MISRA C REPORT

In the MISRA C Report, we've come across several warnings that warrant attention. First and foremost, there's a warning flagging the non-utilization of return values from functions with non-void return types. It's important to note that we've consciously chosen to disregard this warning. The reason behind this choice is that these particular functions have a unique role within our codebase. They are primarily employed for the validation of code changes and debugging purposes, and they do not play a functional role in the operational code. Consequently, their return values are of no consequence to the actual functionality of the program.

Another warning pertains to incomplete parameter analysis within the code. This warning suggests that our code inspects parameters only within "if" conditions but neglects to do so within the corresponding "else" conditions. However, it's essential to clarify that our deliberate approach aligns with our design intent. Our design philosophy and coding standards guide us to structure the code in a specific manner, and this warning, while valid, does not necessitate a change in our approach. We believe that our current implementation adheres to our established design principles and functional requirements.

Additionally, there is a warning regarding external linkage for transitional functions or objects. The typical suggestion accompanying this warning is to segregate these functions into distinct files and call them from the main function. Nevertheless, we've chosen to diverge from this conventional approach. Our preferred methodology revolves around consolidating all functions within a single C file. This organizational choice aligns with our broader objectives of maintaining code simplicity and organization, even if it means forgoing the conventional practice of external linkage.

Lastly, there's an acknowledgment of a warning concerning the usage of basic numerical integer types without typedef declarations prior to initialization. It's important to understand that our coding standards permit and even endorse this particular practice. This decision is in perfect harmony with our established coding practices and project requirements. We've thoroughly considered the implications and made a conscious choice to adhere to our existing standards.

In summary, while these MISRA C warnings have been duly noted, our decisions to disregard or maintain certain practices are rooted in our project-specific design philosophies, coding standards, and organizational preferences.