Tektronix[®]

Getting Started with C++ for Test Automation

GETTING STARTED GUIDE



Contents

What is a Programmatic Interface?	2
Installation and Prerequisites Overview	3
Visual Studio Community Edition	3
VISA	3
Example Code: Saving a Screenshot	4
Setting up the project	4
Linking VISA Libraries	7
Running the Example	9

What is a Programmatic Interface?

A programmatic interface (PI) is a boundary or set of boundaries between two computing systems that can be programmed to execute specific behaviors. For our purposes, it is the bridge between the computer that runs on every piece of Tektronix test equipment and the application written by the end user. To narrow this even further, it is a set of commands that can be sent remotely to an oscilloscope and the system on that oscilloscope that processes and executes them. The PI Stack (Figure 1) shows the flow of information from the host controller down to the instrument. The application code written by the end user defines the behavior of the target instrument. This is usually written in one of the development platforms popular in the industry, such as Python, MATLAB, LabVIEW, C++, or C#. This application will send SCPI (Standard Commands for Programmable Instrumentation) commands, which are the standard format for almost all test and measurement equipment, to the VISA layer. In some cases, the application will call a driver, which will then send one or more SCPI commands to the VISA layer.



Figure 1: The programmatic interface (PI) stack shows the flow of information between a host controller and instrument.

Installation and Prerequisites Overview

This guide will provide you with a source code file to build a C/C++ project with to connect to your oscilloscope to save a screenshot to its local storage and send a copy to your local computer.

The steps outlined in the guide will walk you through the basic installation of free tools, ending with the execution of the code to yield a screenshot specifically on our MSO 5-Series. This code is also compatible with the 2, 4, 5, 6-Series scopes.

A note on the script: We have included code that should be compatible with C as lower common denominator, surely you could do this easier with more modern C++, but we wanted to demonstrate with basic C where possible.

Please download or copy our script from the Tektronix GitHub: <u>Hard Copy Example</u>

We will go through the following:

- 1. IDE (Integrated Development Environment) Installation Visual Studio Community Edition – Free
 - a. Note: This guide will be using the VS compiler, not a true C compiler like mingw
- 2. VISA Installation
 - a. NI-VISA
- 3. Starting a project with Visual Studio
- 4. Linking VISA into the project
- 5. Running the example code

Visual Studio Community Edition

Install Visual Studio Community Edition

- 1. Download & Install Visual Studio Community Edition
 - a. Make sure you install the C/C++ Compiler from the Workloads tab.



Desktop development with C++

Build classic Windows-based applications using the power of the Visual C++ toolset, ATL, and optional features like MFC and...

VISA

Install a VISA Client

- 1. Download and install NI-VISA
 - a. Follow the steps outlined in our <u>"Getting Started Controlling Instruments with VISA</u>" guide for downloading and installing NI-VISA.

Example Code: Saving a Screenshot

Setting up the project

1. Open visual studio community edition and start a new project.



2. Select "Empty Project."

Create a new project		>	
create a new project		Search for templates (Alt+S)	
<u>R</u> ecent project templates		All languages + All platforms + All project types +	
🖏 Empty Project	C++	Empty Project Start from scratch with C++ for Windows. Provides no starting files.	
		Console App Run code in a Windows terminal. Prints "Hello World" by default. C++ Windows Console	
		CMake Project Build modern, cross-platform C++ apps that don't depend on .sln or .vcxproj files. C++ Windows Linux Console	
		***** Windows Desktop Wizard Create your own Windows app using a wizard. C++ Windows Desktop Console Library	
		Windows Desktop Application A project for an application with a graphical user interface that runs on Windows. C++ Windows	
		TypeScript Console Application A basic TypeScript Console application template which can be run with your local node installation.	
		<u>B</u> ack <u>N</u> ext	

And in the next screen, enter your project name and folder.

		- C	- X
Configure your new project			
Empty Project C++ Windows Console			
Project name			
Project1			
Location			
C:\myProjects\c++ test1 •			
Solution na <u>m</u> e 🕥			
✓ Place solution and project in the same <u>d</u> irectory			
Project will be created in "C:\myProjects\c++ test1\Project1\"			
	<u>B</u> ack	<u>C</u> reat	te
	Dack	<u>C</u> reat	2

3. If you do not see the solutions explorer on the right, you can display it with CTRL+ALT+L.

4. Download and import source.c you got from <u>the Tektronix GitHub</u> into your project by either 'right clicking on the project name > Add > Existing Item' and browse to the included source.c file. You can also use the keyboard shortcut SHIFT+ALT+A. This step will also create the 'Source Files' folder if it is not present by default.

File Edit View Git Pro	oject Build Debug Test	Analyze Tools Extensions Window	Help 🖉 🖉 Search 🕶	Project1				GM	
i 🛛 - 🕘 🎁 - 🖆 🗎 目 り	- 🤍 - Debug - x64	🔹 🕨 Local Windows Debugger 🔸	· Þ 🧉 - 🛛 📭 🖓 📮						🖻 Live Share 🖉
								Solution Explorer	- û ×
								Search Solution Explorer (Ctrl+;)	-م
								Solution 'Project1' (1 of 1 project	
								🔺 🖭 Project1	
								▷ □ References	
								External Dependencies Header Files	
	刘 Add Existing Item - Projec						× Resource Files		
	$\leftarrow \rightarrow \cdot \uparrow$. This	s PC > Windows (C:) > myProjects > c+++	test1 → Project1 ~					Source Files	
	Organize 👻 New folde				I ∷ - I	. (2)			
		Name	Date modified	Туре	Size				
	📌 Quick access		10/5/2023 9:51 PM	File folder					
	a testap of	Project1.sln	10/5/2023 9:51 PM	Visual Studio Solu	2 KB				
	A STATUTE OF	Project1.vcxproj	10/5/2023 9:51 PM	VC++ Project	7 KB				
	- Caracter of the	Project1.vcxproj.filters	10/5/2023 9:51 PM	VC++ Project Filte	1 KB				
	Cathor Inter	Project1.vcxproj.user	10/5/2023 9:51 PM	Per-User Project O	1 KB				
	Server at 1	source_riddick.c	10/5/2023 5:00 PM	C Source	3 KB				
	Conception of the								
	And and a second second								
	Table (e)								
	Million (1966								
	- interest								
	📢 Microsoft Visual S 🗸								
	File <u>n</u> a	 ✓ All Files (*.*)	~					
Output				Add	Car	ncel	- 4 ×		
Show output from:		• = = = # 0		100 A					
Show output from:									
💭 Ready								↑ Add to Source Control 🔺 🕅 Sel	ect Repository 🔺 🚨 🔐

5. Open the source.c file. You should see many errors since the VISA libraries are not linked yet.



Linking VISA Libraries

You should see numerous errors like 'cannot find Source file,' 'identifier undefined,' etc.

- 1. Right click on the Project Name and navigate to properties.
- 2. In properties, navigate to Linker > Input > Additional Dependencies, drop down option> Edit and add the path to your NI Visa Library (default: C:\Program Files\IVI Foundation\VISA\Win64\Lib_x64\msc\nivisa64.lib)



3. In the same project properties window, go to C/C++ > General > Additional Include Directories> Add a new line > Browse to your NI VISA headers folder (default: C:\Program Files\IVI Foundation\VISA\Win64\Include)

If you do not see C/C++ tab, you need to make sure you have a .c file opened in the Visual Studio.



4. Hit Apply / Ok to exit out of the properties window.

5. The errors/syntax highlighting will be cleared now.



Running the Example

- 1. Edit the following lines according to your instrument.
 - Line 15: Resource String. Enter your instrument's IP address.
 - Line 19: Target path where the screenshot will be saved. Please ensure this path/folder exists on your local computer.
 - Line 20: This will be the filename of the saved screenshot on your local computer.
 - Line 60: is a direct hardcode to the scope, make sure this directory exists.
- 2. When ready, compile and run your code with CTRL + F5 (Or press the open green triangle).

3. If your code compiles correctly your code should run and end with the code 0(0x0) message.

Eile Edit	<u>V</u> iew <u>G</u> it <u>P</u> roject <u>B</u> uild <u>D</u> ebug Test A <u>n</u> alyze <u>T</u> ools Extensions <u>Wi</u> ndow <u>H</u> elp P Search + Project1		GM	– 🗆 ×
i 🕒 - 🗇 i 👘	・醫 🔠 🧐 「ク・ペー」 Debug 🔹 x64 🔹 → Local Windows Debugger + ▷ 🖉 - 「節 罰 🕫 ♥ ‱ 備 遭 階 Д 役 役 役 💡			🕼 Live Share 🔗
source riddick.c	+ X	- 4	Solution Explorer	• # x ₽
E Project1	 (Global Scope) (Sinthe Cope) 	- +		ignos
7			Search Solution Explorer (Ctrl+;)	tic To
8 🖂	int main() {		Solution 'Project1' (1 of 1 project)	500
10	ViStatus status;	10	🔺 🖭 Project1	
11	ViSession defaultRM, instr; ViUInt32 retCount;		▷ 0-□ References	
12	Vicharsz Hetcourte,		External Dependencies Header Files	
14			Resource Files	
15 16	<pre>char resourceString[] = "TCPIP::192.168.1.222::INSTR"; ViBuf scpi;</pre>		▲ Source Files	
17	vauu sepa,		Isource_riddick.c	
18	// Where to save (PC side)			
19 20	<pre>char target_path[] = "C:\\tempx\\"; char target_name[] = "scope_screenshot.png";</pre>			
20	char cutgec_nemet_i - scope_screenshot.png , char full_path[256];			
22				
23 24	// Initialize VISA session status = viOpenDefaultRM(&defaultRM);;			
24 25 🗉	if (status < VL_SUCCESS) {			
26	<pre>printf("Failed to initialize\n");</pre>			
27	return -1;			
28 29				
30	// Open instrument connection			
31	<pre>status = viOpen(defaultRM, resourceString, VI_NULL, VI_NULL, &instr);</pre>			
32 🖻 33	<pre>if (status < VI_SUCCESS) { printf("Error connecting to instrument\n");</pre>			
33	vistatusDesc(defaultM, status, buffer);			
35	<pre>printf("%s\n", buffer);</pre>			
36	return 0;			
37 38				
39				
100 % 🝷	S 0 ▲ 1 ↑ ↓ 🕉 + 4 ▶ Ln: 22 Ch: 5	TABS CRLF		
Output		- ₽ ×		
Show output from	n: Debug 🔹 🖕 🚔 🚔 🕮 🕲			
	x9898 has exited with code 0 (0x0).			
	xafc8 has exited with code 0 (0x0). xd5fc has exited with code 0 (0x0).			
The thread 0x	xbc88 has exited with code 0 (0x0).			
	xb9a0 has exited with code 0 (0x0). '[67680] Project1.exe' has exited with code 0 (0x0).			
rite program				
4		Þ		
💭 Item(s) Saved			↑ Add to Source Control 🔺 🔟 Select	Repository 🔺 🚨 🔬

A note on repeat runs:

- You will need to delete the generated files on the scope and the local copy. Alternatively, you can build in some code to add the date and time to the filename to make them unique each run.
- You could also implement changes to increment the file names instead of hardcoding them in this quick example.

With this guide we have covered the basics for a small automation task with C/C++. The linking of VISA libraries is one of the most challenging pieces to getting C based languages working with automation.

If you found this useful, you might also want to check out our other language guides on <u>www.tek.com/testautomation</u>.

Contact Information:

Australia 1800 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) 3530-8901 Canada 1800 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 1800 22 55835 Mexico, Central/South America and Caribbean 52 (55) 88 69 35 25 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)26566688 Thailand 1800 011 931 United Kingdom / Ireland* 00800 2255 4835 USA 1800 833 9200 Vietnam 12060128

> * European toll-free number. If not accessible, call: +41 52 675 3777 Rev. 02.2022



Find more valuable resources at <u>TEK.COM</u>

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. 121223 SBG 46W-74051-0