Make a brief posting (150 to 250 words) on the **M5 Discussion Board** describing one of your most technical software debugging experiences. This experience may have been you participation in a software design review or conducting an independent review. What lessons and insights did you learn from the experience?

My most technical software debugging process was when I developed a C++ driver for a SPI transceiver on a Zynq7000 processor. When I began that project I had little experience in C++ except for one C class I had taken in undergraduate. It was certainly a “fake til you make it” experience. But that’s how great engineers are forged: in the fires of industry.

In any case, I had a difficult time establishing the correct order in which to instantiate the SPI configuration functions. In addition, there was very little documentation provided. The source of knowledge and truth came from the Xilinx EmbededSW github repository

<https://github.com/Xilinx/embeddedsw>

where a trove of useful examples including SPI examples are detailed. The SPI links can be found here

[embeddedsw/XilinxProcessorIPLib/drivers/spi/examples at master · Xilinx/embeddedsw · GitHub](https://github.com/Xilinx/embeddedsw/tree/master/XilinxProcessorIPLib/drivers/spi/examples)

While I was working on that code I used a logic analyzer to determine whether or not the SPI transceiver was actually sending signals. It was all great fun and once I got it working I remember thinking that I was going to get my MS in computer engineering.

One technique that helped me was creating a journal showing the screenshots of each attempt and the result. For example, if I were to make the following function call

XSpi\_CfgInitialize(XSpi \*InstancePtr, XSpi\_Config \*Config,UINTPTR EffectiveAddr)

Then I would capture a screenshot of the logic analyzer for analysis.