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

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 HWConfiguration 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*
Jumper	*Component that manually configures peripherals*

Abbreviations

HWConfiguration: Current SW module
ECU: Electronic control unit
SWC: Software components

References

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

3 Realization constraints and targets

This Software component shall be able to identify the correct configuration based on ECU jumpers, moreover, provide the corresponding information to the Door/Window app and setup. The component shall also take care of the configuration, only set it once per operating cycle a never change it through its entire duration

4 SW Conceptual design

The main function of this software component is providing other SWCs with information about the configuration of each ECU and provide an easy way to verify if a specific configuration is present on the module depending on a configuration mask. Other main functions include: the lectures of the jumpers through the corresponding interface and the keeping of the data

The system boundary in this case includes only the operations related to the hardware configuration, the dio interface to get the jumpers states, and the configuration interface to make the configuration public access.

This software component will use the external interfaces: iHwConfig and iDio to communicate with different SWC, and normal int function calls to communicate internally

As global data, this SWC will only use a counter to determine if it´s the first time the function HwConfig_info is being called

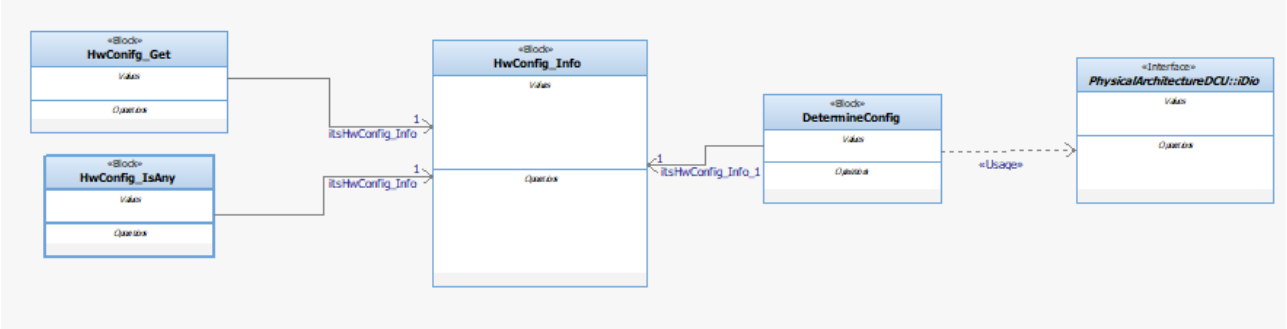
5 SW Component internal breakdown

This Software component may be simple, but the team has decided to separate the functions in to 4 subcomponents:

- *DetermineConfig Determines configuration.*
- *HwConfig_info – Data manager*
- *HwConfig_Get() – Part of the interface unit*
- *HwConfig_IsAny() – Part of the interface unit*

5.1 Functional Decomposition

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



DetermineConfig, Internal operation – Determines the HwConfig based on the jumper reads using the iDio interface
HwConfig_ingfo, internal operation – if the system is on a new operating ciclye, Initalizes the configuration value as undetermined, and calls DetermineConfig to learn the configuration selected. If the system is not on a new operating cycle, the function just returns the actual Config, value
iHwConfig_Get, interface – Part of the interface, calls HwConfig_info, gives format to the Config value copy and makes it accesible to other SWC with a public call.
iHwConfig_IsAny, interface – Part of the interface, reads a config mask, calls_info, and compares the information to return a boolean vañue. Function with public acces.

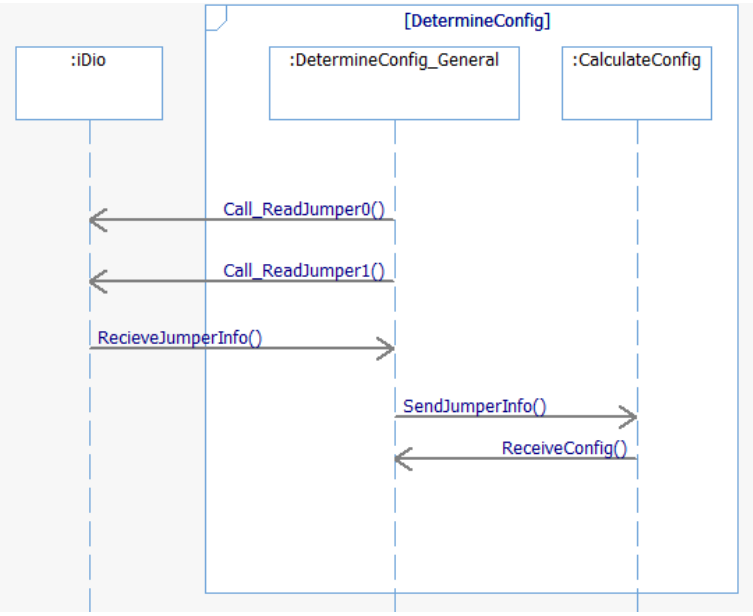
Function Description and Dynamic Behavior

5.2 Function *int DetermineConfig (void)*

Description	Determines the hwconfiguration based on the jumper reads using the iDio interface. Analices the information based on the next table: Jumper 0 Jumper 1 Variant Behavior 0 0 Driver Door 0 1 Passenger Door 1 0 RearLeft Door 1 1 RearRight Door An returns a correspondig int value
Parameter 1 <input| output| inout>	*The function uses the iDio public interface, for that reason there is no need for parameters*

Parameter 2..n	*The function uses the iDio public interface, for that reason there is no need for parameters*
Return Value	Returns 0 if there is an unknown convination, 1 if the fuction identifys the Driver door variant, 1 if the fuction identifys the Driver door variant, 2 if the fuction identifys the passanger door variant, 3 if the fuction identifys the RearLeft Door variant, 4 if the fuction identifys the RearRight Door variant
Precondition	This function can only be called in when the function HwConfig info detects a new operating cycle
Post condition	*No specific post condition.*
Error Conditions	*Error reported of the values gotten from the iDio interface do not relate to the configuration table.*
Requirements	*DCU_SWR_152*

Dynamic Behavior

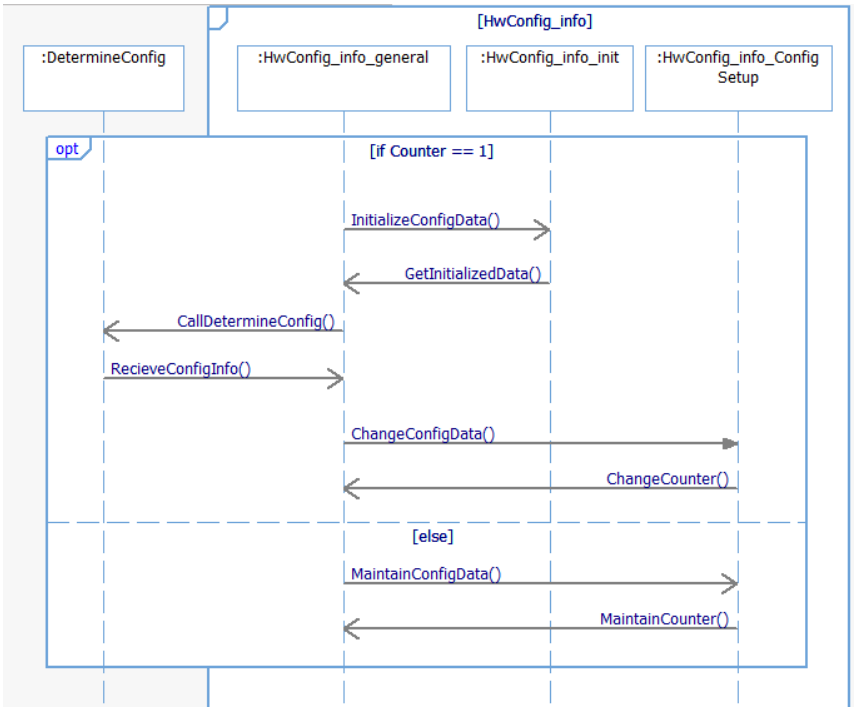


5.3 Function *Int HwConfig_info (void)*

Description	Function that contains the info about the HwConfig. Initializes the configuration value and actualizes it once per operating cycle calling DetermineConfig. If the initialization phase has already occurred, then always keeps the same data.
Parameter 1 <input| output| inout>	No parameters are needed on this function
Parameter 2..n	No parameters are needed on this function
Return Value	*Returns an int value that corresponds with the configuration determination: 0 if there is an unknown convination, 1 if the fuction identifys the Driver door variant, 1 if the fuction identifys the Driver door variant, 2 if the fuction identifys the passanger door variant, 3 if the fuction identifys the RearLeft Door variant, 4 if the fuction identifys the RearRight Door variant.* *Makes it possible for other functions to access a copy of the current config*

	value
Precondition	This function should be called only by other functions in this SWC, the extrernal accesses are made by the interfaces.
Post condition	*No specific post condition.*
Error Conditions	*An error ocurres when the value returned is not one ot the previously mentioned*
Requirements	*Comments DCU_SWR_151 and DCU_SWR_152*

Dynamic Behavior



5.4 Function *char* HwConfig_Get(void)*

Description	Function that provides public access copy of the hardware configurations. Calls HwConfig_info and formats the data to the interface standard.
Parameter 1 <input| output| inout>	No parameters are needed on this function
Parameter 2..n	No parameters are needed on this function
Return Value	Return: HWCONFIG_UNKNOWN: Unknown HW configuration HWCONFIG_DRIVER: HW variant is Driver configuration HWCONFIG_PASSENGER: HW variant is Driver configuration HWCONFIG_REAR_LEFT: HW variant is Driver configuration HWCONFIG_REAR_RIGHT: HW variant is Driver configuration
Precondition	*No specific precondition.*
Post condition	*No specific post condition.*
Error Conditions	*An error occurred when the value returned is not one of the previously mentioned*
Requirements	*The interface standard*

Dynamic Behavior

This function calls the data returned by the HwConfig_info, formats it to be the standard according to the interface and returns this calculation with a public access.

5.5 Function *bool HwConfig_IsAny(char* config, char* ConfigMask)*

Description	Function used to check if a specific configuration is present on the module depending on a configuration mask. Verifies with the saved configuration value
Parameter 1 <input| output| inout>	config: Configuración reference
Parameter 2..n	ConfigMask: Group of configurations to be compare if there are present.
Return Value	*A Boolean value that indicates the presence or the lack of the element.*
Precondition	*No specific precondition.*
Post condition	*No specific post condition.*
Error Conditions	*An error occurred when the value returned is not one of the previously mentioned*
Requirements	*The interface standard*

Dynamic Behavior