MISRA (Motor Industry Software Reliability Association) là một guideline cho việc phát triển phần mềm ô tô sử dụng ngôn ngữ C (hiểu đơn giản là rule khi viết code)

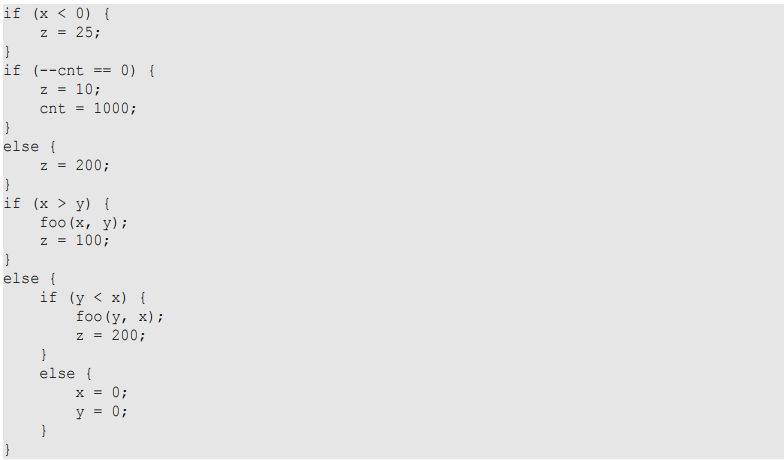
1/ 1 lần tab tương đương với 4 space

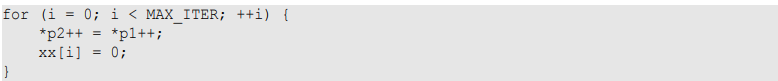
2/ Space cho hợp lí và sử dụng {} cho các lệnh if, else, while, for,…

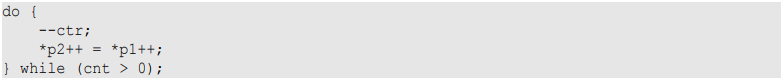
Vd: Ta không viết while(…){ mà viết là while (…) {

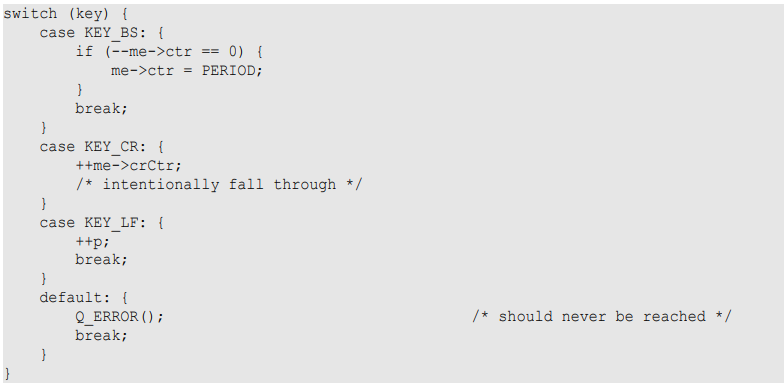


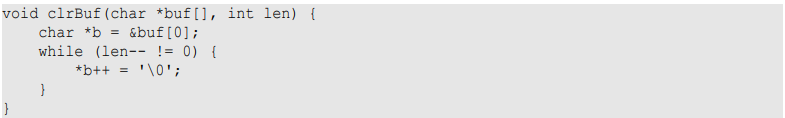


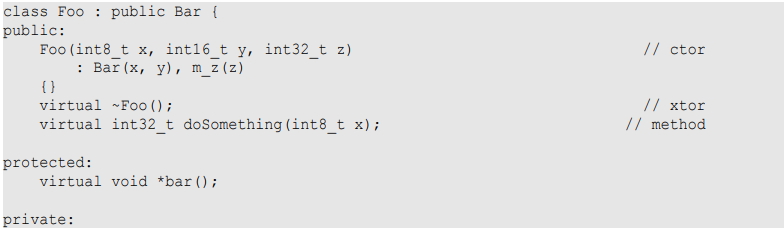
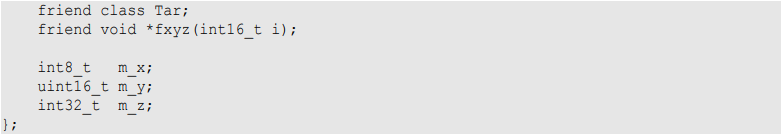




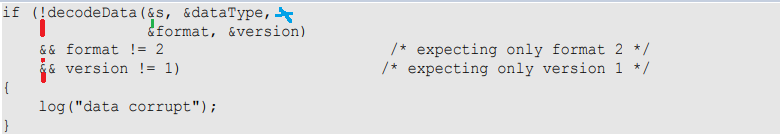








3/ Khi điều kiện trong các cầu điều kiện cùng phân hạng nên đặt thẳng hàng nhau và nếu điều kiện của 1 phân hạng nào đó quá dài thì cân nhắc xuống dòng cho hợp lí và nhớ là dấu “,” phải nằm ở cuối cùng trước khi xuống dòng ->> lúc này {} sẽ có qui tắc khắc với vd ở trên



4/Comment: phân làm 3 mức độ

- Comment có độ phức tạp cao: dành cho những thành phần software chủ chốt như:

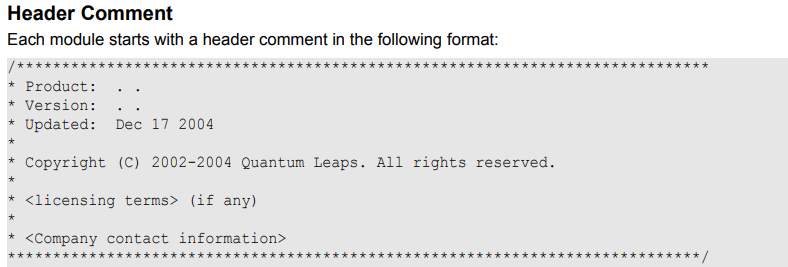
+ modules

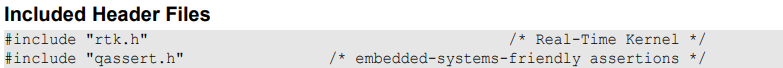
+ classes

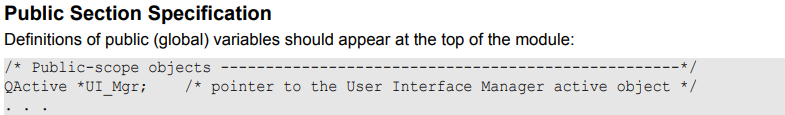
+ exported APIs

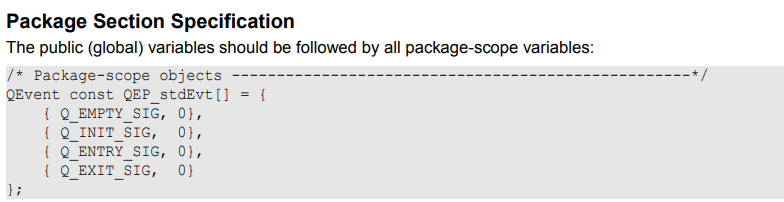
- Comment ngắn gọn xúc tích để giải thích chi tiết về thuật toán của operations thực hiện

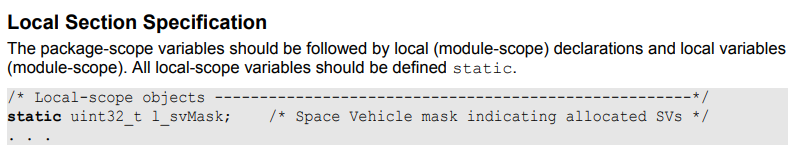
- Comment tài liệu tham khảo và chú thích đính kèm theo code



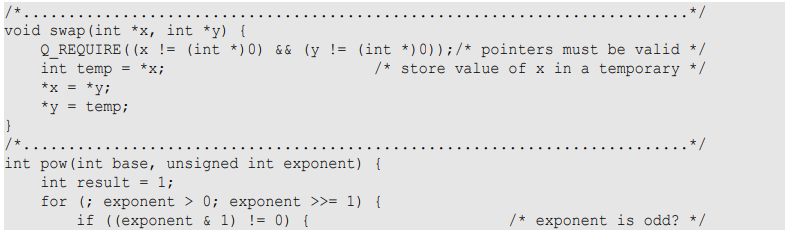
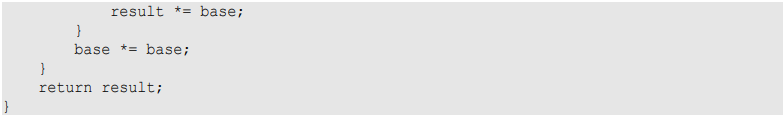


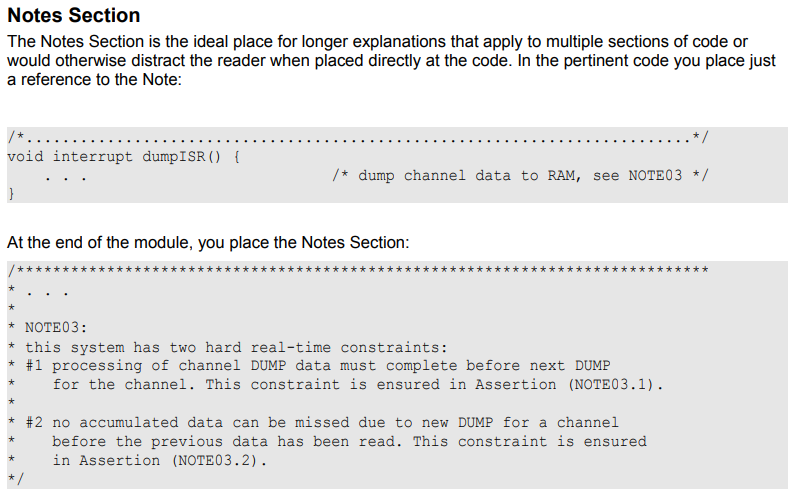




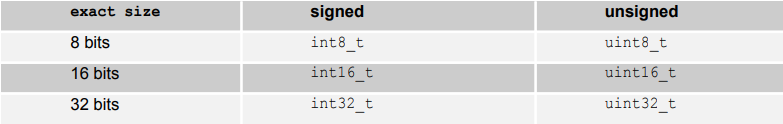








5/ Typedef Data\_Types:



6/ Quy tắc đặt tên:

- Không bắt đầu bằng số (0->9) hay dấu “\_”, không sử dụng các kí tự đặt biệt (ngoại trừ “\_”)

- Tên biến không được dài hơn 31 kí tự (MISRA-C rule 11)

- Không được viết hàm trực tiếp mà phải sử dụng nguyên mẫu hàm, các nguyên mẫu hàm nên đặt ở 1 khu (thường là ngay trên int main(void) {)

- Đối với “Struct, Class, Type” phải bắt đầu bằng chữ in hoa.

Vd: Struct QPeople{

- Các hàm phương thức của class phải bắt đầu bằng kí tự viết thường

Vd: void mathDynamic(void) {

- Các hàm bình thường viết hoa chữ đầu

Vd: void Com\_addMethod(void) {

- Các hàm Global phải bắt đầu bằng tiền tố là kí hiệu của Module và viết hoa + dấu “\_”

Vd: Void QCom\_addMethod(void) {

- Biến thông thường thì bắt đầu bằng chữ viết thường

Vd: uint8\_t var = 10;

- Biến Global phải bắt đầu bằng tiền tố của module và viết hoa + “\_”

Vd: uint32\_t Q\_var = 10;

- Biến Local phải bắt đầu bằng tiền tố “l\_” và khai báo là static (MISRA-C rule 23)

Vd: static uint16\_t l\_var = 10;

- Biến trong class phải bắt đầu bằng tiền tố “m\_”

Vd: uint8\_t m\_var;

- MACRO thì phải viết hoa hết và dùng “\_” phân chia mỗi từ và với global phải có tiền tố module

Vd: Q\_MATH\_BASIC;

- Một số biến sử dụng trong hàm cụ thể

+ vòng lặp: i, j, k, m, n

+ pointer hoặc float: p, q, r, s, t, u, v, w