

How did you approach the project?

I started off by trying to create the functionality of the project without any of the bus/OS features and instead using built in python functionality more like how I would have organized it if I was trying to make this a practical project.

How did you organize the project? Why?

I created a separate file for each process, while there's a lot of places that repeated code could have been cut down, especially in the signalling section, I felt like it would be easier to isolate issues with the separation given that each part would be running in it's own process and I would have to figure out how to capture process output to a terminal or learn to use a debugger in order to get proper debugging info.

What problems did you encounter?

Working with the signaling interface in a direct way really led to a lot of confusion as to how different parts of the project were interacting with each other. When I initially read through the project I figured "pipes I know how those work" and didn't realize that it would involve transferring each byte individually along with a bunch of other signals.

How did you fix them?

Unfortunately I ran out of time in this case, but I would have considered making a short helper program that breaks out the signal bus into an object with all the individual signals clearly labeled and could also do the reverse.

What did you learn doing this project?

Despite working with python a good amount for various raspberry pi projects and whatever I had never tried to use various system functions (like pipes and executing other binaries) from within any language besides C++ and Bash. I suspect in the future a lot of my quick and dirty automations might trend towards python instead of something like bash because of it.

If you did not complete some feature of the project, why not?

Signal Bus

I wish I could have finished a lot more of this since it connects everything together and thus I can't actually show that much in terms of output for my work and I'm kind of working blind as to how functional some of the parts are. Like I mentioned earlier, making something that abstracts the signal bus completely would have made things a lot easier to debug but involved more upfront time than I had. I tend to work with a lot of obscure hardware on some of my personal projects and this is generally the first step I take, but I didn't think it would have been worth the effort for this. It might have been worth trying to convert some of my code meant for JTAG debugging since that standard operates in a similar manner to how the signal bus should work.