I have performed some investigation into vPortSuppressTicksAndSleep. I have added a GPIO toggle around platform_power_down_hook and then a printf statement after the portEXIT_CRITICAL that prints the xExpectedIdleTime versus the adjust_ticks value.

The GPIO toggle and printf's are captured with a logic analyzer and I have attached the capture if anyone has Saleae software.

Observations

1. A lot of sleep intervals lasting < 1mS, generally right around 380uS. Very repeatable. This causes the vTaskStepTick to step by 1, increasing error.

2. When the sleep time is close to the expectedIdleTime value, the real sleep time can be up to 1.5mS longer than what is stepped by vTaskStepTick . This maybe the root cause relating to the earlier issue of adjust_ticks being > xExpectedIdleTime and causing the assert.

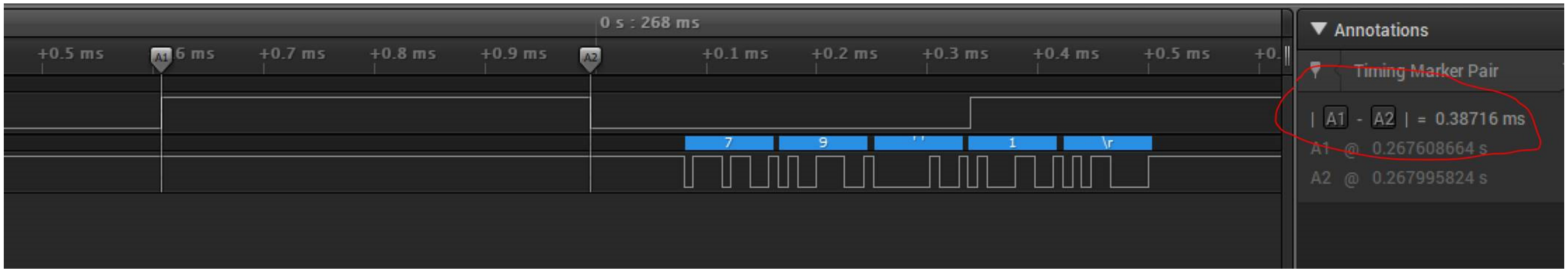
I understand that there is some accuracy loss while using tickless sleep, BUT when there are hundreds of sleep intervals per minute (mostly waking for lwip thread/tcpip timers), the time error is growing too fast.

Captures

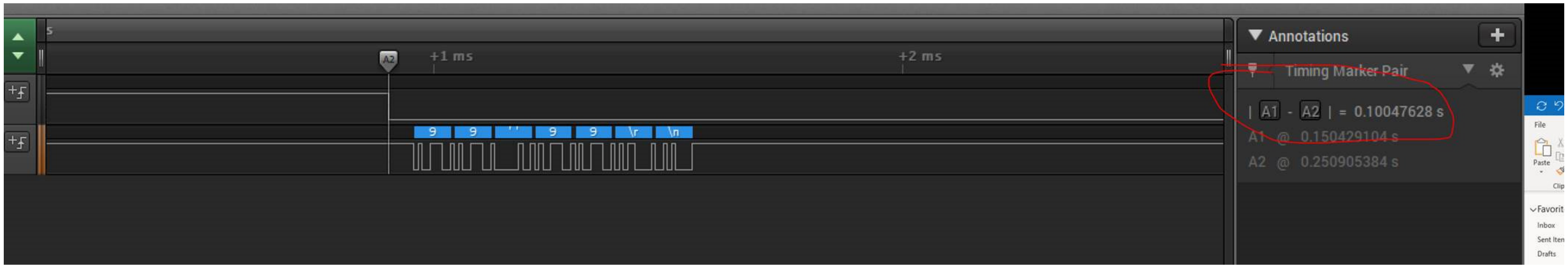
Top line is the GPIO toggle wrapping the sleep time

Bottom line is STDIO output in format of Expected sleep time actual sleep time

I am very curious about the < 1mS intervals. I guess these could be interrupts waking the processor BUT why are they almost always around 380uS and appear to happen prior to a longer sleep interval?



| A1 - A2 | = 0.38716 ms
A1 @ 0.267608664 s
A2 @ 0.267995824 s



| A1 - A2 | = 0.10047628 s
A1 @ 0.150429104 s
A2 @ 0.250905384 s