|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Program | Min Voltage | Max Voltage | Period (s) | Close to 100ms |
| Shell | -130 mV | 3.35 V | 0.238095238 | 138 ms higher |
| Python | -130 mV | 3.25 V | 0.1 | exact |
| C | -160 mV | 3.37 V | 0.1 | exact |

Overall they were close but the difference in shell can be explained by the fact there is an OS that is handling the execution of programs. Since the program isn’t the only one running it gets swapped out and maybe the Python and C have higher priority so they were far more accurate.

|  |  |  |  |
| --- | --- | --- | --- |
| Period | Shell | Python | C |
| 100 ms | 20% | 3% | 3% |
| 10 ms | 66% | 8% | 3% |
| 1 ms | 95% | 20% | 6% |
| .1 ms | No Data | 70% | 23% |

CPU Usage

|  |  |  |  |
| --- | --- | --- | --- |
| Period | Shell | Python | C |
| 100 ms | 238 ms | 100 ms | 100 ms |
| 10 ms | 60 ms | 12 ms | 10 ms |
| 1 ms | 40 ms | 1.35 ms | 1.15 ms |
| .1 ms | No Data | 480 us | 270 us |

Actual Period

SHELL:

The shell seemed fairly stable except for the 1ms. I ran other programs but it didn’t seem to affect it at all. I could really get the program to reduce too much. The lowest I got was 230 ms.

Python:

The most unstable was .1ms. It was seemed pretty unstable when I had multiple things open.

C:

This was by far the fasted and most stable for me. Even with programs running I couldn’t get the CPU very high. I had a few blips at .1ms but for the most part it was way more stable than python.