

**Digital Watch Project PO2\_DGW**

**(SRS)**

Table 1 Status Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document Name** | **Author** | **Version** | **Update Date** | **Status** |
| SRS | Yosuf Ahmad/ Abdollah Mahmoud | 1.0 | 16/02/2023 | Proposed |

**DOCUMENT HISTORY**

Table 2 Document History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Description of Change** | **Author** | **Date** | **Document Status** |
| 1 | Initial creation of SRS Document | Yosuf Ahmad/ Abdollah Mahmoud/ Samira Saeed | 16/02/2023 | Draft |

Table 3 Reference Table

|  |  |  |
| --- | --- | --- |
| Reference Input Documents | Version | Status |
| CYRS | 1.1 | Released |
| HSI | 1.1 | Released |

Table of Contents

[1 INTRODUCTION 4](#_Toc33363453)

[**1.1** **Purpose** 4](#_Toc33363454)

[**1.2** **Glossary** 4](#_Toc33363455)

[**1.2.1** **clock** 4](#_Toc33363456)

[**1.2.2** **alarm** 4](#_Toc33363457)

[**1.2.3** **ring the bell** 4](#_Toc33363458)

[**1.2.4** **clock time** 4](#_Toc33363459)

[**1.3** **General description** 5](#_Toc33363460)

[2 Context Diagram 5](#_Toc33363461)

[3 FUNCTIONAL REQUIREMENTS 6](#_Toc33363462)

[**3.1** **Time-Display Mode Requirements** 6](#_Toc33363463)

[**3.2** **Alarm mode requirements** 6](#_Toc33363464)

[**3.3** **Stop-Watch mode requirements** 7](#_Toc33363465)

[**3.4** **System components (buttons) requirements** 8](#_Toc33363466)

List of Tables

[Table 1 Status Table 2](#_Toc33363415)

[Table 2 Document History 2](#_Toc33363416)

[Table 3 Reference Table 13](#_Toc33363417)

List of Figures

[Figure 1 SW Context Block Diagram 5](file:///F:\Abdelrahman\projectt\Software%20Specification\SRS\SRS.docx#_Toc33363449)

# INTRODUCTION

## **Purpose**

This is version 1 of the requirements specification for a simple Digital watch with alarm with settings and Stop Watch.

## **Glossary**

Definitions, acronyms and abbreviations

Digital watch consists of 3 main Functions:

1-Clock.

2.Alarm.

3.Stop Watch.

### **clock**

a device that has its own value of time in hours, minutes and seconds, which it displays and which it maintains accurately relative to when the time was last reset.

In this document, a "conventional watch" means such a clockwork or electric device in the real world; "watch" means a computer operating system function that returns

at least an absolute or elapsed time as measured by the operating system.

### **alarm**

a device which is a conventional clock with an additional time value setting and an attached alarm bell. When the clock time reaches the time of the "alarm setting" it rings the bell.

In this specification, "alarm clock" means the software application that imitates many of the functions of a conventional clock with an alarm.

### **ring the bell**

the continuous alert sound (the term "audible signal" is too pedantic!) that an alarm clock can make. It may be a buzzer or chime or other tone

(compare mobile phone "ring tones" which no longer resemble a bell).

### **clock time**

the time relative to the last reset of the clock.

## **General description**

The Digital Watch System that has A Clock with a simple alarm clock in an on-screen window and Stop watch. The user can select an option of the three options

(View Clock, setting stop watch, Setting alarm).

The clock provides an alarm and allows the user to set alarms and provide a Stop watch.

# Software Context

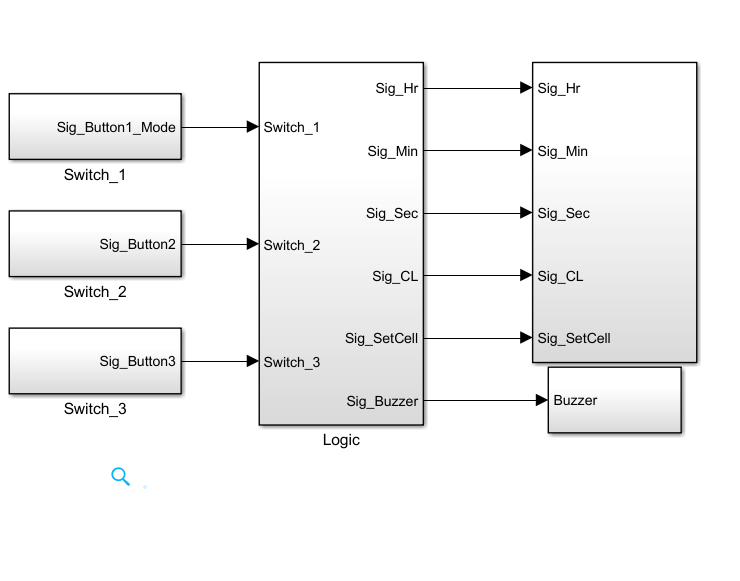


Figure 1 SW Context Block Diagram

# FUNCTIONAL REQUIREMENTS

## **Time-Display Mode Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_01\_V01.0 | **Covers** | Req\_PO3\_DGW\_CYRS\_01\_V01.0 |
| **Author** | Marina Medhat | **Date** | 15/2/2023 |
| **Description:** | The SW shall increment SecondsVar Variable with delay 1 sec.  if (SecondsVar == 60) { SecondsVar=0 and MinutesVar++ }  if (MinutesVar== 60) { MinutesVar=0 and HoursVar++ } | | |
| **inputs** | Sig\_Hr, Sig\_Min, Sig\_Sec | **Outputs** | Sig\_Hr, Sig\_Min, Sig\_Sec |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_02\_V01.0 | **Covers** | Req\_PO3\_DGW\_CYRS\_01\_V01.0 |
| **Author** | Marina Medhat | **Date** | 15/2/2023 |
| **Description:** | The SW shall reset HoursVar to zero if it reaches 12 and the CycleLegnth flag will toggle.  If (CycleLength flag == 0) {Am will be displayed}  if (CycleLength flag == 1) {Pm will be displayed} | | |
| **inputs** | Sig\_Hr, Sig\_CL | **Outputs** | Sig\_Hr, Sig\_CL |

## **Alarm mode requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_03\_V01.0 | **Covers** | Req\_PO3\_DGW\_CYRS\_02\_V01.1 |
| **Author** | Marina Medhat | **Date** | 15/2/2023 |
| **Description:** | The SW shall keep comparing the alarm variables with the time variable  if ((AlHoursVar == HoursVar )&( AlMinutesVar == MinutesVar )&( AlCycleLength == CycleLength)) {the Buzzer\_State = 1} | | |
| **inputs** | Sig\_Hr, Sig\_Min, Sig\_Sec, Sig\_CL | **Outputs** | Sig\_Buzzer |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_04\_V01.0 | **Covers** | Req\_PO3\_DGW\_CYRS\_02\_V01.1 |
| **Author** | Marina Medhat | **Date** | 15/2/2023 |
| **Description:** | If (Buzzer\_state == 1) {the buzzer is enabled}  else {it will be disabled} | | |
| **inputs** | Sig\_Buzzer | **outputs** | - |

## **Stop-Watch mode requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_05\_V01.0 | **Covers** | Req\_PO3\_DGW\_CYRS\_03\_V01.0 |
| **Author** | Marina Medhat | **Date** | 15/2/2023 |
| **Description:** | The SW shall start (SwHoursVar, SwMinutesVar, SwSecondsVar) with zeros.  If (the user start Stopwatch) {The SwSecondsVar Variable starts to increment with delay 1 sec}  if (SwSecondsVar == 60) { SwSecondsVar = 0 and SwMinutesVar++}  if (SwMinutesVar == 60) { SwMinutesVar = 0 and SwHoursVar++ } | | |
| **inputs** | Sig\_Hr, Sig\_Min, Sig\_Sec | **Outputs** | Sig\_Hr, Sig\_Min, Sig\_Sec |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_06\_V01.0 | **Covers** | Req\_PO3\_DGW\_CYRS\_04\_V01.1 |
| **Author** | Marina Medhat | **Date** | 15/2/2023 |
| **Description:** | If(the user press stop button) {The SW shall save the last values of the Stopwatch and displays it on the LCD without incrementing of any variable} | | |
| **inputs** | Sig\_Hr, Sig\_Min, Sig\_Sec | **outputs** | Sig\_Hr, Sig\_Min, Sig\_Sec |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_07\_V01.0 | **Covers** | Req\_PO3\_DGW\_CYRS\_04\_V01.1 |
| **Author** | Marina Medhat | **Date** | 15/2/2023 |
| **Description:** | The SW shall save the last values of the Stopwatch and Continue increment it when the user press to Play the Stopwatch again. | | |
| **inputs** | Sig\_Hr, Sig\_Min, Sig\_Sec | **outputs** | Sig\_Hr, Sig\_Min, Sig\_Sec |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_08\_V01.0 | **Covers** | Req\_PO3\_DGW\_CYRS\_05\_V01.1 |
| **Author** | Marina Medhat | **Date** | 15/2/2023 |
| **Description:** | If (the user press to reset the Stopwatch)  { SwHoursVar = 0, SwMinutesVar = 0, SwSecondsVar = 0 and display them on the LCD} | | |
| **inputs** | Sig\_Hr, Sig\_Min, Sig\_Sec | **outputs** | Sig\_Hr, Sig\_Min, Sig\_Sec |

## **System components (buttons) requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_09\_V01.0 | **Covers** | Req\_PO3\_DGW\_CYRS\_07\_V01.0 |
| **Author** | Marina Medhat | **Date** | 15/2/2023 |
| **Description:** | The SW shall start Current\_Mode with zero  If(Sig\_Button1 == 1) {Current\_Mode++}  if(Current\_Mode == 3) { Current\_Mode=0} | | |
| **inputs** | Sig\_Button1 | **outputs** | Sig\_Button1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_10\_V01.0 | **Covers** | Req\_PO3\_DGW\_CYRS\_08\_V01 |
| **Author** | Marina Medhat | **Date** | 15/2/2023 |
| **Description:** | The SW shall start SetCell with zero in Time display mode.  If(Sig\_Button2 == 1) { SetCell ++}  if(SetCell == 3) { SetCell =0} | | |
| **inputs** | Sig\_SetCell, Sig\_Button2 | **outputs** | Sig\_SetCell |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_11\_V01.0 | **Covers** | Req\_PO3\_DGW\_CYRS\_08\_V01 |
| **Author** | Marina Medhat | **Date** | 15/2/2023 |
| **Description:** | The SW shall compare the SetCell variable and If equal 0, the position on LCD will be in hours’ cell | | |
| **inputs** | Sig\_SetCell | **outputs** | Sig\_Hr, Sig\_Min, Sig\_Sec, Sig\_CL |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_12\_V01.0 | **Covers** | Req\_PO3\_DGW\_CYRS\_08\_V01 |
| **Author** | Marina Medhat | **Date** | 15/2/2023 |
| **Description:** | The SW shall compare the SetCell variable and If equal 1, the position on LCD will be in minutes’ cell. | | |
| **inputs** | Sig\_SetCell | **outputs** | Sig\_Hr, Sig\_Min, Sig\_Sec, Sig\_CL |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_13\_V01.0 | **Covers** | Req\_PO3\_DGW\_CYRS\_08\_V01 |
| **Author** | Marina Medhat | **Date** | 15/2/2023 |
| **Description:** | The SW shall compare the SetCell variable and If equal 2, the position on LCD will be in seconds cell. | | |
| **inputs** | Sig\_SetCell | **outputs** | Sig\_Hr, Sig\_Min, Sig\_Sec, Sig\_CL |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_14\_V01.0 | **Covers** | Req\_PO3\_DGW\_CYRS\_08\_V01 |
| **Author** | Marina Medhat | **Date** | 15/2/2023 |
| **Description:** | The SW shall check SetCell value in Time display mode.  If(Sig\_Button3 pressed)  If(SetCell == 0) {HoursVar++}  If(SetCell == 1) {MinutesVar++}  If(SetCell == 2) {SecondsVar++} | | |
| **inputs** | Sig\_SetCell, Sig\_Button3 | **outputs** | Sig\_Hr, Sig\_Min, Sig\_Sec, Sig\_CL |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_15\_V01.0 | **Covers** | Req\_PO3\_DGW\_CYRS\_09\_V01 |
| **Author** | Marina Medhat | **Date** | 15/2/2023 |
| **Description:** | The SW shall start AlSetCell with zero in Alarm mode.  If(Sig\_Button2) { AlSetCell ++}  if(AlSetCell == 3) { AlSetCell =0} | | |
| **inputs** | Sig\_SetCell, Sig\_Button2 | **Outputs** | Sig\_SetCell |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_16\_V01.0 | **Covers** | Req\_PO3\_DGW\_CYRS\_09\_V01 |
| **Author** | Marina Medhat | **Date** | 15/2/2023 |
| **Description:** | The SW shall compare the AlSetCell variable and If equal 0, the position on LCD will be in hours’ cell. | | |
| **inputs** | Sig\_SetCell | **Outputs** | Sig\_Hr, Sig\_Min, Sig\_CL |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_17\_V01.0 | **Covers** | Req\_PO3\_DGW\_CYRS\_09\_V01 |
| **Author** | Marina Medhat | **Date** | 15/2/2023 |
| **Description:** | The SW shall compare the AlSetCell variable and If equal 1, the position on LCD will be in minutes’ cell. | | |
| **inputs** | Sig\_SetCell | **outputs** | Sig\_Hr, Sig\_Min, Sig\_CL |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_18\_V01.0 | **Covers** | Req\_PO3\_DGW\_CYRS\_09\_V01 |
| **Author** | Marina Medhat | **Date** | 15/2/2023 |
| **Description:** | The SW shall compare the AlSetCell variable and If equal 2, the position on LCD will be in seconds cell. | | |
| **inputs** | Sig\_SetCell | **outputs** | Sig\_Hr, Sig\_Min, Sig\_CL |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_19\_V01.0 | **Covers** | Req\_PO3\_DGW\_CYRS\_09\_V01 |
| **Author** | Marina Medhat | **Date** | 15/2/2023 |
| **Description:** | The SW shall check AlSetCell value in Alarm mode.  If(Sig\_Button3==1)  If(AlSetCell == 0) {AlHoursVar++}  If(AlSetCell == 1) {AlMinutesVar++}  If(AlSetCell == 2) {AlSecondsVar++} | | |
| **inputs** | Sig\_SetCell, Sig\_Button3 | **Outputs** | Sig\_Hr, Sig\_Min, Sig\_CL |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_20\_V01.0 | **Covers** | Req\_PO3\_DGW\_CYRS\_09\_V01 |
| **Author** | Marina Medhat | **Date** | 15/2/2023 |
| **Description:** | The SW shall turn the Buzzer\_Signal to zero if Current\_Mode button is pressed | | |
| **inputs** | Sig\_Button1 | **outputs** | Sig\_Buzzer |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_21\_V01.0 | **Covers** | Req\_PO3\_DGW\_CYRS\_10\_V01 |
| **Author** | Marina Medhat | **Date** | 1/3/2020 |
| **Description:** | The SW shall check Sig\_Button2  First Press at Button2 -> Start Stopwatch mode -> Sig\_SwStatus = 1  Second Press at Button2 -> Stop Stopwatch -> Sig\_SwStatus = 0 | | |
| **inputs** | Sig\_Button2 | **outputs** | Sig\_SwStatus |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_22\_V01.0 | **Covers** | Req\_PO3\_DGW\_CYRS\_10\_V01 |
| **Author** | Marina Medhat | **Date** | 1/3/2020 |
| **Description:** | If (Sig\_SWStatus == 1) {The SW shall start incrementing its values (SwHoursVar, SwMinutesVar, SwSecondsVar) and displays it on LCD.} in case of Stopwatch mode | | |
| **inputs** | Sig\_SWStatus | **outputs** | Sig\_Hr, Sig\_Min, Sig\_Sec |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_23\_V01.0 | **Covers** | Req\_PO3\_DGW\_CYRS\_10\_V01 |
| **Author** | Marina Medhat | **Date** | 1/3/2020 |
| **Description:** | If (Sig\_Button2 == 1 && Sig\_SWStatus == 1) {The SW shall stop incrementing its values (SwHoursVar, SwMinutesVar, SwSecondsVar) and keep it saved } in case of Stopwatch mode | | |
| **inputs** | Sig\_Button2 ,Sig\_SWStatus | **outputs** | \_\_\_\_\_ |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_24\_V01.0 | **Covers** | Req\_PO3\_DGW\_CYRS\_10\_V01 |
| **Author** | Marina Medhat | **Date** | 1/3/2020 |
| **Description:** | The SW will save SwHoursVar, SwMinutesVar and SwSecondsVar  If (Sig\_Button2 == 1) {resume incrementing} | | |
| **inputs** | Sig\_Button2, Sig\_Button3 | **outputs** | Sig\_SwReset, Sig\_Hr, Sig\_Min, Sig\_Sec |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_25\_V01.0 | **Covers** | Req\_PO3\_DGW\_CYRS\_10\_V01 |
| **Author** | Marina Medhat | **Date** | 1/3/2020 |
| **Description:** | If (Sig\_Button3 == 1) { Sig\_SwReset=1} | | |
| **inputs** | Sig\_Button3 | **outputs** | Sig\_Hr, Sig\_Min, Sig\_Sec |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_26\_V01.0 | **Covers** | Req\_PO3\_DGW\_CYRS\_10\_V01 |
| **Author** | Marina Medhat | **Date** | 1/3/2020 |
| **Description:** | If (Sig\_SwReset == 1) {SwHoursVar=0, SwMinutesVar=0 and SwSecondsVar = 0, Sig\_SwReset=0} | | |
| **inputs** | Sig\_SwReset | **outputs** | Sig\_Hr, Sig\_Min, Sig\_Sec. Sig\_SwReset |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_27\_V01.0 | **Covers** | Req\_PO3\_DGW\_HSI\_01\_V01 |
| **Author** | Marina Medhat | **Date** | 15/2/2023 |
| **Description:** | The SW shall configure the pins according to the mapping of the configuration pins on the microcontoller | | |
| **inputs** | --- | **outputs** | --- |