

Motivation

The software that will be on the device will be very large (several hundreds of lines if not into a thousand or so) and will require the merging of several devices/functionalities. There will also be a need for calling of variables, *et cetera*, all throughout the software. On a different level, the software *must* be readable and understandable. It is not good enough to just have it working if nobody, but you knows what is going on or why. This document will detail the general guidelines for any code written that is to be included in the main project software. Any code not found to hold up these standards will not be accepted (*see [GC.2.a](#) for more motivation*). If there are any questions about these guidelines, and/or what qualifies, just ask (with a screenshot, *et cetera*).

Naming/Referencing

1. [Camel Case](#) should be used for all non-keyword names. (Upper/Pascal variation, i.e. CamelCase)
2. [Snake case](#) should be used for all variables using numbers or variables that are modifications of another variable. (i.e. or CamelCase_1 or CamelCase_mod)
3. All functions (“function blocks”) should have a comment explaining the general purpose of the function on the first line of the function.
4. Most lines should include a comment explaining their purpose.
 - a. If the line of code is not obvious, then there should be a comment. It is easier to delete comments than to add them.
 - b. If there is a line that is obvious (i.e. an incrementor in a for loop) it does not need to be commented on unless the line is unusual.
5. Lines that pertain to a similar function (i.e. to handle the LED display) need to be grouped together, separated with some sort of line (i.e. 3 lines of “%%%%” across the screen), and labeled clearly.
 - a. When scrolling through the code the label should be easy to see. Once you give the code to me I will make it uniform in placement/structure, but I need your sections to be easy to format.

Citations

1. **Any** code taken from anyone, *even if modified*, needs to be cited.
 - a. Cite the name and location found in the code.
 - b. Cite the name and location found and/or URL link to the location.
 - c. This is not only for future maintenance needs, but also to not plagiarize.
2. This project will be under the [MIT license](#) structure.
 - a. The primary is because [GitHub](#) explains that it is good for open source projects such as these.
 - b. The second reason is due to convention. If the software is using a piece of code that is under the MIT license the software should be put as under the same license. This is not a must, but in the spirit of an open source community we shall follow this custom.

General Comments

1. Do not use code that is hard to intuitively understand unless absolutely necessary. It only makes it harder for those working on/modifying this software in the future.
2. Do not leave useless comments.
 - a. I was logging logic pulled from an on-site controller and there was a comment that read, “The new vibration guy said this is ok.” Which was dated in 1995! Who is the “new vibration guy?” Why did that change require the “new vibration guy” to approve it? What dummy left that comment? Nobody will ever know. Do not be that person that puts a comment like that (I will kick back your section and tell you to redo it).
3. Your comments should allow anyone, which is reasonably exposed to some form of coding, to “read along” with the code and understand what each part is supposed to do.
 - a. This allows the code to be checked/trouble shoot-ed faster, more easily, and by more people.
 - b. This means if you show someone your code you should not have to speak for them to be able to understand at least what you are *trying* to do.
4. The reason for including all of these comments, *et cetera*, is due to people being careless.
 - a. Most people throw away (or “lose”) instruction manuals, documentation, the papers titled, “DO NOT THROW AWAY! THESE WILL SAVE MAN KIND!” that you paid to have printed, and almost everything else you give them given enough time.
 - b. They do not care, so it is up to the creator(s) to ensure that some level of documentation is left in a place that the client will not got digging through and clearing out.