

**PO3\_DGW\_Digital watch**

**(SRS)**

Table 1 Status Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document Name** | **Author** | **Version** | **Update Date** | **Status** |
| SRS | Mostafa Nader | 1.5 | 20/02/2020 | Proposed |

**DOCUMENT HISTORY**

Table Document History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Description of Change** | **Author** | **Date** | **Document Status** |
| 1 | Initial creation of SRS Document | Marina Medhat  Ahmed Qandeel | 23/01/2020 | Draft |
| 1.1 | Added requirements regarding the SIQ document answers | Marina Medhat  Ahmed Qandeel | 28/01/2020 | Draft |
| 1.2 | Added requirements regarding the SRS Review. | Marina Medhat | 7/02/2020 | Proposed |
| 1.3 | Added head titles for the requirements and updated the table of content after reviewing the SRS document | Mostafa Nader | 7/02/2020 | Proposed |
| 1.4 | Added Context diagram including signals and covered all the inputs/outputs in the requirements with these signals | Mostafa Nader | 20/02/2020 | Proposed |
| 1.5 | Re-Deigned the context diagram and made all the requirements more readable | Mostafa Nader | 23/02/2020 | Proposed |

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.

# Context Diagram

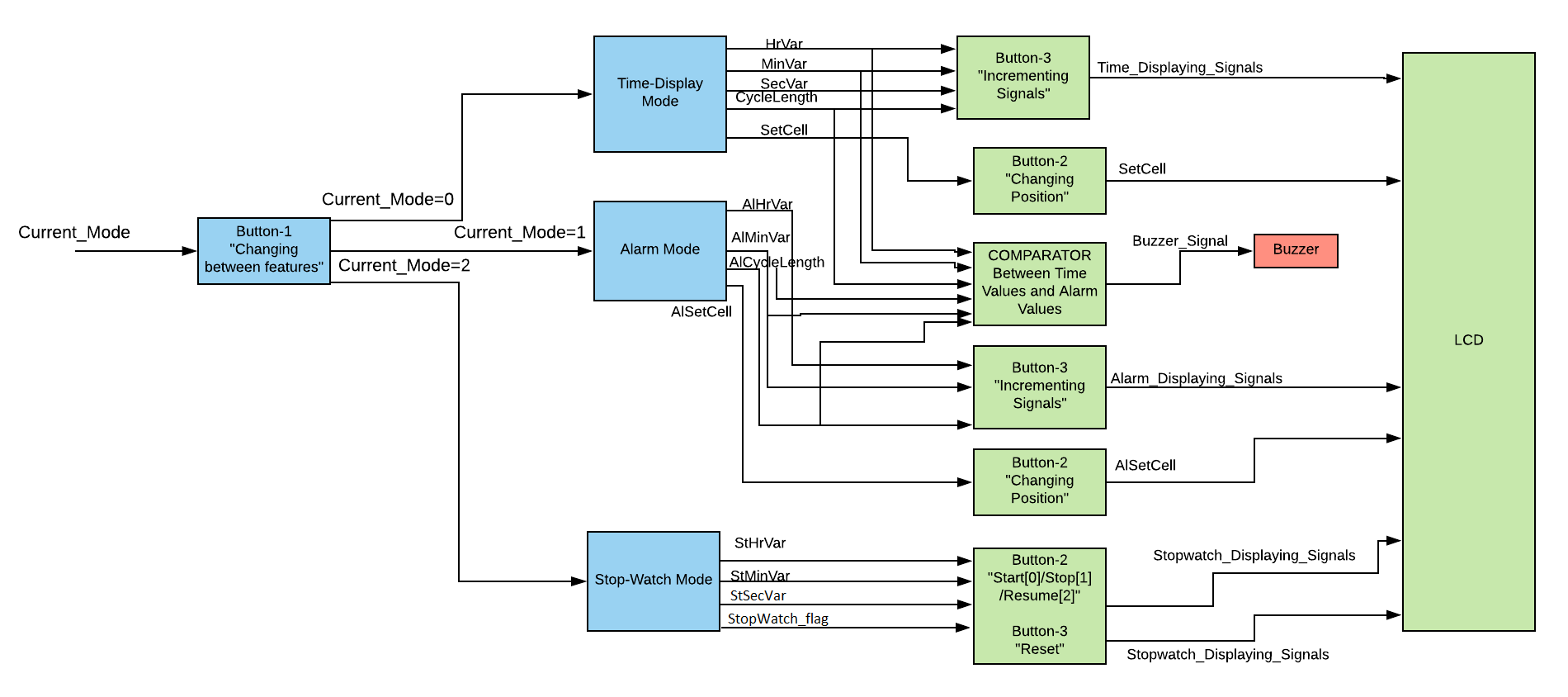


Figure SW Context Block Diagram

# FUNCTIONAL REQUIREMENTS

## **Time-Display Mode Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_01\_V01.2 | **Covers** | Req\_PO3\_DGW\_CYRS\_01\_V01.2 |
| **Author** | Marina Medhat | **Date** | 12/2/2020 |
| **Description:** | The SW shall increment SecVar Variable with delay 1 sec.  if (SecVar == 60) { SecVar=0 and MinVar++ }  if (MinVar== 60) { MinVar=0 and HrVar++ } | | |
| **inputs** | HrVar,MinVar,SecVar | **outputs** | Time\_Displaying Signals |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_02\_V01.2 | **Covers** | Req\_PO3\_DGW\_CYRS\_01\_V01.2 |
| **Author** | Marina Medhat | **Date** | 12/2/2020 |
| **Description:** | The SW shall reset HrVar 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** | HrVar,MinVar,SecVar | **Outputs** | Time\_Displaying Signals |

## **Alarm mode requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_03\_V01.2 | **Covers** | Req\_PO3\_DGW\_CYRS\_02\_V01.1 |
| **Author** | Marina Medhat | **Date** | 12/2/2020 |
| **Description:** | The SW shall keep comparing the alarm variables with the time variable  if ((AlHrVar == HrVar )&( AlMinVar == MinVar )&( AlCycleLength == CycleLength)) {the Buzzer\_Signal = 1} | | |
| **inputs** | HrVar,MinVar,SecVar, AlHrVar,AlMinVar,AlSecVar | **Outputs** | Buzzer\_Signal |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_04\_V01.2 | **Covers** | Req\_PO3\_DGW\_CYRS\_02\_V01.1 |
| **Author** | Marina Medhat | **Date** | 12/2/2020 |
| **Description:** | If (Buzzer\_signal == 1) {the buzzer is enabled}  else {it is not enabled} | | |
| **inputs** | Buzzer\_Signal | **outputs** | - |

## **Stop-Watch mode requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_05\_V01.2 | **Covers** | Req\_PO3\_DGW\_CYRS\_03\_V01.2 |
| **Author** | Marina Medhat | **Date** | 12/2/2020 |
| **Description:** | The SW shall start (StHrVar, StMinVar, StSecVar) with zeros.  If (the user start Stopwatch) {The SecVar Variable starts to increment with delay 1 sec}  if (StSecVar == 60) { StSecVar = 0 and StMinVar++}  if (StMinVar == 60) { StMinVar = 0 and StHrVar++ } | | |
| **inputs** | StHrVar, StMinVar, StSecVar | **Outputs** | StopWatch\_Displaying\_Signals |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_06\_V01.2 | **Covers** | Req\_PO3\_DGW\_CYRS\_04\_V01.1 |
| **Author** | Marina Medhat | **Date** | 12/2/2020 |
| **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** | StHrVar, StMinVar, StSecVar | **outputs** | StopWatch\_Displaying\_Signals |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_07\_V01.2 | **Covers** | Req\_PO3\_DGW\_CYRS\_04\_V01.1 |
| **Author** | Marina Medhat | **Date** | 12/2/2020 |
| **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** | StHrVar, StMinVar, StSecVar | **outputs** | StopWatch\_Displaying\_Signals |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_08\_V01.2 | **Covers** | Req\_PO3\_DGW\_CYRS\_05\_V01.1 |
| **Author** | Marina Medhat | **Date** | 12/2/2020 |
| **Description:** | If(the user press to reset the Stopwatch)  { StHrVar = 0, StMinVar = 0, StSecVar = 0 and display them on the LCD} | | |
| **inputs** | StHrVar, StMinVar, StSecVar | **outputs** | StopWatch\_Displaying\_Signals |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_09\_V01.2 | **Covers** | Req\_PO3\_DGW\_CYRS\_07\_V01.2 |
| **Author** | Marina Medhat | **Date** | 12/2/2020 |
| **Description:** | The SW shall start Current\_Mode with zero  If(mode\_button pressed) {Current\_Mode++}  if(Current\_Mode == 3) { Current\_Mode=0} | | |
| **inputs** | Current\_Mode | **outputs** | Current\_Mode |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_10\_V01.2 | **Covers** | Req\_PO3\_DGW\_CYRS\_08\_V01 |
| **Author** | Marina Medhat | **Date** | 12/2/2020 |
| **Description:** | The SW shall start SetCell with zero in Time display mode.  If(adjust\_button pressed) { SetCell ++}  if(SetCell == 3) { SetCell =0} | | |
| **inputs** | SetCell | **outputs** | SetCell |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_11\_V01.2 | **Covers** | Req\_PO3\_DGW\_CYRS\_08\_V01 |
| **Author** | Marina Medhat | **Date** | 12/2/2020 |
| **Description:** | The SW shall compare the SetCell variable and If equal 0, the position on LCD will be in hours’ cell | | |
| **inputs** | SetCell | **outputs** | Time\_DiSplaying\_Signals |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_12\_V01.2 | **Covers** | Req\_PO3\_DGW\_CYRS\_08\_V01 |
| **Author** | Marina Medhat | **Date** | 12/2/2020 |
| **Description:** | The SW shall compare the SetCell variable and If equal 1, the position on LCD will be in minutes’ cell. | | |
| **inputs** | SetCell | **outputs** | Time\_DiSplaying\_Signals |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_13\_V01.2 | **Covers** | Req\_PO3\_DGW\_CYRS\_08\_V01 |
| **Author** | Marina Medhat | **Date** | 12/2/2020 |
| **Description:** | The SW shall compare the SetCell variable and If equal 2, the position on LCD will be in seconds cell. | | |
| **inputs** | SetCell | **outputs** | Time\_DiSplaying\_Signals |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_14\_V01.2 | **Covers** | Req\_PO3\_DGW\_CYRS\_08\_V01 |
| **Author** | Marina Medhat | **Date** | 12/2/2020 |
| **Description:** | The SW shall check SetCell value in Time display mode.  If(incermenting\_button pressed)  If(SetCell == 0) {HrVar++}  If(SetCell == 1) {MinVar++}  If(SetCell == 2) {SecVar++} | | |
| **inputs** | HrVar,MinVar,SecVar | **outputs** | Time\_DiSplaying\_Signals |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_15\_V01.2 | **Covers** | Req\_PO3\_DGW\_CYRS\_09\_V01 |
| **Author** | Marina Medhat | **Date** | 12/2/2020 |
| **Description:** | The SW shall start AlSetCell with zero in Alarm mode.  If(adjust\_button pressed) { AlSetCell ++}  if(AlSetCell == 3) { AlSetCell =0} | | |
| **inputs** | AlSetCell | **Outputs** | AlSetCell |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_16\_V01.2 | **Covers** | Req\_PO3\_DGW\_CYRS\_09\_V01 |
| **Author** | Marina Medhat | **Date** | 12/2/2020 |
| **Description:** | The SW shall compare the AlSetCell variable and If equal 0, the position on LCD will be in hours’ cell. | | |
| **inputs** | AlSetCell | **Outputs** | Alarm\_Displaying\_Signals |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_17\_V01.2 | **Covers** | Req\_PO3\_DGW\_CYRS\_09\_V01 |
| **Author** | Marina Medhat | **Date** | 12/2/2020 |
| **Description:** | The SW shall compare the AlSetCell variable and If equal 1, the position on LCD will be in minutes’ cell. | | |
| **inputs** | AlSetCell | **outputs** | Alarm\_Displaying\_Signals |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_18\_V01.2 | **Covers** | Req\_PO3\_DGW\_CYRS\_09\_V01 |
| **Author** | Marina Medhat | **Date** | 12/2/2020 |
| **Description:** | The SW shall compare the AlSetCell variable and If equal 2, the position on LCD will be in seconds cell. | | |
| **inputs** | AlSetCell | **outputs** | Alarm\_Displaying\_Signals |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_19\_V01.2 | **Covers** | Req\_PO3\_DGW\_CYRS\_09\_V01 |
| **Author** | Marina Medhat | **Date** | 12/2/2020 |
| **Description:** | The SW shall check AlSetCell value in Alarm mode.  If(incermenting\_button pressed)  If(AlSetCell == 0) {AlHrVar++}  If(AlSetCell == 1) {AlMinVar++}  If(AlSetCell == 2) {AlSecVar++} | | |
| **inputs** | AlMinVar,AlSecVar,AlHrVar | **Outputs** | Alarm\_Displaying\_Signals |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_20\_V01.2 | **Covers** | Req\_PO3\_DGW\_CYRS\_09\_V01 |
| **Author** | Marina Medhat | **Date** | 12/2/2020 |
| **Description:** | The SW shall turn the Buzzer\_Signal to zero if Current\_Mode button is pressed | | |
| **inputs** | Current\_Mode | **outputs** | Buzzer\_Signal |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_21\_V01.2 | **Covers** | Req\_PO3\_DGW\_CYRS\_10\_V01 |
| **Author** | Marina Medhat | **Date** | 12/2/2020 |
| **Description:** | The SW shall start status with zero in Stopwatch mode.  If(adjust\_button pressed) { status ++}  if(status == 3) { status =0} | | |
| **inputs** | Status | **outputs** | Stopwatch\_Displaying\_Signals |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_22\_V01.2 | **Covers** | Req\_PO3\_DGW\_CYRS\_10\_V01 |
| **Author** | Marina Medhat | **Date** | 12/2/2020 |
| **Description:** | The SW shall compare the Status variable and If equal 0, the StopWatch starts incrementing its values (StHrVar, StMinVar, StSecVar). | | |
| **inputs** | Status | **outputs** | Stopwatch\_Displaying\_Signals |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_23\_V01.2 | **Covers** | Req\_PO3\_DGW\_CYRS\_10\_V01 |
| **Author** | Marina Medhat | **Date** | 12/2/2020 |
| **Description:** | If (status==1) {The SW shall stop incrementing its values (StHrVar, StMinVar, StSecVar) and displays it on LCD.} in case of Stopwatch mode | | |
| **inputs** | Status | **outputs** | Stopwatch\_Displaying\_Signals |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_24\_V01.2 | **Covers** | Req\_PO3\_DGW\_CYRS\_10\_V01 |
| **Author** | Marina Medhat | **Date** | 12/2/2020 |
| **Description:** | The SW will compare the Status variable and If equal 2, the StopWatch resume increment its values (StHrVar, StMinVar, StSecVar). | | |
| **inputs** | Status | **outputs** | Stopwatch\_Displaying\_Signals |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_25\_V01.2 | **Covers** | Req\_PO3\_DGW\_CYRS\_10\_V01 |
| **Author** | Marina Medhat | **Date** | 12/2/2020 |
| **Description:** | The SW shall turn the StopWatch\_flag to one when the reset button is pressed in the Stopwatch mode  If (StopWatch\_flag==1) {StHrVar=0, StMinVar=0, StSecVar=0} | | |
| **inputs** | StopWatch\_flag | **outputs** | StHrVar, StMinVar, StSecVar |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO3\_DGW\_SRS\_26\_V01.2 | **Covers** | Req\_PO3\_DGW\_HSI\_01\_V01 |
| **Author** | Marina Medhat | **Date** | 12/2/2020 |
| **Description:** | The SW shall configure the pins according to the mapping of the configuration pins on the microcontoller | | |
| **inputs** | --- | **outputs** | --- |

Table 3 Reference Table

|  |  |  |
| --- | --- | --- |
| Reference Input Documents | Version | Status |
| CYRS | 1.6 | Released |
| HSI | 1.5 | Released |