

Software Unit Test for Digital Watch System

Sun Hwi Lee
Dependable Software Laboratory

Contents

- Introduction
- Test plan for Controllers
- Test plan for Data processes
- Test result
- Appendix

Introduction

- Objective
 - 본 문서는 소프트웨어 공학 개론 B반 T8 팀이 개발한 Digital watch system을 위한 문서이다.
- Reference
 - SRS Ver. DS-2012.DWS.SRS-2.1 (2012-08-17)
 - B. T8 SRA Ver. T8-2012.DWS.SRA-3.6
 - B. T8 SDS Ver. T8-2012.DWS.SDS-3.6

Test plan for Controllers

Identifier	Controller	State	Input	Expected Output
UT.000.000	2.1.1 Time Controller	Time Display	<pre>btn = 'c'; *stateData.cs = TimeDisplay; changeState(*stateData); timeController(*stateData, *timeData);</pre>	<pre>*stateData.cs == StopwatchDisplay; Flag_StopwatchDisplay == 1;</pre>
UT.000.001			<pre>btn = 'a'; *stateData.cs = TimeDisplay; changeState(*stateData); timeController(*stateData, *timeData);</pre>	<pre>*stateData.cs == TimeSetting; Flag_TimeSetting == 1;</pre>
UT.000.002		Time Setting	<pre>btn = 'c'; *stateData.cs = TimeSetting; changeState(*stateData); timeController(*stateData, *timeData);</pre>	<pre>*stateData.cs == TimeSetting; Flag_MoveVariable == 1;</pre>
UT.000.003			<pre>btn = 'b'; *stateData.cs = TimeSetting; changeState(*stateData); timeController(*stateData, *timeData);</pre>	<pre>*stateData.cs == TimeSetting; Flag_addVariable == 1;</pre>
UT.000.004			<pre>btn = 'a'; *stateData.cs = TimeSetting; changeState(*stateData); timeController(*stateData, *timeData);</pre>	<pre>*stateData.cs == TimeDisplay; Flag_TimeDisplay == 1;</pre>

함수의 호출을 확인하기 위해 소스코드에 임시적인 변수 삽입.
 조건이 없이 Transition이 이루어지는 경우는 해당 Data Process를 Testing.

Test plan for Controllers

Identifier	Controller	State	Input	Expected Output
UT.000.005	2.1.1 Time Controller	Stopwatch Display	<pre>btn = 'c'; *stateData.cs = StopwatchDisplay; changeState(*stateData); timeController(*stateData, *timeData);</pre>	<pre>*stateData.cs == TimeDiplay; Flag_TimeDisplay == 1;</pre>
UT.000.006			<pre>btn = 'b'; *stateData.cs = StopwatchDisplay; *stateData.ss = MeasuringTimeOn; *stateData.ps != LapTime; changeState(*stateData); timeController(*stateData, *timeData);</pre>	<pre>*stateData.cs == StopwatchDisplay; Flag_measuringTimeOff == 1;</pre>
UT.000.007			<pre>btn = 'b';Me *stateData.cs = StopwatchDisplay; *stateData.ss = MeasuringTimeOff; *stateData.ps = LapTime; changeState(*stateData); timeController(*stateData, *timeData);</pre>	<pre>*stateData.cs == StopwatchDisplay; Flag_measuringTimeOn == 1;</pre>
UT.000.008		Measuring Time Off	<pre>btn = 'a'; *stateData.cs = StopwatchDisplay; *stateData.ss = MeasuringTimeOff; changeState(*stateData); timeController(*stateData, *timeData);</pre>	<pre>*stateData.cs == MeasuringTimeReset; Flag_measuringTimeReset == 1;</pre>
UT.001.000	2.1.2 BackLight Controller	Backlight On	<pre>Btn = 'd'; *stateData.bs = BacklightOn;</pre>	<pre>Flag_backlightOn == 1;</pre>

Test plan for Data processes

Identifier	Process	Input	Expected Output
UT.002.000	1.5 Change State	btn = 'a'; stateData.cs = TimeDisplay;	stateData.cs = TimeSetting;
UT.002.001		btn = 'a'; stateData.cs = TimeSetting;	stateData.cs = TimeDisplay;
UT.002.002		btn = 'a'; stateData.cs = StopwatchDisplay; stateData.ss = MeasuringTimeOff;	stateData.cs = MeasuringTimeReset;
UT.002.003		btn = 'a'; stateData.cs = StopwatchDisplay; stateData.ss = MeasuringTimeOn;	stateData.cs = Lapttime;
UT.002.004		btn = 'a'; stateData.cs = StopwatchDisplay; stateData.ps = LapTime;	stateData.cs = Lapttime;
UT.002.005		btn = 'b'; stateData.cs = StopwatchDisplay; stateData.ss = MeasuringTimeOn; stateData.ps = LapTime;	stateData.cs = MeasuringTimeOn;
UT.002.006		btn = 'b'; stateData.cs = StopwatchDisplay; stateData.ss = MeasuringTimeOn; stateData.ps != LapTime;	stateData.cs = MeasuringTimeOff;
UT.002.007		btn = 'b'; stateData.cs = StopwatchDisplay; stateData.ss = MeasuringTimeOff;	stateData.cs = MeasuringTimeOn;
UT.002.008		btn = 'b'; stateData.cs = TimeSetting;	stateData.cs = AddVariable;

입력과 관계된 Button Interface(1.1 - 1.4)는 테스트 대상에서 제외하였습니다.

Test plan for Data processes

Identifier	Process	Input	Expected Output
UT.002.009	1.5 Change State	btn = 'c'; stateData.cs = TimeDisplay;	stateData.cs = StopwatchDisplay;
UT.002.010		btn = 'c'; stateData.cs = TimeSetting;	stateData.cs = MoveVariable;
UT.002.011		btn = 'c'; stateData.cs = StopwatchDisplay;	stateData.cs = TimeDisplay;
UT.002.012		btn = 'd';	stateData.bs = BacklightOn;
UT.003.000	2.1.4 Add Variable	timeData.cursor = CSec; timeData.sec = 12;	timeData.sec = 13; stateData.cs = TimeSetting;
UT.003.001		timeData.cursor = CHour; timeData.hour = 1; timeData.pmIndicator = true;	timeData.hour = 2; timeData.pmIndicator = true; stateData.cs = TimeSetting;
UT.003.002		timeData.cursor = CMin; timeData.min = 59;	timeData.min = 0; stateData.cs = TimeSetting;
UT.003.003		timeData.cursor = CYear; timeData.year = 2014;	timeData.year = 2015; stateData.cs = TimeSetting;
UT.003.004		timeData.cursor = CMonth; timeData.month = 1;	timeData.month = 2; stateData.cs = TimeSetting;
UT.003.005		timeData.cursor = CDate; timeData.finalDate = 31; timeData.date = 10;	timeData.date = 11; stateData.cs = TimeSetting;

Test plan for Data processes

Identifier	Process	Input	Expected Output
UT.004.000	2.1.5 Move Variable	timeData.cursor = CDate;	timeData.cursor = CSec; stateData.cs = TimeSetting;
UT.005.000	2.1.8 Measuring Time On	stateData; timeData;	stateData.cs = StopwatchDisplay; stateData.ss = MeasuringTimeOn; stateData.ps = None; timeData.stStatus = true; timeData.lapStatus = false;
UT.006.000	2.1.9 Measuring Time Off	stateData; timeData;	stateData.cs = StopwatchDisplay; stateData.ss = MeasuringTimeOff; timeData.stStatus = false;
UT.007.000	2.1.10 Lap Time	stateData; timeData.stMin = 10; timeData.stSec = 10; timeData.st100Sec = 10;	timeData.lapMin = 10; timeData.lapSec = 10; timeData.lap100Sec = 10; timeData.lapStatus = true; stateData.ps = LapTime; stateData.cs = StopwatchDisplay;
UT.008.000	2.1.11 Measuring Time Reset	stateData; timeData.lapStatus = false; timeData.stMin = 22; timeData.stSec = 34; timeData.st100Sec = 45; timeData.lapMin = 10; timeData.lapSec = 34; timeData.lap100Sec = 20;	stateData.cs = StopwatchDisplay; timeData.lapStatus = false; timeData.stMin = 0; timeData.stSec = 0; timeData.st100Sec = 0; timeData.lapMin = 0; timeData.lapSec = 0; timeData.lap100Sec = 0;

실질적으로 시간을 증가시키는 Data process (2.1.13, 2.1.14)는 제외하였습니다.
Backlight On (2.1.12) 또한 제외하였습니다.

Test Result

CUnit – A Unit testing framework for C.
<http://cunit.sourceforge.net/>

Automated Test Run Results

Running Suite Controllers

Running test DWS.UTC_000_000 ...	Passed
Running test DWS.UTC_000_001 ...	Passed
Running test DWS.UTC_000_002 ...	Passed
Running test DWS.UTC_000_003 ...	Passed
Running test DWS.UTC_000_004 ...	Passed
Running test DWS.UTC_000_005 ...	Passed
Running test DWS.UTC_000_006 ...	Passed
Running test DWS.UTC_000_007 ...	Passed
Running test DWS.UTC_000_008 ...	Passed
Running test DWS.UTC_001_000 ...	Passed

Running Suite DataProcesses

Running test DWS.UTC_002_000 ...	Passed
Running test DWS.UTC_002_001 ...	Passed
Running test DWS.UTC_002_002 ...	Passed
Running test DWS.UTC_002_003 ...	Passed
Running test DWS.UTC_002_004 ...	Passed
Running test DWS.UTC_002_005 ...	Passed
Running test DWS.UTC_002_006 ...	Passed
Running test DWS.UTC_002_007 ...	Passed
Running test DWS.UTC_002_008 ...	Passed
Running test DWS.UTC_002_009 ...	Passed
Running test DWS.UTC_002_010 ...	Passed
Running test DWS.UTC_002_011 ...	Passed
Running test DWS.UTC_002_012 ...	Passed
Running test DWS.UTC_003_000 ...	Passed
Running test DWS.UTC_003_001 ...	Passed
Running test DWS.UTC_003_002 ...	Passed
Running test DWS.UTC_003_003 ...	Passed
Running test DWS.UTC_003_004 ...	Passed
Running test DWS.UTC_003_005 ...	Passed
Running test DWS.UTC_004_000 ...	Passed
Running test DWS.UTC_005_000 ...	Passed
Running test DWS.UTC_006_000 ...	Passed
Running test DWS.UTC_007_000 ...	Passed
Running test DWS.UTC_008_000 ...	Passed

Cumulative Summary for Run

Type	Total	Run	Succeeded	Failed	Inactive
Suites	2	2	- NA -	0	0
Test Cases	34	34	34	0	0
Assertions	34	34	34	0	n/a

Test Result

- Comment
 - 거의 대부분의 경우, SRA, SDS 문서와 소스코드가 일치하였습니다.
 - 모든 Test Case를 Pass 하였습니다.