

Electronic Door Lock System Unit Test Result

2013.12.06

Team Presentation#5

Team 5

강민우 201211324

임동현 201211353

서동현 201211375

Presenter → 함진아 201211389

Contents

- TEST ENVIRONMENT
- MODIFICATION
- TEST CASE
- TEST CODE
- UNIT TEST RESULT

Test Environment

개발 환경	:	Eclipse
컴파일러	:	MinGW
테스트 툴	:	Cunit

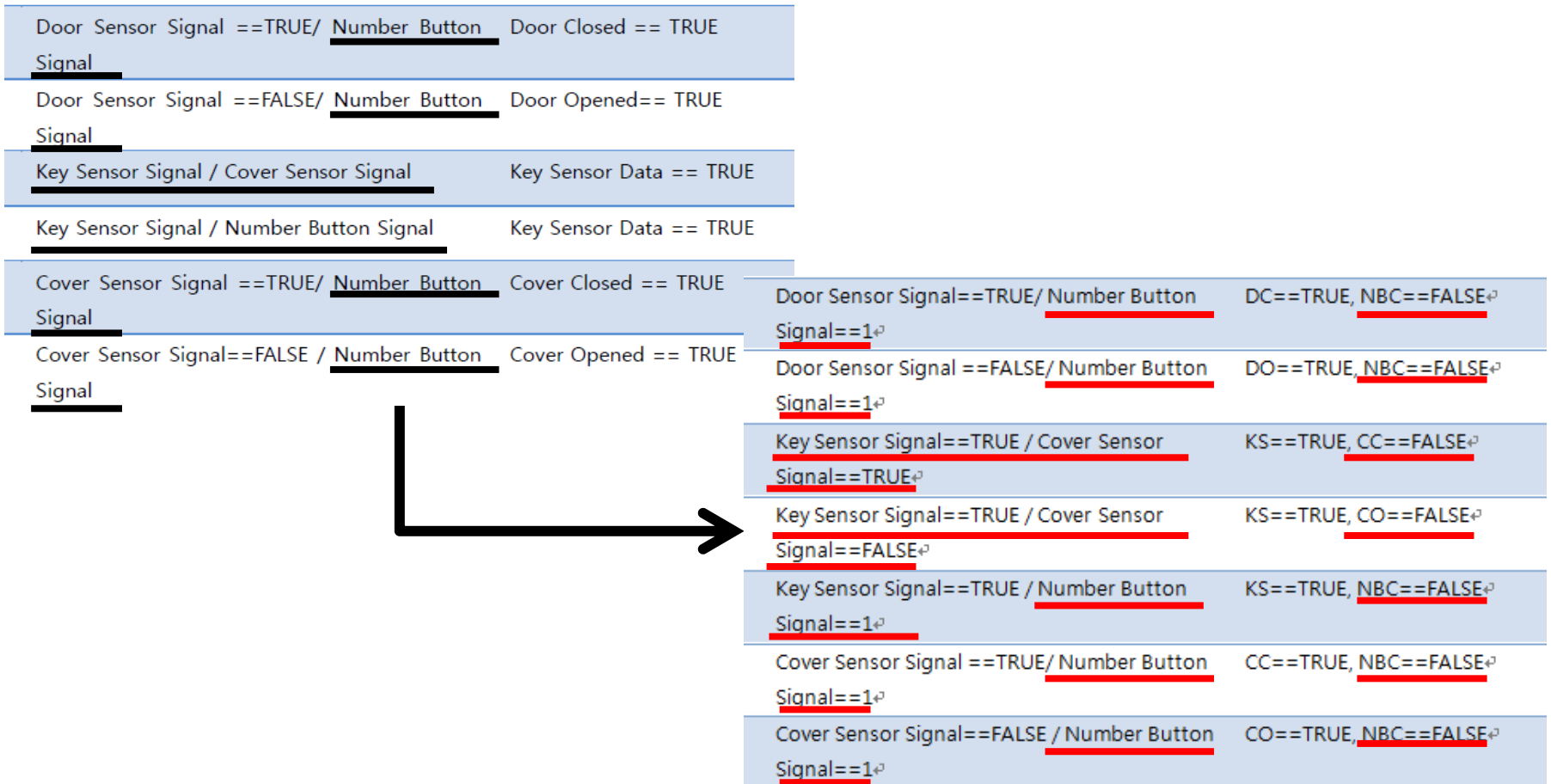
- Cunit 2006년 버전 (Cunit-2.1-0)
- 최신 버전은 윈도우 라이브러리가 존재하지 않기 때문
- MinGW에 include하여 연동

Modification

- UTP-2.0
 - DFD, STD 수정에 따른 Test case 수정 (미완)
 - 1.6 Determine Signal test case 추가
- UTP-3.3
 - SRA, SD 수정에 따른 Test case 수정 완료
 - 애매한 표현에 대한 구체화
 - Test case 79개 → 109개 (세분화 작업)

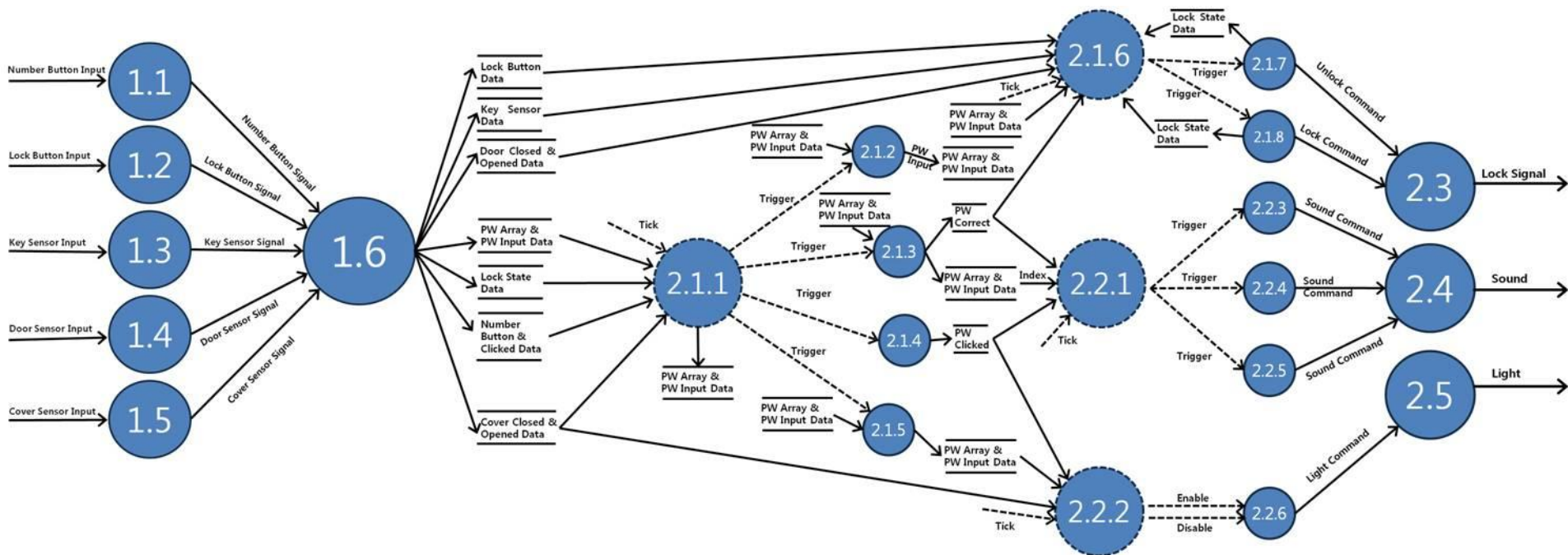
Modification

- 애매한 표현에 대한 구체화의 일부



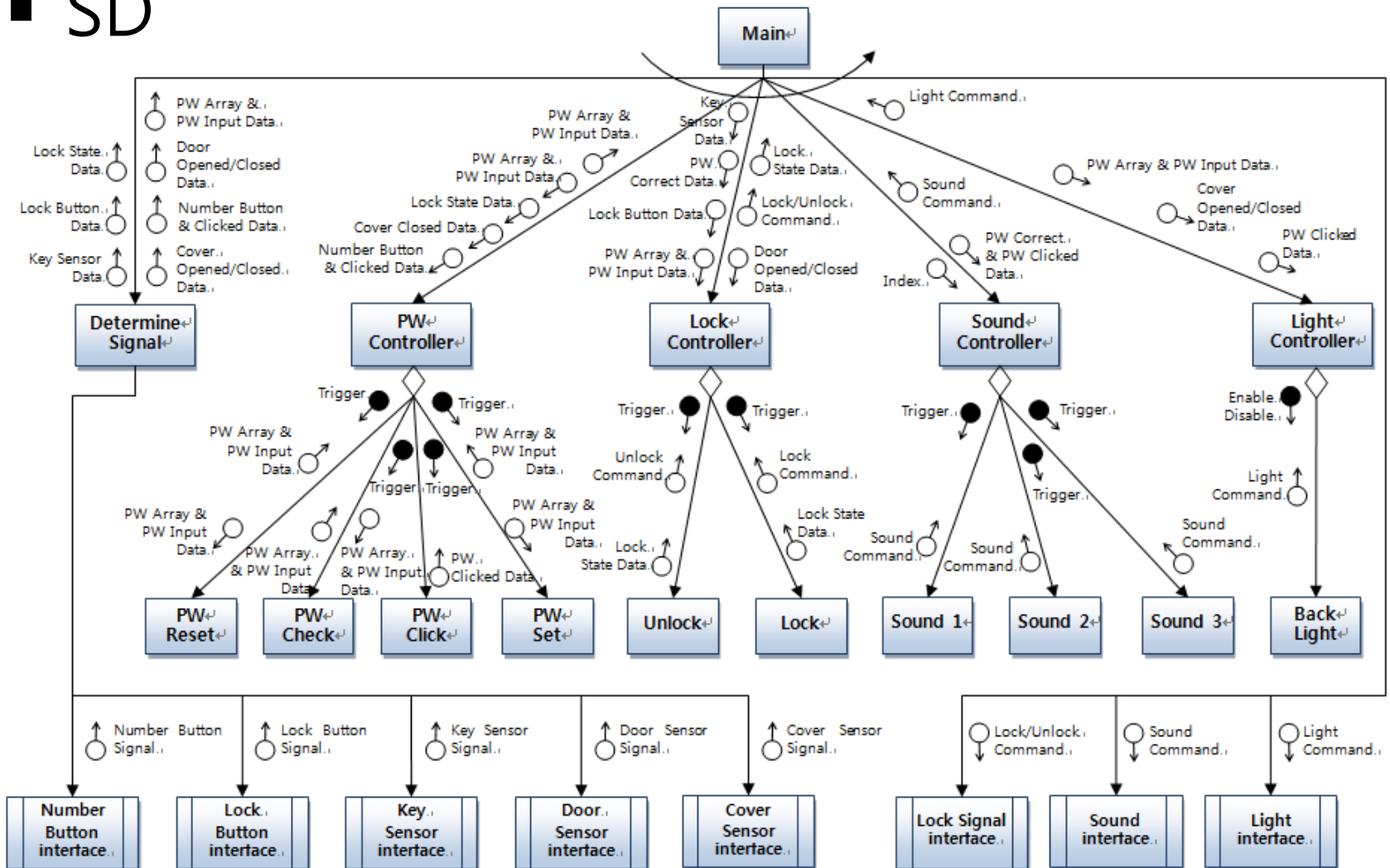
Test Case

Overall DFD



Test Case

■ SD



Test Case

■ 테스트할 Process(DFD) 리스트(1/2)

ID	Name	Description
1.6	Determine Signal	Number Button, Lock Button, Key Sensor, Door Sensor, Cover Sensor Signal들을 받아서 우선순위에 따라 분류하여 가공된 데이터를 넘긴다.
2.1.1	PW Controller	Button Data와 PW Array&PW Input data, Cover Opened&Closed data, 그리고 Lock State Data를 받아서 상태를 통해 적절한 프로세스에게 명령을 전달한다.
2.1.2	PW Reset	PW Controller가 비밀번호를 초기화할때 호출하는 프로세스. Trigger 명령을 받으면 PW Array&PW Input data 안의 PW Input data와 Index를 초기화시킨다
2.1.3	PW Check	PW Controller가 비밀번호를 확인할 때 호출하는 프로세스. Trigger 명령을 받으면 PW Input을 PW Array와 비교하여 PW Correct를 True 혹은 False로 Sound Controller와 Lock Controller에게 보낸다
2.1.4	PW Click	Trigger 명령을 받으면 PW Click Data를 True 상태로 Sound Controller와 Light Controller에게 보낸다.
2.1.5	PW Set	초기 비밀번호를 설정하는 프로세스. Trigger 명령을 받으면 PW Input의 내용을 PW Array에 옮긴다.
2.1.6	Lock Controller	Lock Button Data, Key Sensor Data, Door Closed&Opened Data, PW Correct Data, PW Input&Array Data를 받아 적절한 프로세스에게 명령을 전달한다.
2.1.7	Unlock	Trigger를 받으면 Lock State Data를 False 상태로 변경하고, Unlock Command를 Lock Interface에게 전달한다.
2.1.8	Lock	Trigger를 받으면 Lock State Data를 true 상태로 변경하고, Lock Command를 Lock Interface에게 전달한다.

Test Case

■ 테스트할 Process(DFD) 리스트(2/2)

ID	Name	Description
2.2.1	Sound Controller	PW Correct Data, PW Clicked Data를 받아서 적절한 소리를 판단해 해당하는 프로세스에게 명령을 전달한다.
2.2.2	Light Controller	PW Clicked, Cover Closed&Opened Data, PW Input & PW Array Data를 받아 불을 끌 것인지 켤 것인지 판단하여 프로세스에게 적절한 명령을 보낸다.
2.2.3	Sound 1	Trigger를 받으면 Sound 1 명령을 Sound Interface에 전달한다.
2.2.4	Sound 2	Trigger를 받으면 Sound 2 명령을 Sound Interface에 전달한다.
2.2.5	Sound 3	Trigger를 받으면 Sound 3 명령을 Sound Interface에 전달한다.
2.2.6	Back Light	Enable을 받으면 Light Interface에 불을 키는 명령을 전달하고, Disable을 받으면 불을 끄는 명령을 전달한다.
2.2.1	Sound Controller	PW Correct Data, PW Clicked Data를 받아서 적절한 소리를 판단해 해당하는 프로세스에게 명령을 전달한다.
2.2.2	Light Controller	PW Clicked, Cover Closed&Opened Data, PW Input & PW Array Data를 받아 불을 끌 것인지 켤 것인지 판단하여 프로세스에게 적절한 명령을 보낸다.
2.2.3	Sound 1	Trigger를 받으면 Sound 1 명령을 Sound Interface에 전달한다.

Test Case

■ 테스트하지 않을 Process(DFD) 리스트

ID	Name	Description
1.1	Number Button Interface	Number Button Input을 입력 받아서 Determine Signal에 디지털 신호로 넘겨준다.
1.2	Lock Button Interface	Lock Button Input을 입력 받아서 Determine Signal에 디지털 신호로 넘겨준다.
1.3	Key Sensor Interface	Key Sensor Input을 입력 받아서 Determine Signal에 디지털 신호로 넘겨준다.
1.4	Door Sensor Interface	Door Sensor Input을 입력 받아서 Determine Signal에 디지털 신호로 넘겨준다.
1.5	Cover Sensor Interface	Cover Sensor Input을 입력 받아서 Determine Signal에 디지털 신호로 넘겨준다.
2.3	Lock Signal Interface	Lock command를 받아서 잠금 장치에 잠금 상태를 바꾸는 신호를 보낸다.
2.4	Sound Interface	Sound command를 받아서 소리를 켜는 Sound Signal을 Alarm 장치에 보낸다.
2.5	Light Interface	Light command를 받아서 불을 키고 끄는 Light Signal을 Back Light 장치에 보낸다.

Test Code

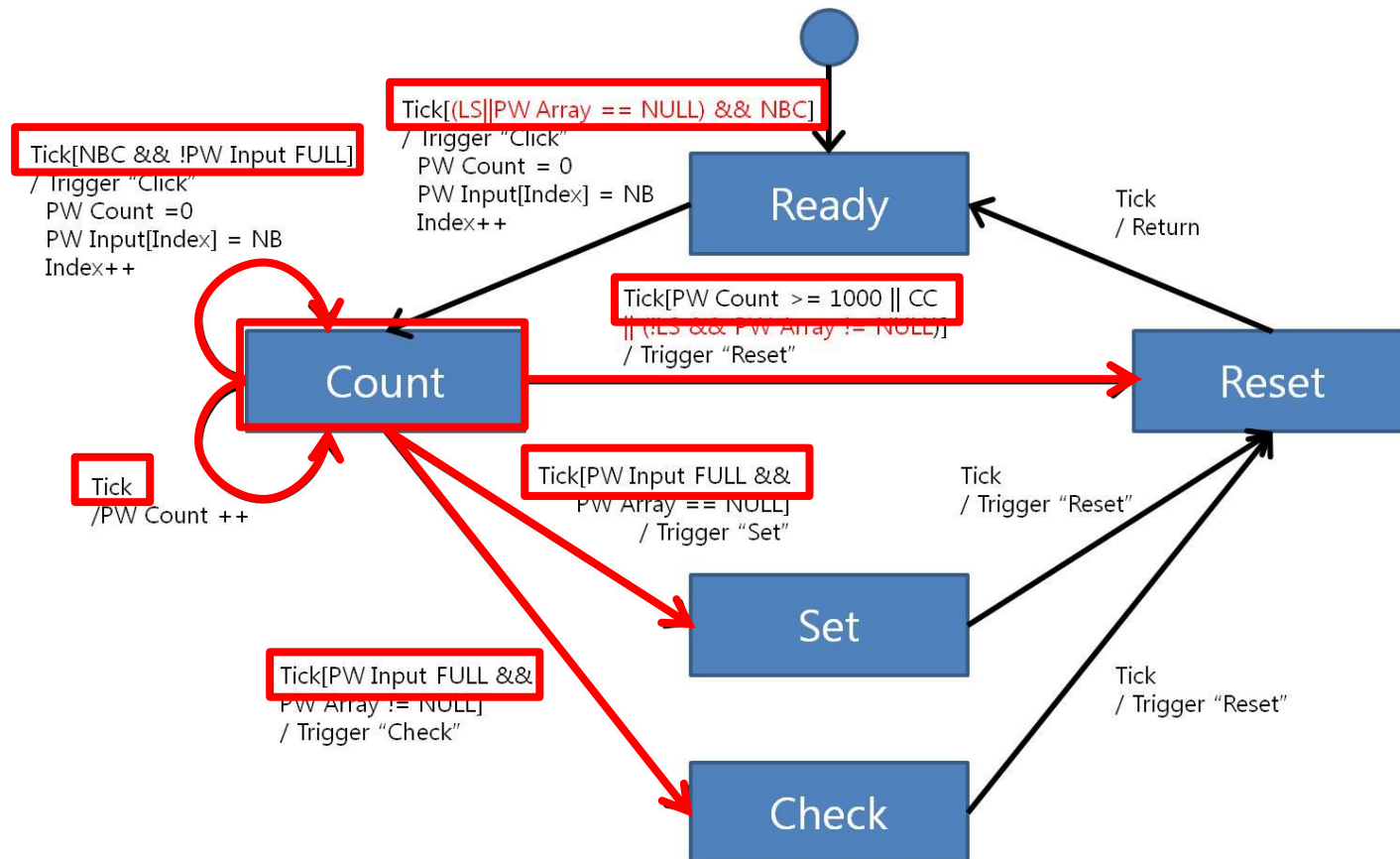
- Test 첫 시도에 나타난 Failures

```
src_file:line# : (suite:test) : failure_condition
1. ..\src\cunittest.c:2152 : (1.6. Determine Signal : EDLS.UTC_160_005) : CU_ASSERT_EQUAL(bss_data->numbtn_data->num_button,-1)
2. ..\src\cunittest.c:2229 : (1.6. Determine Signal : EDLS.UTC_160_012) : CU_ASSERT_EQUAL(bss_data->numbtn_data->num_button,-1)
3. ..\src\cunittest.c:2240 : (1.6. Determine Signal : EDLS.UTC_160_013) : CU_ASSERT_EQUAL(bss_data->numbtn_data->num_button,-1)
4. ..\src\cunittest.c:2273 : (1.6. Determine Signal : EDLS.UTC_160_016) : CU_ASSERT_EQUAL(bss_data->numbtn_data->num_button,-1)
5. ..\src\cunittest.c:2284 : (1.6. Determine Signal : EDLS.UTC_160_017) : CU_ASSERT_EQUAL(bss_data->numbtn_data->num_button,-1)
6. ..\src\cunittest.c:2295 : (1.6. Determine Signal : EDLS.UTC_160_018) : CU_ASSERT_EQUAL(bss_data->numbtn_data->num_button,-1)
7. ..\src\cunittest.c:339 : (2.1.1 PW_Controller : EDLS.UTC_211_010) : pw_state == COUNT
8. ..\src\cunittest.c:1580 : (2.2.1 Sound_Controller : EDLS.UTC_221_004) : state==Sound_Count_Mod
9. ..\src\cunittest.c:1652 : (2.2.1 Sound_Controller : EDLS.UTC_221_008) : state==Sound_Count_Mod
```

Total Number of Failures : 9

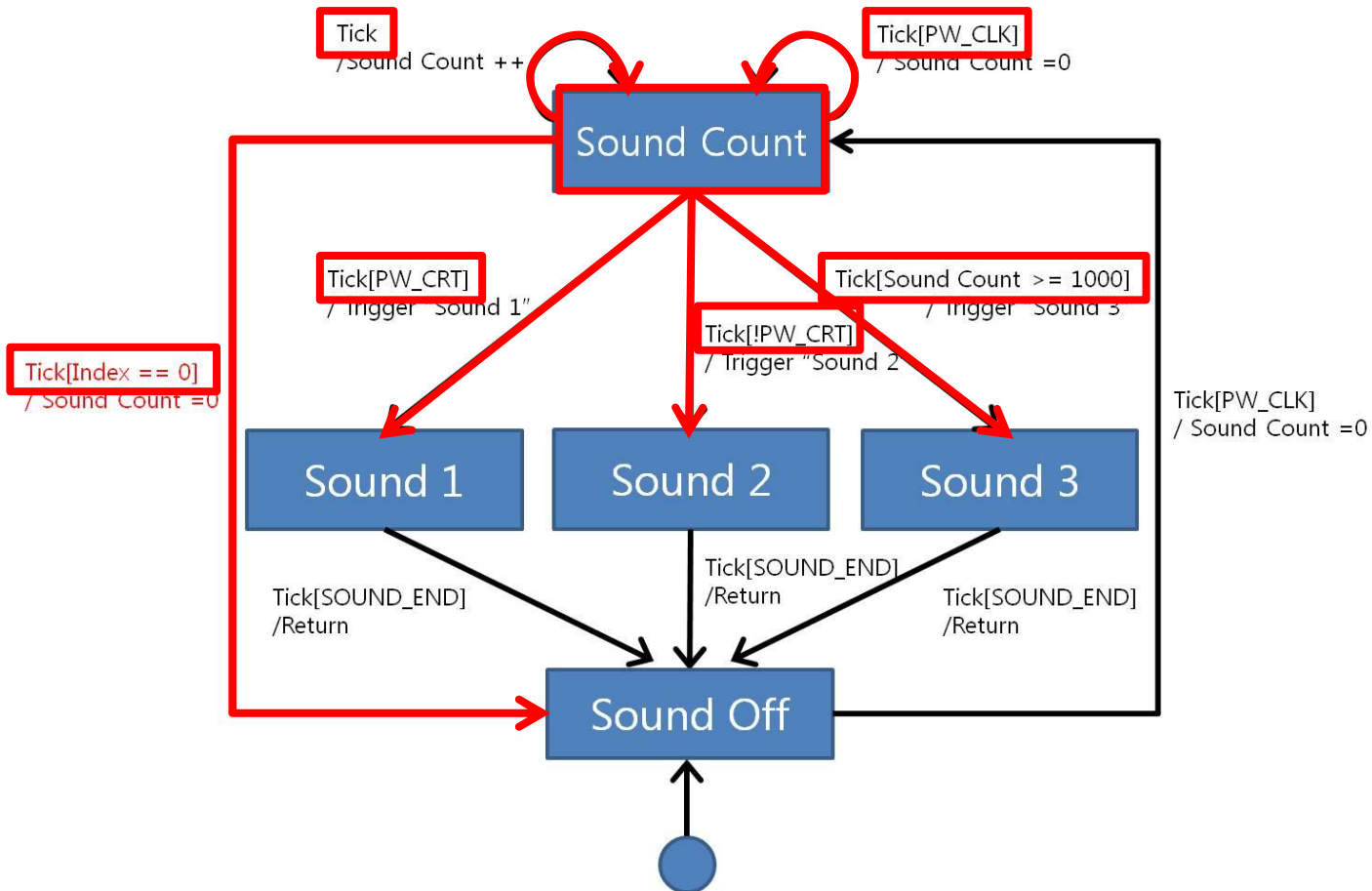
Test Code

1. STD와 Test case간의 차이



Test Code

1. STD와 Test case간의 차이



Test Code

1. STD와 Test case간의 차이

```

case COUNT :
(*pw_count)++; //Tick 조건 : 무조건 증가
if(nbd->num_ck && pwd->index != PW_SIZE) //Count 조건
{
    *pw_clk = pw_click(&nbd->num_ck); //pw_clk 값 반환받아야오기(Trigger)
    *pw_count = 0; //pw_count값 초기화
    pwd->input[pwd->index] = nbd->num_button; //input에 nb 삽입
    (pwd->index)++; //index 증가
    nbd->num_button = NIL; //nb값 초기화(nbc= pw_click에서 초기화 됨)
}
else if(*pw_count >= 1000 || CC || (!ls && pwd->arr != NULL)) //10초 지나거나 커번 압하면
{
    pw_reset(&pwd->index);
    *pw_state = RESET;
}
else if(pwd->index == PW_SIZE && pwd->arr == NULL)
{
    pw_set(pwd);
    *pw_state = SET;
}
else if(pwd->index == PW_SIZE && pwd->arr != NULL)
{
    *pw_crt = pw_check(pwd);
    *pw_state = CHECK;
}
break;

```

Test Code

1. STD와 Test case간의 차이

```

case Sound_Count_Mod :
    (*Sound_Count)++; //tick조건 count 증가

    /*
    pw_crt는 -1,0,1로 가정
    -1 : 아무것도 아님
    0 : PW 불링
    1 : PW 상승
    */
    if(pw_crt==TRUE){
        *state = Sound_1; //상태변화
        // *Sound_Command=Sound1(); //Trigger 'Sound1'
        return Sound1();
    }
    else if(pw_crt==FALSE){
        *state = Sound_2; //상태변화
        // *Sound_Command=Sound2(); //Trigger 'Sound2'
        return Sound2();
    }
    else if(*Sound_Count >= 1000){
        *state = Sound_3; //상태변화
        // *Sound_Command=Sound3(); //Trigger 'Sound3'
        return Sound3();
    }
    else if(pw_clk==TRUE)
        *Sound_Count = 0; //Sound Count 초기화
    else if(index == 0)
        *state = Sound_OFF;
    break;

```

Test Code

2. Trigger 검사

EDLS.UTC_216_005 ↗	State==Lock/LB==True ↗	State=Unlock, Trigger'Unlock' ↗
EDLS.UTC_216_006 ↗	State==Lock/KS==True ↗	State=Unlock, Trigger'Unlock' ↗
EDLS.UTC_216_007 ↗	State==Lock/PW_CRT==True ↗	State=Unlock, Trigger'Unlock' ↗

→ 함수가 실행 됐다는 증거 필요 !

Test Code

- #include, #ifdef, static 변수

```

//////////determine//////////
BSS_DATA * determine(BOOLEAN LB_signal, BOOLEAN DS_signal, BOOLEAN KS_signal, BOOLEAN CS_signal,int NB_signal, BSS_DATA *bss_data);
//////////Lock control//////////
LOCK_CMD * lock_controller(LC_STATE *State, int *Door_Count, STATE_DATA *state,SENSOR_DATA *sens_data, BOOLEAN LB, BOOLEAN pw_crt, LOCK_CMD *cmd);
//////////PW control//////////
int pw_controller(PW_STATE *pw_state, STATE_DATA *lock_and_pw_data, NUM_BUTTON_DATA *numb_data, BOOLEAN CC, BOOLEAN *pw_clk, BOOLEAN *pw_crt, int *pw_count);
//////////Sound control//////////
int sound_controller(SC_STATE *state,int *Sound_Count,BOOLEAN pw_clk,BOOLEAN pw_crt, int index);

```

```

/*
 * cunittest.c
 */

```

```

#include "proto.h"
#include CUNIT_T
#include "Cunit/Cunit.h"
#include "Cunit/Console.h"

```

```

void memset_bss_data(BSS_DATA ** bss_data);
BSS_DATA* free_bss_data(BSS_DATA * bss_data);

```

```

static BSS DATA *g bss data = NULL;
static PWD * g_pwd = NULL;
static int * g_pw_arr = NULL;

```

```

//CUNIT_T를 defines하면 unit test 모드에 들어감
#define CUNIT_T

#ifdef CUNIT_T
char trigger_buffer[512]; //trigger 확인을 위한 전역변수
#endif

```



Test Code

- 전역 변수 메모리 할당 함수

```
//Suite의 초기화 함수
int initialize_data(){
    if(g_bss_data == NULL)
        memset_bss_data(&g_bss_data);

    if(g_pwd==NULL){
        g_pwd = (PWD*)malloc(sizeof(PWD));
        memset(g_pwd,0x0,sizeof(PWD));
    }
    if(g_pw_arr==NULL){
        g_pw_arr = (int*)malloc(sizeof(int)*PW_SIZE);
        memset(g_pw_arr,0x0,sizeof(int)*PW_SIZE);
    }
    return 0;
}
```

Test Code

- 전역 변수 메모리 초기화 함수

```
void memset_bss_data(BSS_DATA ** bss_data){
    if (*bss_data == NULL) {
        //단, NULL일때 즉 한번만 생성
        *bss_data = (BSS_DATA*) malloc(sizeof(BSS_DATA));
        memset(*bss_data, 0x0, sizeof(BSS_DATA));
        //그 내부 구조체 포인터를에도 메모리 할당을 해준 다음에 (이것도 한번만)
        (*bss_data)->numbtn_data = (NUM_BUTTON_DATA*) malloc( sizeof(NUM_BUTTON_DATA));
        memset((*bss_data)->numbtn_data, 0x0, sizeof(NUM_BUTTON_DATA));
        (*bss_data)->state_data = (STATE_DATA*) malloc(sizeof(STATE_DATA));
        memset((*bss_data)->state_data, 0x0, sizeof(STATE_DATA));
        (*bss_data)->state_data->pw_data = (PWD*) malloc(sizeof(PWD));
        memset((*bss_data)->state_data->pw_data, 0x0, sizeof(PWD));
        (*bss_data)->sensor_data = (SENSOR_DATA*) malloc(sizeof(SENSOR_DATA));
        memset((*bss_data)->sensor_data, 0x0, sizeof(SENSOR_DATA));
    }
}
```

Test Code

- 전역 변수 메모리 해제 함수

```
//Suite의 cleanup 함수
int cleanup_data(){
    if(g_bss_data != NULL)
        g_bss_data = free_bss_data(g_bss_data);
    if(g_pwd!=NULL){
        free(g_pwd);
        g_pwd = NULL;
    }
    if(g_pw_arr!=NULL){
        free(g_pw_arr);
        g_pw_arr=NULL;
    }
    return 0;
}
```

Test Code

- 전역 변수 초기화 함수

```
//각 unit test 함수마다 전역변수를 초기화하는 기능
void func_init_data(){
    if(g_bss_data==NULL || g_pwd == NULL || g_pw_arr == NULL){
        printf("\ndata Error!\n");
        return;
    }
    g_bss_data->lockbutton_data = FALSE;
    g_bss_data->numbtn_data->num_button = 0;
    g_bss_data->numbtn_data->num_ck = FALSE;
    g_bss_data->sensor_data->dac_data.cover_closed_data=FALSE;
    g_bss_data->sensor_data->dac_data.cover_opened_data=FALSE;
    g_bss_data->sensor_data->dac_data.door_closed_data=FALSE;
    g_bss_data->sensor_data->dac_data.door_opened_data=FALSE;
    g_bss_data->sensor_data->keysensor = FALSE;
    g_bss_data->state_data->lockstate =FALSE;
    g_bss_data->state_data->pw_data = NULL;

    g_pwd->arr = NULL;
    g_pwd->index = 0;
    memset(g_pwd->input,0x0,sizeof(int)*PW_SIZE);

    memset(g_pw_arr,0x0,sizeof(int)*PW_SIZE);
}
```

Test Code

- 1.6 Determine Signal test code sample

EDLS.UTC_160_012 ↗ 1.6 Determine Signal ↗ Door Sensor Signal==TRUE가 Number Button Signal과 동시에 들어온다. ↗

EDLS.UTC_160_012 ↗ Door Sensor Signal==TRUE/ Number Button Signal==1 ↗ DC==TRUE, NBC==FALSE ↗

```
void EDLS.UTC_160_012()
{
    //DS_signal = TRUE;
    //NB_signal = 1;
    g_bss_data = determine(FALSE, TRUE, FALSE, 0x2, 1, g_bss_data);
    CU_ASSERT_TRUE(g_bss_data->sensor_data->dac_data.door_closed_data);
    CU_ASSERT_FALSE(g_bss_data->numbtn_data->num_ck);
}
```

Test Code

- 2.1.1 PW Controller test code sample

```

//2.1.1 PW Controller
void EDLS.UTC_211_000()
{
    PW_STATE pw_state = READY;
    BOOLEAN pw_clk = NIL;
    int index;
    int pw_count;
    int numb = 1;

    func_init_data(); //전역변수 초기화

    g_bss_data->state_data->pw_data = g_pwd;
    g_bss_data->state_data->lockstate = TRUE;
    g_bss_data->numbtn_data->num_ck = TRUE;
    g_bss_data->numbtn_data->num_button = numb;
    index = g_pwd->index;
    //
    pw_controller(&pw_state,g_bss_data->state_data,g_bss_data->numbtn_data,g_bss_data->sensor_data->dac_data.cover_closed_data,&pw_clk,NULL,&pw_count);

    CU_ASSERT(pw_state == COUNT);
    /*trigger "click"*/
    CU_ASSERT_STRING_EQUAL(trigger_buffer,"click");
    CU_ASSERT(g_pwd->input[index] == numb);
    CU_ASSERT(g_pwd->index == index+1);
    CU_ASSERT(pw_count == 0);

    //
}

```

EDLS.UTC_211_000 ↵ 2.1.1 PW Controller ↵ Ready 상태에서 LS==TRUE 일 때, PW Array==NULL, NBC==TRUE 입력이 들어온다. ↵

EDLS.UTC_211_000 ↵ **1** State==Ready / LS==TRUE, PW Array==NULL,NBC==TRUE ↵ **2** State==Count / Trigger "click", PW Input[Index]==NB, Index++, PW Count==0 ↵

1 g_bss_data->state_data->pw_data = g_pwd;
 g_bss_data->state_data->lockstate = TRUE;
 g_bss_data->numbtn_data->num_ck = TRUE;
 g_bss_data->numbtn_data->num_button = numb;
 index = g_pwd->index;
 //
 pw_controller(&pw_state,g_bss_data->state_data,g_bss_data->numbtn_data,g_bss_data->sensor_data->dac_data.cover_closed_data,&pw_clk,NULL,&pw_count);

2 CU_ASSERT(pw_state == COUNT);
 /*trigger "click"*/
 CU_ASSERT_STRING_EQUAL(trigger_buffer,"click");
 CU_ASSERT(g_pwd->input[index] == numb);
 CU_ASSERT(g_pwd->index == index+1);
 CU_ASSERT(pw_count == 0);

Test Code

- 2.1.2 PW Set & 2.1.3 PW Check test code

EDLS.UTC_212_000 ↵	2.1.2 PW Reset ↵	Trigger 입력이 들어오면, PW Input Data 를 NULL 로 만든다. ↵
EDLS.UTC_213_000 ↵	2.1.3 PW Check ↵	Trigger 입력이 들어오면, PW Input Data 와 PW Array 를 비교해 PW Correct 값을 변경한다. ↵
EDLS.UTC_212_000 ↵	Trigger in ↵	<u>PW Index == 0</u> ↵
EDLS.UTC_213_000 ↵	Trigger in ↵	<u>PW Correct = (PW Input==PW Array)</u> ↵

```

void EDLS.UTC_213_000(){
    BOOLEAN pw_crt = NIL;
    int i=0;
    BOOLEAN t = TRUE;

    func_init_data(); //전역변수 초기화

    g_pwd->arr = g_pw_arr;

    for(i=0;i<PW_SIZE;i++){
        (g_pwd->arr)[i] = rand()%2;
        (g_pwd->input)[i] = rand()%2;
        if(t==TRUE)
            t = (g_pwd->arr)[i]==(g_pwd->input)[i];
    }
    pw_crt = pw_check(g_pwd);

    CU_ASSERT(pw_crt == t);
}

```

```

void EDLS.UTC_212_000(){
    int index = rand()%4+1;

    pw_reset(&index);

    CU_ASSERT(index == 0);
}

```


Test Code

- 2.1.6 Lock Controller test code sample

EDLS.UTC_216_002 ↗ 2.1.6 Lock Controller ↗ Unlock 상태에서 PW Array!=Null 와 LB==True, DC==False 입력이 들어온다 ↗

```
void EDLS.UTC_216_002(){
    LOCK_CMD lock_command = {0};
    LC_STATE state = 0;
    int door_count = 0;
    BOOLEAN LB = FALSE;
    BOOLEAN PW_CRT = NIL;
    int i;

    func_init_data(); //점연변수 초기화

    g_bss_data->state_data->pw_data = g_pwd;
    g_pwd->arr = g_pw_arr;

    for(i = 0; i < PW_SIZE; i++)
        (g_bss_data->state_data->pw_data->arr)[i] = 0;

    state=Unlock;
    LB=TRUE;
    g_bss_data->sensor_data->dac_data.door_closed_data=FALSE;

    lock_controller(&state,&door_count,g_bss_data->state_data,g_bss_data->sensor_data,LB,PW_CRT,&lock_command);

    CU_ASSERT(state==Lock);
    //trigger ↗
    CU_ASSERT_STRING_EQUAL(trigger_buffer,"lock");
}
```

EDLS.UTC_216_002 ↗ State==Unlock/PW Array!=Null, State=Lock, Trigger'Lock' ↗
LB==True,DC==False ↗

```
//Lock Process/
LOCK_CMD *LockP([LOCK_CMD *Loc])
{
    Loc->lockstate = TRUE;
    Loc->lock_command = TRUE;
#ifdef CUNIT_T
    //trigger ↗
    memset(trigger_buffer,0x0,sizeof(trigger_buffer));
    strncpy(trigger_buffer,"lock",20);
#endif
    return Loc;
}
```

Test Code

- 2.2.1 Sound & 2.2.5 Light Controller ...

```

sound_controller(&state,&Sound_Count,pw_clk,pw_crt,index); //실패

CU_ASSERT(state==Sound_2);
//CU_ASSERT('트리거 조건');
CU_ASSERT_STRING_EQUAL(trigger_buffer,"sound2");
}
void EDLS_UTC_221_007(){//Trigger "Sound_3"이 out put에있는데 input trigger 부분을 실패
SC_STATE state=Sound_OFF;
int Sound_Count = 0;
BOOLEAN pw_clk=FALSE;
BOOLEAN pw_crt=NIL;
int index=0;

state=Sound_Count_Mod;//실패변화
Sound_Count=1000;//조건이 1000이상이면 1000으로 변화

sound_controller(&state,&Sound_Count,pw_clk,pw_crt,index);

CU_ASSERT(state==Sound_3);
//CU_ASSERT('트리거 조건');
CU_ASSERT_STRING_EQUAL(trigger_buffer,"sound3");
}
void EDLS_UTC_221_008(){
SC_STATE state=Sound_OFF;
int Sound_Count = 0;
BOOLEAN pw_clk=FALSE;
BOOLEAN pw_crt=NIL;
int index=1;

state=Sound_Count_Mod;//실패변화, 조건이 sound_count이하이면, sound_count를 0으로
sound_controller(&state,&Sound_Count,pw_clk,pw_crt,index);

CU_ASSERT(state==Sound_Count_Mod);
}
void EDLS_UTC_221_009(){
SC_STATE state=Sound_OFF;
int Sound_Count = 0;

void EDLS_UTC_222_001(){
int i;
LTC_STATE state=Light_ON;//첫 시작이 Light On이어서 Light_on으로 초기화
int Light_Count=0;//light_controller의 매개변수 참고해서 만들고 초기화
BOOLEAN CO=FALSE;
BOOLEAN CC=FALSE;
BOOLEAN pw_clk=FALSE;
BOOLEAN light_cmd=FALSE;

func_init_data(); //조건변수 초기화

g_bss_data->state_data->pw_data = g_pwd;
g_pwd->arr = g_pw_arr;

for(i = 0; i < PW_SIZE; i++){//pw_array!=null을 위해서 0을 삽입
(g_bss_data->state_data->pw_data->arr)[i] = 0;
}
pw_clk=FALSE;//클럭 false

light_controller(&state,&Light_Count,CO,CC,pw_clk,g_bss_data->state_data->pw_data,light_cmd);//합수 실패
CU_ASSERT(state==Light_ON);
}
void EDLS_UTC_222_002(){
LTC_STATE state=Light_ON;//첫 시작이 Light On이어서 Light_on으로 초기화
int Light_Count=0;//light_controller의 매개변수 참고해서 만들고 초기화
BOOLEAN CO=FALSE;
BOOLEAN CC=FALSE;
BOOLEAN pw_clk=FALSE;
BOOLEAN light_cmd=FALSE;

func_init_data(); //조건변수 초기화

pw_clk=FALSE;//클럭 false, pw_array=NULL

light_controller(&state,&Light_Count,CO,CC,pw_clk,g_pwd,light_cmd);//합수 실패
CU_ASSERT(state==Light_ON);
}

```

Test Code

- Main 함수
- Case Suite

```
CU_TestInfo test_arr4[] = {
    {"EDLS.UTC_222_000", EDLS.UTC_222_000},
    {"EDLS.UTC_222_001", EDLS.UTC_222_001},
    {"EDLS.UTC_222_002", EDLS.UTC_222_002},
    {"EDLS.UTC_222_003", EDLS.UTC_222_003},
    {"EDLS.UTC_222_004", EDLS.UTC_222_004},
    {"EDLS.UTC_222_005", EDLS.UTC_222_005},
    {"EDLS.UTC_222_006", EDLS.UTC_222_006},
    {"EDLS.UTC_222_007", EDLS.UTC_222_007},
    {"EDLS.UTC_222_008", EDLS.UTC_222_008},
    {"EDLS.UTC_222_009", EDLS.UTC_222_009},
    {"EDLS.UTC_222_010", EDLS.UTC_222_010},
    {"EDLS.UTC_222_011", EDLS.UTC_222_011},
    {"EDLS.UTC_222_012", EDLS.UTC_222_012},
    {"EDLS.UTC_222_013", EDLS.UTC_222_013},
    {"EDLS.UTC_222_014", EDLS.UTC_222_014},
    {"EDLS.UTC_226_000", EDLS.UTC_226_000},
    CU_TEST_INFO_NULL,
};
```

```
CU_SuiteInfo suite_arr[] = {
    {"1.6. Determine Signal", NULL, NULL, test_arr5},
    {"2.1.1 PW_Controller", initialize_data, cleanup_data, test_arr},
    {"2.1.6 Lock_Controller", initialize_data, cleanup_data, test_arr2},
    {"2.2.1 Sound_Controller", NULL, NULL, test_arr3},
    {"2.2.2 Light_Controller", initialize_data, cleanup_data, test_arr4},
    CU_SUITE_INFO_NULL,
};
```

Unit Test Result

- Cunit 실행

CUnit - A Unit testing framework for C - Version 2.1-0
<http://cunit.sourceforge.net/>

```
***** CUNIT CONSOLE - MAIN MENU *****  
(R)un all, (S)elect suite, (L)ist suites, Show (F)ailures, (Q)uit  
Enter Command : r
```

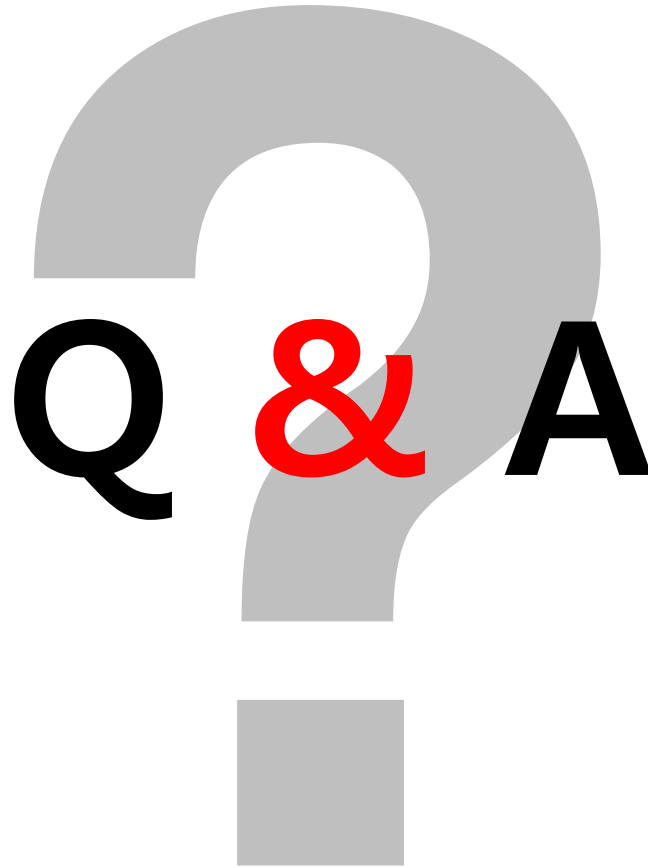
```
Running Suite : 1.6. Determine Signal  
  Running test : EDLS.UTC_160_000  
  Running test : EDLS.UTC_160_001  
  Running test : EDLS.UTC_160_002  
  Running test : EDLS.UTC_160_003  
  Running test : EDLS.UTC_160_004  
  Running test : EDLS.UTC_160_005  
  Running test : EDLS.UTC_160_006  
  Running test : EDLS.UTC_160_007  
  Running test : EDLS.UTC_160_008  
  Running test : EDLS.UTC_160_009  
  Running test : EDLS.UTC_160_010  
  Running test : EDLS.UTC_160_011  
  Running test : EDLS.UTC_160_012  
  Running test : EDLS.UTC_160_013  
  Running test : EDLS.UTC_160_014  
  Running test : EDLS.UTC_160_015  
  Running test : EDLS.UTC_160_016  
  Running test : EDLS.UTC_160_017  
  Running test : EDLS.UTC_160_018  
Running Suite : 2.1.1 PW_Controller  
  Running test : EDLS.UTC_211_000  
  Running test : EDLS.UTC_211_001  
  Running test : EDLS.UTC_211_002  
  Running test : EDLS.UTC_211_003  
  Running test : EDLS.UTC_211_004  
  Running test : EDLS.UTC_211_005  
  Running test : EDLS.UTC_211_006
```

Unit Test Result

- 최종 결과

```
--Run Summary: Type      Total      Ran      Passed    Failed
                 suites         5         5         n/a       0
                 tests       109       109       109       0
                 asserts     185       185       185       0
```

→ **No Fail !**





Thank You