|  |  |
| --- | --- |
| Document Title | Specification of DIO Driver |
| **Document Owner** |  |
| **Document Responsibility** | Mostafa |
| **Document Identification No** |  |
| **Document Version** | 1 |
| **Document Status** |  |
| **Revision** |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Document Change History** | | |
| **Date** | **Version** | **Reviewed by** | **Change Description** |
| 18.04.2017 | 1.0.0 |  | Initial Release |

**Disclaimer**

This specification and the material contained in it is for the purpose of information only. The companies that have contributed to it shall not be liable for any use of the specification.

The material contained in this specification is protected by copyright and other types of Intellectual Property Rights. The commercial exploitation of the material contained in this specification requires a license to such Intellectual Property Rights.

This specification may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only.

For any other purpose, no part of the specification may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

**Table of Contents**

[1 Introduction and functional overview 5](#_Toc511603712)

[1.1 Objective 5](#_Toc511603713)

[1.2 Context Diagram 5](#_Toc511603714)

[1.3 Acronyms and abbreviations 5](#_Toc511603715)

[2 External interfaces 6](#_Toc511603716)

[2.1 Std\_Types.h 6](#_Toc511603717)

[2.1.1 Types 6](#_Toc511603718)

[2.1.2 Interfaces 6](#_Toc511603719)

[2.1.3 Constants 6](#_Toc511603720)

[2.1.4 Variables 6](#_Toc511603721)

[3 Static Design 7](#_Toc511603722)

[3.1 File structure 7](#_Toc511603723)

[3.1.1 Used Files 7](#_Toc511603724)

[3.1.2 File inclusion 7](#_Toc511603725)

[3.2 Types 7](#_Toc511603726)

[3.2.1 Imported types 7](#_Toc511603727)

[3.2.2 Types Definitions 7](#_Toc511603728)

[3.2.2.1 7](#_Toc511603729)

[3.3 Symbol Definition 8](#_Toc511603730)

[3.4 Function definitions 8](#_Toc511603731)

[3.4.1 8](#_Toc511603732)

[3.5 Call-back notifications 9](#_Toc511603733)

[3.6 Scheduled functions 9](#_Toc511603734)

[4 Dynamic Design 9](#_Toc511603735)

[4.1 Mode Management 9](#_Toc511603736)

[4.2 Sequence Diagram 9](#_Toc511603737)

[5 Shared Resources 9](#_Toc511603738)

[6 Configuration specification 10](#_Toc511603739)

[6.1 Containers and configuration parameters 10](#_Toc511603740)

[6.1.1 Variants 10](#_Toc511603741)

[6.1.2 Parameters 10](#_Toc511603742)

[6.1.2.1 10](#_Toc511603743)

[7 Configuration Constraints 11](#_Toc511603744)

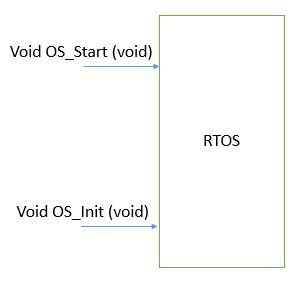
[8 Integration Constraints 11](#_Toc511603745)

# Introduction and functional overview

## Objective

RTOS objective is to design embedded systems that can provide a response to real world events. Events occurring in the real world can have deadlines before which the embedded system must respond and the RTOS scheduling policy must ensure these deadlines are met.

## Context Diagram



## Acronyms and abbreviations

Acronyms and abbreviations that have a local scope.

|  |  |
| --- | --- |
| ***Abbreviation / Acronym:*** | ***Description:*** |
| RTOS | Real Time Operating System |
| OS | Operating System |

# External interfaces

## Std\_Types.h

### Types

|  |  |
| --- | --- |
| Name | Description |
| U8 | Unsigned char. |
| U16 | Unsigned short Int. |
| U32 | Unsigned long Int. |

### Interfaces

None

### Constants

None

### Variables

None

## Timer0.h

### Types

None

### Interfaces

Void Timer0\_SetCallBack (OS\_ISR): Call this function to implement in the OS Tick using Timer0 ISR.

### Constants

None

### Variables

None

## App.h

### Types

None

### Interfaces

Void TaskOne (void): Use this function to be called in the OS scheduler with a certain periodicity.

### Constants

None

### Variables

None

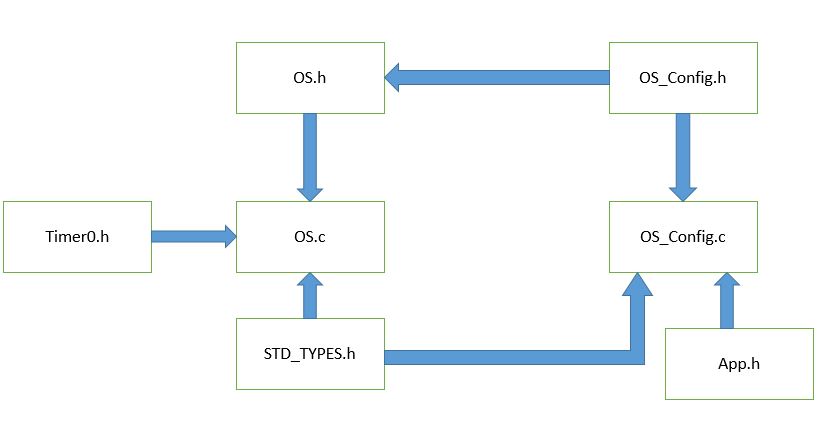
# Static Design

## File structure

### Used Files

|  |  |
| --- | --- |
| ***File*** | ***Description*** |
| OS.h | This file includes all prototype of functions, #defines, extern constant, types which will be accessible to the main. |
| OS.c | This file includes all function implementation and their variables. |
| OS\_Config.h | This file includes the tasks’ struct declaration which includes the tasks prototype and their periodicity plus constant declaration of an instant of an array of that struct and number of tasks. |
| OS\_Config.c | This file has the constant struct array filled with the application tasks. |

### File inclusion



## Types

### Types Definitions

#### OS\_Config\_T

|  |  |
| --- | --- |
| ***Name:*** | OS\_Config\_T |
| ***Type:*** | Struct |
| ***Range:*** | 1 Task |
| ***Description:*** | This is the configuration array of tasks which the scheduler check through it. |

## Symbol Definition

None

## Function definitions

This is a list of functions provided for main.

|  |  |  |
| --- | --- | --- |
| ***Service name:*** | OS\_Init | |
| ***Syntax:*** | Void OS\_Init (void) | |
| ***Service ID[hex]:*** | 0x00 | |
| ***Sync/Async:*** | Sync | |
| ***Reentrancy:*** | Reentrant | |
| ***Parameters (in):*** | None |  |
| ***Parameters (in/out):*** | None | |
| ***Parameters (out):*** | None | |
| ***Return value:*** | None |  |
| ***Description:*** | This function start the OS used to initialized the OS by setting the OS ISR in the Timer0 ISR. | |

### 

|  |  |  |
| --- | --- | --- |
| ***Service name:*** | OS\_Start | |
| ***Syntax:*** | Void OS\_Start (void) | |
| ***Service ID[hex]:*** | 0x01 | |
| ***Sync/Async:*** | Sync | |
| ***Reentrancy:*** | Reentrant | |
| ***Parameters (in):*** | None |  |
| ***Parameters (in/out):*** | None | |
| ***Parameters (out):*** | None | |
| ***Return value:*** | None |  |
| ***Description:*** | This function start the OS using an infinite loop of tasks each is executed depending on the periodicity of each task. | |

## Call-back notifications

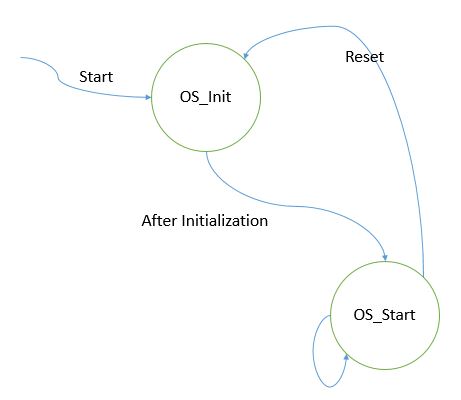
The RTOS module does not provide any callback notifications.

## Scheduled functions

The RTOS module has no scheduled functions.

# Dynamic Design

## Mode Management



## Sequence Diagram

None

# Shared Resources

1. U32 TickCounter

It can be changed while reading task if it was interrupted twice in ISR so we have to make critical section to avoid that when accessing this variable.

# Configuration specification

This chapter defines configuration parameters and their clustering into containers.

## Containers and configuration parameters

The following chapters summarize all configuration parameters.

### Parameters

#### 

|  |  |  |  |
| --- | --- | --- | --- |
| ***Name*** | OS\_Config | | |
| ***Description*** | Array of tasks which the scheduler will check through them. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | Array of Struct | | |
| ***Default value*** | None | | |
| ***Configuration Class***  ***Scope / Dependency*** | ***Link time*** | X | All Array Elements |
| ***Post-build time*** | -- |  |
| Scope: local | -- |  |

# Configuration Constraints

None

# Integration Constraints

1. OS\_Init() called at first in main before OS\_Start().
2. Timer0 is enabled and was ready for OS\_Init() to override the Timer0 ISR.
3. The OS should have at least the application task to work in this project.
4. The sum of Tasks execution time does not exceed the smallest periodicity.