UNIVERSIDADE FEDERAL DE PERNAMBUCO

STELLANTIS

Start Stop Functional Requirements Document

Eliane da Silva Rodrigues Cardozo

Klaus Malone Martins Damasceno

Lucas Edson Silva da Paixao

Marilia Caroline de Almeida Cardoso Silva

Thyago Chrystiann da Silva Lopes

September 10th, 2022

Contents

1 Introduction

Today, saving fuel is considered sensible not only for financial but also for ecological reasons. But when the idea of the automatic start-stop system was born, it was the oil price crisis of the time that provided the need for fuel-saving measures. Toyota invented the first automatic Start-Stop system in the mid-1970s[1].

Volkswagen and Audi introduced their versions of the start-stop system in the 1980s. Together with a 5-speed gearbox, aerodynamic bodywork, and electronic fuel consumption indicator, these improvements were intended to significantly reduce fuel consumption[2].

A new generation of start-stop systems followed in the 2000s. This sophisticated and robust technology marked the beginning of the Start-Stop success story.

1.1 Motivation

Depending on the style of driving, start-stop technology can achieve fuel savings of up to 15%[2]. However, the effectiveness of the system depends on many factors, including the area of use. During urban use, the automatic Start-Stop system causes the engine to switch off at a idle state. Naturally, it is less effective on long motorway journeys.

1.2 Target Audience

Cars equipped with start/stop systems will benefit drivers that use their car for everyday tasks inside city traffic, which demands high idle time.

1.3 User Needs

The system was designed with the aim of reducing fuel consumption, through an automatic engine management system, adapting to the functional needs of vehicle use. In addition, it also reassures the driver regarding the general condition of the car when the system is in use.

2 Conventions Adopted

To facilitate the search and reference of requirements throughout the document, an identification convention will be adopted for each type of requirement:

- System requirements are identified with the tag [SR_#];
- Hardware requirements are identified with the tag [HR_#];
- Software requirements are represented in the format [SWR_#];
- ”#” represents the requirement number.

3 Project outline

The elicitation of requirements present in this document refers to the development of a Start/Stop system for integration in an automatic car. The main purpose of the system is to avoid wasting fuel and excess emissions when the engine is idle. In order to do that, the system needs to use sensors to receive input values such as brake pressure, accelerator position, battery charge level, and engine temperature to determine the best time to turn off and restart the engine based on this information.

In addition to determining the engine shutdown/startup, the system must also provide the user with an option to disable it, using a button, if he does not want to use it. Another aspect present in the application is the concern with life safety, identifying whether the seatbelt is correctly positioned, whether the car’s hood is closed and if the car’s doors are open. All these restrictions will be described in the requirements section.

4 Requirements

4.1 System Requirements

- [SR_001] The system must be able to work perfectly in automatic cars;

- [SR_002] All sensors of the components responsible for the operation of the system must be present and fully operational in the vehicle;
- [SR_003] The user will be able to enable or disable the system, through a button on the panel;
- [SR_004] The system must be able to completely stop the engine when stepping on the brake for at least 5 seconds, and restart it when fully releasing the brake, with a maximum latency of 2 seconds;
- [SR_005] The system must be capable of turning the motor off if the speed falls below 5 km/h;
- [SR_006] The system should only work if the engine temperature is between 90ºC and 104ºC;
- [SR_007] The system must only work if the battery charge status is over 50%;
- [SR_008] The system must not be enabled when the AC is at maximum speed and the state of charge is below 70
- [SR_009] The system must be able to check if the seatbelts, hood, trunk and door are properly closed and if not, it should emit a warning on the main panel;

4.2 Hardware Requirements

- [HR_001] The sensors presently used on the system must be thoroughly verified;
- [HR_002] A failure on the speed sensor, temperature sensor, brakes sensor, battery sensor, or the Start-Stop ECU sensor must entail a complete shutdown;
- [HR_003] In case of functional failure on any of the sensors present in the hood, seatbelts, trunk, doors and/or AC, the system must work normally, although emitting a warning on the main panel;
- [HR_004] In case of any simultaneous failure of the sensors present in the hood, seatbelts, doors and/or AC, the system must not be enabled.

4.3 Software Requirements

- [SWR_001] The software must the developed in C, using the GCC compiler, with the unit tests done on the Unity framework;
- [SWR_002] The system input variables must be stored in a bit field structure, with the size defined in order to allocate the largest predicted value of these variables;
- [SWR_003] All integer variables will be defined according to the stdint.h library;
- [SWR_004] Variables that would be stored as float or double must be stored in custom variables using typedef, following fixed-point logic, with their respective sizes defined in order to allocate the largest predicted value of these variables and with appropriate decimal precision;
- [SWR_005] The software must have the functionality to turn the system on and off completely;
- [SWR_006] The software must be able to determine if the motor's temperature is under the interval determined beforehand for optimal system usage;
- [SWR_007] The software must be able to identify if the battery state of charge is above 50%;
- [SWR_008] The software must be enabled with functions that identify if the hood, doors, or trunk is open or closed and if the seatbelt is fastened or not;
- [SWR_009] The software must be enabled with a function that emits warnings on the main panel, in case the hood, doors, or trunk is open or if the seatbelt is not fastened.
- [SWR_010] The software must be enabled with a function that checks if all the sensors are working perfectly or not;
- [SWR_011] The software must be enabled with a function that receives the motor speed value and determines if it is below the chosen limit of 5 km/h to allow system activation;
- [SWR_012] The software must be enabled with a function that checks if the brakes are being pressed or not, and, in case it is for at least 5 seconds, allows for the system to be activated;

- [SWR_013] The software must be enabled with a function that checks if, after releasing the brakes, the engine is turned on within, at maximum, 2 seconds;
- [SWR_014] The software must be enabled with a function that collects the AC speed and verifies if it is at maximum, in order to allow or not the system activation, should the battery state of charge be below 70

5 Block Diagram

In the following block diagram, we can observe the components that form the Start/Stop system to be designed and how they interact with each other. It is important to note that the subsystems described are the ones that were created to meet the proposed requirements.

The following subsystems were represented:

- checkHardware: Check whether all sensors are fully operational;
- checkEngineTemp: Checks whether the engine temperature is within the specified range;
- checkBattery: Checks the battery charge status;
- checkSpeed: Checks whether the engine speed is below the established limit;
- checkBrake: Checks whether the brake is activated or not;
- checkAirCond: Checks the air conditioning speed and battery status;
- checkSafety: Checks whether the security artifacts are properly configured;
- checkSystemLatency: Variable to indicate whether the system latency value is greater than the allowed limit.

Also represented is the button that activates and deactivates the system (button_pressed), in addition to the variable that indicates whether the system response time is within the determined value (latency_status). There is also the representation of variables that indicate whether the Start/Stop

system is activated or not (start_stop_status), and whether the engine should be turned off or not (set_engine_status).

The systems listed previously have all their outputs compared so that the general output of this relationship only allows the Start/Stop to be activated if all of their conditions are met.

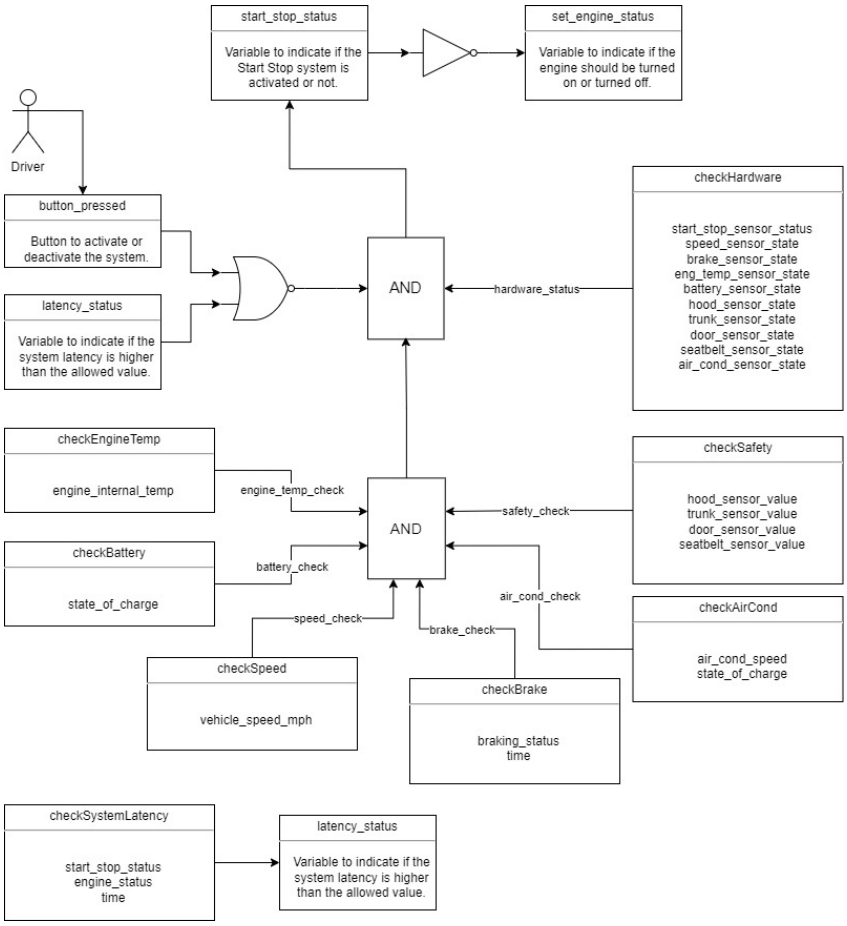

Figure 1: Block diagram for the Start/Stop system.

6 Version Control

To monitor the development of this document, the following table was created, which shows the version number, the date of its creation, the main change that occurred and which team member performed the respective task.

Version	Date	Change Description	Team Member
First Version	06/08/2023	First draft of the requirements document	All team members
Second Version	07/08/2023	Division of requirements into basic, system and software requirements	Eliane and Marilia
Third Version	08/08/2023	Drawing of the block diagram	All team members
Fourth Version	09/08/2023	Creation of introduction, motivation, project explanation sections, etc.	Klaus
Fifth Version	28/08/2023	Addition of more requirements to adapt to the model created in MATLAB	Eliane, Lucas and Marilia
Sixth Version	07/09/2023	Correction of requirements according to Stellantis feedback	All team members
Seventh Version	15/09/2023	Rearrangement of requirements into System, Hardware and Software Requirements, according to code creations	Eliane and Marilia
Eighth Version	28/09/2023	Continuation of document corrections, following the evolution of tests created and code rewriting	Eliane and Marilia
Ninth Version	02/10/2023	Finalization of the requirements document	All team members

Table 1: Table to follow the changes made through the document's development.

7 Table of Variables

Variable	Description	Size	Min Value	Max Value	Unit
button_pressed	Variable to determine if the start/stop button is pressed or not	uint8_t	0	1	binary
engine_on	Variable to determine if the engine should be turned on or off	uint8_t	0	1	binary
start_stop_sensor_status	Variable to determine if the start/stop system is properly working	uint8_t	0	1	binary
brake_sensor_status	Variable to determine if the brake sensor is properly working	uint8_t	0	1	binary
engine_temp_ sensor_status	Variable to determine if the engine sensor is properly working	uint8_t	0	1	binary
battery_sensor_status	Variable to determine if the battery sensor is properly working	uint8_t	0	1	binary
hood_sensor_status	Variable to determine if the hood sensor is properly working	uint8_t	0	1	binary
trunk_sensor_status	Variable to determine if the trunk sensor is properly working	uint8_t	0	1	binary
door_sensor_status	Variable to determine if the door sensor is properly working	uint8_t	0	1	binary

seatbelt_sensor_status	Variable to determine if the seatbelt sensor is properly working	uint8_t	0	1	binary
air_cond_sensor_status	Variable to determine if the air conditioner sensor is properly working	uint8_t	0	1	binary
hood_sensor_value	Variable to determine if the hood is open or not	uint8_t	0	1	binary
trunk_sensor_value	Variable to determine if the trunk is open or not	uint8_t	0	1	binary
door_sensor_value	Variable to determine if the door is open or not	uint8_t	0	1	binary
seatbelt_sensor_value	Variable to determine if the seatbelt is well placed or not	uint8_t	0	1	binary
vehicle_speed_mph	Variable to determine the vehicle speed	unit16_t -> fixed7_9	0	127	mph
braking_status	Variable to determine if the brake is pressed or not	uint8_t	0	1	binary
air_cond_speed	Variable to indicate the air conditioner speed	unit8_t	0	2	count
engine_internal_temp	Variable to determine the engine internal temperature	unit16_t -> fixed7_9	0	127	ºC
time	Variable to indicate the simulation step time	uint16_t	0	65535	seconds
state_of_charge	Variable to indicate the state of charge of the vehicle's battery	uint16_t -> fixed1_15	0	100	percentage

set_engine_status	Variable to indicate if the engine should be turned on or turned off	uint8_t	0	1	binary
start_stop_status	Variable to indicate if the start stop system is activated or not	uint8_t	0	1	binary
latency_status	Variable to indicate if the system latency is higher than the allowed value	uint8_t	0	1	binary

References

[1] Toyota Saga. *Como funciona o sistema start-stop?* 2023. URL: https://www.sagatoyota.com.br/toyota-df-asa-norte/blog/como-funciona-o-sistema-start-stop-.

[2] Varta Battery World. *Why a working Start-Stop system is so important to save fuel*. 2023. URL: https://batteryworld.varta-automotive.com/en-gb/saving-fuel-with-start-stop.