Brant Norris
CEN3907C
March 18, 2022

Pre-Alpha Build

External Interface

Hardware:

The external interface for the hardware portion of this project will be the measurement device that is used to measure the frequency shift of the graphene resonator. When a qubit is coupled with a resonator device, the resonator device's frequency will shift according to the current state of the qubit. Thus, by measuring the frequency shifts in our graphene-based NEMS resonator, we can measure the state of the quantum system. This will be accomplished with a probing signal and an oscilloscope. Thus, the oscilloscope will essentially be the external interface for the hardware portion of this project.

Software:

The external interface for the software portion of the project will be the GUI for the simulation software. This will look similar to the images and renderings from the "Storyboard" section of my Design Mockup.

Persistent State

Hardware:

The persistent state of the hardware portion of the project will be the quantum state in which the qubit that is coupled to the resonator resides in at any given time. This will be defined by its quantum mechanical behavior.

Software:

The persistent state for the software portion of the project will be the storage of the data and state of the simulator. This will be handled in the code utilizing Python libraries such as Qiskit and QuTiP. In order to properly store the data and state of the system being simulated, quantum software objects and circuits will be utilized to store and manipulate information about the system. For example, QuTiP allows developers to create a quantum object that stores information on the state of the system and the qubits in matrix form. It also provides users with the ability to manipulate the data, information, and attributes held by these objects. Similarly, Qiskit provides developers with the ability to create, manipulate, and measure quantum circuits containing qubits and quantum logic gates. These tools and methodologies will allow for the tracking of the persistent state of the system.

Internal Systems

Hardware:

The internal system for the hardware portion of the project will be the qubit itself, along with the circuitry surrounding it (the bias tee, the ground circuit, the measurement scheme, etc) and the graphene NEMS resonator that will be coupled to the qubit. The schematics showing the components of this internal system can be found in my Design Mockup.

Software:

The internal system for the software portion of this project will be the source code that is used to develop the simulator, the tools and libraries that are utilized in its creation, and the platforms on which it is developed and released. The source code will be written primarily in Python. The primary tools and libraries that will be used in this project include numpy, matplotlib, LaTeX, Qiskit, and QuTiP. These tools and libraries, along with their dependencies, will comprise a large portion of the internal systems for the software portion of this project. Additionally, the platform that hosts the program and its developments will comprise a portion of the internal system as well since it will utilize its computational abilities to run the simulation software.

Communication

Hardware:

The hardware portion of this project does not rely heavily on communication between systems. The only communication that will occur is the measurement scheme using the probing signal to measure the frequency of the graphene resonator.

Software:

The only communication that will occur with the software portion of the project will be between the users and the simulation program itself. This will just involve the web-based application on which the simulator will run. This should allow for anyone to use the simulator regardless of the operating system or hardware that they are utilizing on the client-side.

Integrity and Resilience:

Hardware:

The hardware portion of this project will display its integrity and resilience based on how often it provides the expected result. From these results, we can create a matrix and histogram of the frequency of all possible results. This will allow us to estimate the coherence, integrity, and resilience of the system.

Software:

The integrity and resilience of the software portion of the project will be tested during the development process. It should be relatively easy to fix any bugs that affect the integrity and resilience of the simulator. Some of the integrity and resilience may depend on the hosting platform as well, however, which will likely be out of our control to some extent.