

# Unit Testing Plan for Digital Watch System

- Test Plan
- Test Design Specification
- Test Cases Specification

Project Team

**이동아**

Date

**2012-11-09**

---

**Team Information**

이동아: dalee.dslab@gmail.com

## Table of Contents

1	Introduction .....	4
1.1	Objectives.....	4
1.2	Background .....	4
1.3	Scope.....	4
1.4	Project plan .....	4
1.5	Configuration management plan.....	4
1.6	References.....	4
2	Test items .....	4
3	Approach.....	6
4	Item pass/fail criteria .....	6
5	Unit test design specification.....	6
5.1	Test design specification identifier .....	6
5.2	Features to be tested .....	6
5.3	Approach refinements.....	7
5.4	Test identification .....	7
5.5	Feature pass/fail criteria .....	7
6	Unit test case specification.....	7
6.1	Test case specification identifier .....	7
6.2	Test items .....	<b>오류! 책갈피가 정의되어 있지 않습니다.</b>
6.3	Input specifications.....	7
6.4	Output specifications.....	7

7	Testing tasks .....	7
8	Environmental needs .....	7
9	Unit Test deliverables .....	8
10	Schedules .....	8

## 1 Introduction

### 1.1 Objectives

본 문서는 2012년 2학기 SE 수업의 T3가 개발한 두 번째 버전의 Digital Watch System (DWS)을 Unit Testing하기 위한 계획문서이다. T3가 정의한 Unit을 Testing을 수행하기 위하여 Testing Pass/Fail Criteria를 정의하고 이를 수행하기 위한 Test design & test cases를 제작한다.

### 1.2 Background

2012년 2학기 SE수업에서 개발하는 모든 DWS는 SASD기법을 이용하여 개발된다. 기능 별로 나뉜 Unit은 SRA, SDS 문서에 모두 정의되어 있다.

### 1.3 Scope

SRA 및 SDS 문서에 정의된 Unit을 Testing한다.

### 1.4 Project plan

### 1.5 Configuration management plan

### 1.6 References

T2-2012.DWS.SRA-3.4

T2-2012.DWS.SDS-2.1

## 2 Test items

T2가 SASD기법을 이용하여 개발한 DWS을 Testing한다. SA와 SD에서 분류한 각 process/module 별로 Testing을 수행한다. <Figure 1 Overall DFD>은 SA를 이용하여 요구사항을 분석한 결과 중 전체 DFD를 나타낸 그림이며, <Figure 2 Structured Chart오류! 참조 원본을 찾을 수 없습니다.>은 SD의 Basic Structural Chart를 나타낸 그림이다. 각 그림을 참조하여 Unit을 지정하고, 지정한 Unit을 SRA에 명세 된 내용과 같은 동작을 하는지 확인한다.

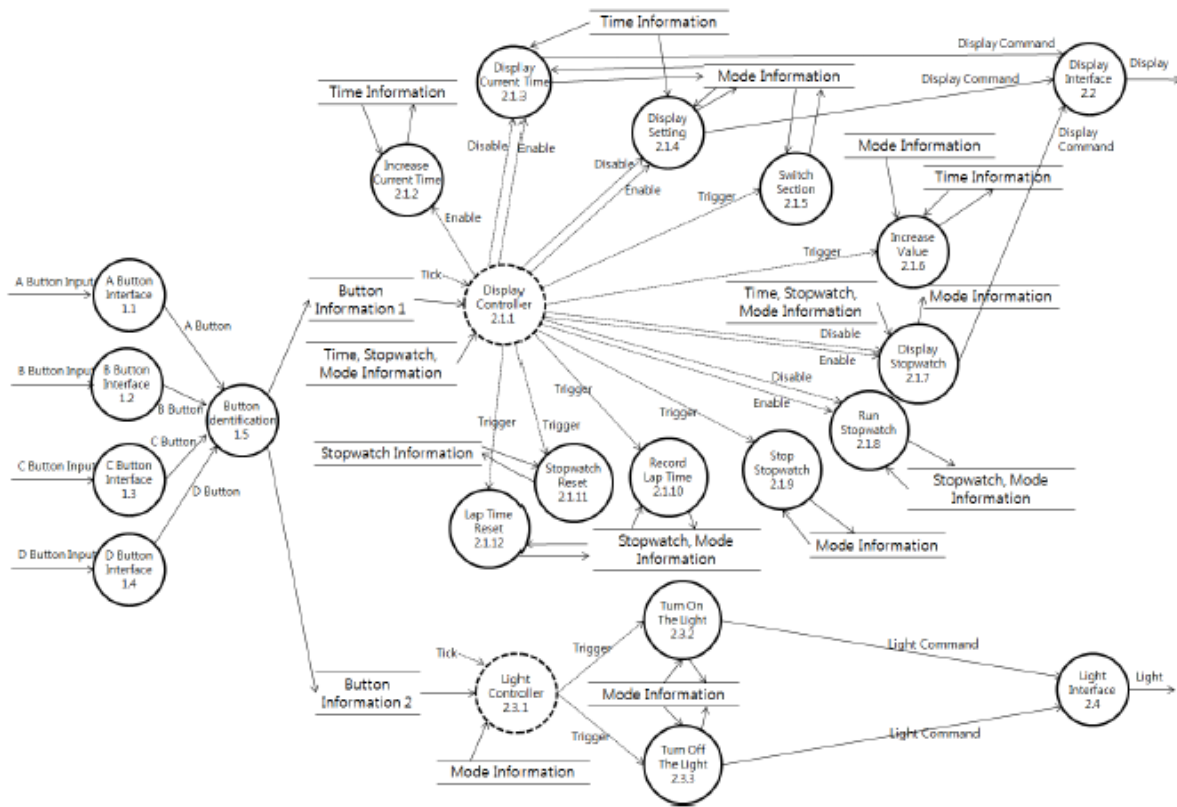


Figure 1 Overall DFD

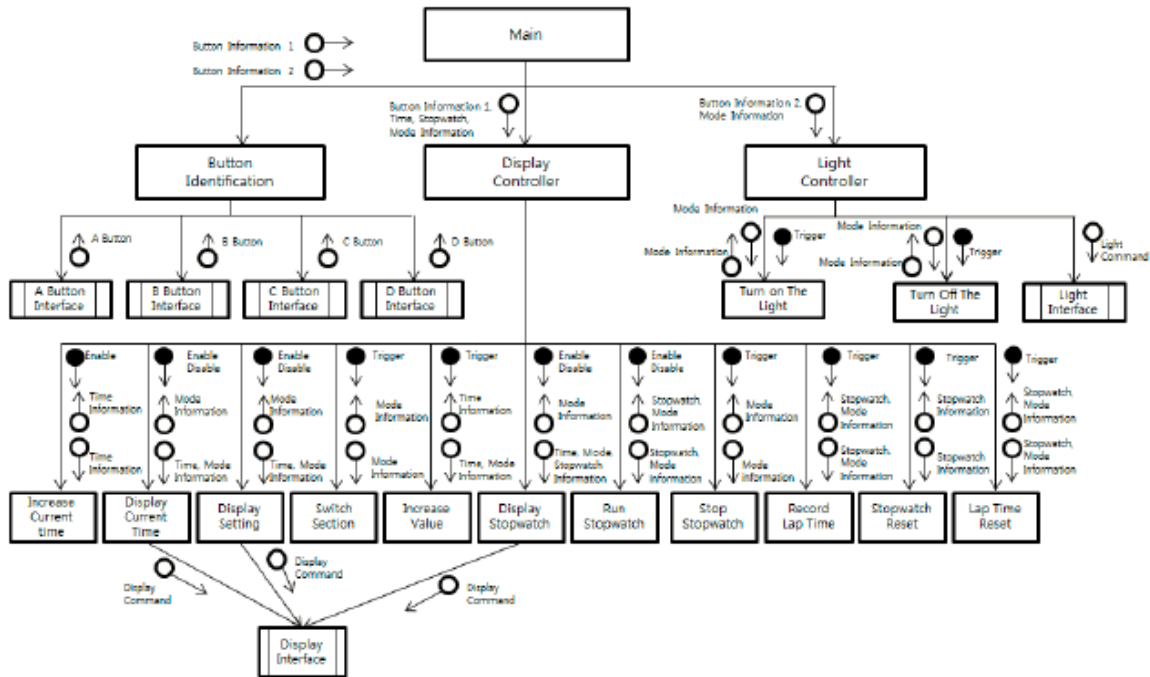


Figure 2 Structured Chart

3 Approach

Brute Force, SRA의 가장 하위 Level의 Process를 대상으로 Unit Test를 수행한다.

4 Item pass/fail criteria

모든 Units은 SRA에 명시된 Process Specification과 동일한 결과를 출력해야 한다. 각 Unit별 Pass/Fail Criteria는 <Table 2 Test Design Specification>를 참조한다.

5 Unit test design specification

5.1 Test design specification identifier

DWS.UTC.Number

5.2 Features to be tested

Process 1.1 – 1.5

Process 2.2, 2.4

Process 2.1.1-2.1.11

Process 2.3.1-2.3.3

5.3 Approach refinements

각 Process Specification에 명시된 내용을 기반으로 Test Design 및 Test Cases를 생성해 낸다.

5.4 Test identification

*DWS.UTC.Number*

5.5 Feature pass/fail criteria

6 Unit test case specification

6.1 Test case specification identifier

*DWS.UTC.Number.Number*

6.2 Input specifications

6.3 Output specifications

7 Testing tasks

**Table 1 Test tasks & schedule**

Task	Predecessor tasks	Special skills	Effort	Finish date
(1) Unit Test Plan 작성	SRA 작성 SDS 작성 DWS 구현		3	
(2) Test design specification	Task 1	DWS 에 대한 이해	5	
(3) Test case specification	Task 2	DWS 에 대한 이해	5	
(4) Test execution	Task 3	Test code 작성 Test tools 에 대한 이해	4	
(5) Test result report	Task 4		1	
(6) 개발팀에 report 전달	Task 5		1	

8 Environmental needs

Eclipse IDE for C/C++ developers

Visual SVN server

Hudson CI

9 Unit Test deliverables

Unit Test Report

10 Schedules

<Table 1 Test tasks & schedule> 참조



Table 2 Test Design Specification

Identifier	Feature	Pass/Fail
DWS.UTC.110	A가 눌리면 A Button에 1을 출력한다.	출력 확인
DWS.UTC.120	B가 눌리면 B Button에 1을 출력한다.	출력 확인
DWS.UTC.130	C가 눌리면 C Button에 1을 출력한다.	출력 확인
DWS.UTC.140	D가 눌리면 D Button에 1을 출력한다.	출력 확인
DWS.UTC.150	A, B, C button 입력을 받아 button information 1에 1(A), 2(B), 3(C)를 저장 D button 입력을 받아 Button Information 2에 저장	A,B,C,D에 따른 출력 (Button Info. 1, Button Info. 2) 확인
DWS.UTC.220	Display commands를 입력 받아서 화면에 출력함.	출력 확인
DWS.UTC.240	Light Command를 받을 때 Light를 출력함	출력 확인
DWS.UTC.211	STD의 동작 확인.	
DWS.UTC.212	Tick이 100번째 들어올 때마다 Time info.의 초를 1 증가시킨다. 초 증가에 따라 년월일시분 도 같이 증가	시간 증가 확인
DWS.UTC.213	Time info.를 Display command에 출력 M_TS == 1 일 때 TS_second =1; TS_others = 0 M_TK = 1; M_STW & M_TS = 0	출력 확인 초기화 확인
DWS.UTC.214	Current time을 Display command에 출력 Selected section에 따라 밑줄 출력 Mode info.의 정보를 이용하면 section 값 알 수 있음 M_TS = 1; M_STW & M_TK = 0	출력 확인 Section 확인 변수 값 확인
DWS.UTC.215	TS_ 의 값들을 순차적으로 변경시킴	변수 값 변경 확인

	TS_second == 1 → TS_hour == 1 → TS_minute == 1 → TS_year == 1 → TS_month == 1 → TS_day == 1 → TS_second == 1	
DWS.UTC.216	TS_ == 1에 해당하는 위치의 값을 +1 시킴	시간 값 변경 확인
DWS.UTC.217	LAP_ON == 0이면 측정 중인 시간 Display Command로 출력 LAP_ON == 1이면 Laptime에 저장된 시간 Display Command로 출력  시, 분 값 출력 M_STW = 1; M_TK & M_TS = 0	LAP_ON 에 따른 출력 값 확인  변수 값 확인
DWS.UTC.218	Stopwatch time의 cs 값 증가, R_STW = 1	시간 증가 확인, 변수 값 확인
DWS.UTC.219	R_STW = 0	변수 값 확인
DWS.UTC.21A	Laptime에 stopwatch time 값 저장	Laptime 값 확인
DWS.UTC.21B	Stopwatch time = 0	시간 값 확인
DWS.UTC.231	STD 동작 확인	
DWS.UTC.232	LIGHT_STATE = 1	변수 값 확인
DWS.UTC.233	LIGHT_STATE = 0	변수 값 확인

Identifier	Input	Output	비고
DWS.UTC.110.00	System function 호출	PASS	
DWS.UTC.120.00	System function 호출	PASS	
DWS.UTC.130.00	System function 호출	PASS	
DWS.UTC.140.00	System function 호출	PASS	
DWS.UTC.150.00	실제 Process spec에 포함 된 내용이 분산되어 구현 됨	FAIL	
DWS.UTC.220.00	System function 호출	PASS	
DWS.UTC.240.00	System function 호출	PASS	
DWS.UTC.211.00	State를 찾을 수 없음, TCs 생성 불가	FAIL	
DWS.UTC.212.00	100회 호출 Time = 2012-11-23 11:34:56	Time = 2012-11-23 11:34:57	<code>void IncreaseCurrentTime(TIME *a)</code>
DWS.UTC.212.01	100회 호출 Time = 2012-11-23 11:34:59	Time = 2012-11-23 11:35:00	
DWS.UTC.212.02	100회 호출 Time = 2012-11-23 11:59:59	Time = 2012-11-23 12:00:00	
DWS.UTC.212.03	100회 호출 Time = 2012-11-23 23:59:59	Time = 2012-11-24 00:00:00	
DWS.UTC.212.04	100회 호출 Time = 2012-11-30 23:59:59	Time = 2012-12-01 00:00:00	
DWS.UTC.212.05	100회 호출 Time = 2012-02-28 23:59:59	Time = 2012-02-29 00:00:00	
DWS.UTC.212.06	100회 호출 Time = 2012-02-29 23:59:59	Time = 2012-03-01 00:00:00	
DWS.UTC.212.07	100회 호출 Time = 2013-02-28 23:59:59	Time = 2013-03-01 00:00:00	
DWS.UTC.213.00	MODE.M_TS == 1	MODE.(TS_year == 0 && TS_month == 0 && TS_day == 0 && TS_hour == 0 && TS_minute == 0 && TS_second == 1) MODE.(M_TK == 1 && M_STW == 0, M_TS == 0)	<code>void DisplayCurrentTime(TIME *time, MODE *mode)</code>

DWS.UTC.213.01	출력 값 확인 통과	PASS	
DWS.UTC.214.00		MODE.( M_TS == 1 && M_STW == 0 && M_TK == 0)	void DisplaySetting(TIME *time, MODE *mode)
DWS.UTC.215.00	1회 호출	MODE.( TS_second == 1)	void SwitchSection(MODE *md)
DWS.UTC.215.01	7회 호출	MODE.( TS_second == 1)	
DWS.UTC.216.00	TS_second == 1 && Time Data = 20-11-22 23:59:00	Time Data = 20-11-22 23:59:01	void IncreaseValue(MODE *m, TIME *t)
DWS.UTC.216.01	TS_second == 1 && Time Data = 20-11-22 23:59:59	Time Data = 20-11-22 23:59:00	
DWS.UTC.216.02	TS_hour == 1 && Time Data = 20-11-22 20:00:00	Time Data = 20-11-22 21:00:00	
DWS.UTC.216.03	TS_hour == 1 && Time Data = 20-11-22 23:00:00	Time Data = 20-11-22 00:00:00	
DWS.UTC.216.04	TS_minute == 1 && Time Data = 20-11-22 23:00:00	Time Data = 20-11-22 23:01:00	
DWS.UTC.216.05	TS_minute == 1 && Time Data = 20-11-22 23:59:00	Time Data = 20-11-22 23:00:00	
DWS.UTC.216.06	TS_year == 1 && Time Data = 20-11-22 23:00:00	Time Data = 21-11-22 23:00:00	
DWS.UTC.216.07	TS_year == 1 && Time Data = 99-11-22 23:00:00	Time Data = 12-11-22 23:00:00	
DWS.UTC.216.08	TS_month == 1 && Time Data = 20-11-22 23:00:00	Time Data = 20-12-22 23:00:00	
DWS.UTC.216.09	TS_month == 1 && Time Data = 20-12-22 23:00:00	Time Data = 20-01-22 23:00:00	
DWS.UTC.216.10	TS_day == 1 && Time Data = 20-12-22 23:00:00	Time Data = 20-12-23 23:00:00	
DWS.UTC.216.11	TS_day == 1 && Time Data = 20-12-31 23:00:00	Time Data = 20-12-01 23:00:00	
DWS.UTC.216.12	TS_day == 1 && Time Data = 20-02-28 23:00:00	Time Data = 20-02-01 23:00:00	
DWS.UTC.216.13	TS_day == 1 && Time Data = 21-02-28 23:00:00	Time Data = 21-02-29 23:00:00	
DWS.UTC.216.14	TS_day == 1 && Time Data = 21-02-29 23:00:00	Time Data = 21-02-01 23:00:00	
DWS.UTC.217.00		M_STW == 1 && M_TK == 0 && M_TS == 0	void DisplayStopwatch(TIME *time, STOPWATCH *stopwatch, MODE *mode)
DWS.UTC.218.00	STOPWATCH == 12:34:78	STOPWATCH == 12:34:79 &&	void RunStopwatch(STOPWATCH

		MODE.R_STW == 1	*sw, MODE *md)
DWS.UTC.218.01	STOPWATCH == 12:34:99	STOPWATCH == 12:35:00 && MODE.R_STW == 1	
DWS.UTC.218.02	STOPWATCH == 12:59:99	STOPWATCH == 13:00:00 && MODE.R_STW == 1	
DWS.UTC.219.00		R_STW == 0	void StopStopwatch(MODE *md)
DWS.UTC.21A.00	STOPWATCH == 12:34:56	LAPTIME == 12:34:56 && LAP_ON == 1	void RecordLapTime(STOPWATCH *sw, MODE *md)
DWS.UTC.21B.00		STOPWATCH == 00:00:00 && LAPTIME == 00:00:00	void StopwatchReset(STOPWATCH *sw)
DWS.UTC.231.00	State 정보를 찾을 수 없음	FAIL	void LightControl(int Button, MODE* md)
DWS.UTC.232.00		LIGHT_STATE == 1	void TurnOnTheLight(MODE *md)
DWS.UTC.233.00		LIGHT_STATE == 0	void TurnOffTheLight(MODE *md)