|  |  |
| --- | --- |
| **Title:** | **Window Lifter Controller** |

**1.- PURPOSE………………………………………………………………………….………………………..4**

**2.-DEFINITIONS AND ABREVIATIONS…………………………………………….……………………….4**

**3.-REALIZATION CONSTRAINTS AND TARGETS………………………………………………………..5**

**4.-SW CONCEPTUAL DESIGN……………………………………………………………………………….6**

**5.- SW Component internal breakdown……………………………………………………………………7**

**5.1.- Diagrams by component…………………………..…………………………………………..7**

**5.1.1.-Complet Diagram…………………………………………………………………………..….8**

## 5.2.- Functional Decomposition…………………………………………………………………………….10

**5.3.- Function Description and Dynamic Behavior……………………………………………………..12**

## 5.3.1- Function voidappUpDown\_void\_manualUp()……………………………….……………………12

## 5.3.2.- Function void appUpDown\_void\_manualDown()…………………..……..……………………12

## 5.3.3.- Function T\_U32 appUpDown\_u32\_PushUpButton()…………………...………………………13

## 5.3.4.-FunctionT\_U32appUpDown\_u32\_PushDownButton()……….…………………………………14

## 5.3.5.- Function T\_U32 appUpDown\_u32\_PushAntipinch()……………………………………………14

## 5.3.6.- Function void appUpDown\_void\_default\_Leds(T\_U32 led)………………..………….………15

## 5.3.7.- Function void appUpDown\_void\_clearGPIO()………………………………...…………….……16

## 5.3.8.- Function void appUpDown\_void\_timer1()……………………………………..……………..…..16

**5.3.9.- Function void appUpDown\_void\_timer0()………………………………………….…..…………17**

## 5.3.10.- Function T\_U32 appUpDown\_u32\_validation10ms()…………………………………..……….18

## 5.3.11.- Function T\_U32 appUpDown\_u32\_validation500ms()…………………………………………18

## 5.3.12.- Function T\_U32 halWinMov\_u32\_ValidateUpButton()………………………….………..……19

## 5.3.13.- Function T\_U32 halWinMov\_u32\_ValidateUpButton()………………..………..………………20

## 5.3.14.- Function T\_U32 halWinMov\_u32\_ValidateAntipinch()*……………*……………..……20

## 5.3.15.- Function void halWinMov\_void\_ToggleBlueLed(T\_U32 u32\_status)…………..……21

## 5.3.16.- Function void halWinMov\_void\_ToggleRedLed(T\_U32 u32\_status)…………………22

## 5.3.17.- Function void halWinMov\_void\_ToggleGreenLed(T\_U32 u32\_status)…………………22

## 5.3.18.- Function void halWinMov\_void\_ToggleLedBar#(T\_U32 u32\_status)……………...……23

## 5.3.19 .- Function void main\_void\_onetouchUp()*……………..………………*…………...……24

## 5.3.20.- Function void main\_void\_onetouchDown()…………………………………………...……25

## 5.3.21.- Function void main\_void\_behaviorUp()….…………………………………………………..26

## 5.3.22.- Function void main\_void\_behaviorDown()….…………………………………………………..26

## 5.3.23.- Function void main\_void\_Antipinch()….…………………………………………………..27

## 5.3.24.- Function void main\_void\_idleState()….…………………………………………………..27

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **History** | | | | |
| **Issue status**  (Index) | **Maturity/Date**  (draft/invalid/valid)  (dd-mmm-yyyy) | **Author**  Department | **Check/Release**  Department | **Description** |
| 1.0 | Draft  04-0ct-09 | Miguel Garcia | Miguel Garcia | Creation of the document. |
| 2.0 | Draft  26-Oct-17 | Rodrigo Mortera | Rodrigo Mortera | Insert Diagrams. |
| 2.1 | Draft  27-Oct-17 | Rafael Sanchez | Rafael Sanchez | Add function descriptions.­­­­­­ |
| 2.2 | Draft  28-Oct-17 | Rodrigo Mortera | Rodrigo Mortera | Updates Diagrams |
| 2.3 | Draft  29-Oct-17 | Rafael Sanchez | Rafael Sanchez | Updates function descriptions and Add flow chart to each function. |
| 3 | Draft 1-Nov.17 | Rodrigo Mortera | Rodrigo Mortera | Update Diagrams |
| 3-1 | Draft 2-Nov-17 | Rafael Sanchez | Rafael Sanchez | Update Diagrams |

**Table of Contents**

# Purpose

This document explains how the window lifter works and also describe each function through the layer programming method.

# Definitions and abbreviations

**Definitions**

|  |  |
| --- | --- |
| *BlueLed* | *Define to identify the port for the Blue Led* |
| *RedLed* | *Define to identify the port for the Red Led* |
| *GreenLed* | *Define to identify the port for the Green Led* |
| *UpButton* | *Define to identify the port for the push button 12* |
| *DownButton* | *Define to identify the port for the push button 13* |
| *LedBar\_1* | *Define to identify the port for the Bar’s first Led* |
| *LedBar\_2* | *Define to identify the port for the Bar’s secondary Led* |
| *LedBar\_3* | *Define to identify the port for the Bar’s third Led* |
| *LedBar\_4* | *Define to identify the port for the Bar’s fourth Led* |
| *LedBar\_5* | *Define to identify the port for the Bar’s fifth Led* |
| *LedBar\_6* | *Define to identify the port for the Bar’s sixth Led* |
| *LedBar\_7* | *Define to identify the port for the Bar’s seventh Led* |
| *LedBar\_8* | *Define to identify the port for the Bar’s eighth Led* |
| *LedBar\_9* | *Define to identify the port for the Bar’s ninth Led* |
| *LedBar\_10* | *Define to identify the port for the Bar’s tenth Led* |
| *malGPIO\_init\_void\_WDOG\_disable* | *Disable watchdog* |
| *malGPIO\_init\_void\_PORT\_init()* | *Port init* |
| *SOSC\_init\_8MHz* | *Clock of 8Mhz* |
| *SPLL\_init\_160MHz* | *Clock of 160Mhz* |
| *NormalRUNmode\_80MHz* | *Set the mode of the timer to normal with 8Mhz* |
| *malGPIO\_init\_void\_LPIT0\_init* | *Initialize timers* |
| *u32\_lpit0\_ch0\_flag\_counter* | *Variable to control timer 0* |
| *u32\_lpit0\_ch1\_flag\_counter* | *Variable to control timer 1* |
| *appUpDown\_void\_clearGPIO()* | *Function to turnoff/turnon BlueLed, GreenLed, RedLed and Ledbars* |
| *appUpDown\_u32\_PushDownButton()* | *Read if Down Button is pushed* |
| *appUpDown\_u32\_PushUpButton()* | *Read if Up Button is pushed* |
| *appUpDown\_u32\_validation10ms()* | *Validate if the button is pushed at least during 10 ms* |
| *appUpDown\_u32\_validation500ms()* | *Validate if the button is pushed at least during 500 ms* |
| *appUpDown\_void\_timer1()* | *Increase the variable u32\_lpit0\_ch1\_flag\_counter each 1 ms* |
| *appUpDown\_void\_default\_Leds(T\_U32 led)* | *Turn on/off Leds in a default phase* |
| *main\_void\_oneTouchUp()* | *Ascending turn on the LedBar leds with just one touch* |
| *main\_void\_oneTouchDown()* | *Descending turn on the LedBar leds with just one touch* |
| *main\_void\_behaviorUp()* | *Select the turn on leds behavior.* |
| *main\_void\_behaviorDown()* | *Select the turn off leds behavior.* |
| *main\_void\_antiPinch()* | *Do the antipinch action* |
| *main\_void\_idleState()* | *Put the program in an idle state* |

**Abbreviations**

|  |  |
| --- | --- |
| CH1 | Channel1 |
| CH2 | Channel2 |
| R&D | Research and Develop |

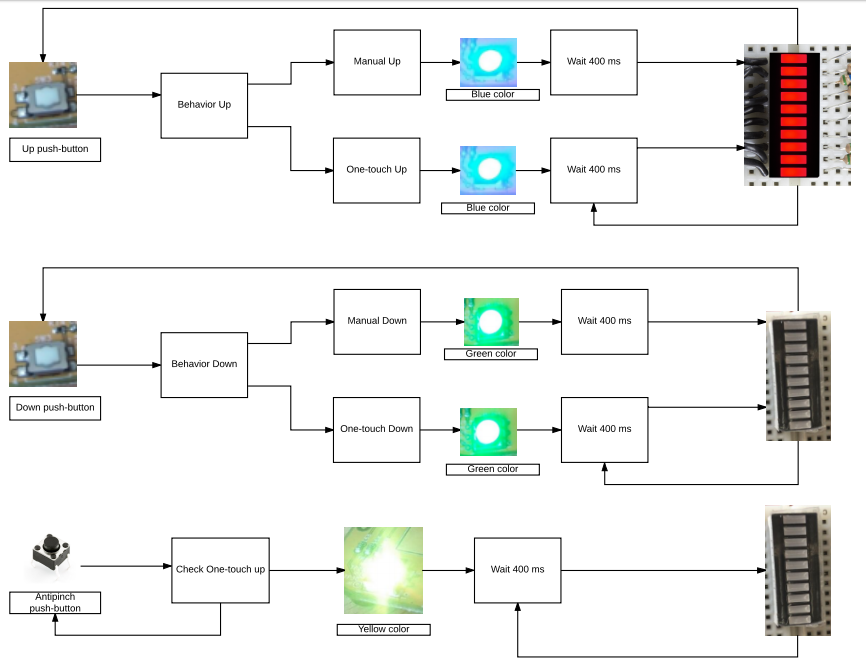
**References**

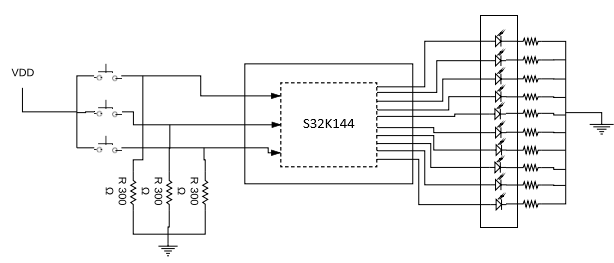
|  |  |  |
| --- | --- | --- |
| **N°** | **Document name** | **Reference** |
| *1* | *USER\_GUIDE\_S32K144* | [Link](https://www.nxp.com/products/microcontrollers-and-processors/arm-based-processors-and-mcus/s32-automotive-processors-and-microcontrollers/32-bit-automotive-general-purpose-microcontrollers:S32K?tab=Documentation_Tab) |
| 2 | [***S32K14x Series Reference Manual***](https://www.nxp.com/docs/en/reference-manual/S32K-RM.pdf) |
| 3 | [***S32K1xx Data Sheet***](https://www.nxp.com/docs/en/data-sheet/S32K-DS.pdf) |
| 4 | ***AN5413, s32k14x Series Cookbook*** |

# Realization constraints and targets

The objective of the task is to achieve to lift and down a window car simulated by 10 red LedBar regarding to the customer request. The development of the project must be on the Development Kit Platform S32K144EVB NXP microcontroller provided by Continental R&D.

# SW Conceptual design





Window lifter block diagram

1. **SW Component internal breakdown**
   1. **Diagrams by component**

Blue Led.

Lines and Functions related with Down behavior and green led.

Lines and Functions related with Up behavior.

Lines and Functions related with Antipinch behavior.

Red Led.

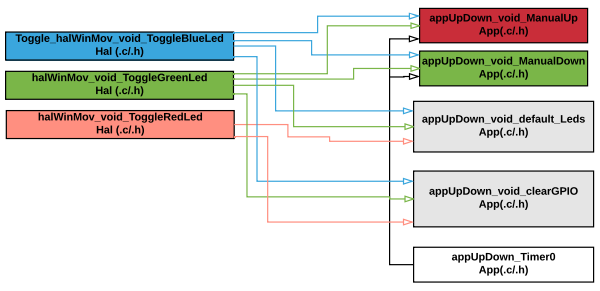
Timer functions and void function.

Status output port led’s function.

* **Validate\_Antipinch function validates the press in Push\_Antipinch function in App.**
* **Validate\_UpButton function validates the press in Push\_UpButton function in App.**
* **Validate\_DownButton function works to validate the press in Push\_DownButton function in App.**

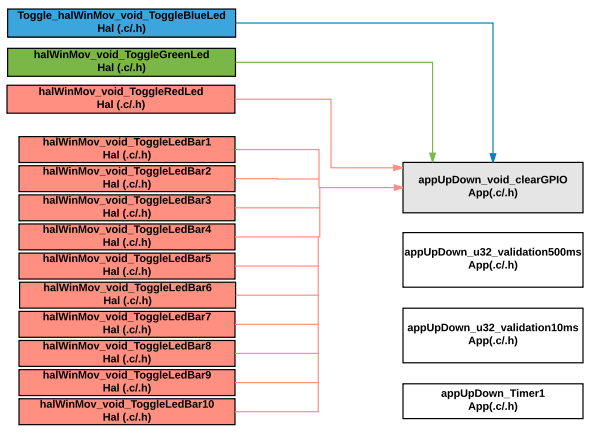
## s

* **Toggle functions execute the functionality to turn on or turn off the led’s, called by either Manual\_Up or Manual\_Down functions in App.**

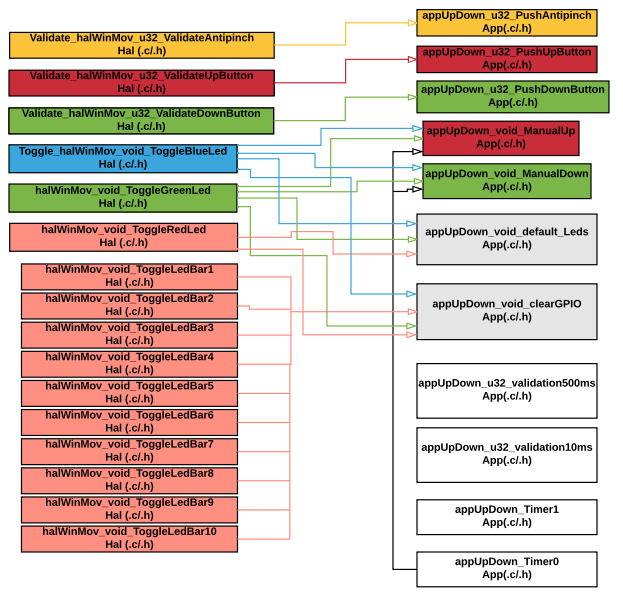


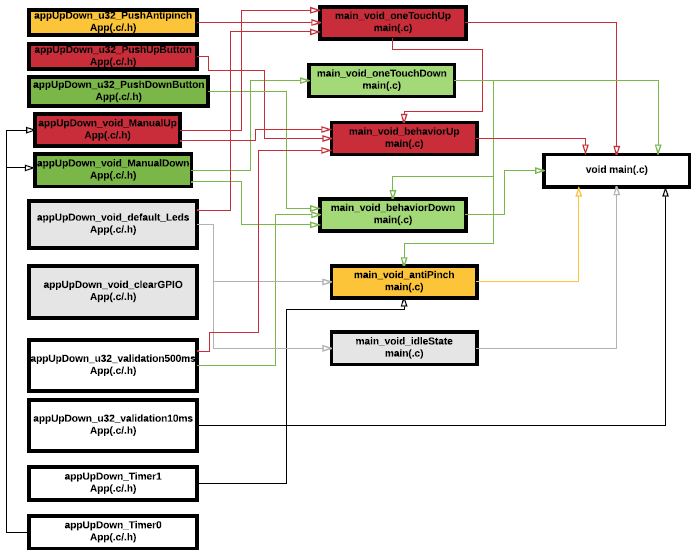
## Complete Diagram

* **The default\_Leds function Works to condition if the leds should be turned on or turned off.**



* **The Clear\_GPIO only Works to clear all the outputs to initialize as turned off the led bar and the RGB led.**

***5.1.1 Complete Diagram***



appUpDown\_Timer0 enables the timer to count 400ms.

appUpDown\_Timer1 enables the timer to count 1ms.

## Functional Decomposition

Black Line makes reference to appUpDown\_void\_clearGPIO() function.

Blue Line makes reference to appUpDown\_void\_default\_Leds() function.

## 

## 

Black Line makes reference to appUpDown\_void\_ManualUp() function.

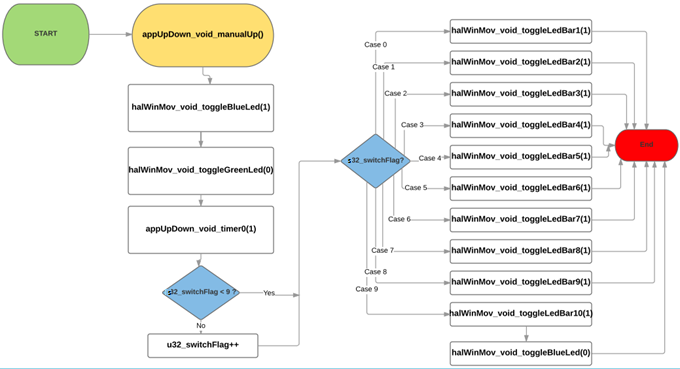
Blue Line makesreference to appUpDown\_void\_ManualDown() function.

* 1. **Function Description and Dynamic Behavior**

## Function void appUpDown\_void\_manualUp()

|  |  |
| --- | --- |
| **Description** | Ascending Turn on the Led Bar using a switch statement, the Led number is proportional with s32\_switchFlag value. |
| **Return Value** | void |
| **Precondition** | The s32\_switchFlag has to be set with a value [-1 to 9] |
| **Post condition** | The LedBar is turned on. |
| **Error Conditions** |  |

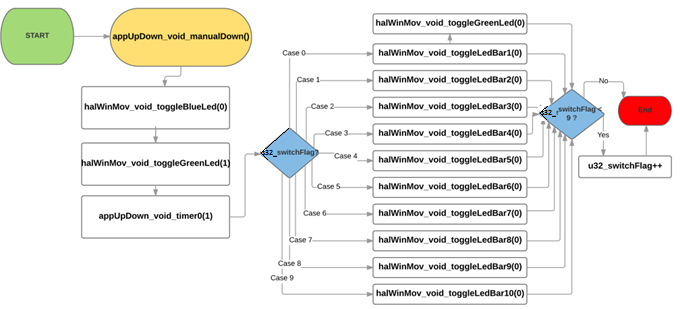
**Dynamic Behavior**

**

## Function void appUpDown\_void\_manualDown()

|  |  |
| --- | --- |
| **Description** | Descending Turn on the Led Bar using a switch statement, the Led number is proportional with s32\_switchFlag value. |
| **Return Value** | void |
| **Precondition** | The s32\_switchFlag has to be set with a value [-1 to 9] |
| **Post condition** | The ledbar is turned off. |
| **Error Conditions** |  |

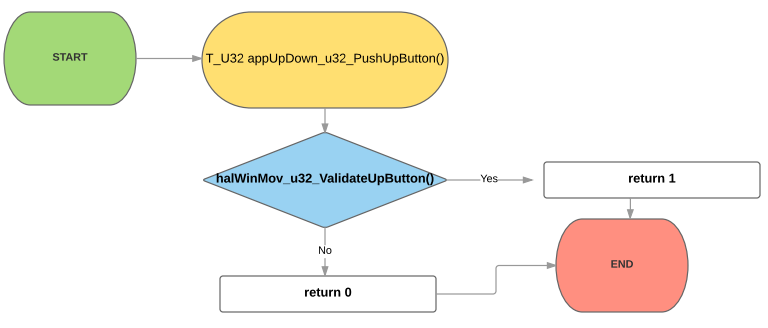
**Dynamic Behavior**



## Function T\_U32 appUpDown\_u32\_PushUpButton()

|  |  |
| --- | --- |
| **Description** | Calls halWinMov\_u32\_ValidateUpButton function, to know if the halWinMov\_u32\_ValidateUpButton()is true or false |
| **Parameter 1** <input| output| inout> |  |
| **Parameter 2..n** |  |
| **Return Value** | T\_U32 1 , 0 |
| **Precondition** |  |
| **Post condition** |  |
| **Error Conditions** |  |

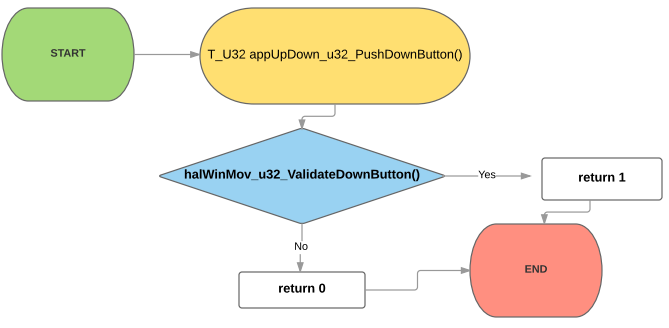
**Dynamic Behavior**



## Function T\_U32 appUpDown\_u32\_PushDownButton()

|  |  |
| --- | --- |
| **Description** | Calls Calls halWinMov\_u32\_ValidateDownButton() function, to know if the Calls halWinMov\_u32\_ValidateDownButton()is true or false |
| **Parameter 1** <input| output| inout> |  |
| **Parameter 2..n** |  |
| **Return Value** | T\_U32 1 , 0 |
| **Precondition** |  |
| **Post condition** |  |
| **Error Conditions** |  |

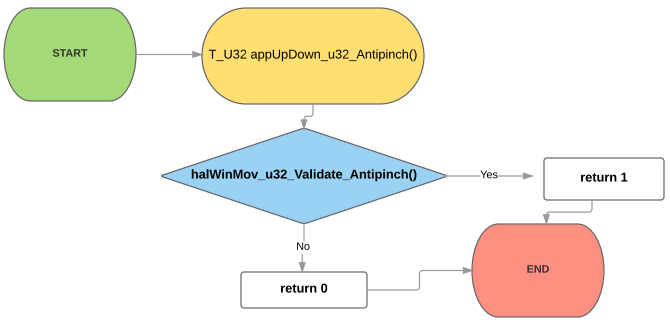
**Dynamic Behavior**



## Function T\_U32 appUpDown\_u32\_PushAntipinch()

|  |  |
| --- | --- |
| **Description** | Calls halWinMov\_u32\_ValidateAntipinch() function, to know if the halWinMov\_u32\_ValidateUpButton()is true or false |
| **Parameter 1** <input| output| inout> |  |
| **Parameter 2..n** |  |
| **Return Value** | T\_U32 1 , 0 |
| **Precondition** |  |
| **Post condition** |  |
| **Error Conditions** |  |

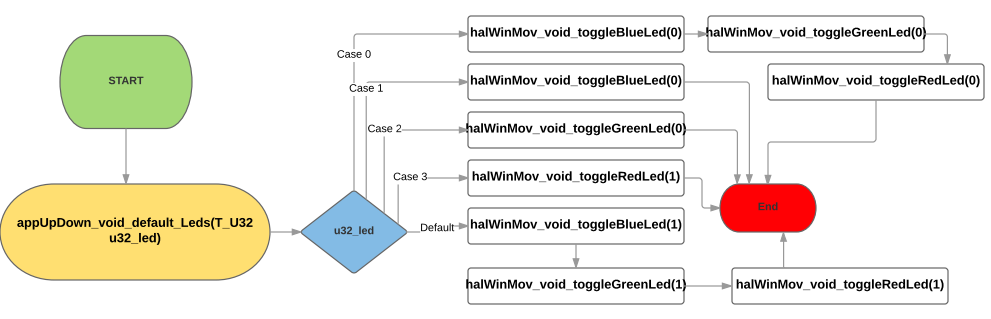
**Dynamic Behavior**



## Function void appUpDown\_void\_default\_Leds(T\_U32 led)

|  |  |
| --- | --- |
| **Description** | Using T\_U32 led, with a switch statement, select case 0,1,2,3 and default.  case 0 → Turn off Blue, Green and Red Led.  case 1 → Turn off Blue.  case 2 → Turn off Green.  case 3 → Turn on Red.  default → Turn on Blue, Green and Red Led. |
| **Parameter 1** <input| output| inout> | input T\_U32 |
| **Parameter 2..n** |  |
| **Return Value** | void |
| **Precondition** |  |
| **Post condition** |  |
| **Error Conditions** |  |

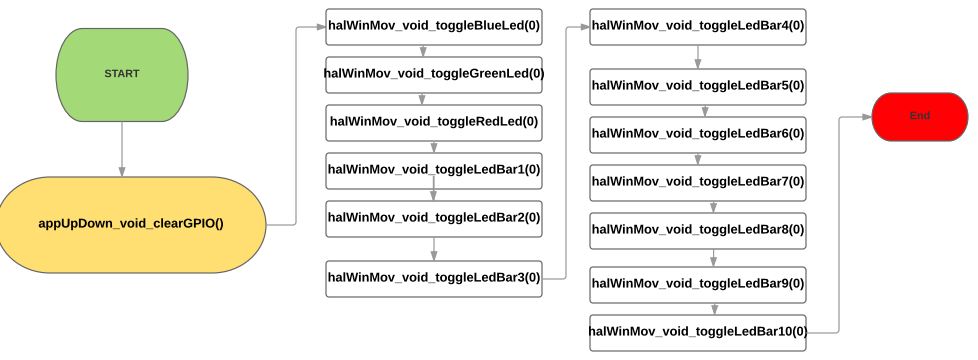
**Dynamic Behavior**



## Function void appUpDown\_void\_clearGPIO()

|  |  |
| --- | --- |
| **Description** | Turn off Blue Led, Green Led, Red Led and LedBar# |
| **Parameter 1** <input| output| inout> |  |
| **Parameter 2..n** |  |
| **Return Value** | void |
| **Precondition** |  |
| **Post condition** |  |
| **Error Conditions** |  |

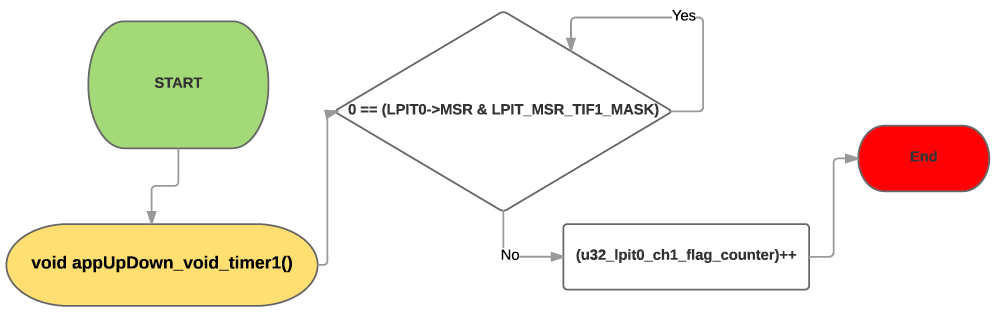
**Dynamic Behavior**



## Function void appUpDown\_void\_timer1()

|  |  |
| --- | --- |
| **Description** | After 40000 cycles has occurred increment u32\_lpit0\_ch1\_flag\_counter variable. (1 ms) |
| **Parameter 1** <input| output| inout> |  |
| **Parameter 2..n** |  |
| **Return Value** | void |
| **Precondition** | The LPIT\_MSR\_TIF1\_MASK timer should be configured and initialized |
| **Post condition** | The LPIT\_MSR\_TIF1\_MASK timer has been cleared |
| **Error Conditions** |  |

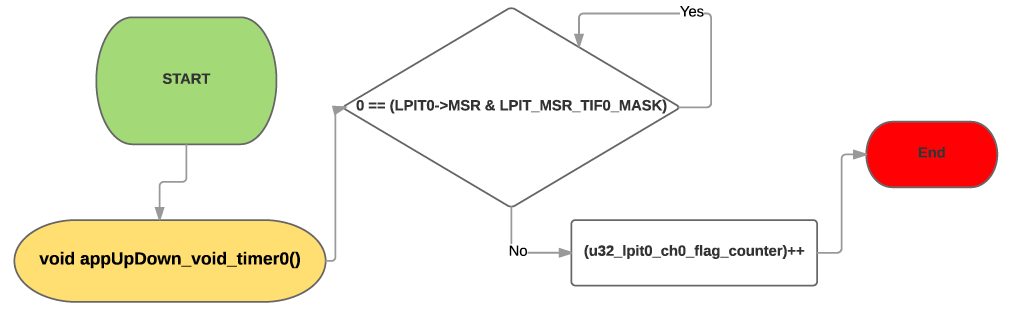
**Dynamic Behavior**



## Function void appUpDown\_void\_timer0()

|  |  |
| --- | --- |
| **Description** | After 16 000 000 cycles has occurred increment u32\_lpit0\_ch0\_flag\_counter variable. (400 ms) |
| **Parameter 1** <input| output| inout> |  |
| **Parameter 2..n** |  |
| **Return Value** | void |
| **Precondition** | The LPIT\_MSR\_TIF0\_MASK timer should be configured and initialized |
| **Post condition** | The LPIT\_MSR\_TIF0\_MASK timer has been cleared |
| **Error Conditions** |  |

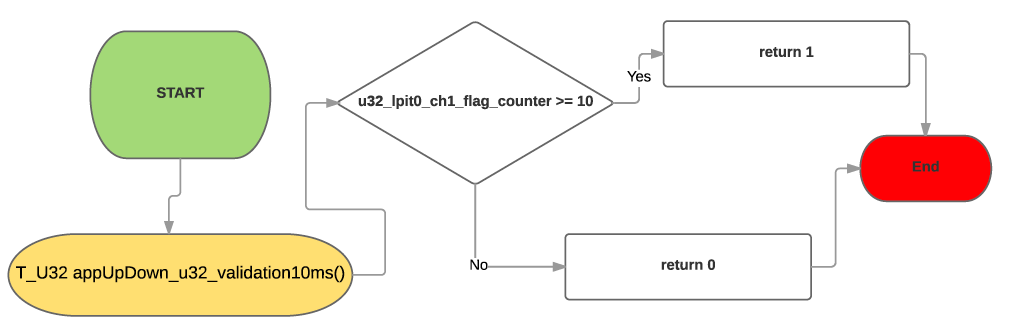
**Dynamic Behavior**



## Function T\_U32 appUpDown\_u32\_validation10ms()

|  |  |
| --- | --- |
| **Description** | Check when u32\_lpit0\_ch1\_flag\_counter is equal or higher to 10. This mean that 10 ms are occurred. |
| **Parameter 1** <input| output| inout> |  |
| **Parameter 2..n** |  |
| **Return Value** | T\_U32 1,0 |
| **Precondition** | The LPIT\_MSR\_TIF1\_MASK timer should be configured and initialized, u32\_lpit0\_ch0\_flag should be 0 at the beginning |
| **Post condition** |  |
| **Error Conditions** |  |

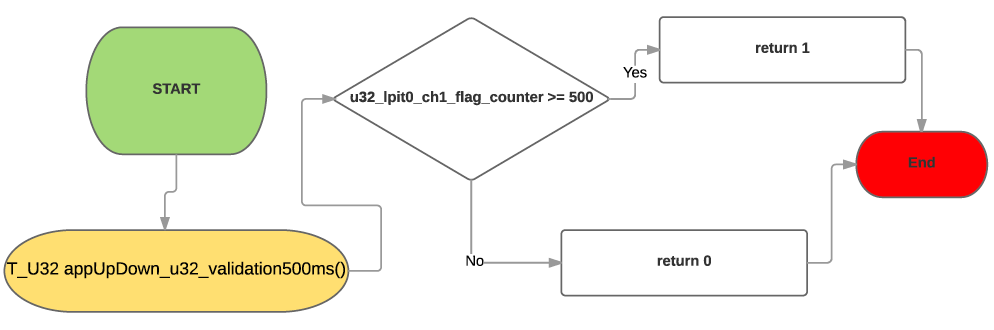
**Dynamic Behavior**



## Function T\_U32 appUpDown\_u32\_validation500ms()

|  |  |
| --- | --- |
| **Description** | Check when u32\_lpit0\_ch1\_flag\_counter is equal or higher to 500. |
| **Parameter 1** <input| output| inout> |  |
| **Parameter 2..n** |  |
| **Return Value** | T\_U32 1,0 |
| **Precondition** | The LPIT\_MSR\_TIF1\_MASK timer should be configured and initialized, u32\_lpit0\_ch1\_flag should be 0 at the beginning |
| **Post condition** |  |
| **Error Conditions** |  |

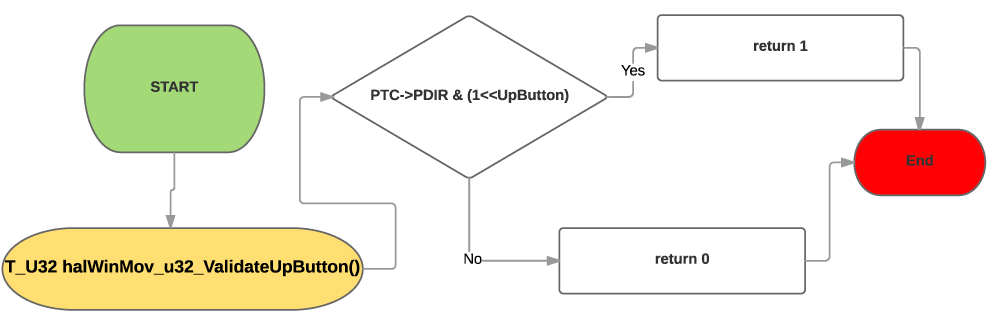
**Dynamic Behavior**



## Function T\_U32 halWinMov\_u32\_ValidateUpButton()

|  |  |
| --- | --- |
| **Description** | Check if PORT C12 value is in High or Low and returns 1 if is High and 0 if is Low |
| **Parameter 1** <input| output| inout> |  |
| **Parameter 2..n** |  |
| **Return Value** | T\_U32 1,0 |
| **Precondition** | The PORTD C12 should be configured as input |
| **Post condition** |  |
| **Error Conditions** |  |

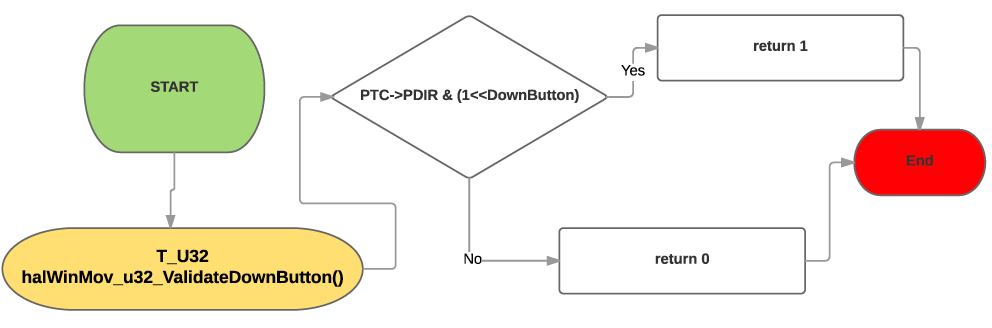
**Dynamic Behavior**



## Function T\_U32 halWinMov\_u32\_ValidateUpButton()

|  |  |
| --- | --- |
| **Description** | Check if PORT C13 value is in High or Low and returns 1 if is High and 0 if is Low |
| **Parameter 1** <input| output| inout> |  |
| **Parameter 2..n** |  |
| **Return Value** | T\_U32 1,0 |
| **Precondition** | The PORTD C13 should be configured as input. |
| **Post condition** |  |
| **Error Conditions** |  |

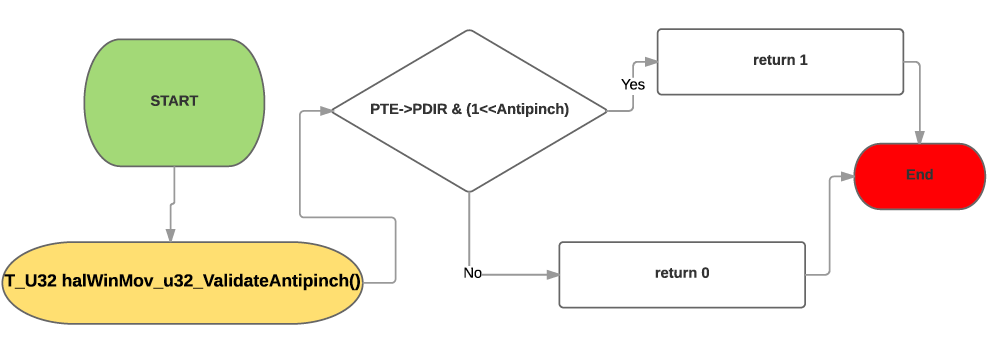
**Dynamic Behavior**



## Function T\_U32 halWinMov\_u32\_ValidateAntipinch()

|  |  |
| --- | --- |
| **Description** | Check if PORT E13 value is in High or Low and returns 1 if is High and 0 if is Low |
| **Parameter 1** <input| output| inout> |  |
| **Parameter 2..n** |  |
| **Return Value** | T\_U32 1,0 |
| **Precondition** | The PORTD E13 should be configured as input. |
| **Post condition** |  |
| **Error Conditions** |  |

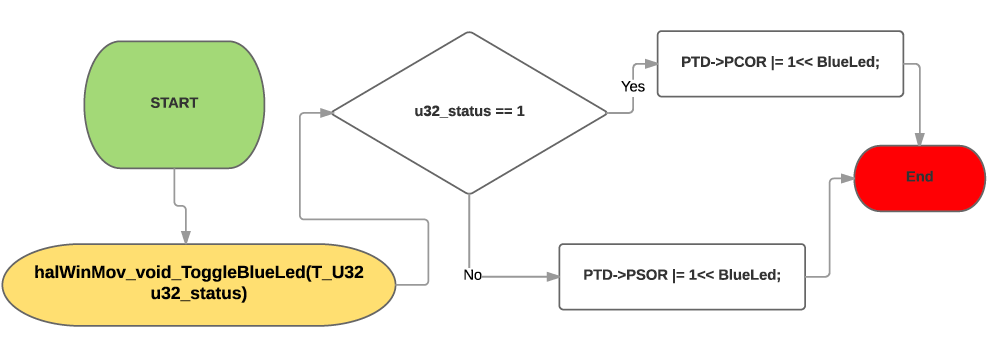
**Dynamic Behavior**



## Function void halWinMov\_void\_ToggleBlueLed(T\_U32 u32\_status)

|  |  |
| --- | --- |
| **Description** | Write in PORT D0:  - High if int status = 1.  - Low if int status = 0. |
| **Parameter 1** <input| output| inout> | Input T\_U32 u32\_status |
| **Parameter 2..n** |  |
| **Return Value** | void |
| **Precondition** | The PORTD D0 should be configured as output. |
| **Post condition** |  |
| **Error Conditions** |  |

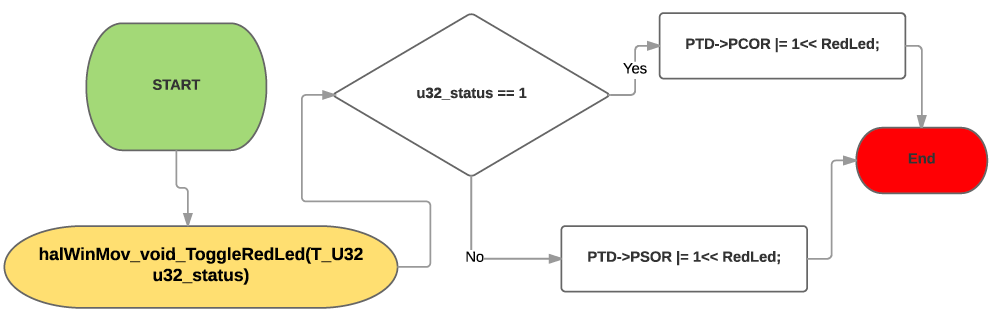
**Dynamic Behavior**



## Function void halWinMov\_void\_ToggleRedLed(T\_U32 u32\_status)

|  |  |
| --- | --- |
| **Description** | Write in PORT D15:  - High if int status = 1.  - Low if int status = 0. |
| **Parameter 1** <input| output| inout> | Input T\_U32 u32\_status |
| **Parameter 2..n** |  |
| **Return Value** | void |
| **Precondition** | The PORTD D15 should be configured as output. |
| **Post condition** |  |
| **Error Conditions** |  |

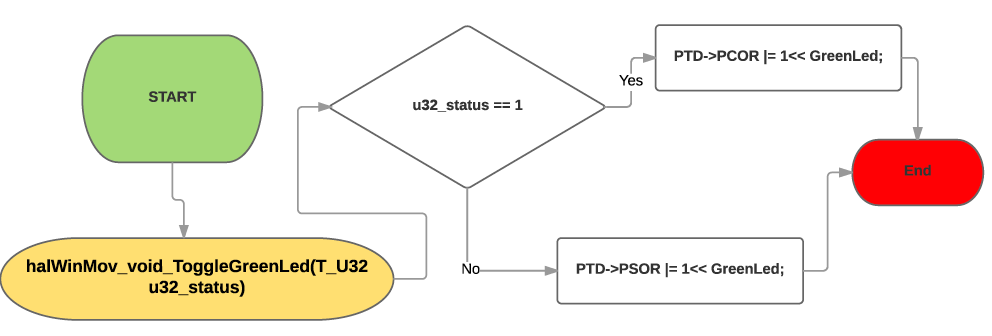
**Dynamic Behavior**



## Function void halWinMov\_void\_ToggleGreenLed(T\_U32 u32\_status)

|  |  |
| --- | --- |
| **Description** | Write in PORT D16:  - High if int status = 1.  - Low if int status = 0. |
| **Parameter 1** <input| output| inout> | Input T\_U32 u32\_status |
| **Parameter 2..n** |  |
| **Return Value** | void |
| **Precondition** | The PORTD D16 should be configured as output. |
| **Post condition** |  |
| **Error Conditions** |  |

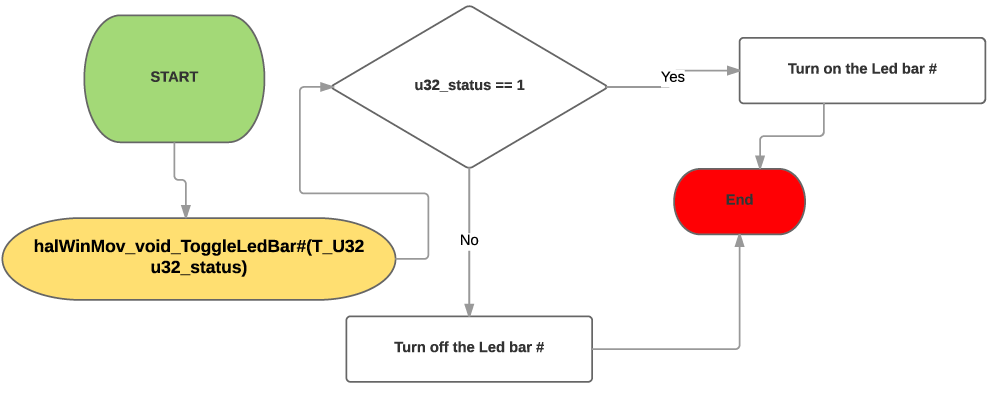
**Dynamic Behavior**



## Function void halWinMov\_void\_ToggleLedBar#(T\_U32 u32\_status)

|  |  |  |
| --- | --- | --- |
| **Description** | Write High or Low in Led number # | |
| LedBar\_1  Write in PORT C7:  - High if int status = 1.  - Low if int status = 0.  LedBar\_2  Write in PORT B17:  - High if int status = 1.  - Low if int status = 0.  LedBar\_3  Write in PORT B14:  - High if int status = 1.  - Low if int status = 0.  LedBar\_4  Write in PORT B15:  - High if int status = 1.  - Low if int status = 0.  LedBar\_5  Write in PORT B16:  - High if int status = 1.  - Low if int status = 0. | LedBar\_6  Write in PORT C14:  - High if int status = 1.  - Low if int status = 0.  LedBar\_7  Write in PORT C3:  - High if int status = 1.  - Low if int status = 0.  LedBar\_8  Write in PORT E16:  - High if int status = 1.  - Low if int status = 0.  LedBar\_9  Write in PORT E15:  - High if int status = 1.  - Low if int status = 0.  LedBar\_10  Write in PORT E14:  - High if int status = 1.  - Low if int status = 0. |
| **Parameter 1** <input| output| inout> | Input T\_U32 u32\_status | |
| **Parameter 2..n** |  | |
| **Return Value** | void | |
| **Precondition** | Should be configured as output: | |
| PORT C7  PORT B17  PORT B14  PORT B15  PORT B16 | PORT C14  PORT C3  PORT E16  PORT E15  PORT E14 |
| **Post condition** |  | |
| **Error Conditions** |  | |

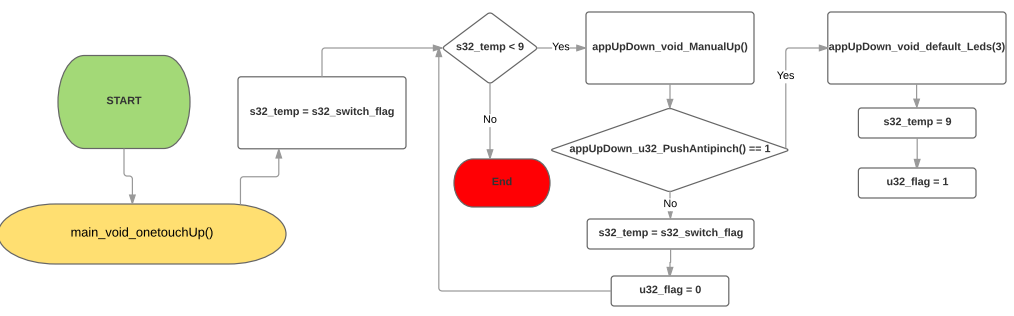
**Dynamic Behavior**



## Function void main\_void\_onetouchUp()

|  |  |
| --- | --- |
| **Description** | Calls appUpDown\_void\_ManualUp() inside a while to similuted the lift movement. And turn on all the led bar |
| **Parameter 1** <input| output| inout> |  |
| **Parameter 2..n** |  |
| **Return Value** | void |
| **Precondition** |  |
| **Post condition** |  |
| **Error Conditions** |  |

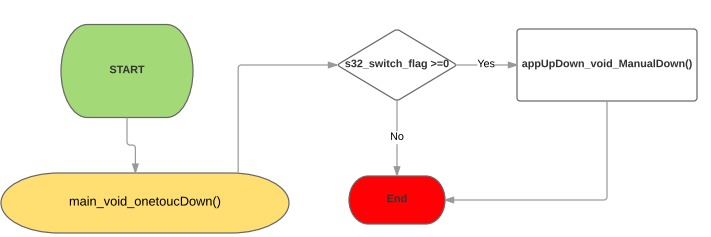
**Dynamic Behavior**



## Function void main\_void\_onetouchDown()

|  |  |
| --- | --- |
| **Description** | Calls appUpDown\_void\_ManualUp() inside a while to similuted the lift movement. And turn on all the led bar |
| **Parameter 1** <input| output| inout> |  |
| **Parameter 2..n** |  |
| **Return Value** | void |
| **Precondition** |  |
| **Post condition** |  |
| **Error Conditions** |  |

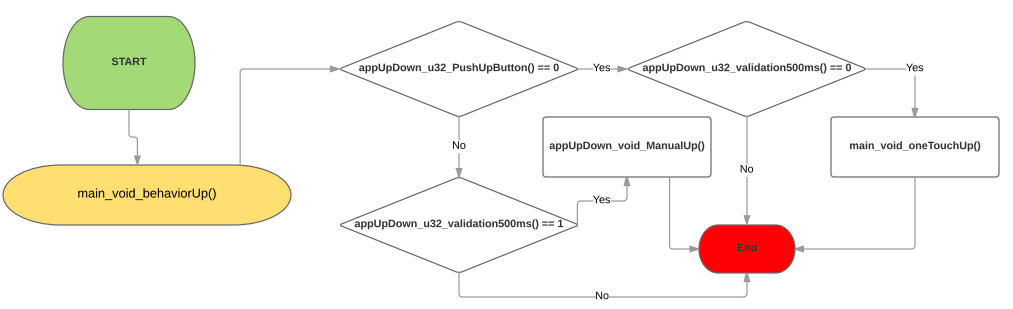
**Dynamic Behavior**



## Function void main\_void\_behaviorUp()

|  |  |
| --- | --- |
| **Description** | Select the behavior between Manual Up and OneTouchUp |
| **Parameter 1** <input| output| inout> |  |
| **Parameter 2..n** |  |
| **Return Value** | void |
| **Precondition** |  |
| **Post condition** |  |
| **Error Conditions** |  |

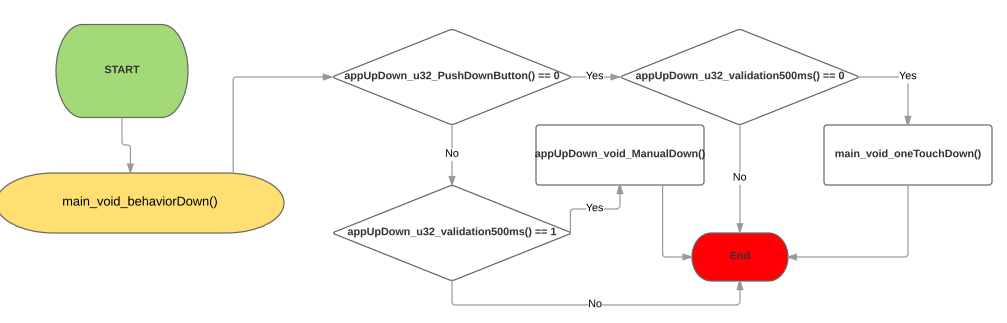
**Dynamic Behavior**



## Function void main\_void\_behaviorDown()

|  |  |
| --- | --- |
| **Description** | Select the behavior between Manual Down and OneTouchDown |
| **Parameter 1** <input| output| inout> |  |
| **Parameter 2..n** |  |
| **Return Value** | void |
| **Precondition** |  |
| **Post condition** |  |
| **Error Conditions** |  |

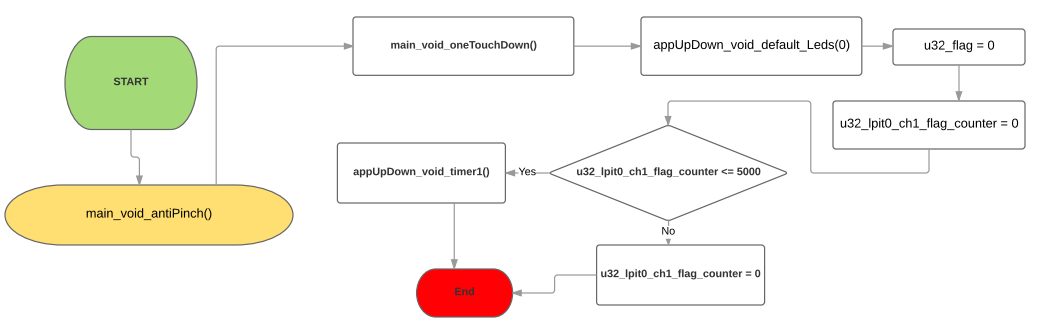
**Dynamic Behavior**



## Function void main\_void\_Antipinch()

|  |  |
| --- | --- |
| **Description** | If the flag antipinch is 1, the action antipinch is going to be activated. During 5 seconds it will not respond to any action. |
| **Parameter 1** <input| output| inout> |  |
| **Parameter 2..n** |  |
| **Return Value** | void |
| **Precondition** |  |
| **Post condition** |  |
| **Error Conditions** |  |

**Dynamic Behavior**



## Function void main\_void\_idleState()

|  |  |
| --- | --- |
| **Description** | Put the all the leds in Low. |
| **Parameter 1** <input| output| inout> |  |
| **Parameter 2..n** |  |
| **Return Value** | void |
| **Precondition** |  |
| **Post condition** |  |
| **Error Conditions** |  |

**Dynamic Behavior**

