



Developers get more Quantum Computing goodies

By Syeda Beenish - March 6, 2018

```
/// A qubit initially in the |0> state that we want to send
/// the state of msg to.
operation ValueTransfer (qubit msg : Qubit) : () {
    body {
        using (var here : Qubit[1]) {
            // Ask for an auxiliary qubit that we can use to prepare
            // for teleportation.
            let there : Qubit[1] = Create();
            // Create an entanglement that we can use to send
            // the message.
            CNOT(here, there);
            // Move our message into the entangled pair.
            CNOT(there, here);
            H(here);
            // Measure out the entanglement.
            if (Measure(there) == One) { Z(there); }
            else { X(there); }

            // Reset our "here" qubit before releasing it.
            Reset(here);
        }
    }
}
```

Microsoft's [Quantum Development Kit](#) (QDK) gets the first major upgrade since its introduction last year. It witnesses addition of several new features designed to open the platform to a wider array of developers.

The updated QDK will be interoperable with the Python computing language.

According to Microsoft, developers have been seeking for long for the kit to be made available for [Linux](#) and macOS. This, newly released QDK includes support for Linux and macOS, as well as additional open source libraries.

The Quantum Development Kit was announced last fall at Microsoft's Ignite Conference.

Supporting interoperability for Python allows developers to access their existing libraries from Q# without having to port it, Friedman pointed out. "These updates support Microsoft's ambition to enable a scalable Quantum solution for the broadest set of customers – eventually realising our goal of delivering Quantum as a Service on [Azure](#)," he further added.

Syeda Beenish

Show/Write Comments