

I tested three different methods of copying an input from one pin to another on the beaglebone. One way was high level with python, one was with mmap and C, the final way was with a kernel module.

The python method used the modified adafruit library with interrupts. On pin change of pin GP1_3, the output of pin GP1_4 was toggled. When it wasn't changing the pin, it slept the program. This method was the slowest (see Table 1), and used ~38% CPU usage which was the second lowest.

The next method that was used was direct memory manipulation with mmap. For this method to work, the program must be scanning the IO constantly, and therefore it demanded the most from the CPU, with close to 100% utilization. This method was in fact the fastest however, with average delay times of less than half a microsecond. See Table 1 for complete results.

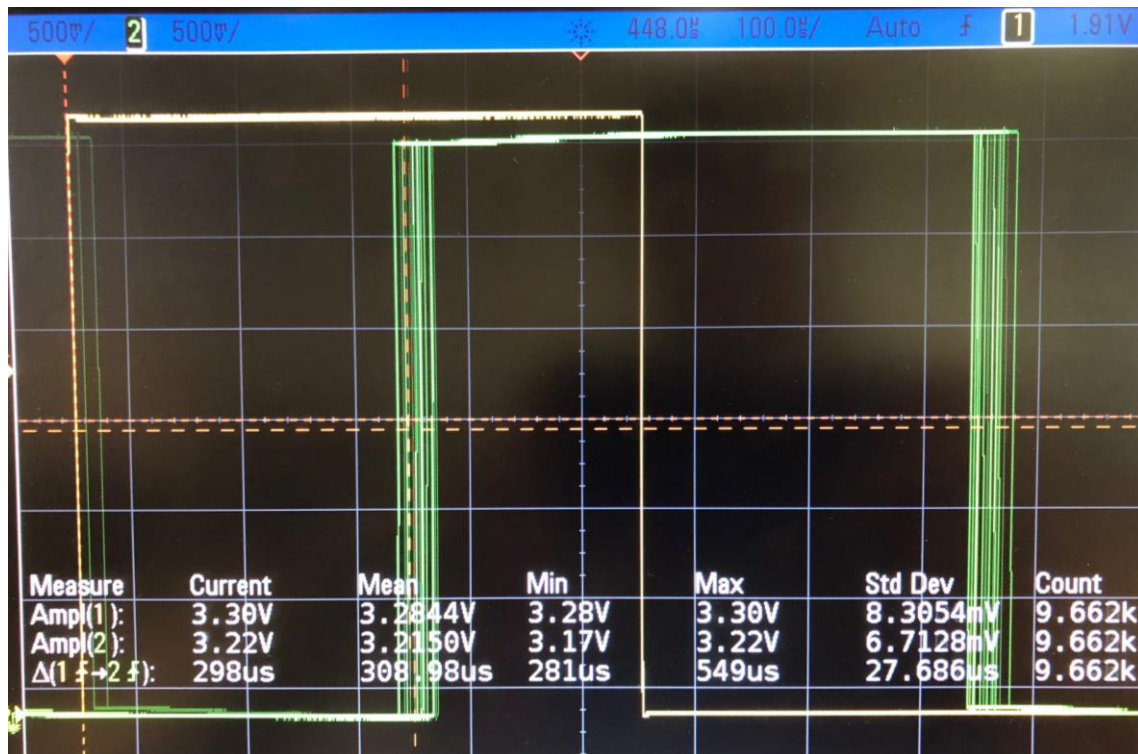


Figure 1: IO delay of kernel module

The final method was what I expected to be the fastest, however it was only a bit better than the python method. It used less CPU than any of the other methods however, at 17%. Figure 1 shows the details of how the results were gathered for the test.

	Min Delay	Average Delay	Max Delay	CPU Usage
Python with interrupts	426,000 ns	749,000 ns	1,328,000 ns	38%
C with mmap	260 ns	600 ns	2,260 ns	99%
Kernel module	281,000 ns	309,000 ns	549,000 ns	17%

Table 1: I/O passthrough delays