**Homework 7 Report**

Below are the speeds and CPU usage for each I/O method we used:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| I/O Method | Min Time | Max Time | Average Time | CPU % |
| Python | 560us | 4.18ms | 752us | 2.6% |
| Mmap() | 240ns | 820ns | 529ns | 100% |
| Kernel | 295us | 550us | 324us | 2.6% |

Looking at the table above, the using a kernel module and Python had very similar results. Each had times in the microseconds and required little CPU usage. Using the mmap() function in C however, used up 100% of the CPU. On a side note, I was using 3 CPU cores on my virtual machine when I ran these programs. The C program was taking up most of the CPUs resources. This makes sense looking at the results because the C program ended up being the fastest. Something else I noticed is that the max time for the python method was much larger than any of the other numbers in the table. I think that this is either a mistake on my part when I was measuring these values on the oscilloscope, or that python very variable on how fast it can handle GPIO input. The last thing I noticed was that even though both the kernel and python method had the same CPU usage, the kernel had faster times. This tells me that the kernel is a more efficient method to use than the python method. This makes since because the kernel interfaces more directly with the beaglebone.