

# Software Requirement Analysis for Digital Watch System

Project Team

**Watch BACK**

Date

**2019-10-29**

---

Team Information

**201811263 백종원**

**201711376 김경진**

**201810568 박용준**

## Table of Contents

1	Introduction _____	5
1.1	Purpose _____	5
1.2	Scope _____	5
1.2.1	Project Team _____	5
1.2.2	Constraints _____	5
1.2.3	Product Utilization _____	5
1.2.4	Development Environment _____	5
1.3	Definition, acronyms, and abbreviations _____	5
1.4	Reference _____	5
1.5	Overview _____	5
2	Overall Description _____	5
2.1	Product Perspective _____	6
2.2	Product functions _____	6
2.2.1	Date-Time _____	6
2.2.2	Stopwatch _____	6
2.2.3	Backlight _____	6
2.2.4	Alarm _____	6
2.3	User characteristics _____	6
2.4	Constraints _____	6
2.5	Assumptions and dependencies _____	7
3	Structured Analysis _____	7
3.1	System Context Diagram _____	7
3.1.1	Basic System Context Diagram _____	7
3.1.2	Event List _____	7
3.1.3	The System Context Diagram _____	8
Introduction to SW Engineering (2019)      Watch BACK		2

3.2	Data Flow Diagram	8
3.2.1	DFD level 0	8
3.2.1.1	DFD	8
3.2.1.2	Process Specification	8
3.2.1.2.1	Process 0	8
3.2.1.3	Data Dictionary	9
3.2.2	DFD Level 1	9
3.2.2.1	DFD	9
3.2.2.2	Process Specification	10
3.2.2.2.1	Process 1	10
3.2.2.2.2	Process 2	10
3.2.2.2.3	Process 3	13
3.2.2.2.4	Process 4	14
3.2.2.3	Data Dictionary	14
3.2.3	DFD level 2	16
3.2.3.1	DFD	16
3.2.3.2	Process Specification	17
3.2.3.2.1	Process 1.1	17
3.2.3.2.2	Process 1.2	17
3.2.3.2.3	Process 1.3	17
3.2.3.2.4	Process 1.4	18
3.2.3.2.5	Process 1.5	18
3.2.3.2.6	Process 2	18
3.2.3.2.7	Process 3.1	21
3.2.3.2.8	Process 3.2	23
3.2.3.2.9	Process 3.3	23

3.2.3.2.10	Process 4.1	23
3.2.3.2.11	Process 4.2	23
3.2.3.3	Data Dictionary	24
3.2.4	DFD Level 3	24
3.2.4.1	DFD	24
3.2.4.2	Process Specification	26
3.2.4.2.1	Process 1.1	26
3.2.4.2.2	Process 1.2	26
3.2.4.2.3	Process 1.3	26
3.2.4.2.4	Process 1.4	27
3.2.4.2.5	Process 1.5	27
3.2.4.2.6	Process 2.1	27
3.2.4.2.7	Process 2.2	30
3.2.4.2.8	Process 2.3	30
3.2.4.2.9	Process 3.1.1	31
3.2.4.2.10	Process 3.1.2	32
3.2.4.2.11	Process 3.1.3	32
3.2.4.2.12	Process 4.1.1	32
3.2.4.2.13	Process 4.1.2	33
3.2.4.3	Data Dictionary	33
3.2.5	DFD Level 4	34
3.2.5.1	State Transition Diagram ( <i>Display &amp; Speaker Controller, Filter Controller</i> )	34
3.2.6	Overall DFD	34

## 1 Introduction

### 1.1 Purpose

본 문서는 2019년 건국대학교의 소프트웨어공학개론 강의의 프로젝트를 설명하기 위해 작성된 문서이다. 프로젝트는 Digital Watch (DW) System에서 사용할 수 있는 SW를 구현하는 것이다. 본 문서는 특히 프로젝트의 structured design을 다룬다.

### 1.2 Scope

#### 1.2.1 Project Team

Watch BACK (Team 1)

#### 1.2.2 Constraints

HW (손목시계)와 연동까지 고려하지 않고, SW로만 구동할 수 있도록 한다.

#### 1.2.3 Product Utilization

개발이 완료된 후 실제 손목시계의 SW를 개발하기 위한 프로토타입으로 삼을 수 있다.

#### 1.2.4 Development Environment

IDE: Eclipse

Compiler: Cygwin

### 1.3 Definition, acronyms, and abbreviations

SW: Software

HW: Hardware

### 1.4 Reference

IEEE Std. 830-1998

### 1.5 Overview

## 2 Overall Description

## 2.1 Product Perspective

대상 제품은 실제 손목시계에 사용될 수 있는 제품이 될 수 있다. HW (버튼)에 의한 동작을 처리하고, 처리한 결과는 HW (화면)에 출력한다. 실제 HW에 의한 동작은 SW 및 console화면으로 처리하여 기능의 동작 유무를 확인하도록 한다. 시계 HW는 4개의 버튼과 LCD 화면을 가진 것으로 한다.

## 2.2 Product functions

### 2.2.1 Date-Time

화면에 일자와 시각을 표시한다.

시각은 24시 표기로 표현한다.

### 2.2.2 Stopwatch

시간의 경과를 알려주고, 특정 순간의 시간을 알려준다.

1/100초 단위로 측정이 가능하다.

Lap time 기록이 가능하다.

### 2.2.3 Backlight

출력하는 문자의 색깔을 노란색으로 표시한다.

### 2.2.4 Alarm

Alarm이 설정되면 alarm indicator가 켜진다.

정해놓은 시각에 소리 (beep 음)로 알려준다.

알람은 5초간 울린다.

알람이 울릴 때 A, B, C, D 중 아무 버튼을 누르면 소리가 꺼진다.

## 2.3 User characteristics

## 2.4 Constraints

날짜의 표기법은 '월-일'이다.

초기 시각은 2019년 01월 01일 00시 00분 00초이다.

2019-1-1부터 2099-12-31까지 표시가 가능하다.

알람을 설정할 때 시와 분을 모두 설정해야 한다.

## 2.5 Assumptions and dependencies

버튼 입력은 키보드 입력으로 대체한다.

버튼이 여러 개 입력되었을 때 우선순위는  $D > C > B > A$ 이다.

시각을 수정할 때 선택된 부분이 깜박이는 표현은 숫자 밑에 밑줄을 표시하는 것으로 대체한다.

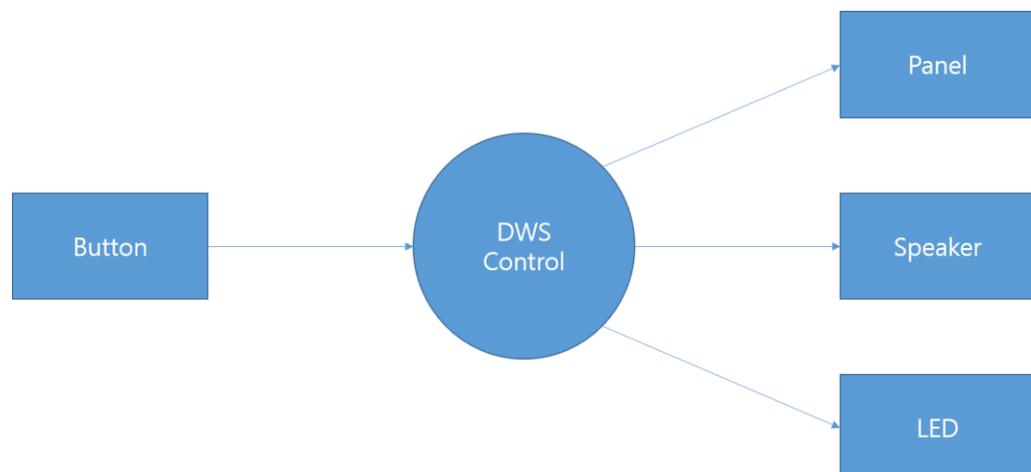
Backlight는 실제 시계의 경우 LED를 켜는 것으로 구현되어야 하지만 출력되는 문자의 색을 바꾸는 것으로 대체한다.

알람음은 PC의 beep으로 대체한다.

## 3 Structured Analysis

### 3.1 System Context Diagram

#### 3.1.1 Basic System Context Diagram

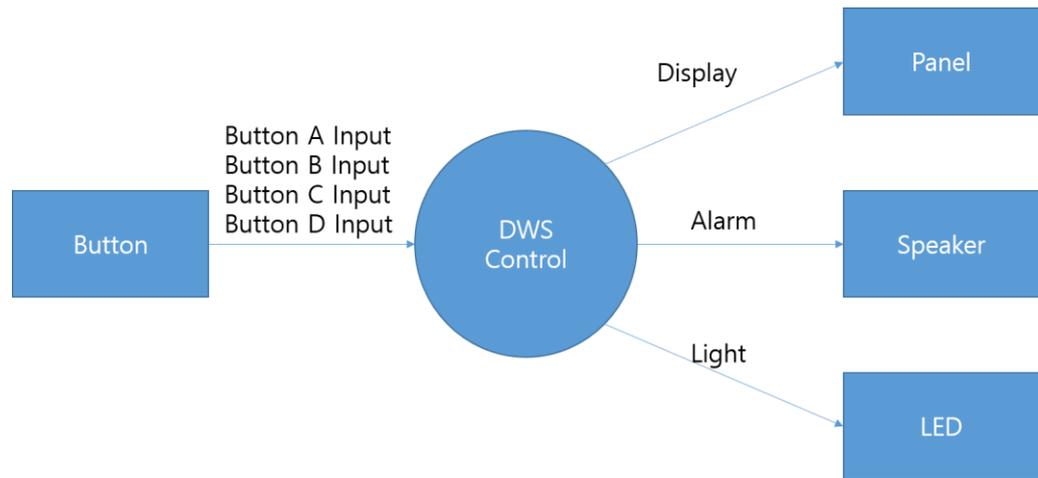


#### 3.1.2 Event List

Input/Output Event	Description
Button A Input	Detects if button a is pressed
Button B Input	Detects if button b is pressed
Button C Input	Detects if button c is pressed

Button D Input	Detects if button d is pressed
Display	Displays current information
Alarm	Rings alarm
Light	Changes font color

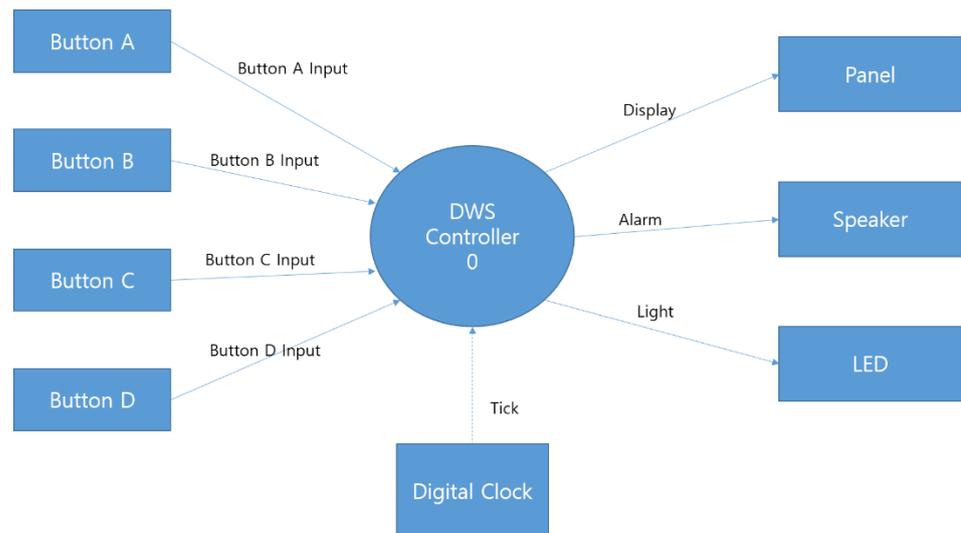
### 3.1.3 The System Context Diagram



## 3.2 Data Flow Diagram

### 3.2.1 DFD level 0

#### 3.2.1.1 DFD



#### 3.2.1.2 Process Specification

##### 3.2.1.2.1 Process 0

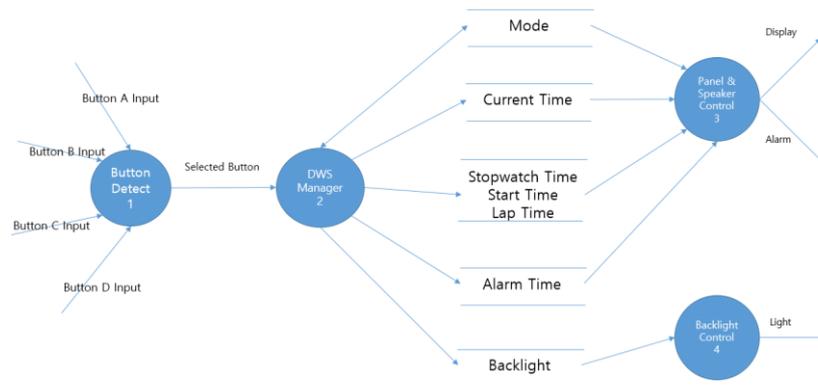
<b>Reference Number</b>	<b>0</b>
Name	DWS Controller
Input	Button A Input, Button B Input, Button C Input, Button D Input
Output	Display, Alarm, Light
Process Specification	버튼 A, B, C, D를 입력으로 받아 그에 대한 정보를 처리하고, 패널, 스피커, LED에 각각 Display, Alarm, Light를 출력하는 시스템

### 3.2.1.3 Data Dictionary

Input/Output Event	Description	Format / Type
Button A Input	Detects if button a is pressed	Boolean
Button B Input	Detects if button b is pressed	Boolean
Button C Input	Detects if button c is pressed	Boolean
Button D Input	Detects if button d is pressed	Boolean
Display	패널에 현재 정보를 출력한다. 패널은 7개의 구역으로 나뉘어 있다. 7개의 구역에 각각에 해당하는 정보의 집합이다. 이 때, 패널에서 바로 출력될 수 있도록 가공되었다.	Modified Data for 7 parts of Panel
Alarm	Rings Alarm with Speaker	On/Off
Light	Changes Text Color with LED	On/Off

## 3.2.2 DFD Level 1

### 3.2.2.1 DFD



### 3.2.2.2 Process Specification

#### 3.2.2.2.1 Process 1

<b>Reference Number</b>	<b>1</b>
<b>Name</b>	Button Detect
<b>Input</b>	Button A Input, Button B Input, Button C Input, Button D Input
<b>Output</b>	Selected Button
<b>Process Specification</b>	<p>대기 시간 동안 하나 이상의 Button Input이 있다면 단위 시간을 시작하고 단위 시간 동안 추가로 Button Input을 받는다. 이때 처음 누른 버튼도 입력 받는 버튼들 중 하나로 취급한다.</p> <p>단위 시간 동안 입력 받는 여러 Button Input들 중 버튼 우선순위 D&gt;C&gt;B&gt;A에 따라 단 하나의 Button Input만 선택해 Selected Button으로 내보낸다.</p> <p>대기 시간 동안 입력이 없는 경우에는 No Button을 내보낸다.</p>

#### 3.2.2.2.2 Process 2

<b>Reference Number</b>	<b>2</b>
<b>Name</b>	DWS Manager
<b>Input</b>	Selected Button
<b>Output</b>	Mode, Current Time, Stopwatch Time, Start Time, Lap Time, Alarm Time, Backlight

Process Specification	<p>input으로 Selected Button과 Mode가 들어온다. 이때 들어온 Selected Button과 Mode에 대해 다음과 같이 반응한다.</p> <p>모드의 0번지가 1인 경우, 들어온 버튼이 있다면 알람 종료 명령을 내린다. 그리고 5초가 지나면 종료한다.</p> <p>그리고 모드의 0번지가 0인 경우, (Mode[1], Mode[2])과 Selected Button에 대해 다음과 같이 반응한다.</p> <ul style="list-style-type: none"> <li>(1,1) timekeeping <ul style="list-style-type: none"> <li>a 모드를 (1,2)로 바꾼다</li> <li>b</li> <li>c 모드를 (2,1)로 바꾼다</li> <li>d Backlight</li> </ul> </li> <li>(1,2) timekeeping_change_sec <ul style="list-style-type: none"> <li>a 모드를 (1,1)로 바꾼다</li> <li>b 현재 시각 초 1 증가</li> <li>c 모드를 (1,3)로 바꾼다</li> <li>d Backlight</li> </ul> </li> <li>(1,3) timekeeping_change_hour <ul style="list-style-type: none"> <li>a 모드를 (1,1)로 바꾼다</li> <li>b 현재 시각 시간 1 증가</li> <li>c 모드를 (1,4)로 바꾼다</li> <li>d Backlight</li> </ul> </li> <li>(1,4) timekeeping_change_min <ul style="list-style-type: none"> <li>a 모드를 (1,1)로 바꾼다</li> <li>b 현재 시각 분 1 증가</li> <li>c 모드를 (1,5)로 바꾼다</li> <li>d Backlight</li> </ul> </li> <li>(1,5) timekeeping_change_year <ul style="list-style-type: none"> <li>a 모드를 (1,1)로 바꾼다</li> <li>b 현재 시각 년 1 증가</li> <li>c 모드를 (1,6)로 바꾼다</li> <li>d Backlight</li> </ul> </li> <li>(1,6) timekeeping_change_month <ul style="list-style-type: none"> <li>a 모드를 (1,1)로 바꾼다</li> <li>b 현재 시각 달 1 증가</li> <li>c 모드를 (1,7)로 바꾼다</li> <li>d Backlight</li> </ul> </li> </ul>
-----------------------	---

	<p>(1,7) timekeeping_Change_day</p> <ul style="list-style-type: none"> <li>a 모드를 (1,1)로 바꾼다</li> <li>b 현재 시각 일 1 증가</li> <li>c 모드를 (1,2)로 바꾼다</li> <li>d Backlight</li> </ul> <p>(2.1) stopwatch</p> <ul style="list-style-type: none"> <li>a if(Stopwatch indi == 0) <ul style="list-style-type: none"> <li>Stopwatch Time = 0</li> <li>Start Time = 0</li> </ul> </li> <li>else if(Stopwatch indi == 1) <ul style="list-style-type: none"> <li>LAP Time = Stopwatch Time</li> <li>모드를 (2,2)로 바꾼다</li> </ul> </li> <li>b if (Stopwatch indi == 0) <ul style="list-style-type: none"> <li>Stopwatch indi = 1</li> <li>Stopwatch Time += Start Time</li> <li>Start Time = Current Time</li> </ul> </li> <li>else if (Stopwatch indi == 1) <ul style="list-style-type: none"> <li>Stopwatch indi = 0</li> <li>Start Time += Current Time - Start Time</li> </ul> </li> </ul> <p>Time</p> <ul style="list-style-type: none"> <li>c 모드를 (3,1)로 바꾼다</li> <li>d Backlight</li> </ul> <p>(2,2) stopwatch_LAP</p> <ul style="list-style-type: none"> <li>a</li> <li>b 모드를 (2,1)로 바꾼다</li> <li>c 모드를 (3,1)로 바꾼다</li> <li>d Backlight</li> </ul> <p>(3.1) alarm</p> <ul style="list-style-type: none"> <li>a 모드를 (3,2)로 바꾼다</li> <li>b</li> <li>c</li> <li>d Backlight</li> </ul> <p>(3.2) alarm_change_hour</p> <ul style="list-style-type: none"> <li>a</li> <li>b 알람 시각 시간 1 증가</li> <li>c 모드를 (3,3)로 바꾼다</li> <li>d Backlight</li> </ul>
--	--

	<p>(3.3) alarm_change_minute</p> <p>a</p> <p>b 알람 시각 분 1 증가</p> <p>c 모드를 (3,2)로 바꾼다</p> <p>d Backlight</p> <p>또한 DWS 매니저는 프로세스 연산 시간 동안 생기는 실제 시간과의 오차를 보정해주는 역할을 하고, 거기에 맞춰 스톱워치와 알람 시간을 관리한다.</p> <p>만약 연산에 시간이 걸리지 않는다면, 1ms마다 받는 tick에 맞춰 현재 시간을 1ms 늘리면 된다. 하지만 연산에는 시간이 걸리기에, 모든 프로세스를 구현한 후, 걸리는 시간의 평균을 구해서 1ms마다 (1 - 평균)을 더하면 된다. 또한 프로세스별로 연산 시간이 다르기에, 자주 쓰는 프로세스에 따라 평균에 가중치를 둘 수 있다.</p> <p>현재 시간 보정 및 스톱워치 알람 관리 순서는 다음과 같다.</p> <p>현재시간 += (1 - a)</p> <p>if(stopwatch indi)</p> <p style="padding-left: 40px;">Stopwatch Time=현재시간-Start Time</p> <p>if(alarm indi)</p> <p style="padding-left: 40px;">if(알람시간&lt;현재시간&lt;알람시간 + 5)</p> <p style="padding-left: 80px;">Alarm Buzzing = 1</p> <p style="padding-left: 40px;">else</p> <p style="padding-left: 80px;">Alarm Buzzing =0</p>
--	---

## 3.2.2.2.3 Process 3

<b>Reference Number</b>	<b>3</b>
Name	Panel & Speaker Control
Input	Mode, Current Time, Stopwatch Time, Lap

	Time, Alarm Time
Output	Display, Alarm
Process Specification	Panel & Speaker Command를 읽어와서, 하드웨어에 맞게 출력하도록 Panel에 Display를 전달한다.

## 3.2.2.2.4 Process 4

<b>Reference Number</b>	<b>4</b>
Name	Backlight Control
Input	Backlight
Output	Light
Process Specification	Backlight의 값에 따라 Light를 실행한다.

## 3.2.2.3 Data Dictionary

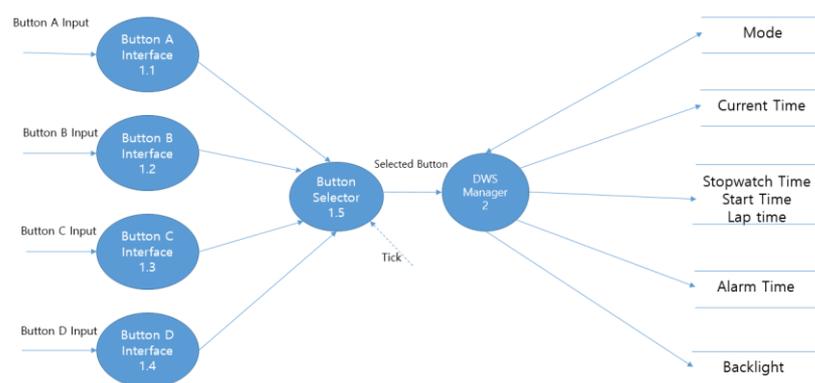
Input/Output Event	Description	Format / Type
Button A Input	Detects if button a is pressed	Boolean
Button B Input	Detects if button b is pressed	Boolean
Button C Input	Detects if button c is pressed	Boolean
Button D Input	Detects if button d is pressed	Boolean
Selected Button	Returns a button if there is at least one button input. Otherwise returns 'No Button'	Button A Button B Button C Button D No Button
Mode	현재 모드에 대한 정보를 저장하는 5칸짜리 배열. 0번지는 알람이 울리고있는지(1), 아닌지(0)를 저장한다. 1번지는 Timekeeping, Stopwatch, Alarm과 같은 큰 분류의 모드에 대한 정보를 저장한다. 2번지는 Timekeeping에서 시간을 설정하는 기능처럼 대 분류에서	integer array size of 5. : Mode[] ""[0]: Alarm Buzzing ""[1]: 대 분류 ""[2]: 소 분류 ""[3]: Stopwatch Indicator

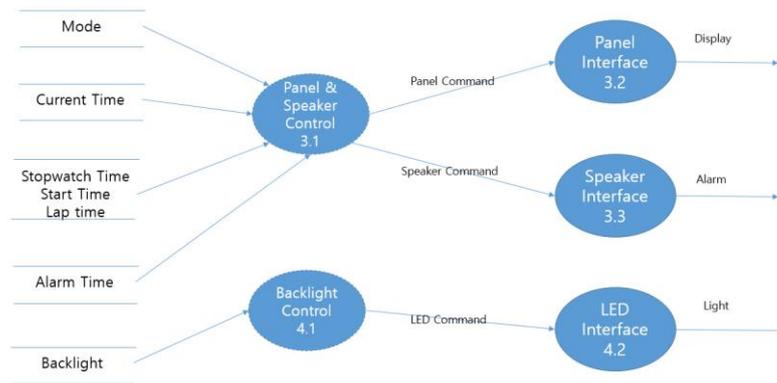
	<p>세부적인 기능을 담당하는 소 분류를 저장한다.</p> <p>3번지는 stopwatch가 켜져 있는지(1), 꺼져 있는지(0).</p> <p>4번지는 alarm indicator가 켜져 있는지(1), 꺼져 있는지(0)</p> <p>대분류 소분류가 가능한 케이스는 다음과 같다. (대분류, 소분류)</p> <p>(1,1): timekeeping -&gt;기본 모드</p> <p>(1,2): timekeeping_change_sec -&gt;시간 설정, 초 변경</p> <p>(1,3): : timekeeping_change_hour -&gt;시간 설정, 시 변경</p> <p>(1,4): timekeeping_change_min -&gt;시간 설정, 초 분 변경</p> <p>(1,5): timekeeping_change_year -&gt;시간 설정, 년 변경</p> <p>(1,6): timekeeping_change_month -&gt;시간 설정, 월 변경</p> <p>(1,7): timekeeping_change_day -&gt;시간 설정, 일 변경</p> <p>(2,1): stopwatch -&gt;스톱워치 기본 모드</p> <p>(2,2): stopwatch_LAP -&gt;스톱워치, 랩 타임 출력</p> <p>(3,1): alarm_change_hour -&gt;알람 시간 설정, 시 변경</p> <p>(3,2): alarm_change_minute -&gt;알람 시각 설정, 분 변경</p>	<p>""[4]: Alarm Indicator</p>
Current Time	Stores current time	Time

		(YY/MM/DD/ HH/MM/SS/MS)
Stopwatch Time	Stores the time Stopwatch was activated	Time()
Lap Time	Stores lap time(s)	Time()
Alarm Time	Stores the time when alarm is set to ring	Time()
Backlight	Stores if the watch is on backlight mode or not	Boolean
Display	패널에 현재 정보를 출력한다. 패널은 7개의 구역으로 나뉘어 있다. 7개의 구역에 각각에 해당하는 정보의 집합이다. 이 때, 패널에서 바로 출력될 수 있도록 가공되었다.	Modified Data for 7 parts of Panel
Alarm	Rings Alarm with Speaker	On/Off
Light	Changes Text Color with LED	On/Off
Start Time	스톱워치가 계산을 시작한 시간을 저장한다.	Time()

### 3.2.3 DFD Level 2

#### 3.2.3.1 DFD





### 3.2.3.2 Process Specification

#### 3.2.3.2.1 Process 1.1

<b>Reference Number</b>	<b>1.1</b>
Name	Button A Interface
Input	Button A Input
Output	Button A input
Process Specification	Button A의 입력을 감지한다.

#### 3.2.3.2.2 Process 1.2

<b>Reference Number</b>	<b>1.2</b>
Name	Button B Interface
Input	Button B Input
Output	Button B input
Process Specification	Button B의 입력을 감지한다.

#### 3.2.3.2.3 Process 1.3

<b>Reference Number</b>	<b>1.3</b>
Name	Button C Interface
Input	Button C Input
Output	Button C input
Process Specification	Button C의 입력을 감지한다.

Specification	
---------------	--

## 3.2.3.2.4 Process 1.4

<b>Reference Number</b>	<b>1.4</b>
Name	Button D Interface
Input	Button D Input
Output	Button D input
Process Specification	Button D의 입력을 감지한다.

## 3.2.3.2.5 Process 1.5

<b>Reference Number</b>	<b>1.5</b>
Name	Button Selector
Input	Button A Input, Button B Input, Button C Input, Button D Input
Output	Selected Button
Process Specification	<p>대기 시간 동안 하나 이상의 Button Input이 있다면 단위 시간을 시작하고 단위 시간 동안 추가로 Button Input을 받는다. 이때 처음 누른 버튼도 입력 받는 버튼들 중 하나로 취급한다.</p> <p>단위 시간 동안 입력 받는 여러 Button Input들 중 버튼 우선순위 <math>D &gt; C &gt; B &gt; A</math>에 따라 단 하나의 Button Input만 선택해 Selected Button으로 내보낸다.</p> <p>대기 시간 동안 입력이 없는 경우에는 No Button을 내보낸다.</p>

## 3.2.3.2.6 Process 2

<b>Reference Number</b>	<b>2</b>
Name	DWS Manager
Input	Mode, Selected Button
Output	Mode, Current Time, Stopwatch Time, Lap Time, Alarm Time, Backlight, Start Time

Process Specification	<p>input으로 Selected Button과 Mode가 들어온다. 이때 들어온 Selected Button과 Mode에 대해 다음과 같이 반응한다.</p> <p>모드의 0번지가 1인 경우, 들어온 버튼이 있다면 알람 종료 명령을 내린다. 그리고 5초가 지나면 종료한다.</p> <p>그리고 모드의 0번지가 0인 경우, (Mode[1], Mode[2])과 Selected Button에 대해 다음과 같이 반응한다.</p> <ul style="list-style-type: none"> <li>(1,1) timekeeping <ul style="list-style-type: none"> <li>a 모드를 (1,2)로 바꾼다</li> <li>b</li> <li>c 모드를 (2,1)로 바꾼다</li> <li>d Backlight</li> </ul> </li> <li>(1,2) timekeeping_change_sec <ul style="list-style-type: none"> <li>a 모드를 (1,1)로 바꾼다</li> <li>b 현재 시각 초 1 증가</li> <li>c 모드를 (1,3)로 바꾼다</li> <li>d Backlight</li> </ul> </li> <li>(1,3) timekeeping_change_hour <ul style="list-style-type: none"> <li>a 모드를 (1,1)로 바꾼다</li> <li>b 현재 시각 시간 1 증가</li> <li>c 모드를 (1,4)로 바꾼다</li> <li>d Backlight</li> </ul> </li> <li>(1,4) timekeeping_change_min <ul style="list-style-type: none"> <li>a 모드를 (1,1)로 바꾼다</li> <li>b 현재 시각 분 1 증가</li> <li>c 모드를 (1,5)로 바꾼다</li> <li>d Backlight</li> </ul> </li> <li>(1,5) timekeeping_change_year <ul style="list-style-type: none"> <li>a 모드를 (1,1)로 바꾼다</li> <li>b 현재 시각 년 1 증가</li> <li>c 모드를 (1,6)로 바꾼다</li> <li>d Backlight</li> </ul> </li> <li>(1,6) timekeeping_change_month <ul style="list-style-type: none"> <li>a 모드를 (1,1)로 바꾼다</li> <li>b 현재 시각 달 1 증가</li> <li>c 모드를 (1,7)로 바꾼다</li> <li>d Backlight</li> </ul> </li> </ul>
-----------------------	---

	<p>(1,7) timekeeping_Change_day</p> <ul style="list-style-type: none"> <li>a 모드를 (1,1)로 바꾼다</li> <li>b 현재 시각 일 1 증가</li> <li>c 모드를 (1,2)로 바꾼다</li> <li>d Backlight</li> </ul> <p>(2.1) stopwatch</p> <ul style="list-style-type: none"> <li>a if(Stopwatch indi == 0) <ul style="list-style-type: none"> <li>Stopwatch Time = 0</li> <li>Start Time = 0</li> </ul> </li> <li>else if(Stopwatch indi == 1) <ul style="list-style-type: none"> <li>LAP Time = Stopwatch Time</li> <li>모드를 (2,2)로 바꾼다</li> </ul> </li> <li>b if (Stopwatch indi == 0) <ul style="list-style-type: none"> <li>Stopwatch indi = 1</li> <li>Stopwatch Time += Start Time</li> <li>Start Time = Current Time</li> </ul> </li> <li>else if (Stopwatch indi == 1) <ul style="list-style-type: none"> <li>Stopwatch indi = 0</li> <li>Start Time += Current Time - Start Time</li> </ul> </li> </ul> <p>Time</p> <ul style="list-style-type: none"> <li>c 모드를 (3,1)로 바꾼다</li> <li>d Backlight</li> </ul> <p>(2,2) stopwatch_LAP</p> <ul style="list-style-type: none"> <li>a</li> <li>b 모드를 (2,1)로 바꾼다</li> <li>c 모드를 (3,1)로 바꾼다</li> <li>d Backlight</li> </ul> <p>(3.1) alarm</p> <ul style="list-style-type: none"> <li>a 모드를 (3,2)로 바꾼다</li> <li>b</li> <li>c</li> <li>d Backlight</li> </ul> <p>(3.2) alarm_change_hour</p> <ul style="list-style-type: none"> <li>a</li> <li>b 알람 시각 시간 1 증가</li> <li>c 모드를 (3,3)로 바꾼다</li> <li>d Backlight</li> </ul>
--	--

	<p>(3.3) alarm_change_minute</p> <ul style="list-style-type: none"> <li>a</li> <li>b 알람 시각 분 1 증가</li> <li>c 모드를 (3,2)로 바꾼다</li> <li>d Backlight</li> </ul> <p>또한 DWS 매니저는 프로세스 연산 시간 동안 생기는 실제 시간과의 오차를 보정해주는 역할을 하고, 거기에 맞춰 스톱워치와 알람 시간을 관리한다.</p> <p>만약 연산에 시간이 걸리지 않는다면, 1ms마다 받는 tick에 맞춰 현재 시간을 1ms 늘리면 된다. 하지만 연산에는 시간이 걸리기에, 모든 프로세스를 구현한 후, 걸리는 시간의 평균을 구해서 1ms마다 (1 - 평균)을 더하면 된다. 또한 프로세스별로 연산 시간이 다르기에, 자주 쓰는 프로세스에 따라 평균에 가중치를 둘 수 있다.</p> <p>현재 시간 보정 및 스톱워치 알람 관리 순서는 다음과 같다.</p> <p>현재시간 += (1 - a)</p> <pre> if(stopwatch indi)     Stopwatch Time=현재시간-Start Time if(alarm indi)     if(알람시간&lt;현재시간&lt;알람시간 + 5)         Alarm Buzzing = 1     else         Alarm Buzzing =0     </pre>
--	--

3.2.3.2.7 Process 3.1

<b>Reference Number</b>	<b>3.1</b>
Name	Panel & Speaker Control
Input	Mode, Current Time, Stopwatch Time, Lap Time, Alarm time, Start Time

Output	Panel Command, Speaker Command
Process Specification	<p>Panel에서 출력될 7 부분의 정보인 Panel Command를 구성하고, 알람을 울릴지 말지에 대한 Speaker Command를 구성한다. 이후 이 두 정보를 Panel Interface, Speaker Interface에 전달한다.</p> <p>Panel Command는 8칸의 배열로 0~6번지는 다음과 같이 구성된다.</p> <p>모드의 대분류 (Mode[1])를 읽고,</p> <p>&lt;Timekeeping&gt;</p> <ul style="list-style-type: none"> <li>1 Day of Week</li> <li>2 Month</li> <li>3 Day</li> <li>4 Alarm Indicator</li> <li>5 Hour</li> <li>6 Min</li> <li>7 Sec</li> </ul> <p>&lt;Stopwatch&gt;</p> <ul style="list-style-type: none"> <li>1 "ST"</li> <li>2 Hour</li> <li>3 Min</li> <li>4 Alarm Indicator</li> <li>5 Stopwatch Min</li> <li>6 Stopwatch Sec</li> <li>7 Stopwatch MilliSec</li> </ul> <p>&lt;Alarm&gt;</p> <ul style="list-style-type: none"> <li>1 "AL"</li> <li>2 Month</li> <li>3 Day</li> <li>4 Alarm Indicator</li> <li>5 Alarm Hour</li> <li>6 Alarm Min</li> <li>7 ""</li> </ul> <p>만약 시간을 수정하고 있다면, 모드의 소분류 (Mode[2])를 읽고, 7번지에 깜빡일 위치를 전달한다.</p> <p>Speaker Command는 다음과 같이 구성된다.</p> <p>Current Time과 Alarm Time이 일치한다면, Speaker Command의 값을 On으로 전달하고,</p>

	그렇지 않은 경우에는 Off로 전달한다.
--	------------------------

## 3.2.3.2.8 Process 3.2

<b>Reference Number</b>	<b>3.2</b>
Name	Panel Interface
Input	Panel Command
Output	Display
Process Specification	Panel & Speaker Command를 읽어와서, 하드웨어에 맞게 출력하도록 Panel에 Display를 전달한다.

## 3.2.3.2.9 Process 3.3

<b>Reference Number</b>	<b>3.3</b>
Name	Speaker Interface
Input	Speaker Command
Output	Alarm
Process Specification	Speaker Command가 On이라면 알람을 울리고, Off라면 울리지 않도록 Speaker에 Alarm을 전달한다.

## 3.2.3.2.10 Process 4.1

<b>Reference Number</b>	<b>4.1</b>
Name	Backlight Control
Input	Backlight
Output	LED Command
Process Specification	버튼 D가 Selected Button으로 들어와 Backlight를 켜야 할 경우, LED Command로 On을 내보내고, 그렇지 않은 경우 Off를 전달한다.

## 3.2.3.2.11 Process 4.2

<b>Reference Number</b>	<b>4.2</b>
-------------------------	------------

Name	LED Interface
Input	LED Command
Output	Light
Process Specification	LED Command가 On인 경우, LED에서 글자의 색을 나타내도록 Light를 On으로 내보낸다. 반대로는 Off를 내보낸다.

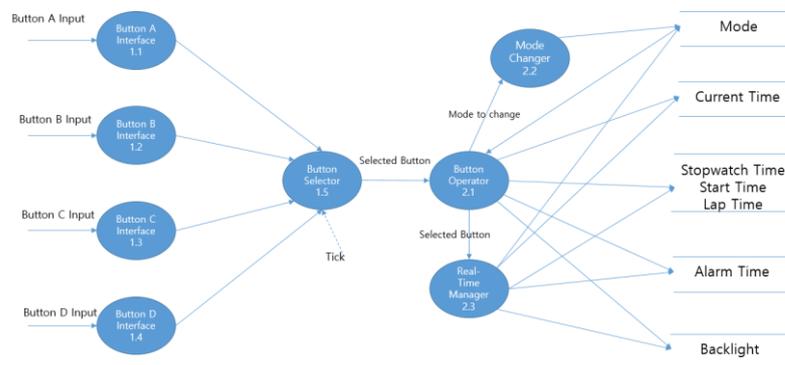
## 3.2.3.3 Data Dictionary

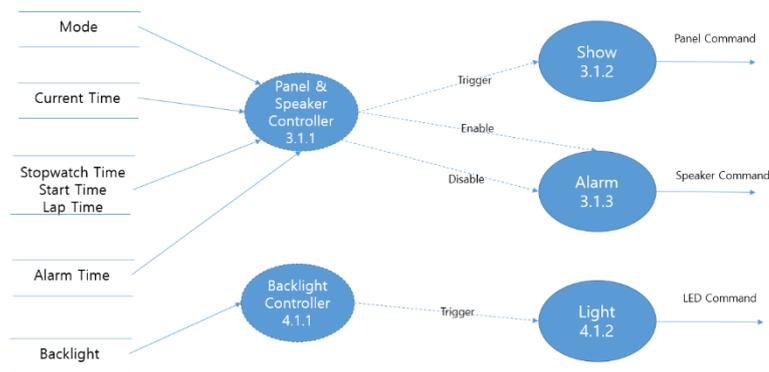
Input/Output Event	Description	Format / Type
Button A Input	Detects if button a is pressed	Boolean
Button B Input	Detects if button b is pressed	Boolean
Button C Input	Detects if button c is pressed	Boolean
Button D Input	Detects if button d is pressed	Boolean
Selected Button	Returns a button if there is at least one button input. Otherwise returns 'No Button'	Button A Button B Button C Button D No Button
Mode	Level 1의 Data Dictionary와 같음.	Level 1의 Data Dictionary와 같음.
Current Time	Stores current time	Time() (YY/MM/DD/HH/MM/SS/MS)
Stopwatch Time	Stores the time Stopwatch was activated	Time()
Lap Time	Stores lap time(s)	Time()
Alarm Time	Stores the time when alarm is set to ring	Time()
Backlight	Stores if the watch is on backlight mode or not	Boolean
Panel Command	출력될 7개 부분에 대한 정보를 저장한다.	Panel_Command[8]. Data for 7 parts of Panel,

		Parts to Highlight
Speaker Command	Sends Alarm Information to HW	On/Off
LED Command	Sends Text Color Information to HW	On/Off
Display	패널에 현재 정보를 출력한다. 패널은 7개의 구역으로 나뉘어 있다. 7개의 구역에 각각에 해당하는 정보의 집합이다. 이 때, 패널에서 바로 출력될 수 있도록 가공되었다.	Modified Data for 7 parts of Panel, parts to highlight
Alarm	Rings Alarm with Speaker	On/Off
Light	Changes Text Color with LED	On/Off
Start Time	스톱워치가 계산을 하기 시작한 시간을 저장한다.	Time()

### 3.2.4 DFD Level 3

#### 3.2.4.1 DFD





### 3.2.4.2 Process Specification

#### 3.2.4.2.1 Process 1.1

<b>Reference Number</b>	<b>1.1</b>
Name	Button A Interface
Input	Button A input
Output	Button A input
Process Specification	Button A의 입력을 감지한다.

#### 3.2.4.2.2 Process 1.2

<b>Reference Number</b>	<b>1.2</b>
Name	Button B Interface
Input	Button B input
Output	Button B input
Process Specification	Button B의 입력을 감지한다.

#### 3.2.4.2.3 Process 1.3

<b>Reference Number</b>	<b>1.3</b>
Name	Button C Interface
Input	Button C input
Output	Button C input
Process Specification	Button C의 입력을 감지한다.

## 3.2.4.2.4 Process 1.4

<b>Reference Number</b>	<b>1.4</b>
Name	Button D Interface
Input	Button D input
Output	Button D input
Process Specification	Button D의 입력을 감지한다.

## 3.2.4.2.5 Process 1.5

<b>Reference Number</b>	<b>1.5</b>
Name	Button Selector
Input	Button A input, Button B input, Button C input, Button D input
Output	Selected Button
Process Specification	<p>대기 시간 동안 하나 이상의 Button Input이 있다면 단위 시간을 시작하고 단위 시간 동안 추가로 Button Input을 받는다. 이때 처음 누른 버튼도 입력 받는 버튼들 중 하나로 취급한다.</p> <p>단위 시간 동안 입력 받는 여러 Button Input들 중 버튼 우선순위 D&gt;C&gt;B&gt;A에 따라 단 하나의 Button Input만 선택해 Selected Button으로 내보낸다.</p> <p>대기 시간 동안 입력이 없는 경우에는 No Button을 내보낸다.</p>

## 3.2.4.2.6 Process 2.1

<b>Reference Number</b>	<b>2.1</b>
Name	Button Operator
Input	Selected Button
Output	Mode to change, Current Time, Stopwatch Time, Lap Time, Alarm Time, Backlight
Process	모드의 0번지가 0인 경우,

Specification	<p>(Mode[1], Mode[2])과 Selected Button에 대해 다음과 같이 반응한다.</p> <p>(1,1) timekeeping</p> <ul style="list-style-type: none"> <li>a 모드를 (1,2)로 바꾼다</li> <li>b</li> <li>c 모드를 (2,1)로 바꾼다</li> <li>d Backlight</li> </ul> <p>(1,2) timekeeping_change_sec</p> <ul style="list-style-type: none"> <li>a 모드를 (1,1)로 바꾼다</li> <li>b 현재 시각 초 1 증가</li> <li>c 모드를 (1,3)로 바꾼다</li> <li>d Backlight</li> </ul> <p>(1,3) timekeeping_change_hour</p> <ul style="list-style-type: none"> <li>a 모드를 (1,1)로 바꾼다</li> <li>b 현재 시각 시간 1 증가</li> <li>c 모드를 (1,4)로 바꾼다</li> <li>d Backlight</li> </ul> <p>(1,4) timekeeping_change_min</p> <ul style="list-style-type: none"> <li>a 모드를 (1,1)로 바꾼다</li> <li>b 현재 시각 분 1 증가</li> <li>c 모드를 (1,5)로 바꾼다</li> <li>d Backlight</li> </ul> <p>(1,5) timekeeping_change_year</p> <ul style="list-style-type: none"> <li>a 모드를 (1,1)로 바꾼다</li> <li>b 현재 시각 년 1 증가</li> <li>c 모드를 (1,6)로 바꾼다</li> <li>d Backlight</li> </ul> <p>(1,6) timekeeping_change_month</p> <ul style="list-style-type: none"> <li>a 모드를 (1,1)로 바꾼다</li> <li>b 현재 시각 달 1 증가</li> <li>c 모드를 (1,7)로 바꾼다</li> <li>d Backlight</li> </ul> <p>(1,7) timekeeping_Change_day</p> <ul style="list-style-type: none"> <li>a 모드를 (1,1)로 바꾼다</li> <li>b 현재 시각 일 1 증가</li> <li>c 모드를 (1,2)로 바꾼다</li> <li>d Backlight</li> </ul>
---------------	--

	<p>(2.1) stopwatch</p> <ul style="list-style-type: none"> <li>a if(Stopwatch indi == 0) <ul style="list-style-type: none"> <li>Stopwatch Time = 0</li> <li>Start Time = 0</li> </ul> </li> <li>else if(Stopwatch indi == 1) <ul style="list-style-type: none"> <li>LAP Time = Stopwatch Time</li> <li>모드를 (2,2)로 바꾼다</li> </ul> </li> <li>b if (Stopwatch indi == 0) <ul style="list-style-type: none"> <li>Stopwatch indi = 1</li> <li>Stopwatch Time += Start Time</li> <li>Start Time = Current Time</li> </ul> </li> <li>else if (Stopwatch indi == 1) <ul style="list-style-type: none"> <li>Stopwatch indi = 0</li> <li>Start Time += Current Time - Start Time</li> </ul> </li> <li>c 모드를 (3,1)로 바꾼다</li> <li>d Backlight</li> </ul> <p>(2,2) stopwatch_LAP</p> <ul style="list-style-type: none"> <li>a</li> <li>b 모드를 (2,1)로 바꾼다</li> <li>c 모드를 (3,1)로 바꾼다</li> <li>d Backlight</li> </ul> <p>(3.1) alarm</p> <ul style="list-style-type: none"> <li>a 모드를 (3,2)로 바꾼다</li> <li>b</li> <li>c</li> <li>d Backlight</li> </ul> <p>(3.2) alarm_change_hour</p> <ul style="list-style-type: none"> <li>a</li> <li>b 알람 시각 시간 1 증가</li> <li>c 모드를 (3,3)로 바꾼다</li> <li>d Backlight</li> </ul> <p>(3.3) alarm_change_minute</p> <ul style="list-style-type: none"> <li>a</li> </ul>
--	--

	b 알람 시각 분 1 증가 c 모드를 (3,2)로 바꾼다 d Backlight
--	---

## 3.2.4.2.7 Process 2.2

<b>Reference Number</b>	<b>2.2</b>
Name	Mode Changer
Input	Mode to change
Output	Mode
Process Specification	입력된 모드로 모드를 바꾼다.

## 3.2.4.2.8 Process 2.3

<b>Reference Number</b>	<b>2.3</b>
Name	Real-Time Manager
Input	Selected Button
Output	Mode, Current time, Stopwatch Time, Lap Time, Alarm Time, Backlight
Process Specification	<p>Real-Time Manager는 프로세스 연산 시간 동안 생기는 실제 시간과의 오차를 보정해주는 역할을 하고, 거기에 맞춰 스톱워치와 알람 시간을 관리한다.</p> <p>만약 연산에 시간이 걸리지 않는다면, 1ms마다 받는 tick에 맞춰 현재 시간을 1ms늘리면 된다. 하지만 연산에는 시간이 걸리기에, 모든 프로세스를 구현한 후, 걸리는 시간의 평균을 구해서 1ms마다 (1 - 평균)을 더하면 된다. 또한 프로세스별로 연산 시간이 다르기에, 자주 쓰는 프로세스에 따라 평균에 가중치를 둘 수 있다.</p> <p>현재 시간 보정 및 스톱워치 알람 관리 순서는 다음과 같다.</p> <p>현재시간 += (1 - a)</p>

	<pre> if(stopwatch indi)     Stopwatch Time=현재시간-Start Time if(alarm indi)     if(알람시간&lt;현재시간&lt;알람시간 + 5)         Alarm Buzzing = 1     else         Alarm Buzzing =0 </pre>
--	--

## 3.2.4.2.9 Process 3.1.1

<b>Reference Number</b>	<b>3.1.1</b>
Name	Panel & Speaker Controller
Input	Mode, Current Time, Stopwatch Time, Lap Time
Output	Enable, Disable, Trigger of Commands
Process Specification	<p>Panel에서 출력될 7 부분의 정보인 Panel Command를 구성하고, 알람을 울릴지 말지에 대한 Speaker Command를 구성한다. 그리고 Show를 Trigger. Alarm을 Enable/Disable 한다.</p> <p>Panel Command는 8칸의 배열로 0~6번지는 다음과 같이 구성된다.</p> <p>모드의 대분류 (Mode[1])를 읽고,</p> <p>&lt;Timekeeping&gt;</p> <ul style="list-style-type: none"> <li>1 Day of Week</li> <li>2 Month</li> <li>3 Day</li> <li>4 Alarm Indicator</li> <li>5 Hour</li> <li>6 Min</li> <li>7 Sec</li> </ul> <p>&lt;Stopwatch&gt;</p> <ul style="list-style-type: none"> <li>1 "ST"</li> <li>2 Hour</li> <li>3 Min</li> <li>4 Alarm Indicator</li> <li>5 Stopwatch Min</li> <li>6 Stopwatch Sec</li> </ul>

	<p>7 Stopwatch MilliSec</p> <p>&lt;Alarm&gt;</p> <p>1 "AL"</p> <p>2 Month</p> <p>3 Day</p> <p>4 Alarm Indicator</p> <p>5 Alarm Hour</p> <p>6 Alarm Min</p> <p>7 ""</p> <p>만약 시간을 수정하고 있다면, 모드의 소분류 (Mode[2])를 읽고, 7번지에 깜빡일 위치를 전달한다.</p> <p>Speaker Command는 다음과 같이 구성된다. Current Time과 Alarm Time이 일치한다면, Speaker Command의 값을 On으로 전달하고, 그렇지 않은 경우에는 Off로 전달한다</p>
--	---

## 3.2.4.2.10 Process 3.1.2

<b>Reference Number</b>	<b>3.1.2</b>
Name	Show
Input	Trigger
Output	Panel Command
Process Specification	패널 화면에 데이터를 보이라는 명령을 내린다.

## 3.2.4.2.11 Process 3.1.3

<b>Reference Number</b>	<b>3.1.3</b>
Name	Alarm
Input	Enable, Disable
Output	Speaker Command
Process Specification	입력 값에 따라, 스피커를 켜거나 끈다. 즉, 알람을 울리거나 끈다.

## 3.2.4.2.12 Process 4.1.1

<b>Reference Number</b>	<b>4.1.1</b>
Name	Backlight Controller
Input	Backlight
Output	Trigger
Process Specification	백라이트 값에 따라, Light를 활성화시킨다. 2초 후 꺼지는 것은 LED에서 구현하는 것으로 한다.

## 3.2.4.2.13 Process 4.1.2

<b>Reference Number</b>	<b>4.1.2</b>
Name	Light
Input	Trigger
Output	LED Command
Process Specification	트리거 된 후, 출력되는 글씨의 색을 바꾼다. 2초 후, 원래대로 돌아온다.

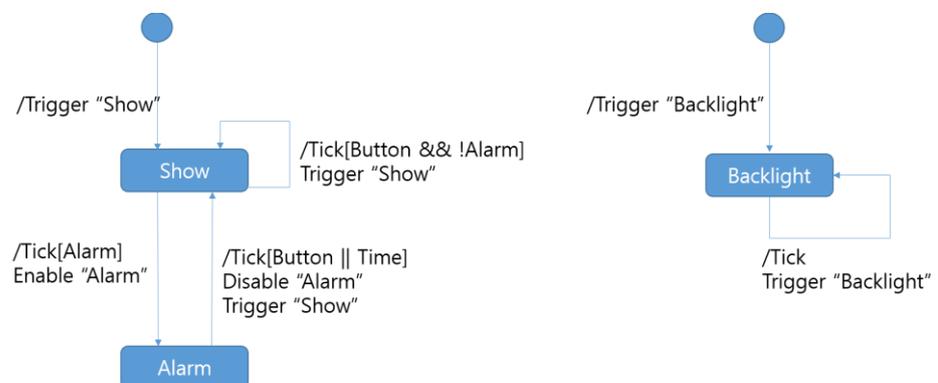
## 3.2.4.3 Data Dictionary

<b>Input/Output Event</b>	<b>Description</b>	<b>Format / Type</b>
Button A Input	Detects if button a is pressed	Boolean
Button B Input	Detects if button b is pressed	Boolean
Button C Input	Detects if button c is pressed	Boolean
Button D Input	Detects if button d is pressed	Boolean
Selected Button	Returns a button if there is at least one button input. Otherwise returns 'No Button'.	Button A Button B Button C Button D No Button
Mode	Level 1의 Data Dictionary와 같음.	
Current Time	Stores current time	Time

		((YY/MM/DD/HH/MM/SS/MS)
Stopwatch Time	Stores the time Stopwatch was activated	Time()
Lap Time	Stores lap time(s)	Time()
Alarm Time	Stores the time when alarm is set to ring	Time()
Backlight	Stores if the watch is on backlight mode or not	Boolean
Panel Command	Sends Current Information to HW	Panel_command[8] Data for 7 parts of Panel, parts to highlight
Speaker Command	Sends Alarm Information to HW	On/Off
LED Command	Sends Text Color Information to HW	On/Off
Mode to Change	Mode that Button Operator wants to change	Timekeeping/ Stopwatch/ Alarm
Start Time	스톱워치가 계산을 하기 시작한 시간을 저장한다.	Time()

3.2.5 DFD Level 4

3.2.5.1 State Transition Diagram (*Display & Speaker Controller, Filter Controller*)



3.2.6 Overall DFD

