

Software Requirement Analysis for Digital Watch System

Project Team

Watch BACK

Date

2019-11-21

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	10
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	11
3.2.2.2.4	Process 4	11
3.2.2.3	Data Dictionary	11
3.2.3	DFD level 2	15
3.2.3.1	DFD	15
3.2.3.2	Process Specification	16
3.2.3.2.1	Process 1.1	16
3.2.3.2.2	Process 1.2	16
3.2.3.2.3	Process 1.3	16
3.2.3.2.4	Process 1.4	17
3.2.3.2.5	Process 1.5	17
3.2.3.2.6	Process 2	17
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	24
3.2.3.3	Data Dictionary	24
3.2.4	DFD Level 3	25
3.2.4.1	DFD	25
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	31
3.2.4.2.9	Process 3.1.1	32
3.2.4.2.10	Process 3.1.2	33
3.2.4.2.11	Process 3.1.3	34
3.2.4.2.12	Process 3.1.4	35
3.2.4.2.13	Process 4.1.1	35
3.2.4.2.14	Process 4.1.2	35
3.2.4.3	Data Dictionary	36
3.2.5	DFD Level 4	37
3.2.5.1	State Transition Diagram (<i>Display & Speaker Controller, Filter Controller</i>)	37
3.2.6	Overall DFD	37

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.

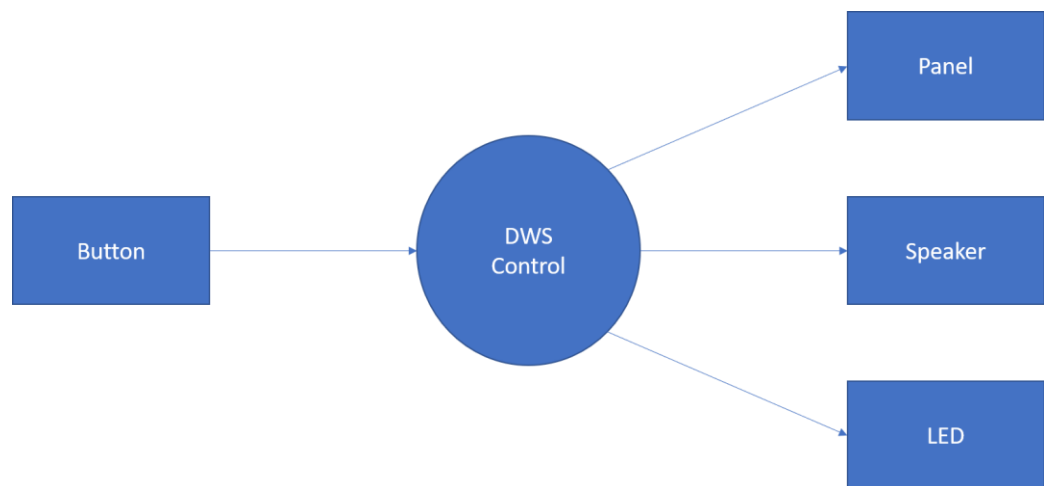
Backlight는 실제 시계의 경우 LED를 켜는 것으로 구현되어야 하지만 출력되는 문자의 색을 바꾸는 것으로 대체한다.

알람음은 PC의 beep으로 대체한다.

3 Structured Analysis

3.1 System Context Diagram

3.1.1 Basic System Context Diagram

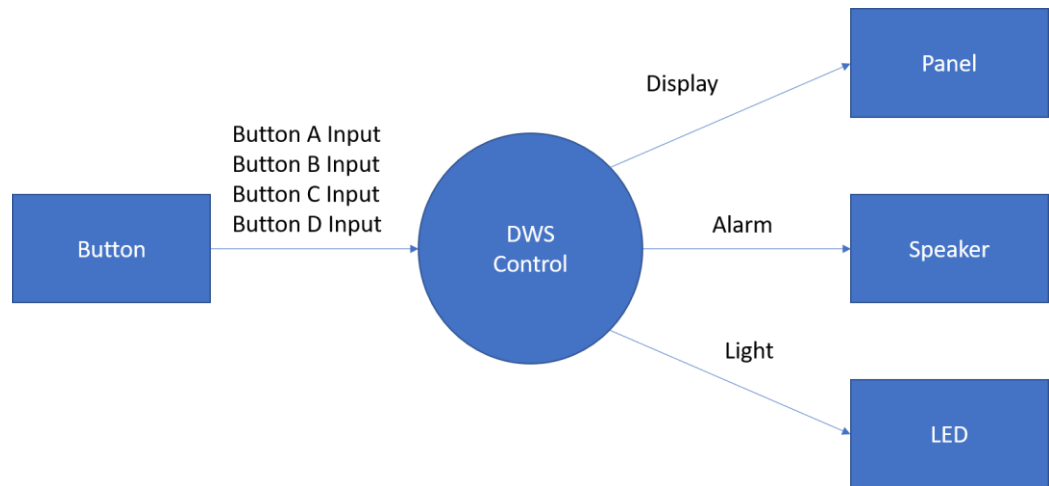


3.1.2 Event List

Input/Output Event	Description
Button A Input	버튼 A의 입력
Button B Input	버튼 B의 입력
Button C Input	버튼 C의 입력

Button D Input	버튼 D의 입력
Display	화면에 현재 정보를 출력
Alarm	현재 알람을 울릴지 말지에 대한 정보 전달
Light	현재 백라이트를 켜지 말지에 대한 정보 전달

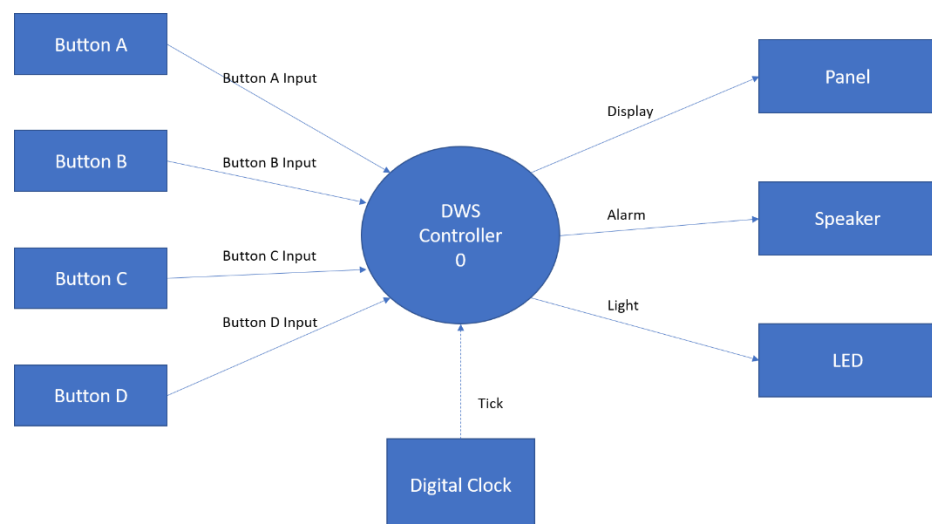
3.1.3 The System Context Diagram



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

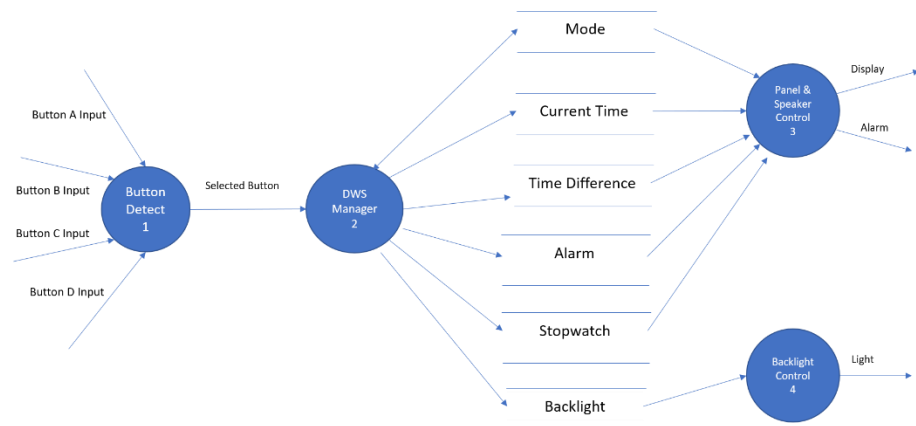
Reference Number	0
Name	DWS Controller
Input	Button A Input, Button B Input, Button C Input, Button D Input
Output	Display, Alarm, Light
Process Specification	버튼 A, B, C, D를 입력으로 받아 그에 대한 정보를 처리하고, 패널, 스피커, LED에 각각 Display, Alarm, Light를 출력하는 시스템이다.

3.2.1.3 Data Dictionary

Input/Output Event	Description	Format / Type
Button A Input	버튼 A의 입력	Bool
Button B Input	버튼 B의 입력	Bool
Button C Input	버튼 C의 입력	Bool
Button D Input	버튼 D의 입력	Bool
Display	패널에 현재 정보를 출력한다. 패널은 8개의 구역으로 나뉘어 있으며, 각각의 모드별로 8개의 구역에 출력해야 하는 항목들이 다르다. 따라서 길이가 2인 문자열 8개의 배열로 각 상황에서 어떤 값을 출력해야하는지를 알려준다. 0-6번지에는 해당 구역에 출력해야 하는 정보를, 그리고 7번지에는 시간을 변경할 때 숫자가 깜빡일 구역을 저장한다. 이때 데이터들은 패널에서 바로 출력될 수 있도록 가공되었다.	char[8][3]
Alarm	현재 알람을 울릴 지 말 지에 대한 정보 전달	On/Off
Light	현재 백라이트를 켤 지 말 지에 대한 정보 전달	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

Reference Number	1
Name	Button Detect
Input	Button A Input, Button B Input, Button C Input, Button D Input
Output	Selected Button
Process Specification	대기하고 있다가 Button Input이 하나라도 감지되면 단위 시간을 시작하고 단위 시간 동안 추가로 Button Input을 받는다. 이렇게 받은 input들 중 우선순위에 따라 단 하나의 Button Input만 선택해 Selected Button으로 내보내며, 대기 시간 동안 입력이 없는 경우에는 No Button을 내보낸다.

3.2.2.2.2 Process 2

Reference Number	2
Name	DWS Manager
Input	Selected Button, Mode
Output	Mode, Current Time, Time Difference, Stopwatch, Alarm, Backlight, Backlight
Process Specification	Button Detect에서 내보낸 Selected Button값과, 모드를 input으로 가지며, Selected Button으로 어떤 동작을 수행해야 하는지를 결정한

	다. 그 다음 프로세스 연산 시간 동안 생기는 실제 오차와 보정하며, 이에 맞춰 stopwatch 와 alarm 시각을 관리한다.
--	--

3.2.2.2.3 Process 3

Reference Number	3
Name	Panel & Speaker Control
Input	Mode, Current Time, Time Difference, Alarm, Stopwatch
Output	Display, Alarm
Process Specification	Panel & Speaker Command를 읽어와서, 하드웨어에 맞게 출력하도록 Panel에 Display를 전달한다.

3.2.2.2.4 Process 4

Reference Number	4
Name	Backlight Control
Input	Backlight
Output	Light
Process Specification	Backlight data store의 값에 따라 Light의 On/Off를 결정한다. Backlight data store에는 백라이트가 꺼져야 하는 시각과 글자 색상정보가 담겨있다.

3.2.2.3 Data Dictionary

Input/Output Event	Description	Format / Type
Button A Input	버튼 A의 입력	Bool
Button B Input	버튼 B의 입력	Bool
Button C Input	버튼 C의 입력	Bool
Button D Input	버튼 D의 입력	Bool
Selected Button	Button Detect에서 단위 시간 내의 button input들 중 우선순위에 따라 정해진 버튼. 버튼	int

	A, B, C, D의 경우 각각 1, 2, 3, 4를, 단위 시간 동안 버튼의 입력이 없었을 경우에는 0으로 설정된다.	
Mode	<p>현재 모드에 대한 정보를 저장하는 5 항목짜리 구조체 data store.</p> <p>첫번째 항목 alarm_buzzing은 알람이 울리고 있는 중인지 (1) 아닌지(0)를 저장한다.</p> <p>두번째 항목 category_alpha는 대분류에 대한 정보를 저장하여 1일 때에는 timekeeping 모드, 2일 때에는 stopwatch 모드, 3일 때에는 alarm 모드를 의미한다.</p> <p>세번째 항목 category_beta는 각 대분류 내의 세부적인 기능을 담당하는 소분류를 저장한다.</p> <p>네번째 항목 stopwatch_indicator는 stopwatch가 켜져 있는지(1), 꺼져 있는지(0)를 저장한다.</p> <p>다섯번째 항목 alarm_indicator는 알람이 설정되어 있는지(1), 그렇지 않은지(0)를 저장한다.</p> <p>category_alpha, category_beta로 조합할 수 있는 경우들은 아래와 같다.</p> <p>(1,1): timekeeping ->기본 모드</p> <p>(1,2): timekeeping_change_sec ->시간 설정 (초 변경)</p> <p>(1,3): timekeeping_change_hour ->시간 설정 (시 변경)</p> <p>(1,4): timekeeping_change_min</p>	<pre>struct Mode { Bool alarm_buzzing; int category_alpha; int category_beta; Bool stopwatch_indicator; Bool alarm_indicator; }</pre>

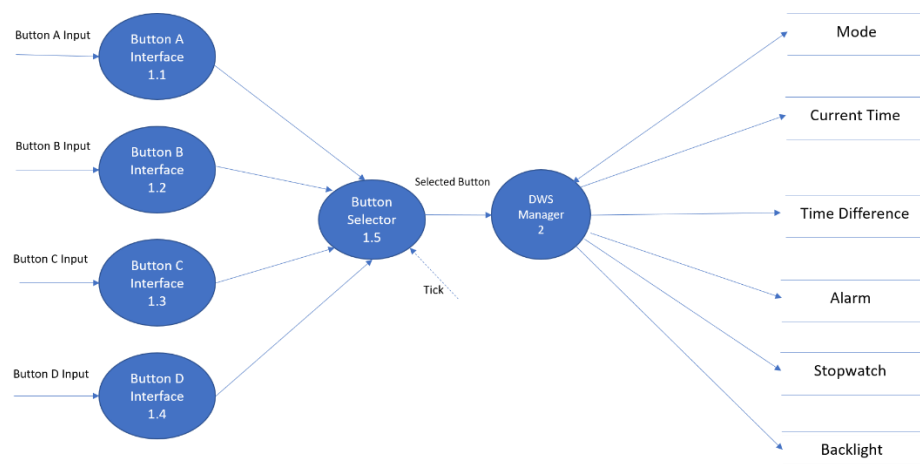
	->시간 설정 (분 변경) (1,5): timekeeping_change_year ->시간 설정 (년 변경) (1,6): timekeeping_change_month ->시간 설정 (월 변경) (1,7): timekeeping_change_day ->시간 설정 (일 변경) (2,1): stopwatch ->스톱워치 기본 모드 (2,2): stopwatch_LAP ->스톱워치 (랩 타임 출력) (3,1): alarm_change_hour ->알람 시간 설정 (시 변경) (3,2): alarm_change_minute ->알람 시각 설정 (분 변경)	
Current Time	Digital watch의 현재 시각을 저장하는 data store이다. 절대 시각과는 차이가 있다.	<pre>struct Time { int YY; // 년도 int MT; // 월 int DD; // 일 int HH; // 시 int MM; // 분 int SS; // 초 int MS; // ms int WD; // 요일 }</pre>
Time Difference	절대 시각과 Current Time의 차이를 저장하는 data store이다.	struct Time
Alarm	알람에 대한 정보를 저장하기 위한 data store이다. 알람이 울리도록 설정한 시각과 알람 설정 시각 후 0-5초 사이에 버튼 입력이 들어오면 false	<pre>struct Alm { Time alarmTime; Bool snooze; Int ring; }</pre>

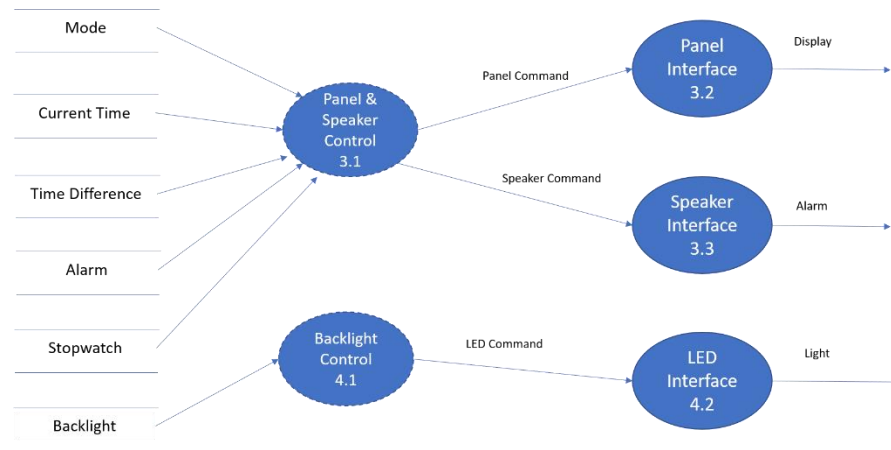
	<p>가 되는 snooze, CT.SS와 비교되어 알람 소리를 일정하게 하는 ring으로 구성되어 있다.</p> <p>Snooze는 Current Time의 초가 0보다 크고 5보다 작을 때 버튼 입력이 들어오면 Current Time의 초가 5가 되기 전까지 false 상태가 된다.</p> <p>Stopwatch로 측정되고 있는 시간을 저장하는 data store이다.</p>	
Stopwatch	<p>스탑워치에 대한 정보를 저장하기 위한 data store이다.</p> <p>Stopwatch 구조체는 Time 타입의 항목 stopwatchTime, startTime, lapTime, initialTime을 가진다.</p> <p>stopwatchTime은 스톱워치로 측정되고 있는 시간을 저장한다. 스톱워치를 켜고 있을 때 현재 흘러가고 있는 시간을 저장한다.</p> <p>startTime은 가장 최근으로 시작을 누른 시각을 저장한다.</p> <p>lapTime은 랩타임을 저장한다.</p> <p>initialTime은 stopwatchTime의 값을 저장한다. 이는 startTime이 가장 최근으로 시작을 누른 시각을 저장하므로 stopwatchTime이 갖던 값을 저장해 이전까지의 기록을 보정해 주기 위함이다. 따라서 initialTime을 이용해 측정을 멈춘 시점의 stopwatchTime을 저장하여, 측정을 멈추더라도 이어서 측정을 할 수 있도록 한다.</p>	<pre>struct Stopwatch { Time stopwatchTime; Time startTime; Time lapTime; Time initialTime; }</pre>
Backlight	<p>Backlight에 대한 정보를 저장하는 data store이다.</p> <p>BacklightTime에는 백라이트가</p>	<pre>Struct Backlight { Time BacklightTime; int value;</pre>

	꺼져야 하는 시각, 즉 버튼 D를 누를 시각 + 2초가 저장되며, value에는 화면에 나타날 글씨의 색에 대한 정보가 담겨있다.	}
Display	패널에 현재 정보를 출력한다. 패널은 8개의 구역으로 나뉘어 있으며, 각각의 모드별로 8개의 구역에 출력해야 하는 항목들이 다르다. 따라서 길이가 2인 문자열 8개의 배열로 각 상황에서 어떤 값을 출력해야 하는지를 알려준다. 0-6번지에는 해당 구역에 출력해야 하는 정보를, 그리고 7번지에는 시간을 변경할 때 숫자가 깜빡일 구역을 저장한다. 이때 데이터들은 패널에서 바로 출력될 수 있도록 가공되었다.	char[8][3]
Alarm	현재 알람을 울릴 지 말 지에 대한 정보를 전달한다.	On/Off
Light	현재 백라이트를 켤 지 말 지에 대한 정보를 전달한다.	On/Off

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

Reference Number	1.1
Name	Button A Interface
Input	Button A Input
Output	Button A input
Process Specification	Button A의 입력을 감지한다. 대문자 또는 소문자 A가 들어오면 true를, 그렇지 않으면 false를 반환한다.

3.2.3.2.2 Process 1.2

Reference Number	1.2
Name	Button B Interface
Input	Button B Input
Output	Button B input
Process Specification	Button B의 입력을 감지한다. 대문자 또는 소문자 B가 들어오면 true를, 그렇지 않으면 false를 반환한다.

3.2.3.2.3 Process 1.3

Reference Number	1.3
Name	Button C Interface
Input	Button C Input
Output	Button C input

Process Specification	Button C의 입력을 감지한다. 대문자 또는 소문자 C가 들어오면 true를, 그렇지 않으면 false를 반환한다.
-----------------------	--

3.2.3.2.4 Process 1.4

Reference Number	1.4
Name	Button D Interface
Input	Button D Input
Output	Button D input
Process Specification	Button D의 입력을 감지한다. 대문자 또는 소문자 D가 들어오면 true를, 그렇지 않으면 false를 반환한다.

3.2.3.2.5 Process 1.5

Reference Number	1.5
Name	Button Selector
Input	Button A Input, Button B Input, Button C Input, Button D Input
Output	Selected Button
Process Specification	<p>Button Selector는 대기하고 있다가 Button Input이 하나라도 들어오면 단위 시간을 시작하고 단위 시간 동안 추가로 Button Input을 받는다 (이때 처음 누른 버튼도 입력 받는 버튼들 중 하나로 취급한다).</p> <p>그리고 단위 시간 동안 입력 받는 모든 input들 중 버튼 우선순위 $D > C > B > A$에 따라 단 하나의 Button Input만 Selected Button으로 선택해 리턴한다. Selected Button이 A, B, C, D인 경우 각각 1, 2, 3, 4를 내보내며, 대기 시간 동안 입력이 없는 경우에는 No Button, 즉 0을 반환한다.</p>

3.2.3.2.6 Process 2

Reference Number	2
-------------------------	----------

Name	DWS Manager
Input	Mode, Selected Button
Output	Mode, Current Time, Time Difference, Alarm, Stopwatch, Backlight
Process Specification	<p>Button Detect에서 내보낸 Selected Button값과, 모드를 input으로 가진다.</p> <p>DWS Manager가 하는 일을 순서대로 나열하면 다음과 같다.</p> <p>1. Selected Button으로 어떤 동작을 수행해야 하는지를 결정한다.</p> <p>알람이 울리는 상태에서는 어떤 버튼을 눌러도 알람이 꺼지므로 알람부터 처리한다. 알람이 울리는 상태에서 버튼의 입력이 있으면 알람을 끄고 Alarm Time을 2019년 1월 1일 0시 0분 0초 00ms로 초기화한다.</p> <p>그리고 Mode의 대분류, 소분류, Selected Button 순서로 읽어가며 각 상황에 맞는 작업을 한다. (대분류, 소분류)와 Selected Button에 대해 다음과 같이 반응한다.</p> <p>(1,1) timekeeping</p> <ul style="list-style-type: none"> a 모드를 (1,2)로 바꾼다 b 변경 사항 없음 c 모드를 (2,1)로 바꾼다 d Backlight 켜기 <p>(1,2) timekeeping_change_sec</p> <ul style="list-style-type: none"> a 모드를 (1,1)로 바꾼다 b 현재 시각 초 1 증가 c 모드를 (1,3)로 바꾼다 d Backlight 켜기 <p>(1,3) timekeeping_change_hour</p> <ul style="list-style-type: none"> a 모드를 (1,1)로 바꾼다 b 현재 시각 시간 1 증가 c 모드를 (1,4)로 바꾼다 d Backlight 켜기 <p>(1,4) timekeeping_change_min</p> <ul style="list-style-type: none"> a 모드를 (1,1)로 바꾼다 b 현재 시각 분 1 증가 c 모드를 (1,5)로 바꾼다 d Backlight 켜기

	<p>(1,5) timekeeping_change_year</p> <ul style="list-style-type: none"> a 모드를 (1,1)로 바꾼다 b 현재 시각 년 1 증가 c 모드를 (1,6)로 바꾼다 d Backlight 켜기 <p>(1,6) timekeeping_change_month</p> <ul style="list-style-type: none"> a 모드를 (1,1)로 바꾼다 b 현재 시각 달 1 증가 c 모드를 (1,7)로 바꾼다 d Backlight 켜기 <p>(1,7) timekeeping_change_day</p> <ul style="list-style-type: none"> a 모드를 (1,1)로 바꾼다 b 현재 시각 일 1 증가 c 모드를 (1,2)로 바꾼다 d Backlight 켜기 <p>백라이트 시작 시각 = 현재 시각</p> <p>(2.1) stopwatch</p> <ul style="list-style-type: none"> a 스톱워치가 안 켜진 경우 스탑워치 구조체의 모든 항목들을 초기화한다. 만약 스톱워치가 켜진 상태이면 stopwatchTime의 값을 initialTime에 저장하고 모드를 (2,2)로 변경한다. b 스톱워치가 안 켜진 경우: 스탑워치를 켜다. 그리고 stopwatchTime의 값을 전부 0으로, startTime에는 Current Time의 값을 저장한다. 스탑워치가 켜진 경우: 스탑워치를 끈다. stopwatchTime의 값을 initialTime에 저장한다. c 모드를 (3,1)로 바꾼다 d Backlight 켜기 <p>(2,2) stopwatch_LAP</p> <ul style="list-style-type: none"> a stopwatchTime의 값을 lapTime에 저장한다. b 모드를 (2,1)로 바꾼다 c 모드를 (3,1)로 바꾼다
--	--

	<p>d Backlight 켜기</p> <p>(3.1) alarm</p> <p>a 모드를 (3,2)로 바꾼다</p> <p>b 알람 설정을 켜거나 끈다.</p> <p>c 모드를 (1,1)로 바꾼다</p> <p>d Backlight 켜기</p> <p>(3.2) alarm_change_hour</p> <p>a 모드를 (3,1)로 바꾼다</p> <p>b 알람 시각 시간 1 증가</p> <p>c 모드를 (3,3)로 바꾼다</p> <p>d Backlight 켜기</p> <p>(3.3) alarm_change_minute</p> <p>a 모드를 (3,1)로 바꾼다</p> <p>b 알람 시각 분 1 증가</p> <p>c 모드를 (3,2)로 바꾼다</p> <p>d Backlight 켜기</p> <p>Timekeeping 모드에서는 현재 시각을 변경할 때 스탑워치 시작 시각과 백라이트가 꺼져야 하는 시각도 함께 변경해, stopwatch로 측정한 시간과 백라이트가 꺼져야 하는 시각의 변동이 없도록 한다. 또한, 날짜를 설정하는 과정에서 실제 달력에 맞게 윤년 및 월별 일 수가 계산되며, 최대치가 된 상태에서 다시 입력하면 최저값으로 초기화해주는 과정이 포함되어 있다.</p> <p>Alarm 설정 모드에서도 알람 시각 세팅 시에 Timekeeping 모드에서처럼 시간 단위를 벗어나기 직전 초기화 과정이 포함되어 있다.</p> <p>한편 알람이 울릴 때를 제외하고 D를 누르면 백라이트가 켜지는데, 이 경우 Backlight 구조체의 value 값을 초록색으로, 백라이트가 꺼져야 하는 시각에는 현재 시각 Current Time에 2초를 더한 값을 저장한 다음 Backlight Time의 값을 60진법에 맞게 보정해준다.</p> <p>2. 프로세스 연산 시간 동안 생기는 실제 오차와의 보정, 이에 맞춰 stopwatch와 alarm 시각을 관리한다.</p>
--	---

	<p>가장 먼저 Current Time을 동기화한다. DWS가 실행될 때마다 실제 세계의 현재 시각을 불러와 Time Difference와의 차이만큼을 Current Time에 저장한다.</p> <p>그 후 시각을 60진법에 맞게 보정한다. 이렇게 Current Time이 보정이 되면 stopwatch가 켜진 경우에 대해 보정된 Current Time을 바탕으로 stopwatchTime도 보정한다.</p> <p>세 번째로 알람을 자동으로 종료하는 작업을 수행한다. 만약 알람이 켜질 수 없는 시간대 (알람은 정각에 울리므로 Current Time의 초 값이 5 이상인 경우) Mode의 alarm_buzzing을 false로 설정해주며, Alarm의 snooze는 평소 값으로 계속 지정한다. 반면에 알람이 울리는 5초 내에 버튼 입력이 들어오면 alarm_buzzing을 false로 설정하고 snooze를 false로 지정한다. 즉 이번 사이클에서 알람을 끈다.</p> <p>알람이 안 울리는 중인데 알람이 설정된 상태에서 알람 시각이 되며, snooze를 true로 설정해 알람을 켜면 된다. 알람이 설정되지 않은 경우에는 알람을 울리게 하면 안되므로 alarm_buzzing을 false로 둔다.</p> <p>가장 마지막으로 Current Time과 BacklightTime의 시, 분이 같은 경우 Backlight를 켜는다.</p>
--	--

3.2.3.2.7 Process 3.1

Reference Number	3.1
Name	Panel & Speaker Control
Input	Mode, Current Time, Time Difference, Alarm, Stopwatch, Backlight
Output	Panel Command, Speaker Command
Process Specification	Data store들의 값들을 읽어와 Panel에서 출력될 일곱 부분의 정보인 Panel Command와, 알람을 울릴지 말지에 대한 Speaker Command를 구성한다. 이후 이 두 정보를

	<p>Panel Interface와 Speaker Interface에 전달한다.</p> <p>Panel Command는 char 타입의 8*3 배열로 현재 화면에 어떤 값을 출력해야 하는지를 알려준다. 0-6번지에는 해당 구역에 출력해야 하는 정보를, 그리고 7번지에는 시간을 변경할 때 숫자가 깜빡일 구역을 저장한다.</p> <p>이때 데이터들은 패널에서 바로 출력될 수 있도록 가공되었다.</p> <p>각 모드별 화면 구성은 아래와 같다.</p> <p><Timekeeping 모드></p> <p>0 Day of Week(요일): 각 요일의 알파벳 앞 두 글자 출력</p> <p>1 Month</p> <p>2 Day</p> <p>3 Alarm Indicator 알람이 설정되었때 A 출력, 아니면 공백</p> <p>4 Hour</p> <p>5 Min</p> <p>6 Sec</p> <p>7 Year</p> <p><Stopwatch 모드></p> <p>0 "ST"</p> <p>1 Hour</p> <p>2 Min</p> <p>3 ""</p> <p>4 Stopwatch Min</p> <p>5 Stopwatch Sec</p> <p>6 Stopwatch MilliSec</p> <p>7 ""</p> <p><Alarm 모드></p> <p>0 "AL"</p> <p>1 Month</p> <p>2 Day</p> <p>3 Alarm Indicator 알람이 설정되었때 A 출력, 아니면 공백</p> <p>4 Alarm Hour</p> <p>5 Alarm Min</p> <p>6 ""</p>
--	--

	<p>7 ""</p> <p>Timekeeping 모드에서 사용자가 시간을 변경하고 있다면 변경 중인 시간 단위의 숫자를 깜빡여야 한다. 따라서 모드의 소분류를 읽어 현재 변경하고 있는 시간 단위를 알아내고, 이를 7번 행에 전달한다. 변경 중인 숫자는 1초에 한 번씩 깜빡인다.</p> <p>Speaker Command에서는 Current Time과 Alarm Time을 비교해 둘이 일치한다면 Speaker Command의 값을 On으로, 그렇지 않은 경우에는 Off로 전달한다.</p>
--	---

3.2.3.2.8 Process 3.2

Reference Number	3.2
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

Reference Number	3.3
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

Reference Number	4.1
Name	Backlight Control

Input	Backlight
Output	LED Command
Process Specification	Backlight data store 값에 들어있는 색으로 숫자의 색을 바꿔 LED Command로 내보낸다.

3.2.3.2.11 Process 4.2

Reference Number	4.2
Name	LED Interface
Input	LED Command
Output	Light
Process Specification	Light Command에 따라 알맞은 색으로 화면 글씨 색상을 설정한다.

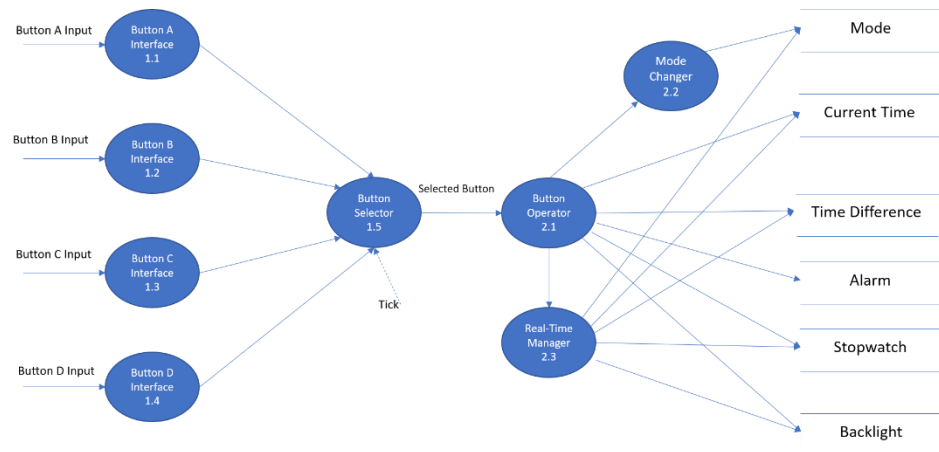
3.2.3.3 Data Dictionary

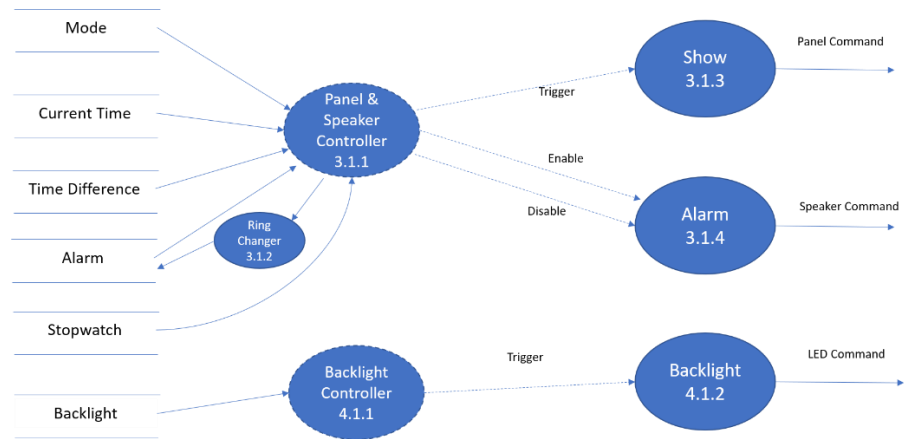
Input/Output Event	Description	Format / Type
Button A Input	DFD Level 1의 Data Dictionary 내용과 동일	Bool
Button B Input	DFD Level 1의 Data Dictionary 내용과 동일	Bool
Button C Input	DFD Level 1의 Data Dictionary 내용과 동일	Bool
Button D Input	DFD Level 1의 Data Dictionary 내용과 동일	Bool
Selected Button	DFD Level 1의 Data Dictionary 내용과 동일	int
Mode	DFD Level 1의 Data Dictionary 내용과 동일	struct Mode
Current Time	DFD Level 1의 Data Dictionary 내용과 동일	struct Time
Time Difference	DFD Level 1의 Data Dictionary 내용과 동일	struct Time
Alarm	DFD Level 1의 Data Dictionary 내용과 동일	struct Alm
Stopwatch	DFD Level 1의 Data	struct Stopwatch

	Dictionary 내용과 동일	
Backlight	DFD Level 1의 Data Dictionary 내용과 동일	struct Backlight
Panel Command	출력될 8개 부분에 대한 정보를 전달한다.	char[8][3]
Speaker Command	HW에 알람을 울려야 하는지(On) 말아야 하는지(Off)를 알린다.	On/Off
LED Command	백라이트 여부 (글씨 색상)를 전달한다.	On/Off
Display	패널에 현재 정보를 출력한다.	char[8][3]
Alarm	알람 울리기	On/Off
Light	Backlight 켜기 (숫자 색깔 바꾸기)	On/Off

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

Reference Number	1.1
Name	Button A Interface
Input	Button A input
Output	Button A input
Process Specification	Button A의 입력을 감지한다. 대문자 또는 소문자 A가 들어오면 true를, 그렇지 않으면 false를 반환한다.

3.2.4.2.2 Process 1.2

Reference Number	1.2
Name	Button B Interface
Input	Button B input
Output	Button B input
Process Specification	Button B의 입력을 감지한다. 대문자 또는 소문자 B가 들어오면 true를, 그렇지 않으면 false를 반환한다.

3.2.4.2.3 Process 1.3

Reference Number	1.3
Name	Button C Interface
Input	Button C input
Output	Button C input

Process Specification	Button C의 입력을 감지한다. 대문자 또는 소문자 C가 들어오면 true를, 그렇지 않으면 false를 반환한다.
-----------------------	--

3.2.4.2.4 Process 1.4

Reference Number	1.4
Name	Button D Interface
Input	Button D input
Output	Button D input
Process Specification	Button D의 입력을 감지한다. 대문자 또는 소문자 D가 들어오면 true를, 그렇지 않으면 false를 반환한다.

3.2.4.2.5 Process 1.5

Reference Number	1.5
Name	Button Selector
Input	Button A input, Button B input, Button C input, Button D input
Output	Selected Button
Process Specification	Button Selector는 대기하고 있다가 Button Input이 하나라도 들어오면 단위 시간을 시작하고 단위 시간 동안 추가로 Button Input을 받는다 (이때 처음 누른 버튼도 입력 받는 버튼들 중 하나로 취급한다). 그리고 단위 시간 동안 입력 받는 모든 input들 중 버튼 우선순위 $D > C > B > A$ 에 따라 단 하나의 Button Input만 Selected Button으로 선택해 리턴한다. Selected Button이 A, B, C, D인 경우 각각 1, 2, 3, 4를 내보내며, 대기 시간 동안 입력이 없는 경우에는 No Button, 즉 0을 반환한다.

3.2.4.2.6 Process 2.1

Reference Number	2.1
-------------------------	------------

Name	Button Operator
Input	Selected Button
Output	Mode to change, Current Time, Time Difference, Alarm, Stopwatch, Backlight
Process Specification	<p>Button Detect에서 내보낸 Selected Button값과, 모드를 input으로 가진다.</p> <p>Button Operator가 하는 일을 순서대로 나열하면 다음과 같다.</p> <p>1. Selected Button으로 어떤 동작을 수행해야 하는지를 결정한다.</p> <p>알람이 울리는 상태에서는 어떤 버튼을 눌러도 알람이 꺼지므로 알람부터 처리한다. 알람이 울리는 상태에서 버튼의 입력이 있으면 알람을 끄고 Alarm Time을 2019년 1월 1일 0시 0분 0초 00ms로 초기화한다.</p> <p>그리고 Mode의 대분류, 소분류, Selected Button 순서로 읽어가며 각 상황에 맞는 작업을 한다. Button Operator는 (대분류, 소분류)와 Selected Button에 대해 다음과 같이 반응한다.</p> <p>(1,1) timekeeping</p> <ul style="list-style-type: none"> a 모드를 (1,2)로 바꾼다 b 변경 사항 없음 c 모드를 (2,1)로 바꾼다 d Backlight 켜기 <p>(1,2) timekeeping_change_sec</p> <ul style="list-style-type: none"> a 모드를 (1,1)로 바꾼다 b 현재 시각 초 1 증가 c 모드를 (1,3)로 바꾼다 d Backlight 켜기 <p>(1,3) timekeeping_change_hour</p> <ul style="list-style-type: none"> a 모드를 (1,1)로 바꾼다 b 현재 시각 시간 1 증가 c 모드를 (1,4)로 바꾼다 d Backlight 켜기 <p>(1,4) timekeeping_change_min</p> <ul style="list-style-type: none"> a 모드를 (1,1)로 바꾼다 b 현재 시각 분 1 증가 c 모드를 (1,5)로 바꾼다

	<p>d Backlight 켜기</p> <p>(1,5) timekeeping_change_year</p> <p>a 모드를 (1,1)로 바꾼다</p> <p>b 현재 시각 년 1 증가</p> <p>c 모드를 (1,6)로 바꾼다</p> <p>d Backlight 켜기</p> <p>(1,6) timekeeping_change_month</p> <p>a 모드를 (1,1)로 바꾼다</p> <p>b 현재 시각 달 1 증가</p> <p>c 모드를 (1,7)로 바꾼다</p> <p>d Backlight 켜기</p> <p>(1,7) timekeeping_change_day</p> <p>a 모드를 (1,1)로 바꾼다</p> <p>b 현재 시각 일 1 증가</p> <p>c 모드를 (1,2)로 바꾼다</p> <p>d Backlight 켜기</p> <p>백라이트 시작 시각 = 현재 시각</p> <p>(2.1) stopwatch</p> <p>a 스톱워치가 안 켜진 경우</p> <p>스탑워치 구조체의 모든 항목들을 초기화한다.</p> <p>만약 스톱워치가 켜진 상태이면 stopwatchTime의 값을 initialTime에 저장하고 모드를 (2,2)로 변경한다.</p> <p>b 스톱워치가 안 켜진 경우:</p> <p>스탑워치를 켜다.</p> <p>그리고 stopwatchTime의 값을 전부 0으로, startTime에는 Current Time의 값을 저장한다.</p> <p>스탑워치가 켜진 경우:</p> <p>스탑워치를 끈다.</p> <p>stopwatchTime의 값을 initialTime에 저장한다.</p> <p>c 모드를 (3,1)로 바꾼다</p> <p>d Backlight 켜기</p> <p>(2,2) stopwatch_LAP</p> <p>a stopwatchTime의 값을 lapTime에 저장한다.</p> <p>b 모드를 (2,1)로 바꾼다</p>
--	--

	<p>c 모드를 (3,1)로 바꾼다 d Backlight 켜기</p> <p>(3.1) alarm</p> <p>a 모드를 (3,2)로 바꾼다 b 알람 설정을 켜거나 끈다. c 모드를 (1,1)로 바꾼다 d Backlight 켜기</p> <p>(3.2) alarm_change_hour</p> <p>a 모드를 (3,1)로 바꾼다 b 알람 시각 시간 1 증가 c 모드를 (3,3)로 바꾼다 d Backlight 켜기</p> <p>(3.3) alarm_change_minute</p> <p>a 모드를 (3,1)로 바꾼다 b 알람 시각 분 1 증가 c 모드를 (3,2)로 바꾼다 d Backlight 켜기</p> <p>Timekeeping 모드에서는 현재 시각을 변경할 때 스탑워치 시작 시각과 백라이트가 꺼져야 하는 시각도 함께 변경해, stopwatch로 측정한 시간과 백라이트가 꺼져야 하는 시각의 변동이 없도록 한다.</p> <p>한편 알람이 울릴 때를 제외하고 D를 누르면 백라이트가 켜지는데, 이 경우 Backlight 구조체의 value 값을 초록색으로, 백라이트가 꺼져야 하는 시각에는 현재 시각 Current Time에 2초를 더한 값을 저장한다.</p>
--	--

3.2.4.2.7 Process 2.2

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

3.2.4.2.8 Process 2.3

Reference Number	2.3
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, 그에 맞춰 스톱워치와 알람 시간을 관리한다.</p> <p>가장 먼저 Current Time을 동기화한다. DWS가 실행될 때마다 실제 세계의 현재 시각을 불러와 Time Difference와의 차이만큼 Current Time에 저장한다.</p> <p>이때 Current Time이 2099년이 초과하였는지를 체크해야 한다. 만약 자연스럽게 시간이 흘러 2100년이 된다면, 아래의 Current Time 보정이 일어나기 전에 Current Time의 YY에는 잠시동안 100 이상의 값이 저장된다. 보정이 끝나면 Current Time.YY에는 2019에서 2099 사이의 값이 저장되겠지만, Time Difference는 자연스럽게 바뀌는 과정이 없다. 따라서 자연스럽게 2099년을 초과하는 경우에는 Time Difference에서 YY -= 81을 해야 한다.</p> <p>Current Time이 2099년을 초과하지 않은 경우에는 Time Difference의 보정 없이 Current Time만 보정해 주면 된다.</p> <p>그 후 시각을 60진법에 맞게 보정한다. 이렇게 Current Time이 보정이 되면 stopwatch가 켜진 경우에 대해 보정된 Current Time을 바탕으로 stopwatchTime도 보정한다.</p> <p>세 번째로 알람을 자동으로 종료하는 작업을 수행한다. 만약 알람이 켜질 수 없는 시간대 (알람은 정각에 울리므로 Current Time의 초 값이 5 이상인 경우) Mode의 alarm_buzzing을 false로 설정해주며, Alarm의 snooze는 평소 값으로 계속 지정한다. 반면에 알람이 울리는 5초 내에 버튼 입력이 들어오</p>

	<p>면 alarm_buzzing을 false로 설정하고 snooze를 false로 지정한다. 즉 이번 사이클에서 알람을 끈다.</p> <p>알람이 안 울리는 중인데 알람이 설정된 상태에서 알람 시각이 되며, snooze를 true로 설정해 알람을 켜면 된다. 알람이 설정되지 않은 경우에는 알람을 울리게 하면 안되므로 alarm_buzzing을 false로 둔다.</p> <p>가장 마지막으로 Current Time과 BacklightTime의 시, 분이 같은 경우 Backlight를 끈다.</p>
--	--

3.2.4.2.9 Process 3.1.1

Reference Number	3.1.1
Name	Panel & Speaker Controller
Input	Mode, Current Time, Stopwatch Time, Lap Time
Output	Enable, Disable, Trigger of Commands
Process Specification	<p>Panel에서 출력될 일곱 부분의 정보인 Panel Command와, 알람을 울릴지 말지에 대한 Speaker Command를 구성한다. 그리고 Show를 Trigger, Alarm을 Enable/Disable 한다.</p> <p>패널은 8개의 구역으로 나뉘어 있으며, 각각의 모드별로 8개의 구역에 출력해야 하는 항목들이 다르다. 따라서 길이가 2인 문자열 8개의 배열로 각 상황에서 어떤 값을 출력해야하는지를 알려준다. 0-6번지에는 해당 구역에 출력해야 하는 정보를, 그리고 7번지에는 시간을 변경할 때 숫자가 깜빡일 구역을 저장한다.</p> <p>각각의 모드일 때 화면에 표시할 값은 아래와 같다.</p> <p><Timekeeping 모드></p> <p>0 Day of Week(요일): 각 요일의 알파벳 앞 두 글자 출력</p> <p>1 Mont</p> <p>2 Day</p>

	<p>3 Alarm Indicator 알람이 설정되었때 A 출력, 아니면 공백</p> <p>4 Hour</p> <p>5 Min</p> <p>6 Sec</p> <p>7 Year</p> <p><Stopwatch 모드></p> <p>0 "ST"</p> <p>1 Hour</p> <p>2 Min</p> <p>3 ""</p> <p>4 Stopwatch Min</p> <p>5 Stopwatch Sec</p> <p>6 Stopwatch MilliSec</p> <p>7 ""</p> <p><Alarm 모드></p> <p>0 "AL"</p> <p>1 Month</p> <p>2 Day</p> <p>3 Alarm Indicator 알람이 설정되었때 A 출력, 아니면 공백</p> <p>4 Alarm Hour</p> <p>5 Alarm Min</p> <p>6 ""</p> <p>7 ""</p> <p>Timekeeping 모드에서 사용자가 시간을 변경하고 있다면 변경 중인 시간 단위의 숫자를 깜빡여야 한다. 따라서 모드의 소분류를 읽어 현재 변경하고 있는 시간 단위를 알아내고, 이를 7번 행에 전달한다. 변경 중인 숫자는 1 초에 한 번씩 깜빡인다.</p> <p>Speaker Command에서는 Current Time과 Alarm Time을 비교해 둘이 일치한다면 Speaker Command의 값을 On으로, 그렇지 않은 경우에는 Off로 전달한다.</p>
--	--

3.2.4.2.10 Process 3.1.2

Reference Number	3.1.2
Name	Ring Changer
Input	Alarm
Output	Alarm
Process Specification	AL의 ring 값을 증가시켜 알람 소리를 일정하게 내는 역할을 한다.

3.2.4.2.11 Process 3.1.3

Reference Number	3.1.3
Name	Show
Input	Trigger
Output	Panel Command
Process Specification	<p>Trigger되면 Panel & Speaker Controller에서의 결과에 따라 화면 구성을 한다. 패널은 8개의 구역으로 나뉘어 있으며, 각각의 모드별로 화면의 8개의 구역에 출력해야 하는 항목들은 아래와 같다.</p> <p><Timekeeping 모드></p> <p>0 Day of Week(요일): 각 요일의 알파벳 앞 두 글자 출력</p> <p>1 Month</p> <p>2 Day</p> <p>3 Alarm Indicator 알람이 설정되었때 A 출력, 아니면 공백</p> <p>4 Hour</p> <p>5 Min</p> <p>6 Sec</p> <p>7 Year</p> <p><Stopwatch 모드></p> <p>0 "ST"</p> <p>1 Hour</p> <p>2 Min</p> <p>3 ""</p> <p>4 Stopwatch Min</p> <p>5 Stopwatch Sec</p> <p>6 Stopwatch MilliSec</p> <p>7 ""</p>

	<p><Alarm 모드></p> <p>0 "AL"</p> <p>1 Month</p> <p>2 Day</p> <p>3 Alarm Indicator 알람이 설정되었때 A 출력, 아니면 공백</p> <p>4 Alarm Hour</p> <p>5 Alarm Min</p> <p>6 ""</p> <p>7 ""</p> <p>Timekeeping 모드에서 사용자가 시간을 변경하는 중에는 panel command의 7번 인덱스에 저장된 위치에 있는 숫자를 깜빡인다. 변경 중인 숫자는 1초에 한 번씩 깜빡인다.</p>
--	---

3.2.4.2.12 Process 3.1.4

Reference Number	3.1.4
Name	Alarm
Input	Enable, Disable
Output	Speaker Command
Process Specification	입력 값에 따라, 스피커를 켜거나 끈다. 즉, 알람을 울리거나 끈다.

3.2.4.2.13 Process 4.1.1

Reference Number	4.1.1
Name	Backlight Controller
Input	Backlight
Output	Trigger
Process Specification	Backlight data store 값에 들어있는 색으로 숫자의 색을 바꿔 LED Command로 내보낸다.

3.2.4.2.14 Process 4.1.2

Reference Number	4.1.2
Name	Light

Input	Trigger
Output	LED Command
Process Specification	LED Command의 값에 의해 정해진 색으로 화면을 나타낸다.

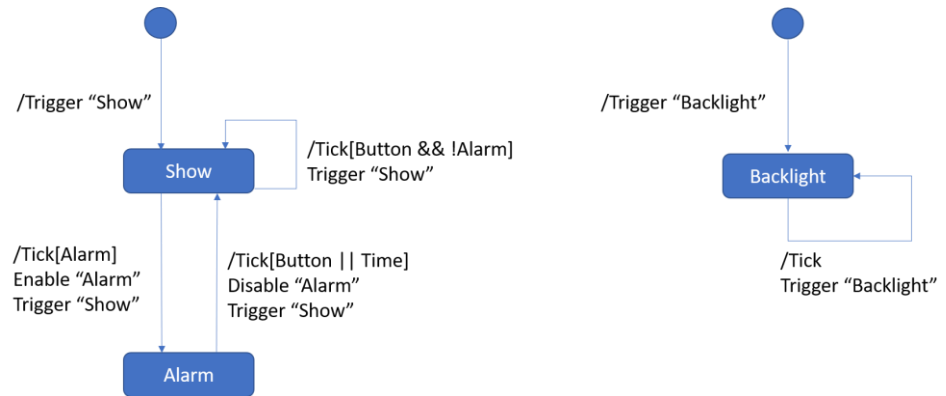
3.2.4.3 Data Dictionary

Input/Output Event	Description	Format / Type
Button A Input	DFD Level 1의 Data Dictionary 내용과 동일	Bool
Button B Input	DFD Level 1의 Data Dictionary 내용과 동일	Bool
Button C Input	DFD Level 1의 Data Dictionary 내용과 동일	Bool
Button D Input	DFD Level 1의 Data Dictionary 내용과 동일	Bool
Selected Button	DFD Level 1의 Data Dictionary 내용과 동일	int
Mode	DFD Level 1의 Data Dictionary 내용과 동일	struct Mode
Current Time	DFD Level 1의 Data Dictionary 내용과 동일	struct Time
Time Difference	DFD Level 1의 Data Dictionary 내용과 동일	struct Time
Alarm	DFD Level 1의 Data Dictionary 내용과 동일	struct Alm
Stopwatch	DFD Level 1의 Data Dictionary 내용과 동일	struct Stopwatch
Backlight	DFD Level 1의 Data Dictionary 내용과 동일	struct Backlight
Mode to change	모드 값을 들어온 값으로 바꾼다	struct Mode
Panel Command	DFD Level 2의 Data Dictionary 내용과 동일	char[8][3]
Speaker Command	DFD Level 2의 Data Dictionary 내용과 동일	On/Off
LED Command	DFD Level 2의 Data Dictionary	On/Off

	내용과 동일	
--	--------	--

3.2.5 DFD Level 4

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



3.2.6 Overall DFD

