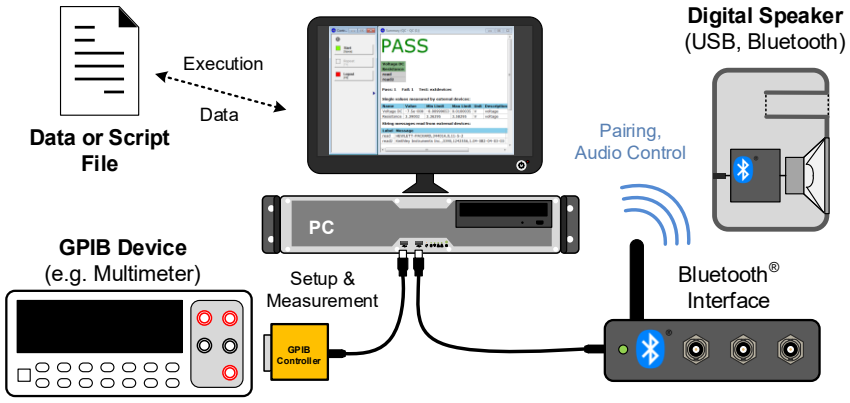
QC – External Devices (EXD) S31

QC Version 7, dB-Lab 212 (Document Revision 1.13)

FEATURES	BENEFITS
• *Bluetooth*® radio and sound device control • GPIB compliant protocols (IEEE 488 & 488.2) • Communicates with any GPIB device for control, measurement and data acquisition • Device status handling (IEEE 488.2) • Results included in Pass/Fail verdict • Flexible limit setting • Simple preset and flexible custom sequences • Shell execution and message box generation • GPIO Control • Versatile stimulus generator • Read measurement data from text file	• Integrate 3rd party instruments • Extend capabilities of QC system • High flexibility through step sequence • Automatic pairing for Bluetooth device test • Control playback volume • Start and get results from external apps or scripts • Remote control and automation • Instruct and prompt operator with message boxes • Control peripherals via GPIO



The *QC External Devices* task is a versatile add-on for the QC framework of the *KLIPPEL Analyzer System* dedicated to interfacing with arbitrary GPIB (IEEE 488) compliant measurement equipment, such as multimeters or power supplies. Simple programmable step sequences enable the user to control 3rd party instruments and to include the measured data in the QC test (*EXD Pro*). Test limits can be applied to generate PASS/FAIL verdicts. In addition to GPIB support, many other interfaces provide simple solutions for test automation and control.

EXD Bluetooth is a limited version of the EXD dedicated to pairing and control of Bluetooth enabled devices with simplified user interface in *Preset Mode*.

APPLICATIONS:

- Testing of complex DUTs (e.g. handsets)
- Electronics testing
- Automatize complex tests
- Bluetooth & USB device testing

Item Numbers	4000-241 (EXD Pro), 4000-251 (EXD Bluetooth)

CONTENTS

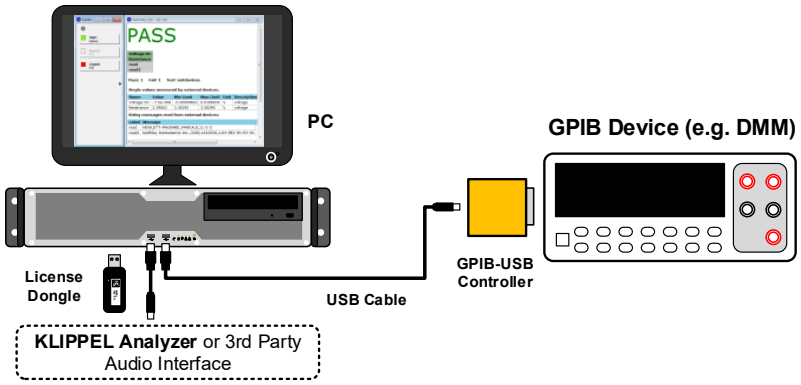
1 Overview

Summary	The *External Devices* (*EXD*) is an add-on to the *Klippel Analyzer System* for the *QC Software* framework. This versatile, step-based QC task provides interfaces for communication with 3rd party devices and applications. A wide range of measurement equipment can be controlled and queried via *GPIB* (*General Purpose Interface Bus*) according to IEEE 488.1 and IEEE 488.2 standard to include external measurement data in the QC System. In addition to GPIB communication, the EXD provides functionality for *Bluetooth*® and sound device handling or accessing the *KLIPPEL Analyzer's Digital I/O* port, creating message boxes for the operator, command line execution and text file-based data import. It also provides different test signal generators for tests with 3rd party instruments. All external data acquired by the EXD is handled like normal QC test results including limit calculation, process control and statistics. With *EXD Pro* license, custom step sequences can be generated using the full feature scope of the EXD empowering the user to create complex test and control sequences. For common straight-forward applications like *Bluetooth* audio control, the EXD also provides a preset mode for minimum effort and quick setup. Some presets can be operated with dedicated licenses (e.g. *EXD Bluetooth*) without requiring the full *EXD Pro* license.

Requirements	**Software**
	• KLIPPEL QC framework
	  ○ *QC Standard* (Item No. 4005-001) or
	  ○ *QC Stand-alone* (Item No. 4005-500)
	• R&D framework: dB-Lab version 210 or higher
	License
	• No EXD license: only Sound Device preset mode
	• EXD Bluetooth: unlocks *Bluetooth Audio* preset mode
	• EXD Pro: unlimited custom sequence, full feature set
	Hardware
	• Production Analyzer, KLIPPEL Analyzer 3 or 3rd party audio interface
	• USB license dongle (or KA3 as license device)
	GPIB (if required):
	• GPIB compatible device (DMM, generator…)

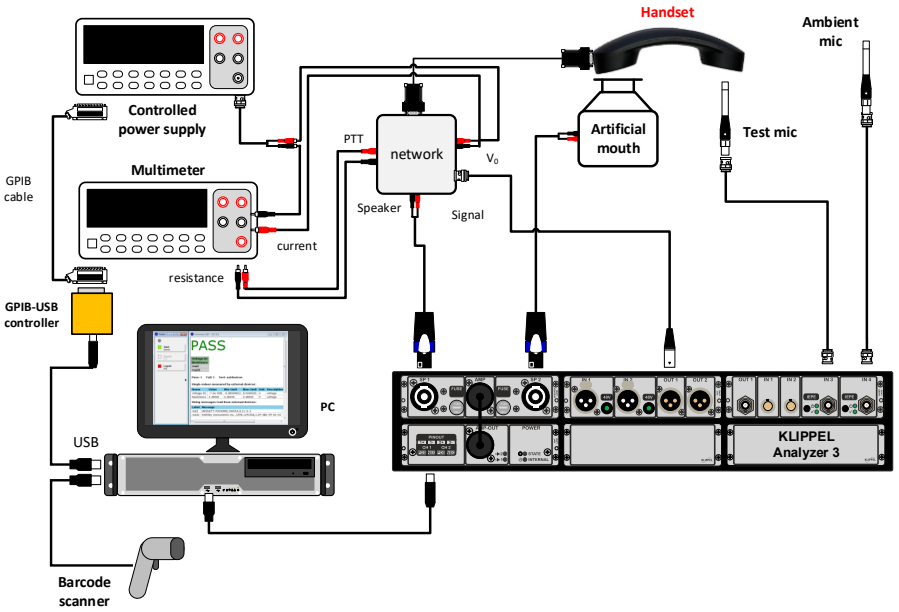
	• GPIB – USB controller interface (see below) **Bluetooth Control** • MegaSig U980 (USB-controlled, analog Bluetooth interface) • Windows Bluetooth radio
Available Step Types	• GPIB: Configure and query arbitrary GPIB devices • Digital I/O (GPIO): Write and poll the state of the Digital I/O port pins of *the Production Analyzer* • Bluetooth: Pairing and audio profile control • Sound device: Volume control and device verification for Bluetooth and other digital audio device tests • Command line: execute command line entries e.g. to start external programs • Message box: Communicate with operator using message boxes that interrupt the execution sequence and import external test data. • File: read data from text file • Stimulus: Generate and configure stimulus signals for Production Analyzer outputs using the internal signal generators or imported wave files • IO Monitor: communicate with external custom IO monitor applications
Limitations	Data acquisition and result processing is limited to single values in plain, standard scientific exponential or hexadecimal notation or as string parameters without numerical processing Parallel GPIB event handling is not supported generally, except bus wide triggering and service request No internal analyzer is provided by the EXD Task. Therefore, no measurement can be performed using the signal in- and outputs of the analyzer (PA or KA3) hardware while the EXD Task is running.

2 Hardware Requirements

Typical Setup	 The figure above shows the minimal equipment required to use the EXD for GPIB control and test applications: • (Klippel Production Analyzer, Klippel Analyzer 3 or 3rd party audio interface) • Personal computer (see separate PC requirements) • USB license dongle • GPIB compatible device • GPIB-USB interface for host computer (see below) • GPIB cable, if multiple devices shall be connected
GPIB Interface for Host PC	 The EXD is dedicated to be operated with *Prologix GPIB-USB Controller*, a cost-effective and versatile GPIB-Bus interface. It does not need any additional host software and can be addressed as a virtual serial port. The interface handles the low-level GPIB communication to send high-level GPIB commands and receive data transmitted by GPIB devices. The configuration and communication is handled by the EXD. Interface versions 6.91 or higher are supported. Older versions may be used as long as the standard instruction set to setup and communicate with the controller is supported.
Bluetooth Interface	 The EXD provides a simple interface for pairing and controlling Bluetooth enabled devices using *MegaSig U980* analog Bluetooth interface or *Windows Bluetooth* stack. The U980 I recommended as a professional device for optimal performance. Find more information in specification *A6 Accessories*.

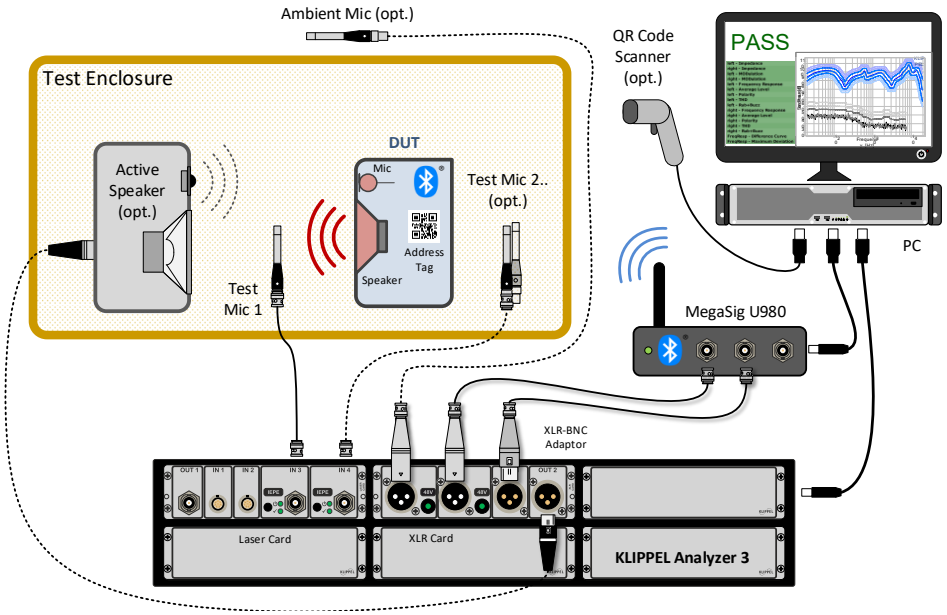
3 Applications

Handset Test



The figure above shows an example test scenario for a handset including electronics and acoustic test. The DUT supply voltage is provided by a programmable power supply with GPIB interface controlled by the EXD task. A digital multimeter is used to test DC current and resistance of the handset electronics. The EXD sequence triggers the measurement and checks the results against limits using to generate PASS/FAIL verdicts. This all is combined with acoustical test of the receiver and microphone using the KA3 and QC Sound Pressure (SPL) task.

Bluetooth Device Test (Pairing & Audio Control)



For test automation of Bluetooth-enabled speakers and headphones, the EXD enhances any QC test sequence with Bluetooth pairing and audio control through *MegaSig U980* analog Bluetooth interface.

At the beginning of the test sequence, the EXD triggers scanning of the DUT address for pairing or the next available device is paired. The A2DP audio profile is activated

	to test the speaker output using a test microphone connected to the KA3. In the next step, the HFP (hands-free) profile is activated for testing the DUT's microphone response. The reference speaker is connected to the output of the analyzer to play back the test signal.

4 Limits, Settings and Results

SETUP PARAMETERS	
Preset Mode	In addition to custom EXD Step Sequence, various preset sequences can be used instead: • Bluetooth Audio • Sound Device Handling
EXD Step Sequence	In addition to the provided preset modes, custom test and control step sequences can be defined with plain ASCII strings following simple format conventions. Each line represents one step (step type: GPIB, Digital IO etc.) with a corresponding action (read, write, value). Each line has the following fixed format: '<Label of test step or measure> | <Step type> | <Step action> | <Step parameter 1> | <opt:Step parameter 2> | <opt: unit> | <opt: comment> ' The setup may be prepared in a simple text editor and copied via clipboard. Find a table of all available step types and parameters in the appendix.
Label	This is a label or measure name to be shown in the verdict list and result table in *Summary* window. The step labels must be unique within one sequence.
Step type	Step type specifies communication channel or mode: • `GPIB`: GPIB device communication • `DigitalIO`: Digital I/O (GPIO) port communication (Production Analyzer) • `Bluetooth`: Pairing and profile (service) control • `SoundDevice`: Device check and volume control for WDM devices • `IOmonitor`: IO monitor communication • `Wait`: wait specified time • `MsgBox`: message box for operator communication • `Execute`: execute any command line (shell command) • `Stimulus`: generate stimulus signal using the output of the analyzer • `File`: read data from text file
Step action	This parameter specifies the action performed in the step: • `write`: write to device or perform action without returning data • `read`: read character string (message/data) • `value`: read numerical string data to create a QC measure
Parameter 1	The interpretation of step parameter 1 depends on the specified type and action. **GPIB: command sequence** Device (SCPI or other GPIB protocol) or bus controller command sequence for device setup, query and measurement. **DigitalIO: bit sequence (only "write")** A bit mask (e.g. "001010011") for Digital I/O (GPIO) port of the analyzer hardware. The bits correspond to specific pins of the port. The corresponding mask

	in *Parameter 2* specifies which bits shall be set. Alternatively, the value "x" keeps the previous state of the pin (e.g. "00xx10x11"). • **Bluetooth: command (only "write")** Pair device by availability, address or friendly name; start and stop profiles like A2DP (audio sink) • **SoundDevice: command** Format: <Playback/Capture> <Device/Volume/Level> <opt:{device name} • **IOmonitor: control matrix (only "write")** Format: <channel value mask aux> Used for communication with your IO monitor application. See IO monitor API documentation for more information. • **Execute: command line** Command line (incl. parameters) to be executed in the windows shell. The task waits until the action is terminated. Use quotation marks in case the path or filename contains white spaces. • **MsgBox: message string** Opens a message box with the specified message string. The task waits until the message box is terminated. • **Wait: time in s (only "write")** The task execution is interrupted for the specified time. • **Stimulus: file path** o **[GenMode fstart, fstop, res, voltage, time, loop]** for internal or o **[GenMode filePath, voltage, loop]** for GenMode="wave" o only "write" Configure the built-in stimulus generator or specify wave file input. • **File** File path (absolute or relative) of file to be read. See section *Examples* for practical examples. Please also see the overview table in the appendix.
Parameter 2	The interpretation of step parameter 2 depends on the specified type and action. **GPIB: device address** The GPIB address (1-30) of the used device in this step is to be specified here. This entry is compulsory for GPIB communication steps. The device address has to be unique within the bus and can be set or determined manually at the particular device. **Bluetooth: device name/address or profile/service ID** (only "write") **SoundDevice: timeout, volume or level** **DigitalIO: bit mask (only "write")** A bit mask of 9 bits (e.g. "111000011") is specified here to set which bits of the Digital Output shall be set according to *Parameter 1*. Only pins with the mask "1" are set, while "0" keeps the previous state. **Stimulus: output routing** (only "write") Set the output of the Production Analyzer to use for stimulus playback (Speaker1/2, Out1/2) **MsgBox: message string** initial value or character string of the data input field

	File: parameter name Name of the parameter in the source file to be read:
Unit (optional)	Specifies the unit of the measured quantity for display purpose in the result table. This entry is obsolete for "`write`" steps.
Comment (optional)	Optionally a comment can be provided which is used for the result table only. This entry is obsolete for "`write`" steps.
Error Mode	In case of a step error the final result may be • force FAIL: All measures are marked as FAIL (red color) • Ignore: All measures are marked as VOID (grey color) • Warning: All measures are marked as Warning (Yellow color) In any case, errors are indicated in the verdict table.
Error Handling	In case of a step error • Abort • Continue • Prompt (*OK, Cancel, Retry*) All error messages are logged and displayed in case of "continue".
Digital Out (amp check)	GPIO setting applied during amplifier check (if *Stimulus* step and *Speaker* routing step are used).
Import Parameters	Activates task and limit parameter import from external parameter file
Parameter File	File path of parameter file for import of settings and limits
GPIB PARAMETERS	
Bus Mode	Select bus mode used for communication with GPIB devices • none • GBIP (IEEE 488.1) - use GPIB via virtual COM interface provided by Prologix GPIB-USB controller • GBIP (IEEE 488.2) - use GPIB via virtual COM interface provided by Prologix GPIB-USB controller, full IEEE 488.2 common command set and status handling is used
Termination Character	Specify GPIB command termination characters. These characters are appended to any command sent to a GPIB device. • None • CR + LF • CR (carriage return) • LF (line feed)
EOI Termination (enable/disable)	Enable assertion of EOI (end of instruction) signal with last character of any command sent over GPIB bus. Some instruments require EOI signal to be asserted in order to properly detect the end of a command.
Status Handling (enable/disable)	Enable internal IEEE 488.2 compliant status register handling to identify state of GPIB device and communication errors. includes: SRQ, Message available, Standard Events Register, Questionable data flag All enable registers are initialized automatically for all standard status bits.
Init Controller	Select the initialization mode of the GPIB bus controller • automatic: automatically detect controller (virtual COM port) • manual: set the virtual COM port number manually
COM Port	Specify virtual COM port number of GPIB bus controller. (according to settings in "*Bus mode*")

	Only available if *Init controller – manual* is selected.
Read Timeout	The *Read timeout* specifies the global time span for serial port or GPIB communication timeouts (read access) in seconds.
BLUETOOTH PARAMETERS	

	• Clear Paired Devices
	• Pairing
	  ○ Auto
	  ○ Address
	  ○ Name
	• Input Mode
	  ○ Enter
	  ○ Prompt
	• Timeout
	• Select Device
	• Unpair (Remove)
	• Activate/Deactivate Profiles
	  ○ A2DP (Audio Sink)
	  ○ HFP (Hands-Free)
	  ○ AVRCP
	  ○ AVRCP Target
	  ○ Custom

SOUND DEVICE PARAMETERS	

	• Verify Playback/Capture Device
	• Timeout
	• Set Playback/Capture Volume
	  ○ Volume
	  ○ Level
	• Input Mode
	  ○ Enter
	  ○ Prompt

LIMIT PARAMETERS (SETUP)	
Limit Setup (“value”)	The limits settings for single value measures are defined similar to the *Measurement sequence*: • Each line defines limit for one “value” step. For multiple steps simply enter more lines. • Each line has the following fixed format: `’<Measure name (step label)> | <Limit calculation mode> | <opt1:Limit parameter min> | <opt2:Limit parameter max> | <opt3:Cpk-PoolSize>} | <opt3:Cpk-Limit> | <opt3:Ppk-Limit> | <opt3:Cpk/Ppk-PassedOnly>’` • The number and order of line entries may be chosen arbitrary • Missing entries will deactivate the limits for the corresponding measure (passive – void verdict) • The measure names/labels must correspond to the unique step labels in the *Measurement sequence* setup matrix
Limit Calculation Mode	• Shift: Values are added to the mean of reference results • Relative:

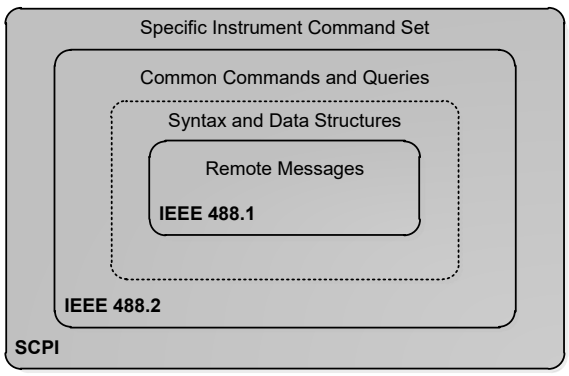
	Values are multiplied: limit = refValue * (1 +/- relative tolerance). Relative tolerance is specified in percent. • Statistics: Values are multiples of the standard deviation of the reference results. At least 2 reference DUTs must be used. • Absolute: Values are directly compared to the measured results. Reference DUTs are not used for limit calculation. No limits are calculated if left empty. For more details on limit calculation, refer to the *QC User Manual*.
Limit Parameter Min	For each Limit calculation mode the upper limit parameter is specified here. • Shift: Shift Min • Relative: Tolerance Min • Statistics: Factor of sigma for Min • Absolute: Min Limit No upper limit is applied if empty.
Limit Parameter Max	For each Limit calculation mode the lower limit parameter is specified here: • Shift: Shift Max • Relative: Tolerance Max • Statistics: Factor of sigma for Max • Absolute: Max Limit No lower limit is applied if field is left empty.
Imported Measured Data	Import external reference data (reference DUTs or mean values)
Imported Standard Deviation	Import external standard deviation (related to mean values of *Imported Measured Data*)
Cpk / Ppk	The Cpk/Ppk process analysis is activated if all four parameters for the corresponding single value measure is specified in in the *Limit setup* • Cpk-Poolsize: • Cpk-Limit: • Ppk-Limit: • PassedOnly
Limit Setup (“read”)	The limits settings for character string messages are defined similar to the *Measurement sequence*: • Each line defines target string for one “read” step. For multiple steps simply enter more lines. • Each line has the following fixed format: `'< Label (step label)>	<Mode>	<Target string>'` • The number and order of line entries may be chosen arbitrary • Missing entries will deactivate the limits for the corresponding measure (passive – void verdict) • The labels must correspond to the unique step labels in the *Measurement sequence* setup matrix
Mode	• equal: Pass if input string equals target string. • notEqual: True if input string differs from target string No limit is set if left empty.
Target String	Here the target string is defined which is to be compared with the input string. Target check is deactivated if left empty.

RESULTS	
Single value measures	For “value” steps the input data is interpreted as numerical data. The results are shown in a measure table on the *Summary Page*. Limits are applied and shown, if specified.
Cpk / Ppk	If Cpk / Ppk values are enabled and can be calculated, they will be displayed in separate columns within the result table in the *Summary* window. The standard processes are applied, please refer to the user manual for more details on Cpk/Ppk.
String messages	For “read” steps the input string data is not interpreted as numerical data. The message is displayed in a results table on the *Summary window*. Limits can be applied in terms of a target string comparison.
Log File	All results are logged in the summary.log – file, if enabled.
Database	All results are stored in the result database, if enabled. Using the *Automation* interface all results and limits can be accessed.

5 GPIB Details

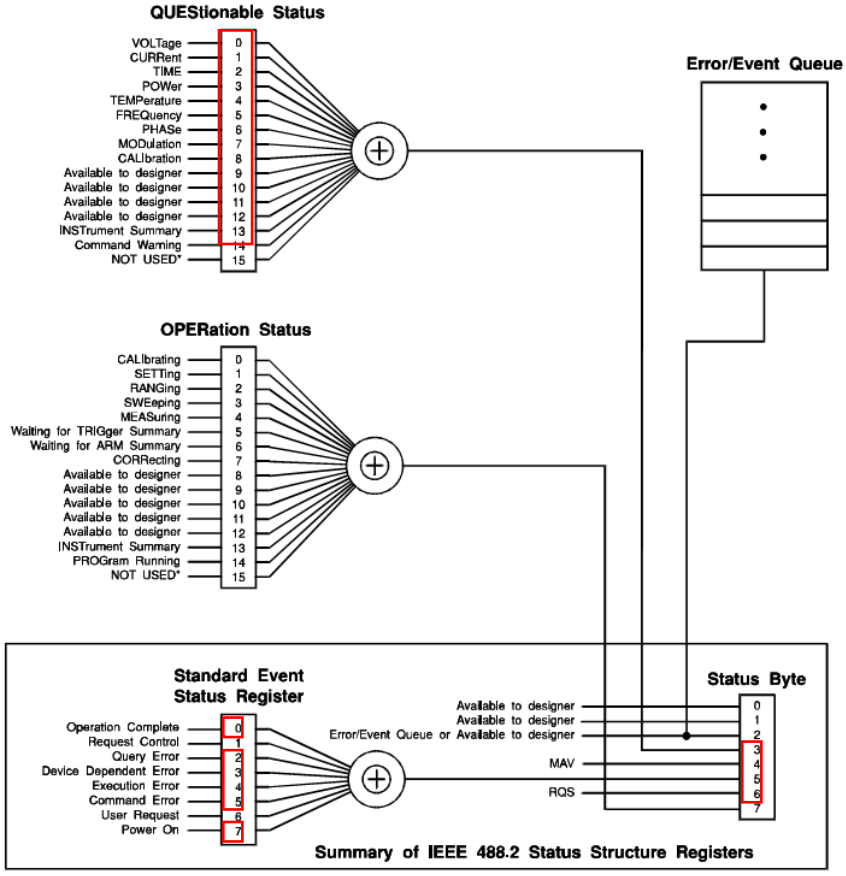
Summary

The *GPIB* (*General Purpose Interface Bus*) according to *IEEE 488* industry standard is a versatile and very common communication bus for all kinds of instrumentation equipment. It specifies the mechanical and electrical bus structure and a selection of hardware level communication protocols. Additionally, the extended *IEEE 488.2* standard specified controller functionality, common commands and functionality as well as a device status reporting system for status information and device error handling. Built on the *IEEE 488.2* standard the *SCPI* (*Standard Commands for Programmable Instruments*) recommendation introduced a simple and versatile device independent programming syntax to minimize manufacturer specific differences and to support portability.



The QC External Devices module combined with the *Prologix GPIB-USB* controller provides a high-level interface for GPIB communication. Compliant devices can be set up and queried to generate QC results. Only the specific commands and the device address have to be specified by the user. All device initialization and status handling is performed automatically according to the task settings. The application of the SCPI status handling is explained in section *Status Handling*.

Additionally, this module offers communication with the *GPIO* port of the *Production Analyzer* hardware and custom *IO Monitor* applications extending the

	communication channels. Furthermore, any shell command can be executed. Operator interaction is provided using message boxes.
Status Handling (IEE 488.2)	With the extended IEEE 488.2 standard, a general GPIB device status handling system was introduced to monitor the device status in terms of service request and synchronization as well as communication and device errors. The standard comprises four *status registers* according to the figure below. Each register mirrors the current device status according to the information represented by the concerning bit. There are device independent common states as well as device specific states defined by the manufacturer. All status register can be queried independently. Additionally, every status register is followed by an *enable register* (not displayed). It acts as a mask controlling which entries in the register are used to build up a sum status bit (logical OR) which is forwarded to the *status byte register*. The *status byte* is summing up the information given in the state registers' or the state of the device's output buffers. The status byte can be read any time by the controller even parallel to pending operations with a *serial poll* to synchronize GPIB communication. Setting the *status byte enable register* mask can be used to generate a *service request* (*SRQ*) interrupt on the bus any time.  Status register structure according to IEEE 488.2[1] The External Devices module supports automatic status handling. The operation status register is excluded. During device initialization all standard (device independent) *enable registers* are set for *service request*. The relevant bits are

	marked in the picture above. The *standard event register* may generate step errors while the *questionable status register* only causes warnings. The status of every device is monitored during communication and the user is informed, if any errors or warnings occurred. Synchronization is given at any time referring to *MAV* (*message available*) and **OPC?* (*operation complete*). The device status is reset at the beginning of the task's sequence. For the purpose of customization and additional feedback all enable registers can be set and all status registers can be queried by the user anytime in the control sequence. For further information please refer to the user manual of the External Devices Task and to the programmer manual of your GPIB device.

6 References

Specifications	Software • C3 - QC Set Hardware • A6 – Accessories for the KLIPPEL Analyzer System • H3 - Klippel Analyzer 3 Other • KLIPPEL Amplifier Requirements
Application Notes	• AN72 Testing Wireless Audio Devices with Klippel R&D System • AN73 QC Headphone Testing All KLIPPEL application notes can be downloaded from www.klippel.de
3rd Party Resources	• Prologix, LLC. Features of GPIB-USB Controller. [Online] [Cited: August 23, 2010.] http://store.prologix.biz/gpco.html. • SCPI Consortium. SCPI Specifications. [Online] 1999. [Cited: August 23, 1010.] http://www.ivifoundation.org/docs/SCPI-99.PDF. • MegaSig U980 Resources [Online] [Cited: January 10, 2020.] http://www.megasig.com/en/product-169-c2632-p1.html

Appendix

Overview of Available Step Types and Parameters

Label/measure name	Step type	Step action	Parameter 1	Parameter 2	Unit (opt)	Comment (opt)
<label >	GPIB	write	<command>	<GPIB address>	-	-
<name>	GPIB	read	<command>	<GPIB address>	<unit>	<comment>
<measure name>	GPIB	value	<command>	<GPIB address>	<unit>	<comment>
<label >	Bluetooth	write	<command>	<name/address> <service ID>	-	-
<label >	SoundDevice	write read value	<Playback/Capture> <Device/Volume/Level> <opt:{name}>	<timeout> <volume/level>	-	-
<label >	Execute	write	<Cmd line entry>	-	-	-
<label >	MsgBox	write	<message>	-	-	-
<label >	MsgBox	read	<message>	<init input>	<unit>	<comment>

<label >	MsgBox	value	<message>	<init input>	<unit>	<comment>
<label >	IOmonitor	write	<chan val mask aux>	-	-	-
<label >	DigitalIO	write	<bit sequence>	<bit mask>	-	-
<name >	DigitalIO	read	-	-	-	<comment>
<label >	Wait	write	<Time>	-	-	-
<label >	Stimulus	write	<config>	<routing>	-	-
<measure name>	File	value	<file path>	<var name>	<unit>	<comment>
<name>	File	read	<file path>	<var name>	<unit>	<comment>

Find explanations for symbols at:
http://www.klippel.de/know-how/literature.html
Last updated: March 10, 2022