Component Design Document

For

<KENOVO BLENDER JOINT>

<Application Layer>

**Table of Contents**

Revision History 2

*1. Introduction* 3

1.1 Objective 3

1.2 Context Diagram 3

2. *External Interface* 4

3. *Static Design* 6

3.1 Files 6

3.2 Types 6

3.3 Symbol Define <#define> 6

3.4 Const 6

3.5 Interface (Services) 6

*4.* *Dynamic Design* 8

4.1 Mode Management 8

4.2 Sequence Diagram 8

*5.* *Shared Resources* 8

5.1 Analysis 8

5.2 Protection 8

6. *Configuration Parameters* 9

6.1 Pre-compile time 9

6.2 Link time 9

6.3 Post-build 9

*7.* *Configuration Constrains* 9

*8.* *Integration Constrains* 9

Revision History

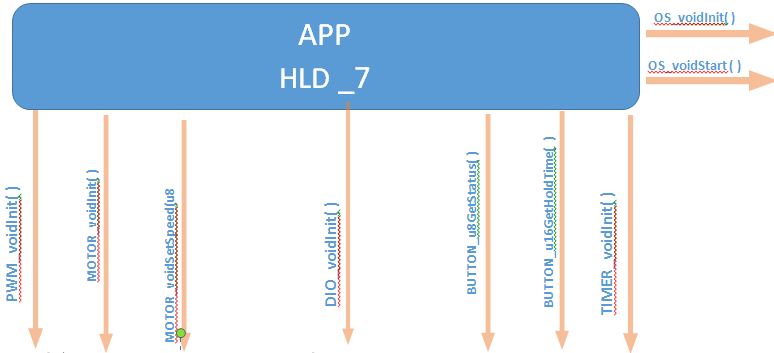
|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Mohammed Elzieny | 19/4/2018 | Creats APP CDD first release | 1.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# *1. Introduction*

## Objective

Application Layer is responsible for handling project functionality and initial all required modules.

## Context Diagram

**

# *External Interface*

## <DIO.h>

### Types: NONE

### Interface

|  |  |
| --- | --- |
| Function | Description |
| DIO\_voidInit() | Initial DIO Registers |

### Constants: NONE

### Symbol: NONE

## <TIMER.h>

### Types: NONE

### Interface

|  |  |
| --- | --- |
| Function | Description |
| TIMER\_voidInit() | Initial TIMER Registers |

### Constants: NONE

### Symbol: NONE

## <PWM.h>

### Types: NONE

### Interface

|  |  |
| --- | --- |
| Function | Description |
| PWM\_voidInit() | Initial PWM Registers |

### Constants: NONE

### Symbol: NONE

## <ButtonH.h>

### Types: NONE

### Interface

|  |  |
| --- | --- |
| Function | Description |
| ButtonH\_voidInit() | Initial ButtonH module |
| BUTTON\_u8GetStatus( ) | Check if switch is currently pressed or not |
| BUTTON\_u16GetHoldTime( ) | Get the period of time that user was holding button |

### Constants: NONE

### Symbol: NONE

## < MotorH.h >

### Types: NONE

### Interface

|  |  |
| --- | --- |
| Function | Description |
| MotorH \_voidInit() | Initial MotorH module |
| MOTOR\_voidSetSpeed(u8 u8SpeedCpy ) | Set motor speed |

### Constants: NONE

### Symbol: NONE

# *Static Design*

## Files

### Used Files

|  |  |
| --- | --- |
| FILE NAME | Description |
| Main.c | Entry function of program |
| Tasks.c | Contains tasks to be added in OS |
| Tasks.h | Contains Tasks Function prototypes |

### Files Inclusion

## Types

NONE

## Symbol Define

NONE

## Constants

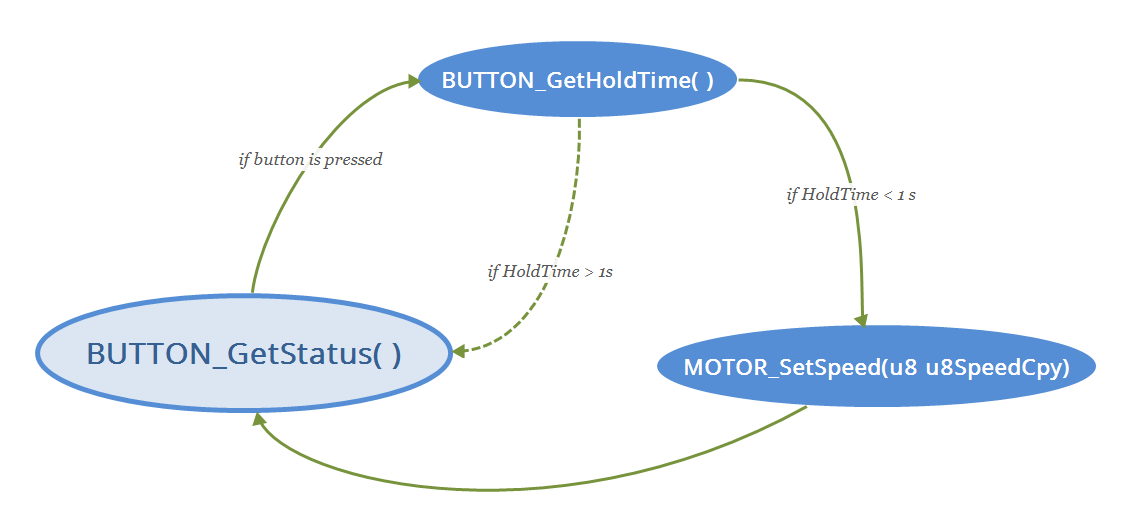
NONE

## Interface (Services)

|  |  |  |
| --- | --- | --- |
| Req ID | APP\_00 | |
| Covers | HLD\_7 | |
| Name/protoTypes | Void TASK\_voidSwitchSpeed(void) | |
| Service ID | 0x00 | |
| Re-entrant / Non re- entrant | Non re- entrant | |
| Synchronous/Asynchronous | Synchronous | |
| Return Value | | NONE |
| Input parameter | | NONE |
| Output parameter | | NONE |
| Input /Output Parameter | | NONE |

# *Dynamic Design*

## Mode Management



## Sequence Diagram

# *Shared Resources*

## Analysis: NONE

## Protection: NONE

# *Configuration Parameters*

## Pre-compile time: NONE

## Link time: NONE

## Post-build: NONE

# Configuration Constrains NONE

# *Integration Constrains*

Initiating modules should be in order as following:  
1-DIO  
2-PWM  
3-MOTOR

4-BUTTON  
5-TIMER  
6-OS