



Gregory T. Ling

gling@ctrl.com

ctrl.com



Interests

Microarchitecture Design

- Integrated a Roomba with a foam dart turret, webcam, FPGA, Raspberry Pi, ESP32, bluetooth game controller, and Quest VR headset for CprE 488
- Started ASIC design club, designed a Verilog ASIC to submit for fabrication
- Pipelined MIPS32 processor in VHDL
- Very low-power intermittently-powered neural network accelerator
- TA for digital logic using Verilog, advanced embedded systems, and neural network microarchitecture design

Firmware Development

- Software lead for PRISUM Solar Car club (register-level embedded firmware using custom libraries)
- Embedded bootloader over CAN with remote programming ability
- Handheld game console with custom USB & SD card bootloader
- Intermittent power communication testbed for ISU research project
- Several custom embedded linux distributions using Yocto build system
- Custom and extensible build system to cross-compile using LLVM, Meson, and Ninja for all PRISUM software

Quantum Firmware

Theory of Quantum Computing at ISU covered nonlocal games in both classical and quantum computing as well as quantum teleportation, quantum information theory, and Kolmogorov Complexity. I have designed a small programming language and am very interested in working more in quantum computing,

Education

Iowa State University (ISU)

B.S. Computer Engineering
M.S. with Thesis in CPR E

Ames, IA

GPA: 3.99
GPA: 4.00

Fall 2020 - Fall 2023
Fall 2023 - Fall 2024

Project Highlights

ISU Solar Car Club (American Solar Challenge PrISUm #9)

- I've been the software lead of ISU's Solar Car club for three years. I guided a transition to a new build system which is focused on manual management of peripherals using the datasheet and CMSIS headers, encouraging programmers to read the datasheet and utilize the full capabilities of these MCUs. I also led the telemetry project which is an embedded linux system built in Yocto running a lightweight web server using C++ and Vue.JS to transmit the car's status over WiFi or LTE in real time.

Sparky! - CPRE 488 Open-Ended Design Project

- For our CprE 488 final project, my group decided to take a Roomba, attach a Digilent ZedBoard, Raspberry Pi, webcam, ESP32, bluetooth serial module, bluetooth game controller, foam dart turret, and CrazyFlie tracker to enable you to drive the roomba remotely via a Quest VR headset, shoot the foam dart turret, and enable automated target detection of a nearby drone

ISU - CPRE 381 Microarchitecture Design

- Designed a MIPS 32-bit processor in VHDL in a small group project over a span of four weeks, I specifically worked on the component integration between the adder, control logic, and the processor memory.
- Learning to implement pipelining, caching, and data forwarding to increase performance.

Personal - NXT/EV3 Infrared Remote Control "Sensor"

- Created an infrared remote control which connects to a LEGO NXT/EV3 as a sensor and allows control of the LEGO Power Functions system from the EV3 programming interface.
- Designed schematic and PCB layout using a TI MSP430 microcontroller and wrote firmware in C using PlatformIO IDE

Personal - Handheld Game Console

- Designed and assembled a handheld game console using an ATSAM21 (32-bit ARM Cortex M0+ on Adafruit Adalogger M0 breakout board) with an LCD display, SD card reader, analog joystick, battery and 3D-printed case
- Challenged myself to use only CMSIS header files and the datasheet; Pong and Brick Breaker currently work

Work Experience

IBM - Firmware Developer Intern

Rochester, MN

Summer 2024

Firmware development for the core hypervisor team on Power systems

Altec - Software Engineer Intern

St. Joseph, MO

Summer 2023

Yocto Linux development, zlib compression in C++, and various kernel hacking projects

ISU Research Assistant (REU)

Ames, IA

Summer 2022 - Spring 2023

Wrote firmware for a test setup of 5 MCUs to research wireless, intermittent, and batteryless communication

Skills

- Experience with a wide range of processors: TM4C, ATSAMx21, ATMEGA, MSP430, STM32, i.mx6, POWER.
- Read and write code based on a datasheet: I've written both a websocket server on a TM4C123 over a serial connection and a websocket client for the Solar Car telemetry system using the websocket specification (RFC 6455), and most of my firmware projects use datasheets as reference
- Git, Agile Development, Language experience in C, C++, ASM (8-bit AVR, 32-bit ARM, 32-bit MIPS, POWER), Verilog, VHDL, Swift, JavaScript (Browser & Node.js), TypeScript, CoffeeScript, Vue.js, HTML, PHP, CSS, Java, Python, UNIX, and more
- Laser cutting, 3D printing (FreeCAD, OpenSCAD, Fusion 360), PCB design (Eagle, KiCad, Altium), HDL (Vivado, Modelsim)
- California Seal of Biliteracy in English and Spanish