question | Like **17** views

functional debugging performance comment / question

I am taking this embedded systems class as a review given I've worked in this area in the past, and also because I wanted to get to know your system you are using for this online class. I am also a bit behind in it for many reasons, so about one week behind the recommended syllabus suggestion (just finishing C9) and in that section you discuss the functional debugging performances.

Yesterday I attended an Agilent presentation on a "spectrum analyzer" which I've never used before, but the person sitting next to me said that that would have been something that you could use to debug a wireless driver I had ported once for an embedded system on which I had worked. Having just watched the performance of debugging video for C9, I'm guessing that the "spectrum analyzer" falls in the realm of non-intrusive debug tool as the logic analyzer is considered for observing outputs from an embedded system. At the time I did this (14 years ago!) I only had a sniffer tool to look at ethernet and USB packets coming out on those wires, but no wireless debug tool, so perhaps that would have been a non-intrusive tool I could have used for that?

Also, I seem to recall and I don't know if it is going to be covered in this class, that when you have to use print statements, the most time consuming and intrusive tool for debugging an embedded system, that your code may not even work correctly if you made the mistake of putting a print statement in an interrupt service routine, because of all the context switching you do saving register state, return address, etc. So even if there is no bug in your code, your code could be so slowed down by putting a print statement in an interrupt service routine that that in itself could cause a bug in your code due to too much time spent saving state combined with the time it takes to do a print statement.

I'm from ancient times though, and these comments were made to me when I was working in those ancient times, but thought I'd mention it, given this was the week (C9--I'm a week behind in my review of taking this class!) for debugging performance.

c9 lecture_videos just_chat

3 hours ago by Karen West

the students' answer, *where students collectively construct a single answer*

The inability to use "print" in interrupt service routines has less to do with speed, and more to do with the fact that "print" frequently relies on interrupts for its own operation. Since simple microcontrollers (say, an AVR on an Arduino) only have one priority level for interrupts, when you are inside an ISR, all other interrupts are disabled. If print() needs interrupts to occur to send characters out the UART, and interrupts can't occur because you are inside some other ISR, everything stops. Which is more than just invasive...

In my experience, invasive debugging is much better than no debugging (especially if it's just "slow" debugging.) The number of bugs that are actually closely timing-limited is much smaller than the total number of bugs.
(we used to have "print" detect when it was running in ISR context, and revert to polled uart IO. Which made ISRs execute literally 100s of times slower than normal. It was still useful.)

1 hour ago by BillW

followup discussions *for lingering questions and comments*