

# Using USB Communication to Control External Instruments with the 4200A-SCS Parameter Analyzer

## APPLICATION NOTE



**KEITHLEY**  
A Tektronix Company

**Tektronix**

## Introduction

Testing advanced technology often requires a variety of different instruments. Beyond current and voltage measurements, a test may include temperature measurements, advanced waveform generation or sensitive low level measurements. Testing setups can require prober control, switching control, and additional lights or electromagnets. The Keithley 4200A-SCS Parameter Analyzer has the ability to control these external instruments and devices using GPIB communication through its built-in software tool, Clarius.

Beginning with Clarius V1.9, the 4200A-SCS will now be able to control instruments through USB communication. The Clarius software includes user libraries to control a generic USB instrument and one that controls the Keithley DMM6500

and DMM7510 for temperature measurements. A user can also write their own user library using NI VISA commands.

This application note will explain how to use the two built-in user libraries for USB control, as well as help the user write their own user library for controlling an external instrument via USB.

## Using the Generic USB Actions and Modules

The first library provided is the generic\_visa\_ulib. This user library contains six modules that allow the user to send commands and receive data directly in Clarius. Each module has been added as an action template in the Clarius library, under the Actions tab, and can be filtered by checking the Communication filter as shown in **Figure 1**.

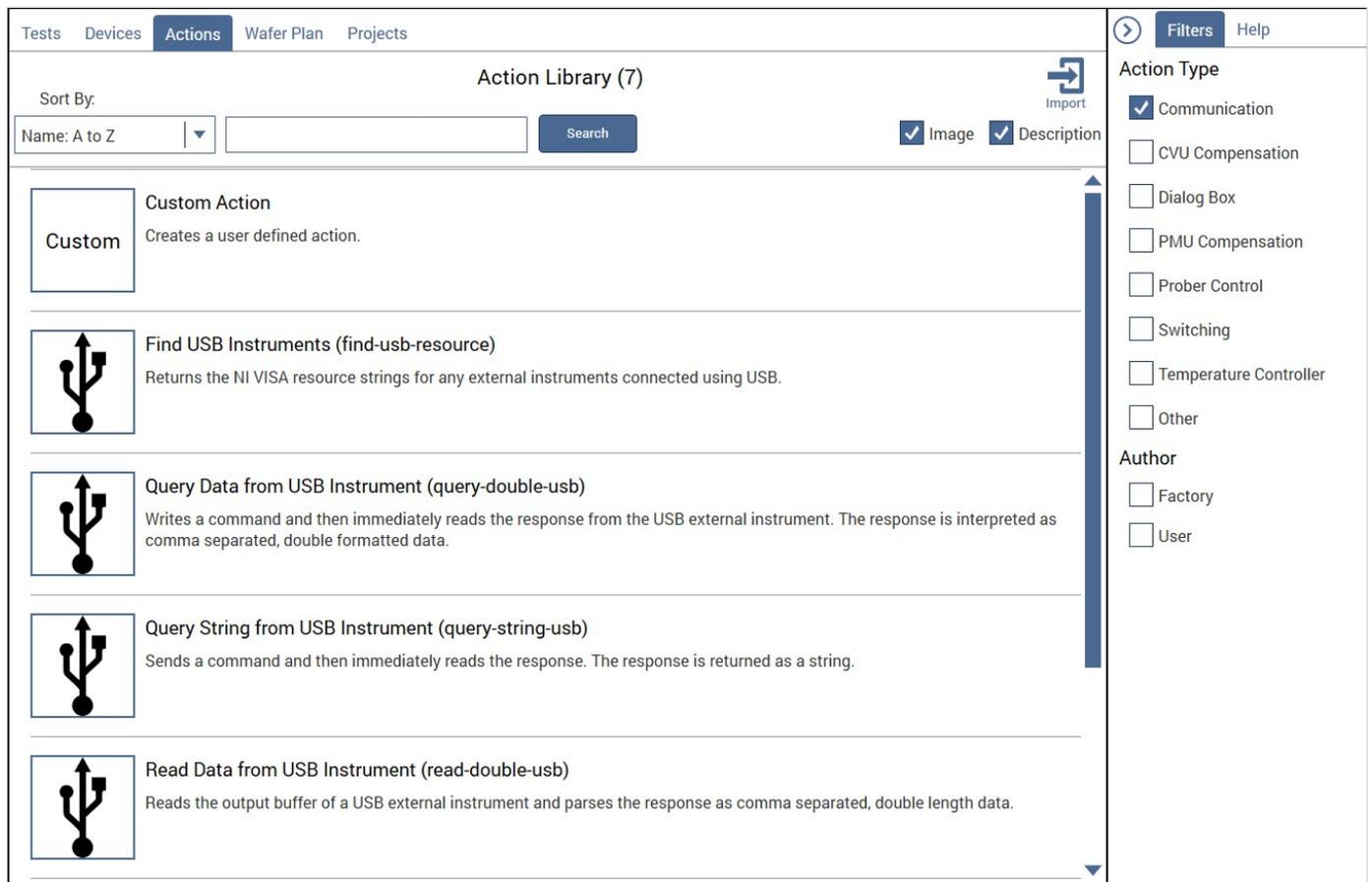


Figure 1: Generic USB actions in Clarius Library.

Using the generic\_visa\_ulib eliminates the need to create a user library and module for small applications such as turning on a power supply or triggering an AFG. The actions can be quickly added to a project and configured. For applications requiring more than a few commands, see the section on creating a custom user library using NI VISA. The actions and their corresponding modules are described in **Table 1**.

Action Name	Module Name	Description
find-usb-resource	find_visa_resource	Finds all USB resources connected to the 4200A-SCS
write-string-usb	write_string_visa	Writes a command or string of commands to an external instrument
read-string-usb	read_string_visa	Reads the external instrument's output buffer and returns the data as a string to the Clarius sheet.
read-double-usb	read_double_visa	Reads the external instrument's output buffer and returns the data. The data string is assumed to be comma separated double data and is parsed into up to 4 columns in the Clarius sheet.
query-string-usb	query_string_visa	Writes a command or string of commands to an external instrument then reads the external instrument's output buffer and returns the data as a string to the Clarius sheet.
query-double-usb	query_double_visa	Writes a command or string of commands to an external instrument then reads the external instrument's output buffer. The returned string is assumed to be comma separated double data and is parsed into up to 4 columns in the Clarius sheet.

Table 1: Generic USB communication actions.

The first action is find-usb-resource. This action returns the NI VISA resource strings for all of the available USB instruments to the Clarius sheet as shown in **Figure 2**. The resource string is what NI VISA uses to identify and communicate with the instrument. The desired resource string must be copied from the Clarius sheet and pasted as a parameter to any of the subsequent actions used.

find-usb-resource#1			
Run1 Formulas List			
	find_visa_reso	num_instr_four	instrument_list
1	0	1	USB0::0x05E6::0x7510::04345321::INSTR

Figure 2: Finding the available USB external instruments

The other five actions write commands and/or read data. For example, the query-double-usb action, shown in the Configure View in **Figure 3**, allows a user to send a command and receive data.

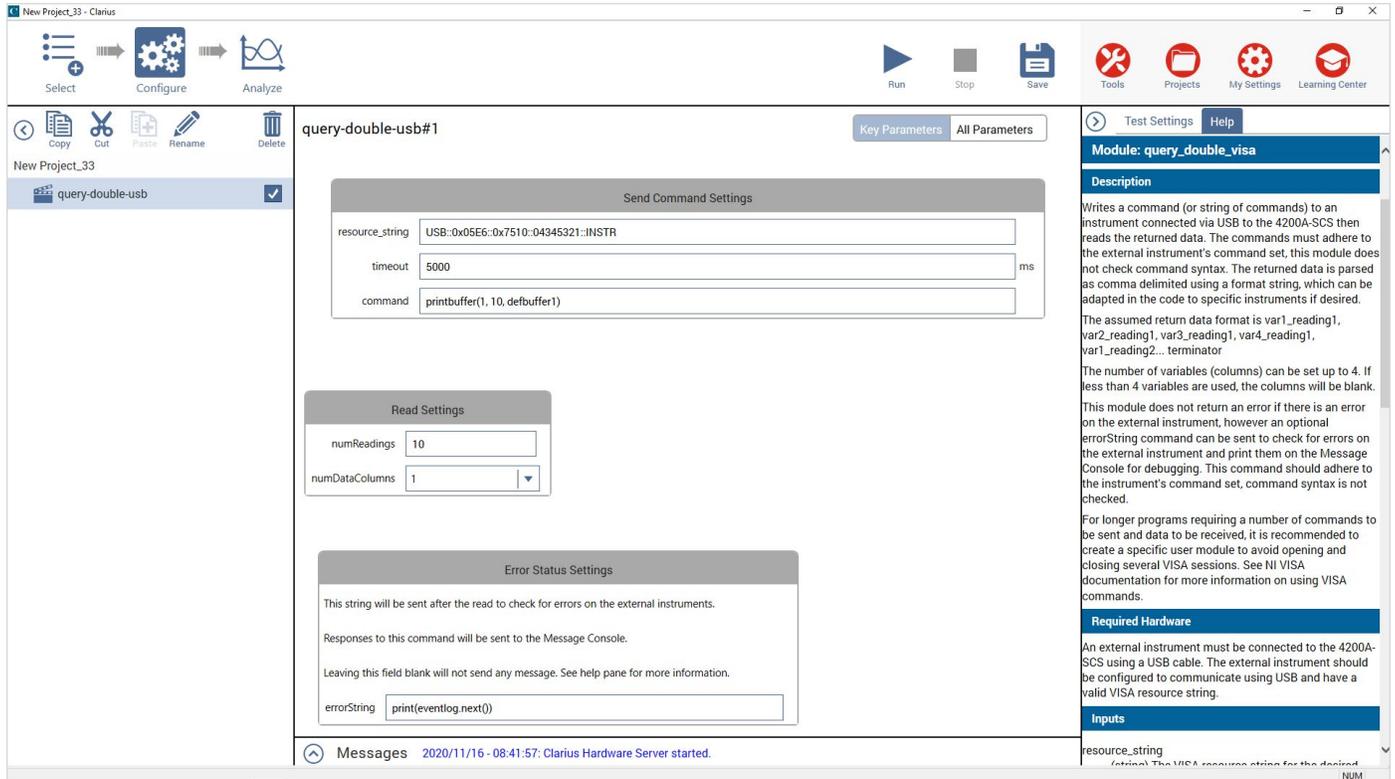


Figure 3: Configure the query-double-usb action.

The command or string of commands are entered in the command parameter. These commands must adhere to the external instrument’s command set and syntax. The generic USB actions do not verify command correctness. The data returned by the query is assumed to be double type data that is comma separated. The data can be parsed into up to four columns using numDataColumns and into a number of rows set by numReadings.

The errors returned by the generic USB library are NI VISA errors. To view errors occurring on the external instrument, users have the option of entering an errorString at the bottom of each of the actions. The errorString should be a command that can return the status of the external instrument. This string is sent to the external instrument after the read or write is completed, and the response is immediately read and returned to the Message Console.

## Using the DMM Temperature Library and Project

Temperature measurements can be done using an external instrument controlled by the 4200A-SCS. The `dmm_6500_7510_temp_ulib`, provides an example of controlling a DMM6500 or DMM7510 digital multimeter with NI VISA. The DMM uses a thermocouple to measure the temperature. The `dmm-meas-temp` project has also been added to the Clarius Library demonstrating each of the modules. **Table 2** shows a description of the project and the modules used.

Test	User Module	Description
time-dmm-meast	meas_time_dmmTemp	Sets up the DMM to measure temperature and returns the measurement and timestamp to the Clarius sheet.
smu-measi-dmm-meast	meas_smul_dmmTemp	Sets up the DMM to measure temperature and the source measure unit (SMU) to bias a voltage and measure current. The measurements are taken at the same time and then the current, temperature and timestamp are returned to the Clarius sheet.

Table 2. Tests in the `dmm-meas-temp` project.

Both tests require the NI VISA resource string as a parameter. This can be found by running the action, `find-usb-resource`, before the tests and then copying and pasting the desired resource string into the tests.

A view of the `smu-measi-dmm-meast` test in the Configure View is shown in **Figure 4**. The voltage and current range can be set for the SMUs and the temperature units and thermocouple type can be set for the DMM. The number of readings to be taken and the time between readings is also set.

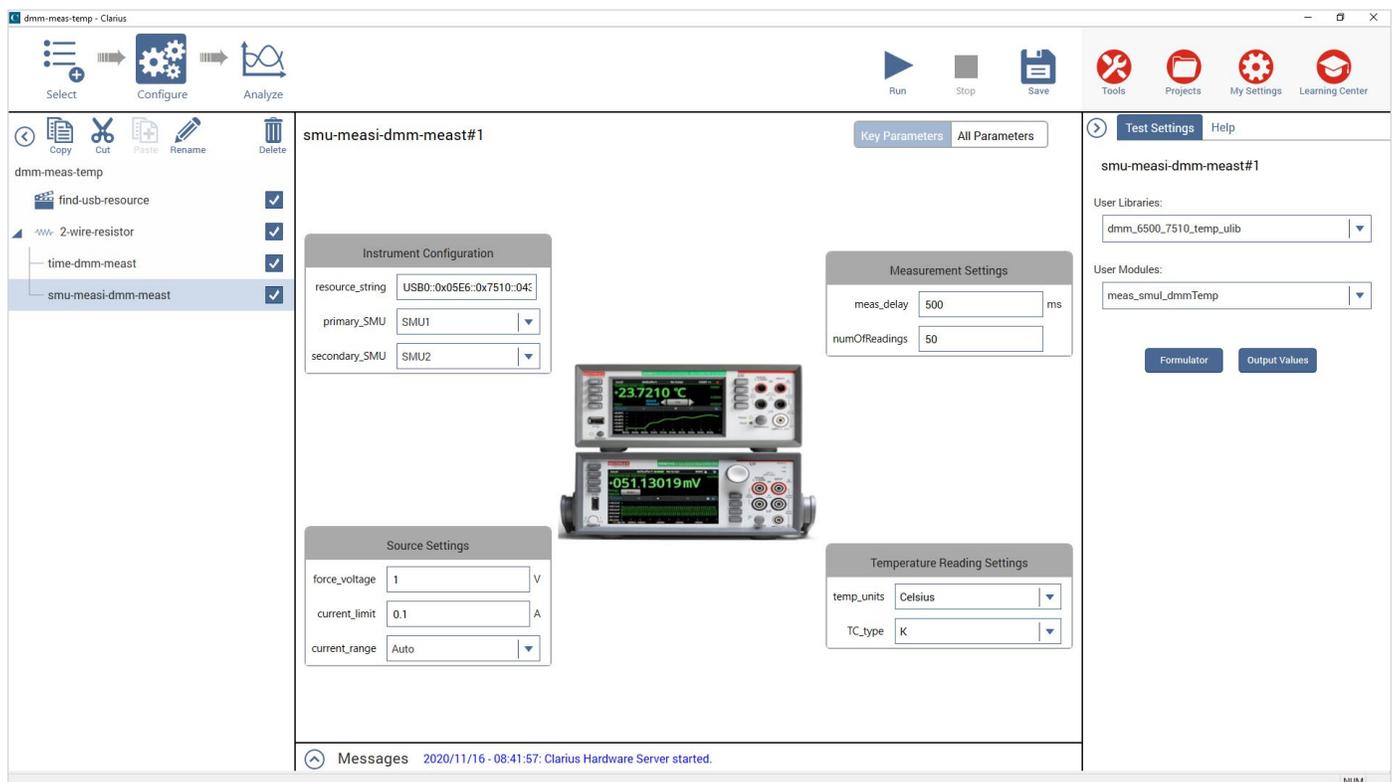


Figure 4. Configuring the `smu-measi-dmm-meast` test.

After the test is run, the current and temperature data is automatically graphed as shown in **Figure 5**.

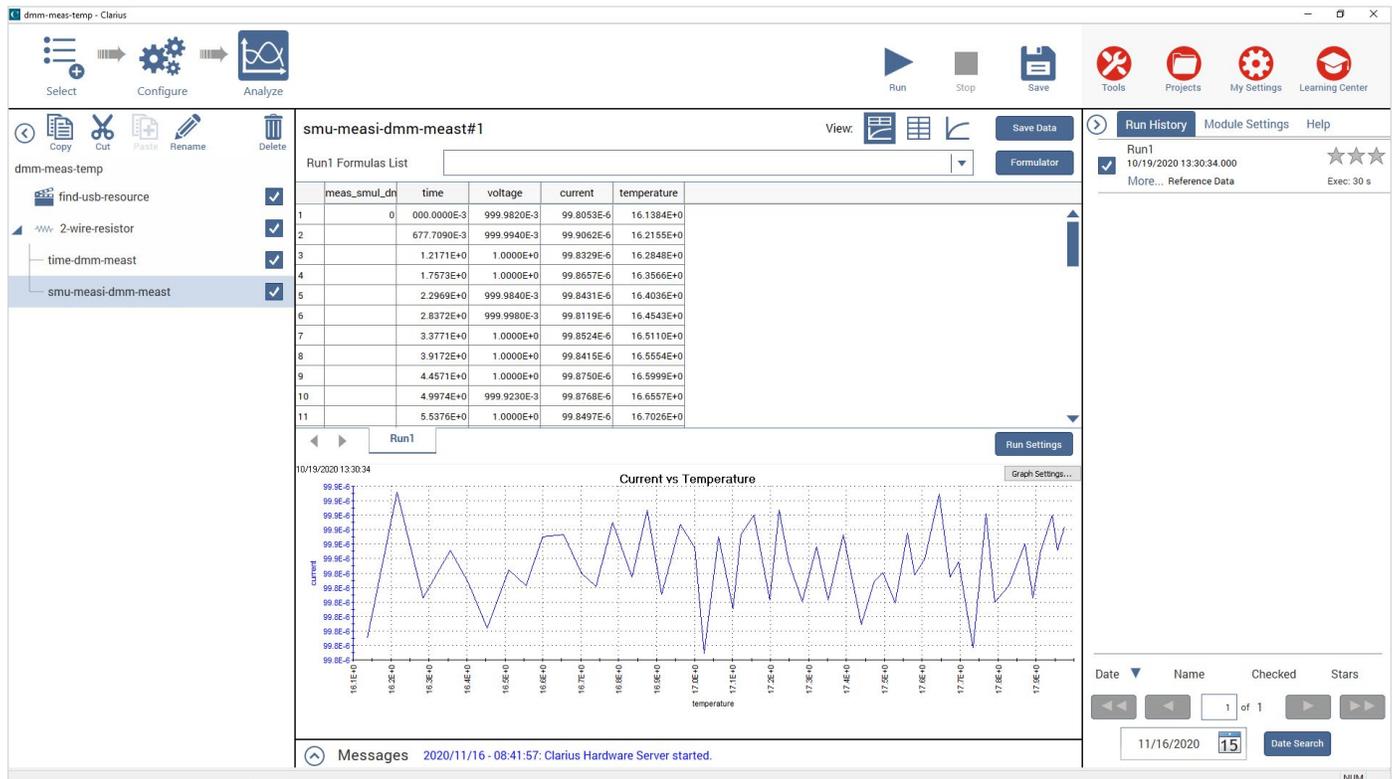


Figure 5. Current versus temperature graphed in the dmm-meas-temp project.

## Creating a Custom User Library

Users can create customer user libraries to communicate with USB-controlled external instruments, using the Keithley User Library Tool (KULT) or the KULT Extension for Visual Studio Code. See the 4200A KULT and KULT Extension Programming Manual for more information on creating user libraries with each of the tools. Users should create a custom user library for tests that require many commands to be sent to the instrument, tests that require the 4200A-SCS instruments to measure in tandem with the external instrument, or tests that require a lot of data return and analysis.

To use NI VISA in a user library, the NI VISA library must be added as a dependency and the NI VISA header files must be included in the module. The library dependency applies to all modules in a library and only needs to be completed once per library. The header files must be included in all modules. To configure a library to use NI VISA in KULT:

1. Close KULT.
2. Go to the kitt\_src folder of the desired library (C:\s4200\kiuser\usrlib\lib\_name\kitt\_src).
3. Open the .mak file for the library in Notepad or another code editor.
4. In the LIBS variable, between the quotes, enter visa32.lib. Enter any other library dependencies you may need. Note that modifying the library dependencies in KULT after this will overwrite NI VISA.
5. Save the file. Reopen the library in KULT.
6. Open the desired module in KULT or create a new module.

7. Click the Includes tab at the bottom of the window.
8. Add the following statements:

```
#include "visa.h"  
#include "visatype.h"
```

To configure a library to use NI VISA using the KULT Extension and Visual Studio Code:

1. Select the desired library in the KULT side bar.
2. Click on the library\_name.mak file in the **Miscellaneous** pane of the KULT side bar to open it in the editor.
3. In the code editor, add the visa32.lib file to the LIBS variable.
4. Save the file.
5. Open the module in the editor.
6. Under the /\* USRLIB MODULE PARAMETER LIST \*/ comment, add the following statements:

```
#include "visa.h"  
#include "visatype.h"
```

Including these files may cause an Intellisense error in Visual Studio Code, because the Intellisense configuration file cannot find the path to the header files. This error will not affect building the library, but it can be removed by opening the c\_cpp\_properties.json file from the Miscellaneous pane on the KULT side bar. In the editor, add the path to the header files: C:/Program Files (x86)/IVI Foundation/VISA/WinNT/include. Included paths should be enclosed in double quotes and separated by commas as shown in **Figure 6**. This will remove the intellisense errors.

```
.vscode > {} c_cpp_properties.json > [ ] configurations > {} 0 > intelliSenseMode  
1 {  
2   "configurations": [  
3     {  
4       "name": "KULT C Configuration",  
5       "includePath": [  
6         "C:/S4200/sys/include",  
7         "c:/s4200/kiuser/usrlib",  
8         "C:/Program Files (x86)/MinGW/**",  
9         "C:/Program Files (x86)/IVI Foundation/VISA/WinNT/include"  
10    ]  
11  },  
12  ]  
13 }
```

Figure 6: Adding NI VISA to c\_cpp\_properties.json configuration file.

Now that NI VISA had been included in the library and module, the NI VISA commands can be used to communicate. The most commonly-used commands are shown in **Table 3**.

73768	Description
viOpenDefaultRM	Initializes VISA. Must be called before any other VISA command.
viFindRsrc	Finds available instruments and returns a list of their resource strings. The list can be filtered to USB only using the format string USB?*
viFindNext	Used to iterate through the returned list of instruments from viFindRsrc to find the desired instrument.
viOpen	Opens a session to the instrument specified by the VISA resource string.
viWrite	Writes data to an external instrument.
viRead	Reads a set number of characters as a string from the external instrument's output buffer.
viClose	Closes a VISA session. This command should be used before exiting a user module.

Table 3: Commonly-used VISA commands.

Please refer to the NI VISA Programmers Manual for more information on NI VISA command syntax, usage, and error codes.

## Conclusion

The ability to control external instruments with the Keithley 4200A-SCS Parameter Analyzer opens the door to new testing possibilities, starting with Clarius V1.9. Using NI VISA gives users the flexibility to work with any USB-enabled instrument, saving time and keeping data in one location. The generic actions send quick commands without the need to write code or exit Clarius. The dmm-meas-temp projects provide a working example of configuring and using an external instrument to measure and collect data. All of these enable users to do more with the 4200A-SCS.

## Contact Information:

**Australia** 1 800 709 465  
**Austria\*** 00800 2255 4835  
**Balkans, Israel, South Africa and other ISE Countries** +41 52 675 3777  
**Belgium\*** 00800 2255 4835  
**Brazil** +55 (11) 3759 7627  
**Canada** 1 800 833 9200  
**Central East Europe / Baltics** +41 52 675 3777  
**Central Europe / Greece** +41 52 675 3777  
**Denmark** +45 80 88 1401  
**Finland** +41 52 675 3777  
**France\*** 00800 2255 4835  
**Germany\*** 00800 2255 4835  
**Hong Kong** 400 820 5835  
**India** 000 800 650 1835  
**Indonesia** 007 803 601 5249  
**Italy** 00800 2255 4835  
**Japan** 81 (3) 6714 3086  
**Luxembourg** +41 52 675 3777  
**Malaysia** 1 800 22 55835  
**Mexico, Central/South America and Caribbean** 52 (55) 56 04 50 90  
**Middle East, Asia, and North Africa** +41 52 675 3777  
**The Netherlands\*** 00800 2255 4835  
**New Zealand** 0800 800 238  
**Norway** 800 16098  
**People's Republic of China** 400 820 5835  
**Philippines** 1 800 1601 0077  
**Poland** +41 52 675 3777  
**Portugal** 80 08 12370  
**Republic of Korea** +82 2 565 1455  
**Russia / CIS** +7 (495) 6647564  
**Singapore** 800 6011 473  
**South Africa** +41 52 675 3777  
**Spain\*** 00800 2255 4835  
**Sweden\*** 00800 2255 4835  
**Switzerland\*** 00800 2255 4835  
**Taiwan** 886 (2) 2656 6688  
**Thailand** 1 800 011 931  
**United Kingdom / Ireland\*** 00800 2255 4835  
**USA** 1 800 833 9200  
**Vietnam** 12060128

\* European toll-free number. If not accessible, call: +41 52 675 3777

Rev. 02.2018



Find more valuable resources at [TEK.COM](http://TEK.COM)

Copyright © Tektronix. All rights reserved. Tektronix products are covered by U.S. and foreign patents, issued and pending. Information in this publication supersedes that in all previously published material. Specification and price change privileges reserved. TEKTRONIX and TEK are registered trademarks of Tektronix, Inc. All other trade names referenced are the service marks, trademarks or registered trademarks of their respective companies.

112020 SBG 1KW-73768-0

