# Window lifter requirements:

Colors convention for functional and non-functional requirements.

Functional personal interpretation shall be highlighted with color yellow

Non - Functional personal interpretation shall be highlighted with color blue

Window lifter is the module responsible to control the window movement. (what: control window movement)

Window lifter is controlled by two switches that indicate the direction of the window movement. (functional, que tenga dos pines de entrada para indicar dirección del movimiento, es necesario) ( que sean switches o botones es no funcional)

## Window behavior:

For this purpose the window has to be emulated using a 10 led bar.

(Functional, window has to be emulated)

(non Functional que sean diez pines asignados de salida, es requisite del cliente, pero no afecta la funcionalidad, que es emular el comportamiento de la ventana)

The color of this led bar has to be RED. (non functional, but required)

The movement of the window has to be simulated turning on/off the LEDS creating the animation of the window movement. (what: simulation of the window movement turning on/off leds (indicators), how: color of leds, type of leds, etc)

The time between each transition shall be 400 msec. (what: transition of simulated movement, how: with a timing of 400 msec)

Window movement graphical description:

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

CLOSED OPEN

There are two possible window movements: (what: 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.

(what: movement indicated through a led color, how: on/off according to direction, color of corresponding direction)

|  |  |
| --- | --- |
| 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. (what: each button has to be pressed at least 10 msec to be valid, how: 10 msec or more, validation with poleo method or something else)

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.

(what: detect fail button press, how: it shall be managed internally, in the code)

In case that a valid button press is detected the module has to follow the next behavior depending on the button pressed (ACTION DETECTED).

(what: LA VENTANA SUBE Y BAJA, how: keep pressed or one touch)

|  |  |  |
| --- | --- | --- |
| Button Press | Time | Action |
| UP | >500 msec | The window shall UP until get totally CLOSED while the button keep press. |
| DOWN | >500 msec | The window shall DOWN until get totally OPEN while the button keep press. |
| UP | <500 msec | The window shall UP until get totally CLOSED automatically. (Function one touch) |
| DOWN | <500 msec | The window shall DOWN until get totally OPEN automatically. (Function one touch) |

## Anti pinch functionality:

Anti pinch is a feature than prevents accidents between window and some human body parts like arms, hands, head….

(what:evitar accidents entre ventana y partes humanas, how: anti-pinch function implemented on code and reflected on pin signals)

In this case the signal than indicates to the module the detection of a pinch will be a push button.

(what: signal needed to indicate the “pinch” event; how: with a push button or something more useful)

Anti pinch button press has to follow the same characteristics than UP and DOWN buttons for valid press.

(what: validation for “anti-pinch” signal, how: same validation for other buttons apply to “anti-pinch” button)

This signal just can be considered as valid when the movement is UP.

(what: validation for “anti-pinch” signal, how: when the movement is UP)

If this signal is valid then the module has to stop the UP Movement and then DOWN the window until the window get totally OPEN.

(what: action corresponding to a valid “anti pinch”=stop UP movement🡪activate DOWN movement UNTIL status is OPEN)

After window is totally OPEN the module has to ignore during 5 seconds all button press.

(WHAT: if window status is OPEN, button presses are ignored during 5 seconds, HOW: internal programming)

After this time the module has to recognize every button press.

(WHAT: after “anti-pinch” ending and window status=open, every button press can be recognized, HOW: internal coding)

**Basic pseudo for antipinch functionality.**

**IF (pinch & validPressForPinchButton & UpMovement ) then (StopUpMovement then (ActivateDownMovement until WindowStatus=OPEN))**

**If (WindowStatus=OPEN) {wait 5 seconds, ignoring all button signals}**

**Module shall recognize all button press.**

**OVERVIEW SOBRE CÓMO ANALIZAR LOS REQUERIMIENTOS.**

Revisión de requerimientos: what y how (funcional y no funcional)

Realizar diagrama de flujo (o máquina de estados simple) considerando sólo las partes funcionales (sólo las acciones o el what, el cómo se realizan se define una vez clarificado el proceso y su estructura, es decir, se profundiza DENTRO DE cada bloque)

Realizar nueva versión del diagrama, incluyendo el cómo

….. luego no se que hacer