OpenQASM Debugger Visual Studio Extension Documentation

V0.7 09/04/2024



© Quantag IT Solutions GmbH 2024.



Contents

Installation

Extension can be downloaded and installed from Microsoft Visual Studio Marketplace https://marketplace.visualstudio.com/items?itemName=QuantagITSolutionsGmbH.openqasmdebug

Description

This VS Code extension connects to QSCore (Quantag Studio Core) component which runs instance of Quantum Virtual Machine (QVM) using DAP (Microsoft Adapter Debugger) protocol and allows:

- debug Quantum circuits written on OpenQASM or Python using <u>Qiskit</u> or <u>PyTKET</u> frameworks.
- perform step debugging on remote Quantum Virtual Machine
- see quantum state on every step
- see compiled OpenQASM code for python circuits in Disassembly View
- see visualized Quantum Circuits directly in VS Code and on web frontend.



The main idea is to establish a quantum development environment in a way like a classic software development environment with sources, assembler code, and direct hardware execution possibility.

How does it work?

After installing extension from Marketplace, it registers itself for .py and .qasm files. *There can be other extensions also installed and registered for same file types*. They can work together; just be sure you choose the right menu items and command for every extension.

Before any action sources from local workplace must be submitted to cloud server where QVM working. To do it we do not use SSH Tunnels which is common way of working with remote file system from VS code, instead we submit all workplace files to server using own microservice. All files submitted after opening new Workplace to avoid delays when browsing files. So be careful with private data, do not open private data with activated extension.

When a user chooses '**Debug circuit**' with python file – source is analyzed on server and *updated* to add OpenQASM-export and rendering circuit commands depends on detected framework.

When choosing '**Debug circuit'** on OpenQASM file – this file imported to Qiskit on server using import-qasm commands and rendered to circuit image which later can be opened using command palette with command 'Render OpenQASM/Qiskit circuit'.

For TKET python sources – circuit generated to HTML and opened in external browser using 'Render TKET circuit.'

Rendered circuits and compiled OpenQASM files transferred from server to VS code using dedicated microservice.

Launching circuit on remote hardware implemented using adding commands to 'Command Palette'.

Hardware configuration for launching taken from workplace root file 'config.json' which contains all information required for launch.

QSCore uses DAP protocol to communicate with VS Code, so it can work with any IDE which supports DAP, for example Microsoft Visual Studio, Eclipse or Emacs. Full list of supported IDEs can be found here: https://microsoft.github.io/debug-adapter-protocol/implementors/tools/



QSCore

QSCore (Quantag Studio Core) is key component which launch instance of QVM and communicates with IDE and web frontend.

It is written in C++ and uses CMake. It is tested on Windows and Linux. It can be run locally or in the cloud.

By default extension connects to QSCore instance deployed in development servers of Quantag IT Solutions. Endpoint configured in file **src/extension.ts** in line 123:

return new vscode.DebugAdapterServer(5555, "cryspprod3.quantag-it.com");

To use local QSCore this line should be changed to

return new vscode.DebugAdapterServer(5555);

where 5555 is port of QSCore for incoming DAP websocket TCP connections.

Activation and starting

Extension activated on .qasm and .py files and can be started by choosing 'Debug Circuit' command from 'Run' menu





Output and debug messages.

After starting debugging session Extension debugging information can be seen in 'Output' View after choosing 'OpenQASM' output channel. *These messages are needed to debug the extension itself, not python or OpenQASM program.*

| 2 | ile Edit Selection View Go F | Run Termina | al Help | | | | | | P [Extension Dev | velopment Host] sampleWork | ipace | | 10 OS | |
|-----|--|-------------|--|---|---|--|--|---|---|----------------------------|-------|-------------------------|---------|----------------|
| Ch | | | | | | | | | | | | | | |
| | ✓ SAMELINGRESTACE ✓ gabt ✓ gab | | | ghzpy This o # This o # (C) Co # # oftai # ofthis ofthi # Any mx copyri that 1 GHZ staf | > code is p oppyright tode is l n a copy is source odificati ight noti they have te exampl | | iskit. , 2023. under the Apa license in th at <u>http://ww</u> erivative wor modified file tered from th so compares r | | sion 2.0. You may le in the root di enses/LICENSE-2.0 wust retain this a notice indicati ment and simulato | | | | | |
| | E testo2qaom | | | from gin from gin # Make a ######## num_qubi qc = Qua # Create qc.h(0) for i in qc.a | skit impo skit prov a quantum its = 5 antumCirco a GHZ s a GHZ s a range(r x(i, i + | <pre>prt Quant fiders.ba n circuit cuit(num_ state state hum_qubit 1)</pre> | umCircuit, tr sic_provider for the GHZ qubits, num_c s - 1): | ranspile import BasicSimu: state. mubits, name="ghz" | | | | | | |
| 8 |) on the | | PROBLEM 99-04- 09-04- 09-04- "stu "pat "pat ", ", ", ", ", ", ", ", ", ", | S 1 2024 09: 2024 09: 2024 09: 2024 09: 2024 09: tus": 0, es": 5, hs": [var/dap/ var/dap/ var/dap/ var/dap/ var/dap/ var/dap/ | 0UTPUT 23:00:64 23:00:64 23:00:64 23:00:84 88a327c2 88a327c2 88a327c2 88a327c2 88a327c2 | DEBUG CONS 15 Startin 16 workpl: 16 Session 11 { -13d4-4d; - | OLE TERMINAL ng OpenQASM d aceFolder: c: nId: 88a327c2 74-8a1c-0ede8 74-8a1c-0ede8 74-8a1c-0ede8 74-8a1c-0ede8 74-8a1c-0ede8 | PORTS ebugging session. .workigtquantagi -13d4-4d74-8alc-6 SS6bcf28/qmSample. SS6bcf28/test02.qu SS6bcf28/diskit/gb SS6bcf28/diskit/ts SS6bcf28/tket/tket | | ode\sampleMorkspace | | OpenAKM | | . A ^ × _ |
| 263 | > TIMELINE | | } | rent : | 09-04-20 | 24 89:23: | .40.209 | | | | | | | |
| 205 | > QUANTUM WORKSPACES | | | | | | | | | | | | | |
| × 1 | 8021 W0 B | | | | | | | | | | | Ln 18, Col 59 Spaces: 4 | UIF-8 L | the tal python |



Debug console.

In Debug Console view you can see messages sent from *remote* Quantum Virtual Machine. For example, OpenQASM parsing errors. These messages sent using <u>OutputEvent</u> DAP command.

| File Edit Selection View Go Run Termin | nal Help | | | | | P [Extension Devel | opment Host] sampleWorkspi | | 🗖 🗖 🗍 08 - 🕫 | |
|---|-----------|---|--|--|--|--|----------------------------|-----|--------------|--|
| EXPLORER ···· | 🔹 gha | | | | | | | | | |
| | | <pre>> ● ghz.py > _ # This cod # # (C) Copy # # This cod # obtain a</pre> | | | | ion 2.0. You may e in the root dire | | | | |
| 刘 File Edit Selection View Go Run Termi | inal Help | | | | | ,Ω [Extension Deve | lopment Host] sampleWorksp | ace | | |
| € Exclusion | | <pre>zzy) x { zzy) x { disp; zzy) x { disp; zzy]</pre> | B Diassembly de is part of 6 yright IBH 201 de is licensed a copy of this source tree of this source the example. It at it isoper Quark to providers b example. It at it isoper Quark to providers b example. It at the source of the this source of the the this source of the this source of the the this source of the the this source of the | kettpy giskit. 7, 2023. under the Appendix and the Appendix and the Appendix and the Appendix and the Appendix and Appendix taread from the said provider 3 second the Appendix for the Append | ◆ testpy he License, Verr LiCENGE.tst fil apache.org/lic s of this code so originals. nuning on experia mapping | | | | | |
| | | | M: [QPP 1:0] | | | | | | | |



Quantum state

Quantum state of the QVM can be seen in view 'Variables'.

After starting debugging session go to 'View' menu and choose 'Open View..' item.

| Ø | File Edit Selection | View Go Run Termina | l Help | ~ | • | √ [Extension Dev | elopment Host] sampleWorks | pace | | × |
|-----------|--|---|---|--|---------------------|------------------|----------------------------|------|--|-------|
| Ch | EXPLORER | Command Palette | Ctrl+Shift+P | | | | | | | |
|) Q 2 4 | SAMPLEWORKSPACE v qiskit g ghz.py test1.py v tket tiet1.py | Open View Appearance Editor Layout Explorer | > > Ctrl+Shift+E | | | | | | | × |
| ¢> | () config.json () config1.json | Search Source Control | Ctrl+Shift+F Ctrl+Shift+G | | | | | | | |
| ₽ B | image.jpg qmSample.qasm temp_17118243 temp_17118243 | Run Extensions Testing | Ctrl+Shift+D Ctrl+Shift+X | | | | | | | |
| ₫ | temp_17122146 temp_17123155 test02.qasm | Problems Output Debug Console | Ctrl+Shift+M Ctrl+Shift+U Ctrl+Shift+Y | e example. It also compar | s running on exper: | | | | | |
| | | Terminal | | <pre>kit.providers.basic_provi</pre> | ler import BasicSim | | | | | |
| 43 |) OUTLINE) TANLINE) TANLINE | | 21 Annuel 22 Finite 23 Finite 23 Annue 40 25 qc - Qu 25 qc - Qu 26 Creat 28 qc. HQ 29 Finite 20 Gc. HQ 20 Finite 20 Gc. HQ 20 Finite 20 Finit 20 Finite 20 Finite 20 Finite 20 Finite 20 Finite 20 Finite 20 F | a guntum circuit for the stars = 5 antumcircuit(num_qubits, + te a GUZ state n range(num_qubits - 1): cuTuT DENKCOMSOL TE QUM: [QUP 1.0] | nul ports | | | | | |
| - 000 | > QUANTUM WORKSP | ACES | | | | | | | 10.22 Col.44 Season 4 1075.9 15 () orthogo 2.120.64 bit | |
| 30 | ₩0 A | | | | | | | | Un 22, Col 44 Spaces: 4 UTF-8 LF (a python 3.12.0 64-bit | |

Then type 'Variables' in opened prompt

| | ile Edit Selection View Go Run Termin | nal Help | view Val | | |
|-----------------|--|---|--|----------|---|
| Ch | EXPLORER ···· | | Variables | Explorer | |
| - | ✓ VARIABLES | | Dev Containers | | |
| Q | ✓ Locals | | | | |
| | | | | | |
| 82 | | 4 # 5 # This code is licensed under the | | | |
| 1 | | | | | |
| 20 | | | | | |
| ₽₽ ⁰ | | | | | |
| | | | | | |
| ٦Ø | | 11 # that they have been altered fro 12 | m the originals. | | |
| π | | | | | |
| A | | 14 GHZ state example. It also compare 15 """ | es running on experiment and simulator. | | |
| | | | | | |
| | | 17 from qiskit import QuantumCircuit | , transpile dan import ParicSimulaton | | |
| | | 19 | | | |
| | | | | | |
| | | 22 # Make a quantum circuit for the | GHZ state. | | |
| | | 23 #################################### | | | |
| | | <pre>24 num_qubits = 5 25 gc = QuantumCircuit(num qubits, n</pre> | um gubits, name="ghz") | | |
| | | | | | |
| | | 27 # Create a GHZ state 28 gc.h(0) | | | |
| | | <pre>29 for i in range(num_qubits - 1):</pre> | | | |
| | SAMPLEWORKSPACE | 30 qc.cx(i, i + 1) | | | |
| | ghz.py | PROBLEMS 1 OUTPUT DEBUG CONSOLE TERM | | | Filter (e.g. text, lexclude) |
| | test1.py | Initialized QVM: [QPP 1.0] | | | |
| | ∼ tket | | | | |
| | tket1.py configure | | | | |
| | () config1.json | | | | |
| | 🖾 image.jpg | | | | |
| | ≣ qmSample.qasm | | | | |
| | temp_1711824308101.html temp_1711804924759.html | | | | |
| | temp_1712214692729.html | | | | |
| 8 | temp_1712315592567.html | | | | |
| | > OUTLINE | | | | |
| | > TIMELINE > OHANTIIM WORKSPACES | | | | |
| × (| 00∆1 ₩0 ₺> | | | | Ln 1, Col 1 Spaces: 4 UTF-8 LF 🚯 python 3.12.0 64-bit 🚨 |
| | | | | | |



Quantum state can be seen in Variables View after parsing circuit after first 'Step' command of debugger.



For example, line $|10000\rangle = 0.707107 + i0.0$ means that during measurement probability to get qubit #0 in state '1' and qubits #1-#4 in state '0' equals ½. (0.707107 ^ 2). More about state of quantum system of N qubits can be found here.



Complied OpenQASM code

After the first 'Step debug' command of python script you can see OpenQASM code in 'Disassembly View' *if python code was compiled correctly to OpenQASM in QVM*.



Disassembly View can be moved as any other View in VS code for convenience.

| Ø | File Edit Selection View Go Run Terminal | Help | | \mathcal{P} [Extension Development Host] sampleWorkspace | | | |
|------------|--|-----------------------|-------------------------------------|--|--|--------------------------|--------------|
| G | EXPLORER ···· | 魯 Disassembly × | | | | | |
| - | VARIABLES | D 100 | OPENQASM 2.0; | | qiskit> ♦ = IÞ ଫ + ↑ ᠑ Π | | |
| 0 | ✓ Locals | | include "qelib1.inc"; | | 17 from niskit import QuantumCircuit tes | ansnile | |
| \sim | 1000001 0.707107 + 10.000000 | | qreg q[5]; | | 18 from giskit.providers.basic provider i | import BasicSimulator | MARCHINE AND |
| A . | | | creg c[5]; | | | | |
| ß | | | h q[0]; | | | | |
| | 00010>: 0.000000 + 10.000000 | | cx q[0],q[1]; | | | | |
| 1 | 00011>: 0.000000 + i0.000000 | | cx q[1],q[2]; | | | | |
| 80 | 00100>: 0.000000 + 10.000000 | | cx q[2],q[3]; | | | | |
| -0 | | 108 | cx q[3],q[4]; | | 24 num_gubits = 5 | | |
| ß | | | barrier q[0],q[1],q[2],q[3],q[4]; | | 25 gc = QuantumCircuit(num gubits, num gu | ubits, name="ghz") | |
| | 00111>: 0,000000 + 10,000000 | | measure q[0] -> c[0]; | | | | |
| | 101000 + 0 000000 + 10 000000 | | <pre>measure q[1] -> c[1];</pre> | | | | |
| 0 | | | measure $q[2] \rightarrow c[2];$ | | 28 gc.h(0) | | |
| | 010013: 0.000000 + 10.000000 | | <pre>measure q[3] -> c[3];</pre> | | <pre>29 for i in range(num_qubits - 1):</pre> | | |
| A | 01010>: 0.000000 + 10.000000 | 114 | measure q[4] -> c[4]; | | | | |
| | 01011>: 0.000000 + 10.000000 | | | | | | |
| | | | | | 32 qc.barrier() | | |
| | | | | | | | |
| | | | | | 34 for i in range(num_qubits): | | |
| | 01111>: 0.000000 + 10.000000 | | | | <pre>35 qc.measure(i, i)</pre> | | |
| | 10000 + 0.707107 + 10.000000 | | | | | | |
| | | | | | <pre>37 sim_backend = BasicSimulator()</pre> | | |
| | | | | | <pre>38 job = sim_backend.run(transpile(qc, si</pre> | im_backend), shots=1024) | |
| | 10010>: 0.000000 + 10.000000 | | | | <pre>39 result = job.result()</pre> | | |
| | 10011>: 0.000000 + i0.000000 | | | | <pre>40 print("Basic simulator : ")</pre> | | |
| | 10100>: 0.000000 + 10.000000 | | | | <pre>41 print(result.get_counts(qc))</pre> | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | ✓ SAMPLEWORKSPACE | | | | | | |
| | v ojskit • | | | | | | |
| | a obz.ov. 1 | PROBLEMS 🕕 OUTPUT | DEBUG CONSOLE TERMINAL PORTS | | | | 🖬 ^ × |
| | a testi nu | Initialized QVM: [QPF | | | | | |
| | v test by | | | | | | |
| | ✓ tket | | | | | | |
| | tket1.py | | | | | | |
| | () config.json | | | | | | |
| | () config1.json | | | | | | |
| | 🖘 image.jpg | | | | | | |
| | ⊑ gmSample.gasm | | | | | | |
| | emp 1711824308101.html | | | | | | |
| | etemp 1711894924759.html | | | | | | |
| | 0 temp 1712214692729 html | | | | | | |
| 0 | A tamp 17122160267 bind | | | | | | |
| 8 | > ortrans | | | | | | |
| | | | | | | | |
| 503 | TIMEUNE | | | | | | |
| | > QUANTUM WORKSPACES | > | | | | | |
| × | ⊗0∆1 ₩0 ♪ | | | | | | 0 |



Command Palette

Extension exposes few commands which are available in Command Palette. To open Command Palette press 'Ctrl+Shift+P' in Windows or open menu 'View' and then 'Command Palette..' item

| Control Contro Control Control | 刘 Fi | le Edit Selection | View Go Run Terminal | Help | $\leftarrow \rightarrow$ | | | Host] sampleWorkspace | D 🗖 🗋 0: | – ø × |
|--|---|--|---|--|--|---|--------|-----------------------|---|---------------------------|
| G ² → watasis pop-/- pop-/- pop-/- P → watasis pop-/- pop-/- pop-/- pop-/- P → watasis pop-/- → watasis pop-/- pop-/- pop-/- P → watasis pop-/- → watasis pop-/- pop-/- pop-/- pop-/- pop-/- P → watasis pop-/- → watasis pop-/- | ц. | EXPLORER | Command Palette | Ctrl+Shift+P | | | | () config1.json | | |
| Image: non-particle in the control of the control | لح | VARIABLES | Open View | | | | | | | |
| C (=000000000000000000000000000000000000 | ر ۰۰ | ✓ Locals 00000>: 0.7€ 00001>: 0.0€ | Appearance Editor Layout | | qiskit import QuantumCircui qiskit.providers.basic_prov | :, transpile ider import BasicSimu | ulator | | | |
| Image: a constraint of the second of the | 。 4787 日 119 | 00010>: 0.00 00011>: 0.00 00100>: 0.00 00101>: 0.00 00110>: 0.00 00110>: 0.00 00111>: 0.00 01000>: 0.00 01001>: 0.00 | Explorer Search Source Control Run Extensions Testing | Ctrl+Shift+E Ctrl+Shift+F Ctrl+Shift+G Ctrl+Shift+D Ctrl+Shift+X | e a quantum circuit for the summary and the summary and the summary Quantum Circuit (num_qubits, ate a GHZ state e) | GH2 state. GH2 state. num_qubits, name="gh2 | | | | |
| Image (mapped map out 1): Image (mapped map out 1): Image (mapped map out 1): Image (mapped map out 1): Image (mapped map out 1): Image (mapped map out 1): Image (mapped map out 1): Image (mapped map out 1): Image (mapped map out 1): Image (mapped map out 1): Image (mapped map out 1): Image (mapped map out 1): Image (mapped map out 1): Image (mapped map out 1): Image (mapped map out 1): Image (mapped map out 1): Image (mapped map out 1): Image (mapped map out 1): Image (mapped map out 1): Image (mapped map out 1): Image (mapped map out 1): Image (mapped map out 1): Image (mapped map out 1): Image (mapped map out 1): Image (mapped map out 1): Image (map out 1): Image (map ou | ₫ | 01018>: 0.06 01011>: 0.06 01108>: 0.06 01101>: 0.06 | Problems Output Debug Console Terminal | Ctrl+Shift+M Ctrl+Shift+U Ctrl+Shift+Y Ctrl+ | <pre>in range(num_qubits - 1): c.cx(i, i + 1) ert a barrier before measur rrier() sure all of the qubits in t in comparison of the pubits in t</pre> | | | | | |
| Interface Interface <t< td=""><td></td><td> 01110>: 0.06</td><td></td><td></td><td><pre>in range(num_qubits): c.measure(i, i)</pre></td><td></td><td></td><td></td><td></td><td></td></t<> | | 01110>: 0.06 | | | <pre>in range(num_qubits): c.measure(i, i)</pre> | | | | | |
| • Upda Poclate © Poclate | | 10000::0,70 10001::0.00 10010::0.00 10010::0.00 10100::0.00 10101::0.00 10110::0.00 10111::0.00 \$MPLEWORSSACE | '107 + 10.000000 0000 + 10.000000 10000 + 10.000000 0000 + 10.000000 0000 + 10.000000 0000 + 10.000000 0000 + 10.000000 | 37 sim_ 38 job 39 resu 40 prin 41 prin 42 | <pre>backend = BasicSimulator() - sim_backend.run(transpile(ult = job.result() nt("Basic simulator : ") nt(result.get_counts(qc))</pre> | ąc, sim_backend), sho | | | | |
| x 60.0.4.1 W0.25 | () () () () () () () () () () () () () (| | s101.html 4759.html 2259.html 2567.html | PROBLEMS () Initializ | D OUTHUT DEBUGGONGOL TH | | | | | |
| | × 0 | 0 ∆ 1 % 0 ₄ > | | | | | | | Ln 32, Col 13 Spaces: 4 UTF-8 LF () pytho | in 3.12.0 64-bit <u>Ω</u> |

Some commands are available only during debugging sessions, some only for '.py' files or only for '.qasm' files. If you can not find some command check that you are in the right mode and correct file is opened.



Render OpenQASM/Qiskit circuit command

This command can be used during debugging session to render OpenQASM or Qiskit circuit within VS Code.

Open 'Commands Palette' and choose 'Quantag Studio: Render OpenQASM/Qiskit circuit'

| Ø | File Edit Selection View Go Run Terminal | Help | | | |
|---|---|--|---|--|-----------|
| Ð | EXPLORER ···· | 🔹 ghz.py 1 🗙 🔮 tket1.py 🔹 test | Quantag Studio: Render OpenQASM/Qiskit circuit | recently used 🔞 | |
| _ | ✓ VARIABLES | qiskit > 💩 ghz.py > | Quantag Studio: Render TKET circuit | ∷ IÞ | C ‡ ↑ D □ |
| | <pre>> took</pre> | <pre>in CHL state example. It also cor</pre> | Cuantag Studio Open Web Circuit JAET Instill Tool Report an size with the .NET Install Tool. Toggle Source Code in Disassembly View Change Language Mode Preferences: Open User Settings (JSON) Accounts: Manage Trusted Extensions For Account Add Cursor Tool User Setting Add Cursors To Bettom Add Cursors To Bettom Add Cursors To Top Add Data Breakpoint at Address Add Function Breakpoint | Ctri + K M offler commands Ctri + At + Upknow Ctri + At + DownAmow Shit + At + I | |
| | [01100:: 0.000000 + i0.000000 [01110:: 0.000000 + i0.000000 [01111:: 0.000000 + i0.000000 [01111:: 0.000000 + i0.000000 [100001:: 0.000000 + i0.000000 [100011:: 0.000000 + i0.000000 [100101:: 0.000000 + i0.000000 [100101:: 0.000000 + i0.000000 | <pre>29 Tor 1 in range(num_quoits - 1 30 qc.cx(i, i + 1) 31 # Insert a barrier before mea 32 qc.barrier() 33 # Measure all of the qubits in 37 sim_backend = BasicSimulator() 39 gob = sim_backend.run(transp1 39 result = job.result() 40 print("Basic Simulator : ") 41 print(result.get_counts(qc)) 4</pre> | Add Selection To Previous Find Match Add Selection To Previous Find Match Add Selection To Previous Find Match Add Selection To Devious Find Match Add Selection To Devious Find Match (qc, sim_backend), shots=1024) | | |
| | 10101>: 0.000000 + i0.000000 | 42 | | | |

If circuit correctly parsed in QVM you will see rendered circuit in new View in VS Code.





You can rearrange visualized circuit like any other View in VS Code to see source code and circuit at same time:





Web frontend

Additionally, QVM has its own web frontend which opens in external browser *after starting debugging session* and can render circuits, show quantum state and source. It is implemented on angular.js and communicates with QSC using web-socket connection.

To open web frontend use command 'Open Web Circuit'.

| × | File Edit Selection View Go Run Terminal | Help | | | | | |
|----------------|--|----------------|--------------------|----------|--|--------------------------|---------------|
| Сh | EXPLORER | 🍨 ghz.py 1 🕽 | < 🍨 tket1.py | 🔹 test | Quantag Studio: Render OpenQASM/Qiskit circuit | recently used 👹 | |
| | ✓ VARIABLES | qiskit > 🗇 ghz | .py ≻ | | Quantag Studio: Render TKET circuit | ≣ Þ | C + ↑ D |
| Q | ✓ Locals | D 1 # Thi | | | Quantag Studio: Open Web Circuit | сэ | |
| 1 | 00000>: 0.707107 + i0.000000 | | | | .NET Install Tool: Report an issue with the .NET Install Tool. | | |
| Ŷ٥ | 00001>: 0.000000 + i0.000000 | 3 # (C) | Copyright IBM 201 | 17, 202 | Toggle Source Code in Disassembly View | | |
| 5 | 00010>: 0.000000 + i0.000000 | | | | Change Language Mode | | |
| | 00011>: 0.000000 + i0.000000 | 6 # obt | ain a copy of this | | Preferences: Open User Settings (JSON) | | |
| - É(1 | 00100>: 0.000000 + i0.000000 | | | | Accounts: Manage Trusted Extensions For Account | | |
| _ 0 | 00101>: 0.000000 + i0.000000 | | | | Add Cursor Above | Ctrl + Alt + UpArrow | |
| B | 00110>: 0.000000 + i0.000000 | | | | Add Cursor Below | Ctrl + Alt + DownArrow | |
| | 00111>: 0.000000 + i0.000000 | | yright notice, and | d modif | Add Cursors To Bottom | | |
| L ^o | 01000>: 0.000000 + i0.000000 | 11 # tha | t they have been a | altered | Add Cursors to Line Ends | | |
| | 01001>: 0.000000 + i0.000000 | | | | Add Cursors To Top | | |
| Д | 01010>: 0.000000 + i0.000000 | 14 GHZ 9 | tate example. It a | | Add Data Breakpoint at Address | | |
| | 01011>: 0.000000 + i0.000000 | | | | Add Function Breakpoint | | |
| | 01100>: 0.000000 + i0.000000 | | | | Add Line Comment | | |
| | 01101>: 0.000000 + i0.000000 | 17 from | qiskit import Quar | ntumCir | Add Selection To Next Find Match | | |
| | 01110>: 0.000000 + i0.000000 | 18 from | qiskit.providers. | basic_pr | Add Selection To Previous Find Match | | |
| | 01111>: 0.000000 + i0.000000 | | | | Add VIID (fatab Decalescint | | |
| | 10000>: 0.707107 + i0.000000 | 21 ##### | **************** | | | | |
| | 10001>: 0.000000 + i0.000000 | | | | | 32 qc.barr | ier() |
| | 10010>: 0.000000 + i0.000000 | | | | | 33 # Measu 34 for i i | n range(num d |
| | 10011>: 0.000000 + i0.000000 | | | | | 35 ac. | measure(i, i) |
| | 10100>: 0.000000 + i0.000000 | | | | | 36 | |

It can render circuits using own renderer in View or Qiskit renderer in 'Alternative View' is available.

| C Quantu | n × + | | |
|---------------------------------|--|---|---------|
| $\leftrightarrow \rightarrow G$ | quantag-it.com/quantum/#/qcd?id=7b8019e6-a797-403c-8639 | -c130c252a827 | ९ 🖈 👻 የ |
| | Quantum Cloud Debugger Connection status: © Orane OpenQASM Code Visualisation | | |
| | Current Line: 2 (00000 :: 0.707107 +0.000000 (0001 :: 0.00000 +0.000000 (0011 :: 0. | 1 OringLos J.9; 1 Biolog Building 1 OringLos J.9; 1 Biolog Building 1 OringLos J.9; 1 OringLos J.9; | |
| | Circuit ALTERNATIVE VIEW ALTERNATIVE VIEW | | |
| | 3kp 1 2 3 4 5 6 7 8 9 10 0(10) H - < | | |



Alternative Web View



Web frontend can be used to integrate QSCore in existing web cloud services.



Limitations and future plans

QSCore component and extension currently under active development. Not all features are fully implemented and stable.

Here is list of main limitations:

- not all circuits are working and supported. QSCore analyzes python sources and modifies them on the fly to add rendering and OpenQASM exporting. If circuit is complex with mixed functions and modules it might fail, and errors will be shown in 'Debug output' view
- For big workplaces file transfer might take some time, and there can be delay at the beginning of session.
- QVM initialization can be slow for big number of qubits in circuit.
- There is no mapping yet between python and compiled OpenQASM sources.

Future plans are:

- Adding commands for Circuit Compression and Qubit-to-Qudits circuit conversions
- Adding support of new hardware for direct execution
- Adding support of cloud-based execution of circuits
- Adding different types of QVM with hardware acceleration (Nvidia cuQuantum https://developer.nvidia.com/cuquantum-sdk) currently we use https://github.com/softwareQinc/qpp as QVM.
- Adding support of hybrid programs (classic + quantum)
- Adding visualization of measurement results in VS code
- Adding circuit step animation of debugging session in VS Code.

Please report found bugs to support@quantag-it.com

