Software Engineer Technical Assessment

RRL-SETA-0225

Instructions To Candidates

Time Limit

- There is no fixed time limit for this assessment.
- Take as much time as needed to ensure that your submission accurately represents your knowledge and skills as an engineer.

Support Files

- This assessment comes with a compressed folder containing all necessary support files. If you have not received this folder, please contact Ross Robotics.

Submission Format

- Submit your completed assessment as a compressed folder (.zip, .tar.gz, etc), containing only source files and any required supporting files.
- Your responses should be written in C++.
- Supporting files may include:
    - CMake files (if applicable)
    - Diagrams
    - Datasheets
    - Instructions for building and running your solution
- Each response to a given question should be placed in its own folder within the main directory.

Use Of Generative AI

- While the use of Generative AI tools (ChatGPT, Github Copilot etc) is not expressly prohibited, you are encouraged to produce your own unique work. Extra credit will be given for your ability to explain and justify your design choices.

Assessment Structure

- Section A: Programming Paradigms
- Section B: Code Refactor
- Section C: Systems Design

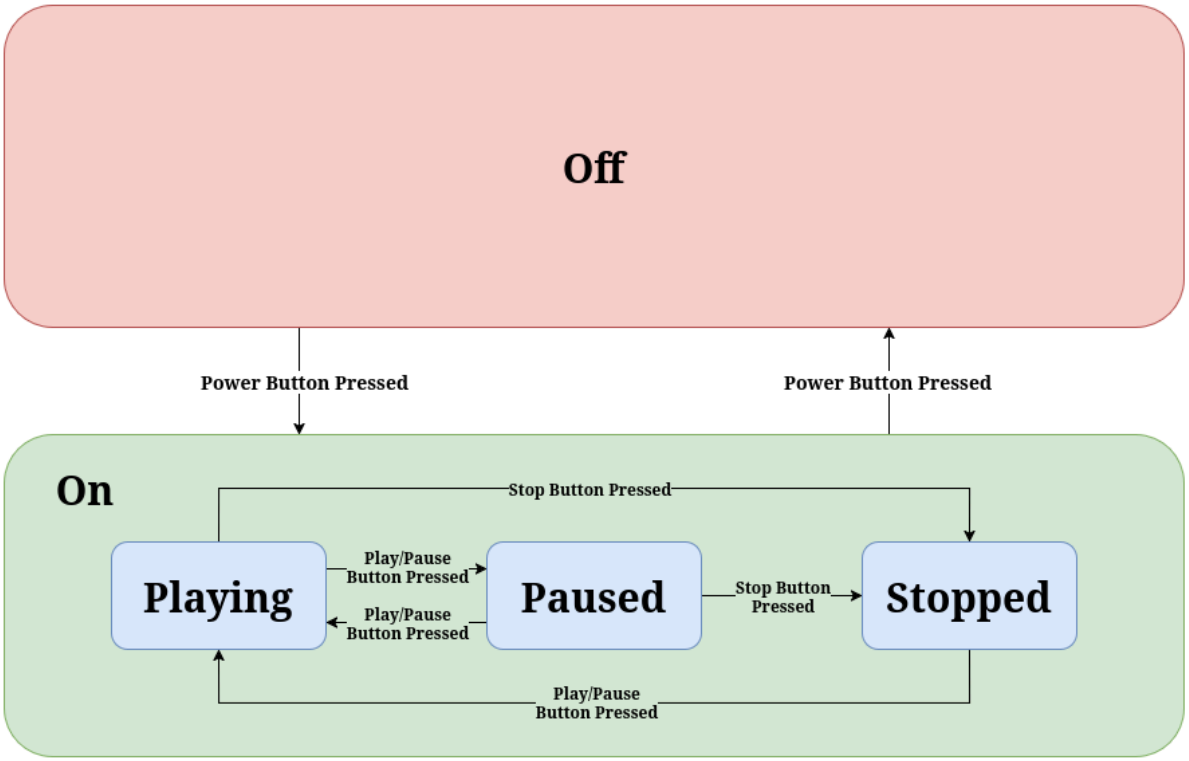
Section A: Programming Paradigms

Implement, in C++, the control logic for a media player device. The device can be either On or Off. When the device is switched on, it can be either: Playing, Paused, or Stopped. The device should respond to the following inputs:

- Power Button Pressed: Toggles the device between Off and On.
- Play/Pause Button Pressed: When in Stopped or Paused mode, the device should transition to Playing. When in Playing mode, it should transition to Paused.
- Stop Button Pressed: When in Playing or Paused mode, the device should transition to Stopped.

Your implementation does not need to handle audio - we are specifically interested in your approach to implementing the control logic. Inputs should be processed from input to the terminal.

Ensure that the system prevents any undefined behavior. Additional credit will be awarded for implementations that are structured to allow for easy extension and integration of new states.



Section B: Code Refactor

In this assessment, we want to verify your skill in reading and understanding code, as well as relating it to physical systems. After all, we work with real robots. Treat this code as a general base for a virtual robot's functionality, but feel free to refactor it as much as you believe is necessary. The goal is to ensure it is deploy-worthy and can be maintained for years to come.

Additional instructions can be found in the file *b_code_refactor/README.md*.

Section C: Systems Design

In this section, we want to assess your system design knowledge and skill. Please update the *system_overview.md* file with a high-level design of a full-stack robotic system, covering everything from hardware interfaces to the Graphical User Interface. Modify the document as needed to ensure your system architecture is well-defined and clearly structured.