel** Lim Lower → FILE + filename
MMEMory:**LOAD**:**LIST**	**EQU**alize**,'filename'** Query: MMEM:LOAD:LIST? EQU	**2.5.4.3** **2.5.4.4** **GEN-Panel** Equal File → FILE + filename
MMEMory:**LOAD**:**LIST**	**ARB**itrary**,'filename'** Query: MMEM:LOAD:LIST? ARB **RAND**om**,'filename'** Query: MMEM:LOAD:LIST? RAND	**2.5.4.9** **2.5.4.10** **GEN-Panel** Shape File → FILE + filename
MMEMory:**LOAD**:**LIST**	**DWEL**l**,'filename'** Query: MMEM:LOAD:LIST? DWEL	**2.5.4.2** **GEN-Panel** Dwell File → FILE + filename
MMEMory:**LOAD**:**LIST**	**FREQ**uency[1|**2**]**,'filename'** Query-Form MMEM:LOAD:LIST? FREQ[1|**2**]	**2.5.4.2** **GEN-Panel** FREQ FILE → FILE + filename
MMEMory:**LOAD**:**LIST**	**FREQ**uency**,'filename'** Query-Form MMEM:LOAD:LIST? FREQ	**2.6.5.3** **2.9.1.3** **ANLR-Panel** SWEEP CTRL → FILE + filename

Command	Parameter	Section
MMEMory:**LOAD**:**LIST**	**INT**erval**,'filename'** Query: MMEM:LOAD:LIST? INT	**2.5.4.5** **2.5.4.6** **GEN-Panel** INTV FILE → FILE + filename
MMEMory:**LOAD**:**LIST**	**ONT**ime**,'filename'** Query: MMEM:LOAD:LIST? ONT	**2.5.4.5** **2.5.4.6** **GEN-Panel** ONTIM FILE → FILE + filename
MMEMory:**LOAD**:**LIST**	**VOLT**age[1|**2**]**,'filename'** Query: MMEM:LOAD:LIST? VOLT[1|**2**]	**2.5.4.2** **GEN-Panel** VOLT FILE → FILE + filename
MMEMory:**LOAD**:**LIST**	**DWEL**l**2,'filename'** Query: MMEM:LOAD:LIST? DWEL2	**2.5.4.2** Sweeps **GEN-Panel** AUX GEN: Dwell File → FILE + filename
MMEMory:**LOAD**:**LIST**	**FREQ**uency**2,'filename'** Query-Form MMEM:LOAD:LIST? FREQ2	**2.5.4.2** Sweeps **GEN-Panel** AUX GEN FREQUENCY FREQ FILE → FILE + filename
MMEMory:**LOAD**:**LIST**	**VOLT**age**2,'filename'** Query: MMEM:LOAD:LIST? VOLT2	**2.5.4.2** Sweeps **GEN-Panel** AUX GEN VOLTAGE | AMPL VOLT FILE → FILE + filename
MMEMory:**LOAD**:**LIST**	**EQU**alize**,'filename'.** Query: MMEM:LOAD:LIST? EQU	**2.5.4.4** **GEN-Panel** Equal.File → FILE + filename
MMEMory**:LOAD:LIST**	**EQU**alize,**'filename'** Query form: MMEM:LOAD:LIST? EQU	**2.5.4.1.1** **GEN-Panel** Equal.File
MMEMory:**LOAD**:**LIST SENS**e,	**'filename'**	**2.6.5.7** **2.6.5.12** **ANL Panel** Equal. file
MMEMory:**LOAD**:**LPGC,**	**'filename'**	**2.5.3.2** **GEN-Panel** Filename
MMEMory:**LOAD**:**PAC,**	**'filename'**	**2.10.8** **DISP-Panel** Proto File

Command	Parameter	Section
MMEMory:**LOAD**:**PAU,**	**'filename'**	**2.10.8** **DISP-Panel** Proto File
MMEMory:**LOAD**:**PGU,**	**'filename'**	**2.5.3.2** **GEN-Panel** Filname
MMEMory:**LOAD**:**RPGC,**	**'filename'**	**2.5.3.2** **GEN-Panel** Filename
MMEMory:**LOAD**:**STAT**e	**0**|**2**|**4, 'filename'** Query: MMEM:LOAD:STAT? 0 MMEM:LOAD:STAT? 2 MMEM:LOAD:STAT? 4	**2.9.1.1** **FILE-Panel** Mode / Filename
MMEMory:**LOAD**:**TRAC**e	**TRAC**e[1|**2**],**'filename'** Query: MMEM:LOAD:TRAC? TRAC[1|**2**]	**2.10.1** **DISP-Panel** TRACE A/B → FILE + Filename
MMEMory:**LOAD**:**TRAC**e[1|**2**]	**REFT**race,**'filename'** Query: MMEM:LOAD:TRAC[1|**2**]? REFT	**2.10.1** **DISP-Panel** Reference → FILE + Reference
MMEMory:**STOR**e:**FORM**at	**BIN** **ASC**ii **EXP**ort	**2.9.1.2** **FILE-Panel** Format → REAL → ASCII → EXPORT
MMEMory:**STOR**e:**INFO**text	**'string'**	**2.9.1.1** FILE-Panel Info Text
MMEMory:**STOR**e:**LIST**	**LIST**[**1**|**2**], **'filename'** **DWEL**l[1|**2**], **'filename'** **ERR**ors, **'filename'** **LIMU**pper, **'filename'** **LIML**ower, **'filename'** **EQU**alize, **'filename'** Query: MMEM:STOR:LIST? LIST[1|**2**] MMEM:STOR:LIST? DWEL[1|**2**] MMEM:STOR:LIST? LIMU MMEM:STOR:LIST? LIML MMEM:STOR:LIST? EQU	**2.9.1.2** **FILE-Panel** Store → X-Axis → Z-Axis → DWEL VALUE → LIM REPORT → LIM UPPER → LIM LOWER → EQUALIZATN
MMEMory:**STOR**e:**STAT**e	**0**|**2,'filename'** Query: MMEM:STOR:STAT? 0 MMEM:STOR.STAT? 2	**2.9.1.1** **FILE-Panel** Mode / Filename

Command	Parameter	Section
MMEMory:**STOR**e:**STAT**e:**RONL**y	**ON** **OFF**	**2.9.1.1** **FILE-Panel** Attrib → READ ONLY → READ/WRITE
MMEMory:**STOR**e:**TRAC**e	**TRAC**e[1|**2**],**'filename'** **TR1A**nd2,**'filename'** Query: MMEM:STOR:TRAC? TRAC[1|**2**] MMEM:STOR:TRAC? TR1A	**2.9.1.2** **FILE-Panel** Store → TRACE A → TRACE B → TRACE A+B
OUTPut	**ON** **OFF**	**2.13** Taste OUTPUT OFF
OUTPut:**AUD**iobits	**<n> =** 8 to 24	**2.5.3** **GEN-Panel** Audio Bits
OUTPut:**DIG**ital:**CSIM**ulator	**OFF** **SIML**ong	**2.5.3** **GEN-Panel** Cable Sim → OFF → LONG CABLE
OUTPut:**DIG**ital:**REF**erence:**FEED**	**AINP**ut **AINR**eclock **AOUT**put **RGEN**erator	**2.5.3** **GEN-Panel** Ref Out → AUDIO IN → AUD IN RCLK → AUDIO OUT → REF GEN
OUTPut:**DIG**ital:**SYNC**:**FEED**	**AIP**ut **GCL**ock **RINP**ut **SPLL**	**2.5.3** **GEN-Panel** Sync Out → AUDIO IN → GEN CLK → REF IN → SYNC PLL
OUTPut:**DIG**ital:**SYNC**:**TYPE**	**WCL**ock **BCL**ock	**2.5.3** **GEN-Panel** Type → WORD CLK → BIPHASE CLK
OUTPut:**DIG**ital:**UNB**alanced:**FEED**	**AOUT**put **AINP**ut	**2.5.3** **GEN-Panel** Unbal Out → AUDIO OUT → AUDIO IN

Command	Parameter	Section
OUTPut:**IMP**edance	**R10** **R200** **R150** (Query response = R200) **R600**	**2.5.2** **GEN-Panel** Impedance → 10 Ω → 200 Ω → 150 Ω → 600 Ω
OUTPut:**SAMP**le:**FREQ**uency	**<nu>** , Opt. UPL-B2 (Digital Audio I/O): 27 kHz to 55 kHz (UPL-B2) Opt. UPL-B29 (Dig. Audio 96 kHz): 40 kHz to 106	**2.5.3** **GEN-Panel**
OUTPut:**SAMP**le[:FREQuency]:**MODE**	**F32** **F44** **F48** **F88** **F96** **EXT**ern **SYNC**hron **VAL**ue	**2.5.3** **GEN-Panel** Sample Freq → 32 kHz → 44.1 kHz → 48 kHz → 88.2 kHz → 96 kHz → EXTERN → SYNCHRON → VALUE:
OUTPut:**SEL**ect	**OFF** **CH1** **CH2** **CH2I**s1	**2.5.2** **2.5.3** **GEN-Panel** Channel(s) → OFF → 1 → 2 → 2 ≡ 1
OUTPut:**SIGN**al:**BAL**anced:**LEV**el	**<nu>**	**2.5.3** **GEN-Panel** Bal Vpp
OUTPut:**SIGN**al:**LEV**el	**<nu>**	**2.5.3** **GEN-Panel** Unbal Vpp
OUTPut:**TYPE**	**BAL**anced **UNB**alanced	**2.5.2** **GEN-Panel** Output → BAL → UNBAL
OUTPut:**VAL**idity	**CH1A**nd2 **NONE**	**2.5.3.2** **GEN-Panel** Validity
OUTPut**2**:**IMP**edance	**R10** **R200** **R600**	**2.5.5** **GEN-Panel** AUX GEN: Impedance → 10 Ω → 200 Ω → 600 Ω

Command	Parameter	Section
OUTPut2:**SEL**ect	**OFF** **CH1** **CH2** **CH2**ls1	**2.5.5** **GEN-Panel** AUX GEN: Channel(s) → OFF → 1 → 2 → 2 ≡ 1
OUTPut**2**:**TYPE**	**UNB**alanced **BAL**anced	**2.5.5** **GEN-Panel** AUX GEN: Output → UNBAL → BAL
SENSe:**DIG**ital:**FEED**	**ADAT**a **JPH**ase **CINP**ut	**2.6.3.1** **ANLR-Panel** Meas Mode → AUDIO DATA → JITTER/PHAS → COMMON/INP
SENSe:**DIG**ital:**SYNC**:**REF**erence	**GCL**ock **PLLV**ari **PLL32** **PLL44** **PLL48** **PLL88** **PLL96**	**2.6.3** **ANLR-Panel** Related to → GEN CLK → VARI (PLL) → 32.0 (PLL) → 44.1 (PLL) → 48.0 (PLL) → 88,2 (PLL) → 96.0 (PLL)
SENSe:**DIG**ital:**SYNC**:**SOUR**ce	**AINP**ut **RINP**ut	**2.6.3** **ANLR-Panel** Sync To → AUDIO IN → REF IN
SENSe:**EQU**alize:**CONT**rol:**POIN**ts**?**	**<n>** 0 to 1024 Query only	**2.9.1.3** **No manual operation**
SENSe:**EQU**alize:**CONT**rol[:DATA]	**<n>**{,**<n>**}	**2.9.1.3** **No manual operation**
SENSe:**FREQ**uency:**FACT**or	<nu> MLT 1 to 20 for RMS-Sel. measurement	**2.6.5.3** **2.6.5.23** **ANLR-Panel** FREQ MODE → Factor
SENSe:**SWE**ep:**SYNC**	**NORM**al **BLOC**k	**2.6.5.1** **2.6.5.23** **ANLR-Panel** Sweep Mode → NORMAL → BLOCK

Command	Parameter	Section
SENSe:**UFIL**ter[1 **to 9**]:**ORD**er	**N4** **N8**	**2.7.2** **FILTER-Panel** Order → 4 → 8
SENSe:**VOLT**age:**EQU**alize:**POIN**ts**?**	**<n>** 0 to 1024 Query only	**2.9.1.3** **No manual operation**
SENSe:**VOLT**age:**EQU**alize[:DATA]	**<n>**{,**<n>**}	**2.9.1.3** **No manual operation**
SENSe**:VOLT**age**:EQU**alize[:STATe]	**ON** **OFF**	**2.6.5.7** **2.6.5.12** **ANL Panel** Equalizer → ON → OFF
SENSe:**VOLT**age:**INTV**:**MODE**	**OFF** **FOR**ever **SMO**oth **EDGE**	**2.6.5.24** **ANLR-Panel** Max Hold → OFF → FOREVER → SLOW DECAY → FAST DECAY
SENSe**:VOLT**age**:INTV:MODE**	**OFF** **ON**	**2.6.5.25** **ANLR-Panel** Max Hold → OFF → ON
SENSe[]:**POW**er:**REF**erence:**RES**istance	**<nu>** 1 mΩ to 100 kΩ	**2.4** (RREF) **2.6.2** **ANLR-Panel** Ref Imped
SENSe[]:**VOLT**age:**RANG**e[1|**2**]:**AUTO**	**ON** **OFF**	**2.6.22.6.2** **ANLR-Panel** Range → AUTO
SENSe[]:**VOLT**age:**RANG**e[1|**2**]:**LOW**er	**<nu>** Value range see 2.6.2 Configuration of the Analog Analyzers	**2.6.2** **ANLR-Panel** Range → LOWER
SENSe[]:**VOLT**age:**RANG**e[1|**2**][:UPPer]	**<nu>** Value range see 2.6.2 Configuration of the Analog Analyzers	**2.6.2** **ANLR-Panel** Range → FIX
SENSe[1]:**BWID**th[:RESolution] gleichbedeutend mit **SENS**e[1]:**BAND**width[:RESolution]	**<nu>** Value range determined by instrument or function	**2.6.5.3** **ANLR-Panel** Bandwidth

Command	Parameter	Section
SENSe[1]:**BWID**th[:RESolution]:**MODE** gleichbedeutend mit **SENS**e[1]:**BANDwid**th[:RESolution]:**MODE**	**PPCT1** **PPCT3** **PTOC**t **POCT1**2 **PFIX** **PFAS**t **SPCT1** **SPCT3** **STOC**t **SOCT1**2 **SFIX** **SFAS**t	**2.6.5.3** **ANLR-Panel** Bandwidth → BP 1% → BP 3 % → BP 1/3 OCT → BP 1/12 OCT → BP FIX: → BP FAST → BS 1% → BS 3 % → BS 1/3 OCT → BS 1/12 OCT → BS FIX: → BS FAST
SENSe[1]:**CHAN**nel:**DEL**ay	**<nu>** -10 to 10 s	**2.6.5.12** **2.6.5.22** **ANLR-Panel** Chan Delay
SENSe[1]:**DATA**1|**2?**	<n> Query only	**3.15.8** **Result display**
SENSe[1]:**FILT**er**<i>**	**<i>***) = 1 to 3 **OFF** Query returns name of the filter switched on: **UFIL1** : **UFIL9** **AWE** **CMES** **CCIT** **CCIR** **CCIU** **DEMP5015** **DEMP50** **DEMP75** **DEMP17** **WRUM** **URUM** **DCN** **CARM** **IECT** **JITT**	**2.7.1** **FILTER-Panel** Filter
SENSe[1]:**FILT**er<i>:**AWE**ighting[:STATe]	<i> 1 to 3 **ON**|**OFF**	**2.7.1** **FILTER-Panel** Filter → A Weighting
SENSe[1]:**FILT**er<i>:**CARM**[:STATe]	<i> 1 to 3 **ON**|**OFF**	**2.7.1** **FILTER-Panel** Filter → CCIR ARM
SENSe[1]:**FILT**er<i>:**CCIR**[:STATe]	<i> 1 to 3 **ON**|**OFF**	**2.7.1** **FILTER-Panel** Filter → CCIR wtd

Command	Parameter	Section
SENSe[1]:**FILT**er<i>:**CCIT**t[:STATe]	<i> 1 to 3 **ON**|**OFF**	**2.7.1** **FILTER-Panel** Filter → CCITT
SENSe[1]:**FILT**er<i>:**CCIU**nweight[:STATe]	<i> 1 to 3 **ON**|**OFF**	**2.7.1** **FILTER-Panel** Filter → CCIR unwtd
SENSe[1]:**FILT**er<i>:**CMES**sage[:STATe]	<i> 1 to 3 **ON**|**OFF**	**2.7.1** **FILTER-Panel** Filter → C MESSAGE
SENSe[1]:**FILT**er<i>:**DCN**oise[:STATe]	<i> 1 to 3 **ON**|**OFF**	**2.7.1** **FILTER-Panel** Filter → DC NOISE HP
SENSe[1]:**FILT**er<i>:**DEMP**hasis**17**[:STATe]	<i> 1 to 3 **ON**|**OFF**	**2.7.1** **FILTER-Panel** Filter → DEEMPH J.17
SENSe[1]:**FILT**er<i>:**DEMP**hasis**50**[:STATe]	<i> 1 to 3 **ON**|**OFF**	**2.7.1** **FILTER-Panel** Filter → DEEMPH 50
SENSe[1]:**FILT**er<i>:**DEMP**hasis**5015**[:STATe]	<i> 1 to 3 **ON**|**OFF**	**2.7.1** **FILTER-Panel** Filter → DEEM 50/15
SENSe[1]:**FILT**er<i>:**DEMP**hasis**75**[:STATe]	<i> 1 to 3 **ON**|**OFF**	**2.7.1** **FILTER-Panel** Filter → DEEMPH 75
SENSe[1]:**FILT**er<i>:**IECTuner**[:STATe]	<i> 1 to 3 **ON**|**OFF**	**2.7.1** **FILTER-Panel** Filter → IEC Tuner
SENSe[1]:**FILT**er**<i>**:**JITT**er[:STATe]	<i> 1 to 4 **ON** | **OFF**	**2.7.1** **FILTER-Panel** Filter → JITTER wtd
SENSe[1]:**FILT**er<i>:**UFIL**ter**1...**:**UFIL**ter**9**[:STATe]	<i> 1 to 3 **ON**|**OFF**	**2.7.1** **FILTER-Panel** Filter
SENSe[1]:**FILT**er<i>:**URUM**ble[:STATe]	<i> 1 to 3 **ON**|**OFF**	**2.7.1** **FILTER-Panel** Filter → RUMBLE unw

Command	Parameter	Section
SENSe[1]:**FILT**er<i>:**WRUM**ble[:STATe]	<i> 1 to 3 **ON**|**OFF**	**2.7.1** **FILTER-Panel** Filter → RUMBLE wtd
SENSe[1]:**FILT**er**2**:.....		**2.7.1** **ANLR panel** Filter
SENSe[1]:**FREQ**uency:**APER**ture:**MODE**	**FAST** **PREC**ision	**2.6.5.19** **ANLR-Panel** Meas Time → FAST → PRECISION
SENSe[1]:**FREQ**uency:**LIMi**t:**LOW**er	**<nu>** Value range determined by instrument or function	**2.6.5.7** **2.6.5.23** **2.6.5.24** **2.6.5.25** **ANLR-Panel** → Frq Lim Low
SENSe[1]:**FREQ**uency:**LIMi**t:**UPP**er	**<nu>** Value range determined by instrument or function	**2.6.5.7** **2.6.5.23** **2.6.5.24** **2.6.5.25** **ANLR-Panel** → Frq Lim Upp
SENSe[1]:**FREQ**uency:**MODE**	**FIX**ed|**CW** **SWE**ep **LIST** **MULT**isine **GENT**rack **CH1** **CH2**	**2.6.5.3** **ANLR panel** SWEEP CTRL → OFF → AUTO SWEEP MANU SWEEP → AUTO LIST MANU LIST →GEN MLTSINE FREQ MODE → GEN TRACK → FREQ CH1 → FREQ CH2
SENSe[1]:**FREQ**uency:**STAR**t **SENS**e[1]:**FREQ**uency:**STOP**	<nu> Value range determined by instrument or function	**2.6.5.3** **ANLR-Panel** SWEEP CTRL → Start | Stop
SENSe[1]:**FREQ**uency[:FIXed|CW]	<nu> Value range determined by instrument or function	**2.6.5.3** **ANLR-Panel** FREQ MODE → FIX

Command	Parameter	Section
SENSe[1]:**FUNC**tion	**'OFF'** **'RMS'** **'RMSS**electiv' **'PEAK'** **'QREak'** **'DC'** **'THD'** **'THDN**sndr' **'MDIS**t' **'DFD'** **'WAF'** **'POL**arity' **'FFT'** **'FILT**ersimulation' **'WAV**eform' **'COH**erence' **'RUBB**uzz' **'PROT**ocol' **'THIR**doct'	**2.6.5** **ANLR-Panel** FUNCTION → OFF → RMS & S/N → RMS SELECT → PEAK & S/N → QPK & S/N → DC → THD → THD+N/SINAD → MOD DIST → DFD → WOW & FL → POLARITY → FFT → FILTER SIM. → WAVEFORM → COHERENCE → RUB & BUZZ → PROTOCOL → THIRD OCT
SENSe[1]:**FUNC**tion	**'OFF'** **'RMS'** **'RMSS**electiv' **'PEAK'** **'QPE**ak' **'DC'** **'THD'** **'THDN**sndr' **'MDIS**t' **'DFD'** **'WAF'** **'POL**arity' **'FFT'** **'FILT**ersim' **'WAV**eform' **'PROT**ocol" **TWEL**vthoct'	**2.6.5** **ANLR-Panel** FUNCTION → OFF → RMS & S/N → RMS SELECT → PEAK & S/N → QPK & S/N → DC → THD → THD+N/SINAD → MOD DIST → DFD → WOW & FL → POLARITY → FFT → FILTER SIM. → WAVEFORM → PROTOCOL → 12th OCTAVE
SENSe[1]:**FUNC**tion:**DCS**uppression	**ON** **OFF**	**2.6.5.1** ANLR-Panel DC Suppres → ON → OFF
SENSe[1]:**FUNC**tion:**DIST**ortion	**<n>**	**2.6.5.6** **ANLR-Panel** → di2468
SENSe[1]:**FUNC**tion:**DMOD**e	**FAST** **PREC**ision	**2.6.5.6** **2.6.5.7** **2.6.5.8** **2.6.5.9** **ANLR-Panel** Dyn Mode → FAST → PRECISION

Command	Parameter	Section
SENSe[1]**:FUNC**tion**:MCO**unt	**T30** **T32**	**2.6.5.24** **ANLR-Panel** Line Count → 30 → 32
SENSe[1]:**FUNC**tion:**MMOD**e	**PPE**ak **NPE**ak **PTOP**eak **PABS**olut **SEL**ectdi **LSEL**ectdi **DALL** **LDAL**l **DODD** **LDOD**d **DEV**en **LDEV**en **THDN** **LTHD**n **SNDR**atio **NOIS**e **LNO**ise **D2_268** alias **D2** **D3_268** alias **D3** **D2_118** **D3_118** **STAN**dard **COMP**ressed **USAM**pl **O33**	**2.6.5.4** **ANLR-Panel** Meas Mode → PK + → PK - → PK to PK → PK abs **2.6.5.6** → SELECT di → LEV SEL di → All di → LEV All di → All odd di → LEV odd di → All even di → LEV even di **2.6.5.7** → THD+N → LEVEL THD+N → SINAD → NOISE → LEVEL NOISE **2.6.5.9** → d2 (IEC268) → d3 (IEC268) → d2 (IEC118) → d3 (IEC118) **2.6.5.14** → STANDARD → COMPRESSED → UNDERSAMP
SENSe[1]:**FUNC**tion:**SETT**ling:...		**2.3.4.2** **ANLR-Panel** Fnct Settl
SENSe[1]:**FUNC**tion:**SETTl**ing:**COUNt**	**<n>** EXP | FLAT: 2 to 6 AVER: 2 to 100	**2.6.5.1** **ANLR-Panel** Samples
SENSe[1]:**FUNC**tion:**SETTl**ing:**MODE**	**OFF** **EXP**onential **FLAT** **AVER**age	**2.6.5.1** **ANLR-Panel** Settling → OFF → EXPonential → FLAT → AVERage
SENSe[1]:**FUNC**tion:**SETTl**ing:**RES**olution	**<nu>** Value range and units are determined by instrument or function see 2.6.5.1	**2.6.5.1** **ANLR-Panel** Resolution

Command	Parameter	Section
SENSe[1]:**FUNC**tion:**SETTl**ing:**TOL**erance	**<n>** 0.001 to 10 %	**2.6.5.1** **ANLR-Panel** Tolerance
SENSe[1]:**FUNC**tion:**SETTl**ing:**TOUT**	**<nu>** 0.001 to 10 s	**2.6.5.1** **ANLR-Panel** Timeout
SENSe[1]:**FUNC**tion:**SNS**equence	**ON** **OFF**	**2.6.5.1** **ANLR-Panel** S/N Sequ → ON → OFF
SENSe[1]:**FUNC**tion:**STAN**dard	**NAB** **JIS** **DIN**iec **SI05** **SI10**	**2.6.5.10** **ANLR-Panel** Rule → NAB → JIS → DIN/IEC → 2 Sigma 5 s → 2 Sigma 10s
SENSe[1]:**FUNC**tion:**WEIG**hting	**ON** **OFF**	**2.6.5.10** **ANLR-Panel** Weighting → ON → OFF
SENSe[1]:**LIST**:**FREQ**uency	**<n>**{,**<n>**}	**2.9.1.3** **No manual control**
SENSe[1]:**LIST**:**FREQ**uency:**POIN**ts**?**	**<n>** 0 to 1023 Query only	**2.9.1.3** **No manual control**
SENSe[1]:**LIST**:**MODE**	**AUTO** **MAN**ual	**2.6.5.3** **ANLR-Panel** SWEEP CTRL → AUTO LIST → MANU LIST
SENSe[1]:**NOTC**h:**FREQ**uency:**FIX**ed	**<nu>** Value range determined by instrument or function	**2.6.5.1** **ANLR-Panel** Notch Freq → VALUE:
SENSe[1]:**NOTC**h:**FREQ**uency:**MODE**	**FIX**ed **GENT**rack	**2.6.5.1** **ANLR-Panel** Notch Freq → VALUE: → GEN TRACK
SENSe[1]:**NOTC**h[:STATe]	**DB0** **DB12** **DB30** **OFF**	**2.6.5.1** **ANLR-Panel** Anlg. Notch → 0 dB → 12 dB → 30 dB → OFF

Command	Parameter	Section
SENSe[1]:**O33?**	Query only Query response: ASCII string.	**No manual control**
SENSe[1]:**SMO**othing:**APER**ture	**N1** **N2** **N4** **N8** **N16** **N32**	**2.6.5.14** **ANLR-Panel** Interpol → 1 → 2 → 4 → 8 → 16 → 32
SENSe[1]:**SWE**ep:**MODE**	**AUTO** **MAN**ual	**2.6.5.3** **ANLR-Panel** SWEEP CTRL → AUTO SWEEP → MANU SWEEP
SENSe[1]:**SWE**ep:**POIN**ts	**<n>** 2 to 1024	**2.6.5.3** **ANLR-Panel** Points
SENSe[1]:**SWE**ep:**SPAC**ing	**LIN**ear **LOG**arithmic	**2.6.5.3** **ANLR-Panel** Spacing → LIN → LOG
SENSe[1]:**SWE**ep:**STEP**	**<nu>** | **<n>**	**2.6.5.3** **ANLR-Panel** Steps
SENSe[1]:**THDN**:**REJ**ection	**NARR**ow **WIDE**	**2.6.5.7** **ANLR-Panel** Rejection → NARROW → WIDE
SENSe[1]:**TRIGger**:**SETTL**ing:**.....**		**2.3.4.2** **ANLR-Panel** Fnct Settl
SENSe[1]:**TRIG**ger:**SETTL**ing:**COUN**t	**<n>** EXP | FLAT: 2 to 6 AVER: 2 to 100	**2.6.5.1** **ANLR-Panel** Samples
SENSe[1]:**TRIG**ger:**SETTL**ing:**MODE**	**OFF** **EXP**onential **FLAT** **AVER**age	**2.6.5.1** **ANLR-Panel** Settling → OFF → EXPONENTIAL → FLAT → AVERAGE
SENSe[1]:**TRIG**ger:**SETTL**ing:**RES**olution	**<nu>** Value range and units determined by instrument or function see 2.6.5.1	**2.6.5.1** **ANLR-Panel** Resolution

Command	Parameter	Section
SENSe[1]:**TRIG**ger:**SETTl**ing:**TOL**erance	**<n>** 0.001 to 10 %	**2.6.5.1** **ANLR-Panel** Tolerance
SENSe[1]:**UFIL**ter<i>:**ATT**enuation	<i> 1 to 9 **<nu>** 3 to 120 dB	**2.7.2.1** **FILTER-Panel** FILTER 1 to 9 → Atten
SENSe[1]:**UFIL**ter<i>:**BPAS**s[:STATe]	<i> 1 to 9 **ON**	**2.7.2** **FILTER-Panel** FILTER 01 to 09
SENSe[1]:**UFIL**ter<i>:**BSTO**p[:STATe]	<i> 1 to 9 **ON**	**2.7.2** **FILTER-Panel** FILTER 01 to 09
SENSe[1]:**UFIL**ter<i>:**CENT**er	<i> 1 to 9 **<nu>** Value range see 2.6.1 Selecting the Analyzer	**2.7.2.4** **2.7.2.5** **FILTER-Panel** FILTER 1 to 9 → Center Frq
SENSe[1]:**UFIL**ter<i>:**DEL**ay	<i> 1 to 9 **<nu>** 0 to 1 s Query only for all filters except for file-defined filters	**2.7.2.7** **2.7.2.1** Query only **FILTER-Panel** FILTER 1 to 9 → Delay
SENSe[1]:**UFIL**ter<i>:**FILE**	**'filename'** <i> 1 to 9	**2.7.2.7** **FILTER-Panel** FILTER 1 to 9 → Filename
SENSe[1]:**UFIL**ter<i>:**FILE**[:STATe]	<i> 1 to 9 **ON**	**2.7.2** **FILTER-Panel** FILTER 01 to 09
SENSe[1]:**UFIL**ter<i>:**HPAS**s[:STATe]	<i> 1 to 9 **ON**	**2.7.2** **FILTER-Panel** FILTER 01 to 09
SENSe[1]:**UFIL**ter<i>:**LPAS**s[:STATe]	<i> 1 to 9 **ON**	**2.7.2** **FILTER-Panel** FILTER 01 to 09
SENSe[1]:**UFIL**ter**<i>**:**NOTC**h[:STATe]	<i> 1 to 9 **ON**	**2.7.2** **FILTER-Panel** FILTER 01 to 09
SENSe[1]:**UFIL**ter<i>:**OCT**av[:STATe]	<i> 1 to 9 **ON**	**2.7.2** **FILTER-Panel** FILTER 01 to 09
SENSe[1]:**UFIL**ter<i>:**PASS**b	<i> 1 to 9 **<nu>** Value range see **2.6.1**	**2.7.2.2** **FILTER-Panel** FILTER 1 to 9 → Passband

Command	Parameter	Section
SENSe[1]:**UFIL**ter<i>:**PASS**b:**LOW**er	<i> 1 to 9 **<nu>** Value range see **2.6.1**	**2.7.2.3** **FILTER-Panel** FILTER 1 to 9 → Passb low
SENSe[1]:**UFIL**ter<i>:**PASS**b:**UPP**er	<i> 1 to 9 **<nu>** Value range see **2.6.1**	**2.7.2.3** **FILTER-Panel** FILTER 1 to 9 → Passb upp
SENSe[1]:**UFIL**ter<i>:**STOP**b:**LOW**er**?**	<i> 1 to 9 **<nu>** Query only	**2.7.2.3** **FILTER-Panel** FILTER 1 to 9 → Stopb low
SENSe[1]:**UFIL**ter<i>:**STOP**b:**UPP**er**?**	<i> 1 to 9 **<nu>** Query only	**2.7.2.3** **FILTER-Panel** FILTER 1 to 9 → Stopb upp
SENSe[1]:**UFIL**ter<i>:**STOP**b**?**	<i> 1 to 9 **<nu>** Query only	**2.7.2.2** **FILTER-Panel** FILTER 1 to 9 → Stopband
SENSe[1]:**UFIL**ter<i>:**TOCT**ave[:STATe]	<i> 1 to 9 **ON**	**2.7.2** **FILTER-Panel** FILTER 01 to 09
SENSe[1]:**UFIL**ter<i>:**WIDT**h	<i> 1 to 9 **<nu>** Value range see **2.6.1**	**2.7.2.4** **2.7.2.5** **FILTER-Panel** FILTER 1 to 9 → Width
SENSe[1]:**UNIT**	**PCT**|**DB**	**2.4** **ANLR-Panel** Unit
SENSe[1]:**VOLT**age:**APER**ture	**<nu>** = >1 ms ... Value range see **2.6.5.2**	**2.6.5.2** **2.6.5.3** **2.6.5.24** **2.6.5.25** **ANLR-Panel** Meas Time
SENSe[1]:**VOLT**age:**APER**ture:**MODE**	**AFAS**t **AUTO** **TRIG**gered **GENT**rack **VAL**ue **FAST**	**2.6.5.2**, **2.6.5.3** **2.6.5.5** **2.6.5.24** **ANLR-Panel** Meas Time → AUTO FAST → AUTO → TRIGGERED → GEN TRACK → VALUE → FIX 200ms

Command	Parameter	Section
SENSe[1]**:VOLT**age**:APER**ture**:MODE**	**VAL**ue	**2.6.5.25** **ANLR-Panel** Meas Time → VALUE:
SENSe[1]:**VOLT**age:**FUND**amental	**<nu>** Value range and units determined by instrument or function	**2.6.5.6** **ANLR-Panel** Fundamentl
SENSe[1]:**VOLT**age:**FUND**amental:**MODE**	**AUTO** **VAL**ue	**2.6.5.6** **2.6.5.7** **ANLR-Panel** Fundamentl → AUTO → VALUE:
SENSe[1]:**VOLT**age:**INTV**time	**<nu>** 20 ms to 10 s 20 ms to 100 s	**2.6.5.4** **2.6.5.24** **ANLR-Panel** Intv Time
SENSe[1]:**VOLT**age:**INTV**time:**MODE**	**SFAS**t **FAST** **SLOW** **FIX**ed **VAL**ue	**2.6.5.4** **ANLR-Panel** Intv Time → FIX 50ms → FIX 200ms → FIX 1000ms → FIX 3 SEC → VALUE:
SENSe[1]:**WAV**eform:**COMP**ression	**<n>** 2 to 1024	**2.6.5.14** **ANLR-Panel** Comp Fact
SENSe[1]:**WAV**eform:**DUR**ation	**<nu>** see 2.6.5.14 WAVEFORM	**2.6.5.14** **ANLR-Panel** Trace Len
SENSe[1][:VOLTage|POWer]:**REF**erence	**<nu>** Analog instruments 100 pV to 1000 V Digital instrument 0.0 to 1.0 FS | 100 pFS to 100 FS	**2.6.5.1** **ANLR-Panel** Reference | Ref Volt
SENSe[1][:VOLTage|POWer]:**REF**erence:**MODE**	**CH1S**tore **CH2S**tore **CH1M**eas **CH2M**eas **STOR**e **GENT**rack **VAL**ue	**2.6.5.1** **ANLR-Panel** Reference → STORE CH1 → STORE CH2 → MEAS CH1 → MEAS CH2 → STORE → GEN TRACK → VALUE:
SENSe[1][:VOLTage|POWer]:**UNIT**[1|**2**]	**PCT** **DB** see 3.10.4 Units for IEC Measurement Results	**2.4** **ANLR-Panel** Unit Ch1/Ch2

Command	Parameter	Section
SENSe[1][:VOLTage|POWer]**:UNIT**[1|**2**]	Analog units: **V** | **DBV** | **DBU** | **DBM** | **W** | **DPCTV** | **DV** | **VVR** | **PCTVVR** | **DPCTW** |**DW** | **PPR** | **PCTPPR** |**DBR** Digital units: **FS** | **PCTFS** | **DBFS** | **DPCT** | **DBR** | **LSBS** | **BITS**	**2.4** **ANLR-Panel** Unit Ch1/Ch2
SENSe**2**:**DATA**1|**2?**	**<nu>** Query only	**3.15.8** **Result display**
SENSe**2**:**FUNC**tion	**'OFF'** **'PEAK**voltage**'** **'RMS'** **'PHAS**etoref**'** **'DIGI**npampl**'**	**2.6.5.18** **ANLR-Panel** INPUT DISP → OFF → PEAK → RMS → PHAS to REF → DIG INP AMP
SENSe**2**:**VOLT**age:**REF**erence	**<nu>** Analog instruments 1 mV to 1000 V Digital instrument 0.0 to 1.0 FS	**2.6.5.1** **ANLR-Panel** Reference
SENSe**2**:**VOLT**age:**REF**erence:**MODE**	**CH1**Store **CH2**Store **STOR**e **CH1M**eas **CH2M**eas **GENT**rack **DIG**outampl **VAL**ue	**2.6.5.1** **ANLR-Panel** Reference → STORE CH1 → STORE CH2 → MEAS CH1 → MEAS CH2 → STORE → GEN TRACK → DIG OUT AMP → VALUE:
SENSe**3**:**DATA**1|**2?**	Query only	**3.15.8** **Result display**
SENSe**3**:**FREQ**uency:**REF**erence	**<nu>** -1 MHz to 1 MHz	**2.6.5.1** **ANLR-Panel** Ref Freq
SENSe**3**:**FREQ**uency:**REF**erence:**MODE**	**CH1S**tore **CH2S**tore **CH1M**eas **CH2M**eas **STOR**e **GENT**rack **VAL**ue	**2.6.5.1** **ANLR-Panel** Reference → STORE CH1 → STORE CH2 → MEAS CH1 → MEAS CH2 → STORE → GEN TRACK → VALUE:

Command	Parameter	Section
SENSe**3**:**FREQ**uency:**REF**erence:**MODE**	**CH1S**tore **CH2S**tore **STOR**e **CH1M**eas **CH2M**eas **GENT**rack **VAL**ue	2.6.5.19 **ANLR-Panel** Ref Freq → STORE CH1 → STORE Ch2 → STORE → MEAS CH1 → MEAS CH2 → GEN TRACK → VALUE:
SENSe**3**:**FREQ**uency:**SETT**ling:.....		**2.3.4.2** **ANLR-Panel** Freq Settl
SENSe**3**:**FREQ**uency:**SETTl**ing:**COUN**t	**<n>** EXP | FLAT: 2 to 6 AVER: 2 to 100	**2.6.5.1** **ANLR-Panel** Samples
SENSe**3**:**FREQ**uency:**SETT**ling:**MODE**	**OFF** **EXP**onential **FLAT** **AVER**age	**2.6.5.1** **ANLR-Panel** Settling → OFF → EXPonential → FLAT → AVERage
SENSe**3**:**FREQ**uency:**SETTl**ing:**RES**olution	**<nu>** Value range and units determined by instrument or function see **2.6.5.1**	**2.6.5.1** **ANLR-Panel** Resolution
SENSe**3**:**FREQ**uency:**SETTl**ing:**TOL**erance	**<nu>** 0.001 to 10 %	**2.6.5.1** **ANLR-Panel** Tolerance
SENSe**3**:**FREQ**uency:**SETTl**ing:**TOUT**	**<nu>** 0.001 to 10 s	**2.6.5.1** **ANLR-Panel** Timeout
SENSe**3**:**FREQ**uency:**UNIT**[1|**2**]	**HZ** **DHZ** **DPCTHZ** **TERZ** **OCT** **DEC** **FFR**	**2.4** **ANLR-Panel** Unit Ch1/Ch2
SENSe**3**:**FUNC**tion	**'OFF'** **'FREQ**uency' **'SFRE**quency' **'FQPH**ase' **'FQGR**oupdelay'	**2.6.5.19** 2.6.5.20 2.6.5.21 **ANLR-Panel** FREQ/PHAS → OFF → FREQ → SAMPLE FREQ → FREQ&PHASE → FREQ&GRPDEL

Command	Parameter	Section
SENSe**3**:**PHAS**e:**FORM**at	**POS**itive **POSN**egative **NEG**ative **RAD** **RADB**ipolar **RADN**egative	**2.6.5.19** **ANLR-Panel** Format Pha → 0 to 360° → -180° to 180° → - 360° to 0° → 0 to 2 π → - π to + π → -2 π το 0
SENSe**3**:**PHAS**e:**REF**erence	**<nu>** -360° to +360°	**2.6.5.1** **ANLR-Panel** Reference
SENSe**3**:**PHAS**e:**REF**erence:**MODE**	**STOR**e **VAL**ue	**2.6.5.1** **ANLR-Panel** Reference → STORE → VALUE:
SENSe**3**:**PHAS**e:**SETTL**ing:.....		**2.3.4.2** **ANLR-Panel** Phas Settl
SENSe**3**:**PHAS**e:**SETTL**ing:**COUN**t	**<n>** EXP | FLAT: 2 to 6 AVER: 2 to 100	**2.6.5.1** **ANLR-Panel** Samples
SENSe**3**:**PHAS**e:**SETTL**ing:**MODE**	**OFF** **EXP**onential **FLAT** **AVER**age	**2.6.5.1** **ANLR-Panel** Settling → OFF → EXPonential → FLAT → AVERage
SENSe**3**:**PHAS**e:**SETTL**ing:**RES**olution	**<nu>** Value range and units determined by instrument or function see **2.6.5.1**	**2.6.5.1** **ANLR-Panel** Resolution
SENSe**3**:**PHAS**e:**SETTL**ing:**TOUT**	**<nu>** 0.001 to 10 s	**2.6.5.1** **ANLR-Panel** Timeout
SENSe**3**:**PHAS**e:**UNIT2**	**DEG** **RAD** **DDEG** **DRAD** **S** **DS**	**2.4** **ANLR-Panel** Unit Ch2
SENSe**4**:**DATA?**	**<nu>** Query only	**3.15.8** **Result display**

Command	Parameter	Section
SOURce**:AM:MODE**	**OFF** **SIN**usoid **BURS**t	**2.5.4.4** **GEN-Panel** Ampl Var → OFF → SINE → BURST
SOURce**:AM:MODE**	**OFF** **SIN**usoid **BURS**t	**2.5.4.9** **2.5.4.10** **GEN-Panel** Ampl Var → OFF → SINE → BURST
SOURce**:COD**edaudio**:CHAN**nel	**CH2** **CH6** **CHL** **CHC** **CHR** **CHLS** **CHRS** **CHLF**	**2.5.4.16** **GEN Panel** Chan Mode → 2/0 192kb/s → 5.1 448kb/s → L 448kb/s → C 448kb/s → R 448kb/s → LS 448kb/s → RS 448kb/s → LFE 448kb/s
SOURce:**COD**edaudio:**FORM**at	**AC3**	**2.5.4.16** **GEN Panel** Format → AC-3
SOURce**:COD**edaudio**:FREQ**uency	**F042** **F997** **F15K**	**2.5.4.16** **GEN Panel** Frequency → 42 Hz → 997 Hz → 15 kHz
SOURce:**DIG**ital:**FEED**	**ADAT**a **JITT**er **PHAS**e **COMM**on	**2.5.3** **GEN-Panel** Src Mode → AUDIO DATA → JITTER ONLY → PHASE → COMMON ONLY
SOURce:**DIG**ital:**REF**erence	**AZER**o **AONE**	**2.5.3** **GEN-Panel** Data → ALL ZERO → ALL ONE
SOURce:**DIG**ital:**SYNC**:**DEL**ay	**<nu>**	**2.5.3.1** **GEN-Panel** PhaseToRef

Command	Parameter	Section
SOURce:**DIG**ital:**SYNC**:**MODE**	**V50** **V60** **WCL**ock **IWCL**ock **F1024**	**2.5.3** **GEN-Panel** Sync Mode → VIDEO 50 → VIDEO 60 → WORD CLK → WRD CLK INV → 1024 kHz
SOURce:**DIG**ital:**SYNC**:**SOUR**ce	**GCL**ock **AINP**ut **RINP**ut **SINP**ut	**2.5.3** **GEN-Panel** Sync Out → GEN CLK → AUDIO IN → REF IN → SYNC IN
SOURce:**EQU**alize:**CONT**rol:**POIN**ts**?**	**<n>** 0 to 1023 Query only	**2.9.1.3** **No manual control**
SOURce:**EQU**alize:**CONT**rol[:DATA]	**<n>**{**,<n>**}	**2.9.1.3** **No manual control**
SOURce**:FREQ**uency	**<new>** 5.21 Hz to 20 kHz at a sampling rate of 48 kHz	**2.5.4.16** **GEN Panel** FREQUENCY
SOURce**:FREQ**uency**:AM**	**<nu>** 1 µHz to f_{max} f_{max} depending on generator	**2.5.4.4** **2.5.4.9** **2.5.4.10** **GEN-Panel** Mod Freq
SOURce:**FREQ**uency:**CH2S**tereo	**<nu>** Value range determined by sample frequency	**2.5.4.13** **GEN-Panel** Freq Ch2
SOURce:**FREQ**uency:**DIFF**erence	**<nu>** Value range and units determined by instrument or function	**2.5.4.8** **GEN-Panel** DIFF FREQ
SOURce:**FREQ**uency:**MEAN**	**<nu>** Value range and units determined by instrument or function	**2.5.4.8** **GEN-Panel** MEAN FREQ
SOURce:**FREQ**uency:**MODE**	**CW** | **FIX**ed **SWE**ep**1** **SWE**ep**2** **LIST1** **LIST2**	**2.5.4.2** **GEN-Panel** SWEEP CTRL X Axis Z Axis
SOURce:**FREQ**uency:**MODE**	**FIX**	**2.5.4.16** **GEN Panel** Vari Mode → FREQUENCY

Command	Parameter	Section
SOURce:**FREQ**uency:**OFFS**et:**STAT**e	**ON** **OFF**	**2.5.4.1** **GEN-Panel** Frq. Offset → + 1000 PPM → OFF
SOURce:**FREQ**uency:**REF**erence	**<nu>** 1 mHz to 1 MHz	**2.5.2** **GEN-Panel** Ref.Freq
SOURce**:FREQ**uency**:SEL**ect	**FQPH** **FQFQ**	**2.5.4.13** **GEN-Panel** Freq Mode FREQ&PHASE FREQ CH1&2
SOURce:**FREQ**uency:**STAR**t **SOUR**ce:**FREQ**uency:**STOP**	**<nu>** Value range and units determined by instrument or function	**2.5.4.2** **GEN-Panel** FREQUENCY → Start → Stop
SOURce:**FREQ**uency[:CW|FIXed]	**<nu>** Value range and units determined by instrument or function	**2.5.4.5** **GEN-Panel** FREQUENCY
SOURce:**FREQ**uency[<i>][:CW|FIXed]	**<i>** 1 to 17 **<nu>** Value range and units determined by instrument or function	**2.5.4.4** **GEN-Panel** Freq No1 to 17
SOURce:**FREQ**uency[1][:CW|FIXed]	**<nu>** Value range and units determined by instrument or function	**2.5.4.8** **GEN-Panel** UPPER FREQ
SOURce:**FREQ**uency[1][:CW|FIXed]	**<nu>** ANALOG-Gen: 240 Hz to 21,75 kHz DIGITAL-Gen: 240 Hz to f_{max} f_{max} see 2.5.1	**2.5.4.7** **GEN-Panel** UPPER FREQ
SOURce:**FREQ**uency**2**[:CW|FIXed]	**<nu>** 0 Hz to f_{max} / 8 f_{max} see 2.5.1	**2.5.4.7** **2.5.4.14** **GEN-Panel** LOWER FREQ Carr Freq

Command	Parameter	Section
SOURce:**FUNC**tion:**MODE**	**EQU**alvoltage **DEF**inedvoltage **IEC2**68 **IEC1**18 **FM** **AM**	**2.5.4.4** **GEN-Panel** Mode → EQUAL VOLT → DEFINE VOLT **2.5.4.1.1** **GEN-Panel** Mode → IEC 268 → IEC 118 **2.5.4.14** **GEN-Panel** Mode → FM → AM
SOURce:**FUNC**tion[:SHAPe]	**SIN**usoid **MULT**isine **BURS**t **S2P**ulse **MDIS**t **DFD** **RAND**om **USER** **POL**arity	**2.5.4** **GEN-Panel** FUNCTION → SINE → MULTISINE → SINE BURST → SINE² BURST → MOD DIST → DFD → RANDOM → ARBITRARY → POLARITY
SOURce:**FUNC**tion[:SHAPe]	**SIN**usoid **STER**eo **MULT**isine **BURS**t **S2P**ulse **MDIS**t **DFD** **RAND**om **USER** **POL**arity **FSK** **FM** **DC** **COD**edaud	**2.5.4** **GEN-Panel** FUNCTION → SINE → STEREO SINE → MULTISINE → SINE BURST → SINE² BURST → MOD DIST → DFD → RANDOM → ARBITRARY → POLARITY → FSK → MODULATION → DC → CODED AUDIO
SOURce**:INT**erval	**<nu>** set burst duration ... 60 s	**2.5.4.4** **2.5.4.9** **2.5.4.10** **GEN-Panel** INTERVAL
SOURce:**INT**erval:**MODE**	**CW** | **FIX**ed **SWE**ep**1** **SWE**ep**2** **LIST1** **LIST2**	**2.5.4.2** **GEN-Panel** SWEEP CTRL X Axis Z Axis

Command	Parameter	Section
SOURce:**INT**erval:**STAR**t	**<nu>** Value range and units determined by instrument or function	**2.5.4.1.3** **2.5.4.5** **2.5.4.6** **GEN-Panel** Start
SOURce:**INT**erval:**STOP**	**<nu>** Value range and units determined by instrument or function	**2.5.4.1.3** **2.5.4.5** **2.5.4.6** **GEN-Panel** Stop
SOURce:**INT**erval[:CW|FIXed]	**<nu>** Value range and units determined by instrument or function	**2.5.4.5** **2.5.4.6** **GEN-Panel** INTERVAL
SOURce:**LIST**:**DWEL**l	**<n>**{,**<n>**}	**2.9.1.3** **No manual control**
SOURce:**LIST**:**DWEL**l:**CONT**rol:**POIN**ts**?**	**<n>** 0 to 1023 Query only	**2.9.1.3** **No manual control**
SOURce:**LIST**:**DWEL**l:**CONT**rol[:DATA]	**<n>**{,**<n>**}	**2.9.1.3** **No manual control**
SOURce:**LIST**:**DWEL**l:**POIN**ts**?**	**<n>** 0 to 1023 Query only	**2.9.1.3** **No manual control**
SOURce:**LIST**:**FREQ**uency	**<n>**{,**<n>**}	**2.9.1.3** **No manual control**
SOURce:**LIST**:**FREQ**uency:**POIN**ts**?**	**<n>** 0 to 1023 Query only	**2.9.1.3** **No manual control**
SOURce:**LIST**:**INT**erval	**<n>**{,**<n>**}	**2.9.1.3** **No manual control**
SOURce:**LIST**:**INT**erval:**POIN**ts**?**	**<n>** 0 to 1023 Query only	**2.9.1.3** **No manual control**
SOURce:**LIST**:**ONT**ime	**<n>**{,**<n>**}	**2.9.1.3** **No manual control**
SOURce:**LIST**:**ONT**ime:**POIN**ts**?**	**<n>** 0 to 1023 Query only	**2.9.1.3** **No manual control**
SOURce:**LIST**:**VOLT**age	**<n>**{,**<n>**}	**2.9.1.3** **No manual control**

Command	Parameter	Section
SOURce:**LIST**:**VOLT**age:**POIN**ts**?**	**<n>** 0 to 1023 Query only	**2.9.1.3** **No manual control**
SOURce:**LOWD**istortion	**ON** **OFF**	**2.5.4.1** **2.5.4.3** **GEN-Panel** Low Dist → ON → OFF
SOURce:**MULT**isine:**COUN**t	**<n>** 1 to 17	**2.5.4.4** **GEN-Panel** No of Sine
SOURce:**MULT**isine:**MODE**	**EQU**alvoltage **DEF**inedvoltage	**2.5.4.4** **GEN-Panel** Mode → EQUAL VOLT → DEFINE VOLT
SOURce:**O33**	'O33 ID code'	**No manual control**
SOURce:**OFF**:**MODE**	**SWE**ep**2** | **LIST2**	**2.5.4.2** **GEN-Panel** Z Axis → OFF
SOURce**:ONT**ime	**<nu>** tmin to tmax Analog generator: tmin= 20.83 µs Digital generator: tmin= 1 / sample frequency tmax: 60 s – tmin	**2.5.4.4** **2.5.4.9** **2.5.4.10** **GEN-Panel** ON TIME
SOURce:**ONT**ime:**DEL**ay	**<nu>** 0 to 60 s	**2.5.4.5** **2.5.4.6** **GEN-Panel** BurstOnDel
SOURce:**ONT**ime:**MODE**	**CW** | **FIX**ed **SWE**ep**1** **SWE**ep**2** **LIST1** **LIST2**	**2.5.4.2** **GEN-Panel** SWEEP CTRL X Axis Z Axis
SOURce:**ONT**ime:**STAR**t **SOUR**ce:**ONT**ime:**STOP**	**<nu>** Value range and units determined by instrument or function	**2.5.4.2** **2.5.4.5** **2.5.4.6** **GEN-Panel** Start Stop
SOURce:**ONT**ime[:CW|FIXed]	**<nu>** Value range and units determined by instrument or function	**2.5.4.5** **GEN-Panel** ON TIME

Command	Parameter	Section
SOURce:PHASe	**<nu>** 0 ° to 360 °	**2.5.4.13** **GEN-Panel** Phas Ch2:1
SOURce:**PHAS**e[**<i>**][:ADJust]	**<i>** 1 to 17 **<nu>** 0 to 360 °	**2.5.4.4** **GEN-Panel** Phas No 1 to 17
SOURce:**PROT**ocol	**OFF** **STAT**ic **ENH**anced	**2.5.3.2** **ANA-Panel** PROTOCOL → PANEL OFF → STATIC → ENHANCED
SOURce:**PROT**ocol:**RCH**annelstatus	**ZERO** **LEQ**ual **AES3** **CRC** **RAW**	**2.5.3.2** **GEN-Panel** Ch Stat. R → ZERO → EQUAL L → FILE+AES3 → FILE+CRC → FILE
SOURce:**PROT**ocol:**UMOD**e	**ZERO** **FILE**	**2.5.3.2** **GEN-Panel** User Mode → ZERO → FILE DEF
SOURce:**RAND**om:**DOM**ain	**FREQ**uency **TIME**	**2.5.4.9** **GEN-Panel** Domain → FREQ → TIME
SOURce:**RAND**om:**FREQ**uency:**LOW**er **SOUR**ce:**RAND**om:**FREQ**uency:**UPP**er	**<nu>** Value range and units determined by instrument or function	**2.5.4.9** **GEN-Panel** Lower Freq Upper Freq
SOURce:**RAND**om:**PDF**	**GAUS**sian **TRI**angle **RECT**angle	**2.5.4.1.1** **2.5.4.9** **GEN-Panel** PDF → GAUSS → TRIANGLE → RECTANGLE
SOURce:**RANDom**:**SHAP**e	**WHIT**e **PINK** **TOCT**ave **ARB**itrary	**2.5.4.9** **GEN-Panel** Equalizatn → WHITE → PINK → THIRD OCT → FILE

Command	Parameter	Section
SOURce:**RAND**om:**SPAC**ing:**FREQ**uency	**<nu>** Lower limit value: analog = 2.93 Hz digital = sampling frequency / 16384	**2.5.4.4** **2.5.4.9** **GEN-Panel** Spacing
SOURce:**RAND**om:**SPAC**ing:**MODE**	**ATR**ack **USER**defined	**2.5.4.4** **2.5.4.9** **GEN-Panel** Spacing → ANLR TRACK → USER DEF
SOURce:**SIN**usoid:**DITH**er	**<nu>** 0 to 1 FS	**2.5.4.1.1** **GEN-Panel** Dither
SOURce:**SIN**usoid:**DITH**er:**STAT**e	**ON** **OFF**	**2.5.4.1.1** **GEN-Panel** Dither → ON → OFF
SOURce:**SWE**ep ...		3.10.1.4 GEN-Panel
SOURce:**SWE**ep:**DWEL**l	**<nu>** 10 ms to 1000 s	**2.5.4.2** **GEN-Panel** Dwell
SOURce:**SWE**ep:**FREQ**uency:**POIN**ts	**<n>** 2 to 1024	**2.5.4.2** **GEN-Panel** Points
SOURce:**SWE**ep:**FREQ**uency:**SPAC**ing	**LIN**ear **LOG**arithmic	**2.5.4.2** **GEN-Panel** Spacing
SOURce:**SWE**ep:**FREQ**uency:**STEP**	**<nu>**	**2.5.4.2** **GEN-Panel** Step
SOURce:**SWE**ep:**INT**erval:**POIN**ts	**<n>** 2 to 1024	**2.5.4.1.3** **GEN-Panel** Points
SOURce:**SWE**ep:**INT**erval:**SPAC**ing	**LIN**ear **LOG**arithmic	**2.5.4.1.3** **GEN-Panel** Spacing → LIN → LOG
SOURce:**SWE**ep:**INT**erval:**STEP**	**<nu>** Value range and units determined by instrument or function	**2.5.4.1.3** **GEN-Panel** Step
SOURce:**SWE**ep:**MODE**	**MAN**ual **AUTO**	**2.5.4.2** **GEN-Panel** Sweep Ctrl

Command	Parameter	Section
SOURce:**SWE**ep:**NEXT**step	**DWEL**l **ASYN**c **LIST**	**2.5.4.2** **GEN-Panel** Next Step → ANLR SYNC → DWELL VALUE → DWELL FILE
SOURce:**SWE**ep:**ONT**ime:**POIN**ts	**<n>** 2 to 1024	**2.5.4.2** **GEN-Panel** Points
SOURce:**SWE**ep:**ONT**ime:**SPAC**ing	**LIN**ear **LOG**arithmic	**2.5.4.2** **GEN-Panel** Spacing → LIN → LOG
SOURce:**SWE**ep:**ONT**ime:**STEP**	**<nu>** Value range and units determined by instrument or function	**2.5.4.1.3** **GEN-Panel** Step
SOURce:**SWE**ep:**VOLT**age:**POIN**ts	**<n>** 2 to 1024	**2.5.4.2** **GEN-Panel** Points
SOURce:**SWE**ep:**VOLT**age:**SPAC**ing	**LIN**ear **LOG**arithmic	**2.5.4.2** **GEN-Panel** Spacing → LIN → LOG
SOURce:**SWE**ep:**VOLT**age:**STEP**	**<nu>** Value range and units determined by instrument or function	**2.5.4.2** **GEN-Panel** Step
SOURce**:VOLT**age**:AM**	**<nu>** -100% to 0%	**2.5.4.4** **2.5.4.9** **2.5.4.10** **GEN-Panel** Variation
SOURce:**VOLT**age:**CH2S**tereo	**<nu>** 0 to 1 FS	**2.5.4.13** **GEN-Panel** VOLT Ch2
SOURce:**VOLT**age:**CRES**tfactor	**<n>** 1 to 100	**2.5.4.4** **GEN-Panel** Crest Fact
SOURce:**VOLT**age:**CRES**tfactor:**MODE**	**MIN**imized **DPH**ase **VAL**ue	**2.5.4.4** **GEN-Panel** Crest Fact → OPTIMIZED → DEFINE PHAS → VALUE
SOURce:**VOLT**age:**EQU**alize:**POIN**ts**?**	**<n>** 0 to 1023 Query only	**2.9.1.3** **No manual control**

Command	Parameter	Section
SOURce:**VOLT**age:**EQU**alize:**STAT**e	**ON** **OFF**	2.5.4.1.1 2.5.4.4 2.5.4.3 **GEN-Panel** Equalizer → ON → OFF
SOURce:**VOLT**age:**EQU**alize[:DATA]	**<n>{,<n>}**	**2.9.1.3** **No manual control**
SOURce:**VOLT**age:**LIM**it[:AMPLitude]	**<nu>** 0 to 20 V 0 to 1 FS	**2.5.2** **GEN-Panel** For Volt Range = AUTO, Max Volt is displayed, for Volt Range = FIX, a numeric value.
SOURce:**VOLT**age:**LOWL**evel	**<nu>** 0 to SOUR:VOLT	**2.5.4.5** **GEN-Panel** Low Level
SOURce:**VOLT**age:**MODE**	**CW** | **FIX**ed **SWE**ep**1** **SWE**ep**2** **LIST1** **LIST2**	**2.5.4.2** **GEN-Panel** SWEEP CTRL X Axis Z Axis
SOURce**:VOLT**age**:MODE**	**FIX**	**2.5.4.16** **GEN Panel** Vari Mode → VOLTAGE
SOURce:**VOLT**age:**RANG**e:**AUTO**	**ON** **OFF**	**2.5.2** **GEN-Panel** Volt Range → AUTO → FIX
SOURce:**VOLT**age:**RAT**io	**<n>** 1 to 10	**2.5.4.7** **2.5.4.13** **GEN-Panel** VOLT LF:UF or Volt Ch2:1
SOURce:**VOLT**age:**REF**erence	**<nu>** 1 mV to 1 MV	**2.5.2** **GEN-Panel** Ref.Volt
SOURce**:VOLT**age**:SEL**ect	**VLRT** **VLVL**	**2.5.4.13** **GEN-Panel** Volt Mode VOLT&RATIO VOLT CH1&2

Command	Parameter	Section
SOURce:**VOLT**age:**STAR**t	**<nu>** Value range determined by instrument or function	**2.5.4.2** **GEN-Panel** VOLTAGE → Start
SOURce:**VOLT**age:**STOP**	**<nu>** Value range determined by instrument or function	**2.5.4.2** **GEN-Panel** VOLTAGE → Stop
SOURce:**VOLT**age:**TOT**al:**GAIN**	**<nu>**	**2.5.4.4** **GEN-Panel** TOTAL GAIN
SOURce:**VOLT**age:**TOT**al:**RMS?**	**<nu>** 0 V to 20 V	**2.5.4.4** **2.5.4.9** **2.5.4.10** **GEN-Panel** TOTAL RMS
SOURce:**VOLT**age:**TOT**al[:LEVel|AMPLitude]	**<nu>** Value range determined by instrument or function	**2.5.4.4** Query only **2.5.4.7** **2.5.4.8** **2.5.4.9** **2.5.4.10** **2.5.4.15** **GEN-Panel** TOTAL VOLT
SOURce:**VOLT**age:**TOT**al[:LEVel|AMPLitude]	**<new>** 1 µFS to 1 FS or -120 dBFS to 0 dBFS	**2.5.4.16** **GEN Panel** TOTAL VOLT
SOURce:**VOLT**age[:LEVel|AMPLitude]	**<nu>** 0 to 12 V 0 to 24 V 0 to 1 FS	**2.5.4.3** **2.5.4.5** **2.5.4.6** **2.5.4.11** **2.5.4.12** **2.5.4.13** **GEN-Panel** VOLTAGE or VOLT Ch1
SOURce:**VOLT**age[:LEVel|AMPLitude]:**OFFS**et	**<n>** -5 V to 5 V –10 V to 10 V –1 FS to 1 FS	**2.5.4.1.1** **GEN-Panel** DC Offset
SOURce:**VOLT**age[:LEVel|AMPLitude]:**OFFS**et:**STAT**e	**OFF** **ON**	**2.5.4.1.1** **GEN-Panel** DC Offset → OFF → ON

Command	Parameter	Section
SOURce:**VOLT**age[**<i>**][:LEVel|AMPLitude]	**<i>** 1 to 17 **<nu>** Analog instr.: OUTP:TYPE UNB 0 to 10 V Analog instr.: OUTP:TYPE BAL 0 to 20 V Digital instrument: 0 to 1 FS	**2.5.4.4** **GEN-Panel** Volt No 1 to 17
SOURce:**VOLT**age**2**	**<nu>** 0 to 5 V 0 to 6V 0 to 0.5 FS	2.5.4.14 **GEN-Panel** Carr Volt
SOURce**2**:**FREQ**uency:**MODE**	**CW** | **FIX**ed **SWE**ep**1** **LIST1**	**2.5.4.2** Sweeps **GEN-Panel** AUX GEN: SWEEP CTRL X Axis
SOURce**2**:**FREQ**uency:**STAR**t	**<nu>** 10 Hz to 110 kHz	**2.5.4.2** Sweeps **GEN-Panel** AUX GEN: FREQUENCY Start
SOURce**2**:**FREQ**uency:**STOP**	**<nu>** 10 Hz to 110 kHz	**2.5.4.2** Sweeps **GEN-Panel** AUX GEN: FREQUENCY Stop
SOURce**2**:**FREQ**uency[:CW|FIXed]	**<nu>** 10 Hz to 110 kHz	**2.5.5** **GEN-Panel** Abhängig von AUX GEN: → Anlg Freq → Comm Freq → JittPkFreq
SOURce2:**FUNC**tion	**OFF** **ANLG**out **COMM**on **JITT**er	**2.5.5** **GEN-Panel** **AUX GEN**: → OFF → ANALOG OUT → COMMON MODE → JITTER
SOURce**2**:**SWE**ep ...		**2.5.4.2 Sweeps** **GEN-Panel** AUX GEN:
SOURce**2**:**SWE**ep:**DWEL**l	**<nu>** 10 ms to 1000 s	**2.5.4.2** Sweeps **GEN-Panel** AUX GEN: Dwell

Command	Parameter	Section
SOURce**2**:**SWE**ep:**FREQ**uency:**POIN**ts	**<n>** 2 to 1024	**2.5.4.2** Sweeps **GEN-Panel** AUX GEN: FREQUENCY Points
SOURce**2**:**SWE**ep:**FREQ**uency:**SPAC**ing	**LIN**ear **LOG**arithmic	**2.5.4.2** Sweeps **GEN-Panel** Spacing → LIN → LOG
SOURce**2**:**SWE**ep:**FREQ**uency:**STEP**	**<nu>** depending on STARt and STOP	**2.5.4.2** Sweeps **GEN-Panel** AUX GEN FREQUENCY Step
SOURce**2**:**SWE**ep:**MODE**	**MAN**ual **AUTO**	**2.5.4.1.3** **GEN-Panel** AUX GEN Sweep Ctrl
SOURce**2**:**SWE**ep:**NEXT**step	**DWEL** **ASYN** **LIST**	**2.5.4.2** Sweeps **GEN-Panel** AUX GEN: Next Step → ANLR SYNC → DWELL VALUE → DWELL FILE
SOURce**2**:**SWE**ep:**VOLT**age:**POIN**ts	**<n>** 2 to 1024	**2.5.4.2** Sweeps **GEN-Panel** AUX GEN VOLTAGE | AMPL Points
SOURce**2**:**SWE**ep:**VOLT**age:**SPAC**ing	**LIN**ear **LOG**arithmic	**2.5.4.2** Sweeps **GEN-Panel** VOLTAGE | AMPL AUX GEN: ANALOG Spacing → LIN → LOG
SOURce**2**:**SWE**ep:**VOLT**age:**STEP**	**<nu>** depending on STARt and STOP	**2.5.4.2** Sweeps **GEN-Panel** AUX GEN VOLTAGE | AMPL Step
SOURce**2**:**VOLT**age:**MODE**	**CW** | **FIX**ed **SWE**ep**1** **LIST1**	**2.5.4.2** Sweeps **GEN-Panel** AUX GEN SWEEP CTRL X Axis

Command	Parameter	Section
SOURce**2**:**VOLT**age:**STAR**t	**<nu>** 0 to 7.07 V 0 to 250 mUI	**2.5.4.2** Sweeps **GEN-Panel** AUX GEN VOLTAGE | AMPL Start
SOURce**2**:**VOLT**age:**STOP**	**<nu>** 0 to 7.07 V 0 to 7.07 V 0 to 250 mUI	**2.5.4.2** Sweeps **GEN-Panel** AUX GEN VOLTAGE | AMPL Stop
SOURce**2**:**VOLT**age[:LEVel|AMPLitude]	**<nu>** 0 to 7.07 V 0 to 7.07 V 0 to 250 mUI	**2.5.5** **GEN-Panel** Abhängig von AUX GEN: → Anlg Ampl → Comm Ampl → JittPkAmpl
STATus:**OPER**ation:**COND?**	**<n>** Query only	**3.7.3.4** **No manual control**
STATus:**OPER**ation:**ENAB**le	**<n>** .... 0 0 **1** 0 0 **1** 0 0 ... .d7 d6 d5 d4 d3 d2 d1 d0 Example: d2 and d5 set: **<n>** = 36 (4 + 32)	**3.7.3.4** **No manual control**
STATus:**OPER**ation:**NTR**ansition	**<n>**	**3.7.3.4** **No manual control**
STATus:**OPER**ation:**PTR**ansition	**<n>**	**3.7.3.4** **No manual control**
STATus:**OPER**ation[:EVENt]**?**	**<n>** Query only	**3.7.3.4** **No manual control**
STATus:**PRES**et		**3.7.5** **No manual control**
STATus:**QUES**tionable:**COND?**	**<n>** Query only	**3.7.3.5** **No manual control**
STATus:**QUES**tionable:**ENAB**le	**<n>**	**3.7.3.5** **No manual control**
STATus:**QUES**tionable:**NTR**ansition	**<n>**	**3.7.3.5** **No manual control**

Command	Parameter	Section
STATus:**QUES**tionable:**PTR**ansition	**<n>**	**3.7.3.5** **No manual control**
STATus:**QUES**tionable[:EVENt]**?**	**<n>** Query only	**3.7.3.5** **No manual control**
STATus:**XQUE**stionable:**COND?**	**<n>** Query only	**3.7.3.6** **No manual control**
STATus:**XQUE**stionable:**ENAB**le	**<n>**	**3.7.3.6** **No manual control**
STATus:**XQUE**Stionable:**NTR**ansition	**<n>**	**3.7.3.6** **No manual control**
STATus:**XQUE**stionable:**PTR**ansition	**<n>**	**3.7.3.6** **No manual control**
STATus:**XQUE**stionable[:EVENt]**?**	**<n>** Query only	**3.7.3.6** **No manual control**
SYSTem:**AHAR**dware:**VERS**ion**?**	**ABO**ard **ACOD**e Query only Query: SYST:AHAR:VERS? ABO SYST:AHAR:VERS? ACOD The response is either the number of a version (eg 0.01) or –NA- (Not Available) if the board is not installed.	**2.15.7** **OPTIONS-Panel** VERSIONS ------ Anlg Board code
SYSTem:**BEEP**er:**STAT**e	**ON** **OFF**	**2.15.2** **OPTIONS-Panel** Beeper → ON → OFF
SYSTem:**COMM**unicate:**GPIB**:**ADDR**ess	**<n>** 0 to 31	**2.15.1** **OPTIONS-Panel** UPL IECadr
SYSTem:**COMM**unicate:**GTL**		**LOCAL-Tastendruck**
SYSTem:**COMM**unicate:**SER**ial**2**: **CONT**rol	**RTS** **XON**	2.15.1 **OPTIONS-Panel** Handshake → RTS/CTS → XON/XOFF

Command	Parameter	Section
SYSTem:**COMM**unicate:**SER**ial**2**: **FEED**:**BITS**	**<n>** n = 7 | 8	2.15.1 **OPTIONS-Panel** Data Bits → 7 → 8
SYSTem:**COMM**unicate:**SER**ial**2**:**FEED**:**BAUD**	**<n>** n = **2400** **3600** **4800** **7200** **9600** **19200** **38400** **56000**	2.15.1 **OPTIONS-Panel** Baud Rate → 2400 Baud → 3600 Baud → 4800 Baud → 7200 Baud → 9600 Baud → 19200 Baud → 38400 Baud → 56000 Baud
SYSTem:**COMM**unicate:**SER**ial**2**:**FEED**:**PAR**ity[:TYPE]	**NONE** **EVEN** **ODD**	2.15.1 **OPTIONS-Panel** Parity → NONE → EVEN → ODD
SYSTem:**COMM**unicate:**SER**ial**2**:**FEED**:**SBIT**s	**<n>** n = 1 | 2	2.15.1 **OPTIONS-Panel** Stop Bits → 1 → 2
SYSTem:**DHAR**dware:**VERS**ion**?**	**CPU**board **DBO**ard Query only Query: SYST:DHAR:VERS? CPU SYST:DHAR:VERS? DBO The response is the number of a version (eg 0.05).	2.15.7 **OPTIONS-Panel** VERSIONS ------ CPU Board Dig. Board
SYSTem:**DISP**lay:**READ**ing:**RATE**	**MAXS**peed **FSTS**peed **MEDS**peed **SLWS**peed	2.15.5 **OPTIONS-Panel** Read Rate → MAX SPEED → 6/s → 3/s → 1/s
SYSTem:**DISP**lay:**READ**ing:**RES**olution	**<n>**	2.15.5 **OPTIONS-Panel** Read Resol
SYSTem:**DISP**lay:**TRAC**e[]:**LOAD**	**MAN**ual **DEF**ault **ACOL**or **ALIN**e	**2.15.5.4** **OPTIONS-Panel** Scan conf → MANUAL → DEFAULT → AUTO COLOR → AUTO LINE

Command	Parameter	Section
SYSTem:**DISP**lay:**TRAC**e[1|**2**]:**COL**or	**GRE**en **YELL**ow **BLUE** **CYAN** **MAG**enta **WHIT**e **BLAC**k **DGR**ay **LGR**ay	**2.15.5.4** **OPTIONS-Panel** Color (A) / (B) → GREEN → YELLOW → BLUE → CYAN → MAGENTA → WHITE → BLACK → DARK GRAY → LIGHT GRAY
SYSTem:**DISP**lay:**TRAC**e[1|**2**]:**LINE**	**SSOL**id **SD** **SP** **SPD** **DSOL**id **DD** **DP** **DPD**	**2.15.5.4** **OPTIONS-Panel** Line (A) / (B) → ——— → - - - - - - → . . . . . . → . - . - . - → ═══ → ====== → : : : : : : → : = : = : =
SYSTem:**DISP**lay:**TRAC**e[1|**2**]:**SEL**ect	**<n>** 1 to 17	**2.15.5.4** **OPTIONS-Panel** Scannr.(A) Scannr.(B)
SYSTem:**ERR**or**?**	**<n>** Query only	3.3.2 **No manual control**
SYSTem:**INFO**text:**STAT**e	**ON** **OFF**	2.9.1.1 **FILE-Panel** Info Displ → ON → OFF
SYSTem:**KEY**:**RDEL**ay	**<nu>** 0.25 to 1.0 s	**2.15.3** **OPTIONS-Panel** Rep Delay
SYSTem:**KEY**:**RRAT**e	**<nu>** 0 to 50 Hz	**2.15.3** **OPTIONS-Panel** Reptn Rate
SYSTem:**LSP**eed	**FAST** **SLOW**	**No manual control**

Command	Parameter	Section
SYSTem:**OPT**ions:**VERS**ion?	**LDG** alias **B1** **REMO**te alias **B4** **DAUD**io alias **B2|B29** **SPE**aker alias **B5** **DAPR**otocol alias **B21** **DAJ**Itter alias **B22** **SQCO**ntrol alias **B10** **B33** **B6** **B8** Query only Query-Form e. g.: SYST:OPT:VERS? LDG The response is either the number of an option (eg 0.01), INST or -NA- (Not Available) if the board or option are not installed.	**2.15.7** **2.6.6** **OPTIONS-Panel** OPTIONS ------- B1 Low Dist B4 Rem Ctrl B2 DigAudio B5 Speaker B21 DA Prot B22 DA Jitt B10 Seq Ctrl ITU-T O33 B6 Coher B8 PhoneTst
SYSTem:**PAR**ameter:**LINK**	**<n>** 0 to 2047	**2.15.7** **OPTIONS-Panel** Param. Link
SYSTem:**PHON**e	**SPKC** **PERM**	**2.6.6** **ANLR-Panel** Phone Out → SPKPhone → PERMANENT
SYSTem:**PRIN**t	**TRAC**e1 **TRAC**e2 **EQU**alize **ERR**ors **DWEL**l **LIML**ower **LIMU**pper **LIST1** **LIST2** **TR1A**nd2 **OFF**	**2.14.5** **OPTIONS-Panel** PRINT Type → TRACE A → TRACE B → EQUALIZATN → LIM REPORT → DWELL VALUE → LIM LOWER → LIM UPPER → X AXIS → Z AXIS → TRACE A+B → OFF
SYSTem:**PROG**ram:**EXEC**ute	**'filename'**	2.16 **OPTIONS-Panel** Exec Macro <filename>
SYSTem:**PROG**ram:**POIN**ts?	**<n>** 0 to 1024 Query only	**No manual control**
SYSTem:**PROG**ram[:DATA]	**<n>**{,**<n>**}	**No manual control**

Command	Parameter	Section
SYSTem:**SOFT**ware:**VERS**ion**?**	**SOFT**ware **SETU**p Query only Query: SYST:SOFT:VERS? SOFT SYST:SOFT:VERS? SETU The response is the number of a version (e.g. 3.05).	**2.15.7** **OPTIONS-Panel** VERSIONS ------ Software Setup
SYSTem:**SPE**aker:**GAIN**	**<nu>** -120 to 120 dB	**2.6.6** **ANLR-Panel** Pre Gain
SYSTem:**SPE**aker:**SOUR**ce	**OFF** **INP1** **INP2** **IN1A**nd2 **FNC1** **FNC2** **FN1A**nd2 **AES1** **AES2** **AE1A**nd2	**2.6.6** **ANLR-Panel** SPEAKER → OFF → INPUT Ch1 → INPUT JITT → INPUT COMM → INPUT Ch2 → INPUT Ch1&2 → FUNCT Ch1 → FUNCT Ch2 → FUNCT Ch1&2 → DIG IN Ch1 → DIG IN Ch2 → DIG Ch1&2
SYSTem:**SPE**aker:**VOL**ume	**<nu>** 0 to 100 %	**2.6.6** **ANLR-Panel** Skp Volume
SYSTem:**SPE**aker[:STATe]	**ON** **OFF**	**2.6.6** **ANLR-Panel** LOCAL-Taste
SYSTem:**VERS**ion**?**	**<n>** Query only	**2.15.7** **No manual control**
TRACe:**POIN**ts**?**	**CREF**erence1 Query only Query: TRAC:POIN? CREF1 Query response <n> = 0 to 1023	**2.9.1.3** **No manual control**
TRACe:**POIN**ts**?**	CREFerence2 Query only Query: TRAC:POIN? CREF2 Query response: <n> = 0 to 1023	**2.9.1.3** **No manual control**

Command	Parameter	Section
TRACe:**POIN**ts**?**	**REF**erence1 Query only Query: TRAC:POIN? REF1 Query response: <n> = 0 to 1023	**2.9.1.3** **No manual control**
TRACe:**POIN**ts**?**	**REF**erence**2** Query only Query: TRAC:POIN? REF2 Query response: <n> = 0 to 1023	**2.9.1.3** **No manual control**
TRACe:**POIN**ts**?**	**TRAC**e**2** Query only Query: TRAC:POIN? TRAC2 Query response: <n> = 0 to 1023 Query only	**2.9.1.3** **No manual control**
TRACe:**POIN**ts**? LIST1**	**LIST1** Query only Query: TRAC:POIN? LIST1 Query response: <n> = 0 to 1023	**2.9.1.3** **No manual control**
TRACe:**POIN**ts**? LIST2**	**LIST2** Query only Query: TRAC:POIN? LIST2 Query response: <n> = 0 to 1023	**2.9.1.3** **No manual control**
TRACe:**POIN**ts**? TRAC**e**1**	**TRAC**e**1** Query only Query: TRAC:POIN? TRAC1 Query response: <n> = 0 to 1023 Query only	**2.9.1.3** **No manual control**
TRACe[:DATA]	**LIST1,<n>**{**,<n>**} Queryform: TRACe? LIST1	**2.9.1.3** **No manual control**
TRACe[:DATA]	**LIST2,<n>**{**,<n>**} Queryform: TRACe? LIST2	**2.9.1.3** **No manual control**
TRACe[:DATA]	**REF**erence**1,<n>**{**,<n>**} Queryform: TRACe? REF1	**2.9.1.3** **No manual control**

Command	Parameter	Section
TRACe[:DATA]	**REF**erence**2,<n>**{**,<n>**} Queryform: TRACe? REF2	**2.9.1.3** **No manual control**
TRACe[:DATA]	**TRAC**e**1,<n>**{**,<n>**} Queryform: TRACe? TRAC1	**2.9.1.3** **No manual control**
TRACe[:DATA]	**TRAC**e**2,<n>**{**,<n>**} Queryform: TRACe? TRAC2	**2.9.1.3** **No manual control**
TRACe[:DATA] **CREF**erence1**,**	**<n>**{**,<n>**} Queryform: TRACe? CREF1	**2.9.1.3** **No manual control**
TRACe[:DATA] **CREF**erence**2,**	**<n>**{**,<n>**} Queryform: TRACe? CREF2	**2.9.1.3** **No manual control**
TRIGger:**CHAN**nel	**CH1** **CH2** **GENB**urst	**2.6.5.14** **ANLR-Panel** Trig Src → CHAN 1 → CHAN 2 → GEN BURST
TRIGger:**COUN**t	**<nu>** 2 to 1024	**2.6.4** **ANLR-Panel** Points
TRIGger:**DEL**ay	**<nu>** 0 to 10 s	**2.6.4** **ANLR-Panel** Delay
TRIGger:**FREQ**uency:**VAR**iation	**<nu>** > 0.1 to 50%	**2.6.4** **ANLR-Panel** Variation
TRIGger:**LEV**el	**<nu>** Analog instruments –50 V to 50 V Digital instrument –1 FS to 1 FS	**2.6.5.14** **ANLR-Panel** Trig Level
TRIGger:**SLOP**e	**POS**itive **NEG**ative	**2.6.5.14** **ANLR-Panel** Trig Slope → RISING → FALLING

Command	Parameter	Section
TRIGger:**SOUR**ce	**IMM**ediate **TIM**er **CH1F**req **CH2F**req **CH1L**evel **CH2L**evel **CH1T**rigger **CH2T**rigger **TCH**art **CH1R**apidfreq **CH2R**apidfreq **CH1E**dgetrigger **CH2E**dgetrigger	**2.6.4** **ANLR-Panel** START COND → AUTO → TIME → CH1Freq | CH2Freq → CH1Level | CH2Level → LEV TRG CH1 | LEV TRG CH2 → TIME CHART → FRQ FST CH1 → FRQ FST CH2 → EDG TRG CH1 → EDG TRG CH2
TRIGger:**TIM**er	**<nu>** 10 ms to 2000 s	**2.6.4** **ANLR-Panel** Timetick
TRIGger:**VOLT**age:**VAR**iation	**<nu>** > 0.1 ,900% oder > 0.01 to 20 dB	**2.6.4** **ANLR-Panel** Variation

3.12 IEC/IEEE-Bus Interface

The UPL is equipped with an IEC/IEEE-bus interface as a standard. The mating 25-pin connector according to standard IEC 625.1/IEEE 488 is at the rear of the instrument. A controller for remote control can be connected via the interface. Interconnection is made via a shielded cable.

3.12.1 Interface Characteristics

- 8-bit parallel data transfer
- bidirectional data transfer
- three line handshake
- high data transfer rate of max. 350 kByte/s
- up to 15 devices can be connected
- maximum length of interconnecting cable 15 m (single connection 2m)
- wired OR if several instruments are interconnected in parallel.

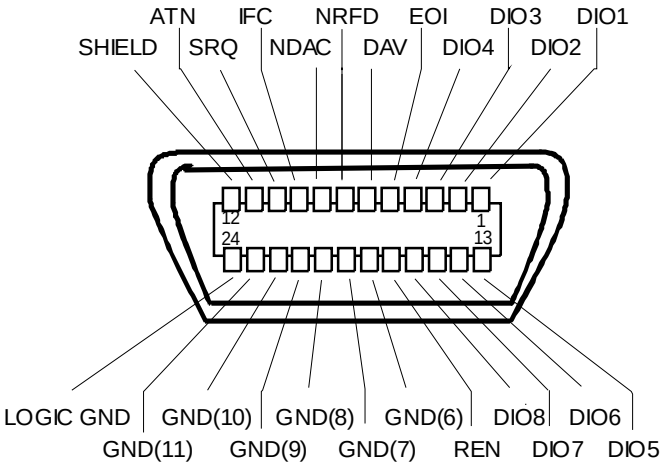


Fig. 3-7 Pin assignment of IEC/IEEE-bus interface

3.12.2 Bus Lines

1. Data bus with 8 lines DIO 1 to DIO 8.
The transmission is bit-parallel and byte-serial in the ASCII/ISO code. DIO1 is the bit of lowest order, DIO8 the bit of highest order.

2. Control bus with 5 lines.
IFC (Interface Clear),
active low resets the interfaces of all connected instruments to the default setting.

ATN (Attention),

active low signals the transmission of interface messages
inactive high signals the transmission of device-dependent messages.
SRQ (Service Request),
active low enables a device connected to send a service request to the controller.

REN (Remote Enable),
active low permits the switchover to remote control.

EOI (End or Identify),
has two functions in connection with ATN:
active low marks the end of data transmission with ATN = HIGH
active low triggers a parallel poll with ATN = LOW.

3. Handshake Bus mit drei Leitungen.
DAV (Data Valid),
active low signals a valid data byte on the data bus.

NRFD (Not Ready For Data),
active low signals that one of the devices connected is not ready for data transfer.

NDAC (Not Data Accepted),
active low as long as the instrument connected is accepting the data present on the data bus.

3.12.3 Interface Functions

Instruments which can be remote-controlled via IEC/IEEE bus can be equipped with different interface functions. Table A-1 lists the interface functions appropriate for the UPL.

Table 3-10 Interface functions

Control character	Interface function
SH1	Handshake source function (source handshake)
AH1	Handshake drain function (acceptor handshake)
L3..L4/LE3..LE4	Listener function.
T5..T8/TE5..TE8	Talker function, ability to respond to serial poll
SR1	Service request function
PP1	Parallel poll function
RL1	Remote/Local switchover function
DC1	Resetting function (Device Clear)
DT1	Trigger function (Device Trigger)
C1...C27	Controller function (with optional software UPL-B10only)

3.13 Interface Messages

Interface messages are transmitted to the UPL on the data lines, with the attention line being active (LOW). They enable the communication between instrument and controller.Das Parsen von Befehlen entfällt, dadurch wird eine hohe Verarbeitungsgeschwindigkeit erreicht.

3.13.1 Common Commands

The common commands are encoded in the range 10 through 1F hex. They are effective for all instruments connected to the bus without addressing them before.

Table 3-11 Common commands

Command	BASIC command for R&S controllers	Effect on UPL
DCL (Device Clear)	`IECDCL`	Aborts the processing of the commands just received and sets the command processing software to a defined initial state. Does not change the instrument setting.
IFC (Interface Clear)	`IECIFC`	Resets the interfaces to the default setting.
LLO (Local Lockout)	`IECLLO`	The REM/LOCAL key is disabled.
SPE (Serial Poll Enable)	`IECSPE`	Ready for serial poll
SPD (Serial Poll Disable)	`IECSPD`	End of serial poll
PPU (Parallel Poll Unconfigure)	`IECPPU`	End of the parallel-poll state

3.13.2 Addressed Commands

The addressed commands are encoded in the range 00 through 0F hex. They are only effective for instruments addressed as listeners.

Table 3-12 Addressed commands

Command	BASIC command for R&S controllers	Effect on UPL
SDC (Selected Device Clear)	`IECLAD 20: IECSDC`	Aborts the processing of the commands just received and sets the command processing software to a defined initial state. Does not change the instrument setting.
GET (Group Execute Trigger)	`IECLAD 20: IECGET`	Triggers all actions which are waiting for a trigger event. This command is identical with the commands INIT and *TRG.
GTL (Go to Local)	`IECLAD 20: IECGTL`	Transition to the "Local" state (manual control)
PPC (Parallel Poll Configure)	`IEC PCON 20,1,6`	Configure instrument for parallel poll In the command to the left, the UPL with an address of 20 is reporting a 1 on line 6.

3.14 List of Error Messages

The following list contains error messages which can be obtained via IEC/IEEE bus or in manual operation. Negative error numbers are SCPI-defined. The positive error number "111" marks device-specific errors.

When and how is the error queue output?

If the controller does not succeed in transmitting a command to UPL, an error message appears in the information line on the screen and a short signal can be heard.

This error message can be indicated in the control program by outputting the error queue after every command sent to UPL. This should be controlled by SRQ, i.e. only when bit 3 (Device Dependent Error), bit 4 (Execution Error) or bit 5 (Command Error) of the event status register is set, but can also happen at any location in the control program.
If no error has been detected, the message is
0, "No Error"

Example:

```
1000Errqueue:
1010 IEC OUT 20,"SYST:ERR?":                    ' Output error queue until it is empty!
1020 IEC IN 20,E$.                              ' Read error information
1030 IF LEFT$(E$,1)="0" THEN RETURN:            ' 0, "No Error", error queue empty!
1040 PRINT "Contents of error queue:"; E$:GOTO Errqueue
```

The table contains the error number in the left-hand column. In the right-hand column, the error text being entered into the error/event queue or being displayed is printed in bold face. Below the error text, there is an explanation as to the respective error.

3.14.1 SCPI-Specific Error Messages

No Error

Error number	**Error query response** Explanation
0	**No error** This message is output if the error queue is completely empty.

3.14.2 Command Error

Command error — sets bit 5 in the ESR register.

Error number	**Error query response** Explanation
-100	**Command Error** The command is faulty or invalid.
-101	**Invalid Character** A syntactic element contains a character which is invalid for that type. Example: A header containing an ampersand, "SOURCE&".
-102	**Syntax error** An unrecognized command or data type was received. Example: A string was received when the device does not accept strings.
-103	**Invalid separator** The device was expecting a separator and received an illegal character. Example: The semicolon was omitted after a program message unit.
-104	**Data type error** The device recognized a data element different than one allowed. Example: Numeric or string data are expected but block data was received.
-105	**GET not allowed** A Group Execute Trigger (GET) was received within a program message.
-112	**Program mnemonic too long** The header contains more than 12 characters.
-113	**Undefined header** The header is syntactically correct, but it is undefined for the UPL. Example: *XYZ is not defined for any device.
-114	**Header suffix out of range** A nonheader character has been encountered in the header element parsed by the device. Example: SOURce3 is not defined for any device.
-123	**Exponent too large** The magnitude of the exponent is larger than 320000.
-124	**Too many digits** The mantissa of a decimal numeric data element contains more than 255 digits (excluding leading zeros).
-128	**Numeric data not allowed** A legal numeric data element was received, but the device does not accept one in this position for the header. Example: Command SOURce:FREQuency:MODE requires the indication of a text parameter.
-131	**Invalid suffix** The suffix is inappropriate for this device. Example: nHz is not defined?
-134	**Suffix too long** The suffix contains more than 12 characters.
-138	**Suffix not allowed** A suffix was received after a numeric element which does not allow suffixes. Example: Command *RCL does not permit indicating a suffix.
-141	**Invalid character data** Either the character data element contains an invalid character or the particular element received is not valid for the header. Example: Write error with parameter indication; SOURce:FREQuency:MODE FIKSed.

Command error (cont.)

Error number	**Error query response** Explanation
-144	**Character data too long** The character data element contains more than 12 characters.
-148	**Character data not allowed** A legal character data element was encountered where prohibited by the device. Example: Command *RCL requires the indication of a number.
-151	**Invalid string data** A string data element was expected, but was invalid for some reason. Example: An END message was received before the terminal quote character.
-158	**String data not allowed** A legal string data element was encountered where prohibited by the device.
-161	**Invalid block data** A block data element was expected, but was invalid for some reason. Example: An END message was received before the length was satisfied.
-168	**Block data not allowed** A legal block data element was encountered where prohibited by the device.
-178	**Expression data not allowed** A legal expression data element was encountered where prohibited by the device.

3.14.3 Execution Error

Execution error — sets bit 4 in the ESR register.

Error number	**Error query response** Explanation
-200	**Execution error** An error occurred when executing a received command.
-220	**Parameter error** A program data element related error occurred.
-221	**Settings conflict** A legal program data element was parsed but could not be executed due to the current device state.
-222	**Data out of range** The received data element was syntactically correct but could not be executed because the value was outside the legal range as defined by the device. Example: Command TRIG:DEL only permits entries in the range of 50 ms to 10 sec.
-241	**Hardware missing** A legal program command or query could not be executed because of missing device hardware; for example, an option was not installed.

3.14.4 Device-Specific Error

Device-specific Error — sets bit 3 in the ESR register.

Error number	**Error query response** Explanation
-300	**Device-specific error** Generic device-dependent error for devices that cannot detect more specific errors.

3.14.5 Query Error

Query Error — sets bit 2 in the ESR register.

Error number	**Error query response** Explanation
-400	**Query error** Generic device-dependent query error for devices that cannot detect more specific errors.
-420	**Query UNTERMINATED** A condition causing an UNTERMINATED query error occurred. Example: The device was addressed to talk and an incomplete program message was received.
-430	**Query DEADLOCKED** A condition causing an DEADLOCKED query error occurred. Example: Both input buffer and output buffer are full and the device cannot continue.

3.14.6 UPL-Specific Error Messages

Device-specific Error — sets bit 3 in the ESR register.

Error number	**Error query response** Explanation
111	After the error queue has been output, all device-dependent errors are shown with the code number 111 and a self-explanatory text having a maximum length of 50 characters, e.g. 111, „Device dep. error; Insufficient disk space! Cannot write file".

3.15 Examples of IEC/IEEE-Bus Programming (Hints and Program Examples)

The examples illustrate UPL programming and may be taken as a basis for solving more complex programming tasks.

All programming examples for IEC/IEEE-bus control in the path C:\UPL\IEC EXAM\EXAM1.BAS ff and those of the UPL software are written in R&S BASIC.

These examples can be run directly on a controller using a suitable IEC/IEEE-bus card and R&S BASIC (see (siehe 3.15.1, R&S BASIC). Files with the extension .SAC are setup files requiring the program examples for setting the UPL. Files with the extension .TXT give the program code of the examples as ASCII file so that they can be accessed with any editor.

If other languages are used for controlling the UPL by means of IEC/IEEE commands, the given sequence of commands may be used but for a few exceptions, as it is independent of the programming language.

3.15.1 R&S BASIC

The following program routines and examples and those contained in the UPL software are ready for use provided R&S BASIC and the R&S IEC/IEEE-bus card have been installed in the controller.

Note:

R&S BASIC and the R&S IEC/IEEE-bus interface card may be ordered from your local sales engineer under the designation ***PAT-B1, Order No. 1007.1150.02****.*

3.15.2 IEC/IEEE-Bus Control after Power-Up

After UPL power-up, messages may be displayed which have to be acknowledged using the ENTER or CANCEL key. If the instrument is to be remote-controlled after power-up irrespective of any messages displayed, it is recommended to start the UPL with the aid of the command line parameter "-r". In this case messages do not require to be acknowledged by a keystoke, the UPL is started with a suitable setup and immediately ready for remote control.

Example: C:\UPL\UPL_UI-r <CR>

3.15.3 Command Logging - Converting UPL-B10 Commands into IEC/IEEE-Bus Commands

All commands required for setting the UPL or for measurements are listed in section **3.10 IEC-bus Commands** and section **3.11 Alphabetical List of IEC-bus Commands** of the UPL manual. To avoid a tedious search for the commands required, the Universal Sequence Controller, UPL-*B10*, which is simply called *B10* below and available as an option, permits all manual setting procedures to be stored as *B10* commands and to be converted into the IEC/IEEE-bus control program.

Note:

The Universal Sequence Controller option UPL-B10 may be ordered from your local sales engineer under the ***Order No. 1078.3856.02****. For installation please refer to the Installation Instruction enclosed with Option UPL-B10.*

The procedure is explained by way of a sweep with subsequent display of the sweep curve:

- Prior to recording the desired command sequence, press key F2 to activate command logging ("**logging on**" displayed at bottom right). All settings performed subsequently in the UPL are recorded as a sequence of B10 commands.
- When the setting sequence has been completed, disable command logging by pressing the F2 key again ("**logging off**" displayed at bottom right).
- Call up the B10 program by pressing F3. Upon pressing F8 (LISTe), a list of commands is displayed - for the time being without comments - which has been generated by means of B10 commands and correspond to the settings just performed (in the example **Settings for a frequency sweep**).

```
10 UPL OUT "*RST"
20 UPL OUT "DISP:MODE COLB"
30 UPL OUT "SENS:VOLT:APER:MODE AFAS"
40 UPL OUT "SENS:FILT:AWE ON"
50 UPL OUT "DISP:TRAC:OPER CURV"
60 UPL OUT "DISP:TRAC:X:SPAC LOG"
70 UPL OUT "SOUR:SWE:MODE AUTO;:SOUR:FREQ:MODE SWE1"
```

Listing of B10 commands without comments.
Comments will we added manually later on.

- Use command ASAVE "A:LOGGING.TXT" to store the listing as ASCII file on a floppy disk.
- Return to the UPL operating level by pressing F3 and quit the UPL (CTRL+F9).
- Using an editor of your own choice at the DOS operating system level, edit the A:LOGGING.TXT file in a form suitable for IEC/IEEE-bus control by replacing |**UPL OUT|** by |**IEC OUT 20,|** and store it on the floppy disk.

```
10 IEC OUT 20,"*RST"
20 IEC OUT 20,"DISP:MODE COLB"
30 IEC OUT 20,"SENS:VOLT:APER:MODE AFAS"
40 IEC OUT 20,"SENS:FILT:AWE ON"
50 IEC OUT 20,"DISP:TRAC:OPER CURV"
60 IEC OUT 20,"DISP:TRAC:X:SPAC LOG"
70 IEC OUT 20,"SOUR:SWE:MODE AUTO;:SOUR:FREQ:MODE SWE1"
```

Listing of IEC/IEEE-bus commands without comments.
Comments will be added manually later on!

- Establish connection to IEC/IEEE-bus controller and start R&S BASIC on the controller.
- Insert floppy holding the "LOGGING.TXT" file into the controller.
- Load the listing as ASCII file from the floppy with command ALOAD "A:LOGGING.TXT" and extend it as required by IEC/IEEE-bus-specific commands and comments.

```
10 IEC TERM 10:'              Controller waits for Line Feed as terminator
20 IEC TIME 10000:'                              IEC/IEEE-bus timeout 10 s
30 IEC OUT 20,"*RST;*WAI":'                              UPL default setup
40 IEC OUT 20,"DISP:MODE COLB":'                         Changes of colour
50 IEC OUT 20,"SENS:VOLT:APER:MODE AFAS":'          High measurement speed
60 IEC OUT 20,"SENS:FILT:AWE ON":'          Switches on A-weighting filter
70 IEC OUT 20,"DISP:TRAC:OPER CURV":'                 Selects trace display
80 IEC OUT 20,"DISP:TRAC:X:SPAC LOG":'                   Logarithmic X axis
0 IEC OUT 20,"SOUR:SWE:MODE AUTO;:SOUR:FREQ:MODE SWE1":'      Autom. Sweep
100 IEC OUT 20,"DISP:CONF AP":'                       Selects graphic window
110 IEC OUT 20,"INIT:CONT OFF;*WAI":'       Starts sweep and waits for end
120 IEC OUT 20,"DISP:TRAC:Y:AUTO ONCE":'                      Autoscaling
130 IEC LAD 20: IEC GTL:'                        Return to manual control
140 END
```

(listing of IEC/IEEE-bus commands with additional commands and comments)

- Start the ready-to-run program with RUN or F2.

3.15.4 Initialization and Default Status

The controller must be informed that the end character for query replies from the UPL is < Line Feed> and that it has to wait for max. 10 s after a trigger command or an IEC-IN command before it signals a timeout. The IEC/IEEE-bus Status Registers and the UPL are reset to the default state.
For default settings of UPL see annex **A UPL Default Setup**

```
10 'Initialization of controller
10 IEC TERM 10:'               Controller waits for Line Feed as terminator
20 IEC TIME 10000:'       After 10 s controller signals IEC/IEEE-bus timeout
10 'Initialization of UPL
20 IECOUT 20,"*CLS":'                                  Resets Status Register
30 IECOUT 20,"*RST:*WAI":'    Resets device and waits for end of calibration
     :
```

3.15.5 Sending Instrument Setting Commands

In this routine, the UPL is set to maximum speed for triggered measurements.

```
10 IEC TERM 10:'                  Terminator for query replies is Line Feed
20 IEC TIME 10000:'             Max. waiting time for query replies is 10 s
30 IEC OUT 20,"*CLS":'                  Resets IEC/IEEE-bus Status Register
40 IEC OUT 20,"*RST;*WAI":'     UPL default setting, *WAI waits for calibr.
50 IEC OUT 20,"DISP:ANN OFF":'                   Switches off result display
60 IEC OUT 20,"SENS2:FUNC 'OFF'":'       Switches off input-peak measurement
70 IEC OUT 20,"SENS3:FUNC 'OFF'":'        Switches off frequency measurement
80 IEC OUT 20,"SENS:VOLT:APER:MODE VAL"
90 IEC OUT 20,"SENS:VOLT:APER 1ms":'        Sets a measurement speed of 1 ms
     :
```

3.15.6 Switchover to Manual Control

```
REM ------ Switching the instrument to manual control -------
     :
100 IEC LAD 20:'  Addresses UPL
110 IEC GTL:'     Sets UPL to local
     :
```

3.15.7 Readout of Instrument Settings

The settings made in section **3.15.5 Sending Instrument Setting** Commands, are read out. In this case the short form of the commands is used.

```
     :
110 '------- Readout of instrument settings ---------
120 IEC OUT 20,"DISP:ANN?":'           Query for setting the display Update
130 IEC IN 20,A$: PRINT A$:'                                Displays OFF
140 IEC OUT 20,"SENS2:FUNC?":'             Query for input peak measurement
150 IEC IN 20,A$: PRINT A$:'                                Displays OFF
160 IEC OUT 20,"SENS:VOLT:APER?":'              Query for measurement speed
170 IEC IN 20,A$: PRINT A$:'                                Displays 1.E-3
     :
```

3.15.8 Readout of Measurement Results

Numeric results of a specific measurement function, ie input peak, input RMS or frequency and phase measurements, can be taken from the UPL in a triggered or non-triggered form.

For triggering measurements and sweeps see section **3.6.7 Triggering a Measurement/Sweep**
Waiting for the end of a measurement or sweep is described in section **3.6.8.2 Wait for End** of Measurement/Sweep.

The display below gives the IEC/IEEE-bus commands used for selecting and calling up measurement results.

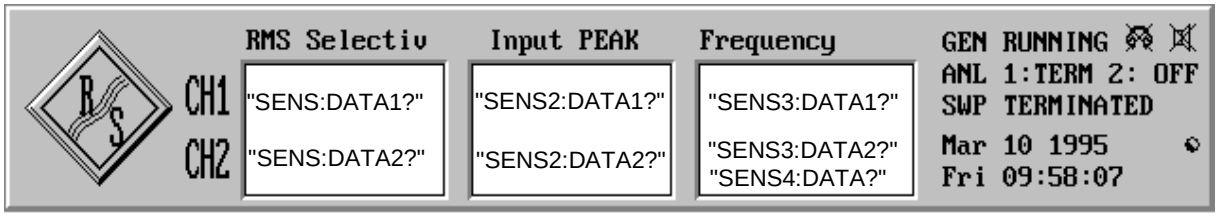


Fig. 3-8 Result display and associated IEC/IEEE-bus commands

Table 3-13 Selecting measurement results

Measurement function	Channel setting	Selection and readout of measurement results
Function	IECOUT 20,"INP:SEL CH1"	IECOUT 20,"SENS:DATA1?":IECIN 20,Func$
IECOUT 20, "SENS:FUNC 'RMS'"	IECOUT 20,"INP:SEL CH2"	IECOUT 20,"SENS:DATA2?":IECIN 20,Func$
"SENS:FUNC 'RMSS'" "SENS:FUNC 'PEAK'" "SENS:FUNC 'QPE'" "SENS:FUNC 'DC'" "SENS:FUNC 'THD'" "SENS:FUNC 'THDN'" "SENS:FUNC 'MDIST'" "SENS:FUNC 'DFD'" "SENS:FUNC 'DIM'" "SENS:FUNC 'WAF'" "SENS:FUNC 'POL'" "SENS:FUNC 'FFT'" "SENS:FUNC 'FILT'" "SENS:FUNC 'WAV'" "SENS:FUNC 'COHE'"	IECOUT 20,"INP:SEL CH1A" or IECOUT 20,"INP:SEL CH1I" or IECOUT 20,"INP:SEL CH2I" or IECOUT 20,"INP.SEL BOTH"	IECOUT 20,"SENS:DATA1?":IECIN 20,FuncA$ IECOUT 20,"SENS:DATA2?":IECIN 20,FuncB$
Input PEAK or Input RMS	IECOUT 20,"INP:SEL CH1"	IECOUT 20,"SENS2:DATA1?":IECIN 20,Ip$
IECOUT 20,	IECOUT 20,"INP:SEL CH2"	IECOUT 20,"SENS2:DATA2?":IECIN 20,Ip$
"SENS2:FUNC 'PEAK'" "SENS2:FUNC 'RMS'"	IECOUT 20,"INP:SEL CH1A" or IECOUT 20,"INP:SEL CH1I" or IECOUT 20,"INP:SEL CH2I" or IECOUT 20,"INP.SEL BOTH"	IECOUT 20,"SENS2:DATA1?":IECIN 20,IpA$ IECOUT 20,"SENS2:DATA2?":IECIN 20,IpB$
Frequency measurement	IECOUT 20,"INP:SEL CH1"	IECOUT 20,"SENS3:DATA1?":IECIN 20,Freq$
IECOUT 20,	IECOUT 20,"INP:SEL CH2"	IECOUT 20,"SENS3:DATA2?":IECIN 20,Freq$
"SENS3:FUNC 'FREQ'"	IECOUT 20,"INP:SEL CH1A" or IECOUT 20,"INP:SEL CH1I" or IECOUT 20,"INP:SEL CH2I" or IECOUT 20,"INP.SEL BOTH"	IECOUT 20,"SENS3:DATA1?":IECIN 20,FreqA$ IECOUT 20,"SENS3:DATA2?":IECIN 20,FreqB$

Measurement function	Channel setting	Selection and readout of measurement results
Frequency and phase measurement IECOUT 20, "SENS3:FUNC 'FQPH'" with two-channel measurements only	IECOUT 20,"INP:SEL CH1A" or IECOUT 20,"INP:SEL CH1I" or IECOUT 20,"INP:SEL CH2I" or IECOUT 20,"INP.SEL BOTH"	IECOUT 20,"SENS3:DATA1?":IECIN 20,Freq$ IECOUT 20,"SENS4:DATA?":IECIN 20,Phas$
Frequency and group delay measurement IECOUT 20, "SENS3:FUNC 'FQGR'" with two-channel measurements only	IECOUT 20,"INP:SEL CH1A" oder IECOUT 20,"INP:SEL CH1I" oder IECOUT 20,"INP:SEL CH2I" oder IECOUT 20,"INP.SEL BOTH"	IECOUT 20,"SENS3:DATA1?":IECIN 20,**Freq$** IECOUT 20,"SENS4:DATA?":IECIN 20,**Grpl$**

3.15.8.1 Readout of Triggered Measurements

Readout of triggered measurement is demonstrated by way of an RMS measurement:

```
     :
100 IEC OUT 20,"SENS:FUNC 'RMS'":'                    Sets RMS measurement
110 IEC OUT 20,"INPUT:SELECT CH1":'                        Sets channel 1
      :
210 'One of three trigger modes can be selected
220 INPUT "Select trigger mode INIT [I], GET [G] or *TRG [T]:";Tg$
      :
330 IEC OUT 20,"INIT:CONT OFF;*WAI":'       Selects single-measurement mode
      :
480 IF Tg$="I" THEN IEC OUT 20,"INIT":   GOTO In
490 IF Tg$="G" THEN IEC LAD 20: IEC GET :GOTO In
500 IF Tg$="T" THEN IEC OUT 20,"*TRG":   GOTO In
      :
620In:
630 IEC OUT 20,"*WAI":'          Sends next IEC/IEEE-bus command only after
640 '                                        a result has been obtained.
650 IEC OUT 20,"SENS:DATA1?":'               Selects result of channel 1
660 IEC IN 20,Mwert$:'                       Reads in measurement result
667 PRINT Mwert$:'                            Outputs measurement result
      :
```

3.15.8.2 Readout of Non-Triggered Measurements

Contrary to triggered measurements, results of non-triggered measurements are read from the buffer at maximum speed without considering settling so that the same value is displayed several times before a new value is output.
Since results of non-triggered measurements are read out as an exception, no detailed description will be given. Reading out triggered measurement results as described in section **3.15.8.1 Readout of Triggered** Measurements, is to be preferred in any case.

```
    :
190 IEC OUT 20,"*RST;*WAI": ’                        Sets up RMS measurement
200 IEC OUT 20,"DISP:ANN OFF":’                   Switches off result display
210 IEC OUT 20,"SENS2:FUNC ’OFF’":’             Switches off PEAK measurement
220 IEC OUT 20,"SENS3:FUNC ’OFF’":’       Switches off frequency measurement
230 IEC OUT 20,"SENS:VOLT:APER:MODE AFAS":’   Selects high measurement speed
240 IEC OUT 20,"INIT:CONT ON":’              Selects continuous measurements
250 IEC LAD 20: IEC GET :’             Triggers with Group Executive Trigger
260 ’"*WAI" omitted!
270 IEC OUT 20,"SENS:DATA1?":’               Selects RMS result on channel 1
280 IEC IN 20,Mwert$:’                           Reads in measurement result
    :
```

3.15.9 Sweep Setting/Trigger

3.15.9.1 Generator Sweep

With a generator sweep of the UPL, output frequency, output level, burst duration, etc. are swept between the start/stop values .
A sweep is triggered with `INIT`, `*TRG` or `GET` (see section **3.6.7 Triggering a Measurement/Sweep**).

```
    :
50 IEC OUT 20,"SOUR:SWE:MODE AUTO;:SOUR:FREQ:MODE SWE1":’       Freq. sweep
60 IEC OUT 20,"SOUR:FREQ:STAR 100 HZ":’         Sweep start frequency 100 Hz
70 IEC OUT 20,"SOUR:FREQ:STOP 10 KHZ":’          Sweep stop frequency 10 kHz
80 IEC OUT 20,"SOUR:SWE:FREQ:POIN 15":’                      15 sweep points
90 IEC OUT 20,"SENS:FILT:AWE ON": ’              Selects A-weighting filter
100 IEC OUT 20,"DISP:TRAC:OPER CURV": ’               Selects trace display
110 IEC OUT 20,"INIT:CONT OFF;*WAI":’  Triggers single sweep, waits for end
120 IEC OUT 20,"DISP:CONF AP":’ Selects analyzer display with graphic window
130 IEC LAD 20: IEC GTL: ’                             Displays sweep curve
    :
```

See also section **3.15.10.3 Configuration for Maximum** Sweep Speed and **3.15.10.3.1 Generator Sweep**.

3.15.9.2 External Sweep

With external sweep of UPL, measurement results are obtained by frequency and level variations at channel 1 or 2 of the analyzer input.
The example below illustrates an external frequency sweep.

```
     :
300 '*** Setting parameters for external sweep
310 IEC OUT 20,"TRIG:SOUR CH1F": '        Setup for external frequency sweep
320 IEC OUT 20,"ARM:LEV:MIN 100 mV": '                  Minimum level 100 mV
330 IEC OUT 20,"ARM:FREQ:STAR 100 Hz": '              Start frequency 100 Hz
340 IEC OUT 20,"ARM:FREQ:STOP 16 kHz": '               Stop frequency 16 kHz
350 IEC OUT 20,"TRIG:FREQ:VAR 4.5": '       Var. just below sweep step width
360 IEC OUT 20,"DISP:CONF AP": '                      Selects graphic window
370 PRINT "External sweep is started - recording is in progress!"
380 IEC OUT 20,"INIT:CONT OFF;*WAI": '         Triggers external single sweep
390 IEC OUT 20,"SYST:BEEP:STAT ON": ' Waits for sweep end with dummy command
400 IEC LAD 20: IEC GTL : '                                    Displays curve
     :
```

See also section **3.15.10.3 Configuration for Maximum** Sweep Speed and **3.15.10.3.2 External Sweep**.

3.15.9.3 RMS-Selektiv-Sweep

With the RMS selective sweep, the center frequency of a bandpass or bandstop filter is swept between the start/stop frequencies and an RMS measurement is carried out after each sweep step.
The sweep is triggered with `"INIT"`, `"*TRG"` or `GET` (see section **3.6.7 Triggering a Measurement/Sweep**)

```
     :
100 IEC OUT 20,"SENS:FUNC 'RMSS':'      Setup for RMS selective measurement
110 IEC OUT 20,"SENS:BAND:MODE PPCT1":'      Bandwidth of bandpass filter 1%
120 IEC OUT 20,"SENS:FREQ:MODE SWE;:SENS:SWE:MODE AUTO":'        Sweep setup
130 IEC OUT 20,"SENS:SWE:SPAC LOG;POIN 50":'       Log. sweep with 50 points
140 IEC OUT 20,"SENS:FREQ:STAR 4000Hz;STOP 16000Hz":'       Start/stop freq.
150 IEC OUT 20,"INIT:CONT OFF;*WAI":'       Triggers sweep and waits for end
160 IEC OUT 20,"DISP:TRAC:OPER CURV":'                  Selects trace display
170 IEC OUT 20,"DISP:CONF AP":'   Selects analyzer panel with graphic window
180 IEC LAD 20: IEC GTL:'                                      Displays curve
     :
```

See also section **3.15.10.3 Configuration for Maximum** Sweep Speed and **3.15.10.3.3 RMS-selective Sweep**.

3.15.10 Tuning - Setup for Maximum Measurement Speed

3.15.10.1 Configuration for Maximum Measurement Speed

To obtain maximum measurement speed, all unnecessary measurements and result displays are to be avoided. This is demonstrated by way of a fast RMS measurement:

```
    :
190 IEC OUT 20,"*RST;*WAI": ’                        Sets up RMS measurement
200 IEC OUT 20,"DISP:ANN OFF":’                   Switches off result display
210 IEC OUT 20,"SENS2:FUNC ’OFF’":’             Switches off PEAK measurement
220 IEC OUT 20,"SENS3:FUNC ’OFF’":’        Switches off frequency measurement
230 IEC OUT 20,"SENS:VOLT:APER:MODE AFAS":’   Selects high measurement speed
240 IEC OUT 20,"INIT:CONT OFF":’                  Selects single measurement
250 IEC LAD 20: IEC GET :’          Triggers sweep with Group Execute Trigger
260 IEC OUT 20,"*WAI":’ *WAI                     Waits for measurement results
270 IEC OUT 20,"SENS:DATA1?":’               Selects RMS results on channel 1
280 IEC IN 20,Mwert$:’                               Reads in measured value
     :
```

3.15.10.2 Adapting Measurement Speed to Signal Frequency

Table 3-14 Hints for matching measurement speed and signal frequency

Automatic adaptation of measurement speed to signal frequency	
IEC/IEEE-bus command	**Used for**
`"SENSe[1]:VOLTage:APERture:MODE AFASt"`	RMS and RMS selective measurements: Automatic matching of measurement time and signal frequency by taking into account the signal period. The measurement time is optimally adapted to the input signal. An algorithmic error of max. 1% may occur
`"SENSe[1]:VOLTage:APERture:MODE AUTO"`	RMS and RMS selective measurements: Same as AFASt but with an algorithmic error of max. 0.1%.

Fixed measurement speed	
IEC/IEEE-bus command	**Used for**
`"SENSe[1]:VOLTage:APERture:MODE SFASt"`	RMS and RMS selective measurements: measurement time 50 ms
`"SENSe[1]:VOLTage:APERture:MODE FAST"`	RMS, RMS selective and DC measurements: measurement time 200 ms
`"SENSe[1]:VOLTage:APERture:MODE SLOW"`	RMS and RMS selective measurements: measurement time 1000 ms
`"SENSe[1]:VOLTage:APERture:MODE VALue"` `"SENSe[1]:VOLTage:APERture xxx ms"`	RMS-, RMS selective and DC measurements: measurement time freely selectable

VALue is a **fixed integration time** irrespective of the signal period.

RMS and RMS selective measurements:

- If the measurement time is an integral multiple of the signal period, optimum integration and therefore a steady display is obtained.
- If the measurement time is longer and not an integral multiple of the signal period, an integration is obtained with a beat effect in the display.

DC measurements:

If an AC voltage is superimposed on the DC, the measurement time as integration period has different effects with respect to the signal period of the AC voltage shows different effects:

- If the measurement time is an integral multiple of the signal period, optimum integration is obtained. The AC voltage does not influence the DC measurement result and the display is steady.
- If the measurement time is longer and not an integral multiple of the signal period, an integration is obtained with a beat effect in the display. The AC voltage has no effect on the DC measurement result.
- If the measurement time is shorter than the signal period, the measurement result follows the signal curve. The AC voltage affects the DC measurement result.

Fixed monitor time	
IEC/IEEE-bus command	**Used for**
`"SENSe[1]:VOLTage:INTVtime:MODE SFASt"`	PEAK measurements: time 50 ms
`"SENSe[1]:VOLTage:INTVtime:MODE FAST"`	PEAK measurements: time 200 ms
`"SENSe[1]:VOLTage:INTVtime:MODE SLOW"`	PEAK measurements: time 1000 ms
`"SENSe[1]:VOLTage:INTVtime:MODE FIXed"`	QPK measurements: time 3000 ms
`"SENSe[1]:VOLTage:INTVtime:MODE VALue"` `"SENSe[1]:VOLTage:INTVtime xxx ms"`	PEAK and QPK measurements: time freely selectable Generally no recommendation can be made as to the most suitable monitor time for peak values, as it depends on the input signal and on the measurement itself.

3.15.10.3 Configuration for Maximum Sweep Speed

3.15.10.3.1 Generator Sweep

To obtain maximum sweep speed, switch off all "slowing-down" settings. Particularly the frequency measurement and settling of the low-distortion generator are very time-consuming. If permitted by the measurement, these functions should be switched off.

The following example illustrates which functions are to be switched on or off to obtain maximum sweep speed.
A typical example is a

- **single-channel linear frequency sweep** on channel 1 of
- **100 points,**
- **200 Hz to 4 kHz** and
- **RMS measurement** in
- **AUTO FAST**.

Each speed-reducing function can be switched off separately by means of a command after the *RST command. In this case a single sweep should be triggered once with "INIT:CONT OFF;*WAI" (program line 230) before sweeping is started to avoid the setting times of this command influencing the sweep time. The sweep performed at maximum speed is then triggered with another INIT command (program line 300).

```
 :
100   IEC OUT 20,"*RST;*WAI"
110   IEC OUT 20,"INP:TYPE GEN1"
120   IEC OUT 20,"OUTP:SEL CH2"
130   IEC OUT 20,"SENS:VOLT:RANG 1V": ’                      Prevent ranging
140   IEC OUT 20,"SOUR:LOWD OFF": ’                Low-distortion generator off
150   IEC OUT 20,"SENS:VOLT:APER:MODE GENT": ’      Measurement speed GEN TRACK
160   IEC OUT 20,"SENS2:FUNC ’OFF’": ’               Input peak measurement off
170   IEC OUT 20,"SENS3:FUNC ’OFF’": ’               Frequency measurement off
180   IEC OUT 20,"DISP:ANN OFF": ’                No display of measured values
190   IEC OUT 20,"SOUR:SWE:MODE AUTO;:SOUR:FREQ:MODE SWE1": ’Frequency sweep
200   IEC OUT 20,"SOUR:FREQ:STAR 200;STOP 4000": ’      Sweep 200Hz to 4kHz
210   IEC OUT 20,"SOUR:SWE:FREQ:SPAC LIN": ’                    Linear sweep
220   IEC OUT 20,"SOUR:SWE:FREQ:POIN 100": ’                 100 Sweep points
230   IEC OUT 20,"INIT:CONT OFF;*WAI": ’                Triggers single sweep
240   IEC OUT 20,"SYST:BEEP:STAT OFF": ’     Dummy command waits for sweep end
 :
300 IEC OUT 20,"INIT;*WAI": ’ Einzelswp mit max. Geschwindigkeit ausloesen
310 IEC OUT 20,"SYST:BEEP:STAT OFF": ’ Dummy command waits for sweep end
 :
```

The fastest way to set a suitable sweep is to load an ACTUAL SETUP in which the required settings have already been made.

```
 :
100 IEC OUT 20,"MMEM:LOAD:STAT 0,’C:\UPL\USER\MAXSWP.SAC’;*WAI"
110 IEC OUT 20,"INIT;*WAI": ’                          Triggers single sweep
120 IEC OUT 20,"SYST:BEEP:STAT OFF": ’     Dummy command waits for sweep end
 :
```

The maximum sweep speed attained in this example is approx. 25 ms/step when an UPL with 386 board is used and approx. 8 ms/step when a 486 board is used.

3.15.10.3.2 External Sweep

The example below demonstrates the setup for a fast external frequency sweep with a sweep-signal sequence obtained, for instance, from a CD or tape.

Matching the external frequency sweep to the signal sequence:

- Set a start value which should corresponds to the expected lowest frequency of the sweep-signal sequence.
- Select a stop value slightly below the expected highest sequence frequency (approx. 0.1%) to provide a safe halt criterion.
- elect a variation that is approx. 5 to 10 % lower than the expected frequency variation of the signal sequence to ensure safe triggering and avoid unwanted intermediate values.

The example below illustrates the settings for matching the external sweep of the UPL to the signal sequence generated by a sweep generator for demonstration purposes.
Connect the sweep generator to UPL input UNBAL Ch1, and set and start a

- continuous logarithmic sweep of 100 Hz to 16.1 kHz,
- with a step width 5%,
- and 120-ms time tick

The external sweep parameters of the UPL are set as recommended above and the external single sweep is started.

- Start frequency of ext. sweep = 100 Hz (expected lowest frequency in this case 100 Hz)
- Stop frequency of ext. sweep = 16 kHz (0.1% below the expected highest frequency, in this case 16.1 kHz)
- Variation of ext. sweep = 4.5% (10% below the expected frequency variation of the sequence, in this case 5 %)

Each time the frequency varies by more than 4.5% on channel 1, UPL carries out an RMS measurement until a frequency greater than 16 kHz is measured. After this the external sweep is terminated and the sweep curve is displayed.

With the speed-increasing settings in lines 270 to 290 and the fast frequency measurement for the external sweep in line 310, a signal sequence with a **minimum time tick of 120 ms** can still be reliably measured. For all other settings a longer time is required.

```
     :
260 '*** Speed-increasing settings
270 IEC OUT 20,"DISP:ANN OFF": '                          Display Update off
280 IEC OUT 20,"SENS3:FUNC 'OFF'": '                Frequency measurement off
290 IEC OUT 20,"SENS:VOLT:APER:MODE AFAS": '       RMS meas. speed AUTO FAST
300 '*** Setting of external sweep parameters
310 IEC OUT 20,"TRIG:SOUR CH1R": '    Ext. sweep with fast freq. measurement
320 IEC OUT 20,"ARM:LEV:MIN 100 mV": '              Measurement above 100 mV
330 IEC OUT 20,"ARM:FREQ:STAR 100 Hz": '              Start frequency 100 Hz
340 IEC OUT 20,"ARM:FREQ:STOP 16 kHz": '               Stop frequency 16 kHz
350 IEC OUT 20,"TRIG:FREQ:VAR 4.5": '      Var. just below sweep step width
360 IEC OUT 20,"DISP:CONF AP": '                       Selects graphic window
370 PRINT "External sweep is started - recording goes on!"
380 IEC OUT 20,"INIT:CONT OFF;*WAI": '        Triggers external single sweep
390 IEC OUT 20,"SYST:BEEP:STAT ON": '      Dummy command waits for sweep end
400 IEC LAD 20: IEC GTL : '                                  Displays curve
     :
```

3.15.10.3.3 RMS Selective Sweep

In the example below settings are made for a fast RMS selective sweep. The sweep speed mainly depends on the

- bandwidth of the set bandpass /bandstop filter and the
- center frequency.

The bandpass or bandstop filters are very steep filters with an attenuation of 100 dB. The higher the center frequency the wider the passband range and the faster the settling and therefore the RMS selective measurement.
The example below illustrates a fast RMS selective sweep of a 5-kHz squarewave signal produced by the UPL generator. The narrowband RMS measurement from 4 to 16 kHz yields a spectrum display of the fundamental with the 2nd and 3rd harmonic of the squarewave signal.

```
10 IEC TERM 10: IEC TIME 60000
20 IEC OUT 20,"*RST;*WAI;:DISP:MODE COLB;:INP:TYPE GEN2"
30 IEC OUT 20,"SOUR:FUNC SQU;:SOUR:FREQ 5000Hz;:SOUR:VOLT 1V"
70 IEC OUT 20,"SENS:FUNC 'RMSS':'     Switches on RMS selective measurement
80 '*** Speed-increasing measures
90 IEC OUT 20,"SENS:VOLT:APER:MODE AFAS":'             Fast RMS measurement
100 IEC OUT 20,"SENS2:FUNC 'OFF':'               Input-peak measurement off
110 IEC OUT 20,"SENS3:FUNC 'OFF':'                Frequency measurement off
120 IEC OUT 20,"DISP:ANN OFF":'                          Display Update off
130 '*** Settings for RMS selective sweep
140 IEC OUT 20,"SENS:BAND:MODE PPCT1":'                   Bandpass filter 1%
150 IEC OUT 20,"SENS:FREQ:MODE SWE;:SENS:SWE:MODE AUTO":'         Auto sweep
160 IEC OUT 20,"SENS:SWE:SPAC LOG;POIN 50":'       Log. sweep over 50 points
180 IEC OUT 20,"SENS:FREQ:STAR 4000Hz;STOP 16000Hz":'  Start/stop frequency
190 '*** Settings for graphics display
200 IEC OUT 20,"DISP:TRAC:OPER CURV"
210 IEC OUT 20,"DISP:TRAC:Y:UNIT DBV;:DISP:TRAC:X:AUTO OFF"
220 IEC OUT 20,"DISP:TRAC:X:LEFT 3000Hz;RIGH 17000Hz":'              X scale
230'Measuring the sweep time
240 Z1=TIME
250 IEC OUT 20,"INIT:CONT OFF;*WAI"
260 IEC OUT 20,"DISP:CONF AP":'  Selects analyzer panel with graphic window
270 Z2=TIME: IEC LAD 20: IEC GTL:'                          Displays spectrum
280 PRINT (Z2-Z1)/100;" Sec pro Sweep": END
```

Speed-increasing measures become more effective towards higher center frequencies (> 5 kHz)!

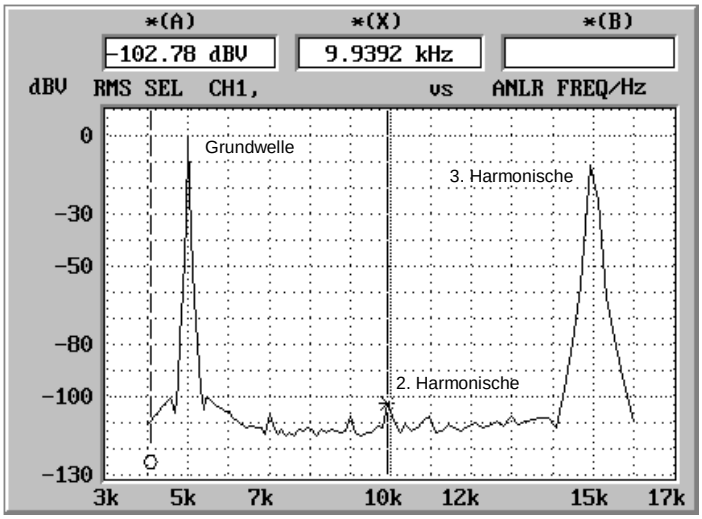


Fig. 3-9 Spectrum of 5-kHz squarewave obtained through RMS selective sweep

3.15.10.3.4 Measurement Speed with Reference to Sampling Mode

If the HIGH RATE sampling mode (CONF:DAI HRM) is used with option UPL-B29 (digital audio 96 kHz), the increased performance requirements caused by the higher clock rates lead to a certain reduction of the measurement speed as compared to the BASE RATE mode (CONF:DAI BRM).See also chapter **2.6.8 Optimizing the Measurement Speed**, section 5, Optimized Utilization of DSP Performance with Reference to the Clock Rate.

3.15.11 List Management

3.15.11.1 Loading Lists into the UPL

3.15.11.1.1 Loading Sweep Lists into the UPL

Depending on the application, data are loaded into the UPL by a variety of commands that can be looked up in section 3.10.6 Commands for Graphical Representation of Results. Loading the frequency values for a frequency sweep is used as an example for demonstrating the procedure.
Load command:
"SOURce:LIST:FREQuency 100.0,300.0,500.0, ... ,20000"
permits a maximum of 1024 values to be loaded.

Use DATA and READ for handling a greater number of frequency values in program code:

```
     :
8110 DATA 100,300,500,700,800,900,1000,2000,3000,4000,5000,6000,7000
8120 DATA 10000,13000,15000,17000,20000,0
8150 IEC OUT 20,"SOUR:SWE:MODE AUTO":'                          AUTO sweep
8160 IEC OUT 20,"SOUR:FREQ:MODE LIST1":'          LIST sweep of frequencies
8170 Bef$="SOUR:LIST:FREQ": '               Lists block data of frequencies
8180 READ Frq
8190 Loop1:
8200 IF Frq<>0 THEN Bef$=Bef$+STR$(Frq)
8210 READ Frq: IF Frq<>0 THEN Bef$=Bef$+",": GOTO Loop1
8230 IEC OUT 20,Bef$:'                              Outputs block command
     :
```

3.15.11.1.2 Loading and Display of Several Traces in the UPL

Not only curves generated in the UPL by a sweep or FFT, or those stored in a file can be graphically displayed but also any data sequences loaded into the UPL by the control program. All UPL capabilities for scaling and unit conversion are used for the graphic display.
The following routine demonstrates loading of three traces into the UPL and their graphic display.

```
     :
200 '********************** Loading traces ***************************
210 IEC OUT 20,"DISP:TRAC:COUN 3":'       Sets number of traces to be loaded
220 IEC OUT 20,"DISP:TRAC:IND 0":'               Selects trace with index 0
230 IEC OUT 20,"TRAC LIST1, 100,1000,5000,15000":'     X values for trace 0
240 IEC OUT 20,"TRAC TRAC1, 0.001,0.01,0.01,0.001":    'Y values for trace 0
250 IEC OUT 20,"DISP:TRAC:IND 1":'               Selects trace with index 1
260 IEC OUT 20,"TRAC LIST1, 100,1500,5500,15000":'     X values for trace 1
270 IEC OUT 20,"TRAC TRAC1, 0.001,0.02,0.02,0.001":    'Y values for trace 1
280 IEC OUT 20,"DISP:TRAC:IND 2":'               Selects trace with index 2
290 IEC OUT 20,"TRAC LIST1, 100,1800,6000,15200":'     X values for trace 2
300 IEC OUT 20,"TRAC TRAC1, 0.001,0.03,0.03,0.001":    'Y values for trace 2
310 IEC OUT 20,"DISP:TRAC:OPER CURV":'             Selects the display mode
320 IEC OUT 20,"DISP:TRAC:Y:AUTO ONCE":'                    Optimizes scale
```

```
330 IEC OUT 20,"DISP:CONF AP":'    Activates analyzer panel + graphic window
340 IEC LAD 20:IEC GTL:'                                   Displays curve
     :
```

Note:

If traces are loaded into the UPL under program control and a sweep is subsequently selected, the trace in the display will be erased and the sweep curve displayed.

3.15.11.1.3 Loading and Displaying of Trace Pairs in the UPL

The program below demonstrates loading of three trace pairs into the UPL, the graphic display, subsequent selection of another unit, rescaling and changing from linear to logarithmic display on the X axis.

```
     :
290 IEC OUT 20,"DISP:TRAC1:FEED 'SENS:DATA'":'              Enables trace A
300 IEC OUT 20,"DISP:TRAC2:FEED 'SENS:DATA'":'              Enables trace B
310 IEC OUT 20,"DISP:TRAC:COUN 3":'Three trace pairs
320 IEC OUT 20,"DISP:TRAC:IND 0":'          Selects trace pair with index 0
330 IEC OUT 20,"TRAC LIST1, 100,1000,5000,15000":'      X values of trace A
340 IEC OUT 20,"TRAC TRAC1, 0.001,0.01,0.01,0.001":'    Y values of trace A
350 IEC OUT 20,"TRAC LIST2, 100,1100,5100,15000":'      X values of trace B
360 IEC OUT 20,"TRAC TRAC2, 0.001,0.02,0.02,0.001":'    Y values of trace B
370 IEC OUT 20,"DISP:TRAc:IND 1":'          Selects trace pair with index 1
380 IEC OUT 20,"TRAC LIST1, 100,1500,5500,15000":'      X values of trace A
390 IEC OUT 20,"TRAC TRAC1, 0.001,0.03,0.03,0.001":'    Y values of trace A
400 IEC OUT 20,"TRAC LIST2, 100,1600,5600,15000":'      X values of trace B
410 IEC OUT 20,"TRAC TRAC2, 0.001,0.04,0.04,0.001":'    Y values of trace B
420 IEC OUT 20,"DISP:TRAc:IND 2":'          Selects trace pair with index 2
430 IEC OUT 20,"TRAC LIST1, 100,1800,6000,15200":'      X values of trace A
440 IEC OUT 20,"TRAC TRAC1, 0.001,0.05,0.05,0.001":'    Y values of trace A
450 IEC OUT 20,"TRAC LIST2, 100,1900,6100,15200":'      X values of trace B
460 IEC OUT 20,"TRAC TRAC2, 0.001,0.06,0.06,0.001":'    Y values of trace B
470 IEC OUT 20,"DISP:TRAC:OPER CURV":'                Selects trace display
480 IEC OUT 20,"DISP:TRAC2:Y:EQU ON":'     Scaling of trace B same as for A
490 IEC OUT 20,"DISP:TRAC1:Y:AUTO ONCE":'                       Autoscaling
500 IEC OUT 20,"DISP:CONF DP":'   Selects DISPLAY panel with graphic window
510 IEC LAD 20: IEC GTL: HOLD 5000:'                Result display for 5 s
520 IEC OUT 20,"DISP:TRAC1:Y:UNIT W":'     Conversion of Y values into Watt
530 IEC OUT 20,"DISP:TRAC1:Y:AUTO ONCE":'                       Autoscaling
540 IEC LAD 20: IEC GTL:HOLD 5000:'                 Result display for 5 s
550 IEC OUT 20,"DISP:TRAC:X:SPAC LOG":'             Log display on X scale
560 IEC OUT 20,"DISP:TRAC1:Y:AUTO ONCE":'                       Autoscaling
570 IEC LAD 20: IEC GTL
     :
```

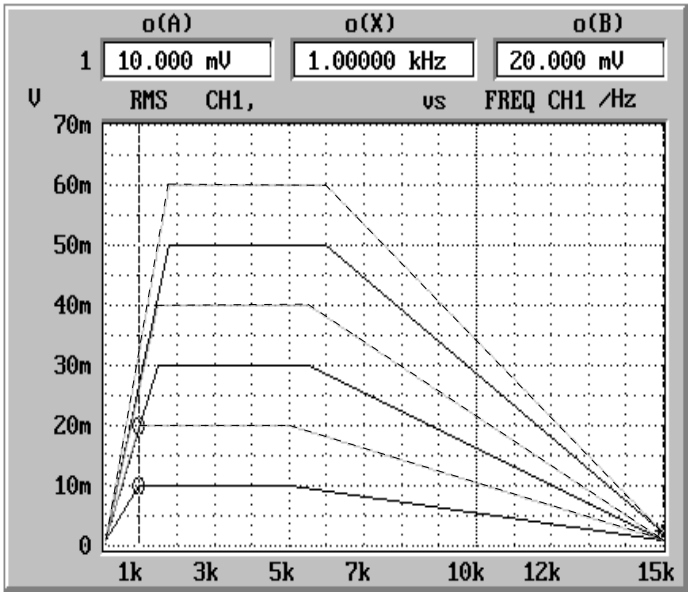


Fig. 3-10 Trace pairs loaded into UPL by control program

3.15.11.2 Readout of Data Lists from the UPL

When data lists are read from the UPL, in compliance with SCPI values are always transferred in the basic unit even if other units have been selected for the trace display. The table below shows the basic units for transferring data of various functions from the UPL to the controller.

Table 3-15 Basic units of data

Measurement function/sweep settings	**Data with basic unit for analog/digital instruments**
"SENS:FUNC 'RMS'"	V/FS
"SENS:FUNC 'RMSS'"	V/FS
"SENS:FUNC 'PEAK'"	V/FS
"SENS:FUNC 'QPE'"	V/FS
"SENS:FUNC 'DC'"	V/FS
"SENS:FUNC 'THD'"	%
"SENS:FUNC 'THDN'"	%
"SENS:FUNC:MMOD THDN|NOIS	%
"SENS:FUNC:MMOD SNDR	% (große Werte)
"SENS:FUNC:MMOD LTHD|LNOI	V/FS
"SENS:FUNC 'DFD'"	%
"SENS:FUNC 'DIM'"	%
"SENS:FUNC 'MDIS'"	%
"SENS:FUNC 'WAF'"	%
"SENS:FUNC 'POL'"	keine Daten
"SENS:FUNC 'FFT'"	V/FS
"SENS:FUNC 'WAV'"	V/FS
"SENS:FUNC 'COHE'"	%
"SENS:FUNC 'RUBB'"	V
"SENS2:FUNC 'PEAK'"	V/FS
"SENS2:FUNC 'RMS'"	V/FS

"SENSe2:FUNCtion 'DIGInpamp'"	V
"SENSe2:FUNCtion 'PHASetoref'"	UI
"SENS3:FUNC 'FREQ'"	Hz
"SENS3:FUNC 'FQPH'"	Grad
"SENS3:FUNC 'FQGR'"	s
"SOUR:FREQ:MODE ..."	Hz
"SOUR:VOLT:MODE ..."	V/FS
"SOUR:ONT:MODE ..."	s
"SOUR:INT:MODE ..."	s
"SENS:FREQ:MODE ..."	Hz

When reading out lists remember that with commands
- "SOUR:LIST:FREQ?"
- "SOUR:LIST:INT?"
- "SOUR:LIST:ONT?"
- "SOUR:LIST:VOLT?"
- "SENS:LIST:FREQ?"

always the X values of the set sweep are read, contrary to commands
- "TRAC? LIST1"
- "TRAC? LIST2"

which cause the X values of the current graphic display to be read.
Note:

Normally, X values are identical for both command groups. They are only different if ***other than the sweep curve*** *is* ***subsequently*** *selected by means of program control or by loading a file while the sweep mode is on .*

3.15.11.2.1 Readout of Lists of up to 1024 Values

A great number of application-specific commands are available for reading out sweep data, FFT data, data loaded from a file or by the control program (see section 3.10.6 Commands for Graphical Representation of Results). The procedure is illustrated by an example for reading out level values of a frequency sweep.
The readout procedure is as follows

"TRAC? TRAC" permits 1024 values to be read.

```
     :
8270 IEC OUT 20,"INIT:CONT OFF;*WAI": '             Triggers a single sweep
     :
8420 IEC OUT 20,"TRAC? TRAC":'                Reads in level data of trace A
8430 IEC IN 20,S$:'    S$ comprises an ASCII string with level values in the
8440 '                        form "1.1234E-003,2.3456E-002,3.4567E-001 ..."
     :
```

3.15.11.2.2 Readout of FFT Lists of more than 1024 Values

The number of values that can be transferred is limited to 1024 lines. If more than 1024 lines are to be read, the data have to be divided in blocks of 1024 values. The table in section 2.6.5.12 FFT informs on the number of lines of the selected FFT which are a function of FFT size and zooming.

In the R&S BASIC program below, the 7488 lines of a 8k-zoom FFT with 8 blocks each (7 x 1024 and 1 x 320 lines) are read and stored in the form of a string
(eg "5.50884e-004,4.1273e-004,1.64638e-004,...") in files FFT_Y1.TXT ... FFT_Y8.TXT.

```
      :
```

```
500 FOR Blkidx=0 TO 7
510  IEC OUT 20,"DISP:TRAC:IND"+STR$(Blkidx):'    Selects block index 0 to 7
520  '                      Reads out FFT lines and stores in string Fftdat$
530  IEC OUT 20,"TRAC? TRAC"
540  IEC IN 20,Fftdat$:'                    Reads in FFT data as ASCII string
550  Filename$="FFT_Y"+RIGHT$(STR$(Blkidx+1),1)+".TXT":'   Defines file name
560  OPENO# 1,Filename$: PRINT# 1,Fftdat$: CLOSE# 1
570 NEXT Blkidx
       :
```

FFT line frequencies are read out in the same way with command
"TRAC? LIST1"

3.15.11.2.3 FFT Lists with Suppressed Noise Floor

Since in most cases the noise floor of an FFT is of no interest, the number of lines can be considerably reduced by including only values exceeding a certain limit in the trace, eg 0.1 V.
To do so set the UPL as described below:

```
                :
100 IECOUT 20,"DISPlay:TRACe:OPERation FFTErrors":'             Sets limits
110 IECOUT 20,"CALCulate:LIMit:UPPer:VALue 0.1V":'..                 >0.1 V
                :
510 IEC OUT 20,"DISP:TRAC:IND 0":'                             Block index 0
520 IEC OUT 20,"TRAC? TRAC":'                  Stores FFT lines above 0.1 V
530 IEC IN 20,Fftdat$:'                        as string data under Fftdat$
                :
```

FFT lines frequencies are read out in the same way using command
"TRAC? LIST1"

3.15.11.2.4 Readout of Several Traces from UPL

If several traces are displayed on the UPL (**"DISP:TRAC:COUN > 1"** set), the required trace can be selected with command "**DISP:TRAC:IND 0 to 17"** and read out with commands **"TRAC? LIST1"** and **"TRACE? TRAC"**:

```
     :
200 '***************** Readout of traces ********************
220 IEC OUT 20,"DISP:TRAC:IND 0":'               Selects trace with index 0
230 IEC OUT 20,"TRAC? LIST1":'       Selects X values of trace with index 0
240 IEC IN 20,X0$:'                Stores X values as ASCII string under X0$
250 IEC OUT 20,"TRAC? TRAC":'        Selects Y values of trace with index 0
260 IEC IN 20,Y0$:'                 Stores Y values as ASCII string under Y0
270 IEC OUT 20,"DISP:TRAC:IND 1":'               Selects trace with index 1
280 IEC OUT 20,"TRAC? LIST1":'       Selects X values of trace with index 1
290 IEC IN 20,X1$:'                Stores X values as ASCII string under X1$
300 IEC OUT 20,"TRAC? TRAC":'        Selects Y values of trace with index 1
310 IEC IN 20,Y1$:'                Stores Y values as ASCII string under Y1$
320 IEC OUT 20,"DISP:TRAC:IND 2":'               Selects trace with index 2
330 IEC OUT 20,"TRAC? LIST1":'       Selects X values of trace with index 2
340 IEC IN 20,X2$:'                Stores X values as ASCII string under X1$
350 IEC OUT 20,"TRAC? TRAC":'        Selects Y values of trace with index 2
     :
```

3.15.11.2.5 Readout of Trace Pairs from UPL

Trace pairs are read out in the way described in section 3.15.11.2.4 Readout of Several Traces from UPL, with the difference
"**TRAC? LIST1**" and "**TRAC? TRAC1**" reads out the X and Y values of trace A
"**TRAC? LIST2**" and "**TRAC? TRAC2**" reads out the X and Y values of trace B

```
     :
200 '******************** Readout of trace pairs**********************
220 IEC OUT 20,"DISP:TRAC:IND 0":'          Selects trace pair with index 0
230 IEC OUT 20,"TRAC? LIST1":'     Selects X values of trace A with index 0
240 IEC IN 20,Xa0$:'              Stores X values as ASCII string under Xa0$
250 IEC OUT 20,"TRAC? TRAC1":'     Selects Y values of trace A with index 0
260 IEC IN 20,Ya0$:'              Stores Y values as ASCII string under Ya0$
270 IEC OUT 20,"TRAC? LIST2":'     Selects X values of trace B with index 0
280 IEC IN 20,Xb0$:'              Stores X values as ASCII string under Xb0$
290 IEC OUT 20,"TRAC? TRAC2":'     Selects Y values of trace B with index 0
300 IEC IN 20,Yb0$:'              Stores Y values as ASCII string under Yb0$
310 '
320 IEC OUT 20,"DISP:TRAC:IND 1":'          Selects trace pair with index 1
330 IEC OUT 20,"TRAC? LIST1":'     Selects X values of trace A with index 1
340 IEC IN 20,Xa1$:'              Stores X values as ASCII string under Xa0$
350 IEC OUT 20,"TRAC? TRAC1":'     Selects Y values of trace A with index 1
360 IEC IN 20,Ya1$:'              Stores Y values as ASCII string under Ya0$
370 IEC OUT 20,"TRAC? LIST2":'     Selects X values of trace B with index 1
380 IEC IN 20,Xb1$:'              Stores X values as ASCII string under Xb0$
390 IEC OUT 20,"TRAC? TRAC2":'     Selects Y values of trace B with index 1
400 IEC IN 20,Yb1$:'              Stores Y values as ASCII string under Yb0$
410 '
420 IEC OUT 20,"DISP:TRAC:IND 2":'          Selects trace pair with index 2
430 IEC OUT 20,"TRAC? LIST1":'     Selects X values of trace A with index 2
440 IEC IN 20,Xa2$:'              Stores X values as ASCII string under Xa0$
450 IEC OUT 20,"TRAC? TRAC1":'     Selects Y values of trace A with index 2
460 IEC IN 20,Ya2$:'              Stores Y values as ASCII string under Xa0$
470 IEC OUT 20,"TRAC? LIST2":'     Selects X values of trace B with index 2
480 IEC IN 20,Xb2$:'              Stores X values as ASCII string under Xa0$
490 IEC OUT 20,"TRAC? TRAC2":'     Selects Y values of trace B with index 2
500 IEC IN 20,Yb2$:'              Stores Y values as ASCII string under Xa2$
     :
```

3.15.12 Filter Settings

In the case of RMS measurements, the analyzer permits a maximum of 4 filters to be switched into the measurement path, 3 filters with PEAK and QPK measurements and one filter with THDN measurements. Fixed filters like CCITT, CCIR or WRUMble or user-defined filters can be used.
The example below illustrates a customized filters made up of a bandpass filter of 11 to 15 kHz and the two notch filters of 12 and 14 kHz.

```
     :
300 IEC OUT 20,"*RST;*WAI": ’             *WAI waits for end of calibration
310 IEC OUT 20,"DISP:MODE COLB":’                   Coloured user interface
315 ’----- User filter No. 1: Bandpass filter 11 to 15 kHz, Atten. 100 dB
320 IEC OUT 20,"SENS:UFIL1:BPAS ON"
330 IEC OUT 20,"SENS:UFIL1:PASS:LOW 11 KHZ"
340 IEC OUT 20,"SENS:UFIL1:PASS:UPP 15 KHZ"
350 IEC OUT 20,"SENS:UFIL1:ATT 100 DB"
355 ’----- User filter No. 2: 12-kHz notch filter
360 IEC OUT 20,"SENS:UFIL2:NOTC ON"
370 IEC OUT 20,"SENS:UFIL2:CENT 12 KHZ"
380 IEC OUT 20,"SENS:UFIL2:WIDT 500 HZ"
390 IEC OUT 20,"SENS:UFIL2:ATT 100 DB"
395 ’----- User filter No. 3: 14-kHz notch filter
400 IEC OUT 20,"SENS:UFIL3:NOTC ON"
410 IEC OUT 20,"SENS:UFIL3:CENT 14 KHZ"
420 IEC OUT 20,"SENS:UFIL3:WIDT 500 HZ"
430 IEC OUT 20,"SENS:UFIL3:ATT 100 DB"
435 ’The 3 customized filters defined above are used for RMS
436 ’measurements; the bandpass filter for increasing the filter slope
437 ’is used twice.
440 IEC OUT 20,"SENS:FUNC ’RMS’"
450 IEC OUT 20,"SENS:FILT1:UFIL1 ON":’            Two bandpass filters for
460 IEC OUT 20,"SENS:FILT2:UFIL1 ON":’          increasing the filter slope
470 IEC OUT 20,"SENS:FILT3:UFIL2 ON":’                  12-kHz notch filter
480 IEC OUT 20,"SENS:FILT4:UFIL3 ON":’                  14-kHz notch filter
     :
```

Filter curve obtained in a sweep from 9 to 17 kHz:

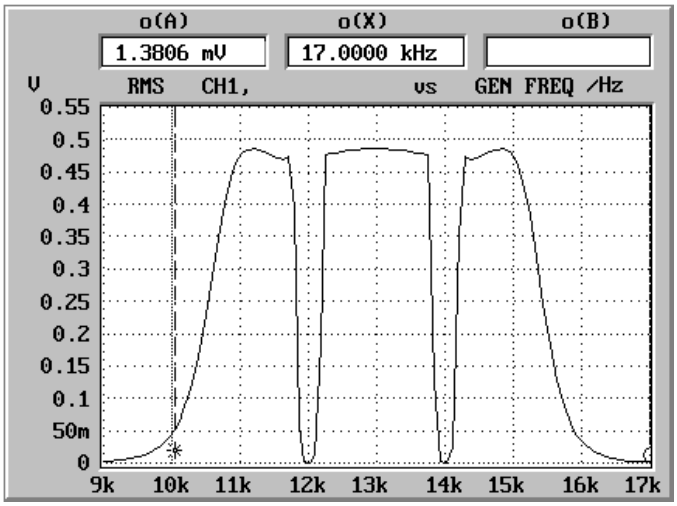


Fig. 3-11 Filter curve: steep bandpass filter + 2 notch filters

3.15.13 Finding a File

UPL provides no special command to find out whether a file has been stored on the UPL hard disk or on a floppy.
Remedy:
If an attempt to copy the file in a temporary file is not followed by an error message, the file already exists.

```
     :
100 File$ = "'C:\UPL\USER\MY.SCO':' File of interest
110 IECOUT 20,"MMEM:COPY "+File$+",'TMP.TMP'"
120 IECOUT 20,"SYST:ERROR?": IECIN 20,E$
130 IF LEFT$(E$,1)="0" THEN
140   PRINT "File available!"
150 ELSE
160   PRINT "File not available!"
170 ENDIF
     :
```

3.15.14 Readout of Error Queue

The error queue can be read out after each command or by means of an SRQ interrupt routine if an error has occurred (see section **3.7.4.5 Error Queue Query**)
The program below is a routine for reading out the error queue until it is empty.

```
     :
1290Errqueue:
1300 IEC OUT 20,"SYST:ERR?": '    Reads out error queue until queue is empty
1310 IEC IN 20,E$
1320 IF LEFT$(E$,1)="0" THEN RETURN:'                    Quits error routine
1330 PRINT "Contents of error queue: ";E$: GOTO Errqueue
     :
```

3.15.15 Command Synchronization

The synchronization modes realized in the example below are described in section 3.6.8, Command Synchronization.
Use commands *WAI, *OPC? or *OPC with SRQ to terminate a specific action before a new one is executed. Through suitable programming the controller can be made to wait for a specific action to be completed (see section **3.6.8.3 Comparison of Synchronization** Capabilities).
There are two events in the UPL which have to be waited for before the next command can be executed:

- End of calibration
- End of measurement

Selection of an instrument with automatic calibration is used as an example for demonstrating the three synchronization methods. The following command should only be sent when the automatic calibration is completed. For more detailed information refer to section **3.6.8.1 Wait for End** of Calibration.

3.15.15.1 Command Synchronization with *WAI

```
IECOUT 20,"INSTrument2 A100;*WAI":'       Selects new analog instrument and
                                        waits with *WAI for end of calibration
```

3.15.15.2 Command Synchronization with *OPC?

```
IECOUT 20,"INSTrument2 A100":'                Selects new analog instrument
IECOUT 20,"*OPC?":'              Sends OPC?. Calibration is terminated when
IECIN 20,A$:'                                 the response "1" is received.
```

3.15.15.3 Command Synchronization with *OPC and SRQ

Command synchronization with *OPC and SRQ is described in advance of section 3.15.16 Service Request, which should best be read through first. Waiting for end of calibration with *OPC and SRQ after an instrument selection is again used as an example.

Procedure:

- set Operation Complete bit (OPC) in the Event Status Register,
- set ESB bit 5 in the Status Byte Register
- activate SRQ handler,
- call up change of instrument with automatic calibration,
- output synchronization command *OPC,
- wait in a loop for SRQ (end of calibration).

```
1057 '***************** Setting up SRQ ********************************
1058 IEC TERM 10:'                                Line Feed as terminator
1059 IEC TIME 10000.'                           IEC/IEEE-bus timeout 10 s
1060 IEC OUT 20,"*CLS"
1061  'Enable OPC (Operation Complete) in the Event Status Register
1062  '                                 +---+---+---+---+---+---+---+---+
1063  '                                 | d7| Event Status Register | d0|
1064  '                                 |POW|USR|CME|EXE|DDE|QUE|   |OPC|
1065  IEC OUT 20,"*ESE 1": '            | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
1066  '                                 +---+---+---+---+---+---+---+---+
1067  'Trigger SRQ through entry in the Event Status Register (d5=1)
1068  '                                +---+---+---+---+---+---+---+---+
1069  '                                | d7|  Status Byte Register | d0|
1070  '                                |SOR|RQS|ESB|   |SQR|   |   |   |
1071  IEC OUT 20,"*SRE 96": '          | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
1072  '                                +---+---+---+---+---+---+---+---+
1073 ON SRQ1 GOSUB Srqintr:'   Activate SRQ handler of IEC/IEEE bus No. 1
  :
1080 IECOUT 20,"INSTrument2 A100":' Instrument change with autom. calibr.
1090 IECOUT 20,"*OPC":'                          Synchronization command
  :
1100 REPEAT
1110  'Other tasks may be performed as long as no SRQ is appears.
1120  'Signalled by a count on screen.
1130  Count=Count+1: PRINT Count
1140 UNTIL Srqflag=0:'           Flag = 1 when calibration is completed
  :
2000 '** Standard SRQ interrupt routine shown in section 4.15.1.2 ***
2010 Srqintr:
2020 IEC SPL 20,Sb%: '               Reads in Status Byte via Serial Poll
2040                                    :
```

3.15.16 Service Request

As can be seen from the introduction to the SRQ standard routine below, a Service Request routine requires an extended initialization of the UPL.
The SRQ standard routine uses Serial Poll for processing SRQ . This SRQ routine is used in almost all demo programs but for the sake of clarity it is not listed each time (remark in program code). The program example in section 3.15.16.2 SRQ Interrupt Routine with Parallel Poll, demonstrates SRQ handling by means of Parallel Poll which should be used to speed up identification of the instrument raising the SRQ when several IEC/IEEE-bus instruments are connected.
Same as with all other program examples it is assumed that IEC/IEEE-bus address 20 is set on the UPL to be controlled.

3.15.16.1 SRQ Interrupt Routine with Serial Poll

The examples below for initializing an SRQ and the SRQ interrupt routine are suggestions which can be modified as required by the specific application.
Serial Poll SRQ and the Serial Poll SRQ interrupt routine are initialized in this or a similar form in almost all program examples.

3.15.16.1.1 Initialization of Serial Poll SRQ

```
     :
100 '************ Initializing Serial Poll SRQ ****************
110 IEC TERM 10:' IEC/IEEE-bus terminator = Line Feed
120 IEC TIME 10000: ' IEC/IEEE-bus waiting time 10 s
130 IEC OUT 20,"*CLS":'Resets Status Register
140 'Enables error bits in the Event Status Register
150 '                                +---+---+---+---+---+---+---+---+
160 '                                | d7| Event Status Register | d0|
170 '                                |POW|USR|CME|EXE|DDE|QUE|   |OPC|
180 IEC OUT 20,"*ESE 61": '          | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
190 '                                +---+---+---+---+---+---+---+---+
200 'Enables d5 for SRQ trigger through Event Status Register
210 '                                +---+---+---+---+---+---+---+---+
220 '                                | d7|  Status Byte Register | d0|
230 '                                |SOR|RQS|ESB|   |SQR|   |   |   |
240 IEC OUT 20,"*SRE 96": '          | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
250 '                                +---+---+---+---+---+---+---+---+
260 Srqflag=0
270 ON SRQ1 GOSUB Srqintr: ' Activates SRQ handler
     :
```

3.15.16.1.2 Serial Poll SRQ Routine

The following standard SRQ interrupt routine is used in almost all the program examples. It displays the reason for the SRQ and the contents of the error queue on the controller monitor and signals to the main program with `Srqflag = 1`, that a SRQ has occurred.

```
1000Srqintr:
1010 ' ***********************************************
1020 ' ******* Standard SRQ Interrupt Routine ********
1030 ' ***********************************************
1040 IEC SPL 20,Sb%: ' Read-in of Status Byte
1050 IF (Sb% AND 64)=0 THEN GOTO Ret: ' No response in the case of false
alarm
1060 Srqflag=1
1070 PRINT "Status Byte Register = ";Sb%
1080 IF (Sb% AND 1)   THEN PRINT " SRQ->Not used"
1090 IF (Sb% AND 2)   THEN PRINT " SRQ->Not used"
1100 IF (Sb% AND 4)   THEN PRINT " SRQ->Not used"
1110 IF (Sb% AND 8)   THEN PRINT " SRQ->Questionable-status bit"
1120 IF (Sb% AND 16)  THEN PRINT " SRQ->Not used"
1130 IF (Sb% AND 32)  THEN PRINT " SRQ->Event-status bit"
1140 IF (Sb% AND 64)  THEN PRINT " SRQ->Summary bit"
1150 IF (Sb% AND 128) THEN PRINT " SRQ->Operation-status bit"
1160 '
1170 IEC OUT 20,"*ESR?": ' Read-in of Status Register
1180 IEC IN 20,Es$
1190 PRINT "Event Status Register = ";Es$
1200 IF (VAL(Es$) AND 1)   THEN PRINT " ESR->Operation-complete bit"
1210 IF (VAL(Es$) AND 2)   THEN PRINT " ESR->Not used"
1220 IF (VAL(Es$) AND 4)   THEN PRINT " ESR->Query-error bit"
1230 IF (VAL(Es$) AND 8)   THEN PRINT " ESR->Device-dep. error bit"
1240 IF (VAL(Es$) AND 16)  THEN PRINT " ESR->Execution-error bit"
1250 IF (VAL(Es$) AND 32)  THEN PRINT " ESR->Command-error bit"
1260 IF (VAL(Es$) AND 64)  THEN PRINT " ESR->User-request bit"
1270 IF (VAL(Es$) AND 128) THEN PRINT " ESR->Power-on bit"
1280 '
1290Errqueue:
1300 IEC OUT 20,"SYST:ERR?": ' Readout of error queue until queue is empty!
1310 IEC IN 20,E$
1320 IF LEFT$(E$,1)="0" THEN GOTO Ret
1330 PRINT "Contents of Error Queue:"
1340 PRINT " ";E$: GOTO Errqueue
1350 '
1360Ret: ON SRQ1 GOSUB Srqintr: RETURN:' Reactivates SRQ!
```

3.15.16.2 SRQ Interrupt Routine with Parallel Poll

3.15.16.2.1 Initialization of Parallel Poll SRQ

```
     :
100 '************ Initialization of Parallel Poll SRQ ********************
110 IEC TERM 10:'                        IEC/IEEE-bus terminator = Line Feed
120 IEC TIME 10000: '                         IEC/IEEE-bus waiting time 10 s
130 IEC OUT 20,"*CLS":'                                Resets Status Register
140 IEC OUT 20,"*ESE 121":' Enables OPC,DDE,EXE,CMD in the Event Status Reg.
150 IEC OUT 20,"*SRE 32":'             Enables Event Status bit as SRQ event
160 IEC OUT 20,"*PRE 255":'                  Enables all Parallel Poll lines
170 IEC PCON 20,1,6:'UPL identifies itself with 1 on line 6
180 IEC PCON 10,1,3:'Device with address 10 ident. itself with 1 on line 3
190 ON SRQ1 GOSUB Srqintr:'                              SRQ handler activated
     :
```

3.15.16.2.2 Parallel Poll SRQ Routine

```
740 '*********************************************************************
750 '******* Standard Parallel-Poll SRQ Interrupt Routine *************
760 '*********************************************************************
770Srqintr:
790 PRINT "SRQ has occurred!"
800 IEC PPL Pp%
810 IF (Pp% AND 32)<>0 THEN GOSUB UPLsrq
820 IF (Pp% AND 4)<>0 THEN GOSUB Adr10srq
825 ON SRQ1 GOSUB Srqintr: RETURN:' Reactivates SRQ
826 '
830UPLsrq:
840 '*********************************************************************
850 '**********         SRQ sent by UPL      ******************
860 '*********************************************************************
1040 IEC SPL 20,Sb%: ' Read-in of Status Byte
1060 Srqflag=1
1070 PRINT "Status Byte Register = ";Sb%
1080 IF (Sb% AND 1)   THEN PRINT " SRQ->Not used"
1090 IF (Sb% AND 2)   THEN PRINT " SRQ->Not used"
1100 IF (Sb% AND 4)   THEN PRINT " SRQ->Not used"
1110 IF (Sb% AND 8)   THEN PRINT " SRQ->Questionable status"
1120 IF (Sb% AND 16)  THEN PRINT " SRQ->Not used"
1130 IF (Sb% AND 32)  THEN PRINT " SRQ->Event Status"
1140 IF (Sb% AND 64)  THEN PRINT " SRQ->Summary"
1150 IF (Sb% AND 128) THEN PRINT " SRQ->Operation Status"
1160 '
```

```
1170 IEC OUT 20,"*ESR?": ’ Read-in of Event Status Register
1180 IEC IN 20,Es$
1190 PRINT "Event Status Register = ";Es$
1200 IF (VAL(Es$) AND 1)   THEN PRINT " ESR->Operation complete"
1210 IF (VAL(Es$) AND 2)   THEN PRINT " ESR->Not used"
1220 IF (VAL(Es$) AND 4)   THEN PRINT " ESR->Query error"
1230 IF (VAL(Es$) AND 8)   THEN PRINT " ESR->Device-dep. error"
1240 IF (VAL(Es$) AND 16)  THEN PRINT " ESR->Execution error"
1250 IF (VAL(Es$) AND 32)  THEN PRINT " ESR->Command error"
1260 IF (VAL(Es$) AND 64)  THEN PRINT " ESR->User request"
1270 IF (VAL(Es$) AND 128) THEN PRINT " ESR->Power on"
1280 ’
1290Errqueue:
1300 IEC OUT 20,"SYST:ERR?": ’ Read-out of error queue until it is empty
1310 IEC IN 20,E$
1320 IF LEFT$(E$,1)="0" THEN RETURN
1330 PRINT "Contents of error queue:"
1340 PRINT " ";E$: GOTO Errqueue
1250 RETURN
1260 ’
1270Adr10srq:
1280 ’*************************************************************
1290 ’*******  SRQ sent by device with the address 10   ************
1300 ’*************************************************************
1310 IEC SPL 10,Sb%: ’      Reset SRQ conditions for device with address 10
1320 ’SRQ evaluation for device with the address 10
1330 ’
1340 ’
1350 RETURN
```

3.15.17 Readout of Cursor Position and Values

The values of a curve displayed on the UPL can be read by the controller, no matter whether the curve has been generated by a sweep or FFT or loaded into the UPL from a file or the controller. To do so the o- or * cursor has to be positioned as required. The value at the cursor crossing the curve or the difference value can be read out.
The great number of commands available for positioning the cursor are listed below.
To simplify the program examples for the various cursor display modes and the respective intercept points, the cursor position is indicated directly in the form of a value.

Table 3-16 Positioning the cursor on the displayed curve

Positioning the cursor for curve display	
Positioning the o cursor:	**Positioning the *-cursor:**
"DISP:TRAC:CURS1:POS:MODE MIN1" Sets the horizontal o-cursor to the minimum value of **curve A** on the X axis.	**"DISP:TRAC:CURS2:POS:MODE MIN1"** Sets the horizontal *-cursor for modes "DISP:TRAC:CURS2:POS:MODE N12|D12|C12" to the minimum value of **curve A** on the **X axis .** Sets vertical *-cursor for modes "DISP:TRAC:CURS2:POS:MODE HL1|HL2|HLD1|HLD2" to minimum of **curve A** on the **Y axis.**
"DISP:TRAC:CURS1:POS:MODE MAX1" Sets horizontal o-cursor to maximum of **curve A.**	**"DISP:TRAC:CURS2:POS:MODE MAX1"** Sets horizontal *-cursor for modes ☞ "DISP:TRAC:CURS2:POS:MODE N12|D12|C12" to maximum of **curve A** on the **X axis.** Sets vertical *-cursor for modes "DISP:TRAC:CURS2:POS:MODE HL1|HL2|HLD1|HLD2" to maximum of **curve A** on the **Y axis**.
"DISP:TRAC:CURS1:POS:MODE MIN2" Sets horizontal o-cursor to minimum of **curve B.**	**"DISP:TRAC:CURS2:POS:MODE MIN2"** Sets horizontal *-cursor for modes "DISP:TRAC:CURS2:POS:MODE N12|D12|C12" to maximum of **curve B** on the **X axis.** Sets vertical *-cursor for modes "DISP:TRAC:CURS2:POS:MODE HL1|HL2|HLD1|HLD2" to minimum of **curve B** on the **Y axis.**
"DISP:TRAC:CURS1:POS:MODE MAX2" Sets horizontal o-cursor to maximum of **curve B.**	**"DISP:TRAC:CURS2:POS:MODE MAX2"** Sets horizontal *-cursor for modes "DISP:TRAC:CURS2:POS:MODE N12|D12|C12" to maximum of **curve B** on the **X axis.** Sets vertical *-cursor for modes ☞ "DISP:TRAC:CURS2:POS:MODE HL1|HL2|HLD1|HLD2" to maximum of **curve B** on the **Y axis.**
"DISP:TRAC:CURS1:POS:MODE VALue" **"DISP:TRAC:CURS1:POS 1000kHz"** Sets horizontal o-cursor for modes "DISP:TRAC:CURS1:POS:MODE N12|D12|C12" to specified position on the **X axis.**	**"DISP:TRAC:CURS2:POS:MODE VALue"** **"DISP:TRAC:CURS2:POS 1000kHz"** Sets horizontal *-cursor for modes "DISP:TRAC:CURS1:POS:MODE N12|D12|C12" to specified **X position.** Sets vertical *-cursor for modes ☞ "DISP:TRAC:CURS2:POS:MODE HL1|HL2|HLD1|HLD2 to specified position on the **Y axis.**

Table 3-17 Positioning the cursor for FFT spectrum display

Positioning the cursor for FFT spectrum display	
Positioning the o-cursor:	**Positioning the *-cursor:**
"DISP:TRAC:CURS1:POS:MODE MARKer1" Sets vertical o-cursor to **X position** of marker if the latter was switched on with "DISP:TRAC1|2:MARK:MODE MAX|CURS".	**"DISP:TRAC:CURS2:POS:MODE MARKer1"** Sets vertical *-cursor to **X position** of marker if the latter was switched on with "DISP:TRAC1|2:MARK:MODE MAX|CURS".
"DISP:TRAC:CURS1:POS:MODE NEXTmarker" Sets vertical o-cursor to **X position** of next harmonic if harmonics display was switched on with DISP:TRAC1|2:MARK:HARM ON.	**"DISP:TRAC:CURS2:POS:MODE NEXTmarker"** Sets vertical *-cursor to **X position** of next harmonic if harmonics display was switched on with "DISP:TRAC1|2:MARK:HARM ON".
"DISP:TRAC:CURS1:POS:MODE IMAX1" Sets vertical o-cursor to **X position** of highest Y value of **FFT curve A.**	**"DISP:TRAC:CURS2:POS:MODE IMAX1"** Sets vertical *-cursor to **X position** of highest Y value of **FFT curve A.**
"DISP:TRAC:CURS1:POS:MODE IMAX2" Sets vertical o-cursor to **X position of** highest Y value of **FFT curve B.**	**"DISP:TRAC:CURS2:POS:MODE IMAX2"** Sets vertical *-cursor to **X position** of highest Y value of **FFT curve B.**

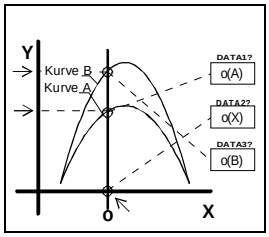
Note:

*Positioning of *-cursor partly depends on the set cursor mode "DISP:TRAC:CURS2:POS:MODE N12|D12|C12|HL1|HL2|HLD1|HLD2". Function and effect of the individual cursor modes can be seen from the following diagrams and the associated program line 110*

Abbreviations used in the diagrams below:

o(A) = Y value at crosspoint of vertical o-cursor on curve A
o(B) = Y value at crosspoint of vertical o-cursor on curve B
o(X) = X value of vertical o-cursor
*(A) = Y value at crosspoint of vertical *-cursor on curve A
*(B) = Y value at crosspoint of vertical *-cursor on curve B
*(X) = X value of vertical *-cursor
*(Y) = Y value of horizontal *-cursor
*(X)AL = X value at left crosspoint of horizontal *-cursor on curve A
*(X)AR = X value at right crosspoint of horizontal *-cursor on curve A
*(X)BL = X value at left crosspoint of horizontal *-cursor on curve B
*(X)BR = X value at right crosspoint of horizontal *-cursor on curve B

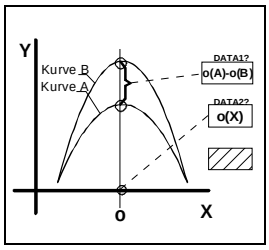
Cursor data for traces can be read out without restrictions as from UPL program version 2.10 onwards.!




```
100 IEC OUT 20,"DISP:TRAC:CURS1 ACT": ' Activates o-cursor 1
110 IEC OUT 20,"DISP:TRAC:CURS1:MODE N12"
120 IEC OUT 20,"DISP:TRAC:CURS1:POS:MODE VAL": 'at 1000 Hz
130 IEC OUT 20,"DISP:TRAC:CURS1:POS 1000 Hz"
```

Analogously, **CURS2** activates the ***-cursor** yielding the values *(A), *(X) and *(B)

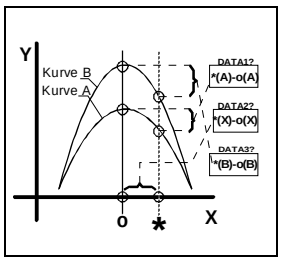
Fig. 3-12 Cursor data o(A), o(X), o(B), *(A), *(X), *(B)




```
100 IEC OUT 20,"DISP:TRAC:CURS1 ACT": ' Activates o-cursor 1
110 IEC OUT 20,"DISP:TRAC:CURS1:MODE D12"
120 IEC OUT 20,"DISP:TRAC:CURS1:POS:MODE VAL": 'at 1000 Hz
130 IEC OUT 20,"DISP:TRAC:CURS1:POS 1000 Hz"
```

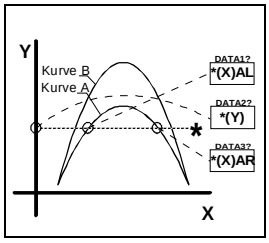
Analogously, **CURS2** activates the ***-cursor** yielding the values *(A) - *(B) and *(X)

Fig. 3-13 Cursor data o(A)-o(B), o(X), *(A)-*(B), *(X)




```
100 IEC OUT 20,"DISP:TRAC:CURS1 ACT;CURS2 ACT": 'Activates
                                                  o- and *-cursors
110 IEC OUT 20,"DISP:TRAC:CURS2:MODE C12"
120 IEC OUT 20,"DISP:TRAC:CURS1:POS:MODE VAL": 'o-cursor 1 kHz
130 IEC OUT 20,"DISP:TRAC:CURS1:POS 1000 Hz"
140 IEC OUT 20,"DISP:TRAC:CURS2:POS:MODE VAL": '*-cursor 2 kHz
150 IEC OUT 20,"DISP:TRAC:CURS2:POS 5000 Hz"
```

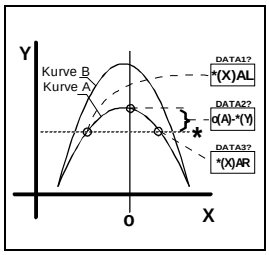
Fig. 3-14 Cursor data *(A)-o(A), *(X)-o(X), *(B)-o(B)




```
100 IEC OUT 20,"DISP:TRAC:CURS2 ACT": ' Activates *-cursor
110 IEC OUT 20,"DISP:TRAC:CURS2:MODE HL1"
120 IEC OUT 20,"DISP:TRAC:CURS2:POS:MODE VAL": 'Positions *-
cursor to
                                        Y value 0.2 V
130 IEC OUT 20,"DISP:TRAC:CURS2:POS 0.2 V"
```

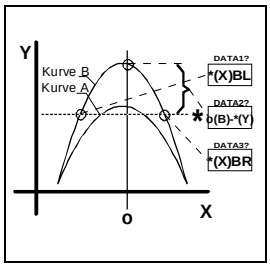
Analogously, cursor ...**CURS2:MODE HL2** yields values *(X)BL, *(Y) and *(X)BR for **curve B.**

Fig. 3-15 Cursor data *(X)AL, *(Y), *(X)AR




```
100 IEC OUT 20,"DISP:TRAC:CURS1 ACT;CURS2 ACT": 'Activates
                                                  o- and *-cursors
110 IEC OUT 20,"DISP:TRAC:CURS2:MODE HLD1"
120 IEC OUT 20,"DISP:TRAC:CURS1:POS:MODE VAL": 'Positions
                                          o-cursor on 1000 Hz
130 IEC OUT 20,"DISP:TRAC:CURS1:POS 1000.0 Hz"
140 IEC OUT 20,"DISP:TRAC:CURS2:POS:MODE VAL": 'Positions
                                       *-cursor on Y value 0.2 V.
150 IEC OUT 20,"DISP:TRAC:CURS2:POS 0.2 V"
```

Fig. 3-16 Cursor data *(X)AL, o(A)-*(Y), *(X)AR




```
100 IEC OUT 20,"DISP:TRAC:CURS1 ACT;CURS2 ACT": 'Activates
                                                  o- and *-cursors
110 IEC OUT 20,"DISP:TRAC:CURS2:MODE HLD2"
120 IEC OUT 20,"DISP:TRAC:CURS1:POS:MODE VAL": 'Positions
                                                 o-cursor on 1000 Hz
130 IEC OUT 20,"DISP:TRAC:CURS1:POS 1000.0 Hz"
140 IEC OUT 20,"DISP:TRAC:CURS2:POS:MODE VAL": 'Positions
                                               *-cursor on Y value 0.2 V
150 IEC OUT 20,"DISP:TRAC:CURS2:POS 0.2 V"
```

Fig. 3-17 Cursor data *(X)BL, o(B)-*(Y), *(X)BR

Readout of cursor values

DATA1?, **DATA2?** and **DATA3?** values are read in with the following commands:

```
IECOUT 20,"DISP:TRAC:CURS:DATA1?":IEC IN 20,"D1$
IECOUT 20,"DISP:TRAC:CURS:DATA2?":IEC IN 20,"D2$
IECOUT 20,"DISP:TRAC:CURS:DATA3?":IEC IN 20,"D3$
```

Values are output with the unit indicated in the display.

3.15.18 Call a BASIC-Macro

With the UPL, setting and measurement sequences can be written as BASIC programs or recorded using the built-in program generator (see 3.15.3 Command Logging - Converting B10 into IEC/IEEE-Bus Commands). Option UPL-K2 (Universal Autorun Control) is required. The generated BASIC programs can be stored (preferred file extension: .BAS) and called and used in various ways (see Macro operating).

The following example illustrates how a BASIC macro is called by means of an IEC/IEEE-bus control program in the programming language C and the IEC/IEEE-bus driver GPIB.COM from National Instruments:

Example 1:
BASIC macro transfers a measurement result in a measurement-result buffer to the control program

BASIC macro:

A short program is written under Universal Autorun Control UPL-B10 to trigger a level in channel 1. To demonstrate that any data can be transferred to the IEC/IEEE-bus control program as floating-point values via the measurement-result buffers, the level of channel 1 (line 30) is copied into the measurement-result buffer of channel 2 (line 40), from where it is read with the aid of the IEC/IEEE-bus control program.
This BASIC macro is stored in the UPL under the file name LEV_CH1.BAS.

```
10 UPL OUT "INIT:CONT OFF;*WAI"
20 UPL OUT "*TRG;*WAI"
30 UPL OUT "SENS:DATA?": UPL IN A$: ' level of channel 1 ...
40 UPL OUT "SENS:DATA2 "+A$: '... copying into buffer of channel 2
50 END
```

How to proceed:
- Press the F3 key to switch from the UPL user interface to Universal Autorun Control.
- Type the five lines shown above.
- Store program with SAVE LEV_CH1.BAS.
- Press the F3 key again to return to UPL user interface.

The following IEC/IEEE-bus control program calls the BASIC macro in the UPL with the command SYST:PROG:EXEC 'LEV_CH1.BAS'. There is a delay until serial polling indicates that bit 14 (RUN) has changed from 1 to 0 in the OPERation register; this indicates that the BASIC macro has been executed.
The measurement result is read from channel 2's measurement-result buffer and displayed on the screen.

IEC/IEEE-bus control program in controller:

```
/*************************************************************************
* A BASIC program in UPL triggering a level-measurement result in channel 1
* To be started as a BASIC macro from the controller
* Measurement result to be output at the controller
*************************************************************************/
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <conio.h>
#include <bios.h>
#include "C:\NI-GPIB\C\DECL.H"

void report_error(int fd, char *errmsg)
{

   fprintf(stderr, "Error %d: %s\n", iberr, errmsg);
   if (fd != -1) {
      printf("Cleanup: taking board off-line\n");
      ibonl(fd,0);
   }
getch();
exit(1);              /* abort program */
}

void befout (int upl, char *befstr)
{
   ibwrt(upl, befstr,(long)strlen(befstr));
   if (ibsta & ERR)
      report_error (upl, "Could not initialize UPL");
}

void queryin (int upl, char* reading)
{
 ibrd(upl, reading, 20L);
  if (ibsta & ERR)
   report_error (upl, "Could not read data from UPL");
  reading[ibcnt-1] = '\0'; /* Overwrites line feed with string terminator */
}

void main()
{
int    upl;           /* File descriptor for UPL */
int    i;
char   reading[20]; /* UPL measurement results */
long count = 0;
char stb;

if ((upl = ibdev(0, 20, 0, T10s, 1, 0)) < 0)
 report_error (upl, "Could not initialize UPL");

befout (upl,"*ESE 0"); //Disables information from event status register
befout (upl,"*SRE 0"); //Disables SRQ
```

```
/* The 1 -> 0 transition of bit 14 (RUN) in the OPERation register
   should set bit 7 (OPER) in the STB. The STB is read by means
   of serial poll until the event has occurred. */

befout (upl,"STAT:OPER:NTR 16384");  /* Enables 1->0 transition of bit 14 */
befout (upl,"STAT:OPER:PTR 0");       /* Disables 0->1 transition of bit 14
*/
befout (upl,"STAT:OPER:ENAB 16384"); /* Enables bit 14 for STB */

/* The BASIC program LEV_CH1.BAS in the UPL working directory writes
   the level measured in channel 1 to the measurement-result buffer of
channel 2
   to demonstrate data transfer via the measurement-result buffers. */

for (i=1; i <= 10; i++)
 {/* Reads 10 measurement results via the BASIC macro LEV_CH1.BAS */
  /* Reading the EVENt part of the OPERation register deletes the
     OPER bit in the status byte register! */
  befout (upl,"STAT:OPER:EVEN?");
  queryin (upl,reading);

  befout (upl,"SYST:PROG:EXEC 'LEV_CH1.BAS'"); // Starts the BASIC macro

  /* When the RUN bit (bit 14) in the OPERation register changes from 1 to
     0, the BASIC macro has been executed and the measurement
     result can be read from the measurement-result buffer. */
  stb = 0;
  while ((stb & 0x80) == 0)   // Serial poll is performed until
   {                          // bit 7 (OPER) in the STB is set to 1.
    ibrsp (upl,&stb);         // Serial poll of the status byte register
    if ((count++ % 100) == 0) // Progress counter while
     printf ("+");            // waiting for bit 7 = 1 */
   }

  /* The level of channel 1 can be read from the channel-2 measurement-
      result buffer, where it was stored by the BASIC macro. */
  befout (upl,"SENS:DATA2?");
  queryin (upl,reading);
  printf ("\n%s\n",reading);
 }

printf ("Any key:\n");
getch ();
ibonl(upl, 0);  /* Take UPL off-line */
}
```

Data transfer between BASIC macro and controller via the measurement-result buffers:

The measurement-result buffers can be written to so as to transfer data between the BASIC macro and the controller. The measurement results calculated by the macro can thus then be displayed in the UPL environment, which is familiar to the user.

Furthermore, fast exchange of floating-point parameters and floating-point measurement results between the UPL macro and the controller is possible via the measurement-result buffers. The following commands are available for the data exchange:
SENS1:DATA1, SENS1:DATA2
SENS2:DATA1, SENS2:DATA2
SENS3:DATA1, SENS3:DATA2

Note:
If the measurement results are not to be overwritten by the UPL measurement task, make sure the measurement task is halted, ie no measurement or sweep is being performed, while the measurement results are being written.

Example 2:
BASIC macro transfers a set of data in a block buffer to the control program

BASIC macro:

A short program is written under Universal Autorun Control UPL-B10. This program writes any set of data into the block buffer specially generated for BASIC macros. The data are then read from this buffer by the IEC/IEEE-bus control program.
This BASIC macro is stored in the UPL under BLK.BAS.

```
10 DIM A(1000): Frq=100: A(0)=X
20 FOR I=1 TO 999: ' 1000 log. frequency values  ...
30 Frq=Frq*1.00503: A(I)=Frq: ' ... 100 Hz to 15 kHz
40 NEXT I
50 UPL BLOCKOUT A(0),1000
60 UPL OUT "SYST:PROG"
70 END
```

Proceed as for example 1.

The following IEC/IEEE-bus control program calls the BASIC macro in the UPL with the command SYST:PROG:EXEC 'BLK.BAS'. There is a delay until serial polling indicates that bit 14 (RUN) in the OPERation register has changed from 1 to 0. This shows that the BASIC macro has been executed.

The block data are read from the block buffer and displayed on the screen.

IEC/IEEE-bus control program in the controller:

```
First part of program as in example 1
            :
            :
befout (upl,"SYST:PROG:EXEC 'BLK.BAS'"); // Starts the BASIC macro
stb = 0;
while ((stb & 0x80) == 0) // Serial poll is performed until
   {                      // bit 7 (OPER) in the STB is set to 1.
    ibrsp (upl,&stb);
    if ((count++ % 100) == 0) // Progress counter while
     printf ("_");            // waiting for bit 7 = 1  */
   }

  /* Determines number of values in the block buffer */
  befout (upl, "SYST:PROG:POIN?");
  queryin (upl,reading);
  points = atoi (reading);
  printf ("\nBlock buffer contains %d values. Display values...\n",points);
  getch();

 /* Read values from block buffer */
  befout (upl,"SYST:PROG?"); /* Fetch contents from block buffer.
                    The values are available as ASCII characters separated
                    by commas */
  ibeos (upl,0x142C); //Stringterminator = ','
  for (i = 0; i < points-1; i++)
```

```
   {// Each value is read up to the comma
    queryin (upl,reading);
    fltvalfield[i] = atof (reading);
   }
  // Before the last value has been read, the string terminator ...
  ibeos (upl,0x140A); //... is reset to AF.
  queryin (upl,reading);
  fltvalfield[i] = atof (reading);

  // Values are output on the screen.
  for (i = 0; i < points; i++)
    printf ("%d: %f\n", i+1,  fltvalfield[i]);

printf ("Any key:\n");
getch ();
ibonl(upl, 0);  /* Take UPL off-line */
}
```

3.15.19 Third analysis - Output of Block Data

Example of Programming for Universal Autorun Control UPL-B10:

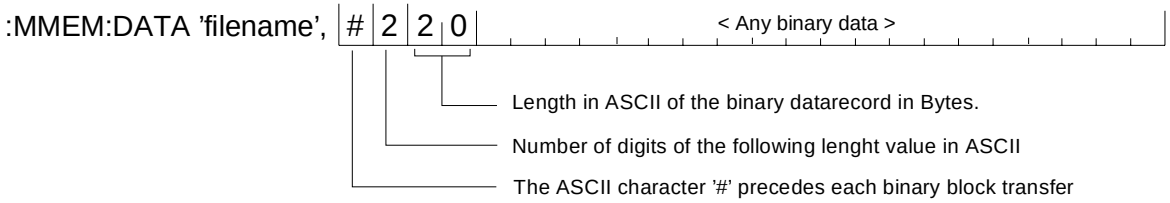
The following program example shows the settings for the third analysis measurement function of a noise signal (1/3 OCTAVE measurement function in ANALYZER panel).

Important!
The current level values of the 1/3 octave analysis are available under Scan Count 1 (line 200), the maximum level values of the max. hold function (line 60) under Scan Count 2 (line 210). The frequency list is queried with TRAC? LIST (line 140) , the max. hold level values with TRAC? TRAC (line 260).

```
10 REM ******** Third analysis - read out of Block Data ******************
20 UPL OUT "*RST": ’                                        UPL default setup
30 UPL OUT "SOUR:FUNC RAND": ’                    Noise signal for generator
40 UPL OUT "INP:TYPE GEN2": ’   Internal connection to generator channel 2
50 UPL OUT "SENS:FUNC ’THIR’": ’                  Switches on Third Analysis
60 UPL OUT "SENS:VOLT:INTV:MODE FOR": ’      Max-hold function switched on
70 UPL OUT "DISP:TRAC:OPER FFTL": ’               FFT data in tabular form
80 UPL OUT "INIT:CONT OFF": ’              Selects single-measurement mode
90 UPL OUT "*TRG;*WAI": ’                     Triggers single-measurement
100 UPL OUT "TRAC:POIN? LIST": ’         Request count of frequency values
110 UPL IN A$: Count=VAL(A$): ’         Read out count of frequency values
120 PRINT "Count of freq. values:";Count: INPUT "Go on......";A$
130 DIM X(Count): ’                                Field for frequency values
140 UPL OUT "TRAC? LIST": ’                         Read out frequency values
150 UPL BLOCKIN X(0): ’   Loads frequency block data into the data field X
160 FOR I=0 TO Count-1
170 PRINT X(I);"Hz ": ’                              Display frequency values
180 NEXT I
190 UPL OUT "DISP:TRAC:OPER SPEC": ’   COUNT selection only avail. in SPEC
200 REM UPL OUT "DISP:TRAC:COUN 1": ’ Choose current values of T. Analysis
210 UPL OUT "DISP:TRAC:COUN 2": ’ Choose Max-hold values of Third Analysis
220 UPL OUT "TRAC:POIN? TRAC": ’              Request count of level values
230 UPL IN A$: Count=VAL(A$): ’             Read out count of level values
240 PRINT : PRINT "Count of level values: ";Count: INPUT "Go on......";A$
250 DIM Y(Count): ’                                 Field for level values
260 UPL OUT "TRAC? TRAC": ’                  Read out Max-hold level values
270 UPL BLOCKIN Y(0): ’       Loads level block data into the data field Y
280 FOR I=0 TO Count-1
290 PRINT Y(I);"V ": ’                               Display level values
300 NEXT I
310 END
```

3.15.20 Binary Data via IEC/IEEE-Bus Interface

Using the IEC bus command **MMEM**ory**:DATA 'filename', #<lele><le><Binärdaten>**, it is possible to transfer any binary data and files from the host to the UPL.



Using the IEC bus command **MMEM**ory**:CHEC**k**? '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.

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	(kopiert von einem Steuerrechner aus eine Datei über RS232-Schnittstelle zum UPL)
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`

3.16 Automatic Control of UPL with R&S BASIC

Important:

The software described below is an accessory for UPL and available under the designation UPL-B10 . It is not part of the equipment supplied with UPL.

The program examples in the paths C:\UPL\B10_EXAM\EXAM1.BAS ff of the UPL software can be directly run on the UPL with Universal Sequence Controller UPL-B10. Files with the extension .SAC are setup files required by the 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.

3.16.1 Use

Executing frequent test sequences in a fast and reproducible way, summing up the results and creating a valuable documentation, these are the applications of the UPL universal sequence controller using R&S BASIC. These automatic measurements, consisting of generator and analyzer functions of the UPL, are used for a full characterization of instruments and components in production or the test shop and for ensuring and monitoring the characteristics of system and transmission devices.

A universal sequence controller for automatic measurements does not only have to control the instrument functions, but must also be able to evaluate the measurement results and branch in the program. Besides, operator prompting with confirmations and indications is expected. Furthermore, synchronization with a time base or external events may be required. Thus, some programming is sometimes required, but it should be as simple as possible. Therefore, a complete BASIC interpreter with optimally integrated commands is used for operation of the measuring instrument. A simple keystroke permits to change between normal operation of the measuring instrument and BASIC. The command extensions for the instrument control feature the same structure as the IEC/IEEE-bus commands, which in turn comply with the international SCPI standard.

3.16.2 Scope of Functions

The UPL provides about 600 elements (ie functions in the programming language) and almost as many keywords as parameters. Therefore, users not wishing to do any programming, but also experts will appreciate the integrated program generator. Every input via front panel or keyboard for setting the UPL is recorded in logging mode and added to the program as a complete program line. Simple test sequences are thus completely programmed without having typed a single line. There is no need to check the correct syntax, the created program can be easily read due to the standard SCPI notation and is thus simple to modify and supplement.

R&S BASIC with easy-to-handle IEC/IEEE-bus commands that are optimally incorporated into the syntax can also control further IEC/IEEE-bus devices without the need for an external controller. (This requires the Remote Control option UPL-B4). Likewise, it is straightforward to operate the serial interface and write and read files for connection with peripheral devices or other programs.

For graphical output, BASIC can fully make use of the UPL software: Graphs with sophisticated scaling and labelling, bargraphs, bargraphs with trailing pointers, all of them also with automatic scaling depending on the measured value, are still available. In addition, the graphics commands belonging to BASIC can also be used.

If the UPL is to be controlled by an external controller in a test system, two REPLACE commands can be used to convert all UPL IN/OUT instructions into IEC/IEEE commands (IEC IN/OUT). This constitutes the basic program for controlling the UPL.

3.16.3 Preparation for Use

If the Universal Sequence Controller UPL-B10 is ordered together with the UPL it is immediately ready for use. When retrofitted, it has to be enabled by means of an installation key. A key matching the serial number of the UPL is supplied to the user to activate the software.

Subsequently, a memory model for BASIC can be selected by calling UPLSET. The user has to indicate how much memory he wants to reserve for the BASIC program and BASIC data (variables). Since the UPL cannot simultaneously be remote-controlled via the IEC/IEEE bus and the Universal Sequence Controller UPL-B10, the user must choose between the two modes by means of UPLSET. The files CONFIG.SYS and UPL.BAT are thus changed.

-UPLSET contains a menu through which the user is guided:

- Enter "UPLSET" and "↵" after the prompt c:>.
  The selected operating mode is maintained even after instrument switch-off.

To estimate the required memory, the following empirical values are given: A typical BASIC line requires about 25 bytes. A 13k program memory is thus sufficient for about 500 lines or 10 pages of program. The remaining empty memory can be polled in BASIC with FRE(1). A variable in BASIC requires about 15 bytes (depending on the length of the name), and a field with floating-point numbers requires 8 bytes for each index. FRE(o) indicates the remaining storage area.

The memory should not be oversized, since the program may be limited in its speed from a certain size onwards (see also Section 3.16.4.11 UPL/Basic Memory Management - UPL-B10). If the preset values are not exceeded, the UPL operates at full speed.

In the case of first installation, the UPL must be booted anew; otherwise this is only necessary when the memory model is changed. UPLSET can also be called in order to poll about the currently active memory allocation. If the UPL is to operate again as a measuring instrument controlled externally via IEC/IEEE bus, UPLSET can be used to return to this mode.

3.16.4 Operation

In the following, a distinction is made between BASIC and the UPL program, the latter including all routines except BASIC (i.e. the test, readout, graphics output and input routines).

3.16.4.1 Switchover between UPL and BASIC Entry Mode

- **Switchover from UPL to BASIC entry mode**:
  Use function key **F3** on the **external keyboard** or **BACKSP** on the front panel. With BASIC in the entry mode, the fields for reading out measured values are displayed at the upper edge irrespective of whether full graphics display is selected in the UPL or not. The field below down to the softkeys is available to BASIC.
  Possible error messages after pressing the F3 key: When the message "BASIC not installed" is displayed, BASIC has not been installed at all or incorrectly. "Memory not available" indicates that a memory size exceeding the available space has been selected with UPLSET.
- **Entry in the BASIC mode:**
  All characters can be entered from the external keyboard as required. A limited control is also possible from the UPL front panel:
  Keys of the front-panel key blocks DATA/PANEL and EDIT as well as the CURSOR keys have the common functions.
  Exceptions:
  SELECT = blank
  +/- = - (minus)
  The keys of the CONTROL block as well as HELP, the tabulator and PgUp/Dn keys in the CURSOR/VARIATION block have no function.

  Letters cannot be output with the front-panel keys but the entry of numerals opens up plenty of possibilities for controlling a BASIC program without the burden of an external keyboard.

  While BASIC waits for a line entry (as after pressing of the "↵"-key), the UPL program continuous to run in the background and the measurement results are displayed. The effects of the settings made by BASIC can thus be observed immediately. However, after the first character has been entered, the UPL program is not called up any more. No measurements are performed and the printer spooler (HCOPY) does not run in the background until the entry is terminated with "↵".

  Note:

  When the UPL is controlled with BASIC, printing with HCOPY is considerably slowed down as the available computing time must be divided up. There will be more time for the printout if the measurement is stopped using the STOP key on the front panel.

- **Switching back from BASIC to UPL entry mode:**
  Key **F3** on the **external keyboard** or the keys **ENTER** and then **LOCAL** on the **front panel**.
  When switching back to UPL control the panels are completely restored to show the current settings as there may have been changes under BASIC. BASIC is inactive but the last-set status will be retained.

3.16.4.2 First Steps (Readout of Measurement Results)

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

- Press key F3 of the external keyboard from the UPL user interface. The result display field appears at the top of the screen, the softkey lines at the bottom. "R&S-BASIC version..." is displayed in the screen center.
- Enter the following program (including line numbers 10-90).

```
10 UPL OUT "*RST": ’                                  UPL default setting
20 UPL OUT "INP:TYPE GEN2": ’Internal connection to generator channel 2
30 FOR I=1 TO 10
40  UPL OUT "INIT:CONT OFF;*WAI": ’       Triggers a single measurement
50  UPL OUT "SENS:DATA?":  Requests function test result from channel 1
60  UPL IN M$: ’                            Reads in measurement result
70 PRINT M$: ’                               Prints measurement result
80 NEXT
90 END
```

- Start the program with F6:

Due to the default setting made with *RST (see Appendix **A UPL Default Setup**), the UPL generator produces a 1-kHz sinewave signal with a level of 0.5 V.
Command INP:TYPE GEN2 internally links generator channel 2 and analyzer channel 1 so that cabling between inputs and outputs is not required for this first test. The UPL analyzer performs 10 RMS measurements and the results are displayed on the UPL screen.

Note:

To display a measurement result on the screen it must first be triggered (line 40). After triggering a settled result is available which can be requested (line 50), read out (line 60) and displayed on the UPL screen (line 70).

- Return to manual operation: Press F3 on the external keyboard or ENTER and then LOCAL on the front panel.

3.16.4.3 Logging Mode

Function key F2 switches the logging mode on or off. The respective mode is indicated in the bottom righthand corner above the softkeys. In the case of "on", all entries used for setting the UPL are appended to the BASIC program as a command line. After switching to the BASIC mode, these new lines are displayed automatically and may be modified.

The BASIC commands for automatic control of the UPL differ only slightly from the commands for remote control via the IEC/IEEE bus. The program can easily be converted into the other commands using the BASIC command REPLACE (eg for controlling the UPL with an external controller). See also UPL-specific modifications of the BASIC manual, paragraph REPLACE. A detailed example is given in section 3.15.3 Command Logging - Converting B10 into IEC/IEEE-Bus Commands.

3.16.4.4 Differences to IEC/IEEE-Bus Remote Control

The differences between the R&S BASIC commands of the universal sequence controller and the R&S BASIC commands of the IEC/IEEE-bus control are illustrated by way of examples:

Delimiters of character strings, Timeout

R&S BASIC command	R&S IEC/IEEE-bus command
A delimiter need not be specified for the transfer of a character string.	10 IEC TERM 10 Controller expects LF as delimiter for an UPL response
Waits indefinitely for a response. However, in special cases the time can be monitored by querying bit d0 (OPC) of the Event Status Register in a loop until the bit assumes the value 1 which signals that a measurement result is available. The program example is given under "Other differences to IEC/IEEE-bus remote control" further down in this section.	10 IEC TIME 5000 The controller does not wait longer than 5 s for a response from the UPL before an IEC/IEEE-bus timeout is signalled.

Output of commands

R&S BASIC command	R&S IEC/IEEE bus command
10 **UPL OUT** "SOUR:FREQ 1000Hz" (sets generator frequency) Transfers a character string (constant in quotes, variable denoted with $ or a character string expression) to the UPL program.	10 IEC OUT 20, " SOUR:FREQ 1000Hz "

Reading in responses

R&S BASIC command	R&S IEC/IEEE-bus command
100 UPL OUT "SENS:DATA2?" 110 **UPL IN** A$ (transfers the measured value of channel 2 to the variable A$ for processing) Takes a character string from the UPL program. This may be a measured value or a queried setting. The information to be taken must first be defined in a query (command with question mark).	100 IEC OUT 20, "SENS:DATA2?" 110 IEC IN 20, A$

Output of block data

R&S BASIC command	R&S IEC/IEEE-bus command
10 DIM A(20) 20 Frqval = 20 30 FOR I = 0 TO 19 40 A(I) = Frqval 50 Frqval = Frqval*1.44 60 NEXT I 70 **UPL BLOCKOUT** A(0),20 80 UPL OUT "sour:list:freq" (transfer of 20 values for a frequency list sweep to the UPL) **UPL BLOCKOUT <array(i)>[,n]** Stores block data in a reserved communication range of the UPL so that they can be transferred subsequently from BASIC to the UPL with an UPL OUT "...." command. This applies, for instance, to the transfer of lists or values for graphic display. The index **i** indicates the value from which onwards the data field is to be transferred. **n** defines the number of values. If **n** is not specified, the quantity defined with DIM will be used.	10 DIM A(20) 20 ' Combines block data to a string 30 Bef$="SOUR:LIST:FREQ" 40 Frqval=20 50 FOR I=0 TO 19 60 Bef$=Bef$+STR$(INT(Frqval)) 70 IF I<19 THEN Bef$=Bef$+"," 80 Frqval=Frqval*1.44 90 NEXT I 100 IEC OUT 20,Bef$

Reading out block data

R&S BASIC command	R&S IEC/IEEE-bus command
10 DIM A(200) 20 UPL OUT "TRAC? TRAC" 30 **UPL BLOCKIN** A(0) 40 UPL OUT "TRAC:POIN? TRAC" 50 UPLIN A$:Count = VAL(A$) (loads a sweep list in the form of block data from the UPL into field A( ) of BASIC and loads the data into Count) **UPL BLOCKIN <array(i)>** loads block data (ie lists or a measurement sequence) from the UPL program into a data field (index variable) for further processing. Same as with UPL IN the data to be read must first be defined in a query (sour:list:freq?). **i** denotes the index from which the first value of a block is stored. The list is always used in full length, ie a sufficiently large data field has first to be defined with DIM.	10 DIM A (200) 20 IEC OUT 20,"TRAC:POIN? TRAC" 30 IEC IN 20,Count$: Count=VAL(Count$) 40 IEC OUT 20,"TRAC? TRAC" 50 IEC TERM 44: ' Sets string term. to ',' (0x2C = 44d) 60 FOR I=0 TO Count-2: ' Reads in n - 1 values 70 IEC IN 20,A$: DIM(I) = VAL(A$) 80 NEXT I 90 'Reads last value 100 IEC TERM 10: 'Resets string term. to LF 110 IEC IN 20,A$: DIM(Count-1) = VAL(A$)

Switchover to UPL user interface

R&S BASIC command	R&S IEC/IEEE-bus command
UPL GTL (Go To Local) With this command BASIC is quit and the UPL screen is displayed. Control is also transferred to the UPL and entries can be made in the UPL panels. To return to BASIC the F3 key has to be pressed. This means that in the program mode this command has the same function as the F3 key (switchover from BASIC to UPL). If the operator now presses F3, control is switched back and BASIC continues with the command that follows UPL GTL in the program. When entered in the BASIC direct mode, this command has the same function as the F3 key. ***Note:*** *After a BASIC program has been started with RUN, key F3 is disabled. It is enabled again only after END, STOP or abort (with Ctrl/Break)in the BASIC entry mode.* **UPL GTL U** (Go To Local, temporarily with UPL screen) This command is required when the UPL display should be used under BASIC. Thus an ongoing sweep or FFT can be monitored (provided a measurement was started before, the display is continuously updated) or values computed in BASIC are to be displayed. (These values have to be transferred first from BASIC to UPL using the UPL BLOCKOUT command). The panel displayed at the left of the graphics window can be used by BASIC for PRINT outputs. A line must not be longer than 26 characters, however, as otherwise the graphics window will be overwritten. After the UPL GTL U command, the screen scroll is stopped to avoid the graphic display being shifted and destroyed when the first or the last line is reached (in the case of uncontrolled PRINT outputs). Status lines 1 and 2 cannot be used either as this space is occupied by the UPL display. With this command the BASIC display is stored and the UPL screen displayed. Control is not transferred to the UPL, however, and entries cannot be made in the UPL panels. Immediately after display of the UPL screen, control is handed back to BASIC without the BASIC display being restored. Use the GTL B command described below to restore the BASIC display. After an UPL GTL U command and before terminating the program with END, STOP or abort (with Ctrl/Break or in case of a fault) the operator should restore the BASIC display as otherwise no entries can be made, eg in the UPL full-screen mode. As an aid for the operator, the BASIC display is automatically restored when the BASIC entry mode is reached (entry of commands or program instructions) with the consequence that UPL GTL B is performed immediately when UPL GTL U command is entered in the direct mode and the entry has no effect. **UPL OUT "DISP:ACT ON | OFF"** This command is of interest in conjunction with GTL U. The OFF state prevents the graphics display being updated after every control command as this might be disturbing and slow down program execution. With ON the display is completely restored and then continually updated. The OFF and ON commands are to be used together as a pair as otherwise the UPL graphics display is not restored, not even when the UPL is manually controlled.	`IEC LAD 20` `IEC GTL`

UPL GTL B (set BASIC screen) This command is only needed with UPL GTL U. It restores the BASIC screen displayed before the UPL display set up with UPL GTL, UPL GTL U or with the F3 key (change from BASIC to UPL control). **UPL GTL G (draw UPL graphic)** This command is needed when the UPL graphics display is to be used under BASIC. The display is restored but not subsequently updated as is the case with UPL GTL U. Thus values computed in BASIC, for instance, can be displayed (the values have to be transferred first from BASIC to the UPL using the UPL BLOCKOUT command). The operator can make full use of the UPL graphics display with scales and labels. The graphics display must be switched on of course (with command UPL OUT "disp:conf ...." using parameters xP or P, or previously in manual control). Note: Upon loading a setup, the 3-panel display may be selected which has the effect of switching off the graphics display. In case of doubt check the setting by changing to the UPL mode. When the result output has been activated (set to display) using UPLOUT "disp:ann on" also the measurement results in the display field at the screen top are updated. Contrary to F3, UPL GTL or UPL GTL U, the BASIC screen is not stored and the UPL display set up, but the UPL graphics display is integrated in the BASIC screen. Like any other graphic drawn under BASIC, the display is shifted together with the text when the cursor reaches the top or bottom edge. The graphic may also be overwritten by text (in this case the background is blanked) irrespective of whether this is useful or not. When the display is shifted, which can always be prevented by adequately positioning the cursor (see section 3.16.4.8 STRINX.SYS Driver for Screen and Keyboard - UPL-B10), parts of the graphics display remain visible at the upper or lower edge of the scroll window as the scrolled parts are shifted in multiples of the text size, which do not correspond to the UPL graphic displayed. Since the UPL graphic has become part of the BASIC screen it is always restored when the UPL display is switched over to BASIC with the F3 key. It is cleared together with the BASIC text screen using the sequence PRINT "Esc[2J". CLEAR also clears the screen or parts thereof (see section 3.16.4.5 UPL-specific Modifications to the Basic Manual - UPL-B10 under CLEAR).	

Further differences to IEC/IEEE-bus remote control:

- An SRQ procedure cannot be programmed in R&S BASIC of the Universal Sequence Controller UPL-B10, ie the control program cannot perform other tasks while waiting for measurement results or error messages from the UPL. Error queue or the status registers must be queried cyclically in the control program.
- Since the UPL does not generate an SRQ for the Universal Sequence Controller UPL-B10, there is no need to determine the sender or source of the SRQ by means of a serial or parallel poll. This should not be mixed up with the possibility of using the UPL as an IEC/IEEE-bus controller and to control other instruments on the IEC/IEEE bus from the Universal Sequence Controller UPL-B10 (described in the manual R&S BASIC Interpreter supplied with the UPL-B10 option).
- Line messages like REN, GET, DCL, etc are not possible.
- The line message EOI denoting the end of a binary block cannot be used.
  Binary block data cannot be received. The command " FORMat REAL" is ineffective.
  Block data can only be received with command UPL BLOCKIN described above.

- In R&S BASIC of the Universal Sequence Controller UPL-B10 all common commands referring to SRQ control are not relevant. Respective information can be obtained from the table under 3.9 Common Commands.
  An exception is the *OPC command. Although mainly intended to trigger an SRQ when a measurement result is relevant, under R&S BASIC this command may be used to wait for a measurement result in a loop by querying the bit d0 of the Event Status Register:

  Example:
```
110 UPL OUT "*OPC;*TRG": ’ Triggers measurement result
120 Brk=0: I=0
130 WHILE (Brk=0) AND (I<=100)
140  UPL OUT "*ESR?"
150  I=I+1: UPL IN Esr$: IF (VAL(Esr$) AND 1)<>0 THEN Brk=1: ’ Queries OPC
160 WEND
170 IF I>100 THEN PRINT "Timeout": STOP
180 UPL OUT "SENS:DATA?": ’ Requests measured value
190 UPL IN M$: PRINT M$: ’ Reads and output measured value
```

- Common commands (3.13.1) and addressed commands (3.13.2) are not available for the Universal Sequence Controller UPL-B10.

- All registers and commands of the status reporting system which are not related to SRQ generation can be used:

*STB?	not usable
*SRE	not usable
*PRE	not usable
*IST?	not usable
***ESR?**	**usable**
*ESE	not usable
STATus:**OPER**ation**?**	**usable**
STATus:**QUES**tionable**?**	**usable**
STATus:**XQUE**stionable**?**	**usable**
SYSTem:**ERR**or**?**	**usable**

- ***Note:***

  *If UPL with Universal Sequence Controller UPL-B10 is used as a controller for **other IEC/IEEE-bus devices** (described in the R&S BASIC Interpreter manual supplied with the UPL-B10 option), and should **again** be controlled by another controller as a talker/listener **on the IEC/IEEE bus**, IEC/IEEE-bus control must be released by the Universal Sequence Controller UPL-B10 with command IECRLC (IEC ReLease Control) .*

3.16.4.5 UPL-Specific Modifications to the BASIC Manual

There are only a few modifications to the supplied standard BASIC manual which result from different conditions. In the part of the manual dealing with process controllers, the BASIC manual of the PSA and PAT controller versions is valid.

Softkey labelling and function keys
As against the standard BASIC manual, the function keys are shifted by 4 keys as F1 to F4 are assigned different functions in the UPL. The softkey labelling has been adapted accordingly for the UPL. Switchover between alphanumeric and graphics mode (F8) is not provided in the UPL.

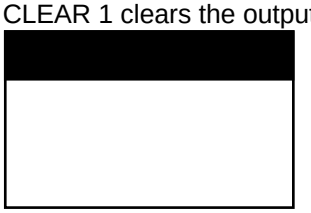
BYE
is a synonym for EXIT; description see under EXIT.

CLEAR [1 | 2 | 3]
This command clears the screen or parts of it. The size of the parts corresponds to the UPL panels or fields. The commands always clear the indicated parts irrespective of whether they are assigned UPL fields or not.

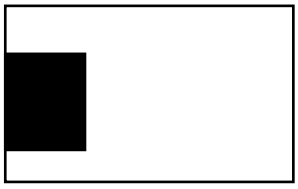
Without parameter specified, the upper part of the screen is cleared, however without the output field for measured values.



CLEAR 1 clears the output field for measured values



CLEAR 2 clears the field at the left of the graphic display.



CLEAR 3 clears the UPL graphics.



COLOR
should not be modified to avoid changing of the UPL graphics output. The colours are assigned as follows:

Pen	UPL colour mode	UPL b/w mode	
0	white	white	(background)
1	dark grey	white	
2	white	black	
3	red	black	
4	grey	grey	
5	yellow	light grey	
6	dark grey	dark grey	
7	yellow	dark grey	
8	green	grey	
9	green	black	
10	blue	black	
11	green	black	
12	yellow	grey	
13	cyan	dark grey	
14	black	black	
15	black	black	(preselected colour)

COPYOUT
is not supported. See GSAVE "LPT1".

EXIT (Synonym for BYE)
leaves the BASIC mode and returns to the UPL input mode and not to MS-DOS.

GRAPHIC
The interface name for putting out graphics on the screen is no longer GRAPH but GRAPX with the UPL.

GSAVE on LPT
is not supported. Instead, the HCOP:DEST <> remote-control command should be used for a printout of the display.

HELP
is not supported as a command.

HOLD
***Note**:*

During the wait time the routines are not continued. Therefore, with long times, the wait time should better be implemented with a loop using TIME.

REPLACE
To allow also the comma to be contained in the REPLACE command as part of the string (and not as separator between the new and the old string), it has to be preceded by a backslash (\,).

Example 1:
old program: `100 UPL OUT A$`
`REPLACE UPL OUT, IEC OUT 20,`
new program: `100 IEC OUT 20,A$`

Example 2:
old program: `100 IEC IN 20, A$`
`REPLACE IEC IN 20\,, UPL IN`
new program: `100 UPL IN A$`

SCREEN
is not supported; SCREEN 18 (VGA mode with 16 colours/grey shades) is always set.

SET
The colour of the pen is selected from the colours described above under COLOR.

SHELL
is only supported with restrictions, since the remaining memory of approx. 60 Kbytes is too small; the MS-DOS command interpreter together with the program called must not exceed this memory size. However, this is the case with the internal and a few external MS-DOS commands (dir, del, md, cd etc, see MS-DOS manual).

VIEWPORT
The upper limit for y2 should be 294 so that the upper field remains vacant for readout of the measured values. In principal, there are no restrictions to the BASIC graphics commands, it is up to the user whether the area used by the UPL graphics is overwritten.

WINDOW
The preselected values are 0,639,0,**293.**

ZOOM
is not supported

3.16.4.6 BASIC Screen

The screen contains 30 text lines, 5 at the lower edge being reserved for softkeys and two lines for status indication. One status line is used by BASIC, the other is available to the user (see labels of status lines and softkeys).

Two modes are provided for the upper 25 lines: either BASIC uses all lines or the UPL builds up a field for the output of measured values at the top edge, where the measurement results are continuously updated. This field comprises 7 text lines and reduces the BASIC text window to 18 lines. The second mode is activated using UPL OUT "disp:ann on".

Inside the 25/18-line window, the text is scrolled when the cursor reaches the top or bottom edge. If this window contains a graphic, the latter is shifted as well. This also applies to graphics drawn by the UPL software. However, since the graphics are slightly larger than the text window, which is variable in steps of 16 pixels only (text size), they seem to be somewhat "torn". By positioning the text appropriately, the user must ensure not to write outside the text window (which causes a shift).

The command PRINT "Esc (2)" clears the 18- or 25-line window depending on the mode. The CLEAR command (without parameter) only clears the 18-line window (without the area reserved for the output of measured values). The BASIC extensions CLEAR 1½ 2 ½ 3 clear the panel fields or the output field for measured values irrespective of whether they contain UPL panels or not (see section 3.16.4.5 UPL-specific Modifications to the Basic Manual - UPL-B10).

If text is entered for BASIC, BASIC does not get the keyboard entries but reads out the screen contents. If graphics are superimposed on this text, it is possible that the character is not identified and BASIC responds with an error message. Even the space between the last character and the right-hand screen edge is significant if it can be interpreted as a blank or character. Therefore, the user should write at a "clean" position, or go to a position by scrolling the text before writing.

3.16.4.7 Control Commands Unsuitable for Logging

The control commands are described in detail in the UPL manual, classified according to their function and then in alphabetical order. Furthermore, almost all commands can be logged in manual operation, the commands being created in the correct notation and added to the program.

Commands for reading out measured values and control commands via the front-panel keys in the CONTROL block cannot be logged. They are briefly summed up in the following.

CONTROL key commands:

START	UPL OUT "init:cont on"
	UPL OUT "init"
SNGL	UPL OUT "init:cont off"
STOP	UPL OUT "abort"
HCOPY	UPL OUT "hcop"
LCD ON/OFF	UPL OUT "disp:enab on" adv. off
OUTPUT ON/OFF	-----
LOCAL	UPL GTL

Commands for reading out single measurement results:

Function CH1:	UPL OUT "sens:data?":UPL IN A¤
Function CH2:	UPL OUT "sens:data2?":UPL IN A¤
Input PEAK CH1:	UPL OUT "sens2:data?":UPL IN A¤
Input PEAK CH2:	UPL OUT "sens2:data2?":UPL IN A¤
Freq CH1:	UPL OUT "sens3:data?":UPL IN A¤
Freq CH2:	UPL OUT "sens3:data2?":UPL IN A¤
Phase:	UPL OUT "sens4:data2?":UPL IN A¤

Commands for reading out block data:

Trace A:	UPL OUT "trac? trac1":UPL BLOCKIN A(0)
Trace B:	UPL OUT "trac? trac2":UPL BLOCKIN B(0)
X values:	UPL OUT "trac? list1":UPL BLOCKIN X(0)
Z values:	UPL OUT "trac? list2":UPL BLOCKIN Z(0)

Command for synchronization:

UPL OUT "*WAI"

3.16.4.8 Driver for Screen and Keyboard STRINX.SYS

An internationally standardized software interface based on the ANSI standard X 3.41-1774 is provided for addressing the screen. This standard defines all functions required to operate a terminal. The most important functions of this standard as well as certain functions mainly required for operating the BASIC editor are implemented in the STRINX device driver.

This driver is loaded when booting the operating system, if
`DEVICE = C:\UPL\DRIVER\STRINX.SYS`
is contained in the configuration file CONFIG.SYS.

The following function groups are supported:

- Cursor control
- Labelling of status lines and softkeys
- Clearing screen areas
- Editing screen
- Setting the video attributes.

If the appropriate ANSI sequence can be assigned parameters P1; P2 ...; Pn, they must be entered as decimal numbers with one or two digits. The individual parameters are separated by semicolons.

The STRINX.SYS driver is to be used exclusively for programming in R&S BASIC. With this driver, the UPL is largely compatible with the PSA. ANSI.SYS has to be loaded for programs using the control sequences of the ANSI standard.

Cursor Control
The cursor can be set to absolute or relative positions. The output of new characters is continued starting at the set position.

Table 3.18 Cursor Control (UPL-B10)

Cursor function	**Sequence**	**Example in BASIC**
Free positioning (one parameter may be omitted)	ESC[P_1; P_2H *)	?"E_c[05;32H"; Column Row
Shift cursor by P_n positions ↑ Shift cursor by P_n positions ↓ Shift cursor by P_n positions → Shift cursor by P_n positions ← (P_n with one digit only)	ESC[P_n A ESC[P_n B ESC[P_n C ESC[P_n D	?"E_c[5A"; ?"E_c[3B"; ?"E_c[3C"; ?"E_c[5D";
Shift cursor by one position ↑ ↓ with scrolling	ESC[: A ESC[: B	?"E_c[:A"; ?"E_c[:B";
Store cursor position / recall cursor position	ESC[s ESC[u	?"E_c[s"; ?"E_c[u";
Switch off cursor Switch on cursor	ESC[h ESC[l	?"E_c[h"; ?"E_c[l";

*) ESC means the key "Esc" or the keyboard code 1BH or 27 decimal

Labelling of Status Lines and Softkeys

Table 3.19 Labelling of Status Lines and Softkeys (UPL-B10)

Function	Sequence	Example in BASIC
Label status lines	ESCQP$_n$TEXT	?"E$_c$Q3TEXT"
Label softkeys	ESCRP$_n$TEXT	?"E$_c$R4TEXT"

Note:

The sequences must be terminated by LF (ASCII code 10).

Softkeys 1 to 4 cannot be labelled as they have been assigned important basic functions that must be available in all program states.

Status and softkey lines are labelled as follows:

Table 3.20 Labelling of Softkeys (UPL-B10)

Screen lines	**VGA graphics mode *)**
First Last	25 Q1 26 Q2, Q 27 Q3 28 Q4, softk. 29 Q5

*) Compatible with PSA mode

If one of the softkeys is labelled, the PSA compatibility mode is switched on (return with sequence "ESC[1j"). The keyboard codes are converted to the corresponding values of the PSA according to the following table:

Table 3.21 Labelling of Softkeys, Conversion table (UPL-B10)

PSA code	PC comp. scan code	Keyboard labels	
		German keyboard	American keyboard
0E0H	3B	F1	F1
0E1H	3C	F2	F2
0E2H	3D	F3	F3
0E3H	3E	F4	F5
0E4H	3F	F5	F4
0E5H	40	F6	F6
0E6H	41	F7	F7
0E7H	42	F8	F8
0E8H	43	F9	F9
0E9H	44	F10	F10
0EAH	45	F11	F11
0EBH	46	F12	F12
0B7H	47	Pos 1 7	Home
0B8H	48	↑ 8	↑
0B9H	49	Bild ↑ 9	Pg Up
0B4H	4B	← 4	←
0B6H	4D	→ 6	→
0B1H	4F	Ende 1	End
0B2H	50	↓ 2	↓
0B3H	51	Bild ↓ 3	Pg Dn
0B0H	52	Einfg 0	Ins
0AEH	53	Entf .	Del

Editing the screen

Table 3.22 Editing the screen (UPL-B10)

Action	Sequence	Example
Insert blank lines	ESC[P_nL	?"E_c[5L"
Delete lines	ESC[P_nM	?"E_c[3M"

Clearing Screen Area

Table 3.23 Clearing Screen Area (UPL-B10)

Cursor function	Sequence	Example in R&S BASIC
Clear screen, Cursor ↖	ESC[2J	?"E_c[2J"
Clear complete video RAM, Cursor ↖	ESC[3J	?"E_c[3J"
Clear from cursor to end of screen	ESC[J	?"E_c[J"
Clear from cursor to end of line	ESC[K	?"E_c[K"
Clear status and softkey lines **Note:** The sequence must be terminated by LF.	ESC[y	?"E_c[y"

Setting the colours

Colours may be assigned to each character output on the screen. The colours are also set via the ANSI interface.

Table 3.24 Setting the colours (UPL-B10)

Colour	b/w	Sequence	Example in R&S BASIC
light grey	light grey	ESC[91m	?"E_c[91m"
light grey	light grey	ESC[92m	?"E_c[92m"
yellow	white	ESC[93m	?"E_c[93m"
----	----	ESC[94m	?"E_c[94m"
blue	black	ESC[95m	?"E_c[95m"
dark grey	dark grey	ESC[96m	?"E_c[96m"
black	black	ESC[97m	?"E_c[97m"
black	black	ESC[98m	?"E_c[98m"

3.16.4.9 Operation of Serial Interfaces COM1 and COM2

The transmission rate, parity bit, number of data and stop bits as well as the error handling mode of the serial interfaces can be configured using the MS-DOS MODE program, which is described in section **2.17 Connecting External Devices** or in the DOS manual (DOS commands). Configuration is made either automatically on calling the AUTOEXEC.BAT program, in the operating system by means of an entry via the keyboard, or in BASIC using the SHELL command. Example:

```
10 SHELL "mode com2: baud=24 parity=e data=7 stop=1 retry=n >NUL"
```

The last instruction for rerouting to the NUL device ensures that MODE does not output the response on the screen, but suppresses it.

The serial interface designated COM1 or COM2 is prepared for the output using the PRINT# command. As with every PRINT command, CR and LF are added to the string to be output if the line is not terminated by a comma or semicolon. Example:

```
10 OPENO #1,"com1:"
20 PRINT #1,A$
```

If the acknowledge lines DSR and CTS are not active, the operating system outputs the error message "ERROR 74 IN LINE xx: DOS: write fault".

The interfaces are prepared for reception using the OPEN command. With the subsequent INPUT# command, characters are read in until a CR is received. If the sending device continues to transmit characters, these characters must immediately be read in using the next INPUT# command, otherwise they are lost and an error is signalled. Timing becomes more critical the higher the transmission rates. If no characters are received (timeout), a zero is stored every 100 ms.

Example:

```
10 OPENI #1,"com1:"
20 INPUT #1,A$
```

The entry described above is for line-oriented text as characters are accepted until reception of CR. However, if the number of characters is known, the INPUT$() function ensures that any character and almost any number of characters can be received.

Example:

```
10 OPENI #1, "com2:"
20 A$=INPUT$( 100, #1)
```

Precisely 100 characters are read in. If less characters are received, the device waits 100 ms (specified timeout) for every expected character (and enters zero).

To wait for the start of the transmission, a character can be entered in a loop until the character is no longer zero.

Example:

```
20 REPEAT
30 A$=INPUT( 1, #1)
40 UNTIL ASC( A$) > 0
40 INPUT #1,B$
50 B$=A$+B$
```

3.16.4.10 UPL-Specific Error Messages From BASIC

ERROR 83: 'Instrument command allowed in this context'

The command is not allowed in this instrument state and depends on other settings. ('Execution error' with IEC/IEEE-bus control)

ERROR 84: 'Instrument param not within valid range'

The value of the command parameter is illegal.

ERROR 85: 'Instrument unit not allowed in this context'

The indicated unit is not allowed (in this state).

ERROR 87: 'Instrument option not installed'

The option required for this command is not fitted.

ERROR 88: 'Instrument ??? user error'

A user-correctable error has occurred during command execution (eg file not found).

ERROR 89: 'Instrument ??? system error'

An error has occurred in the MS-DOS or UPL software during command execution.

ERROR 90: 'Instrument invalid header string'

The actual command was not recognized (possibly because of a notation error).

ERROR 91: 'Instrument invalid parameter string'

The parameter of the command (string) is invalid.

ERROR 92: 'Instrument invalid unit string'

The unit of the command (string) is invalid.

3.16.4.11 UPL/BASIC Memory Management

The user need not be familiar with all the details, as the user automatically receives the CONFIG.SYS to be used and associated batch files with the program UPLSET described in section 3.16.3.

Since the memory available to MS-DOS programs is limited, the overlay technique is used in the UPL program. Furthermore, BASIC requires memory for the user program and its data (variables). These memory areas are assigned the parameters

-bp<n> for the program memory and
-bd<n> for the data memory.

Example:

```
upl_ui -bp16 -bd8
```

reserves 16k main memory for the program and 8k for the data, BASIC itself needing about 3k for its own management.

The minimum values are about 8k program and 4k data. BASIC can manage a maximum of 64k. However, with a size of about 2 times 32 k, the overlay memory for the UPL program decreases, reducing the program speed. More details cannot be given, since the available total memory, which may be occupied by resident programs and device drivers, may be very different.

The memory management of the UPL program is to be briefly explained in the following so that an experienced user can optimize his own configuration. The program and data memory used for BASIC is first reserved in the UMB area. If this is not possible (because the line DOS=HIGH,UMB is missing in the CONFIG.SYS, or too may other programs have been loaded into this area by LOADHIGH or DEVICEHIGH), the space in the conventional memory (below 640k) is used. If the remaining memory for the UPL program thus becomes too small, no memory is reserved for BASIC at all. The attempt to switch to BASIC then produces the error message "not enough memory for BASIC".

In memory models 64k plus 32k (or 32k plus 64k) also the device drivers are loaded into the UMB area. With 64k plus 64k the upper memory is completely full and the device drivers have to be stored in the conventional memory.

If space is to be used in the conventional memory, the size of the overlay memory is reduced. The UPL program may be restricted in its speed from a certain size onwards, which also depends on the memory required by other resident programs.

When executing BASIC, other instruments can also be controlled via the IEC/IEEE-bus interface. In this case, the UPL is the system controller via BASIC, i.e. it can no longer be remote-controlled by an external controller. The parser program UPL_IEC.EXE is no longer needed and must not be loaded any more when starting the UPL. Instead, BASIC requires the following device drivers:

STRINX.SYS	as BASIC editor
IECX.SYS	as IEC/IEEE-bus controller
GRAPHX.SYS	for the BASIC commands for graphics output
BEEPX.SYS	for audio outputs.

3.17 Remote Control via RS-232 Interface

As from UPL version 1.0 onwards, when an option UPL-B4 is purchased, remote control is not only possible via the IEC/IEEE bus but also via the RS-232 interface at the COM2 port at the rear of the instrument.

3.17.1 Preparation for Use

To activate the COM2 interface of the UPL for remote control select

```
Remote via      COM2
```

in the options panel.

To allow communication between controller and UPL via the RS-232 interface, the parameters of the two COM2 interfaces must be matched. The UPL parameters can be set with

```
COM2   PARAMETER   -------------
Baud Rate   2400    ...    56000
Parity      EVEN|ODD|NONE
Data Bits   7|8
Stop Bits   1|2
Handshake   XON/XOFF|RTS/CTS
```

in the options panel.
The meaning of the parameters can be seen in section 2.15.1 IEC/IEEE-bus Address

Use a zero-modem cable with the following assignment for interconnecting controller and UPL (2 x 9-pin female connector, R&S Order No. 1050.0346). This cable is suitable for RTS/CTS and XON/XOFF handshake.

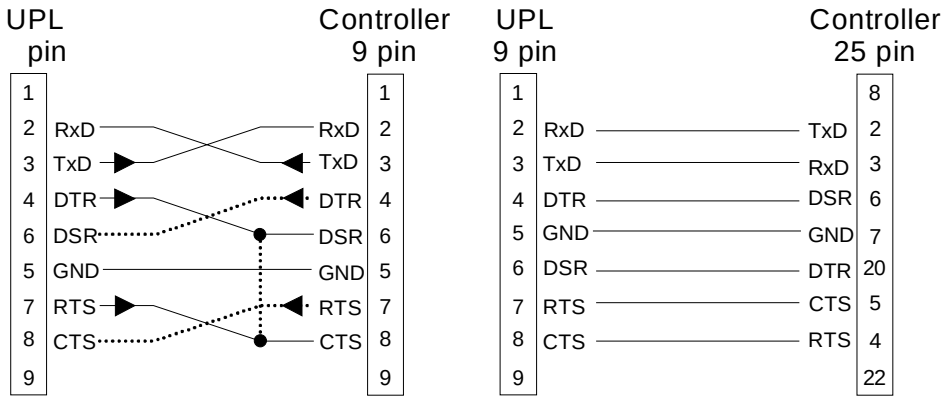


Fig. 3-18 Universal RS-232 cable, suitable for RTS/CTS and XON/XOFF handshake

RTS (request to send) is an output of the UPL which is set to TRUE (+12V) upon UPL switch-on. With **handshake = RTS/CTS** selected, UPL sets RTS to FALSE (-12V) when the UPL buffer is full during a data transmission from the controller to the UPL. In this case the controller must stop data transmission immediately until the contents of the data buffer have been processed by the UPL and RTS has been reset to TRUE. Normally, the RTS line of the UPL is connected to the CTS line of the controller. With **handshake = XON/XOFF**, RTS is not served by the UPL and remains set to TRUE.

DTR (data terminal ready) is an output of the UPL which is set to TRUE (+12V) upon UPL switch-on and which does not change. If the RS-232 interface of the controller requires the DSR (data nor ready) input to be set to TRUE, it is best to connect DTR of the UPL to DSR of the controller. With **Handshake = XON/XOFF** selected, RTS of UPL is always TRUE. Thus DSR and CTS of the controller could be linked by means of a jumper on the PC connector (see dotted line).

CTS (clear to send) of the UPL is an input. If the UPL wishes to send data to the controller or plotter with **handshake = RTS/CTS** selected, it first checks whether the addressed device is ready to receive data. State-of-the-art devices normally signal their readiness to receive via the RTS output (older devices often via the DTR output). If the CTS input of the UPL is TRUE (+12V), UPL starts data transmission. If the data buffer of the addressed device is full, the RTS output is reset and the CTS input of the UPL is set to FALSE (-12V). As soon as the data buffer of the addressed device is empty, the device resets the RTS output and thus the CTS input of the UPL to TRUE and UPL continues data transmission. With **handshake = XON/XOFF** selected, the CTS input of the UPL is irrelevant and the connection CTS - ·- · RTS (dash-point line) is not required.

DSR (data set ready) of the UPL is an input which is not evaluated however. The line DSR - ··- ·· DTR (dash-dot-dot line) is not required.

3.17.2 Switchover to Remote Control

After power-up, the UPL is always in the manual operating mode (LOCAL state)") and can be controlled from the front panel or a keyboard. Control is switched to REMOTE as soon as a signal is identified at the RS-232 interface via COM2 in the remote control mode. During remote control the front-panel keys are disabled. The UPL remains in the REMOTE control mode.

3.17.3 Return to Manual Operation

UPL remains in the REMOTE control mode until switched to manual mode by means of the LOCAL key on the front panel or command SYSTem:GTL. A change from manual to remote control and vice versa does not change the instrument settings.

3.17.4 First Steps (Readout of Measurement Results)

A simple test of the RS-232 interface can first be performed with DOS commands from the PC.
Example:

- Connect the COM1 interface of the PC and the COM2 interface of the UPL using the zero-modem cable. If the COM1 interface of the PC is assigned to a mouse, the mouse may be connected to the COM2 interface of the PC using the adapter cable (9 → 25-way) normally supplied with the mouse.

- Set UPL in the OPTIONS panel to remote control using the RS-232 interface. The standard parameters of the COM2 interface, 9600 baud, even parity, 7 data bits, 1 stop bit, are set in the UPL. This standard setting is made when the BACKSPACE key is pressed during UPL switch-on (loading DEFAULT setup), *UPL -d* is entered at the DOS level and after (new) installation of an UPL software. On switching the UPL off and on or upon loading a setup, the "Remote via" settings and the parameters of the serial COM2 interface remain unchanged.

```
Remote via      COM2
          :
          :
PARAMETER    ------------
Baud Rate       9600
Parity          EVEN
Data Bits       7
Stop Bits       1
Handshake       RTS/CTS
```

- Adapt the COM1 interface of the PC to the COM2 interface of the UPL with DOS command *mode*:
  mode com1
  :9600,e,7,1
- Create an ASCII file with the name COMOUT.TXT at the PC using the DOS editor, which comprises the characters "*RST" (set UPL to default setting).
- Send the contents of file COMOUT.TXT to the COM1 interface:
- *copy COMOUT.TXT com1:*

UPL goes to the REMOTE mode and to the default setting.

- Although the described method allows control of the UPL, data cannot be read out as the copy command of DOS requires received characters to be terminated with the delimiter CTRL Z (1A hex) and UPL terminates its responses with a Line Feed (0A hex).

The following program examples demonstrate UPL control and read-out of measurement results via the RS-232 interface. For the three examples the RS-232 interface parameter should be set as described in section 3.17.4.

Because of the default setting with "*RST" (see Appendix **A UPL Default Setup**), the UPL generator produces a 1-kHz sinewave signal with a level of 0.5 V.
Command "INP:TYPE GEN2" internally links generator channel 2 and analyzer channel 1 so that no cabling of outputs and inputs is required. The UPL analyzer performs 10 RMS measurements and the results are displayed in the result field and on the controller screen.

Note:

*A measurement result must be triggered before it can be displayed on the screen ("INIT:CONT OFF;*WAI"). After triggering, a settled result is available which can be queried, ("SENS1:DATA1?"), read in (comin) and displayed on the screen of the controller. (PRINT...).*

3.17.4.1 Readout of Measurement Results in QuickBASIC

Enter the following program listing in Microsoft QuickBASIC (most of the PCs will use **QuickBASIC** under MS-DOS on the PC) and start with Shift-F5:

In QuickBASIC a maximum transfer rate of 19200 baud can be set (`OPEN "COM1:`**`19200`**`,E,7,1,RB100" FOR RANDOM AS #1).`
Since QuickBASIC does not support the hardware handshake RTS/CTS the size of the communication buffer (**RB value** in byte) had to be adapted to the expected data quantity and the controller speed: the slower the controller the larger the data quantity and the higher the RB value.

Every output string must be terminated with an LF (line feed) so that UPL is able identify the end of the command. In this program example the line feed is added generally to the output string in the `Comout` output routine (`PRINT #1, A$;` **`CHR$(10)`**).

```
'************************************************
'* Triggering and output of 10 measurement results *
'************************************************
DECLARE FUNCTION Comin$ ()
DECLARE SUB Comout (A$)

OPEN "COM1:9600,E,7,1,RB100" FOR RANDOM AS #1: '    Parameter of COM1
'The size of the communication buffer   (RB value in bytes) has to be
'adapted to the expected data quantity: The slower the controller the larger
the data quantity and the higher the RB value.
'For a single measurement result a few bytes will be sufficient,
'for a block data transfer of 1024 Y values of an FFT in ASCII
'format with a total length of more than 12000 bytes approx. RB5000
is required.
Comout ("*RST;*WAI"): '                          Loads default setup
Comout ("INP:TYPE GEN2"):'Connects analyzer input to generator output
FOR I = 1 TO 10
  Comout ("INIT:CONT OFF;*WAI"): '      Triggers and waits for result
  Comout ("SENS1:DATA1?"): '  Queries measurement result of channel 1
  PRINT Comin$: '            Reads out and outputs measurement result
NEXT I
CLOSE
END

FUNCTION Comin$
'*** Reading the response string of UPL at COM1 ***
' UPL terminates each output string with NL (CHR$(10)).
' Characters are read from the COM1 interface until NL is received.
X$ = ""
NZ: Z$ = INPUT$(1, 1): '           Reads single characters from COM1
IF Z$ <> CHR$(10) THEN X$ = X$ + Z$: GOTO NZ: '       Next character
Comin$ = X$: '                                Returns complete string
END FUNCTION

SUB Comout (A$) STATIC
'*** Output of ASCII string at the COM1 interface ***
 PRINT #1, A$; CHR$(10): 'Each string must be terminated with NL (CHR$(10)).
END SUB
```

3.17.4.2 Readout of Measurement Results in R&S BASIC

Enter the following program listing in R&S BASIC and start with F2 (RUN).

When R&S BASIC is already installed on the controller, the COMX.SYS device driver allows the maximum UPL transmission rate of 56000 baud (`OPENI# 1,"com1:56000,e,7,1,2000"`) to be used.

Every output string must be terminated with LF so that the UPL is able to identify the end of a command (eg `"INP:TYPE GEN2"+CHR$(10);`).

```
10 '************************************************************************
30 '* Triggering and output of 10 measurement results                      *
40 '************************************************************************
50 OPENO# 2,"com1:"
60 OPENI# 1,"com1:9600,e,7,1,2000": '                    Set COM1 parameter
70 PRINT# 2,"*RST"+CHR$(10);: '  Sets default setup with internal connection
80 PRINT# 2,"INP:TYPE GEN2"+CHR$(10);: '      between generator and analyzer
90 FOR I=1 TO 10: '    Outputs 10 individually triggered measurement results
100 PRINT# 2,"INIT;*WAI"+CHR$(10);: '            Triggers measurement result
110 PRINT# 2,"SENS:DATA?"+CHR$(10);: '            Selects measurement result
120 GOSUB Comin: '                               Reads out measurement result
```

```
130 PRINT Instr$: ’                                  Outputs measurement result
140 NEXT I
150 END
160 ’
170 ’************** Reading in the UPL response *****************
180 Comin:
190 C$="": Instr$=""
200 Nexchar:
210 C$=INPUT$(1,#1): ’                             Reads individual characters
220 IF C$=CHR$(10) THEN RETURN :  ’Terminates reading when an LF is received
230 Instr$=Instr$+C$: GOTO Nexchar: ’Links individual characters to a string
```

3.17.4.3 Readout of Measurement Results in Borland-C 3.0

Enter the following program listing in Borland-C 3.0 and start with CTRL-F9 (RUN).

In Borland-C 3.0 a maximum transmission rate of 9600 baud can be set.

Each output string must be terminated with a line feed so that UPL can identify the command end (eg "INP:TYPE GEN2**n**").

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <bios.h>
#include <dos.h>
#include <conio.h>

/************************************************************************/
/*                              Declarations                            */
/************************************************************************/

#define COM_1 0
#define COM_2 1
#define SETTINGS (_COM_9600 | _COM_CHR7 | _COM_STOP1 | _COM_EVENPARITY)

/*The following settings can be made in bios.h: */
// _COM_CHR7       0x02    /* 7 data bits */
// _COM_CHR8       0x03    /* 8 data bits */
// _COM_STOP1      0x00    /* 1 stop bit */
// _COM_STOP2      0x04    /* 2 stop bits */
// _COM_NOPARITY   0x00    /* no parity */
// _COM_EVENPARITY 0x18    /* even parity */
// _COM_ODDPARITY  0x08    /* odd parity */
// _COM_110        0x00    /* 110 baud */
// _COM_150        0x20    /* 150 baud */
// _COM_300        0x40    /* 300 baud */
// _COM_600        0x60    /* 600 baud */
// _COM_1200       0x80    /* 1200 baud */
// _COM_2400       0xa0    /* 2400 baud */
// _COM_4800       0xc0    /* 4800 baud */
// _COM_9600       0xe0    /* 9600 baud */
```

```
/*************************************************************************/
/* Initialization of interface */
/*************************************************************************/
void init_com(int port)
{
 _bios_serialcom(_COM_INIT, port, SETTINGS);}

/*************************************************************************/
/* Output of character */
/*************************************************************************/
void outp_char(int port,char c)
{
 _bios_serialcom(_COM_SEND, port, c);
}

/*************************************************************************/
/* Reading a character */
/*************************************************************************/
int inp_char(int port,char *to_rec)
{
unsigned int status;

 while (1)
 { /* Read until a character of error-free status is received */
  status = (_bios_serialcom(_COM_RECEIVE, port, 0) & 0x9fff);
  if ((status & 0x9f00) == 0)
   {
    *to_rec = (char)status;
    return (0);
   }
 }
}

/*************************************************************************/
/* Output of string */
/*************************************************************************/
void comout(int port, char *strptr)
{
 while (*strptr != '\0')
   outp_char(port,*strptr++);
}

/*************************************************************************/
/* Read string until a line feed is received */
/*************************************************************************/
void comin (int port, char *recptr)
{
int idx = 0;
char c = 0;

  while (1)
   {
    inp_char (port,&c);
    recptr[idx] = c;
    if (c == '\n') // Abort when LF is received
     break;
    idx++;
   }
  recptr[idx] = '\0'; // Overwrite NL with '\0'
}
```

```
/***************** Main program ******************/
/* Triggering and output of 10 measurement results */
/*************************************************/
int main ()
{
char recstring[100];
int i;

init_com(COM_1);

comout (COM_1, "*RST;*WAI\n");
comout (COM_1, "INP:TYPE GEN2\n");

for (i = 1; i <= 10; i++)
 {
  comout (COM_1, "INIT;*WAI\n");
  comout (COM_1, "SENS:DATA?\n");
  comin (COM_1, recstring);
  printf ("%s\n",recstring);
 }
printf ("Continue: ");
getch ();
return (1);
}
```

3.17.5 Binary Data via RS232 Interface

First steps

To obtain the programs RS232_BT.EXE and RS232_BT.BAS, which are required for a transmission of binary data via RS232 interface, proceed as described under 3.15.20 Binary Data via IEC/IEEE-Bus Interface.

Transmitting a File to UPL via RS232 Interface

RS232_BT.BAS is written in the programming language QuickBASIC from Microsoft, which are under MS-DOS on most PCs.
The source file RS232_BT.BAS runs under QuickBASIC and can be adapted to the user's needs whenever necessary.

For transmitting binary data via RS232_BT.EXE, the following settings must be made in the OPTIONS panel of UPL:

```
Remote via      COM2
COM2 PARAMETER -------
Baud Rate       19200   (max. permissible baudrate for QuickBASIC)
Parity          NONE    (required for binary transmission)
Data Bits       8       (required for binary transmission)
Stop Bits       1
Handshake       RTS/CTS
```

When transmitting binary data via the RS232 interface, the XON/XOFF handshake must **not** be set in the OPTIONS panel of UPL, since XON/XOFF functions by means of exchanging binary characters, which may affect characters in the binary data stream.

As an RS232 cable connecting UPL and the control PC, a modem-bypass cable with R&S order number 1050.0346 should be used, or a cable with a pin assignment as described in 3.17.1 Preparation for Use, Fig. 3-42.

After RS232_BT has been called, the program requests the following:

- Selection of the desired interface COM1 or COM2 at the controller as well as the desired baud rate (make sure the baud rate selected in UPL is used). The program now attempts to contact the UPL. If this is successfully done, UPL measurements are halted to attain maximum transmission speed.
- Path and file name of the file to be transmitted are polled as well as the name of the file to be generated with this content in the UPL working directory.
- The program now generates a temporary file with the name TEMP.OUT, which contains the RS232 command required for UPL and the data set to be transmitted.
- TEMP.OUT is now transmitted to UPL. As the transmission takes considerably more time than via IEC/IEEE bus, the program has a progress bar. The file TEMP.OUT is retained in the controller also after the transmission is completed so that it can be viewed by means of an editor with binary capability.
- Following the transmission, signatures are generated (see MD5 signature technique) using the original file on the controller and the file stored in UPL. If these two files have the same signatures, it may be assumed with great certainty that the file are absolutely identical and no transmission errors have occurred.

MD5 Signature Method

To check whether a file was transferred error-free from the process controller to the UPL, the MD5 signature method can be used, see 3.15.20 Binary Data via IEC/IEEE-Bus Interface.
UPMD5.EXE can conveniently be called up as "child process" (SHELL) from its RS232 control program in order to compate the signature with the signature of the file transmitted to the UPL.

Example in QuickBASIC see source code RS232_BT.BAS lines 53 and 54

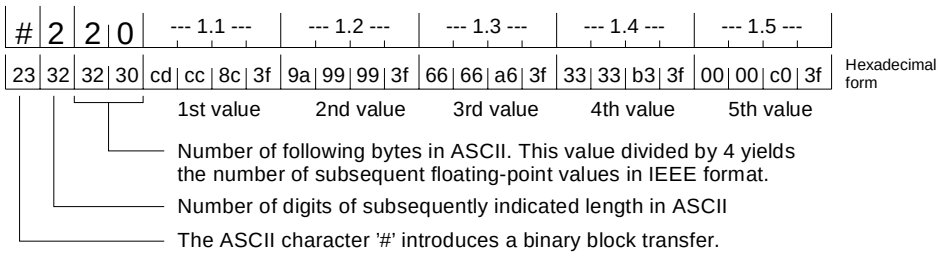
```
shellcmd$ = "UPMD5.EXE " + hostfina$ + "> " + hostcheckfina$
SHELL (shellcmd$)
```

To fetch the signature of a file stored on the UPL under RS232 control, the command `"MMEMory:CHECk? 'filename'"`
is to be sent to the UPL.

3.17.6 Differences to Remote Control via IEC/IEEE Bus

- A command line sent to the UPL must always be terminated with <New Line> (ASCII code 10, decimal). Since the character <Carriage Return> (ASCII code 13, decimal) before the delimiter is a filler without significance, the combination <Carriage Return> <New Line> is also permissible.

- **ASCII strings** sent by the UPL as a response to the controller via the RS-232 interface are always terminated with <New Line> (ASCII code 10, decimal).
  Block data in binary form are sent to the controller via the RS-232 interface without delimiter. The number of bytes can be seen from the digits sent ahead of the block data:



Program examples 3.17.4.1 Readout of Measurement Results in QuickBASIC and 3.17.4.3 Readout of Measurement Results in Borland-C 3.0 illustrate the readout of binary block data.

- With remote control via RS-232 all the common commands related to SRQ control are not useful. This can be seen in the table under 3.9 Common Commands . An exception is the *OPC command. The main task of this command is to trigger an SRQ when a measurement result is available, but in the case of RS-232 control, it may be used to wait in a loop for a measurement result by querying bit d0 of the Event Status Register, see 3.17.4.1 Readout of Measurement Results in QuickBASIC.

- Common commands (3.13.1) and addressed commands (3.13.2) are not used in RS-232 control.

- With RS-232 remote control, an SRQ procedure cannot be programmed, ie the control program is not able to carry out another task while it waits for measurement results or error messages from the UPL.
  Error queue or status register must be queried cyclically in the control program.

- Since an SRQ routine is not available, there is no need to determine the sender or the source of an SRQ in a serial or parallel poll.

- Line messages like REN, GET, DCL, etc are not possible.

- The line message EOI denoting the end of a binary block cannot be used.
  If binary data are to be received nevertheless, the expected block length must be used instead.
  Remember that data bits = 8 and parity = NONE must be set for the receipt of binary data.

- All registers and commands of the status reporting system not referring to SRQ generation may be used:

*STB?	not usable
*SRE	not usable
*PRE	not usable
*IST?	not usable
***ESR?**	**usable**
*ESE	not usable
STATus:**OPER**ation**?**	**usable**
STATus:**QUES**tionable**?**	**usable**
STATus:**XQUE**stionable**?**	**usable**
SYSTem:**ERR**or**?**	**usable**