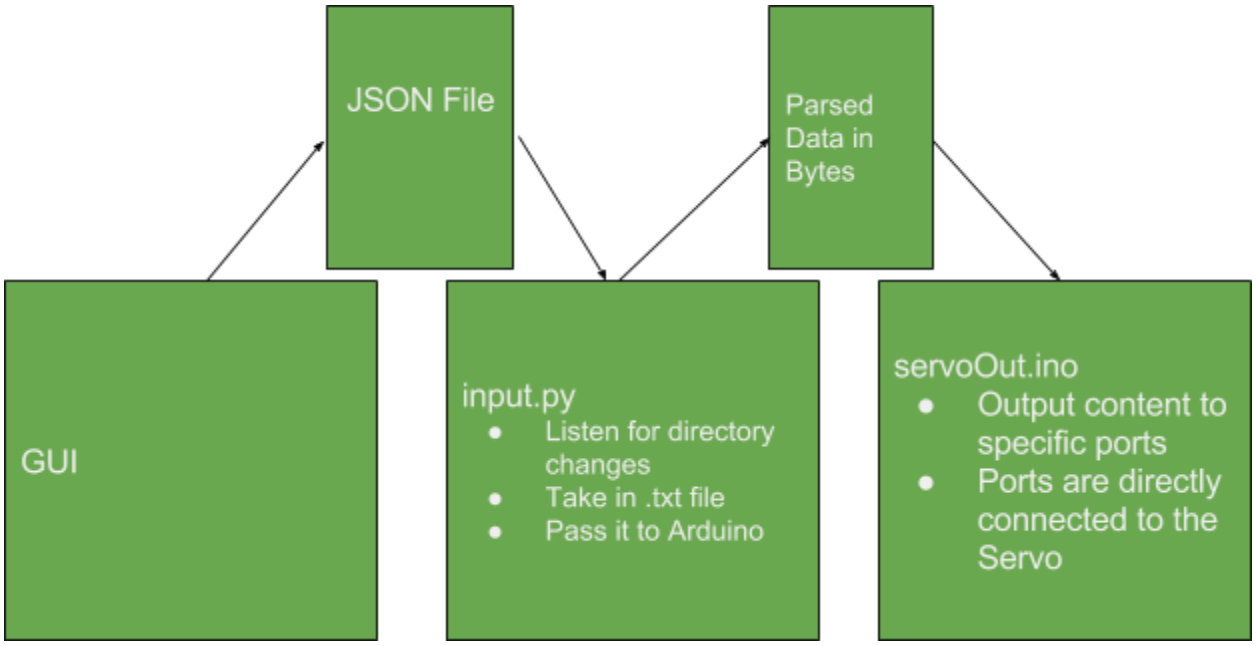
Alex Saladna

Senior Design Documentation

May 2nd, 2016

Flow Diagram



Input.py Overview:

Input.py was primarily tasked with handling data coming in from the GUI. This data was stored in JSON files and whenever a new JSON file was added, input.py would have to pull that file and extract the needed data. It would then pass it along as Bytes to servoOut.ino so that it could write the data to our microcontroller.

Input.py Challenges Faced:

Passing the data to the Arduino was the biggest challenge during this part. I would constantly get errors when I would try to pass values and I eventually discovered that I needed to specify the size of my message before sending it. Another problem I faced was figuring out how to listen to a directory for changes. I found a great library called watchdog that was designed to do this very thing and cut down the amount of code I needed significantly.

ServoOut.ino Overview:

For this task, I needed to take in an array and pass the data to 16 servos at the same time. I began by using the millis() operator which relies on the arduino's internal timer to write to the servos. The issue soon became apparent that doing this would be extremely expensive since each servo would require a for loop nested within the other. I decided to do some research and discovered that a 16 pin servo controller existed. This made my life much easier and also allowed me to move all 16 servos with a fraction of the memory cost.

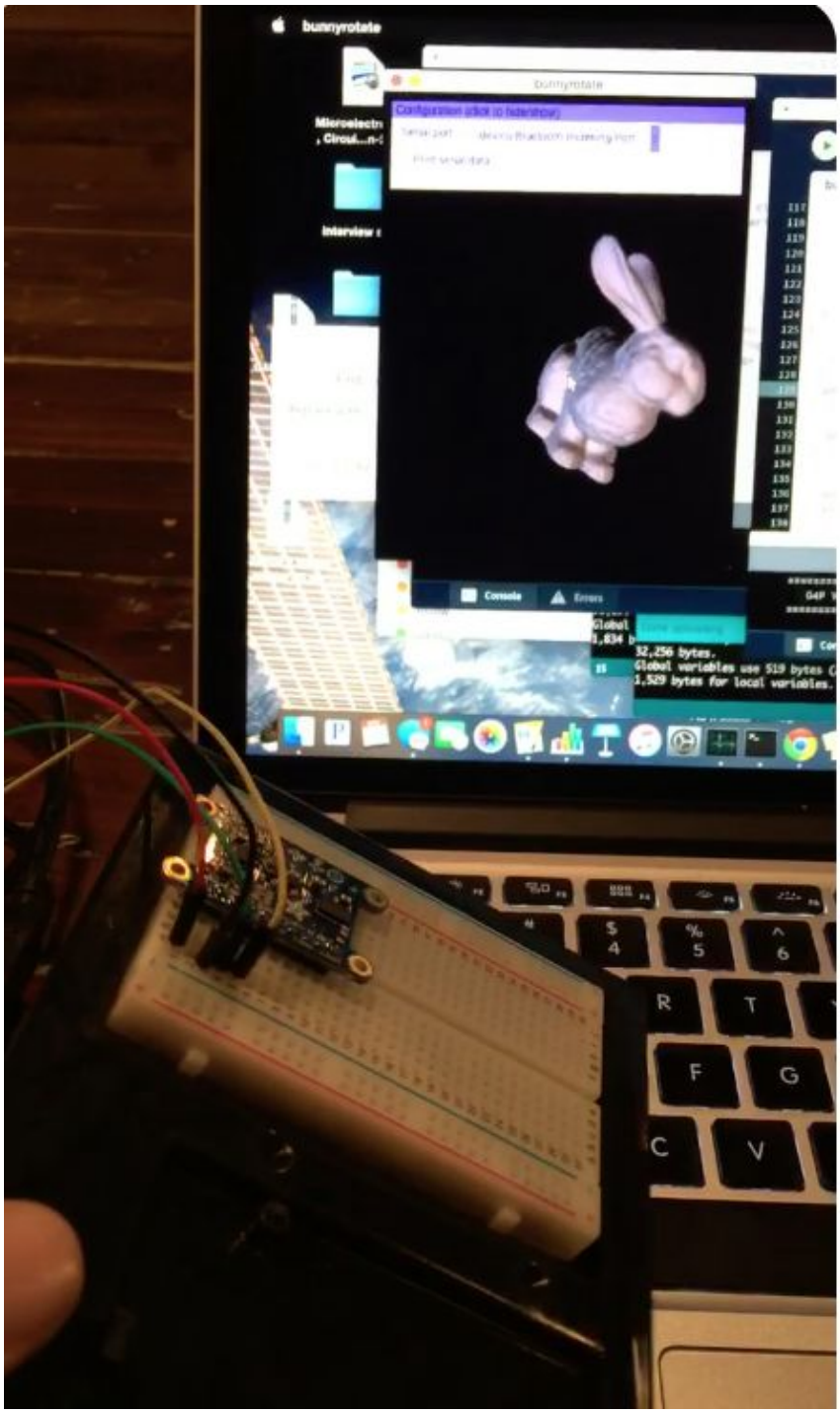
ServoOut.ino Challenges:

The millis() operator took up a large amount of my time since I did not realize that servo controllers existed. I also ran into an issue where servos would move when they were not told to, but I eventually discovered this was a minor bug within my code. My final issue was processing power. If I moved all of the servos at the same time the arduino uno would sometimes run into problems. As a result, I switched out the uno for an arduino mega which fixed my performance problems.

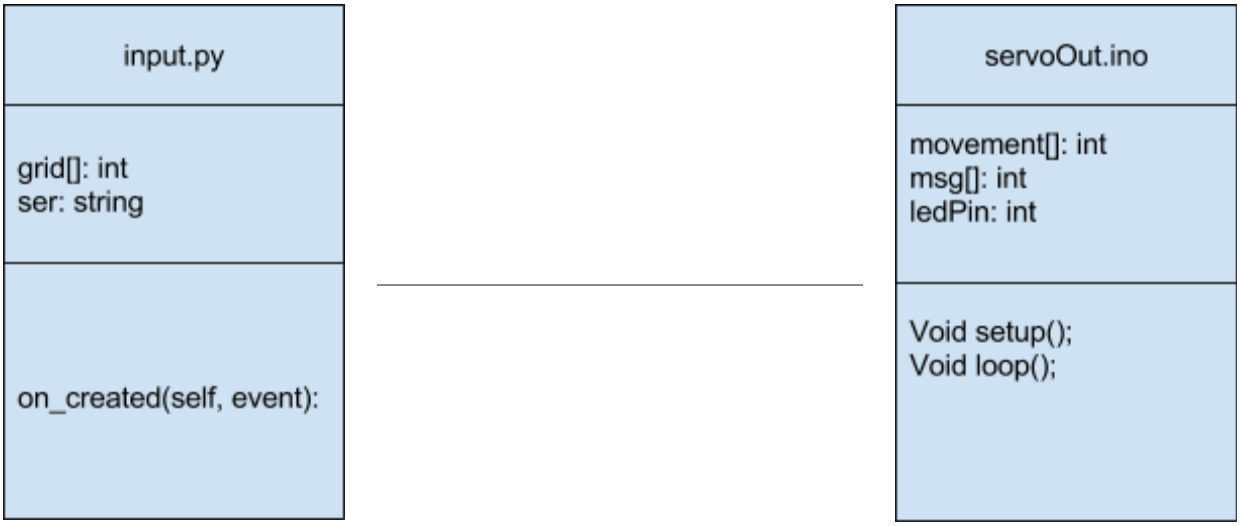
Additional Features:

I did implement a 3d interface that responded to a triple axis compass. This would have allowed our GUI to mimic the device being tilted and made it more responsive. Unfortunately, this did not make it to our final product and was eventually scrapped.

Triple Axis Compass Test Output



UML Diagram



Basic Wiring for Arduino Mega & Servo Controller