SE/EE/CPR E/CYB E 492 – Spring 2024
PrairieLearn Senior Design Team
Week 17 Report

March 22 - March 29

Faculty Advisors: Phillip Jones

Team Members:

Chris Costa - Auto-Drawing
Matt Graham - Emulator
Mitch Hudson - Technical Lead, ARM Assembly Auto Grading
Carter Murawski - Note Taker, Emulator
Tyler Weberski - Project Manager, Auto-Drawing
Andrew Winters - ARM Assembly Auto Grading

Summary for Progress this Week

- Configured emulator environment and added to git
- Set up the emulator to be able to run any standard C file
- Completed emulator usage writeup **(see the end of this document here)**
- Reviewed homework 8
- Reviewed homework 6
- Reviewed homework 11

Past Week Accomplishments

- Completed midterm peer review assignment
- Reviewed homework 12
- Looked more into the emulator
- Worked on HW5 and using GPIO code

Individual Contributions

Team Member	Contribution	Weekly Hours	Total Hours
Chris Costa	worked on implementing hw5 q1 and getting better interactiveness, continued editing hw9	5	95
Matt Graham	Got the emulator environment configured and added it to git under the pico-emulator-dev branch, set up the emulator to be able to run any standard C file, and created a writeup of how to use the emulator once you have it pulled from the repository **(see the end of this document here)**	5	98
Mitch Hudson	Wrote hw 6 and 8 Responded to peer reviews for hw 6, 8, 11 Linked PrairieLearn to Canvas environment	18	235

	Started working on QEMU emulation platform for Tiva TM4C123G microcontroller Started writing Python system for handling configuration register setup		
Carter Murawski	Worked on emulator environment, it can now run C files and helped Matt with the write up about how to set it up	6	88
Tyler Weberski	Reviewed homeworks 6, 8, and 11. On top of this continued working with GPIO code, making my own code work for the problem, still working on integrating	4	90
Andrew Winters			77

Comments and Extended Discussion

- N/A

Plans for Coming Week

- Work on getting the emulator working with more advanced C files, such as UART
- Continue to work through peer reviews and homework development
- Finish HW 5 and 9 peer reviews
- Work on QEMU emulation integration

Summary of Weekly Advisor Meeting

- Add lightning talks to website(slides with audio)
- Gave first demo of PL implementation
- Get last years questions
- Keep moving forard with PL question
- Improve communication and come to next meeting with more of a plan
- Try to find different meeting time

How to Setup The Emulator and Run Any C File

- Open Terminal
- Run “cd ~”
- Git clone https://git.ece.iastate.edu/sd/sdmay24-33.git
- Run “cd sdmay24-33”
- Run “git pull”
- Run “git fetch”
- Run “git checkout “pico-emulator-dev”
- Run “sudo apt update”

- Run “sudo apt install”
- Run “sudo apt install cmake gcc-arm-none-eabi libnewlib-arm-none-eabi build-essential”
- Run “sudo nano ~/.bashrc”
- Add the following line to the end of the file:
    - “export PICO_SDK_PATH=/home/*user*/sdmay24-33/emulator/pico-sdk”
    - Replace “*user*” with the account name you are using
    - Note: this command assumes you have cloned the sdmay24-33 directory in your user’s home directory as specified above. If this is not the case, modify the path accordingly
- Open Visual Studio Code
- Select “File”
- Select “Open Folder”
- Choose the “sdmay24-33/emulator/build-hex-files” directory from your filesystem
- Here, you can edit the “build-hex-files.c” file to contain any C code you wish to run
- Select “Build” from the bottom of the screen (gear icon)
- You will see a prompt for “Select a kit for build-hex-files”
- Select “GCC 10.3.1 arm-none-eabi” from the dropdown