Component Design Document

For

Timer Module

**Table of Contents**

Revision History 1

*1. Introduction* 2

1.1 Objective 2

1.2 Context Diagram 2

2. *External Interface* 2

2.1 <file name.h> 2

3. *Static Design* 2

3.1 Files 2

3.2 Types 3

3.3 Symbol Define <#define> 3

3.4 Const 3

3.5 Interface (Services) 3

*4.* *Dynamic Design* 4

4.1 Mode Management 4

4.2 Sequence Diagram 4

*5.* *Shared Resources* 4

5.1 Analysis 4

5.2 Protection 4

6. *Configuration Parameters* 4

6.1 Pre-compile time 4

6.2 Link time 5

6.3 Post-build 5

*7.* *Configuration Constrains* 5

*8.* *Integration Constrains* 5

*9.* *History* 5

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Abdulla fathy Mohamed | 16th April 2018 | Create the file | 1.0 |
| Youssef Medhat | 20th April 2018 | Review on the file | 1.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# *1. Introduction*

## Objective

It’s used for calculating time to drive the operating system (OS) module.

## Context Diagram

## 

## 

# *External Interface*

## File name : STD\_TYPES.h

### Types

|  |  |
| --- | --- |
| U8 | unsigned char |
| U16 | unsigned short int |
| U32 | unsigned long int |

### Interface

|  |  |
| --- | --- |
|  |  |
|  |  |

### Const

### Symbol

# *Static Design*

## Files

### Used Files

|  |  |
| --- | --- |
| File name | Description |
| Timer\_register.h | Contains all the registers of the timer driver |
| Timer\_priv.h | Contains all the private Symbols of the timer driver |
| Timer\_config.h | Contains all the configurable symbols for the timer driver such as timer mode. |
| Timer\_int.h | Contains all the public interfaces and symbols to be available for anyone who uses the driver |
| Timer.c | Contains the implementation of all the interfaces of the timer interface. |

### Files Inclusion

## 

## 3.2 Types

## Symbol

|  |  |
| --- | --- |
| Symbol name | Description |
| Timer0\_NoDIV | To make the timer work without prescaler |
| Timer0\_DIV\_8 | To make the timer work with prescaler of 8 to divide the frequency by 8 |
| Timer0\_DIV\_64 | To make the timer work with prescaler of 64 to divide the frequency by 64 |
| Timer0\_DIV\_256 | To make the timer work with prescaler of 256 to divide the frequency by 256 |
| Timer0\_DIV\_1024 | To make the timer work with prescaler of 1024 to divide the frequency by 1024 |
| Timer\_Res | To indicate the size of the timer register |

## Const

## Interface (Services)

|  |  |  |  |
| --- | --- | --- | --- |
| Req ID | TIM\_001 | | |
| Covers |  | | |
| Name/prototype | void Timer\_voidInitialize(void) | | |
| Service ID |  | | |
| Re-entrant / Non re- entrant |  | | |
| Synchronous/Asynchronous |  | | |
| Return Value | | void | No return value |
| Input parameter | | void | No input parameters |
| Output parameter | | void | No output parameters |
| Input /Output Parameter | | void | No input//Output parameters |

|  |  |  |  |
| --- | --- | --- | --- |
| Req ID | TIM\_002 | | |
| Covers |  | | |
| Name/prototype | void Timer\_voidEnableInt(void); | | |
| Service ID |  | | |
| Re-entrant / Non re- entrant |  | | |
| Synchronous/Asynchronous |  | | |
| Return Value | | void | No return value |
| Input parameter | | void | No input parameters |
| Output parameter | | void | No output parameters |
| Input /Output Parameter | | void | No input//Output parameters |

|  |  |  |  |
| --- | --- | --- | --- |
| Req ID | TIM\_003 | | |
| Covers |  | | |
| Name/prototype | void Timer\_voidDisableInt(void); | | |
| Service ID |  | | |
| Re-entrant / Non re- entrant |  | | |
| Synchronous/Asynchronous |  | | |
| Return Value | | void | No return value |
| Input parameter | | void | No input parameters |
| Output parameter | | void | No output parameters |
| Input /Output Parameter | | void | No input//Output parameters |

# *Dynamic Design*

## Mode Management

## Sequence Diagram

# *Shared Resources*

## Analysis

## Protection

# *Configuration Parameters*

## Pre-compile time

|  |  |
| --- | --- |
| Symbol name | Description |
| Timer\_u16Prescaler | To change the prescaler that the timer will work on |
| Timer\_CMP\_Value | To change the compare value that the timer will use in the CTC mode |

## Link time

## Post-build

# *Configuration Constrains*

We can use this module only to configure one timer (timer0).

# *Integration Constrains*

Init() function must be called before using any of the module’s interfaces.

# *History*