

Unit Test Report for Digital Watch System

- Test Cases Specification
- Test Summary Report

Project Team
Dependable Software Laboratory

Latest update on:
2012-11-11

Team Information

Sanghyun Yoon shyoon.dslab@gmail.com

Table of Contents

1	Introduction	3
1.1	Objectives.....	3
1.2	References.....	3
2	Features to be tested.....	3
3	Features not to be tested	4
4	Unit test case specification.....	5
4.1	Test case specification identifier.....	5
4.2	Test items	7
4.3	Input specifications.....	8
4.4	Output specifications.....	8
5	Environmental needs.....	8
6	Unit test summary report	8
6.1	Test summary report identifier.....	9
6.2	Evaluation.....	9

1 Introduction

1.1 Objectives

본 문서는 digital watch system의 unit test를 수행한 결과에 대한 report 문서이다. Test 요소들에 대한 test case와 test수행 결과에 대한 내용을 담고 있다.

1.2 References

DS.2012.DWS.SRS-2.1

T2_SRA

T2_SDS

2 Features to be tested

Table 1 테스트할 Process 목록

ID	Name	Description
2.1.3.1	Current time controller	시간구조체 now_time을 이용하여 시스템에서 현재시간을 받아오고, 시간을 증가시키는 등의 프로세스 들을 관장하는 컨트롤. Start_setup 플래그를 이용, 프로그램 시간을 받아왔는지 여부를 보고 받아왔으면 그 때부터 시간 증가 프로세스를 시간한다. Second_check 틱을 이용하여 1초가 경과하였는지 체크한다.
2.1.3.2	Increase time	시간 구조체 wt 의 sec 를 증가시키는 프로세스. 현재 시간증가를 의미한다.
2.1.4	Mode Select	TimeKeep 상태 일 때 key에따라 mode_state를 Stopwatch mode나Change time mode로바꿔주고, 키를 초기화한다.
2.1.5	Time keep	매 tick 마다 wt 를 초, 분, 시, 날짜 범위에 범위에 맞게 수정한다
2.1.6.1	Change time controller	스탑워치의 시간을 바꿔주고, 바꾸는 대상을 변경하는 등 시간변경 프로세스들을 관장하는 컨트롤러. Key 값 변수를 이용 다시 time keep 모드로 돌아갈것인지 시간증가를 할것인지 대상 변경을 할 것인지 등을 결정한다.
2.1.6.2	Change Object	스탑워치 구조체의 pos 변수를 변경하는 프로세스. 시간변경 대상을 바꿔준다.
2.1.6.3	Increase Digit	스탑워치 구조체의 pos 에 맞는 대상을 1 씩 증가시켜주는 프로세스. 시간 변경을 의미하며, 단위변경도 한다.
2.1.7.1	Stopwatch Controller	스탑워치 기능을 관장하는 컨트롤러. Key 값을 이용 다시 timekeep 모드로 돌아갈 것인지, 스톱워치 시간을 증가시키거나 정지시킬 것인지, 랩타임을 출력할 것인지를 컨트롤한다.
2.1.7.2.1	SW increase & stop	스탑워치의 시간증가 및 정지를 조절하는 컨트롤러. 스톱워치

	Controller	구조체, s_time 내의 start_state 값을 비교하여 0 이면 정지상태, 1 이면 동작상태를 구현한다.
2.1.7.2.2	SW second change	스탑워치내 초를 증가시키는 프로세스. S_time 구조체내의 밀리세컨트, Sw_ss 가 100 이 되면 1 초를 증가시킨다.
2.1.7.2.3	SW minute change	스탑워치내 분을 증가시키는 프로세스. S_time 구조체내의 초, Sw_s 가 60 이 되면 1 분을 증가시킨다.
2.1.8.1	Backlight controller	백라이트에 관련되는 프로세스들을 관장하는 컨트롤러. Light status flag 를 이용하여 현재 백라이트가 켜지는 상태인지를 확인하고, light_tick 을 이용하여 2 초를 계산한다.

3 Features not to be tested

외부 장치 드라이버, 단순 데이터 전달 프로세스 등은 test에서 제외한다. 또는 SRA, SDS 문서와 다르게 실제구현에서는 unit으로 나누어 있지 않은 프로세스를 제외한다.

Table 2 Test 하지 않을 process 목록

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 바꿔 전달한다.
1.5	Button Type Detect	입력된 버튼정보를 컨트롤 프로세스로 전달한다.
2.1.2	Button detection	키보드로부터 A, B, C, D 버튼을 입력 받아 key 변수에 최우선 순위 버튼을 저장한다.
2.1.3.3	Get current time	시스템시간이 들어있는 시간구조체 tm)time 을 이용하여 스탑워치 시간을 초기화 하는 프로세스. 스탑워치 구조체 wt 의 각 변수들에 시간 구조체 변수들을 대입한다. 다른 프로세스의 설명으로 추측.
2.1.9.2	TK Print	현재시간을 출력하는 프로세스, 스탑워치 구조체, wt 를 받아와 각각의 변수들을 출력한다.
2.1.9.3	CT Print	시간변경을 위한 화면을 출력하는 프로세스. 타임워치 구조체, wt 를 받아와 각각의 변수들을 출력하며, 시간변경대상 즉 wt 의 pos 를 이용한 변경대상에 밑줄을 친다.
2.1.9.4	SW Print	스탑워치를 위한 화면을 출력하는 프로세스. 스탑워치타임 구조체, s_time 를 받아 start_state 를 비교하여 Lap 이면 laptime 을, Lap 이

		아니면 스탑워치시간의 변수를 출력하며, 타임워치 구조체, wt 를 받아 현재시간도 출력한다.
2.1.1	Main controller	매 Tick 마다 반복되어 실행된다. 2.1.2 부터 2.1.9 의프로세스를매 Tick 마다실행한다. 출력 값이 없음. 함수들을 조건없이 호출하고 필요한 인자 값들을 넘겨줌
2.1.7.4	Lap time	현재 스탑워치 시간을 스탑워치 구조체, s_time 내 Lap time 변수에 저장하는 프로세스.
2.1.7.3	Reset Stopwatch	스탑워치 시간을 리셋시키는 프로세스. 스탑워치 구조체, s_time 내의 모든 변수들 값을 0 으로 초기화 시킨다.
2.1.8.2	Turn on/off	백라이트를 키거나 끄는 프로세스. 실제론 글자색을 의미하는 정수 i를 받아 그 색을 출력하게끔 한다.
2.1.9.1	Display controller	출력에 관한 프로세스들을 관장하는 컨트롤러. State_mode 를 비교하여 맞는 state 의 출력 프로세스를 실행시킨다. 현재시간 구조체인 wt 와 스탑워치 구조체 s_time 을 인자로 쓴다.

4 Unit test case specification

4.1 Test case specification identifier

Table 3 Test Case Identification

Test case identifier	Input specification	Output specification
DWS.UTC_00.00	start_setup==STOP	start_setup==START
DWS.UTC_00.01	start_setup==START,	
DWS.UTC_01.00	wt->sec = 0	wt->sec = 1
DWS.UTC_02.00	mode_state = TK, key = 'a'	mode_state = CT
DWS.UTC_02.01	mode_state = TK, key = 'c'	mode_state =SW
DWS.UTC_03.00	wt->year = 2012	Timekeep_endday [1]=29
DWS.UTC_03.01	wt->hour = 12	wt->PM = 1
DWS.UTC_03.02	Wt->hour = 1, wt-min = 60	Wt->min = 0, wt->hour =2
DWS.UTC_03.03	Wt->min = 23, Wt->sec = 60	Wt->min = 24, wt->sec = 0
DWS.UTC_03.04	Wt->year=2012, Wt->month = 13	Wt->year=2013, Wt->month=1
DWS.UTC_03.05	wt->month = 12, wt->mday = 31	Wt->month=1, wt->mday = 1
DWS.UTC_04.00	mode_state = CT, key = a	Mode_state = TK
DWS.UTC_04.01	mode_state = CT, key = b	Mode_state = CT
DWS.UTC_04.02	mode_state = CT, key = c	Mode_state = CT
DWS.UTC_05.00	Wt->pos = 0	Wt->pos = 1
DWS.UTC_05.01	Wt->pos = 6	Wt->pos = 0
DWS.UTC_06.00	Wt->pos =0, Wt->sec = 0	Wt->sec = 1
DWS.UTC_06.01	Wt->pos =1, Wt->hour = 0	Wt->hour = 1
DWS.UTC_06.02	Wt->pos =2, Wt->min = 0	Wt->min = 1
DWS.UTC_06.03	Wt->pos =3, Wt->year = 0	Wt->year = 1

DWS.UTC_06.04	Wt->pos =4, Wt->month = 0	Wt->month = 1
DWS.UTC_06.05	Wt->pos =5, Wt->mday = 0	Wt->mday = 1
DWS.UTC_06.06	Wt->pos =0, Wt->sec = 59, wt->min =0	Wt->sec = 0, wt->min = 1
DWS.UTC_06.07	Wt->pos =1, Wt->hour = 11	Wt->hour = 12, wt->pm =1
DWS.UTC_06.08	Wt->pos =2, Wt->min = 59, wt->hour=1	Wt->min = 0, wt->hour = 2
DWS.UTC_06.09	Wt->pos =3, Wt->year = 2099	Wt->year = 2012
DWS.UTC_06.10	Wt->pos =4, Wt->month = 12, wt->year =12	Wt->month = 1, wt->year = 13
DWS.UTC_06.11	Wt->pos =5, Wt->mday = 28, wt->year =12, wt->month = 2	Wt->mday = 29, wt->month = 2
DWS.UTC_06.12	Wt->pos =5, Wt->mday = 28, wt->year =13, wt->month=2	Wt->mday = 1, wt->month = 3
DWS.UTC_07.00	mode_state == SW, s_time->start_state = START, key = a	s_time->start_state=LAP
DWS.UTC_07.01	mode_state == SW, s_time->start_state= START, key = b	s_time->start_state =STOP
DWS.UTC_07.02	mode_state == SW, s_time->start_state= START, key = c	mode_state = TK
DWS.UTC_07.03	mode_state == SW, s_time->start_state=STOP, key = a	s_time->start_state=STOP
DWS.UTC_07.04	mode_state == SW, s_time->start_state=STOP, key = b	s_time->start_state=START
DWS.UTC_07.05	mode_state == SW, s_time->start_state=STOP, key = c	mode_state = TK
DWS.UTC_07.06	mode_state == SW, s_time->start_state=LAP, key = a	s_time->start_state=LAP
DWS.UTC_07.07	mode_state == SW, s_time->start_state=LAP, key = b	s_time->start_state=START
DWS.UTC_07.08	mode_state == SW, s_time->start_state=LAP, key = c	mode_state = TK
DWS.UTC_08.00	S_time->SW_ss = 0	S_time->SW_ss = 1
DWS.UTC_09.00	s_time->SW_ss = 100, s_time->SW_s=0	s_time->SW_ss = 0, s_time->SW_s=1
DWS.UTC_10.00	s_time->SW_m = 0, s_time->SW_s=60	s_time->SW_m = 1, s_time->SW_s=0
DWS.UTC_11.00	s_time	s_time->SW_m=0; s_time->SW_s=0; s_time->SW_ss=0; s_time->start_state=STOP; s_time->Lap_m=0; s_time->Lap_s=0; s_time->Lap_ss=0;
DWS.UTC_12.00	Key = d, light_status = 0;	light_status = 1;

4.2 Test items

Table 4 Test Design Identification

Identifier	Feature (Process ID in DFD)	Valid/ Invalid value
DWS.UTC_00.00	2.1.3.1 Current time controller	*start_setup==STOP이 들어온다.
DWS.UTC_00.01	2.1.3.1 Current time controller	*start_setup==START가 들어온다. 1초가 지난다.
DWS.UTC_01.00	2.1.3.2 Increase time	함수가 호출 된다.
DWS.UTC_02.00	2.1.4 Mode Select	* mode_state = TK(timekeeping)인 상태에서 a 버튼 입력(key=a)이 들어온다.
DWS.UTC_02.01	2.1.4 Mode Select	* mode_state = TK 인 상태에서 c 버튼 입력이 들어온다.
DWS.UTC_03.00	2.1.5 Time keep	*wt 구조체에 윤년이 들어온다.
DWS.UTC_03.01	2.1.5 Time keep	*wt 구조체의 hour 가 12 이상이다.
DWS.UTC_03.02	2.1.5 Time keep	*wt 구조체의 min 이 60 이상이다.
DWS.UTC_03.03	2.1.5 Time keep	*wt 구조체의 sec 가 60 이상이다.
DWS.UTC_03.04	2.1.5 Time keep	*wt 구조체의 month 가 13 이상이다.
DWS.UTC_03.05	2.1.5 Time keep	*wt 구조체의 mday(일)이 해당 달의 최대일보다 크다.
DWS.UTC_04.00	2.1.6.1 Change time controller	*mode_state = CT (현재시간변경)인 상태에서 a 버튼 입력이 들어온다.
DWS.UTC_04.01	2.1.6.1 Change time controller	*mode_state = CT 상태에서 b 버튼 입력이 들어온다.
DWS.UTC_04.02	2.1.6.1 Change time controller	*mode_state = CT 상태에서 c 버튼 입력이 들어온다.
DWS.UTC_05.00	2.1.6.2 Change Object	함수가 호출된다.
DWS.UTC_05.01	2.1.6.2 Change Object	pos 값이 최대인 상태에서 함수가 호출 된다.
DWS.UTC_06.00	2.1.6.3. Increase Digit	'초'가 선택된 상태에서 함수가 호출된다.
DWS.UTC_06.01	2.1.6.3. Increase Digit	'시'가 선택된 상태에서 함수가 호출된다.
DWS.UTC_06.02	2.1.6.3. Increase Digit	'분'이 선택된 상태에서 함수가 호출된다.
DWS.UTC_06.03	2.1.6.3. Increase Digit	'년'이 선택된 상태에서 함수가 호출된다.
DWS.UTC_06.04	2.1.6.3. Increase Digit	'월'이 선택된 상태에서 함수가 호출된다.
DWS.UTC_06.05	2.1.6.3. Increase Digit	'일'이 선택된 상태에서 함수가 호출된다.
DWS.UTC_06.06	2.1.6.3. Increase Digit	'초'가 59 이고 '초'가 선택된 상태에서 함수가 호출된다.
DWS.UTC_06.07	2.1.6.3. Increase Digit	'시'가 11 이고 '시'가 선택된 상태에서 함수가 호출된다.
DWS.UTC_06.08	2.1.6.3. Increase Digit	'분'이 59 이고 '분'이 선택된 상태에서 함수가 호출된다.
DWS.UTC_06.09	2.1.6.3. Increase Digit	'년'이 2099 이고 '년'이 선택된 상태에서 함수가 호출된다.
DWS.UTC_06.10	2.1.6.3. Increase Digit	'월'이 12 이고 '월'이 선택된 상태에서 함수가

		호출된다.
DWS.UTC_06.11	2.1.6.3. Increase Digit	윤년 2 월에 '일'이 28 일이고 '일'이 선택된 상태에서 함수가 호출된다.
DWS.UTC_06.12	2.1.6.3. Increase Digit	윤년이 아닌 2 월에 '일' 28 일이고 '일'이 선택된 상태에서 함수가 호출된다.
DWS.UTC_07.00	2.1.7.1. Stopwatch Controller	스탑워치가 동작하는 상태에서 a 버튼 입력이 들어온다.
DWS.UTC_07.01	2.1.7.1. Stopwatch Controller	스탑워치가 동작하는 상태에서 b 버튼 입력이 들어온다.
DWS.UTC_07.02	2.1.7.1. Stopwatch Controller	스탑워치가 동작하는 상태에서 c 버튼 입력이 들어온다.
DWS.UTC_07.03	2.1.7.1. Stopwatch Controller	스탑워치가 멈춘 상태에서 a 버튼 입력이 들어온다.
DWS.UTC_07.04	2.1.7.1. Stopwatch Controller	스탑워치가 멈춘 상태에서 b 버튼 입력이 들어온다.
DWS.UTC_07.05	2.1.7.1. Stopwatch Controller	스탑워치가 멈춘 상태에서 c 버튼 입력이 들어온다.
DWS.UTC_07.06	2.1.7.1. Stopwatch Controller	스탑워치가 Lapt time 을 보여주는 상태에서 a 버튼 입력이 들어온다.
DWS.UTC_07.07	2.1.7.1. Stopwatch Controller	스탑워치가 Lapt time 을 보여주는 상태에서 b 버튼 입력이 들어온다.
DWS.UTC_07.08	2.1.7.1. Stopwatch Controller	스탑워치가 Lapt time 을 보여주는 상태에서 c 버튼 입력이 들어온다.
DWS.UTC_08.00	2.1.7.2.1 SW increase & stop Controller	함수가 호출 된다.
DWS.UTC_09.00	2.1.7.2.2 SW second change	100 분의 1 초(s_time->SW_ss)가 100 이다.
DWS.UTC_10.00	2.1.7.2.3 SW minute change	'초' (s_time->SW_s)가 60 이다.
DWS.UTC_11.00	2.1.7.3 Reset Stopwatch	함수가 호출 된다.
DWS.UTC_12.00	2.1.8.1 Backlight controller	d 버튼 입력이 들어온다.

4.3 Input specifications

<Table 3 Test Case Identification> 참조.

4.4 Output specifications

<Table 3 Test Case Identification> 참조.

5 Environmental needs

6 Unit test summary report

6.1 Test summary report identifier

<Table 3 참조>

6.2 Evaluation

Pos에 대한 설명이 data dictionary에 없습니다. 시간 수정 시 선택된 위치로 해석하여 테스트 하였습니다.

1초 증가 시키는 부분에서 시스템 시간을 사용하고 있습니다. Tick이 몇 번 지났는지 확인하는 방식으로 바뀌어야 됩니다. 따라서, DWS.UTC_00.01는 invalid.

2.1.7.2.1 SW increase & stop Controller 의 설명이 잘못되어있습니다. 정지도 관리한다고 되어 있지만 실제로는 시간을 증가시키는 용도로만 쓰입니다.

DWS.UTC_03.00 등에서 endday를 접근할 수가 없어서 실제 테스트에서는 소스를 약간 수정하여 확인하였습니다.

*wt 를 controller에서 wt를 받아 호출하는 process들에게 넘겨주는 형태로 되어 있는 경우가 있습니다. 엄밀히 말하면 그림과 다른 구현입니다.

Identifier	PASS/ FAIL
DWS.UTC_00.00	PASS
DWS.UTC_00.01	PASS
DWS.UTC_01.00	PASS
DWS.UTC_02.00	PASS
DWS.UTC_02.01	PASS
DWS.UTC_03.00	PASS
DWS.UTC_03.01	PASS
DWS.UTC_03.02	PASS
DWS.UTC_03.03	PASS
DWS.UTC_03.04	PASS
DWS.UTC_03.05	PASS
DWS.UTC_04.00	PASS
DWS.UTC_04.01	PASS
DWS.UTC_04.02	PASS
DWS.UTC_05.00	PASS
DWS.UTC_05.01	PASS
DWS.UTC_06.00	PASS
DWS.UTC_06.01	PASS
DWS.UTC_06.02	PASS

DWS.UTC_06.03	PASS
DWS.UTC_06.04	PASS
DWS.UTC_06.05	PASS
DWS.UTC_06.06	PASS
DWS.UTC_06.07	PASS
DWS.UTC_06.08	PASS
DWS.UTC_06.09	PASS
DWS.UTC_06.10	PASS
DWS.UTC_06.11	PASS
DWS.UTC_06.12	PASS
DWS.UTC_07.00	PASS
DWS.UTC_07.01	PASS
DWS.UTC_07.02	PASS
DWS.UTC_07.03	PASS
DWS.UTC_07.04	PASS
DWS.UTC_07.05	PASS
DWS.UTC_07.06	PASS
DWS.UTC_07.07	PASS
DWS.UTC_07.08	PASS
DWS.UTC_08.00	PASS
DWS.UTC_09.00	PASS
DWS.UTC_10.00	PASS
DWS.UTC_11.00	PASS
DWS.UTC_12.00	PASS