

Measuring a GPIO pin on an Oscilloscope

Shell

1. min voltage = -200mV max voltage = 3.48V
2. T = 135.6ms
3. Delta = 35.6ms, 30.22% difference (not good)
4. The shell is very slow at interfacing with GPIO because of the process it must go through to change the values. It can't operate fast enough to achieve the desired period of 100ms.
5. Used 3.2% of the CPU on togglegpio.sh with 100ms desired period.

expected T	actual T	CPU used
100 ms	135.6 ms	5.20%
50 ms	85.10 ms	8.40%
200 us	35.2 ms	20.90%
100 us	35.4 ms	20.50%

- 6.
7. The last two trials for the period were not very stable. It varied between 35 ms and about 85 ms.
8. When vi launched the period became very unstable, but overtime it began to settle back at the level from before vi was launched.
9. I saw minimal impact on the period. The greatest change was shortening the period by < 1ms.
10. Yes, the period was noticeably shorter using sh instead of bash
11. The shortest period I observed was 25.30ms.

Python

1. min voltage = -200mV max voltage = 3.56V
2. T = 100.5ms
3. Delta = 0.5ms, 0.50% difference (good)
4. Python is much faster interfacing with GPIO than the shell, which is why it was able to come much closer to the desired period. And as a bonus it is also to interface with as a user.
5. Used 2.6% of the CPU on togglegpio.py with 100ms desired period.

expected T	actual T	CPU used
100 ms	100.5 ms	0.80%
50 ms	50.40 ms	0.70%
200 us	548.2 ms	35.90%
100 us	441.0 ms	47.80%
1 us	168.6 us	95.50%

- 6.
7. My first three trials the period was very stable. But when I began to Python's limits the period began to jump and jitter greatly.
8. Launching vi hardly seemed to affect the stability of the period with Python, if any at all.
9. Nothing to remove from togglegpio.py

10. N/A
11. The shortest period I observed was 157.6 us.

C

1. min voltage = -120mV max voltage = 3.48V
2. T = 100.4ms
3. Delta = 0.4ms, 0.40% difference (good)
4. C is also very fast at interfacing with GPIO, and much faster than the Shell as evident by the accuracy in achieving the desired period.
5. Used 0.7% of the CPU on togglegpio.sh with 100ms desired period.

expected T	actual T	CPU used
100 ms	100.4 ms	0.70%
50 ms	50.30 ms	0.70%
200 us	481.3 us	33.30%
100 us	377.1 us	60.20%
1 us	277.3 us	72.10%
100 ns	276.4 us	93.90%

- 6.
7. The period only began to get very unstable on the trials after 100 us. Especially with the 100 ns trial, the period got very jump as the limit of the timing was reached.
8. When launching vi, there was one very high initial spike in the period, but it quickly leveled back to normal activity with no other effects on the period's stability.
9. Nothing to remove from togglegpio.c
10. N/A
11. The shortest period I obtained on the oscilloscope was 275.8 us.

Script Results

	<u>Shell</u>	<u>Python</u>	<u>C</u>	<u>Shell</u>	<u>Python</u>	<u>C</u>
expected T	actual T	actual T	actual T	CPU used	CPU used	CPU used
100 ms	135.6 ms	100.5 ms	100.4 ms	5.20%	0.80%	0.70%
50 ms	85.10 ms	50.40 ms	50.30 ms	8.40%	0.70%	0.70%
200 us	35.2 ms	548.2 ms	481.3 us	20.90%	35.90%	33.30%
100 us	35.4 ms	441.0 ms	377.1 us	20.50%	47.80%	60.20%
1 us	-----	168.6 us	277.3 us	-----	95.50%	72.10%
100 ns	-----	-----	276.4 us	-----	-----	93.90%