# Window lifter requirements:

Window lifter is the module responsible to control the window movement. **(The module shall control the window movement in two directions up and down --- Functional) check**

Window lifter is controlled by two switches that indicate the direction of the window movement. **(Two push buttons shall indicate the window’s movement – Non Functional) check**

## Window behavior:

For this purpose the window has to be emulated **(Window shall be emulated – Functional) check**

Using a 10 led bar. **(A ten LED Bar should be used to emulate behavior – Non Functional) check**

The color of this led bar has to be RED. **(The color used to indicate movement shall be red – Non Functional) check**

The movement of the window has to be simulated turning on/off the LEDS creating the animation of the window movement. **(The bar’s LEDs shall turn on and off to emulate movement -- Functional) check**

**LED turned on (red) indicates closed, LED turned off (blank) indicates open. Both states combined emulate approximate window height – Non Functional) check**

**(There should be well defined state changes for turning on/off LEDs of the LED bar-- Functional) check**

The time between each transition shall be 400 msec. **(State changes shall be determined by a fixed time constraint of 400 ms between each change of state (ambiguous) – Non Functional) check**

Window movement graphical description:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

CLOSED OPEN

There are two possible window movements:

-Up

-Down

Each window movement has to be indicated trough a led color. Depending on movement each led has to be turn on. **(Window movement should also be indicated with additional LEDs (one for each movement) – Functional) check**

**(LED’s color should change depending on the direction the window is moving -- Functional) check**

**(To indicate the window’s upward movement, LED indicator (or LED #1) should turn on blue. To indicate the window’s movement in downward direction, LED indicator (LED #2) should turn on green. If none of these conditions are met, LED indicator (or LEDs #1 & #2) should be turned off. – Non Functional) check**

|  |  |
| --- | --- |
| Movement | LED indicator color |
| UP | BLUE |
| Down | GREEN |

## Button Behavior:

In order to consider a validate button press; the button has to be pressed at least 10 msec. **(A well-defined anti-bouncing algorithm shall be implemented in order to avoid accidental activation of the window’s movement -- Functional) check**

The module has to be able to detect fail button press. In that case the button press or button combination has to be considered as invalid. **(All button combinations shall be approached and defined in order to avoid erratic behavior -- Functional)**  **check**

In case that a valid button press is detected the module has to follow the next behavior depending on the button pressed. **(Different behaviors shall be invoked when a valid button press is detected -- Functional) check**

**(Behaviors shall be determined taking into account well defined time constraints and input behavior from the user in order to validate continuous touch, one touch behaviors and when necessary movement interruption -- Functional). check**

|  |  |  |
| --- | --- | --- |
| Button Press | Time | Action |
| UP | >= 500 msec | **(The window shall go UP until it gets totally CLOSED while the button keeps being pressed -- Non Functional) check** |
| DOWN | >= 500 msec | **(The window shall go DOWN until it gets totally OPEN while the button keeps being pressed – Non Functional) check** |
| UP | <500 msec | **(The window shall go UP until it gets totally CLOSED automatically (Function one touch) – Non Functional) check** |
| DOWN | <500 msec | **(The window shall go DOWN until it gets totally OPEN automatically (Function one touch) – Non Functional) check** |

## Anti pinch functionality:

Anti pinch is a feature that prevents accidents between window and some human body parts like arms, hands, head…. **(A safety functionality algorithm shall be implemented within the module in order to prevent health risks to the user. Specifically, damage of the body’s limbs -- Functional) check**

**(A third signal should activate the "Anti Pinch" functionality - Functional) check**

In this case the signal that indicates the module the detection of a pinch will be a push button. **(A third button will act as a sensor to indicate the activation of the anti-pinch functionality – Non Functional) check**

Anti pinch button press has to follow the same characteristics than UP and DOWN buttons for valid press. **(A well-defined anti-bouncing algorithm shall also be implemented with this third button in order to avoid accidental de-activation of the window’s movement -- Functional) check**

This signal can just be considered as valid when the movement is UP. **(Anti-pinch functionality shall only activate when the window moves up with either continuous or one touch behaviors and the third button being pressed – Non Functional)** **check**

If this signal is valid then the module has to stop the upward movement and then move the window downwards, until the window gets totally OPEN. **(Anti pinch functionality should immediately stop the window’s movement and then move it downwards until it gets completely open – Functional)** **check**

After window is totally OPEN the module has to ignore during 5 seconds all button press. **(After the window gets completely open, all button presses should be ignored during 5 seconds as an additional safety feature – Functional) check**

After this time the module has to recognize every button press. **(The module should work normally after the 5 seconds have elapsed –Functional) check**