# SDD Review Document

## Summary

|  |  |
| --- | --- |
| **Date** | 04/Sep/21 |
| **Effort** | 1hr |
| **Room/Location** | NA |
| **Review Status** | Closed |
| **Review name** | SDD\_WindowApp.doc |
| **Method** | WT |
| **Release** | Initial |
| **Responsible** | Berenice Jimenez |
| **Project** | Door Control Module |
| **Reason of Review** | New Design document. |

## Comment List

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Reference** | **Comments / Actions** | **Classification (E)rror/Risk / (R)emark** | **Responsible person/Planned date for completion** | **Completion(Name/Date)** |
| 1 | Section 5.2 | As per base Architecture document, the Init function for WindowApp is a handler of Init Window module, and should be called WindowHandler\_Init. | Remark | Edgar Reyes |  |
| 2 | Section 5.2 | Current precondition can be ambiguous, can be specified that will happen immediately after power-up. | Remark | Edgar Reyes |  |
| 3 | Section 5.3 | As per base Architecture document, the Init function for WindowApp is a handler of Run Window module, and should be called WindowHandler\_Run. | Remark | Edgar Reyes |  |
| 4 | Section 5.4  Section 5.5  Section 5.6 | Add prefix WindowApp to function name. | Remark | Edgar Reyes |  |
| 5 | Section 5.4 | Current precondition can be ambiguous. This interface will be executed only when OPEN\_BTN from any window detects the event. | Remark | Edgar Reyes |  |
| 6 | Section 5.4 | Post Condition sentence is incomplete. | Remark | Edgar Reyes |  |
| 7 | Section 5.5  Section 5.6  Section 5.7 | Current precondition can be ambiguous. This interface will be executed only when CLOSE\_BTN from any window detects the event. | Remark | Edgar Reyes |  |
| 8 | Section 5.5  Section 5.6  Section 5.7 | Post Condition can be the change in the Window\_State after the function is successfully executed. | Remark | Edgar Reyes |  |

## Check List

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Description | OK / NOK / NR | Comment | Responsible person /  Planned date for completion | Status |
| 1 | Does the design comply to the SW architecture? (interfaces, scheduling...) | NOK | Different interface name for Init and Run | Edgar Reyes | Open |
| 2 | Are all requirements allocated to Desing elements? | OK |  |  | Closed |
| 3 | Are all operations described in an adequate detail and with the adequate notation? | OK |  |  | Closed |
| 4 | Is the coupling level between SW parts (internal or externals) reduced to the minimum?  Is the justification of all global data written in the design document? | OK |  |  | Closed |
| 5 | Is each data owned by one unit?  If a data is public (for read and/or for write operations), is its access made using a method provided by the owner?  (if a method is provided for read and write operations on the same pubilc data, the data has to be private) | OK |  |  | Closed |
| 6 | How are the variables initialized? If not initialized, is the reason explained? | OK |  |  | Closed |
| 7 | Is the mechanism to initialise the functionality (when needed) described?  (eg: function calls, data acquisition …) | OK |  |  | Closed |
| 8 | In case of global variable (shared or not shared) used in reentrance function (reentrance raised by an ISR), is there a mechanism to avoid data modification during its treatment? | OK |  |  | Closed |
| 9 | Are Tasks, ISRs and event notification function kept as short as possible? | OK |  |  | Closed |
| 10 | Is the state variable only used in one single module?  (If the state variable needs to be visible from another module (to be avoided), indicate it in the design and use the mechanism of read copy on that variable). | OK |  |  | Closed |
| 11 | Is the event memorization (ex: flag) consumed at the end of each reccurence of a state machine?  Otherwise, the risk is to use an obsolete event (ex: event memorization consumption conditionned by a state transition). | OK |  |  | Closed |
| 12 | In case of asynchronous reception of the same event by several objects (ex: state machine, C function called periodicly…), has each object its own memorization mechanism (ex: separate flags). | OK |  |  | Closed |