

UTR

PRESENTATION

Team#1

201211371 이유민

201211338 김주호

201111352 박소은

201211346 박선민

INDEX

- > NOTE**
- > Environments**
- > References**
- > Unit Test Case Specification
& Test Code**
- > Test Result**



01

Note

01Note

- ✓ 설계한 구조와 실제 구현한 구조가 많이 다름.
 - ✓ **SRA, SDS, UTP**
실제 구현할 때 필요한 input 과 output을 추가하고,
필요 없는 인자 값들은 제거함.
 - ✓ **Source Code**
수정된 SRA와 SDS를 기반으로 수정.
 - ✓ **Unit Test case**
수정된 Source Code를 기반으로 수정.



02

Environment

02 Environmets

- ✓ Window
- ✓ Eclipse IDE
- ✓ MinGW
- ✓ CUnit unit test framework for C



03

References

03References

- ✓ Team1-2013.EDLS.SRS - 1.0
- ✓ Team1 - 2013.EDLS.SRA - 3.0
- ✓ Team1 - 2013.EDLS.SDS - 3.0
- ✓ Team1 - 2013.EDLS.UTP - 2.0

A large red circle is positioned on the left side of the slide, partially overlapping the text. The number '04' is written in white inside the circle.

04

Unit Test Case Specification & Test Code

04-0 Index

- ✓ 04-1 Lock Controller
- ✓ 04-2 Display Controller
- ✓ 04-3 Alarm Controller
- ✓ 04-4 Backlight Controller
- ✓ 04-5 Password Controller

04-1 Lock Controller

✓ Test Case Specification Identification

Identifier	Feature(Process ID in DFD)	Valid/Invalid value
EDLS.UTC_000_000	2.1.1 Lock Controller	Lock Mode에서 [BO]==FALSE Input
EDLS.UTC_000_001	2.1.1 Lock Controller	Lock Mode에서 [L]==TURE&&[C]==FALSE 일 때 [K]==TRUE Input
EDLS.UTC_000_002	2.1.1 Lock Controller	Unlock Mode에서[L]==FALSE&&[D]==FALSE일때[LB]==TRUE Input
EDLS.UTC_000_003	2.1.1 Lock Controller	Unlock Mode에서 [BO]==TRUE Input
EDLS.UTC_000_004	2.1.1 Lock Controller	Unlock Mode에서 [N]==FALSE Input
EDLS.UTC_000_005	2.1.1 Lock Controller	Lock Mode에서 [L]==TRUE일 때[C]==FALSE &&[P]==TRUE Input
EDLS.UTC_000_006	2.1.1 Lock Controller	Lock Mode에서 [L]==TRUE일 때 [C]==FALSE&&[K]==TRUE Input
EDLS.UTC_001_000	2.1.2 Locking	Enable Input

04-1 Lock Controller

✓ Test Items

Test case Identifier	Input Specification	Output Specification
EDLS.UTC_000_000	State=Lock/ [BO]==FALSE	State=Lock
EDLS.UTC_000_001	State=Lock&&[C]==FLASE&&[L]==TRUE/ [K]==T URE	Disable / Locking==1
EDLS.UTC_000_002	State=Unlock&&[L]==FLASE &&[D]==FALSE/ [LB]==TRUE	Enable / Locking==1
EDLS.UTC_000_003	State=Unlock/ [BO]==TRUE	State=Unlock
EDLS.UTC_000_004	State=Unlock/ [N]==FALSE	State=Unlock
EDLS.UTC_000_005	State=Lock&&[L]==TURE/ [C]==FALSE&&[P]==TRUE	Enable / Locking==1
EDLS.UTC_000_006	State=Lock&&[L]==TRUE/ [C]==TRUE&& [K]==TRUE	Enable / Locking==1
EDLS.UTC_001_000	Enable	[L], Lock command

04-1 Lock Controller

✓ Test Code – Invalid case

```
EDLS.UTC_000_000()
{
    int tick=0;

    KEY_SENSOR* key = (KEY_SENSOR*)calloc(1,sizeof(KEY_SENSOR));
    DOOR_SENSOR* door = (DOOR_SENSOR*)calloc(1,sizeof(DOOR_SENSOR));
    COVER_SENSOR* cover = (COVER_SENSOR*)calloc(1,sizeof(COVER_SENSOR));
    LOCKBUTTON_SENSOR* lockbutton = (LOCKBUTTON_SENSOR*)calloc(1,sizeof(LOCKBUTTON_SENSOR));

    STATE_DATA * state = (STATE_DATA*)calloc(1,sizeof(STATE_DATA));
    PASSWORD_DATA* password =(PASSWORD_DATA*)calloc(1,sizeof(PASSWORD_DATA)) ;

    door->D=FALSE;
    state->L=TRUE;
    lockbutton->LB=FALSE;
    key->K=WAIT;
    state->BO=FALSE;
    password->P=WAIT;
    cover->C=TRUE;

    input=lock_controller(state,password,key,door,cover,lockbutton,tick);

    CU_ASSERT(input==1); //LockMode
}
```

04-1 Lock Controller

✓ Test Code – valid case

```
EDLS.UTC_000_001()
{
    int tick=0;

    KEY_SENSOR* key = (KEY_SENSOR*)calloc(1,sizeof(KEY_SENSOR));
    DOOR_SENSOR* door = (DOOR_SENSOR*)calloc(1,sizeof(DOOR_SENSOR));
    COVER_SENSOR* cover = (COVER_SENSOR*)calloc(1,sizeof(COVER_SENSOR));
    LOCKBUTTON_SENSOR* lockbutton = (LOCKBUTTON_SENSOR*)calloc(1,sizeof(LOCKBUTTON_SENSOR));

    STATE_DATA * state = (STATE_DATA*)calloc(1,sizeof(STATE_DATA));
    PASSWORD_DATA* password =(PASSWORD_DATA*)calloc(1,sizeof(PASSWORD_DATA)) ;

    door->D=FALSE;
    state->L=TRUE;
    lockbutton->LB=FALSE;

    key->K=TRUE;
    cover->C=FALSE;

    input=lock_controller(state,password,key,door,cover,lockbutton,tick);

    CU_ASSERT(input==DISABLE);
}
```

04-2 Display Controller

✓ Test Case Specification Identification

Identifier	Feature(Process ID in DFD)	Valid/Invalid value
EDLS.UTC_002_000	2.2.1 Display Controller	Yellow Mode에서 [C]==TRUE && [K]==TRUE Input
EDLS.UTC_002_001	2.2.1 Display Controller	Yellow Mode에서 [C]==FALSE && [K]==TRUE Input
EDLS.UTC_002_002	2.2.1 Display Controller	Yellow Mode에서 [C]==FALSE && [K]==FALSE Input
EDLS.UTC_002_003	2.2.1 Display Controller	Yellow Mode에서 [C]==TRUE && [K]==FALSE Input
EDLS.UTC_005_000	2.2.4 Yellow color	Enable Input
EDLS.UTC_006_000	2.2.5 Red color	Trigger Input
EDLS.UTC_007_000	2.2.6 Green color	Trigger Input

04-2 Display Controller

✓ Test Items

Test case Identifier	Input Specification	Output Specification
EDLS.UTC_002_000	State==Yellow/ [C]==TRUE&&[K]==TRUE	Enable / State == Yellow
EDLS.UTC_002_001	State==Yellow/ [C]==FALSE&&[K]==TRUE	Trigger / State == Green
EDLS.UTC_002_002	State==Yellow/ [C]==FALSE&&[K]==FALSE	Trigger / State == Red
EDLS.UTC_002_003	State==Yellow/ [C]==TRUE&&[K]==FALSE	Enable / State == Yellow
EDLS.UTC_005_000	Enable	Yellow color command
EDLS.UTC_006_000	Red Color Trigger	Red color command
EDLS.UTC_007_000	Green Color Trigger	Green color command

04-2 Display Controller

✓ Test Code – valid case

```
void EDLS.UTC_002_001(){  
  
    KEY_SENSOR* key = (KEY_SENSOR*)calloc(1,sizeof(KEY_SENSOR));  
    COVER_SENSOR* cover = (COVER_SENSOR*)calloc(1,sizeof(COVER_SENSOR));  
    key->K=TRUE;  
    cover->C=FALSE;  
    input=display_controller(key, cover);  
  
    CU_ASSERT(input == GREEN);  
  
    //reset  
    key->K = WAIT;  
    cover->C = FALSE;  
}
```

04-3 Alarm Controller

✓ Test Case Specification Identification

Identifier	Feature(Process ID in DFD)	Valid/Invalid value
EDLS.UTC_003_000	2.2.2 Alarm Controller	Wait Mode에서 [L]==TURE Input
EDLS.UTC_003_001	2.2.2 Alarm Controller	Wait Mode에서 [AC]==TURE Input
EDLS.UTC_003_002	2.2.2 Alarm Controller	Wait Mode에서 [Input_End]==TURE Input
EDLS.UTC_003_003	2.2.2 Alarm Controller	Wait Mode에서 [C_flag]==0 Input
EDLS.UTC_003_004	2.2.2 Alarm Controller	Wait Mode에서 [C]==FALSE&&[P]==FALSE Input
EDLS.UTC_003_005	2.2.2 Alarm Controller	Wait Mode에서 [BO]==TRUE Input
EDLS.UTC_003_006	2.2.2 Alarm Controller	Wait Mode에서 [C]==FALSE&&[P]==TRUE Input
EDLS.UTC_008_000	2.2.7 Alarm1	Trigger Input
EDLS.UTC_009_000	2.2.8 Alarm2	Trigger Input
EDLS.UTC_010_000	2.2.9 Alarm3	Trigger Input

04-3 Alarm Controller

✓ Test items

Test case Identifier	Input Specification	Output Specification
EDLS.UTC_003_000	State==Wait/ [L]==TRUE	State == Wait
EDLS.UTC_003_001	State==Wait / [AC]==TURE	State == Wait
EDLS.UTC_003_002	State==Wait / [Input_End]==TRUE	State == Wait
EDLS.UTC_003_003	State==Wait / [C_flag]==0	State == Wait
EDLS.UTC_003_004	State==Wait/ [C]==FALSE&&[P]==FLASE	Trigger/ State == Error Sound
EDLS.UTC_003_005	State==Wait/ [BO]==TRUE	Trigger/ State == Warning Sound
EDLS.UTC_003_006	State==Wait/ [C]==FALSE&&[P]==TRUE	Trigger/ State==Correct Sound
EDLS.UTC_008_000	Alarm1 Trigger	Alarm1 command
EDLS.UTC_009_000	Alarm2 Trigger	Alarm2 command
EDLS.UTC_010_000	Alarm3 Trigger	Alarm3 command

04-3 Alarm Controller

✓ Test Code – invalid case

```
void EDLS_UTC_003_000(){
    COVER_SENSOR* cover = (COVER_SENSOR*)calloc(1,sizeof(COVER_SENSOR));
    STATE_DATA * state = (STATE_DATA*)calloc(1,sizeof(STATE_DATA));
    PASSWORD_DATA* password =(PASSWORD_DATA*)calloc(1,sizeof(PASSWORD_DATA)) ;
    //set Wait
    password->P=WAIT;
    state->L=TRUE;
                                cover->C=FALSE;
    input=alarm_controller(password,state,cover);

    CU_ASSERT(input==ALARM1);

    //reset
    password->P = WAIT;
    cover->C = FALSE;
}
```

04-3 Alarm Controller

✓ Test Code – valid case

```
void EDLS.UTC_003_004(){  
  
    COVER_SENSOR* cover = (COVER_SENSOR*)calloc(1,sizeof(COVER_SENSOR));  
    STATE_DATA * state = (STATE_DATA*)calloc(1,sizeof(STATE_DATA));  
    PASSWORD_DATA* password =(PASSWORD_DATA*)calloc(1,sizeof(PASSWORD_DATA)) ;  
    //set Wait  
    password->P=FALSE;  
    state->L=TRUE;  
    cover->C=FALSE;  
    input=alarm_controller(password, state,cover);  
  
    CU_ASSERT(input == ALARM3);  
  
    //reset  
    password->P = WAIT;  
    cover->C = FALSE;  
}
```

04-4 Backlight Controller

✓ Test Case Specification Identifier

Identifier	Feature(Process ID in DFD)	Valid/ Invalid value
EDLS.UTC_004_000	2.2.3 Backlight Controller	Backlight off Mode에서 [C_flag]==0&&[C]==TRUE일 때[N]==TURE Input
EDLS.UTC_004_001	2.2.3 Backlight Controller	Backlight off Mode에서 [C]==TRUE&&[N]==TRUE일 때 [C_flag]==0] Input
EDLS.UTC_004_002	2.2.3 Backlight Controller	Backlight off Mode에서 [K]==TURE Input
EDLS.UTC_004_003	2.2.3 Backlight Controller	Backlight on Mode에서 [C]==FALSE&&[BO]==FALSE일 때 [N]==FALSE Input
EDLS.UTC_004_004	2.2.3 Backlight Controller	Backlight on Mode에서 [C]==FALSE&&[N]==FALSE일 때 [BO]==TRUE Input
EDLS.UTC_011_000	2.2.10 Light On	Enable Input

04-4 Backlight Controller

✓ Test items

Test case Identifier	Input Specification	Output Specification
EDLS.UTC_004_000	State == Backlight Off&& [C_flag]==0&&[C]==TRUE/ [N]==TURE	Enable/ State==Backlight On
EDLS.UTC_004_001	State==Backlight Off&&[C]==TURE&& [N]==TURE/ [C_flag]==0]	Enable/ State==Backlight On
EDLS.UTC_004_002	State==Backlight Off/ [K]==TURE	Disable/ State==Backlight Off
EDLS.UTC_004_003	State==Backlight On&& [C]==FALSE&&[BO]==FALSE/ [N]==FALSE	Tick counting==0/ State==Backlight Off
EDLS.UTC_004_004	State==Backlight On&& [C]==FALSE&&[N]==FLASE/ [C_flag]==1]	Tick counting++
EDLS.UTC_011_000	State==Disable/ Enable	Light On Command/ State==Enable

04-4 Backlight Controller

✓ Test Code – invalid case

```
void EDLS.UTC_004_002() //Backlight off Mode에서 [K]==TURE Input
{
    KEY_SENSOR* key = (KEY_SENSOR*)calloc(1,sizeof(KEY_SENSOR));
    COVER_SENSOR* cover = (COVER_SENSOR*)calloc(1,sizeof(COVER_SENSOR));
    STATE_DATA * state = (STATE_DATA*)calloc(1,sizeof(STATE_DATA));
    PASSWORD_DATA* password =(PASSWORD_DATA*)calloc(1,sizeof(PASSWORD_DATA)) ;
    unsigned int tick = 0 ;
    state->BO = TRUE;
    C_flag = 1;
    cover->C = FALSE;
    key->K = TRUE;
    CU_ASSERT_TRUE(backlight_controller(state, password, cover, key, tick));
}
```


04-4 Backlight Controller

✓ Test Code – valid case

```
void EDLS.UTC_004_000() //Backlight off Mode에서 [C_flag]=0&&[C]==TRUE일 때 [N]==TURE Input
{
    KEY_SENSOR* key = (KEY_SENSOR*)calloc(1,sizeof(KEY_SENSOR));
    COVER_SENSOR* cover = (COVER_SENSOR*)calloc(1,sizeof(COVER_SENSOR));
    STATE_DATA * state = (STATE_DATA*)calloc(1,sizeof(STATE_DATA));
    PASSWORD_DATA* password =(PASSWORD_DATA*)calloc(1,sizeof(PASSWORD_DATA)) ;
    unsigned int tick = 0 ;
    state->BO = TRUE;
    C_flag = 0;
    cover->C = TRUE;
    password->N = TRUE;
    CU_ASSERT_TRUE(backlight_controller(state, password, cover, key, tick));
}
```

04-5 Password Controller

✓ Test Case Specification Identifier

Identifier	Feature(Process ID in DFD)	Valid/ Invalid value
EDLS_UTC_012_000	2.3.1 Password Controller	Wait Mode에서 [L]==TRUE&&[C]==FALSE &&[INPUT_END]==FALSE Input
EDLS_UTC_012_001	2.3.1 Password Controller	Wait Mode에서 [L]==TRUE&&[C]==FALSE &&[INPUT_END]==TRUE Input
EDLS_UTC_012_002	2.3.1 Password Controller	[Password] == NULL Input
EDLS_UTC_012_003	2.3.1 Password Controller	Wait Mode에서 [LB] == FALSE Input
EDLS_UTC_012_004	2.3.1 Password Controller	Wait Mode에서 [K]==TRUE Input
EDLS_UTC_012_005	2.3.1 Password Controller	Password Setting Mode에서 [K]==TRUE Input
EDLS_UTC_012_006	2.3.1 Password Controller	Password Setting Mode에서 [LB]==TRUE Input
EDLS_UTC_013_000	2.3.2 Compare Password	Trigger Input
EDLS_UTC_014_000	2.3.3 Reset Password	Trigger Input

04-5 Password Controller

✓ Test items

Test case Identifier	Input Specification	Output Specification
EDLS.UTC_012_000	State == wait/ [L]==TRUE&&[C]==FALSE&& [INPUT_END]==FALSE	Trigger "Reset Password"/ State == Reset Mode
EDLS.UTC_012_001	State==wait/ [L]==TRUE&&[C]==FALSE&& [INPUT_END]==TURE	Trigger / "Compare Password"/ State==Compare Mode
EDLS.UTC_012_002	State==wait/ [PASSWORD]==NULL	State==Password Setting Mode
EDLS.UTC_012_003	State==wait/ [LB]==FALSE	State==wait
EDLS.UTC_012_004	State==wait/ [K]==TRUE	State==wait
EDLS.UTC_012_005	State==Password Setting/ [K]==TRUE	State==Wait
EDLS.UTC_012_006	State==Password Setting/ [LB]==TRUE	State==Wait
EDLS.UTC_013_000	Reset password Trigger	[P]
EDLS.UTC_014_000	Trigger	INPUT_PASSWORD==NULL

04-5 Password Controller

✓ Test Code – invalid case

```
void EDLS.UTC_012_003()
{
    D == FALSE ;

    P == WAIT ; // WAIT 상태일때 //

    result = Password_controller(L == FALSE,C==TRUE ); // invalid 값
    CU-ASSERT(output == result) ;
    D == FALSE ;

}
```

04-5 Password Controller

✓ Test Code – valid case

```
void EDLS.UTC_012_000()
{
    D == FALSE ;

    P == WAIT ; // WAIT 상태일때 //

    result = Password_controller(L == TRUE,C==FALSE ,INPUT_END == FALSE); // resetpassword trigger 시킴
    CU-ASSERT(output == result) ;
    D == FALSE ;
}
```



05

Test Result

06 Test result

✓ Run

The screenshot shows the Eclipse IDE interface. The main editor displays the following C code:

```
input=alarm_controller(password, state, cover);

CU_ASSERT(input == ALARM2);

//reset
password->P = WAIT;
cover->C = FALSE;
}

void EDLS_UTC_003_006(){

COVER_SENSOR* cover = (COVER_SENSOR*)calloc(1,sizeof(COVER_SENSOR));
STATE_DATA * state = (STATE_DATA*)calloc(1,sizeof(STATE_DATA));
PASSWORD_DATA* password =(PASSWORD_DATA*)calloc(1,sizeof(PASSWORD_DATA));
//set Wait
password->P=TRUE; state->L=TRUE; cover->C==FALSE;
}
```

The Console window shows the following output:

```
DDLUnitTest.exe [C/C++ Application] S:\sunmin#\sunmin#\eclipse-cpp-kepler-SR1-win32#\workspace#\DDLUnitTest#\Debug#\DDLUnitTest.exe (13. 12. 6. 오후 3:13)
Running Test : EDLS_UTC_012_006()
Running Test : EDLS_UTC_013_000()
Running Test : EDLS_UTC_014_000()

Run Summary:  Type  Total  Ran  Passed  Failed  Inactive
              suites  1      1  n/a      0      0
              tests  40     40  32      8      0
              asserts 23     23  15      8      n/a

Elapsed time = 0.317 seconds

***** CUNIT CONSOLE - MAIN MENU *****
(R)un (S)elect (L)ist (A)ctivate (F)ailures (O)ptions (H)elp (Q)uit
Enter command:
```

The Project Explorer on the left shows the project structure for DDLUnitTest, including src files like data.h, EDLS.C, EDLSUnitTest.h, and func.h. The Properties window on the right shows the project's configuration, including the C/C++ compiler and linker settings.

06 Test result

✓ Fail

The screenshot shows the Eclipse IDE interface. The main editor displays the source code for `DDLUnitTest.c`. The code defines two functions: `yellow_color` and `red_color`. The `yellow_color` function checks if a condition is `ENABLE` (printing "Yellow Display") or `DISABLE` (printing "Yellow Display Off"). The `red_color` function checks if a trigger is `TRIGGER` (printing "Red Display") and then sleeps for 100ms.

The console output shows the test results for `DDLUnitTest.exe`. It lists 8 failures, each corresponding to a specific test case. The failures are:

1. ..\src\DDLUnitTest000.c:412 : ([CUnitTest] : EDLS_UTC_002_002()) : input == RED
2. ..\src\DDLUnitTest000.c:432 : ([CUnitTest] : EDLS_UTC_002_003()) : input == YELLOW
3. ..\src\DDLUnitTest000.c:449 : ([CUnitTest] : EDLS_UTC_003_000()) : input==ALARM1
4. ..\src\DDLUnitTest000.c:482 : ([CUnitTest] : EDLS_UTC_003_002()) : state == ALARM0
5. ..\src\DDLUnitTest000.c:502 : ([CUnitTest] : EDLS_UTC_003_003()) : state == ALARM0
6. ..\src\DDLUnitTest000.c:522 : ([CUnitTest] : EDLS_UTC_003_004()) : state == ALARM3
7. ..\src\DDLUnitTest000.c:538 : ([CUnitTest] : EDLS_UTC_003_005()) : input == ALARM2
8. ..\src\DDLUnitTest000.c:603 : ([CUnitTest] : EDLS_UTC_004_002()) : CU_ASSERT_TRUE(backlight_controller(state, password, cover, key, tick))

Total Number of Failures : 8

***** CUNIT CONSOLE - MAIN MENU *****
(R)un (S)elect (L)ist (A)ctivate (F)ailures (O)ptions (H)elp (Q)uit
Enter command:

06 Test result

- ✓ Fail된 부분
Alarm / Display / Backlight Controller
- ✓ 시스템 테스트 할 때, 에러가 났던 부분.
- ✓ Unit testing 을 통해 구체적으로 어떤 부분에서 에러가 나는지 알게 됨.
- ✓ 수정

Q&A

THANK YOU