Plamedi Luis

+2774-934-9868 — plamedi.luis11@gmail.com — linkedin.com/in/plamedi-luis-385353160 — github.com/Kapongoboy

Summary — Plamedi is an Electronic Engineering graduate from the University of Pretoria with a strong passion for embedded platforms and the software industry. A quick learner with a thirst for knowledge, he thrives on challenges and continuously seeks opportunities to expand his expertise.

Proficient in C/C++, Python, Rust, and Linux, he is also expanding his knowledge of Java through courses. He has hands-on experience in embedded systems development and has collaborated on team-based university projects, including the digital filtering module on a digital transceiver, GPU device programming for a Non-Intrusive Monitoring System, and wire-based UART communication in VHDL for the NLFSR with Chaos Generator.

Plamedi gained industry exposure as a QA intern at *AEC Electronics*, where he tested and repaired electronic detonators and utility devices (*BO-01, CT-01, CI-01, etc.*) to ensure reliability and performance.

Skills

Languages	C/C++, Python, Rust, VHDL, SQL, TypeScript, MATLAB, Octave	**Machine Learning**	PyTorch, Tensorflow, Tinygrad, CUDA, Roboflow
Web	HTML, CSS, React, Flask, Supabase, NextJS	**misc**	Automation, Data Structures, Research
Tools	Linux, Git, Github, OrCAD, LTSpice, Simulink		

Projects

Autonomous robot vacuum cleaner with computer vision

– Main platform was the Odroid N2+, using UART to communicate to an STM32F429I-DISC1.
– STM32CubeMX used for initial C code generation for STM32F429I-DISC1.
– arm-none-eabi-gcc used to compile the STM32's program using make.
– st-link (provides st-flash) was used to flash the embedded device.
– openocd and gdb were used for live debugging the STM32F429I-DISC1 on a Linux host device (EndeavourOS).

Non-intrusive monitoring system

– Yolov3-Tiny was implemented in CUDA/C++ running on the Nvidia Jetson Nano embedded platform.
– The main program was written in Python and communicated with the detector program through UNIX sockets and shared memory.
– An OSNet CNN was implemented for re-identification feature extraction in Python (NumPY).
– Libfreenect was used to interface with the Microsoft Xbox Kinect sensors (for RGB-D image capture).

NLFSR with chaos generator

– DE0-Nano-SoC development board programmed with VHDL ran the main NLFSR operations.
– Quartus-prime used for programming FPGA platform.
– ModelSim-Intel used for FPGA system simulation and verification.
– 16-bit synced data communication between platform and host device to enable testing and verification.

Modulated morse-code transceiver

– Two STM32F429I-DISC1 were used, for transmission and reception.
– STM32CubeMX used for initial C code generation.
– ARM CMSIS-DSP library used for FFT functions.
– STM32 HAL used for graphics programming of ILI9341 (tft lcd driver).
– ADC/DAC functionality setup using STM32CubeMX.

Line following robot

– 8-bit PIC18F45K22 microprocessor programmed using PIC assembly.
– Initial setup of peripherals and clocks done through MPLABX graphical pin-out interface.
– Interfaced with external EEPROM IC for expanded persistent storage.

Respiratory sensor system

– 8-bit PIC18F45K22 microprocesser programmed with PIC C.
– MPLAB Code Configurator (MCC) used for initial C code generation.
– MPLAB XC8 compiler used on Windows host device.
– All algorithms interacting with hardware interfaced using PIC HAL.

Education

University of Pretoria
Bachelor of Engineering in Electronic Engineering

Experience

QA Intern   **December 2022 - February 2023**
AEC Electronics

- Executed comprehensive manual testing procedures for the CT-01, CI-01, BO-01 and other products using in house test jigs according to design specifications.
- Reviewed and optimized testing protocols to enhance efficiency and effectiveness, including submitting a report for the update of a historic products test specification.
- Collaborated with repair teams to diagnose and address defects using oscilloscopes, multi-meters and debuggers. As well as working hands on to re-solder and repair light physical defects.
- Maintained meticulous records of test results and device histories, streamlining future audits, and storing data categorically and efficiently.