Component Design Document

For

OPERATING SYSTEM

KENEVO ELECTRIC BLENDER

**Table of Contents**

Revision History 1

1. Introduction 3

1.1 Objective 3

1.2 Context Diagram 3

2. External Interface 3

2.1 std\_types.h 3

3. Static Design 4

3.1 Files 4

3.2 Types 5

3.3 Const 5

3.4 Interface (Services) 6

4. Dynamic Design 7

4.1 Sequence Diagram 7

5. Shared Resources 7

5.1 Analysis 7

5.2 Protection 7

6. Configuration Parameters 8

6.1 Pre-compile time 8

6.2 Link time 8

6.3 Post-build 8

7. Configuration Constrains 8

8. Integration Constrains 8

9. History 8

Revision History

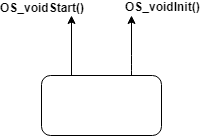
|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Khaled hossam | 18th April 2018 | Create the CDD for Operating System | 1.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 1. Introduction

## Objective

This document describes in detail the **Operating System** used to control this project and all the APIs and services provided to help software development.

## Context Diagram



# External Interface

## Std\_types.h

### Types

|  |  |
| --- | --- |
| Type | Description |
| u8 | Unsigned char |
| u16 | unsigned short int |
| u32 | unsigned long |
| s8 | signed char |
| s16 | signed int |
| s32 | signed long |
| f32 | float |
| f64 | double |

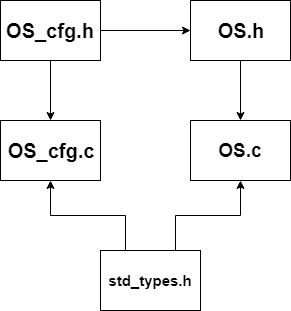
# Static Design

## Files

### Used Files

|  |  |
| --- | --- |
| Filename | Description |
| OS.c | Include APIs implementation and timer ISR |
| OS.h | Include APIs declarations |
| OS\_cfg.h | Include task and configuration type declaration |
| OS\_cfg.c | Include tasks configuration |

### Files Inclusion



## Types

|  |  |
| --- | --- |
| Req ID | OS\_001 |
| Covers | HLD\_015 |
| Name | os\_task{} |
| Type | struct |
| Range | -------- |
| Description | This type define the task configuration paramteres ( Task function, Periodicity) |

## Const

|  |  |
| --- | --- |
| Req ID | OS\_002 |
| Covers | HLD\_015 |
| Name | os\_Tasks[] |
| Type | Const array of struct |
| Range | -------- |
| Description | This is an array of all the tasks in the system with their configuration parameters |

|  |  |
| --- | --- |
| Req ID | OS\_003 |
| Covers | Derived requirement |
| Name | tasks\_max |
| Type | Const u16 |
| Range | -------- |
| Description | Holds the number of tasks in the system |

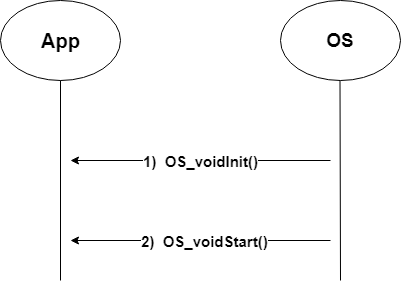
## Interface (Services)

|  |  |
| --- | --- |
| Req ID | OS\_004 |
| Covers | HLD\_015 |
| Name | OS\_voidInit() |
| Service ID | 0x00 |
| Re-entrant / Non-Reentrant | Re-entrant |
| Synchronus / Asynchronus | Synchronus |
| Input parameters | void |
| Ouput parameters | void |
| Input / Output parameters | none |
| Description | Function used to initialize the system operating system |

|  |  |
| --- | --- |
| Req ID | OS\_005 |
| Covers | HLD\_015 |
| Name | OS\_voidStart() |
| Service ID | 0x01 |
| Re-entrant / Non-Reentrant | Re-entrant |
| Synchronus / Asynchronus | Synchronus |
| Input parameters | void |
| Ouput parameters | void |
| Input / Output parameters | none |
| Description | Function to start the operating system and run the tasks defined using os scheduler |

# Dynamic Design

## Sequence Diagram



# Shared Resources

## Analysis

|  |  |
| --- | --- |
| Variable name | sysTick |
| Type | u32 |
| Filename | OS.c |
| Description | Variable to track the ticks of the OS for accurate task scheduling |

## Protection

|  |  |
| --- | --- |
| Variable name | sysTick |
| Protection | None |

# Configuration Parameters

## Pre-compile time

No pre-compile time configuration needed.

## Link time

|  |  |
| --- | --- |
| Name | os\_Tasks |
| Type | Const array of struct |
| Parameters | * Pointer to function * Periodicity |

|  |  |
| --- | --- |
| Name | tasks\_max |
| Type | Const u16 |
| Parameters | * Number of tasks |

## Post-build

No post-build time configuration needed.

# Configuration Constrains

No configuration constrains required.

# Integration Constrains

Integration requires the following:

* Motor and PWM and Timer drivers must be initialized.
* TIMER\_voidInit() must be called before OS\_voidInit().