



Fraunhofer
FOKUS

Qrisp

Höhere Programmiersprache für
Quantencomputer

Was ist Qrisp?

Qrisp ist eine höhere Programmiersprache für die Programmierung von Quantencomputern. Sie ermöglicht Entwicklerinnen und Entwicklern komplexe Quantenalgorithmen mit einer leicht zu erlernenden, modernen Syntax zu schreiben und diese bis auf Schaltkreisebene zu kompilieren.

Durch die Automatisierung eines Großteils der Low-Level-Programmieraufgaben, wie das hinzufügen einzelner Gatteroperationen oder das Qubit-Management, soll das Forschungsgebiet für ein breites Publikum geöffnet werden. Qrisp ist ein Open-Source Projekt bei der Eclipse Foundation, wird hauptsächlich vom Fraunhofer FOKUS entwickelt und durch das Bundesministerium für Wirtschaft und Klimaschutz (BMWK) gefördert.

Überblick über das Framework

Die zentrale Datenstruktur für die abstrakte Quantenprogrammierung ist die »QuantumVariable«. Der Lebenszyklus der »QuantumVariable« wird, neben anderen Aspekten, von der Klasse »QuantumSession« gesteuert, die auch für die Interaktion mit der QPU (Quantum processing unit) zuständig ist. Dank eines ausgeklügelten Systems zur Verwaltung von »QuantumSessions« müssen die Nutzerinnen und Nutzer von Qrisp keine »QuantumSession«-Objekte verwenden und entsprechend Qubits selbst verwalten, sondern können direkt die »QuantumVariable« nutzen.

In vielen Fällen sind aber reine »QuantumVariablen« nicht sehr hilfreich, da sie aufgrund ihrer Allgemeinheit nur sehr wenige fortgeschrittene Datenverarbeitungsmöglichkeiten bieten. Man kann sie sich als abstrakte Basisklasse für spezifischere Datentypen vorstellen.

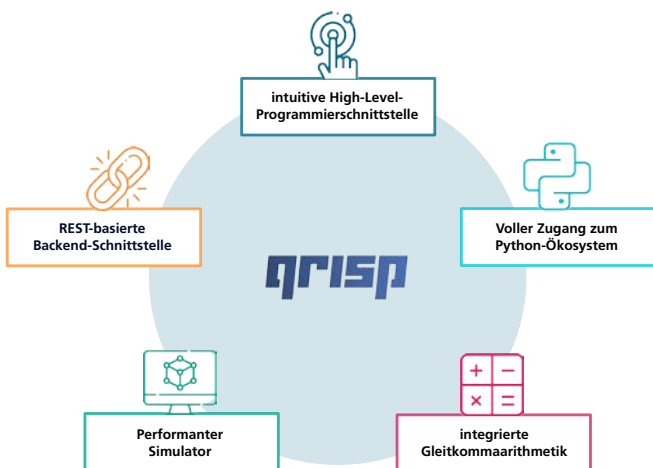


Abbildung 1: Überblick über das Qrisp-Programmierframework

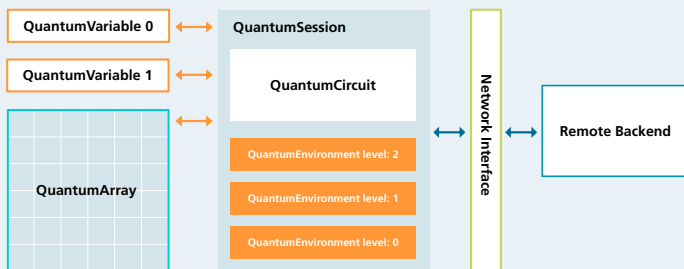


Abbildung 2: Strukturübersicht des Frameworks



Die Entwicklung von Quantenalgorithmen mithilfe von Variablen ermöglicht eine strukturierte, modulare und damit professionelle Herangehensweise an komplexe Algorithmen.«

Raphael Seidel,
wissenschaftlicher Mitarbeiter (Senior) im Geschäftsbereich SQC

Qrisp bietet vier fortgeschrittene Quantum-Datentypen:

1. *QuantumFloat*: Darstellung und Verarbeitung von Zahlen mit beliebiger Genauigkeit
2. *QuantumBool*: Darstellung boolescher Werte
3. *QuantumChar*: Darstellung von Buchstaben und Schriftzeichen / Characters
4. *QuantumString*: Darstellung von Zeichenketten

»*QuantumVariablen*« desselben Typs können in der Klasse »*QuantumArray*« verwaltet werden. Diese Klasse bietet viele praktische und bewährte Funktionen wie »*Slicing*«, also der einfache Zugriff auf einen bestimmten Teil des Arrays, oder Reshaping, das Anpassen der Größe des Arrays.

Mit dem Konzept der »*QuantumEnvironments*« ist es möglich, viele der etablierten Methoden der klassischen Programmierung zu nutzen, wie die bedingte Ausführung von Codeblöcken.

Da sich Quantenschaltkreise bei der Implementierung und Forschung zu Quantenalgorithmen etabliert haben, bieten wir mit dem »*Circuit-Construction-Modul*« eine Möglichkeit zur Konstruktion solcher Quantenschaltkreise. Die Struktur und Benennung der Klassen und Methoden in Qrisp wurden für eine bessere Kompatibilität so weit wie möglich vom bekannten Programmierframework für Quantencomputer »Qiskit« übernommen.

Um eine anwendungsorientierte Algorithmenentwicklung zu gewährleisten, verfügt Qrisp über eine Netzwerkschnittstelle, um Quantenprogramme auf realen (Remote)-Quantenbackends auszuführen. Diese wird realisiert, indem der Backend-Anbieter einen Backend Server in seiner Netzinfrastruktur betreibt, mit dem der Benutzer sich über ein Backend-Client-Objekt verbinden kann.

Darüber hinaus unterstützt Qrisp die Ausführung von Quantenschaltkreisen auf den Backends von etablierten Anbietern unter Verwendung der Klasse »*VirtualBackend*«.

Vorteile von Qrisp

Mit Qrisp können sich Programmiererinnen und Programmierer auf die entscheidenden Aspekte ihres Codes konzentrieren und den Arbeitsaufwand für das Management einzelner Qubits und Quantengatter reduzieren. Durch ein ausgeklügeltes Qubit-Management-System werden recycelte Quantenressourcen automatisch über Funktionen hinweg wiederverwendet. So lässt sich Qrisp-Code effektiv modularisieren. Auf diese Weise können Programme in kleinere Module unterteilt werden, die sich dann flexibel einsetzen lassen und nicht immer wieder neu implementiert werden müssen. Kombiniert mit einem Typensystem, welches nahtlos in die Python-Infrastruktur integriert ist, wird die Entwicklung skalierbarer Algorithmen zu einem unkomplizierten Prozess. Qrisp ist einfach zu bedienen und hat einen schnellen, komfortablen Entwicklungsablauf.

Zusammenfassung

Mit Qrisp können Entwicklerinnen und Entwickler leicht mit Quantenalgorithmen arbeiten, auch wenn sie bisher keine Erfahrungen mit der Programmierung von Quantencomputern haben. Das Framework senkt nicht nur die Einstiegshürde, sondern eröffnet neue Ebenen der Komplexität in Quantenalgorithmen, die bisher ungeahnte Quantenvorteile aufdecken können. Qrisp ist als Open-Source-Code verfügbar und offen für Beteiligungen aus der Community.

More information can be found at:



<https://qrisp.eu>



<https://github.com/eclipse-qrisp/Qrisp>

Kontakt

Prof. Dr.-Ing. Nikolay Tcholtchev
Gruppenleiter Quality Engineering
für Urbane IKT und Quantencomputing
Geschäftsbereich SQC
Tel. +49 30 3463-7175
nikolay.tcholtchev@fokus.fraunhofer.de

Sebastian Bock
wissenschaftlicher Mitarbeiter (Senior)
Geschäftsbereich SQC
sebastian.bock@fokus.fraunhofer.de

Raphael Seidel
wissenschaftlicher Mitarbeiter (Senior)
Geschäftsbereich SQC
raphael.seidel@fokus.fraunhofer.de

Fraunhofer FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin

www.fokus.fraunhofer.de

Wir
vernetzen
alles