Global Design Document

**Version 1.2**

**February 25, 2020**

**Electric Blender**

**PO2\_EBL**

**By: Mostafa Ramadan**

**Status: Draft**

**Document Status:**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Status | Author | Date |
| 1.2 | **Draft** | **Mostafa Ramadan** | **09– 03 - 2020** |

**Document History:**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Date | Change Description |
| 1.0 | **Mostafa Ramadan** | **25-02-2020** | **Adding document status and document history table** |
| 1.0 | **Mostafa Ramadan** | **27-02-2020** | **Adding the Static architecture & s/w sequence diagram** |
| 1.0 | **Fatima Gomaa** | **27-02-2020** | **Adding the s/w context diagram** |
| 1.0 | **Mostafa Ramadan** | **28-02-2020** | **Adding the Public APIs for the software** |
| 1.1 | **Mostafa Ramadan** | **02-03-2020** | **Adding the APIs of the components from the static architecture (PWM-ADC-GPIO-Timer- HSwitch-Voltage Shifter - Motor Driver – Button – Voltage analyzer – Blender)** |
| 1.2 | **Mostafa Ramadan** | **09-03-2020** | * **Change the version of the GDD document to V1.2** * **Remove the CYRS document from the reference table as the GDD document doesn’t depend on it** * **Edit the sequence diagram to return arrows for the blender component** * **Edit the naming of the static architecture components** * **Edit the APIs (add the value – range and the description for all of the APIs)** * **Remove the delay component from the static architecture and its API** * **Edit the S/W Feature diagram with adding the way the switch component would communicate with the Timer component** * **Change the alignment of the APIs from stack to layers** |

**Reference Table:**

|  |  |  |
| --- | --- | --- |
| Document | Version | Author |
| SRS | **1.6** | **Mostafa Ramadan** |

**Table Of Contents**

[Table of Contents iii](#_Toc32164885)

[Table of Figures iv](#_Toc32164886)

[1.0. Introduction 6](#_Toc32164887)

[1.1.Purpose 6](#_Toc32164888)

2.0 Software Feature Diagram …….………………………6  
3.0 Static Architecture..……………………………………7  
4.0 Software Sequence Diagram...……………………...…8  
5.0 Public APIs…………………………………………….9  
 5.1 APP Layer …………………………………………...9  
 5.1.1 Button Component…………………………..…9  
 5.1.2 Blender Component………………………….…10  
 5.1.3 Voltage Analyzer Component………………… 12

5.2 ECUAL Layer..……….………………………………13  
 5.2.1 HSwitch Component …………………………...13  
 5.2.2 Motor Driver Component……………………….14  
 5.2.3 Voltage Shifter Component……………………..16

5.3 MCAL Layer…………………………………………..17

5.3.1 PWM Component…...……………..……………17

5.3.2 Timer Component……………………………..…19

5.3.3 ADC Component………………………………..20

**Table of Figures**

[Figure 1 – Software Feature Diagram 6](#_Toc33267998)

[Figure 2 – Static Architecture 7](#_Toc33267999)

[Figure 3 – Software Sequence diagram 8](#_Toc33268000)

1. ***Introduction***
   1. ***Purpose***

The purpose of this document is to present the flow of the software through the system and shows the APIs that will cover the functionality of the software. And how the software shall act and react against certain parameters.

This document will explain the features of the system from the functional software manner.

1. ***Software Feature Diagram***
2. ***Static Architecture***
3. ***Software Sequence Diagram***
4. ***Public APIs***
   1. ***APP Layer***
      1. *Button Component*

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_ PO2\_EBL\_Electric\_Blender\_GDD\_001-1.0 | **Covers** | Covers\_ PO2\_EBL\_Electric\_Blender\_SRS\_001-1.0 |
| **Author** | Mostafa Ramadan | **DATE** | 28/2/2020 |
| **API** | errState Button\_Pressed(u8 Button\_CH, u8 \*BData); | | |
| **Description** | This Function shall read if the button is pressed or not. | | |
| **Inputs** | u8 Switch\_CH:-  **Range:** 0 – 255 **Value:** would be macro defined as  #define SWITCH\_CH0 0  u8 \* BData :- **Range:** 0 - 255 **Value :**  pressed(0) - not pressed(1) | **Outputs** | errStatus:- Range: NOK(0) – OK(1) |

* + 1. Blender Component

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_ PO2\_EBL\_Electric\_Blender\_GDD\_002-1.0 | **Covers** | Req\_ PO2\_EBL\_Electric\_Blender\_SRS\_006-1.4 |
| **Author** | Mostafa Ramadan | **DATE** | 2/3/2020 |
| **API** | errState BlenderStart(u8 speed); | | |
| **Description** | This Function shall control the blender motor speed | | |
| **Inputs** | u16 speed:   **Range:** 0 - 255  **Value:**   1. Speed 1 = 30% \* 255 = 77 2. Speed 2 = 60% \*255 = 153 3. Speed 3 = 100% \*255 = 255 | **Outputs** | errStatus:- Range: NOK(0) – OK(1) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Req\_ID** | Req\_ PO2\_EBL\_Electric\_Blender\_GDD\_003-1.0 | **Covers** | Req\_ PO2\_EBL\_Electric\_Blender\_SRS\_009-1.4 & Req\_ PO2\_EBL\_Electric\_Blender\_SRS\_011-1.5 | | |
| **Author** | Mostafa Ramadan | **DATE** | 2/3/2020 | | |
| **API** | errState SystemShutdown(void); | | | | |
| **Description** | This Function shall shutdown the blender if there is voltage error to the system | | | | |
| **Inputs** | n/a | | | **Outputs** | errStatus:- Range: NOK(0) – OK(1) |

* + 1. Voltage Analyzer Component

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Req\_ID** | Req\_ PO2\_EBL\_Electric\_Blender\_GDD\_004-1.0 | **Covers** | Req\_ PO2\_EBL\_Electric\_Blender\_SRS\_008-1.5 & Req\_ PO2\_EBL\_Electric\_Blender\_SRS\_010-1.5 | | |
| **Author** | Mostafa Ramadan | **DATE** | 2/3/2020 | | |
| **API** | errState ReadCurrentVoltage(u8 VCC\_CH,u16 \*VData); | | | | |
| **Description** | This Function shall Read the input voltage to the blender | | | | |
| **Inputs** | U8 VCC\_CH:  **Range:** 0 - 255  **Value:** shall be defined as macro #define VCC\_CH0 0 **Description:** the current voltage is connected to which channel  U16 \* VData: **Range :** 0 - 65535  **Description:** the value of the current voltage applied to the system | | | **Outputs** | errStatus:- Range: NOK(0) – OK(1) |

* 1. ECUAL Layer
     1. HSwitch Component

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_ PO2\_EBL\_Electric\_Blender\_GDD\_005-1.0 | **Covers** | Covers\_ PO2\_EBL\_Electric\_Blender\_SRS\_001-1.0 |
| **Author** | Mostafa Ramadan | **DATE** | 28/2/2020 |
| **API** | errState Read\_SwitchState(u8 Switch\_CH, u8 \*BData); | | |
| **Description** | This Function shall get the status of the Switch and return an error status in case it met the required functionality or not. | | |
| **Inputs** | u8 Switch\_CH: **Range:** 0 – 255 **Value:** would be macro defined as  #define SWITCH\_CH0 0  u8 \* BData : **Range:** 0 – 255 **Value:** 0(pressed) – 1(not pressed) | **Outputs** | errStatus:- Range: NOK(0) – OK(1) |

* + 1. Motor Driver Component

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Req\_ID** | Req\_ PO2\_EBL\_Electric\_Blender\_GDD\_006-1.0 | **Covers** | Req\_ PO2\_EBL\_Electric\_Blender\_SRS\_005-1.4 | | |
| **Author** | Mostafa Ramadan | **DATE** | 2/3/2020 | | |
| **API** | errState SetMotorSpeed(u8 SpeedState); | | | | |
| **Description** | This Function shall control the duty cycle of PWM to feed the motor speed | | | | |
| **Inputs** | U8 SpeedState: - **Range:** 0 - 255  **Value:** would be Macro defined as:  SPEED\_1 🡺 from 0% up to 30% of the duty cycle.  SPEED\_2 🡺 from 30% up to 60% of the duty cycle  SPEED\_3 🡺 from 60% up to 100% of the duty cycle | | | **Outputs** | errStatus:- Range: NOK(0) – OK(1) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Req\_ID** | Req\_ PO2\_EBL\_Electric\_Blender\_GDD\_007-1.0 | **Covers** | Req\_ PO2\_EBL\_Electric\_Blender\_SRS\_009-1.4 & Req\_ PO2\_EBL\_Electric\_Blender\_SRS\_011-1.5 | | |
| **Author** | Mostafa Ramadan | **DATE** | 2/3/2020 | | |
| **API** | errState StopMotor(void); | | | | |
| **Description** | This Function shall stop the motor of the blender | | | | |
| **Inputs** | n/a | | | **Outputs** | errStatus:- Range: NOK(0) – OK(1) |

* + 1. Voltage shifter Component

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Req\_ID** | Req\_ PO2\_EBL\_Electric\_Blender\_GDD\_008-1.0 | **Covers** | Req\_ PO2\_EBL\_Electric\_Blender\_SRS\_008-1.5 & Req\_ PO2\_EBL\_Electric\_Blender\_SRS\_010-1.5 | | |
| **Author** | Mostafa Ramadan | **DATE** | 2/3/2020 | | |
| **API** | errState ReadInputVoltage(u8 Vin\_CH,u16 \*VData); | | | | |
| **Description** | This Function shall Read the input voltage to the system | | | | |
| **Inputs** | U8 Vin\_CH:  **Range:** 0 - 255  **Value:** shall be defined as macro #define VCC\_CH0 0 **Description:** the input voltage is connected to which channel  U16 \* VData:  **Range :** 0 - 65535  **Description:** the value of the current voltage applied to the system | | | **Outputs** | errStatus:- Range: NOK(0) – OK(1) |

* 1. MCAL Layer  
     1. PWM Component

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Req\_ID** | Req\_ PO2\_EBL\_Electric\_Blender\_GDD\_009-1.0 | **Covers** | Req\_ PO2\_EBL\_Electric\_Blender\_SRS\_005-1.4 | | |
| **Author** | Mostafa Ramadan | **DATE** | 2/3/2020 | | |
| **API** | errState SetDutyCycle(u8 Duty\_Cycle); | | | | |
| **Description** | This Function shall Set the Duty Cycle of the PWM. | | | | |
| **Inputs** | u8 Duty\_Cycle:  **Range:** 0 - 255  **Value:** would be Macro defined as:  SPEED\_1 🡺 from 0% up to 30% of the duty cycle.  SPEED\_2 🡺 from 30% up to 60% of the duty cycle  SPEED\_3 🡺 from 60% up to 100% of the duty cycle | | | **Outputs** | errStatus:- Range: NOK(0) – OK(1) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Req\_ID** | Req\_ PO2\_EBL\_Electric\_Blender\_GDD\_010-1.0 | **Covers** | Req\_ PO2\_EBL\_Electric\_Blender\_SRS\_009-1.4 & Req\_ PO2\_EBL\_Electric\_Blender\_SRS\_011-1.5 | | |
| **Author** | Mostafa Ramadan | **DATE** | 2/3/2020 | | |
| **API** | errState StopPWM(void); | | | | |
| **Description** | This Function shall shutdown the PWM peripheral. | | | | |
| **Inputs** | n/a | | | **Outputs** | errStatus:- Range: NOK(0) – OK(1) |

* + 1. Timer Component

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_ PO2\_EBL\_Electric\_Blender\_GDD\_011-1.0 | **Covers** | Req\_ PO2\_EBL\_Electric\_Blender\_SRS\_002-1.5 & Req\_ PO2\_EBL\_Electric\_Blender\_SRS\_014-1.5 |
| **Author** | Mostafa Ramadan | **DATE** | 2/3/2020 |
| **API** | errState Start\_Timer(u16 time\_Value); | | |
| **Description** | This Function shall start the timer. | | |
| **Inputs** | U16 time\_Value :-  **Range:** 0 - 65535  **Value:** 200ms | **Outputs** | errStatus:- Range: NOK(0) – OK(1) |

* + 1. ADC Component

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Req\_ID** | Req\_ PO2\_EBL\_Electric\_Blender\_GDD\_012-1.0 | **Covers** | Req\_ PO2\_EBL\_Electric\_Blender\_SRS\_008-1.5 & Req\_ PO2\_EBL\_Electric\_Blender\_SRS\_010-1.5 | | |
| **Author** | Mostafa Ramadan | **DATE** | 2/3/2020 | | |
| **API** | errState GetADCValue(u8 ADC\_CH,u16 \*VData); | | | | |
| **Description** | This Function shall get the value applied to the ADC | | | | |
| **Inputs** | U8 ADC\_CH:  **Range:** 0 - 255  **Value:** shall be defined as macro #define ADC\_CH0 0 **Description:** the input Signal is connected to which channel  U16 \* VData: **Range :** 0 - 65535  **Description:**  the value of the current Signal applied to the ADC | | | **Outputs** | errStatus:- Range: NOK(0) – OK(1) |