Application

|  |  |
| --- | --- |
| **Document Name** |  |
| **Version** | 2 |
| **Author** | Ahmed Adel |
| **Status** | Completed |
| **Date** | 4/8/2016 |

Document History

| Version | Date | Author | Changes |
| --- | --- | --- | --- |
| 1.00 | 01/04/2016 | Ahmed Adel | Initial |

Table of content

[1 Introduction 4](#_Toc447289905)

[1.1 Goals and Objectives 4](#_Toc447289906)

[1.2 Component APIs and variables 4](#_Toc447289907)

[1.3 File Description 7](#_Toc447289912)

# Introduction

This document is created to demonstrate the Application and its used APIs also to demonstrate the dependant drivers for this module.

## Goals and Objectives

To provide demonstration of the used APIs and components to make the system work in 3 modes Timer, Stop watch and Time editor.

## Component APIs and variables

|  |  |
| --- | --- |
| **Interface** | **Description** |
| *LCD driver* | *The driver is to write characters on the LCD and interface with the DIO driver.* |
| *DIO Driver* | *It is required to interface with microcontroller through DIO peripheral.* |
| *Timer Driver* | *It is used for measuring time at different modes* |
| *Tactile Switch Driver* | *It is used to measure if switch is pressed or released* |

Table 1: Interfaces description

|  |  |  |  |
| --- | --- | --- | --- |
| ***API*** | ***Description*** | ***Input arguments*** | ***Return Value*** |
| **APP\_voidInit** | This function is used to initialize the Timer, DIO and LCD. | None | None |
| **APP\_u8Timer** | This function is used to update and display time; it will keep working till mode switch is pressed .at this moment it will switch to APP\_STOP\_WATCH\_STANDBY state. | None | Next state to be executed in the state machine |
| **APP\_u8APP\_STOP\_WATCH\_STANDBY** | This function used to write 0 to lcd ,update system time, wait for mode switch or start stop watch switch to be pressed, if mode switch is pressed it will change next state to APP\_EDIT\_TIME state but if start stop watch is pressed it will change next state to APP\_STOP\_WATCH | None | Next state to be executed in the state machine |
| **APP\_u8APP\_STOP\_WATCH** | This function used to write stop watch time to lcd ,update system time, wait for mode switch or start stop watch switch to be pressed, if mode switch is pressed it will change next state to APP\_EDIT\_TIME state but if start stop watch is pressed it will change next state to APP\_STOP\_WATCH\_PAUSE | None | Next state to be executed in the state machine |
| **APP\_u8APP\_STOP\_WATCH\_PAUSE** | This function used to write to lcd last stop watch time before it was paused, update system time, wait for mode switch or start stop watch switch to be pressed, if mode switch is pressed it will change next state to APP\_EDIT\_TIME state and reset stop watch timer but if start stop watch is pressed it will change next state to APP\_RESUME\_STAND\_BY and start a temp timer. | None | Next state to be executed in the state machine |
| **APP\_u8APP\_RESUME\_STAND\_BY** | This function used to write last stop watch time to lcd ,update system time, wait for start stop watch switch to be released. if it is released before maximum time it will stop temp timer then it will switch next state directly to APP\_STOP\_WATCH but if switch was pressed for more than maximum time it will stop temp timer then it will rest stop watch timer, after that it will switch next state to APP\_STOP\_WATCH | None | Next state to be executed in the state machine |
| **APP\_u8APP\_EDIT\_TIME** | This function used to take snapshot of time, write it to lcd ,allow the user to edit it .then finally switch system to APP\_TIMER | None | Next state to be executed in the state machine |
| **APP\_voidUpdateTimers** | Update number of MS passed since last time this function was called, If 1000 MS have passed since last timer update, it add a second to the timer.it updates system timer, stop watch timer if it was enabled and temp timer if it was enabled | None | None |
| **APP\_voidDisplay** | Display input time at lcd | Array of u8 representing hours as first array element, minutes as second array element, seconds as third array element, AM or PM as fourth array element | None |
| **APP\_u8ReadSwitch** | Check if switch is pressed or released at (single press mode) | Switch ID | State of Switch |
| **APP\_voidConvertto12HoursSystem** | Convert 24 hours system clock to 12 hours system clock | Time in seconds as the first argument and a pointer for an array to store conversion result as the second argument | None |
| **APP\_u8TimeUpdate** | If 1000 MS have passed since last system timer update, it add a second to the system timer | Number of MS passed since last system timer update | None |
| **APP\_u8CheckFlag** | update number of seconds passed since this function was called | None | number of seconds passed since this function was called |
| **APP\_u8StopWatchUpdate** | If 1000 MS have passed since last stop watch timer update, it add a second to the stop watch timer | Number of MS passed since last stop watch timer update | None |
| **APP\_voidDisplayFlasher** | Display time at lcd | Pointer to array storing the time | None |
| **APP\_voidChangeTime** | Increment or decrement time item (hours, minutes, seconds, AM or PM) | Pointer to array storing the time as the first argument, index of the item that need to be changed as the second argument ,state (if this item need to be incremented or decremented) as the third argument | None |
| **APP\_u32Convertto24hourssystem** | Convert from12 hours system to 24hours system | Pointer to array storing the time | U32 variable representing time in 24 hours system (in seconds) |

Table 2: API description

**Dependant Modules API’s:**

* DIO driver:

|  |  |  |  |
| --- | --- | --- | --- |
| **API** | Description | Inputs | Outputs |
| **DIO\_VoidInit** | API which is used to Initialize DIO pins | None | None |

* Timer driver:

|  |  |  |  |
| --- | --- | --- | --- |
| **API** | Description | Inputs | Outputs |
| **TIMER\_voidInit(void (\*Copy\_pfcallback)(void))** | API which is used to Initialize Timer in CTC mode without prescaler ,enable timer interrupt and set pointer of call back function | Pointer to call back function | None |

* Lcd driver:

|  |  |  |  |
| --- | --- | --- | --- |
| **API** | Description | Inputs | Outputs |
| **LCD\_voidInit(void )** | API which is used to Initialize Timer LCD | None | None |
| LCD\_voidWriteCommand(u8) | API used to control lcd through some orders | U8(order number) | None |
| LCD\_voidWriteChar(u8) | Write char to current position on lcd | U8(character to be written on lcd) | None |

|  |  |  |
| --- | --- | --- |
| **Variable** | **Type** | **Description** |
| App\_u8StopWatchFlag | u8 | Flag representing if stop watch is working or not |
| App\_u8TempTimerFlag | u8 | Flag representing if temp timer is working or not |
| APP\_u81MilliCounter | u8 | This variable is incremented by timer interrupt every 1 MS |
| APP\_u32Timer | U32 | Variable containing number of seconds passed since system clock started |
| APP\_u32StopWatchTimer | U32 | Variable containing number of seconds passed since stop watch clock started |
| APP\_u32TempTimer | U32 | Variable containing number of seconds passed since temp clock started |
| APP\_u161MilliSecondCounterStopWatch | U16 | Variable containing number of Milliseconds passed since last stop watch clock update |
| APP\_u161MilliSecondTempCounter | U16 | Variable containing number of Milliseconds passed since last temp clock update |
| APP\_u8SwitchState | Array containing 3 elements of type u8 | Array containing states of the 3 switches in system |

Table 3: Variables of the component

## File Description

|  |  |
| --- | --- |
| **File Name** | **Description** |
| APP\_interface.h | It contains the prototypes of the APIs. |
| APP\_private.h | It contains private configuration parameters. |
| APP\_prog.c | It contains the written code of the APIs. |
| APP.c | It contains main function |

Table 5: Files Description

{END\_VAL\_SPECIFICATION\_DOC}