Title: Door Control Module GOC Motors Automotive
SW Component - Window

History				
Issue status (Index)	**Maturity/Date** (draft/invalid/valid) (dd-mmm-yyyy)	**Author** Department	**Check/Release** Department	**Description**
1.0	Draft 04-0ct-21	Ariel Gonzalez	Ariel Gonzalez	Creation of the document

Table of Contents

1 Purpose

This document has been created to show the detailed design of the software component Window stated on the architecture and describe the units, functions, interfaces and information flow

2 Definitions and abbreviations

Definitions

Special Byte	*Special byte received to change the baud rate 55h*
Antipinch	*Special system that prevents window accidents*

Abbreviations

ECU: Electronic control unit
SWC: Software components
ADC: Analog to digital converter

References

N°	Document name	Reference
1	*Proyecto_DoorControlModule*	
2.	*Traceability matrix template*	
3.	*DesignReviewChecklist_Window*	

3 Realization constraints and targets

This Software component shall be able to provide at least the following functionalities:

- *Attend request from application to operate window control operations.*
- *Determine and provide window status to Application.*
- *Implements Antipinch functionality.*

All of them at an ECU level of abstraction and accessible for other SWC via public interface. The software component is also responsible for the setup for the window actuations. (The lead bar simulation) and the direct response for remote and antipinch request

4 SW Conceptual design

The main function of this software component is to cover the window functionalities at an ECU abstraction level, to make sure key operations are available in the window app and to respond directly to certain types of signals and requests. This means making independent processes based on the information available and publishing some of the new data and operation for public use, and for this reason, this SWC will take a highly encapsulated approach.

The system boundary in this case includes the operations related to providing the window app basic functions and setups, as well as state/Operation information, the response actuation to remote request and the execution of the antipinch system. This system also will use information provided by the HwConfig, ADC and Dio interfaces making calls to important get functions.

As global data, this SWC will only use the following: Window_Status, Window_Requests, and variables representing initialization specific values and important actuation conditions. All to have a common reference point for important data.

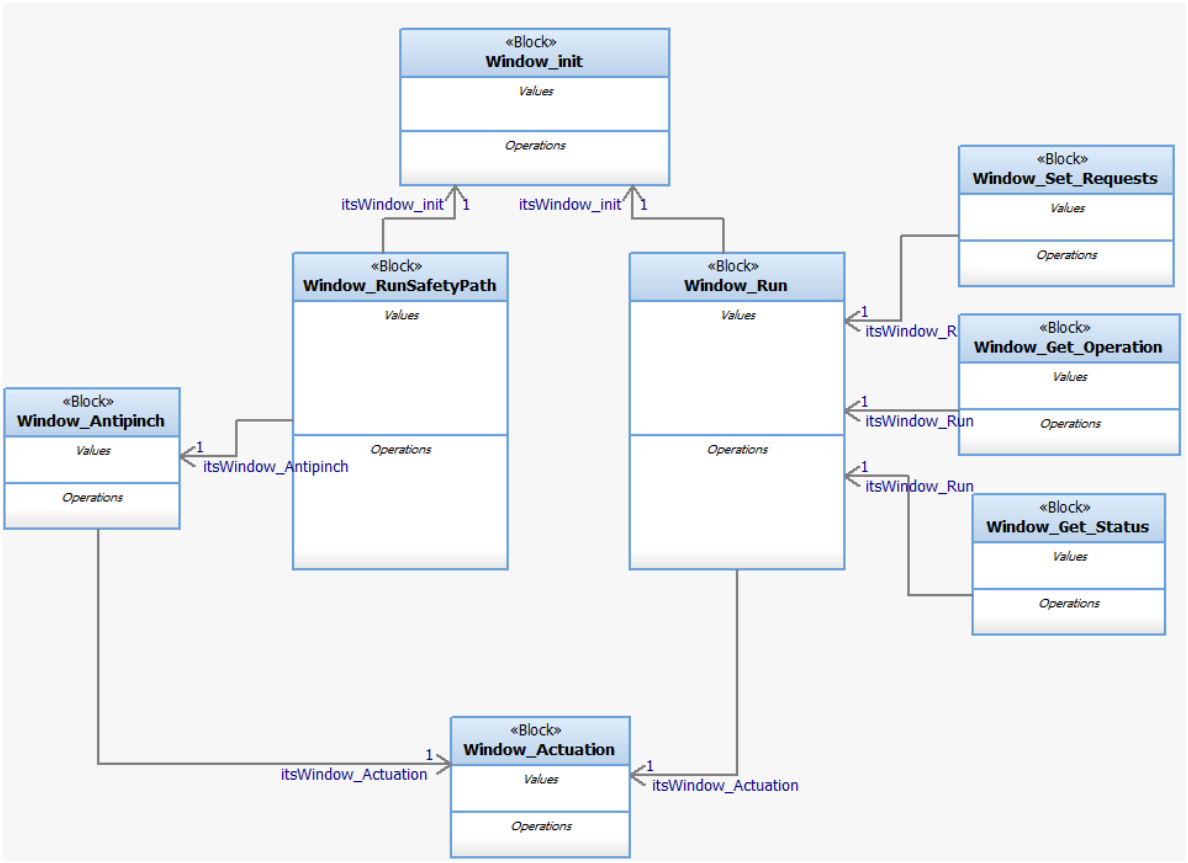
5 SW Component internal breakdown

This Software component will be separated in the following subsections:

- *Window_Get_Operation - indicates current window operation based on actuations and status*
- *Window_Get_Status – indicates current window status based on current position*
- *Window_Init– performs Window SWC initialization*
- *Window_Run - performs Window SWC Periodic Operations*
- *Window_RunSafetyPath - performs Window SWC Periodic safety Operations*
- *Window_Set_Requets – sets a Window Operation*
- *Window_Actuation – Executes window actuation (In the led model)*
- *Window_Antipinch – Runs the antipinch subroutine actuation*

5.1 Functional Decomposition

Overview of functions and their dependencies shown by a Static Function Tree



Window_Get_Operation, interface - indicates current window operation and serves as a public data access.

Window_Get_Status, interface– indicates current window status based and serves as a public data access.
Window_Init service– Initializes al values related to the SWC component ant sets up the correct data using the adc and dio interfaces
Window_Run, periodical 10ms - performs the periodic window Operation, specifically, everything related to the constant checking of request and Hardware Configuration, the related determination of status and operation, moreover the calling for the corresponding actuation.
Window_RunSafetyPath, periodical 10ms– performs the periodic window safety Operation, in this case, the constant checking for antipinch signals and the execution of the antipinch subprotocol when necessary
Window_Set_Requets, interface – sets a Window Operation, used only over the Window_Run
Window_Actuation, internal operation – Executes a window actuation following the led pattern stated on the architecture
Window_Antipinch, internal operation – Runs the antipinch subroutine, administrates all the operations related to this action, and secures a sequential operation

Function Description and Dynamic Behavior

5.2 Function *char* Window_Get_Operation (void)*

Description	Function that returns the current operation value and formats it to accommodate the interface
Parameter 1 <input| output| inout>	*This interface function requires no parameters*
Parameter 2..n	*This interface function requires no parameters*
Return Value	Returns one of the following strings: WINDOW_OPERATION_IDLE WINDOW_OPERATION_UP WINDOW_OPERATION_DOWN
Precondition	*No specific precondition.*
Post condition	*No specific post condition.*
Error Conditions	*An error should be riced when the string returned is not one from the previous stated list.*
Requirements	*The interface standard*

Dynamic Behavior

This function gives access to the window operation being executed through request, formats it to be the standard according to the interface and returns the correct string

5.3 Function *char* Window_Get_Status (void)*

Description	Function that returns the current window status based on the window specific set of positions
Parameter 1 <input| output| inout>	*This interface function requires no parameters*
Parameter 2..n	*This interface function requires no parameters*
Return Value	Returns one of the following strings: WINDOW_POSITION_OPEN WINDOW_POSITION_1 WINDOW_POSITION_2 WINDOW_POSITION_3 WINDOW_POSITION_4

	WINDOW_POSITION_5 WINDOW_POSITION_6 WINDOW_POSITION_7 WINDOW_POSITION_8 WINDOW_POSITION_9 WINDOW_POSITION_CLOSED WINDOW_POSITION_ERROR Depending on the position determined previously
Precondition	*No specific precondition.*
Post condition	*No specific post condition.*
Error Conditions	*An error should be riced when the string returned is not one from the previous stated list.*
Requirements	*The interface standard*

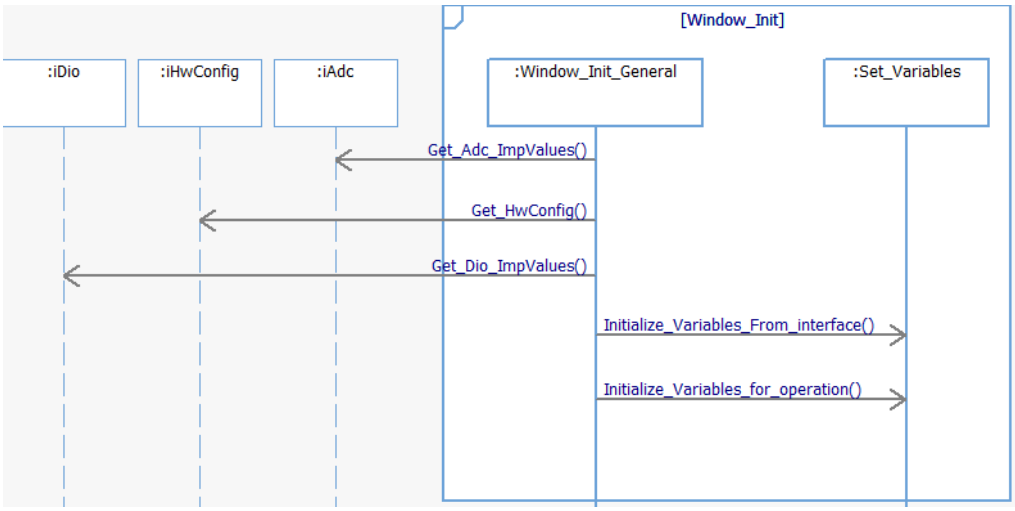
Dynamic Behavior

This function gives access to the window status being based on the current position, formats it to be the standard according to the interface and returns the correct string

5.4 Function *void Window_Init (void)*

Description	Function that initializes all common variables and data for the operation of the window. Sets up the global data, and uses the adc, HwCongfig and dio interfaces.
Parameter 1 <input| output| inout>	No parameters are needed on this function
Parameter 2.n	No parameters are needed on this function
Return Value	No return is specified, the function make all variable initialization through the modification of the global variables.
Precondition	*This function shall only be called once per operation cycle and SWC function, there is no need to make use of this operation twice. Every SWC that calls this function shall have the required global variables to modify through this operation*
Post condition	*No specific post condition.*
Error Conditions	*An error occurs when a value initialized is not compatible with the format of the function that makes the function call*
Requirements	*The interface standard*

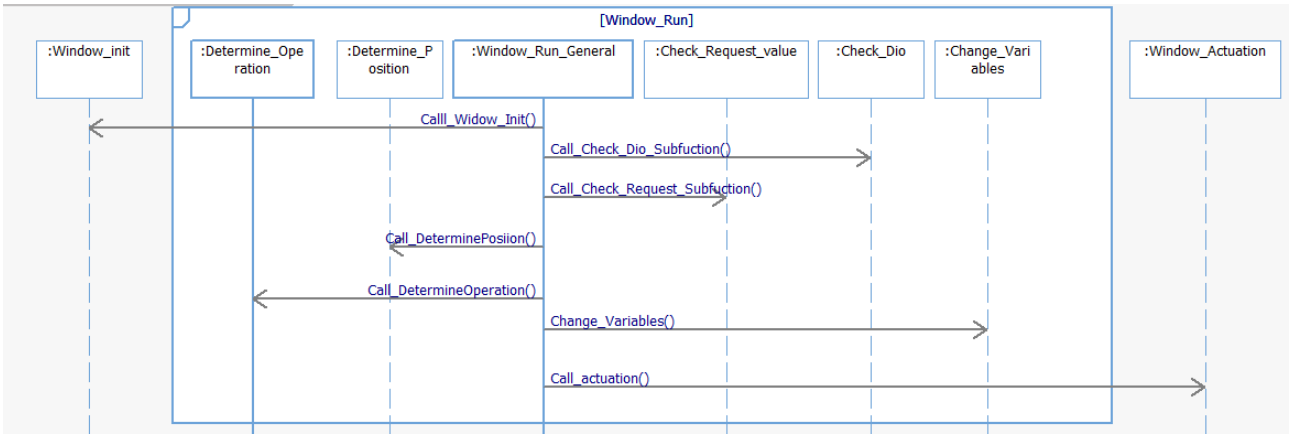
Dynamic Behavior



5.5 Function *void Window_Run(void)*

Description	Function used for the periodical checks of status and requests, the determination of the new operations, status and position, the call for actuation when needed an overall used to be the central point of execution of ECU window operations (Init, Calculations and actuations)
Parameter 1 <input| output| inout>	No parameters are needed on this function, designed to work independently when called.
Parameter 2.n	No parameters are needed on this function, designed to work independently when called.
Return Value	*This function gives no return.*
Precondition	*Every SWC that calls this function shall have the required global variables to reference and operate them when needed*
Post condition	*No specific post condition.*
Error Conditions	*An error should be reported when the system is unable to execute the function at the correct time and rate*
Requirements	*Operation and status determination related: DCU_SWR_111, DCU_SWR_112, DCU_SWR_113, DCU_SWR_114*

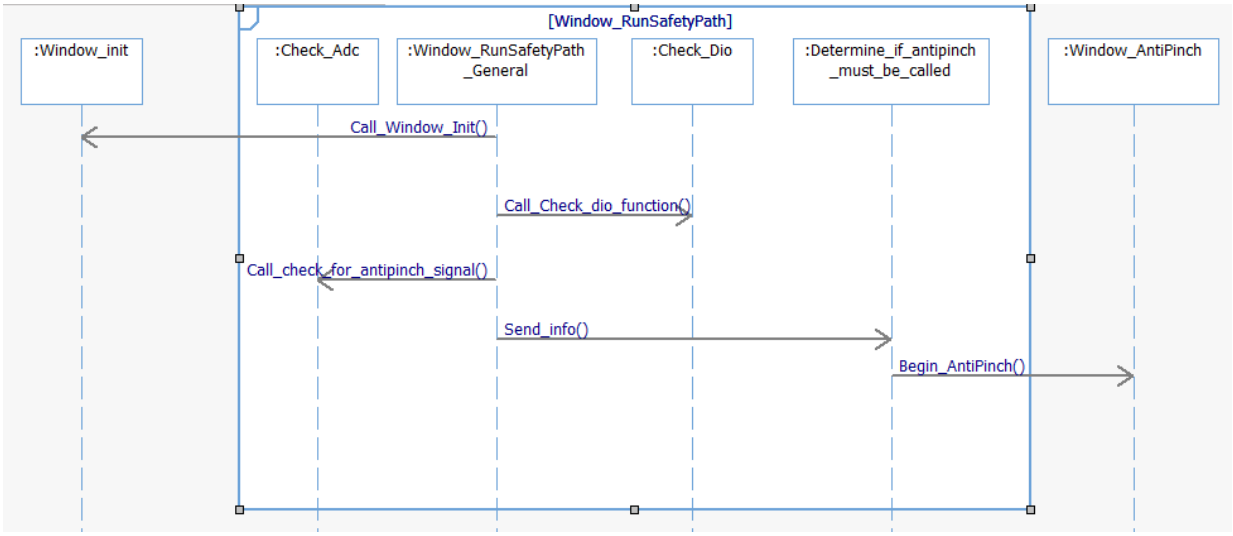
Dynamic Behavior



5.6 Function *void Window_RunSafetyPath(void)*

Description	Function used for the periodical checks of signals to execute the antipinch system when needed. In this case, the function takes care only of the detection for signals, the correct setup for the antipinch system, and the desition to call the subroutine
Parameter 1 <input| output| inout>	No parameters are needed on this function, designed to work independently when called.
Parameter 2.	No parameters are needed on this function, designed to work independently when called.
Return Value	*This function gives no return.*
Precondition	*Every SWC that calls this function shall have the required global variables to reference and operate them when needed*
Post condition	*No specific post condition.*
Error Conditions	*An error should be reported when the system is unable to execute the function at the correct time and rate*
Requirements	*No requirement directly filled.*

Dynamic Behavior



5.7 Function *char* Window_Set_Request (char*)*

Description	Function that sets a comand that the function run should use to decide and initiate a window request actuation.
Parameter 1 <input| output| inout>	*The parameter 1 should be a string with one of the following formats:* WINDOW_REQUEST_IDLE WINDOW_REQUEST_UP WINDOW_REQUEST_DOWN
Parameter 2..n	*This interface function requires no second parameter*
Return Value	Returns one of the following strings: WINDOW_OPERATION_IDLE WINDOW_OPERATION_UP WINDOW_OPERATION_DOWN

	Depending on the parameter given by the user
Precondition	*No specific precondition.*
Post condition	*No specific post condition.*
Error Conditions	*An error should be riced when the string returned is not one from the previous stated list.*
Requirements	*The interface standard*

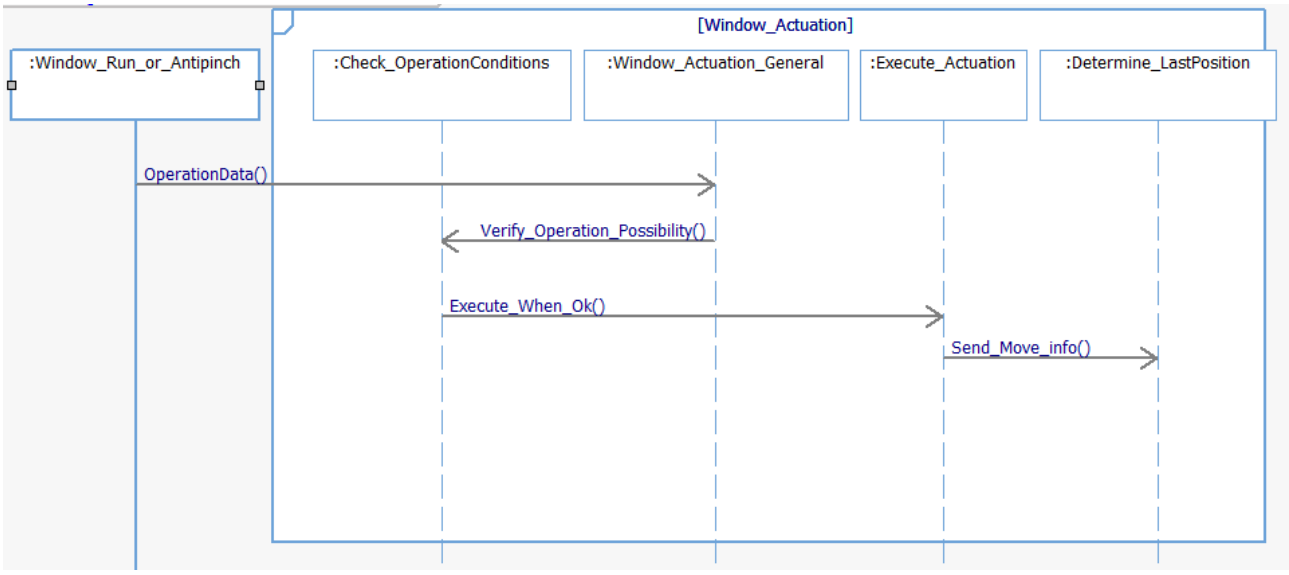
Dynamic Behavior

This function sets a window operation based on an external request, formats it to be the standard according to the interface and returns the information the Window_Run function needs.

5.8 Function *void Window_Actuation(char* Operation, char* Position)*

Description	Function used to execute the window actuations indicated by the operation following the position conditions. This function is design to be an internal operation of the SWC that operates always together with a run or a runsafety function. Uses the correct configuration set on the global variables to work correctly
Parameter 1 <input| output| inout>	char* Operation is a string which indicates the action that the actuation module should produce
Parameter 2..n	char* Position gives the actuation module the necessary information to work around certain conditions
Return Value	*This function gives no return.*
Precondition	*This function shall only be called directly by the Window_Run and Window_RunSafetly functions, even though these are both of public access*
Post condition	*No specific post condition.*
Error Conditions	*An error should be reported when the system is unable to execute the correct window actuation with the parameters given*
Requirements	*DCU_SWR_141, DCU_SWR_142*

Dynamic Behavior



5.9 Function *void Window_AntiPinch (void)*

Description	Function used to administrate all the actions related to the antipinch routine. This operation shall only be called by the function Window_RunSafetyPath and is responsible of the sequenced window actuations with preset instructions, as well as the inhibitions of actuations for 15 seconds
Parameter 1 <input| output| inout>	No parameters are needed on this function
Parameter 2..n	No parameters are needed on this function
Return Value	*This function gives no return, all changes shall be made to the important global variables.*
Precondition	*This function shall only be called by the Window_RunSafetyPath function and never directly*
Post condition	*No specific post condition.*
Error Conditions	*An error should be reported when the system is unable to execute the full set of actuations and the inhibition of further actuations during 15 seconds*
Requirements	*DCU_SWR_110*

Dynamic Behavior