PABLO A. MAURIN

PABLO@MAURIN.ORG

EMPLOYMENT

Avionics Firmware Engineer **Zipline International** **Feb 2015 – May 2016**

Avionics Firmware - Micrium µC/OS-II RTOS on ARM, C

- Developed firmware for a number of subsystems in unmanned autonomous aerial vehicles (drones)
- Worked closely with electrical engineers to bring up new board designs and board revisions
- Wrote board test procedures for technicians to ensure all avionics hardware is flight worthy before vehicle assembly
- Microcontrollers: TI Hercules TMS570 (Cortex-R4); Atmel ATSAM4E (Cortex-M4); STM32F427 (Cortex-M4)
- Took ownership and sole development responsibility for subsystems including, but not limited to:

Emergency Pilot RC Controls

- Developed driver to decode proprietary Futaba S.Bus serial protocol from RC receiver
- Implemented low level autopilot bypass for manual RC commands
- This was the first avionics code to fly in the critical path

Control Surface Actuators

- Wrote CAN driver for STM32F4 used by first iteration of actuator control module
- Created calibration routines to map autopilot commands to control surface angle and PWM servo output
- Developed Python scripts for CAN frame injection used in development and automated preflight
- This submodule is the last link in the control chain. All human and autopilot commands depend on this module to move control surfaces to the correct position

Power System Telemetry

- Wrote code to measure and report real-time power telemetry at system and board level (all power rails)
  - ADC for voltage and current measurements
  - I^2C for external temperature sensor measurement
  - GPIOs for failover power bus status
- Discovered design flaw on failover status lines and worked with electrical engineers to fix in next revision
- Real-time measurements communicated to ground tracking station and saved to flight logs for analysis

Navigation Sensors

- Contacted vendors and selected GPS/GLONASS modules and antennas to meet performance requirements and design constraints (low mass, high resolution, 10Hz+ update rates)
- Implemented high precision RTK GPS/GLONASS infrastructure
- Augmented existing ground to vehicle communication links (vehicle, base station, and backend servers) to support GNSS corrections
- Wrote RS-232 serial driver for Novatel GPS/GLONASS receiver module
- Ported vendor supplied INS module source drivers from POSIX threads to µC/OS-II, resulting in code reuse and quick integration of sensor module
- Both GPS and INS sensors are critical inputs to flight control loops

ALL of these subsystems are currently flying, and Zipline would not be in the air if not for my contributions

Software Engineer **Applied Minds** **Sep 2005 – Aug 2014**

Image Processing - GNU/Linux on ARM, C

- Worked closely with electrical engineer to port stereo image algorithms to Xilinx Zynq SoC (ARM A9 + FPGA)
- Implemented a zero-copy device driver for image capture and streaming to FPGA accelerated functions
- Transmitted computed results over gigabit ethernet to downstream compute nodes
- Video processing pipeline latency cut in half, frame rate doubled, spatial resolution quadrupled

Robotics (multiple projects) - GNU/Linux on x86/x86_64, C / Python

- Architected software framework for sensor (LIDAR, GPS, INS) data, compute nodes, and physical actuators
- Implemented using shared memory protected by Multiple Reader One Writer (MROW) locking
- Enabled zero-copy messaging and parallel execution for downstream nodes
- Authored tools for logging and playback of recorded and simulated missions
- Designed and implemented dynamic routing system for automatic selection of 900Mhz, 2.4GHz, and hardwire links. This enabled reliable communication at over 2km and fast multi GB log file downloads

Hardware Drivers- Android OS on ARM, C++

- Developed driver for supplemental capacitive touch device built by electrical engineering team
- Driver implemented on bare Android Linux, allowing all Android apps to work with new hardware
- Produced novel intellectual property for client portfolio

Image Processing on GPU - GNU/Linux on x86_64, C++ (QT)

- Developed low latency real-time video processing pipeline for a color night vision system
- OpenGL Pixel Buffer Objects (PBO) used to eliminate a memory copy operation and reduce latency
- Performed image processing on GPU using OpenGL Shading Language (GLSL)
- Achieved 60Hz real-time colorspace processing and image fusion
- Patent Pending: Application US20150350629

EDUCATION

Miami, FL **Florida International University** **Dec 2002**

- B.S. in Computer Science with Minor in Mathematics, Graduated *Cum Laude*

Languages and Technologies

- Programming Languages: C, C++, Python, Shell scripting
- Compilers/IDEs: IAR, Visual Studio, Eclipse, GCC
- Target Platforms: GNU/Linux, Micrium µC/OS-II, NuttX, Atmel SAM4E, ST32F4xxxx, AVR
- Human Languages: English, Spanish