

# Matthew Geib

Cell: N/A | [Matt@mageiber.com](mailto:Matt@mageiber.com)

**Brief:** I currently live and work in Oregon as an Embedded Software Engineer developing software for embedded avionics, currently on the new product development team. My role includes creating embedded software for aviation displays and the related certification tasks for releasing the software.

## Education:

- **M.S.**, Electrical Engineering – Embedded Systems, University of Colorado, Boulder 2024
- **B.S.**, Computer Science and Engineering, University of Nevada, Reno 2017
- **Embedded:** Microprocessor system design, embedded game development, networks, RTOS, SOC, FPGA, Battery technology, alternative energy sources, embedded Linux, sensors
- **Programming:** Data structures, object oriented, algorithms, operating systems, computer vision

## Employment:

### **Garmin AT**

*Software Engineer II*, June 2021-Present

- **Software/Firmware:** Code (Embedded C, ARM) for embedded avionics spanning low level to application and UI in a custom/proprietary RTOS. Safety critical software that interfaces with other avionics units and sensors(Ethernet, CAN, UART, SPI). Utilize software concepts necessary for multiprocessor, multi-core, and multi-threaded capabilities. Root cause defects based on field observations, propose solutions, and implement fixes. New product development and current product software updates. Shipped two major software updates, one new product, and various minor updates.
- **Test:** Create and run test code to exercise embedded code as required based DO-178B and criticality level. Requirements and code coverage based testing. Unit testing and automated integration testing.
- **Process:** Derive software requirements from system requirements. Review code with respect to criticality level. Estimate tasks, features, and defects via project tracking software. Documentation in knowledge management/collaborative software. Git version control. Jenkins continuous integration. Mentoring interns and new hires.

### **Hyster-Yale Group**

*Software Engineer*, June 2018-June 2021

- **Firmware:** Low level code (Embedded C, ARM) for controller application level, bootloader, drivers, and communication protocols. RTOS (Keil RTX) and microcontroller debugging. Configuration of microcontrollers according to datasheets.
- **UI/Display:** Application level code(Lua) for forklift user display that interfaces to the main controller. User interface storyboard for the display and internal/external tool UI design(WPF).
- **Control Software:** High level control system software(Simulink/Matlab) for forklifts that controls vehicle and hydraulic movement, lights, sounds, error(fault) states, and communication with other controllers.
- **Tools:** Internal/external tool code(C#, C++, and Python) for interfacing to controllers via custom protocols, generating configuration data, embedded memory management, and auto-generation of code.
- **Build and Test:** Integration of various code bases with Jenkins for build automation, unit testing, and release pipeline. Automation of display software testing.
- **Documentation:** Gathering and writing requirements for both internal tools and external software releases to ensure business, team, and customer needs are met. Documenting software for easy maintenance and use. Defect and feature tracking via ticket system.

- Process: Agile and Kanban workflows. Software life cycle from requirements to released product support. Participation in code reviews and pair programming. Task estimation and prioritization. Mentored interns and new hires.

**Skills:**

- Programming languages: C/C++, Python, C#, Simulink/MATLAB, Lua
- Software: Visual Studio/VS Code, PyCharm, Git, Beyond Compare, Jenkins, BullseyeCoverage, Microsoft Office, KiCad EDA, FreeCAD, Lauterbach TRACE32, Keil  $\mu$ Vision, Simulink/MATLAB, Agile PLM, Adobe Design Suite, Xilinx ISE, Modelsim, Quartus Prime.
- Operating Systems: Windows, Linux, Embedded Linux, Mac OSX
- Embedded Systems/Firmware: 8-bit ATmega, 32-bit and 64-bit ARM, RTOS-Keil RTX and proprietary, CAN/CANopen, Encoders, I2C, SPI, UART, DMA, debuggers, oscilloscopes, digital logic analyzers, power profiling, FPGA.