

SOFTWARE ENGINEERING

Practice #5 (Cunit Test) – Electronic Door Lock

201114188 김종연
201114191 정재욱
201114192 정재철
201114195 홍호탁

www.themegallery.com

Contents

1

개발환경

2

Unit Test Plan

3

Source Code

- ❖ **Eclipse IDE for C/C++ Developers**
- ❖ **MinGW 4.7.0**
- ❖ **CUnit-2.1-2**

Test Function

❖ **int BacklightControl**

int Number(입력 숫자),
int CoverState(개폐 여부),
int tick(시간)

❖ **int LockControl**

int Number(입력 숫자), **int KeyState**(카드키),
int LockState(수동 잠금 버튼),
int DoorState(문 여닫 상태), **int tick**(시간)

Test Case Identification (1/7)

```
void EDLS.UTC_000_000() {  
    state = WAIT_SET_PWD;  
    InNumCount = 4;  
    BacklightControl(NONE, TRUE, 2);  
  
    CU_ASSERT_EQUAL(state, PWD_SET);  
}  
  
void EDLS.UTC_000_001(){  
    state = WAIT_SET_PWD;  
    InNumCount = 3;  
    BacklightControl(NONE, TRUE, 2);  
  
    CU_ASSERT_EQUAL(state, WAIT_SET_PWD);  
}
```

STATE 값을 TEST CASE와
동일하게 설정하고,
버튼 입력 4개를 실시간으로
확인할 수 없어,
InNumCount = Len(n)

Test case identifier	Input specification	Output specification
EDLS.UTC_000_000	state == WAIT_SET_PWD/ Len(N) == 4	state == PWD_SET
EDLS.UTC_000_001	state == WAIT_SET_PWD/ Len(N) == 3	state == WAIT_SET_PWD

Test Case Identification (2/7)

```
void EDSL_UTC_000_002(){
    state = PWD_SET;
    BacklightControl( 1 , TRUE , 0);
    CU_ASSERT_EQUAL(state, TURN_ON);
}
void EDSL_UTC_000_003(){

    state = PWD_SET;
    BacklightControl( 1 , TRUE , 5);
    CU_ASSERT_EQUAL(state, TURN_ON);
}
void EDSL_UTC_000_004(){
    state = PWD_SET;
    BacklightControl( 1 , TRUE , 9);
    CU_ASSERT_EQUAL(state, TURN_ON);
}
void EDSL_UTC_000_005(){
    state = PWD_SET;
    BacklightControl( NONE , TRUE , 5);
    CU_ASSERT_EQUAL(state, PWD_SET);
}
void EDSL_UTC_000_006(){ //
    state = PWD_SET;

    TickCount = 10*TickPerSec;
    CU_ASSERT_EQUAL(BacklightControl( 1 , TRUE , 10), FALSE);
    CU_ASSERT_EQUAL(state, TURN_OFF);
}
void EDSL_UTC_000_007(){
    state = PWD_SET;
    CU_ASSERT_EQUAL(BacklightControl( NONE , FALSE , 4) , FALSE );
    CU_ASSERT_EQUAL(state, TURN_OFF);
}
```

002~007까지 PWD_SET 테스트.
IN은 Number의 입력유무 이며, True는 0~9 , False는 NONE(-1)이다.

Enable 은 True
Disable은 False

Test case identifier	Input specification	Output specification
EDLS_UTC_000_002	state == PWD_SET / tick == 0 / IN == TRUE / c == TRUE	state == TURN_ON
EDLS_UTC_000_003	state == PWD_SET / tick == 5 / IN == TRUE / c == TRUE	state == TURN_ON
EDLS_UTC_000_004	state == PWD_SET / tick == 9 / IN == TRUE / c == TRUE	state == TURN_ON
EDLS_UTC_000_005	state == PWD_SET / tick == 5 / IN == FALSE / c == TRUE	state == PWD_SET
EDLS_UTC_000_006	state == PWD_SET / tick == 10 / c == TRUE / in == TRUE	state == TURN_OFF/ Disable "backlight on"
EDLS_UTC_000_007	state == PWD_SET / C == FALSE / tick = true / in == FALSE	state == TURN_OFF/ Disable "backlight on"

Test Case Identification (3/7)

```
void EDSL.UTC_000_008(){ // tick0: 0~9
    state = TURN_OFF;
    CU_ASSERT_EQUAL(BacklightControl( 1 , TRUE , 5) , TRUE );
    CU_ASSERT_EQUAL(state, TURN_ON);
}

void EDSL.UTC_000_009(){
    state = TURN_OFF;
    CU_ASSERT_EQUAL(BacklightControl( 1 , TRUE , 5), TRUE);
    CU_ASSERT_EQUAL(state, TURN_ON);
}

void EDSL.UTC_000_010(){
    state = TURN_OFF;
    BacklightControl( NONE , FALSE , 5);
    CU_ASSERT_EQUAL(state, TURN_OFF);
}
```

008~010까지 TURN_OFF 테스트.

Test case identifier	Input specification	Output specification
EDLS.UTC_000_008	state == TURN_OFF / C == TRUE / Tick == TRUE / IN == TRUE	state == TURN_ON / Enable "backlight on"
EDLS.UTC_000_009	state == TURN_OFF / IN == TRUE / C == TRUE / Tick == TRUE	state == TURN_ON / Enable "backlight on"
EDLS.UTC_000_010	state == TURN_OFF / C == FALSE / IN == FALSE / Tick == TRUE	State == TURN_OFF

Test Case Identification (4/7)

```
void EDSL.UTC_000_011(){
    state = TURN_ON;
    CU_ASSERT_EQUAL(BacklightControl( NONE , FALSE , 5) , FALSE );
    CU_ASSERT_EQUAL(state, TURN_OFF);
}

void EDSL.UTC_000_012(){
    state = TURN_ON;
    TickCount = 10*TickPerSec;
    CU_ASSERT_EQUAL(BacklightControl( NONE , TRUE , 5) , FALSE );
    CU_ASSERT_EQUAL(state, TURN_OFF);
}

void EDSL.UTC_000_013(){
    state = TURN_ON;
    TickCount = 9*TickPerSec ;
    BacklightControl( NONE , TRUE , 0);
    CU_ASSERT_EQUAL(state, TURN_ON);
}
```

011~013까지 **TURN_ON** 테스트.

Test case identifier	Input specification	Output specification
EDLS.UTC_000_011	state == TURN_ON / C == false	state == TURN_OFF / Disable "backlight on"
EDLS.UTC_000_012	state == TURN_ON / 10tick[IN==false]	state == TURN_OFF / Disable "backlight on"
EDLS.UTC_000_013	state == TURN_ON / 9tick[IN==false]	state == TURN_ON

Test Case Identification (5/7)

000~013 WAIT_SET_PWD 테스트.

```
void EDLS_UTC_001_000(){
    state = WAIT_SET_PWD;
    InNumCount = 4;
    TickCount = 3*TickPerSec;
    CU_ASSERT_EQUAL(LockControl(NONE, NONE, NONE, TRUE, NONE)
        & LOCK_DOOR, LOCK_DOOR);
    CU_ASSERT_EQUAL(state, LOCKED);
}
```

```
void EDLS_UTC_001_001(){
    state = WAIT_SET_PWD;
    InNumCount = 3;
    TickCount = 3*TickPerSec;
    LockControl(TRUE, NONE, NONE, TRUE, NONE);
    CU_ASSERT_EQUAL(state, WAIT_SET_PWD);
}
```

```
void EDLS_UTC_001_002(){
    state = WAIT_SET_PWD;
    InNumCount = 4;
    TickCount = 2*TickPerSec;
    LockControl(NONE, NONE, NONE, TRUE, NONE);
    CU_ASSERT_EQUAL(state, WAIT_SET_PWD);
}
```

```
void EDLS_UTC_001_003(){
    state = WAIT_SET_PWD;
    InNumCount = 4;
    TickCount = 3*TickPerSec;
    LockControl(NONE, NONE, NONE, FALSE, NONE);
    CU_ASSERT_EQUAL(state, WAIT_SET_PWD);
}
```

Test case identifier	Input specification	Output specification
EDLS_UTC_001_000	state == WAIT_SET_PWD / Len(N) == 4 / tick == 3 / D == true	state == Locked / Trigger "Lock door"
EDLS_UTC_001_001	state == WAIT_SET_PWD / Len(N) == 3 / tick == 3 / D == true	state == WAIT_SET_PWD
EDLS_UTC_001_002	state == WAIT_SET_PWD / Len(N) == 4 / tick == 2 / D == true	state == WAIT_SET_PWD
EDLS_UTC_001_003	state == WAIT_SET_PWD / Len(N) == 4 / tick == 3 / D == false	state == WAIT_SET_PWD

Change Control for test

```
case LOCKED:
    if (tick){
        TickCount++;
    }
    if (0 <= Number && Number <= 9){
        TickCount = 0;
        if (InNumCount == 4){
            if (Correct){
                trigger |= RING_ALERT_1;
                trigger |= UNLOCK_DOOR;
                state = UNLOCKED;
            }
            else{
                trigger |= RING_ALERT_2;
            }
            for (i = 0; i < 4; i++){
                inputNum[i] = NONE;
            }
            InNumCount = 0;
        }
    }
    else if (KeyState || LockState){
        TickCount = 0;
        trigger |= RING_ALERT_1;
        trigger |= UNLOCK_DOOR;
        state = UNLOCKED;
    }
    else if (TickCount >= 10*TickPerSec && 0 < InNumCount && InNumCount < 4){
        TickCount = 0;
        InNumCount = 0;
        trigger |= RING_ALERT_3;
    }
    break;
```

inputNum[InNumCount++] = Number를 삭제

NumCmp(password, inputNum)를 변경

Test Case Identification (6/7)

```

void EDLS.UTC_001_009(){
    int result;
    state = LOCKED;
    result = LockControl( NONE, TRUE, NONE, NONE, NONE);
    CU_ASSERT_EQUAL(result & RING_ALERT_1, RING_ALERT_1);
}
void EDLS.UTC_001_010(){
    int result;
    state = LOCKED;
    result = LockControl( NONE, NONE, TRUE, NONE, NONE);
    CU_ASSERT_EQUAL(result & RING_ALERT_1, RING_ALERT_1);
    CU_ASSERT_EQUAL(result & UNLOCK_DOOR, UNLOCK_DOOR);
    CU_ASSERT_EQUAL(state, UNLOCKED);
}
void EDLS.UTC_001_011(){
    state = LOCKED;
    Correct = FALSE;
    InNumCount = 3;
    LockControl(TRUE, FALSE, FALSE, NONE, NONE);
    CU_ASSERT_EQUAL(state, LOCKED);
}
void EDLS.UTC_001_008(){
    int result;
    state = LOCKED;
    InNumCount = 4;
    Correct = TRUE;
    result = LockControl( TRUE, NONE, NONE, NONE, NONE);
    CU_ASSERT_EQUAL(result & RING_ALERT_1, RING_ALERT_1);
    CU_ASSERT_EQUAL(result & UNLOCK_DOOR, UNLOCK_DOOR);
    CU_ASSERT_EQUAL(state, UNLOCKED);
}
    
```

001~011까지 Locked 테스트.

Test case identifier	Input specification	Output specification
EDLS.UTC_001_004	state == Locked / Len(N) == 4 / N != P / IN == true	Trigger "Ring Alert2"
EDLS.UTC_001_005	state == Locked / tick == 10 / Len(N) == 1 / IN == none / K == false / O == false	Trigger "Ring Alert3"
EDLS.UTC_001_006	state == Locked / tick == 10 / Len(N) == 2 / IN == none / K == false / O == false	Trigger "Ring Alert3"
EDLS.UTC_001_007	state == Locked / tick == 10 / Len(N) == 4 / IN == none / K == false / O == false	state == Locked
EDLS.UTC_001_008	state == Locked / N == P / IN == true / Len(N) == 4	state == Unlocked / Trigger "Ring Alert1" / Trigger "Unlock door"
EDLS.UTC_001_009	state == Locked / K == true / IN == false	state == Unlocked / Trigger "Ring Alert1" / Trigger "Unlock door"
EDLS.UTC_001_010	state == Locked / O == true / IN == false	state == Unlocked / Trigger "Ring Alert1" / Trigger "Unlock door"
EDLS.UTC_001_011	state == Locked / N != P / K == false / O == false / Len(N) != 4 / IN == true	state == Locked

Test Case Identification (7/7)

```
void EDLS.UTC_001_012(){
    state = UNLOCKED;
    CU_ASSERT_EQUAL(LockControl(NONE, NONE, TRUE, TRUE, NONE)
        & LOCK_DOOR, LOCK_DOOR);
    CU_ASSERT_EQUAL(state, LOCKED);
}
```

```
void EDLS.UTC_001_013(){
    state = UNLOCKED;
    TickCount = 3*TickPerSec;
    CU_ASSERT_EQUAL(LockControl(NONE, NONE, TRUE, NONE, NONE)
        & LOCK_DOOR, LOCK_DOOR);
    CU_ASSERT_EQUAL(state, LOCKED);
}
```

```
void EDLS.UTC_001_014(){
    state = UNLOCKED;
    LockControl(NONE, NONE, TRUE, FALSE, NONE);
    CU_ASSERT_EQUAL(state, UNLOCKED);
}
```

```
void EDLS.UTC_001_015(){
    state = UNLOCKED;
    TickCount = 1*TickPerSec;
    LockControl(NONE, NONE, FALSE, TRUE, NONE);
    CU_ASSERT_EQUAL(state, UNLOCKED);
}
```

```
void EDLS.UTC_001_016(){
    state = UNLOCKED;
    TickCount = 2*TickPerSec;
    LockControl(NONE, NONE, FALSE, TRUE, NONE);
    CU_ASSERT_EQUAL(state, UNLOCKED);
}
```

```
void EDLS.UTC_001_017(){
    state = UNLOCKED;
    TickCount = 3*TickPerSec;
    LockControl(NONE, NONE, NONE, FALSE, NONE);
    CU_ASSERT_EQUAL(state, UNLOCKED);
}
```

012~017까지 **Unlocked** 테스트.

Test case identifier	Input specification	Output specification
EDLS.UTC_001_012	state == Unlocked / D == true / O == true	state == Locked / Trigger "Lock door"
EDLS.UTC_001_013	state == Unlocked / tick == 3 / D == true	state == Locked / Trigger "Lock door"
EDLS.UTC_001_014	state == Unlocked / D == false / O == true	State == Unlocked
EDLS.UTC_001_015	state == Unlocked / D == true / O == false / tick != 3	State == Unlocked
EDLS.UTC_001_016	state == Unlocked / tick == 2 / D == true / O == false	State == Unlocked
EDLS.UTC_001_017	state == Unlocked / tick == 3 / D == false	State == Unlocked

Test Result – Basic Type

```
Problems @ Javadoc Declaration Console
<terminated> T2_Unit.exe [C/C++ Application] C:\Users\Administrator#wor

CUnit - A unit testing framework for C - Version 2.1-2
http://cunit.sourceforge.net/

Suite: testing a suite
Test: EDSL.UTC_000_000 ...passed
Test: EDSL.UTC_000_001 ...passed
Test: EDSL.UTC_000_003 ...passed
Test: EDSL.UTC_000_004 ...passed
Test: EDSL.UTC_000_005 ...passed
Test: EDSL.UTC_000_006 ...passed
Test: EDSL.UTC_001_000 ...passed
Test: EDSL.UTC_001_001 ...passed
Test: EDSL.UTC_001_002 ...passed
Test: EDSL.UTC_001_003 ...passed
Test: EDSL.UTC_001_004 ...passed
Test: EDSL.UTC_001_005 ...passed
Test: EDSL.UTC_001_006 ...passed
Test: EDSL.UTC_001_007 ...passed
Test: EDSL.UTC_001_008 ...passed
Test: EDSL.UTC_001_009 ...passed
Test: EDSL.UTC_001_010 ...passed
Test: EDSL.UTC_001_011 ...passed
Test: EDSL.UTC_001_012 ...passed
Test: EDSL.UTC_001_013 ...passed
Test: EDSL.UTC_001_014 ...passed
Test: EDSL.UTC_001_015 ...passed
Test: EDSL.UTC_001_016 ...passed
Test: EDSL.UTC_001_017 ...passed

Run Summary:
  Type      Total      Ran      Passed      Failed      Inactive
  suites      1          1         n/a         0           0
  tests      24         24        24          0           0
  asserts    34         34        34          0           n/a

Elapsed time = 0.040 seconds
```

Thank You !