



GRL Platform Solutions

Python Programming API User Guide For Using the GRL 5-Port Switch Board Test Fixture (GRL-USB-PD-MULT)

Published on 25 June 2026

DISCLAIMER

This document is provided “as is” with no warranties whatsoever, including any warranty of merchantability, no infringement, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. The GRL Platform Solutions disclaims all liability for infringement of proprietary rights, relating to use of information in this specification. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted herein.

All product names are trademarks, registered trademarks, or service marks of their respective owners.



Copyright © 2026 GRL Platform Solutions. All rights reserved.

TABLE OF CONTENTS

1	SCOPE OF THIS USER GUIDE	5
2	GETTING STARTED WITH GRL-USB-PD-MULT PYTHON APIS	6
2.1	PYTHON SOFTWARE INSTALLATION	6
2.2	HARDWARE CONNECTION SETUP	8
2.2.1	Connect Ethernet Cable and Turn On Test Controller	9
2.2.2	Verify Test Controller Ethernet Installation.....	9
2.2.3	Connect GRL-USB-PD-MULT Board to Test Controller and Control Computer 11	
2.3	USING GRL-USB-PD-MULT PYTHON AUTOMATION CONTROL SCRIPTS.....	12
2.3.1	Running MainProgram.py Script.....	12
2.3.2	Running C2MultiDUT_Automation.py Script.....	15
2.3.3	Running JSON Script	16

REVISION RECORD

Version	Date	Description of Changes	Author(s)
1.0	01/2020	New User Guide Creation	Pooja SM (GRL) pooja@graniteriverlabs.in Ong Gaik Pheng (GRL Platform Solutions) gpong@grlps.com
2.0	06/2026	User Guide Rebranding to GRL Platform Solutions	Ong Gaik Pheng (GRL Platform Solutions) gpong@grlps.com

1 Scope of this User Guide

This User Guide helps you get started with Python programming APIs for the GRL Platform Solutions 5-Port Switch Board (GRL-USB-PD-MULT option).

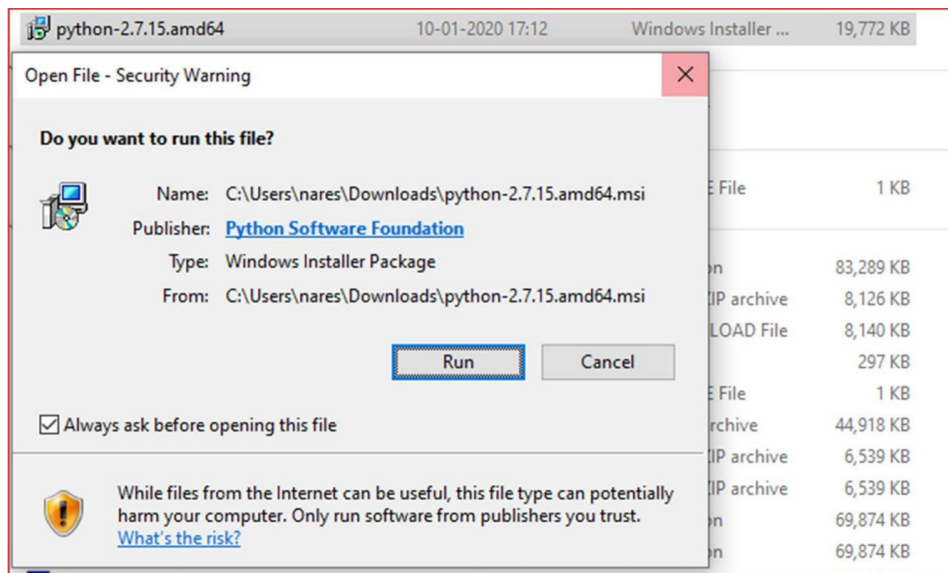
The GRL-USB-PD-MULT option is provided as a separate test fixture accessory for the GRL-USB-PD-C2-EPR test controller. The GRL-USB-PD-MULT option consists of a 5- port switch extension fixture that is plugged in to the GRL-USB-PD-C2-EPR test controller and used to connect up to five Unit Under Tests (UUTs) to perform switching during tests. For more information on the GRL-USB-PD-MULT option, please contact support@grlps.com.

2 Getting Started with GRL-USB-PD-MULT Python APIs

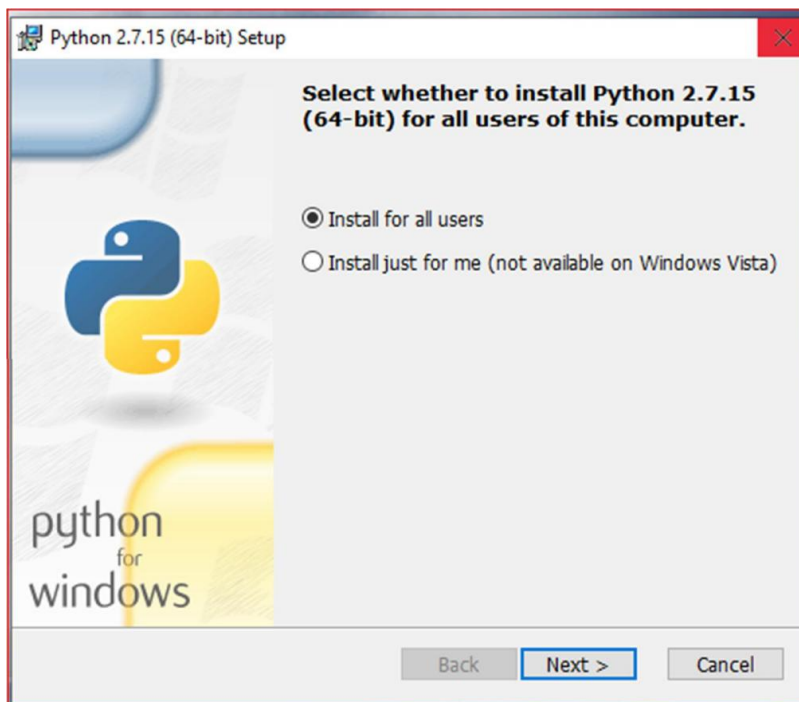
This section explains how to install and use the following Python APIs to create test cases to meet more customized test requirements.

2.1 Python Software Installation

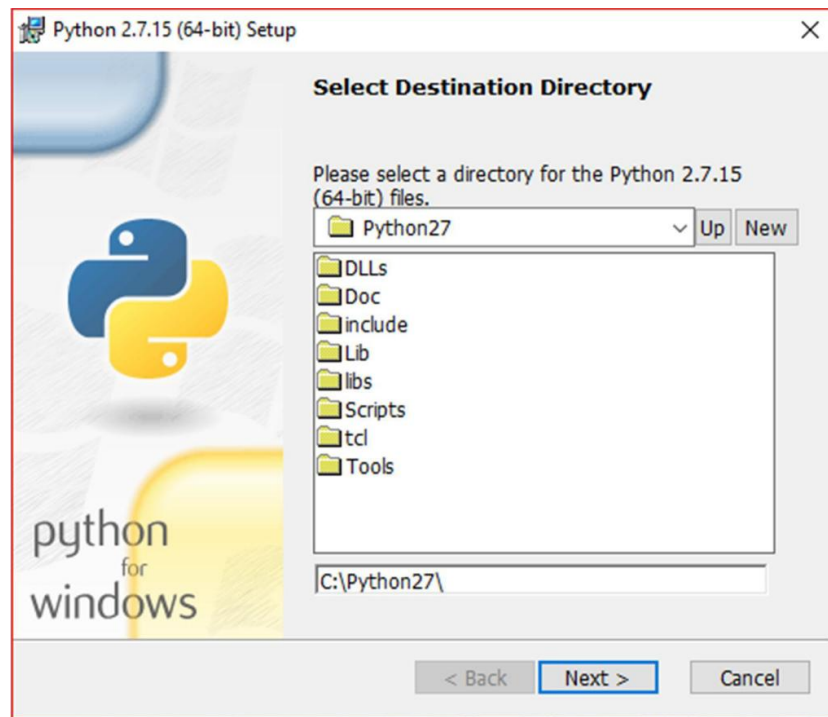
1. Download the Python software version 2.7.15 and install it on your PC.



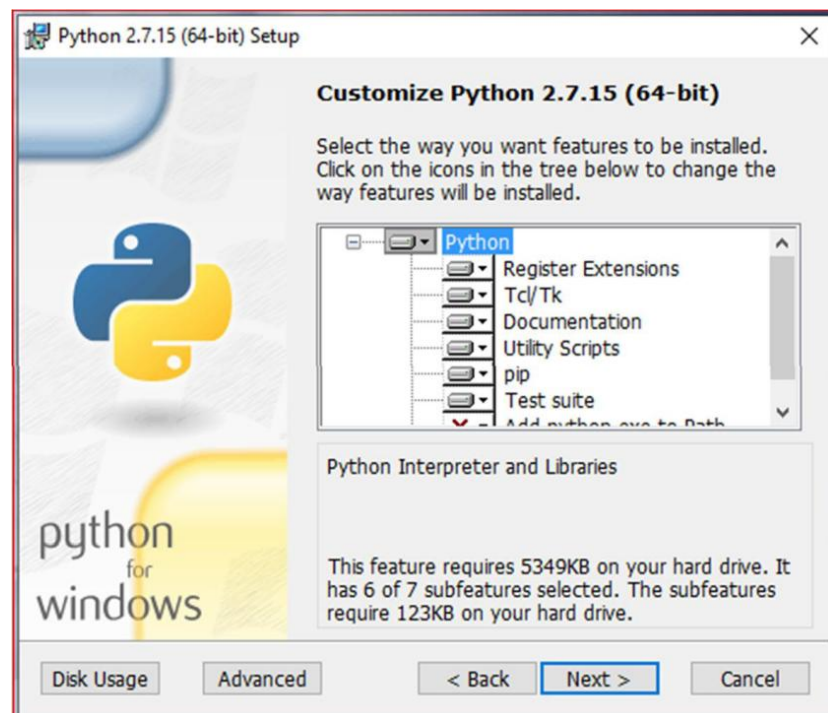
2. Click **Next** to install the Python software.



3. Select the directory for the Python files and then click **Next**.



4. Set up the libraries for the Python features and then click **Next**.



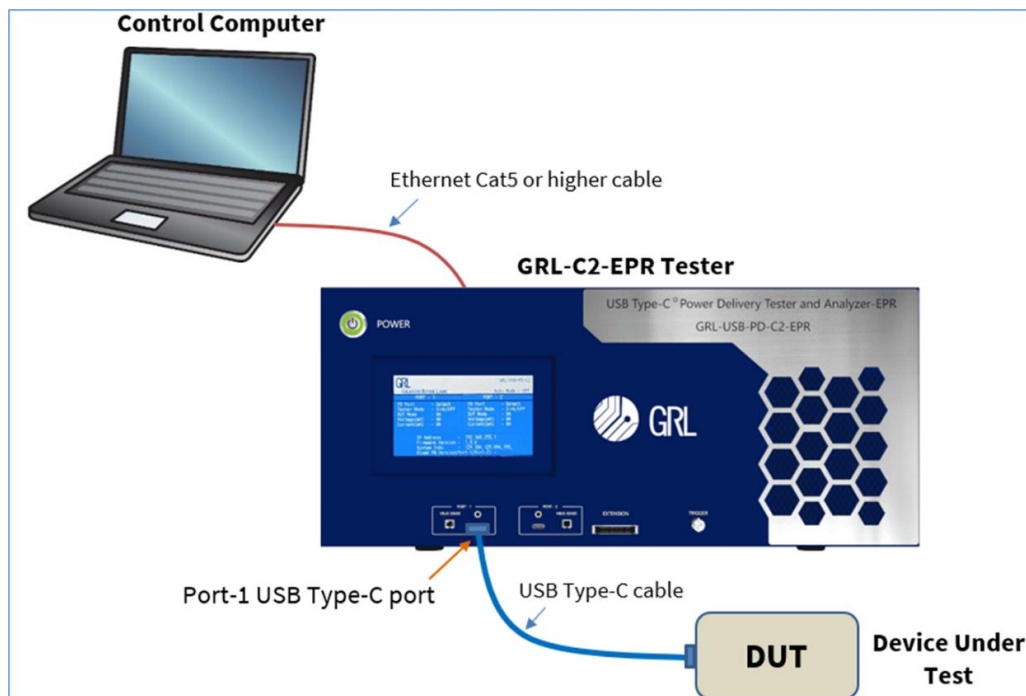
5. Follow the remaining instructions to complete the Python installation.
6. Once installation has completed, open the GRL-USB-PD-C2-EPR remote API Python scripts from C:\scripts\Python\3. Python Script\C2_RemoteAPIScript.

Windows (C:) > scripts > Python > 3. Python Script > C2_RemoteAPIScript

Name	Date modified	Type	Size
.vs	10-01-2019 19:13	File folder	
.vscode	10-01-2019 19:13	File folder	
__pycache__	10-01-2019 19:13	File folder	
Python27	10-01-2019 19:13	File folder	
C2_RemoteAPIScript	25-10-2018 16:27	Python Project	3 KB
C2_RemoteAPIScript	18-06-2018 07:55	Microsoft Visual S...	1 KB
C2MultiDUT_Automation	04-12-2018 18:05	Python File	11 KB
C2MultiDUT_Automation	04-12-2018 18:10	Compiled Python ...	6 KB
C2TestAutomationModule	01-12-2018 19:13	Python File	30 KB
C2TestAutomationModule	02-12-2018 14:57	Compiled Python ...	22 KB
C2TestExecutionModule	01-12-2018 19:06	Python File	5 KB
C2TestExecutionModule	01-12-2018 19:06	Compiled Python ...	3 KB
cmd.exe	25-10-2018 12:46	Shortcut	2 KB
GlobalVars	25-10-2018 19:57	Python File	1 KB
GlobalVars	13-11-2018 17:52	Compiled Python ...	1 KB
GrlPdAppCli	01-12-2018 18:10	Python File	7 KB
GrlPdAppCli	01-12-2018 18:13	Compiled Python ...	4 KB
Json	04-12-2018 18:09	JSON File	11 KB
MainProgram	04-12-2018 18:08	Python File	10 KB
SharedEnums	06-08-2018 23:19	Python File	1 KB
SharedEnums	22-10-2018 18:53	Compiled Python ...	1 KB
TestcaseDetail	01-12-2018 18:10	Python File	2 KB
TestcaseDetail	01-12-2018 18:13	Compiled Python ...	3 KB
TestScriptCable	01-12-2018 18:10	Python File	4 KB

2.2 Hardware Connection Setup

Figure below shows an example hardware setup for testing a USB Power Delivery Provider, Consumer or Dual Role Powered device, or USB Type-C hub or cable using the Python APIs running on a control computer and connected via Ethernet to the GRL-USB-PD-C2-EPR test controller that is attached to the USB device to be tested through one of the two controller USB Type-C test ports.



Below is the procedure for connecting the hardware and verifying proper hardware connections.

1. Connect a power supply to the GRL-USB-PD-C2-EPR test controller power interface using the 24 V, 280 W power brick included with the controller.
2. Connect the GRL-USB-PD-C2-EPR test controller to the control computer using a physical Ethernet connection.

Note: Automation of the UUT power supply switching in the GRL-USB-PD-C2-EPR test controller is handled internally to the controller. Thus, there is no Ethernet, USB or GPIB connection attached to the power supply.

2.2.1 Connect Ethernet Cable and Turn On Test Controller

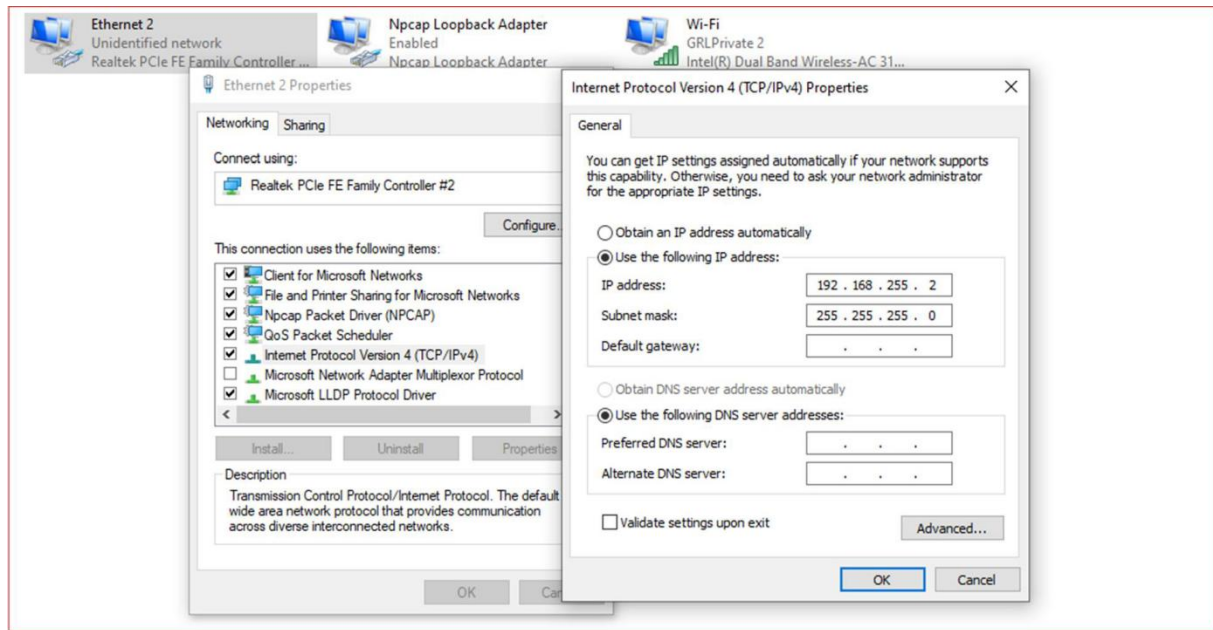
1. Connect the Ethernet (RJ-45) connector to one of the control computer's Ethernet ports. A USB to Ethernet adapter can be used if there are no native Ethernet ports on the control computer.
2. Turn on the GRL-USB-PD-C2-EPR test controller using the Power button on the front of the controller.



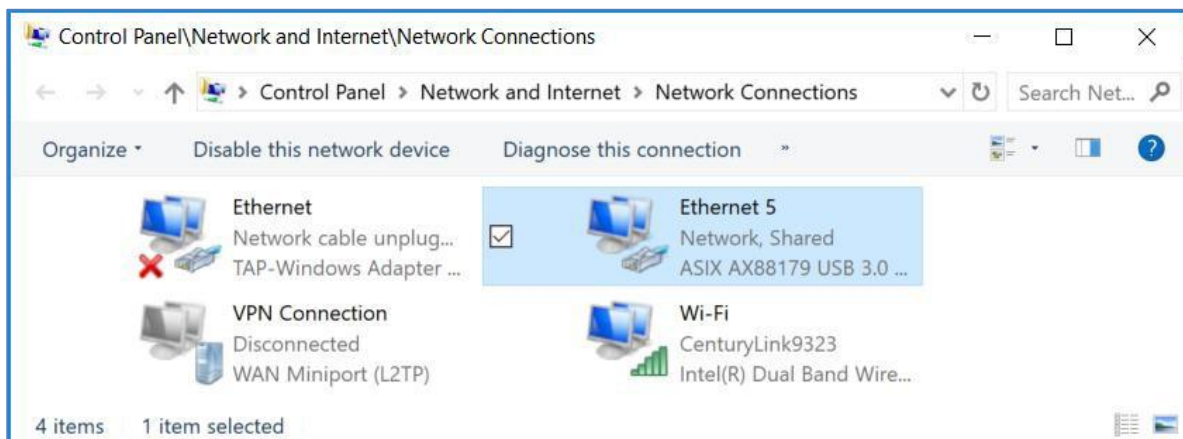
2.2.2 Verify Test Controller Ethernet Installation

The Ethernet port on the control computer needs to be configured correctly for the GRL-USB-PD-C2-EPR test controller to recognize the control computer and vice versa.

To make sure the network connection is set up correctly, open the Network Connections panel from the control panel and set up the Ethernet properties as shown below:



Click on the 'OK' button on the Internet Protocol Properties and close the Ethernet Properties. Make sure the GRL-USB-PD-C2-EPR test controller is powered on and completely booted up (front panel display showing firmware version number), then connect the Ethernet cable from the GRL-USB-PD-C2-EPR test controller to the control computer's Ethernet port that was just set up. The network connections panel should now look as shown below:



The GRL-USB-PD-C2-EPR test controller is now set up and ready for use.

2.2.3 Connect GRL-USB-PD-MULT Board to Test Controller and Control Computer

The GRL-USB-PD-MULT board contains five USB Type-C ports that can be used to connect and test five UUTs at one time (one after another) or a single port UUT.



1. Connect the GRL-USB-PD-MULT board to one of the test ports on the GRL-USB-PD-C2-EPR test controller as shown above.
2. Connect a Mini-B USB cable from the GRL-USB-PD-MULT board to the control computer.
3. Make sure the Ethernet cable has been connected from the GRL-USB-PD-C2-EPR test controller to the control computer's Ethernet port (as described in previous sections) before running the five-port/single port switch function.

2.3 Using GRL-USB-PD-MULT Python Automation Control Scripts

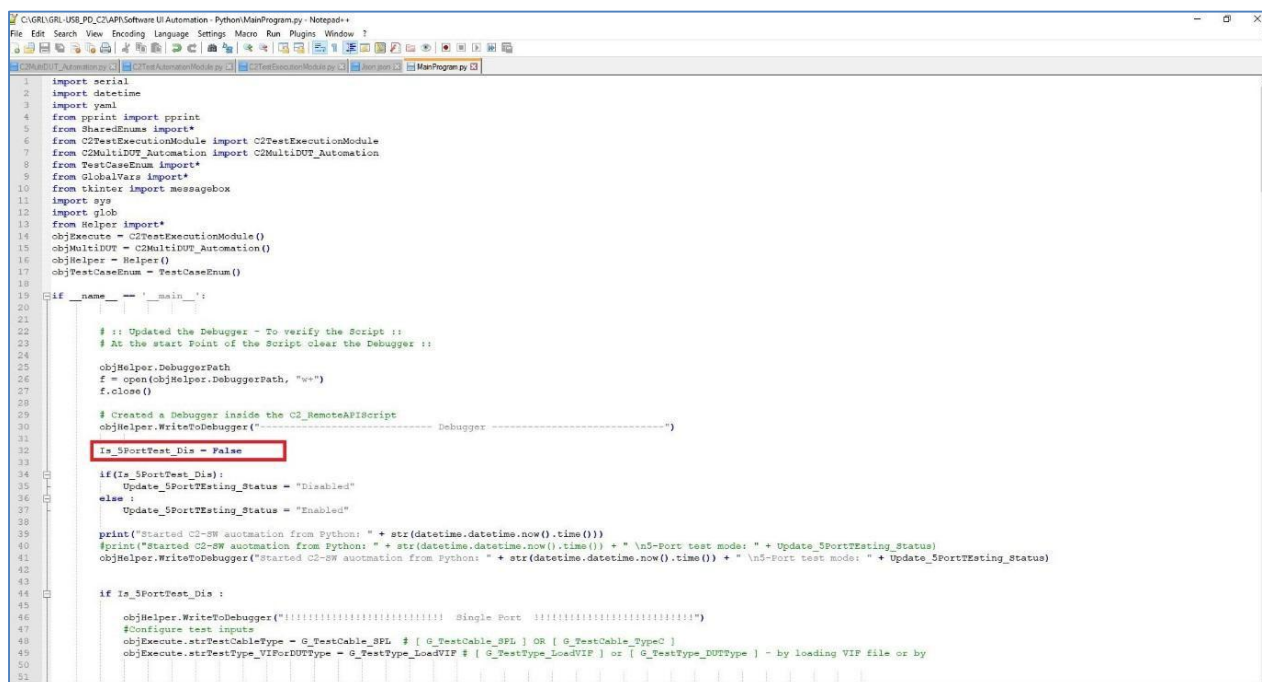
The GRL-USB-PD-MULT Python test scripts allow automation control for single port and five-port testing with the GRL Platform Solutions 5-Port Switch Board.

2.3.1 Running MainProgram.py Script

The MainProgram.py script is used to run scripts with the requirements given in the respective fields.

- Go to “C:\scripts\Python\3. Python Script\C2_RemoteAPIScript\MainProgram.py” and open **MainProgram.py** in Notepad++.

*Note: Single port testing is enabled by default in the MainProgram.py script; to enable five-port testing, set the `Is_5PortTest_Dis` field code to **False** as shown below:*



```

1  import serial
2  import datetime
3  import yaml
4  from pprint import pprint
5  from SharedEnums import*
6  from C2TestExecutionModule import C2TestExecutionModule
7  from C2MultiDUT_Automation import C2MultiDUT_Automation
8  from TestCaseEnum import*
9  from GlobalVars import*
10 from tkinter import messagebox
11 import sys
12 import glob
13 from Helper import*
14 objExecute = C2TestExecutionModule()
15 objMultiDUT = C2MultiDUT_Automation()
16 objHelper = Helper()
17 objTestCaseEnum = TestCaseEnum()
18
19 if __name__ == '__main__':
20
21     # :: Updated the Debugger - To verify the Script ::
22     # At the start Point of the Script clear the Debugger ::
23
24     objHelper.DebuggerPath
25     f = open(objHelper.DebuggerPath, "w+")
26     f.close()
27
28     # Created a Debugger inside the C2_RemoteAPIScript
29     objHelper.WriteToDebugger("----- Debugger -----")
30
31     Is_5PortTest_Dis = False
32
33     if(Is_5PortTest_Dis):
34         Update_5PortTesting_Status = "Disabled"
35     else:
36         Update_5PortTesting_Status = "Enabled"
37
38     print("Started C2-SW automation from Python: " + str(datetime.datetime.now().time()))
39     print("Started C2-SW automation from Python: " + str(datetime.datetime.now().time()) + " \n5-Port test mode: " + Update_5PortTesting_Status)
40     objHelper.WriteToDebugger("Started C2-SW automation from Python: " + str(datetime.datetime.now().time()) + " \n5-Port test mode: " + Update_5PortTesting_Status)
41
42
43     if Is_5PortTest_Dis:
44
45         objHelper.WriteToDebugger("!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Single Port !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!")
46         #Configure test inputs
47         objExecute.strTestCableType = G_TestCable_SPL # [ G_TestCable_SPL ] OR [ G_TestCable_TypeC ]
48         objExecute.strTestType_VIForDUTType = G_TestType_LoadVIP # [ G_TestType_LoadVIP ] or [ G_TestType_DUTType ] - by loading VIP file or by
49
50
51

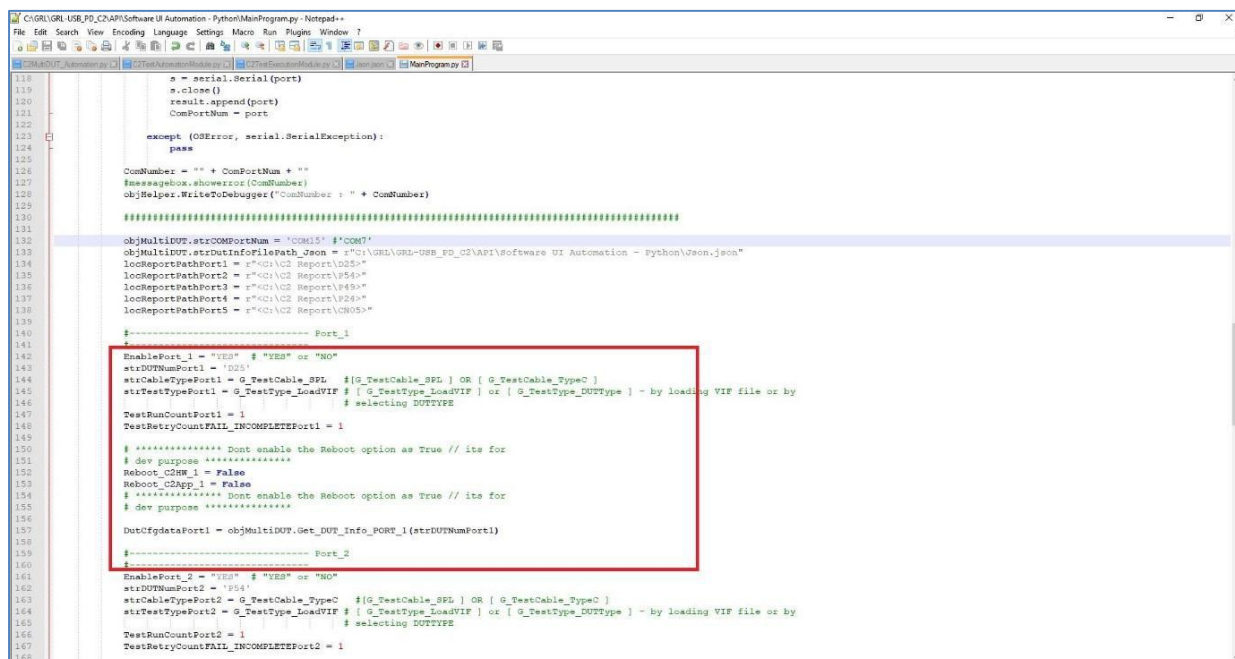
```

- Change to the correct COM Port Number according to the Cable connected to the control computer’s port from the GRL 5-Port Switch Board fixture.

- Define the test report file directory as shown in below example based on the location that you want to save the reports. The JSON file directory should also refer to where the JSON file is located.

```
C:\GRL\GRU_USB_F0_G2API\Software UI Automation - Python\MainProgram.py - Notepad++  
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?  
  
C:\TestAutomation\Software UI Automation - Python\MainProgram.py C:\TestAutomation\Software UI Automation - Python\MainProgram.py  
118 s = serial.Serial(port)  
119 s.close()  
120 result.append(port)  
121 ComPortNum = port  
122  
123 except (OSError, serial.SerialException):  
124     pass  
125  
126 ComNumber = "" + ComPortNum + "  
127 fmessagebox.showerror(ComNumber)  
128 objHelpes.WriteToDebugger("ComNumber : " + ComNumber)  
129  
130  
131  
132 objMultiDUT.strCOMPortNum = 'COM15' #'COM7'  
133 objMultiDUT.strDeviceInfoFilePath_Jeon = "C:\\GRL\\GRU_USB_F0_G2API\\Software UI Automation - Python\\Jeon.json"  
134 locReportPathPort1 = r"C:\\G2 Report\\J025"  
135 locReportPathPort2 = r"C:\\G2 Report\\P54">  
136 locReportPathPort3 = r"C:\\G2 Report\\P49">  
137 locReportPathPort4 = r"C:\\G2 Report\\P56">  
138 locReportPathPort5 = r"C:\\G2 Report\\CN053"  
139  
140  
141 ----- Port_1  
142 EnablePort_1 = "YES" # "YES" or "NO"  
143 strDUTNumPort1 = 'D25'  
144 strCableTypePort1 = G_TestCable_SPL #[G_TestCable_SPL] OR [ G_TestCable_TypeC ]  
145 strTestTypePort1 = G_TestType_LoadVIF # [ G_TestType_LoadVIF ] or [ G_TestType_DUTType ] - by loading VIF file or by  
146                                     # selecting DUTTYPE  
147 TestRunCountPort1 = 1  
148 TestRetryCountFAIL_INCOMPLETEPort1 = 1  
149  
150 # ***** Dont enable the Reboot option as True // its for  
151 # dev purpose *****  
152 Reboot_CDRW_1 = False  
153 Reboot_CHApp_1 = False  
154 # ***** Dont enable the Reboot option as True // its for  
155 # dev purpose *****  
156  
157 DutCfgDataPort1 = objMultiDUT.Get_DUT_Info_PORT_1(strDUTNumPort1)  
158  
159 ----- Port_2  
160  
161 EnablePort_2 = "YES" # "YES" or "NO"  
162 strDUTNumPort2 = 'P54'  
163 strCableTypePort2 = G_TestCable_TypeC #[G_TestCable_SPL] OR [ G_TestCable_TypeC ]  
164 strTestTypePort2 = G_TestType_LoadVIF # [ G_TestType_LoadVIF ] or [ G_TestType_DUTType ] - by loading VIF file or by  
165                                     # selecting DUTTYPE  
166 TestRunCountPort2 = 1  
167 TestRetryCountFAIL_INCOMPLETEPort2 = 1  
168
```

- Based on the DUT Type and Cables used, the following fields need to be modified for each Port (using Port 1 as example):
 - Set the EnablePort_1 field code to **“YES”** to run the UUT connected to Port 1.
 - Modify the UUT’s TID name in the strDUTNumPort1 field code according to the respective JSON file.
 - If the UUT is connected using the GRL-SPL-EPR cable, use the strCableTypePort1 = G_TestCable_SPL field code. If using the Captive cable, use the strCableTypePort1 = G_TestCable_TypeC field code.
 - If a VIF file is available for the UUT, use the strTestTypePort1 = G_TestType_LoadVIF field code to load the VIF file. If a VIF file is not available, use the strTestTypePort1 = G_TestType_DUTType field code to define the device type.
 - The TestRunCountPort1 field code allows you to define the number of times to repeat running tests while the TestRetryCountFAIL_INCOMPLETEPort1 field code allows you to set the number of times to repeat running the Fail and Incomplete test cases.
 - If you want to reboot the GRL-USB-PD-C2-EPR test controller and application for each Port, set the Reboot_C2HW_1 and Reboot_C2APP_1 codes to **“True”** or otherwise set to **“False”**.
- Similarly follow the procedure as above for other Ports.



```

118 s = serial.Serial(port)
119 s.close()
120 result.append(port)
121 ComPortNum = port
122
123 except (OSError, serial.SerialException):
124     pass
125
126 ComNumber = "" + ComPortNum + ""
127 ErrorMessage.ShowError(ComNumber)
128 objHelper.WriteToDebugger("ComNumber : " + ComNumber)
129
130 #####
131
132 objMultiDUT.strCOMPortNum = "COM15" # "COM7"
133 objMultiDUT.strDeviceInfoFilePath_Json = r"C:\GRL\GRL-USB-PD-C2-API\Software UI Automation - python\Json.Json"
134 locReportPathPort1 = r"C:\V2 Report\005"
135 locReportPathPort2 = r"C:\V2 Report\054"
136 locReportPathPort3 = r"C:\V2 Report\049"
137 locReportPathPort4 = r"C:\V2 Report\029"
138 locReportPathPort5 = r"C:\V2 Report\005"
139
140 #----- Port_1
141 #####
142 EnablePort_1 = "YES" # "YES" or "NO"
143 strDUTNumPort1 = "D05"
144 strCableTypePort1 = G_TestCable_SPL if [G_TestCable_SPL] or [G_TestCable_TypeC]
145 strTestTypePort1 = G_TestType_LoadVIF if [G_TestType_LoadVIF] or [G_TestType_DUTType] - by loading VIF file or by
146 # selecting DUTTYPE
147 TestRunCountPort1 = 1
148 TestRetryCountFAIL_INCOMPLETEPort1 = 1
149
150 # ***** Dont enable the Reboot option as True // its for
151 # dev purpose *****
152 Reboot_C2HW_1 = False
153 Reboot_C2App_1 = False
154 # ***** Dont enable the Reboot option as True // its for
155 # dev purpose *****
156
157 DutCfgdataPort1 = objMultiDUT.Get_DUT_Info_PORT_1(strDUTNumPort1)
158
159 #----- Port_2
160 #####
161 EnablePort_2 = "YES" # "YES" or "NO"
162 strDUTNumPort2 = "D04"
163 strCableTypePort2 = G_TestCable_TypeC if [G_TestCable_SPL] or [G_TestCable_TypeC]
164 strTestTypePort2 = G_TestType_LoadVIF if [G_TestType_LoadVIF] or [G_TestType_DUTType] - by loading VIF file or by
165 # selecting DUTTYPE
166 TestRunCountPort2 = 1
167 TestRetryCountFAIL_INCOMPLETEPort2 = 1
168
  
```

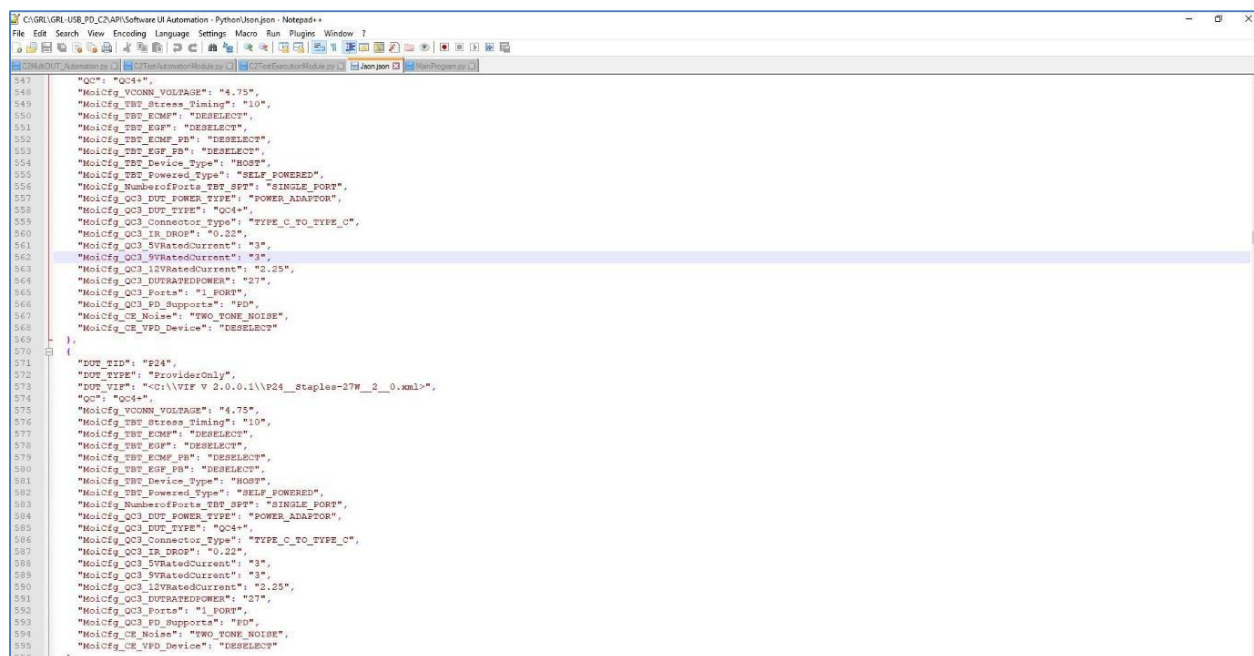
Page.15

2.3.3 Running JSON Script

The JSON script contains all VIF file information and is used to call the VIF file from the given directory to the MainProgram.py script.

- Go to the “C:\scripts\Python\3. Python Script\C2_RemoteAPIScript\Json.json” and open **Json.json** in Notepad++.
- You will need to modify the VIF information based on the UUT before running tests.
- Modify required fields in the JSON file, for example DUT_VIF, DUT_TID and DUT_TYPE.
- Make sure to specifically define the DUT_TID and DUT_TYPE field codes based on the UUT.
- Specify the VIF file directory where the VIF files are stored using the DUT_VIF field code. Make sure the JSON script has VIF file directories specified for all UUTs.

Note: Use “\\” instead of “\” and “_” instead of spacing when entering the VIF file directory.



```

547 {
548   "QC": "QC4",
549   "MoIcfig_VCONN_VOLTAGE": "4.75",
550   "MoIcfig_TBT_Stream_Timing": "10",
551   "MoIcfig_TBT_ECOMP": "DESELECT",
552   "MoIcfig_TBT_ECOMP_FB": "DESELECT",
553   "MoIcfig_TBT_ESP_FB": "DESELECT",
554   "MoIcfig_TBT_Device_Type": "HOST",
555   "MoIcfig_TBT_Powered_Type": "SELF POWERED",
556   "MoIcfig_NumberOfPorts_TBT_SFP": "SINGLE_PORT",
557   "MoIcfig_QC3_DUT_POWER_TYPE": "POWER ADAPTOR",
558   "MoIcfig_QC3_DUT_TYPE": "QC4",
559   "MoIcfig_QC3_Connector_Type": "TYPE_C_TO_TYPE_C",
560   "MoIcfig_QC3_IR_DROP": "0.22",
561   "MoIcfig_QC3_VRatedCurrent": "3",
562   "MoIcfig_QC3_VRatedCurrent": "2.25",
563   "MoIcfig_QC3_OTRATEDPOWER": "27",
564   "MoIcfig_QC3_Ports": "1_PORT",
565   "MoIcfig_QC3_PD_Supports": "PD",
566   "MoIcfig_QC_Noise": "TWO_TONE_NOISE",
567   "MoIcfig_QC_VFD_Device": "DESELECT"
568 },
569 {
570   "DUT_TID": "P24",
571   "DUT_TYPE": "ProviderOnly",
572   "DUT_VIF": "C:\\VIF V 2.0.0.1\\P24_Staples-27W_2_0.xml",
573   "QC": "QC4",
574   "MoIcfig_VCONN_VOLTAGE": "4.75",
575   "MoIcfig_TBT_Stream_Timing": "10",
576   "MoIcfig_TBT_ECOMP": "DESELECT",
577   "MoIcfig_TBT_ECOMP_FB": "DESELECT",
578   "MoIcfig_TBT_ESP_FB": "DESELECT",
579   "MoIcfig_TBT_Device_Type": "HOST",
580   "MoIcfig_TBT_Powered_Type": "SELF POWERED",
581   "MoIcfig_NumberOfPorts_TBT_SFP": "SINGLE_PORT",
582   "MoIcfig_QC3_DUT_POWER_TYPE": "POWER ADAPTOR",
583   "MoIcfig_QC3_DUT_TYPE": "QC4",
584   "MoIcfig_QC3_Connector_Type": "TYPE_C_TO_TYPE_C",
585   "MoIcfig_QC3_IR_DROP": "0.22",
586   "MoIcfig_QC3_VRatedCurrent": "3",
587   "MoIcfig_QC3_VRatedCurrent": "2.25",
588   "MoIcfig_QC3_OTRATEDPOWER": "27",
589   "MoIcfig_QC3_Ports": "1_PORT",
590   "MoIcfig_QC3_PD_Supports": "PD",
591   "MoIcfig_QC_Noise": "TWO_TONE_NOISE",
592   "MoIcfig_QC_VFD_Device": "DESELECT"
593 }
594 }
595

```

END_OF_DOCUMENT