CS-350-T3346 Emerging Systems Architectures & Technologies

4-2 Journal: Best Coding Practices

Eric Wallace

Jan. 20, 2023

As I started to research best coding practices in EmbeddedC there was two names that kept showing up, one of which is the ESCR (Embedded System Development Coding Reference) Guide. This guide complies with the ISO /IEC 25 010:2011 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) (Software Reliability Enhancement Center, 2018) which is the second name I seen referenced on search results. So, I started reading the article referenced below and it started making sense. According to this guide, there is four characteristics to which coding practices should follow and they are:

**Reliability:** Reduce the number of malfunctions code causes on the system and minimize the impact of bugs created by code have on other areas of the system. Examples of reliability is:

* Use types that can represent the data
* Pay attention to pointer types.
* Describe initializations.
* Do not perform recursive calls.
* Pay attention to order of evaluations.
* Check for restrictions when functions are called.
* Describe how conditions are to be handled when they do not follow predefined conditions.

**Maintainability:** Code should be easily read and modified, a developer should be able to add new functions or find existing function very efficiently. Examples of maintainability is:

* Do not write code in a cryptic manner.
* Do not reuse names.
* No magic numbers.
* Describe statements even if they aren’t compiled.
* Do structured programming.
* Write functions that differ in purpose differently.
* Unify coding styles, comments, and naming conventions.
* Use the same style for writing declarations, pointers, and directives.
* Be careful when using dynamic memory allocations.

**Portability:** Create code that can be move from one environment to another with little no errors. As the number of embedded devices being used increases, being able to take code and apply it to multiple devices with very little errors is paramount. Below are some examples of how to do this:

* Write in a style that does not depend on the environment to which it is compiled.
* Only use characters and escape sequences defined the programming language.
* Localize code that has issues with portability.

**Efficiency:** Performance relative to the number of resources used under certain conditions. Code must consider the memory size and resources available to the device.

* Code must be written thoughtfully with consideration as to how much memory is available. (Creating code where the file size is more than memory on the device)
* Do not write multiple functions that process the same data and produce the same result.
* Use the correct statements for the context in which it is written. (In other words, don’t use a switch statement when you should be using an if statement)
* Do not write code that consume large amounts of processing time.

Common pitfalls in writing code are a whole lot easier to find than the best coding practices, in my opinion, mostly since practices and standards are more subjective than pitfalls. Pitfalls are pretty much given because when code produces a bad result, well, what caused it. Below is some common coding pitfalls:

* Using meaningless expressions or statements that are not executed which causes confusion or errors in the code.
* Write code that is not executed.
* Write and using statements whose results are never used.
* Values passed as arguments are never used.
* Incorrect use of operators in logic conditions
* Declaring a function type that returns a different data type of that function.
* Reference or update ranges outside the bounds of an array
* Using the incorrect address of an automatic variable
* Accidentally overwriting string literals
* Using bitwise operators and logical operators incorrectly.
* Defining the same name multiple times

Coding practices are endless and change depending on who wrote them. The best practices I showed above are specific to Embedded C and use the ESCR (Software Reliability Enhancement Center, 2018) for the writing of this journal. Another popular coding standard is the set of standards created by the Barr Group. In my opinion, they all have the same goals just different roads taken to get there.

References

Barr Group Software Experts. (2018). *Embedded C coding standard*. Barr Group.

Software Reliability Enhancement Center. (2018). Embedded system development  
coding reference guide [Abstract]. *3* 16-17. <https://www.ipa.go.jp/files/000065271.pdf>