[Experience]

[Describe a skill or knowledge you acquired recently that has been impactful for you. Why did you make this investment? What has the outcome been?]

Well, it is not technical. I just got my pilot license for Ultralight flying. I did this investment because it was one of my chilhood dreams and I wanted to get into a aeronautical company. And I got an offer as a Software Architect.

[What new skill would you like to learn? Why do you think this is important or timely or interesting? Why do you think you will be good at it?]

Computer Vision. It is where my interest lies within. I strongly believe that it will be a critical skill to have in the future.

[Describe your experience as a developer on embedded Linux.]

I have 7 years experience in the automotive industry in Germany , where I used embedded Linux in all developments such as communications or autonomous driving.

[Describe your experience with Linux kernel development and debugging.]

Unfortunatelly I do not have such experience.

[Describe your experience of low-level boot processes and BIOS / firmware.]

I have 6 years in developing firmware for ARM Cortex M3 processesors. it was my first job.
Level of understanding is from understanding design schematics (Last tool used is Eagle) to writing device drivers (Display, AD converter, keyboard). RTOS was also a task to use for custom aplications. Writing device drivers implied reading and understanding the processor manual and writing&adjusting the drivers within the RTOS.

[How do you address software performance in your coding practice?]

Softare performace is adressed by the following : Coding, Testing and Documentation.

*Coding
- Object Oriented environment performance - software performance is achieved by using Design Patterns. Sometimes, depending on the programing language (C++ templates), software performance can be improved by metaprogramming level, speeding up also compile times.
- low level environment - use only data structures which can be mapped directly on the processor design. Such examples are for instance: data types used by the ALU unit, to the data used for Transer between interfaces SPI, RS232, ..., memory Word length within Flash, ...

* Tesing - Unit-tests and code coverage. End-to-end tests.
* Documentation: Some form of Architecture Documantation (PlantUML), System and Test Documentation, Code documentation (Clean code manners)

[How do you prefer to drive documentation for your products?]

Documentation lies within the art and form how one writes the code ("clean code a handbook of agile software craftsmanship" as a reference). Also, at a block level of understanding timings and Architecture interaction, one form of UML I personally use.

[How do you think about and ensure quality in your software products?]

By providing a Chain of Testability, Code Architecture and Documentation.
- Testability - Unit-tests with a high degree of code coverage. The mode must be design in separate modules in respect to the Design Patterns, to avoid dependency as much as possible between modules. Another level of Testability is End-to/End testing, by testing the Software product as a stand/alone entity (binary for example) or black-box model. At this level inputs are simulated, and outputs results are tested.
- Architecture design - use known practices for developing the code -> object oriented Design Patterns, clean code manner of writing.
- Documentation - Requirements -> System design -> Architecture Design -> Testing.

[Describe a case where it was very difficult to test code you were writing, but you found a reliable way to do it.]

- In a templated class definition - All types are templated. I used mock classes, at a abstract level to test certain callbacks.
- In a multithreaded environment - Where timings are importand in a certain order. I test fixtures for defining an abstract model to test the environment.
- End-to-End - When the product has a lot of dependencies, which require third party applications. Finding a way to simulate the commuinication inbetween apllication is difficult.

[Describe your C/C++ software development experience to date. How would you rate your competency with C/C++?]

I have 6 years of experience in C++17. As the C++ community prgressed towards other C++ verions, I am limited on the level of libraries and Versions of C++. I cannot see myself at a higher experience as this community always thrives in progress. It is like compairing double data structures on a processor ;)

[Describe your experience in Golang and Python.]

I used only Python for tests. At a test level only.

[Which Linux middleware/user space stacks are you the most familiar with? For example gstreamer, libinput, audio subsystems etc]

I only understand the concept of middleware. Such as DDS, ROS, SomeIp. I am experinced in using this protocols between process communication.

[What interesting syscalls are used by the “uname” binary? How did you find out?]
-

[Have you created .deb or .rpm packages? Please describe your experience with Linux packaging.]

No. I only used yocto for adjusting recepies.

[What kinds of software projects have you worked on before? Which operating systems, development environments, languages, databases?]

- Projects - Industrial Scales. Starting from Product design to product development. 10 years.
            - Car manufacter. Inside and outside car communication. Autonomous driving, sign recognition.

- OS - Mostly Linux. Open source and toolchains. VSCode,Docker, Git, PlantUML, Jenkins ...
- Langages - Asembler, C, C++, Java, Delphi, Python.