

Unit Testing Plan for Digital Watch System

- Test Plan
- Test Design Specification
- Test Cases Specification

Date

2012-10-28

Team Information

Sanghyun Yoon shyoon.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.....	5
2	Test items	5
3	Features to be tested.....	7
4	Features not to be tested	7
5	Approach.....	9
6	Item pass/fail criteria	9
7	Unit test design specification.....	9
7.1	Test design specification identifier	9
7.2	Features to be tested	10
7.2.1	Processes in SRA.....	10
7.3	Approach refinements.....	10
7.3.1	Brute force testing.....	10
7.4	Test identification	10
7.5	Feature pass/fail criteria	10
8	Unit test case specification.....	10

8.1	Test case specification identifier	10
8.2	Test items	10
8.3	Input specifications.....	11
8.4	Output specifications.....	11
9	Testing tasks	11
10	Environmental needs	11
11	Unit Test deliverables.....	11
11.1	Unit test plan.....	11
11.2	Unit test design specification	12
11.3	Unit test case specification	12
11.4	Unit test summary report.....	12
12	Schedules	12

1 Introduction

1.1 Objectives

본 문서는 digital watch system의 unit test를 수행하기 위한 계획 문서이며 다음과 같은 목적을 갖는다.

- (1) Digital watch system의 unit test를 수행하기 위해 필요한 활동 및 자원을 정의한다.
- (2) Digital watch system의 unit test를 수행하기 위한 test approach 및 techniques을 정의한다.
- (3) Digital watch system의 unit test를 수행하기 위한 환경적인 요구사항 및 test 도구들을 정의한다.

1.2 Background

Digital watch system은 현재 시각을 알려주는 기본 기능 외에도 알람, 스톱워치 등의 다양한 기능을 가지고 있는 시계이다. 이 시스템은 여러 개의 입력을 가지고 있으며 실행 시간, 반응 시간 등도 시스템의 성능을 좌우하는 중요한 feature들이다. Unit test는 시스템을 구성하는 최소 단위 모듈들을 대상으로 하는 test이며, 시스템의 성능을 좌우하는 feature들이 요구사항을 만족하는지를 확인할 수 있는 기본적인 test approach이다.

1.3 Scope

이 계획 문서는 digital watch system의 unit test를 수행하기 위한 모든 것을 포함한다. Digital watch system의 unit test를 수행하기 위한 자원과 절차, test approach와 technique과 필요로 하는 환경 및 도구 등을 정의한다. Digital watch system의 unit test는 시스템을 구성하는 최소 단위의 모듈들을 대상으로 하며, 구현된 모듈이 요구사항을 만족하는지를 test한다.

1.4 Project plan

1.5 Configuration management plan

Digital watch system의 program source code 및 unit test를 위한 test code는 CTIP (Continuous Testing & Integration Platform) 환경에서 이루어지며, program source code/test code의 변경 및 수정 사항은 지속적으로 통합되고 test된다.

- (1) Program source code의 변경

Program source code에 변경 및 수정 발생 시, 이를 통합하고 수동적으로 unit test를

수행한다.

(2) 일정 주기

CI server에 의해 관리되는 program source code는 일정 주기를 가지고 자동적으로 build 및 unit test를 수행한다.

1.6 References

Test plan은 다음 문서들을 기반으로 작성되었으며, unit test 수행을 위한 자원으로 사용된다.

DS.2012.DWS.SRS-2.0

T2_SRA

T2_SDS

IEEE 829-1998 Standard for Software Test Documentation

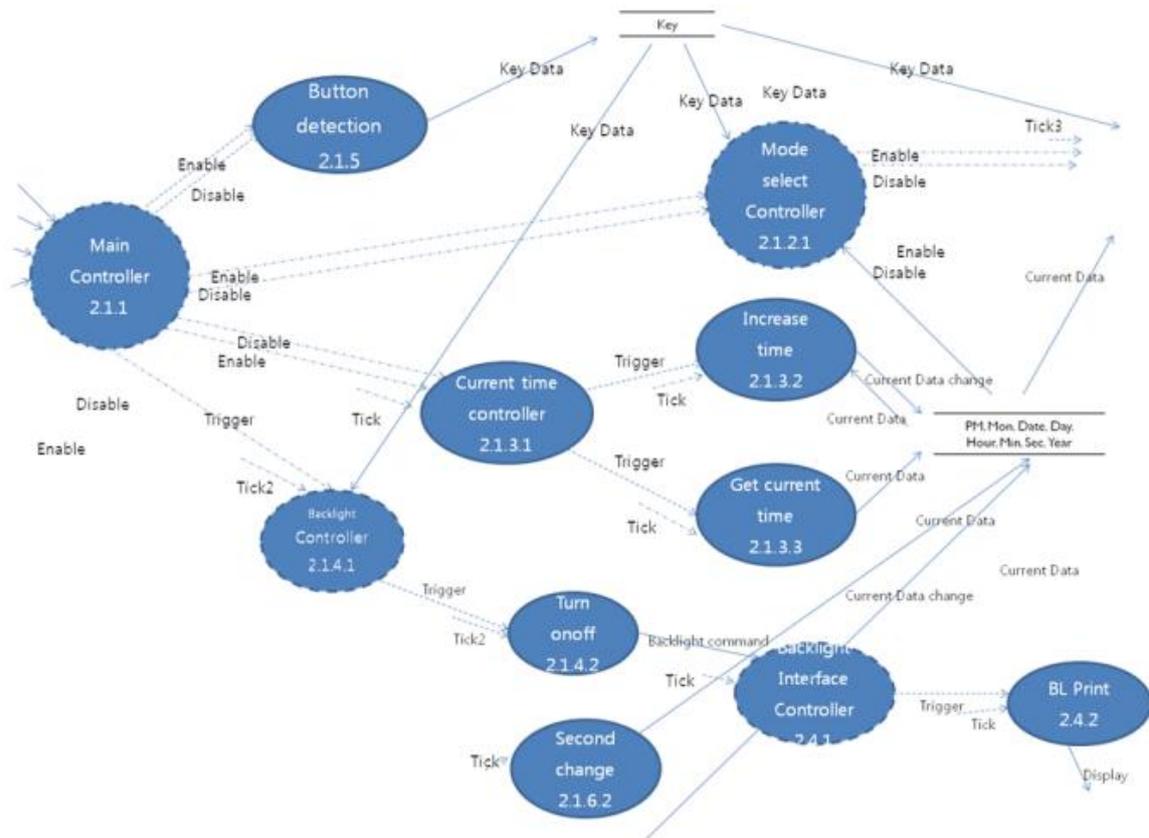
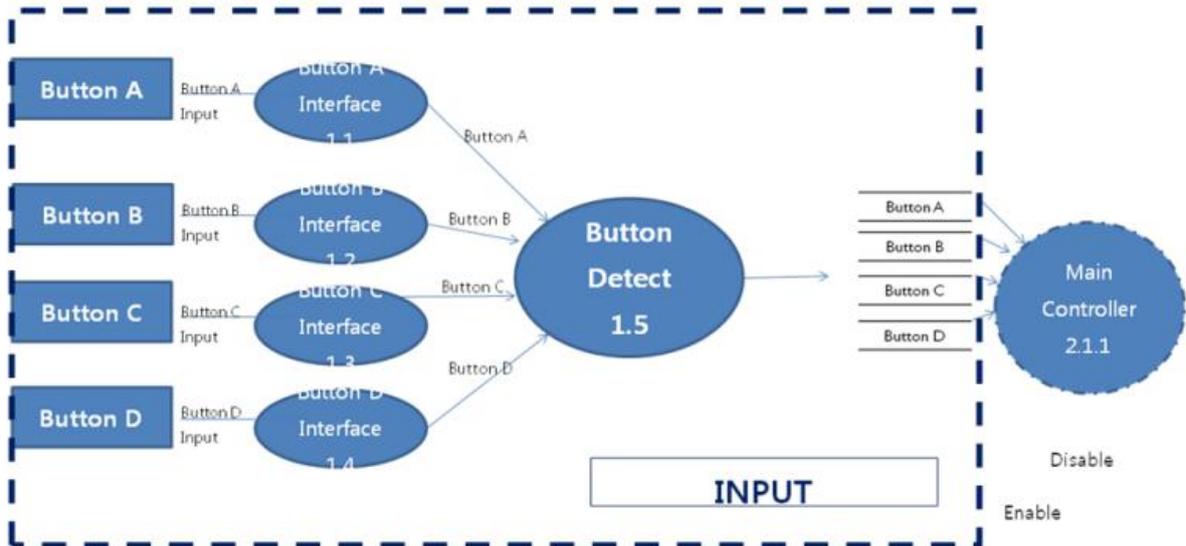
IEEE 1012-2012 Standard for System and Software Verification and Validation

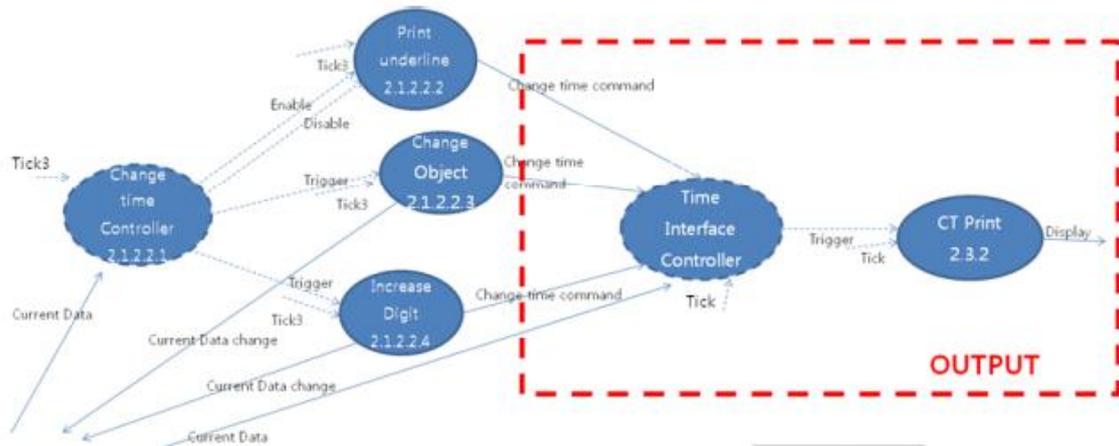
2 Test items

Digital watch system을 구성하는 최소 단위의 모듈들이 unit test의 대상이 된다. 각 모듈의 요구 사항을 만족하는지를 test하며, test item은 다음 자료들로부터 작성되었다.

(1) Functionality of modules – T2_SRA: Process specification, DFD

아래 그림은 그 일부를 참조한 것임.





(2) Module interface – T2_SRA: Process specification, Structure chart

3 Features to be tested

- (1) Processes in SRA: 각 프로세스가 가지고 있는 요구사항을 만족하는지를 test한다
- (2) Modules in SDS: 각 모듈이 가지고 있는 데이터 인터페이스를 test한다. <Table 1 테스트할 Process (DFD) 리스트>의 Process name 참조

Table 1 테스트할 Process (DFD) 리스트

ID	Name	Description

4 Features not to be tested

- (1) Processes in SRA: 외부 장치 드라이버, 단순 데이터 전달 프로세스 등은 test에서 제외한다. 또는 SRA, SDS 문서와 다르게 실제구현에서는 unit으로 나누어 있지 않은 프로세스를 제외한다.
- (2) Modules in SDS: <Table 2 테스트하지 않을 Process (DFD) 리스트>의 Process name 참조

Table 2 테스트하지 않을 Process (DFD) 리스트

ID	Name	Description
1.1	Button A Interface	아날로그 신호 버튼 A 를 받아 디지털 신호 Button A 로 바꿔 전달한다.
1.2	Button B Interface	아날로그 신호 버튼 B 를 받아 디지털 신호 Button B 로 바꿔 전달한다.
1.3	Button C Interface	아날로그 신호 버튼 C 를 받아 디지털 신호 Button C 로 바꿔 전달한다.
1.4	Button D Interface	아날로그 신호 버튼 D 를 받아 디지털 신호 Button D 로

		바뀌 전달한다.
1.5	Button Type Detect	입력된 버튼정보를 컨트롤 프로세스로 전달한다.
2.1.3.3	Get current time	시스템에서 받아온 현재시간 값을 각 Output 변수에 저장
2.2.1	TimeKeep interface controller	Timekeep command 의 실행결과를 출력하는 Trigger 를 발생한다.
2.2.2	TK Print	Timekeep command 의 실행결과를 출력하는 Trigger 를 입력 받아 결과를 화면에 출력한다.
2.3.1	Change time interface controller	Change time command 의 실행결과를 출력하는 Trigger 를 발생한다.
2.3.2	CT Print	Change time command 의 실행결과를 출력하는 Trigger 를 입력 받아 결과를 화면에 출력한다.
2.1.2.3.7	Stop	Process spec 없음
2.4.1	Back light interface controller	Back light command 의 실행결과를 출력하는 Trigger 를 발생한다.
2.4.2	BL Print	Backlight command 의 실행결과를 출력하는 Trigger 를 입력 받아 결과를 화면에 출력한다.
2.5.1	Stopwatch interface controller	Stopwatch command 의 실행결과를 출력하는 Trigger 를 발생한다.
2.5.2	SW print	Stopwatch 의 실행결과를 출력하는 트리거를 입력 받아 각 스탑워치 측정 시간을 화면에 출력한다.
2.1.5	Button detection	입력 받은 버튼의 우선순위를 결정하여 하나의 버튼데이터만 Key 데이터에 저장한다
	Time Interface controller	Process id 없음, process spec 없음
	Stopwatch	Process id 없음, process spec 없음
	Increase time	Process id 없음, process spec 없음
2.1.2.2.6	Lap time	스톱워치가 진행되는 동안 순간 기록을 보여주는 프로세스
2.1.6.2	Second change	60 초에 도달하면 초를 0 으로 초기화 분을 1 분 증가해주는 프로세스
	Minute change	Process id 없음, process spec 없음
2.1.1	Main controller	<p>입력된 버튼 정보에 따라 2 가지 모드(Stop watch mode, Change time mode)중 하나를 선택하는 Mode select process 를 실행한다.</p> <p>Time keep 을 실행한다.</p> <p>Button detection 버튼의 우선순위를 결정하는 process 실행한다.</p> <p>항상 Get current time process 를 통해 현재 시간을 업데이트 한다.</p> <p>Backlight process 로 백라이트 기능을 수행한다.</p> <p>구현 된 부분에서 일치하는 부분 찾기 어려움. Main.c 의</p>

		main()으로 추측되기는 하나, DFD 에는 2.1.1 에 key 값이 들어가지 않는데 main()에는 key 값이 사용되고 있음.
2.1.2.1	Mode Select Controller	프로세스들(Change time process, Stop watch process) 중, 입력 버튼에 해당하는 프로세스를 실행시킨다. 중요한 역할을 하나 문서와 상이. Report 의 evaluation part 참조.
2.1.3.1	Current time controller	현재 시간 출력에 관한 프로세스를 관장하는 컨트롤러 DFD 와 다르게 너무 많은 곳에서 호출, 초 분 시 등 시간
2.1.3.2	Increase time	진행되는 시간을 표현하기 위해 초 값을 증가시키는 프로세스. 계산을 컴퓨터 시간을 가져와서 리셋시키는 것으로 보임.
2.1.4.1	Backlight Controller	라이트를 켜주는 것과 관련된 프로세스를 관장하는 컨트롤러. Light_status flag 를 이용하여 켜짐, 꺼짐 상태에 맞는 행동을 취한다.
2.1.4.2	Turn onoff	라이트를 켜주거나 꺼주는 프로세스. 실제론 색을 인자로 받아 해당색의 문자를 출력하게끔 한다.
2.1.6.2	Second Change	60 초에 도달하면 초를 0 으로 초기화 분을 1 분 증가해주는 프로세스
2.1.2.2.1	Change time Controller	현재 시간을 원하는 값으로 변경해주는 프로세스
2.1.2.3.1	Stopwatch Controller	스톱워치 기능을 담당하는 컨트롤러

5 Approach

Digital watch system의 program source code 및 unit test를 위한 test code는 CTIP (Continuous Testing & Integration Platform) 환경에서 이루어지며, program source code/test code 의 변경 및 수정 사항은 지속적으로 통합되고 test된다.

(1) Brute force testing: 각 모듈의 요구사항을 만족하는지를 확인할 수 있는 test case를 작성한다. 그 이외의 예외상황에 대해서는 test하지 않는다.

6 Item pass/fail criteria

Functional test pass/fail criteria: 각 모듈은 요구사항을 모두 만족하여야 한다.

7 Unit test design specification

7.1 Test design specification identifier

DWS.UTD.00.00

7.2 Features to be tested

7.2.1 Processes in SRA

<Table 1 테스트할 Process (DFD) 리스트> 참조

7.3 Approach refinements

7.3.1 Brute force testing

DWS의 각 모듈이 요구사항을 만족하는지를 확인하기 위하여, 요구사항에 정의된 내용에 기반하여 test case를 작성한다. 그 이외의 예외 상황에 대해서는 test case를 작성하지 않는다.

7.4 Test identification

Table 3 Test Design Identification

Identifier	Feature (Process ID in DFD)	Valid/Invalid value

7.5 Feature pass/fail criteria

DWS의 각 모듈(프로세스)은 SRA에 정의되어 있는 요구사항 (입력 / 출력 및 동작) 을 모두 만족해야 한다. 각 모듈(프로세스)의 입력 / 출력 및 동작은 SRA의 process description 항목을 참조한다.

8 Unit test case specification

8.1 Test case specification identifier

Table 4 Test Case Identification

--	--	--

8.2 Test items

<Table 3 Test Design Identification> 참조

8.3 Input specifications

<Table 4 Test Case Identification> 참조

8.4 Output specifications

<Table 4 Test Case Identification> 참조

9 Testing tasks

Table 5 Testing tasks & Schedule

Task	Predecessor tasks	Special skills	Effort	Finish date
(1) Unit Test Plan 작성	DS-DWS-2012-SRS 작성 DS-DWS-2012-SRA 작성 DS-DWS-2012-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	

10 Environmental needs

Digital watch system의 unit test를 위한 환경적 요구사항은 다음과 같다.

(1) Hardware & Platform, Eclipse IDE (Integrated Development Environment)

gcc compiler/linker

(2) CTIP (Continuous Testing & Integration Platform) Environment

CI server

SVN repository server

CI/SVN server에 접근 가능한 단말 PC

Test tools

CUnit unit test framework for C

gCov code coverage measurement tool

11 Unit Test deliverables

11.1 Unit test plan

11.2 Unit test design specification

11.3 Unit test case specification

11.4 Unit test summary report

12 Schedules

<Table 3 Testing tasks & Schedule> 참조