# Applicant Questionnaire: Software Developer

## Instructions

For some of our positions, we refer to portfolios or code samples when determining if interested people will fit with our team. We have found that some people's previous experience doesn't lend itself towards the creation of an online presence, so we have prepared a brief questionnaire.

Please answer the following questions to the best of your abilities. Complete and email your response to [klee@clearpath.ai](mailto:klee@clearpath.ai). Good luck; may the force be with you!

|  |  |
| --- | --- |
| **Applicant Name:** | Rose Vayampothanal Thomas |

|  |
| --- |
| **Our main programming languages at Clearpath are Python and C++. What factors would you consider when choosing between these for a given project?** |
| 1. If the application is time critical, I would prefer C++ 2. If the application needs to run on cross-platform, I will choose Python 3. For scripting and automation, python would be the first choice |

|  |
| --- |
| **What is the most interesting problem you had to resolve through specialized tools, whether in-house or commercial? How were they used to explore and resolve the issues? Tell your debugging war story.** |
| I would like to explain a recent story. My project was running on Classic CAN protocol for several years. And we got the requirement to replace the protocol with CAN Flexible Data-Rate. Since our previous microcontroller would not support the new protocol, we were forced to replace the microcontroller with another one within the same Renesas family. As a result, all the registers have been changed. After the code development for the new protocol, we tried to test on the new board with replaced microcontroller, but alas, that was not an easy job. Initially, I tested the hardware using oscilloscope to know whether at least any sort of transmission occurs. I could see from oscilloscope that pin configuration is correct, and when I tried to start the vehicle communication using CANOE simulation tool, the board is waking up as expected in response to the transmitted message. So, I have concluded that hardware has no issue. Next, I tried debugging the Software using Renesas E1 debugger. After some struggles, I got to know that, the registers are not configured correctly in order to support CAN FD. Those were still configured for Classic CAN. Also, the APIs has been changed, so we ended up reintegrating the software modules to the existing system. |

## Team Project:

Please focus on a recent project you completed as part of a larger team.

|  |
| --- |
| **Please describe your contributions to the project. What components were you personally responsible for, and what were their interfaces to the other components? How did those interfaces get defined?** |
| I am currently working for the communication network section of Instrument Cluster Electronic Control Unit (ECU). Cluster consists of 2 microcontroller – one for basic Software and another for Graphics. Both controllers are synchronized using SPI. Cluster communicates with other ECUs using CAN, LIN and Ethernet networks. I am the Scrum Product Owner and Lead Software Engineer for Vehicle Network Team. My job involves both team management as well as development. As a Product Owner, I act as a bridge between my team and project management. Since the entire project team consists of 100+ members and are distributed across the globe in different time zones, it is indeed important to coordinate between different stakeholders and customer. I create tickets for the team as well as decide the priority for the tasks.  As the lead developer, I work on the design, development and testing of vehicle network and diagnostics related application modules. I do the review for other colleagues and mentor them as well. |

|  |
| --- |
| **What development environment and programming language did you use? What costs and benefits did this choice have to the overall project outcome?** |
| Currently I am using MSVC 2013 IDE and the language is C++. Since the instrument cluster ECU is safety critical, C++ is a faster solution. The debugger is Renesas Multi and it is user friendly. |

|  |
| --- |
| **What steps did you take to future-proof your code? Were there areas where you (and the team) chose specifically not to (or not to attempt to) future-proof? What was the reasoning in each case? Is there an example where this strategy paid off, or you got bitten?** |
| As I explained above, since future proofing was not done properly, we ended up in a mess while we replaced the protocol. 2We grabbed it as an experience to learn and we have decided to add an abstraction layer between driver code and application. This would save a lot of effort in case if we want to replace any peripheral driver or even microcontroller itself. The idea is to reuse the application modules above abstraction layer with minimal effort. |

## Individual Project

Please focus on a project completed individually (or mostly individually). Side projects, open source contributions, stuff for school, and stuff for previous employers are all valid.

|  |
| --- |
| **What was the project? What was the motivation and what did you accomplish? How did it turn out differently from your plan going in?** |
| I would like to describe about the Programmable power supply (PPS) project. The project was not so big, so I did it alone. PPS is a 3-phase programmable power source for generating 3 phase voltage and current with phase angle of Unity Power Factor, 0.5 Lag & 0.8 Lead. In order to do this project, I relearned all the fundamentals of electrical engineering. Most of the coding was related to complex electrical calculations and hence it made me strong in using mathematical equations in coding. Another important thing that I learned from this project is on how to reuse the maximum code from an existing Software. I reused the Software components from another project whenever possible. |

|  |
| --- |
| **What is something that you know now that could have helped you with this project if you had known it at the time? Please focus on technical knowledge rather than personal development (eg, time management).** |
| Since I did the above explained project in the initial days of my career, I think I could improve the implementation a lot.  Firstly, I considered the entire requirement as a whole big requirement. Now I know I could fragment big requirements into small tasks and proceed with the development and testing step by step. It would be always easier to manage well planned small tasks with a clear timeline.  Another important thing that I learned is to do unit/module testing in between. DO NOT wait till the end thinking to do testing in the last.  Third thing that I learned is that never underestimate the power of a debugger. In fact, debugger is a heavenly gift to any embedded software engineer. |

|  |
| --- |
| **How did you perform system testing and verification?** |
| I have done unit testing, module testing as well as system testing for different projects. System testing is a grey-box testing wherein I would test the system in a full framework that consists of complete software code additionally all real-time operating system (RTOS) and platform-related pieces such as interrupts, tasking mechanisms, communications and so on. For the system test plan, I would refer the requirements captured in DOORS. Each functional requirement will have a dedicated system test case. I would reuse the test plan if possible, from other projects and adapt for the new project. System testing is a mix of automated and non-automated test cases. |

## Lightning Round

|  |
| --- |
| **What is your favorite network protocol debugging tool and why? If you don’t have one, what might you try out next? Where would using one be effective in troubleshooting a problem with an application or service?** |
| E1(along with Multi) and CANOE are my favorite debugging tools. I love E1 because I could put breakpoints and edit the breakpoints to print, calculate time period (based on Software timer). I could get the trace history as well which help me a lot to understand the code flow. Vector CANOE simulation is the networking tool that I use for testing and debugging. It gives report on the bus statistics, I could see the messages and signals real time, and I even get the error report on hardware. Debugging tools are useful for both application and services. |

|  |
| --- |
| **Have you ever reverse engineered a protocol or some other aspect of a piece of unknown hardware or software? What was your process and what did you discover?** |
| There were a lot of redundant PCBs with different functionalities. I had collected them, tested each PCB whether it works or not, then made a test rack, inserted the working boards into the rack, made connections using wire, made a cover, and finally I turned those scrapped PCBs into a very useful and handy test box for my team. In the process, I got exposure to PCBs, how to test whether they work or not, how to use schematics and layouts, how to make a harness and connectors in a neat and aesthetic way and so on. However, I would consider learning about different types of motors as the best part of that task |

|  |
| --- |
| **Have you ever successfully evangelized a tool or practice to a team you were working on?** |
| As a product owner, I had introduced knowledge sharing session within team once in every 2 months. The topic should be related to the work. We nominate a colleague for each session, and everyone would get a chance. This helped to improve the bonding and cooperation within the team and helped in our work as well. |

|  |
| --- |
| **What's the most exotic programming language you’ve encountered? What made it unique, interesting, or particularly well-suited to a given problem domain?** |
| My personal favorite is C. I learned it in my college days and embedded C is helpful for many driver related codes. In my opinion, both C and C++ are the best programs in embedded software domain. |

|  |
| --- |
| **What is your salary expectations for this role?** |
| 70,000 CAD |

|  |
| --- |
| **What is your favorite robot, real or fictional, and in one sentence tell us why?** |
| Robot vacuum and mops because it makes my day to day life way easier. |