Object-Oriented Analysis and Design for Digital Watch System

Project Team
Team A2

Date
2020-06-08

Team Information
201410935 조현종
201511903 박재영
201612368 이지우
201814122 이예인

Activity 2055. Write Unit Test Code

1. Controller Test Code

```
@Before
public void setUp() throws Exception {
    this.controller = new Controller();
    Timer update = new Timer();
    update.scheduleAtFixedRate(this.controller, 0, 10);
}



@Test
public void getTimeKeepingTime() throws InterruptedException {
    Thread.sleep(10);
    assertThat(controller.getSegment1(),
            is(LocalDateTime.now().format(DateTimeFormatter.ofPattern("HHmmss"))));
}
```

```
@Test
public void setTimeKeepingTime() throws InterruptedException {
    Thread.sleep( millis: 100);
    controller.reqSetting();
    int seg;
    //시
    Thread.sleep( millis: 1000);
    seg= Integer.parseInt(controller.getSegment1());
    controller.changeUnitValue( changeValue: +1);

    Thread.sleep( millis: 500);
    seg = Integer.parseInt(controller.getSegment1()) - seg;
    assertThat(seg, either(is( value: 10000)).or(is( value: -230000)));
    controller.changeUnitValue( changeValue: -1);

    //분
    controller.nextUnit();
    Thread.sleep( millis: 500);
     seg = Integer.parseInt(controller.getSegment1());
    controller.changeUnitValue( changeValue: +1);

    Thread.sleep( millis: 500);
    seg = Integer.parseInt(controller.getSegment1()) - seg;
    assertThat(seg, either(is( value: 100)).or(is( value: -5900)));
    controller.changeUnitValue( changeValue: -1);

    //초
    controller.nextUnit();
    Thread.sleep( millis: 500);
    seg = Integer.parseInt(controller.getSegment1());
    controller.changeUnitValue( changeValue: +1);

    Thread.sleep( millis: 500);
    seg = Integer.parseInt(controller.getSegment1()) - seg;
    assertThat(seg, either(is( value: 1)).or(is( value: -59)));
    controller.changeUnitValue( changeValue: -1);
```

```
    //일
    controller.nextUnit();
    Thread.sleep( millis: 500);
    seg = Integer.parseInt(controller.getSegment2().substring(7,9));
    controller.changeUnitValue( changeValue: +1);

    Thread.sleep( millis: 1000);
    seg = Integer.parseInt(controller.getSegment2().substring(7,9)) - seg;
    assertThat(seg, either(is( value: 1)).or(is( value: -27)).or(is( value: -28)).or(is( value: -29)).or(is( value: -30)));
    controller.changeUnitValue( changeValue: -1);

    //월
    controller.nextUnit();
    Thread.sleep( millis: 500);
    seg = Integer.parseInt(controller.getSegment2().substring(5,7));
    controller.changeUnitValue( changeValue: +1);

    Thread.sleep( millis: 1000);
    seg = Integer.parseInt(controller.getSegment2().substring(5,7)) - seg;
    assertThat(seg, either(is( value: 1)).or(is( value: -11)));
    controller.changeUnitValue( changeValue: -1);

    //년
    controller.nextUnit();
    Thread.sleep( millis: 500);
    seg = Integer.parseInt(controller.getSegment2().substring(3,5));
    controller.changeUnitValue( changeValue: +1);

    Thread.sleep( millis: 1000);
    seg = Integer.parseInt(controller.getSegment2().substring(3,5)) - seg;
    assertThat(seg, is( value: 1));
    controller.changeUnitValue( changeValue: -1);
}
```

```
@Test
public void setAlarmPage() throws InterruptedException {
    this.controller.reqModeSwitch();
    Thread.sleep( millis: 10);
    controller.reqSetting();
    String seg;
    //시
    Thread.sleep( millis: 1000);
    controller.changeUnitValue( changeValue: +1);
    controller.reqCompleteSetting();
    Thread.sleep( millis: 1000);
    seg = controller.getSegment1();
    for (int i = 0; i < 4; i++)
        controller.reqChangeIndicatedAlarm();

    Thread.sleep( millis: 1000);

    assertThat(seg,is(controller.getSegment1()));
}
```

```java
@Test
public void setAlarmTime() throws InterruptedException {
    Thread.sleep( millis: 100);
    this.controller.reqModeSwitch();

    int seg;
    Thread.sleep( millis: 100);
    controller.reqSetting();
    //시
    Thread.sleep( millis: 1000);
    seg= Integer.parseInt(controller.getSegment1());
    controller.changeUnitValue( changeValue: +1);

    Thread.sleep( millis: 500);
    seg = Integer.parseInt(controller.getSegment1()) - seg;
    assertThat(seg, either(is( value: 10000)).or(is( value: -230000)));
    controller.changeUnitValue( changeValue: -1);

    //분
    controller.nextUnit();
    Thread.sleep( millis: 500);
    seg = Integer.parseInt(controller.getSegment1());
    controller.changeUnitValue( changeValue: +1);

    Thread.sleep( millis: 500);
    seg = Integer.parseInt(controller.getSegment1()) - seg;
    assertThat(seg, either(is( value: 100)).or(is( value: -5900)));
    controller.changeUnitValue( changeValue: -1);

}
```

```java
@Test
public void getStopwatchTime() throws InterruptedException {
    for(int i = 0; i<2; i++)
        this.controller.reqModeSwitch();

    assertThat(controller.getSegment1(),
            is(LocalTime.of( hour: 0, minute: 0, second: 0).format(DateTimeFormatter.ofPattern("HHmmss"))));
}
@Test
public void startStopwatch() throws InterruptedException {
    for(int i = 0; i<2; i++)
        this.controller.reqModeSwitch();
    controller.reqStartStopWatch();
    Thread.sleep( millis: 2000);
    assertThat(controller.getSegment1(),
            is(not(LocalTime.of( hour: 0, minute: 0, second: 0).format(DateTimeFormatter.ofPattern("HHmmss")))));
}


@Test
public void getTimerTime() throws InterruptedException {
    for(int i = 0; i<3; i++)
        this.controller.reqModeSwitch();

    assertThat(controller.getSegment1(),
            is(LocalTime.of( hour: 0, minute: 0, second: 0).format(DateTimeFormatter.ofPattern("HHmmss"))));
}
```

```java
@Test
public void setTimerTime() throws InterruptedException {
    for(int i = 0; i<3; i++)
        this.controller.reqModeSwitch();

    Thread.sleep( millis: 100);
    controller.reqSetting();

    int seg;

    //시
    Thread.sleep( millis: 1000);
    seg= Integer.parseInt(controller.getSegment1());
    controller.changeUnitValue( changeValue: +1);

    Thread.sleep( millis: 500);
    seg = Integer.parseInt(controller.getSegment1()) - seg;
    assertThat(seg, either(is( value: 10000)).or(is( value: -230000)));

    //분
    controller.nextUnit();
    Thread.sleep( millis: 500);
    seg = Integer.parseInt(controller.getSegment1());
    controller.changeUnitValue( changeValue: +1);

    Thread.sleep( millis: 500);
    seg = Integer.parseInt(controller.getSegment1()) - seg;
    assertThat(seg, either(is( value: 100)).or(is( value: -5900)));

    //초
    controller.nextUnit();
    Thread.sleep( millis: 500);
    seg = Integer.parseInt(controller.getSegment1());
    controller.changeUnitValue( changeValue: +1);

    Thread.sleep( millis: 500);
    seg = Integer.parseInt(controller.getSegment1()) - seg;
    assertThat(seg, either(is( value: 1)).or(is( value: -59)));
```

```java
    controller.reqCompleteSetting();

    assertThat( actual: 10101, is(Integer.parseInt(controller.getSegment1())));

    controller.reqStartTimer();
    Thread.sleep( millis: 2000);
    assertThat( actual: 10101, is(not(Integer.parseInt(controller.getSegment1()))));

}


public void modeindi(){
    this.controller.reqSetIndicateMode();
    this.controller.reqNextIndicator();
    this.controller.reqNextIndicator();
    this.controller.reqSelectMode();
    this.controller.reqNextIndicator();
    this.controller.reqSelectMode();
    this.controller.reqNextIndicator();
    this.controller.reqSelectMode();
}
@Test
public void modeIndicateChange() throws InterruptedException {
    this.controller.reqSetIndicateMode();
    this.controller.reqNextIndicator();
    this.controller.reqNextIndicator();
    this.controller.reqSelectMode();
    this.controller.reqNextIndicator();
    this.controller.reqSelectMode();
    this.controller.reqNextIndicator();
    this.controller.reqSelectMode();
```

```
    for(int i = 0; i<3; i++)
        this.controller.reqModeSwitch();
    Thread.sleep( millis: 1000);
    assertThat(controller.getSegment1(),
            is( value: "-000--"));
}

@Test
public void getWorldTime() throws InterruptedException {
    modeindi();
    for(int i = 0; i<2; i++)
        this.controller.reqModeSwitch();

    Thread.sleep( millis: 1000);
    assertThat(controller.getSegment1(),
            is(LocalDateTime.now().format(DateTimeFormatter.ofPattern("HHmmss"))));

}
@Test
public void changeWorldTime() throws InterruptedException {
    modeindi();
    for(int i = 0; i<2; i++)
        this.controller.reqModeSwitch();

    controller.reqChangeWorldTime();
    Thread.sleep( millis: 1000);
    assertThat(controller.getSegment1(),
            is(ZonedDateTime.now(ZoneId.of("+10")).format(DateTimeFormatter.ofPattern("HHmmss"))));

    for(int i = 0; i<24; i++)
        controller.reqChangeWorldTime();
    Thread.sleep( millis: 1000);
    assertThat(controller.getSegment1(),
            is(LocalDateTime.now().format(DateTimeFormatter.ofPattern("HHmmss"))));
}
```

```
@Test
public void setWorldTimeZone() throws InterruptedException {
    modeindi();
    for(int i = 0; i<2; i++)
        this.controller.reqModeSwitch();
    controller.reqChangeWorldTime();
    controller.reqChangeTimeZone();

    for(int i = 0; i<2; i++)
        this.controller.reqModeSwitch();
    Thread.sleep( millis: 1000);
    assertThat(controller.getSegment1(),
            is(ZonedDateTime.now(ZoneId.of("+10")).format(DateTimeFormatter.ofPattern("HHmmss"))));

}

@Test
public void setTurnip() throws InterruptedException {
    modeindi();
    for(int i = 0; i<3; i++)
        this.controller.reqModeSwitch();

    controller.reqSetting();
    controller.reqChangePriceValue( changeValue: +1);

    Thread.sleep( millis: 1000);
    assertThat(controller.getSegment1(),
            is( value: "-091--"));
}
```

```java
@Test
public void getTurnipCalc() throws InterruptedException {
    modeindi();
    for(int i = 0; i<3; i++)
        this.controller.reqModeSwitch();

    controller.reqSetting();
    controller.reqChangePriceValue( changeValue: +10);
    controller.reqCompleteSetting();
    controller.reqChangeDate();


    controller.reqSetting();
    controller.reqCompleteSetting();
    controller.reqChangeDate();


    controller.reqSetting();
    controller.reqChangePriceValue( changeValue: -10);
    controller.reqCompleteSetting();
    controller.reqChangeDate();


    controller.reqSetting();
    controller.reqChangePriceValue( changeValue: -20);
    controller.reqCompleteSetting();
    controller.reqChangeDate();


    controller.reqSetting();
    controller.reqChangePriceValue( changeValue: -30);
    controller.reqCompleteSetting();
    controller.reqChangeDate();


    controller.reqSetting();
    controller.reqChangePriceValue( changeValue: -20);
    controller.reqCompleteSetting();
    controller.reqChangeDate();
```

```java
    Thread.sleep( millis: 1000);
    assertThat(controller.getSegment1(),
            is(not( value: "-000--")));
}
```

2. TimeKeeping Test Code

```
@Before
public void setUp(){
    this.timeKeeping = TimeKeeping.getInstance();
}

@Test
public void getCurrentTime() {
    // given
    //when
    this.timeKeeping.setTimeZone(ZoneId.of("+9"));
    //then
    assertThat(ZonedDateTime.now().toLocalDateTime(), is(this.timeKeeping.getCurrentTime().toLocalDateTime()));

}
@Test
public void setTimeZone() {
    // given

    for(int i = -12; i <= 12; i++) {
        // when
        ZoneId id = ZoneId.of(i<0?Integer.toString(i):"+"+i);
        this.timeKeeping.setTimeZone(id);
        ZoneId result = this.timeKeeping.getCurrentTime().getZone();

        //then

        assertThat(result, is(id));
    }
}
```

3. Alarm Test Code

```
@Before
public void setUp(){
    this.alarm = new Alarm();
}

//test time set and get
@Test
public void saveGetAlarmTime() {
    //given
    LocalTime time = LocalTime.now();

    //when
    alarm.setAlarmTime(time);

    //then
    assertThat(time, is(alarm.getAlarmTime()));
    assertThat( actual: true, is(alarm.getActivated()));
}

@Test
public void activateAlarm() {
    //when
    this.alarm.activateAlarm();
    //then
    assertThat( actual: true, is(alarm.getActivated()));
}

@Test
public void deactivateAlarm() {

    this.alarm.activateAlarm();
    this.alarm.deactivateAlarm();
    assertThat( actual: false, is(alarm.getActivated()));
}
```

4. Stopwatch Test Code

```
@Before
public void setUp() { this.stopwatch = new Stopwatch(); }

@Test
public void getStopwatchTime() {
    time=LocalTime.of( hour: 0, minute: 0, second: 0);
    assertThat(time, is(stopwatch.getStopwatchTime()));
}

@Test
public void getLapTime() throws InterruptedException {
    time=LocalTime.of( hour: 0, minute: 0, second: 1);
    stopwatch.startStopwatch();

    Thread.sleep( millis: 1000);
    stopwatch.lapTime();
    assertThat(time.getSecond(), is(stopwatch.getLapTime().getSecond()));
}

@Test
public void getIsStartedStopwatch() {
    stopwatch.startStopwatch();

    assertThat( actual: true, is(stopwatch.getIsStartedStopwatch()));
}
```

```
@Test
public void startStopwatch() throws InterruptedException {
    time=LocalTime.of( hour: 0, minute: 0, second: 0);
    stopwatch.startStopwatch();
    Thread.sleep( millis: 1000);
    assertThat( actual: true, is(time.isBefore(stopwatch.getStopwatchTime())));

}

@Test
public void pauseStopwatch() throws InterruptedException {
    time=LocalTime.of( hour: 0, minute: 0, second: 0);
    stopwatch.startStopwatch();

    Thread.sleep( millis: 1000);
    time=stopwatch.getStopwatchTime();

    stopwatch.pauseStopwatch();

    assertThat(time.getSecond(), is(stopwatch.getStopwatchTime().getSecond()));

}

@Test
public void resetStopwatch() throws InterruptedException {
    time=LocalTime.of( hour: 0, minute: 0, second: 0);
    stopwatch.startStopwatch();

    Thread.sleep( millis: 1000);

    assertThat(time, is(not(stopwatch.getStopwatchTime())));
    stopwatch.resetStopwatch();
    assertThat(time, is(stopwatch.getStopwatchTime()));
}
```

5. Timer Test Code

```
@Before
public void setUp() { this.timer = new Timer(); }

@Test
public void getTimerTime() {
    time=LocalTime.of( hour: 0, minute: 0, second: 0);
    assertThat(time, is(timer.getTimerTime()));
}

@Test
public void setTimerTime() {
    //given
    time=LocalTime.of( hour: 0, minute: 10, second: 0);
    //when
    timer.setTimerTime(time);
    //then
    assertThat(time, is(timer.getTimerTime()));
}

@Test
public void getRunTime() {
    //given
    time=LocalTime.of( hour: 0, minute: 10, second: 0);
    //when
    timer.setTimerTime(time);
    //then
    assertThat(time, is(timer.getRunTime()));
}
```

```
@Test
public void getIsStartedTimer() {
    //given
    time=LocalTime.of( hour: 0, minute: 10, second: 0);
    //when
    timer.setTimerTime(time);
    timer.startTimer(time);
    //then
    assertThat( actual: true, is(timer.getIsStartedTimer()));
}

@Test
public void startTimer() throws InterruptedException {
    //given
    time=LocalTime.of( hour: 0, minute: 10, second: 0);
    //when
    timer.setTimerTime(time);
    timer.startTimer(timer.getTimerTime());
    Thread.sleep( millis: 1000);
    //then
    assertThat(time, is(not(timer.getRunTime())));
}
```

```java
@Test
public void pauseTimer() throws InterruptedException {
    //given
    time=LocalTime.of( hour: 0, minute: 10, second: 0);
    //when
    timer.setTimerTime(time);
    timer.startTimer(time);
    Thread.sleep( millis: 2000);

    timer.pauseTimer();
    time=timer.getRunTime();
    Thread.sleep( millis: 2000);
    //then
    assertThat(time, is(timer.getRunTime()));
}

@Test
public void resetTimer() throws InterruptedException {
    //given
    time=LocalTime.of( hour: 0, minute: 10, second: 0);
    //when
    timer.setTimerTime(time);
    timer.startTimer(time);
    Thread.sleep( millis: 2000);
    timer.pauseTimer();
    timer.resetTimer();

    Thread.sleep( millis: 500);
    //then
    assertThat(time, is(timer.getRunTime()));
}
```

6. TurnipPrice Test Code

```java
@Before
public void setUp() { this.turnipPrice = new TurnipPrice(); }

@Test
public void getTurnipPrice() {
    assertThat(this.turnipPrice.getTurnipPrice(),is( value: 0));

}

@Test
public void getTurnipDay() {

    assertThat(this.turnipPrice.getTurnipDay(),isA(String.class));

}

@Test
public void savePrice() {

    this.turnipPrice.setTurnipPrice(100);

    assertThat(this.turnipPrice.getTurnipPrice(),is( value: 100));
}

@Test
public void resetPrice() {
    //given
    this.turnipPrice.setTurnipPrice(100);
    //when
    this.turnipPrice.resetPrice();
    //then
    assertThat(this.turnipPrice.getTurnipPrice(),is( value: 0));
}
```

```java
@Test
public void nextPrice() {
    //given
    this.turnipPrice.setTurnipPrice(100);
    //when
    for(int i = 13;i<13;i++){
        this.turnipPrice.nextPrice();
    }
    //then
    assertThat(this.turnipPrice.getTurnipPrice(),is( value: 100));
}
```

7. WorldTime Test Code

```
@Before
public void setUp(){
    this.worldTime = new WorldTime();
    this.timeKeeping = TimeKeeping.getInstance();
}

@Test
public void getWorldTime() {
    assertThat(ZonedDateTime.now().toLocalDateTime().withNano(0),
            is(this.worldTime.getWorldTime().toLocalDateTime().withNano(0)));
}

@Test
public void getUTCString() { assertThat(worldTime.getUTCString(),containsString( substring: "UTC")  ); }

@Test
public void nextWorldTime() {
    ZonedDateTime time;
    time = worldTime.getWorldTime();

    worldTime.nextWorldTime();
    assertThat(time,is(not(worldTime.getWorldTime())));
}
```

```
@Test
public void changeTimeZone() {
    ZonedDateTime time;
    worldTime.nextWorldTime();
    worldTime.nextWorldTime();

    time = worldTime.getWorldTime();
    worldTime.changeTimeZone();

    assertThat(time.getZone(),is(timeKeeping.getCurrentTime().getZone()));

}
```

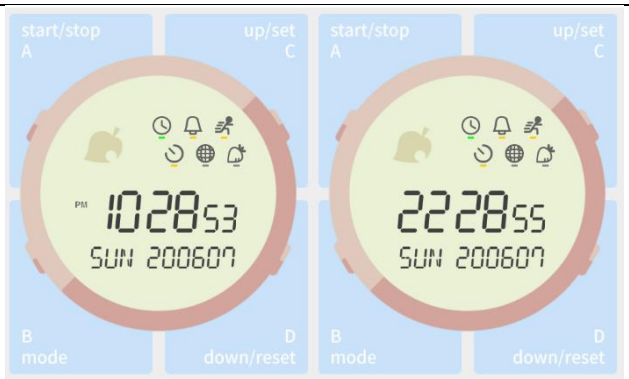
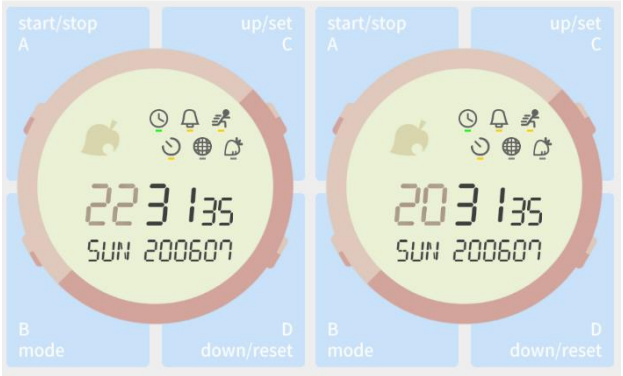
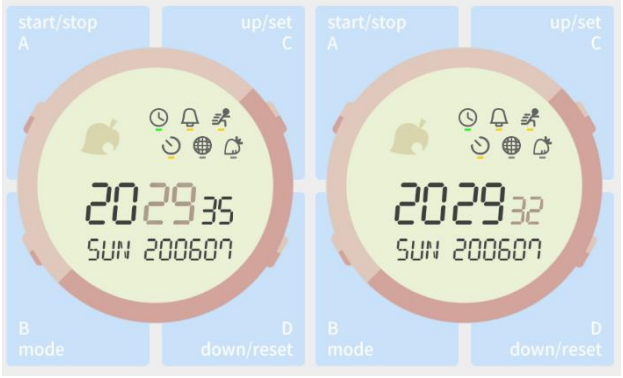
8. ModeSwitch Test Code

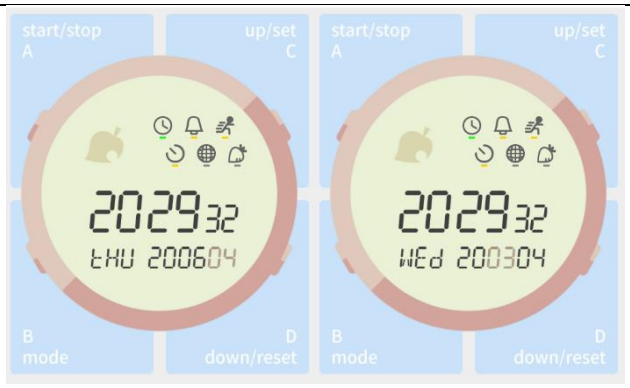
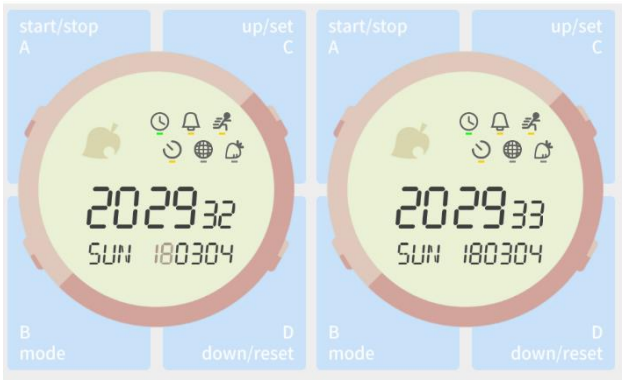
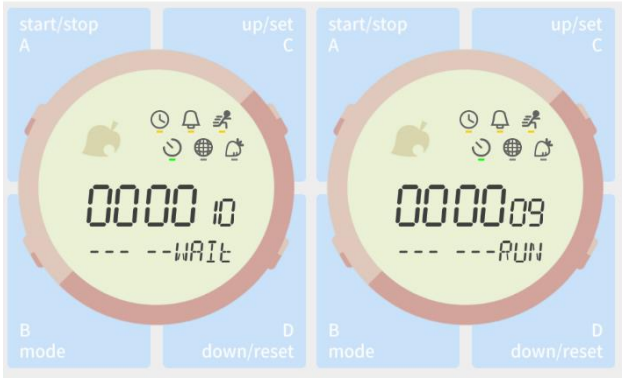
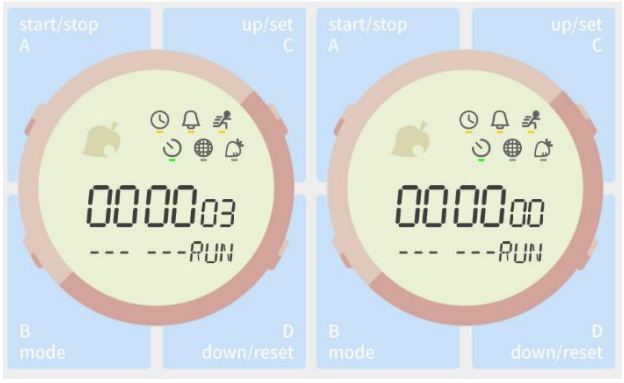
```
@Before
public void setUp() {
    this.modeSwitch = new  ModeSwitch();
}
@Test
public void getMode() {

    assertThat(modeSwitch.getMode(),is( value: 0));
    modeSwitch.nextMode();
    assertThat(modeSwitch.getMode(),is( value: 1));

}

@Test
public void getEnabledMode() {

    int enMod=0;
    int[] en = modeSwitch.getEnabledMode();
    for(int i = 0; i<6; i++){
        if (en[i]==1) enMod += 1;
    }
    assertThat(enMod, is( value: 4));
}
```

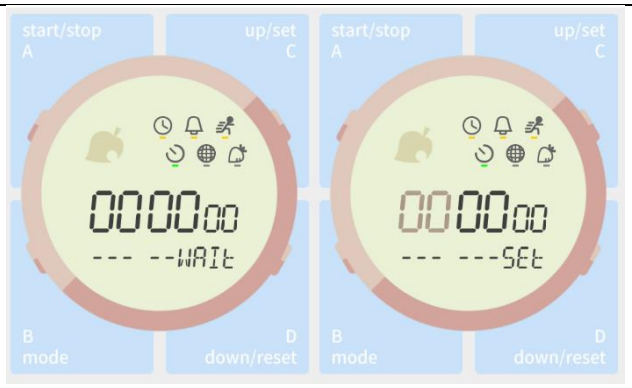
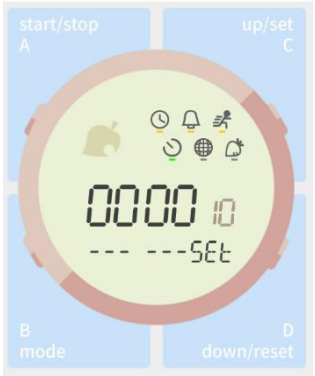
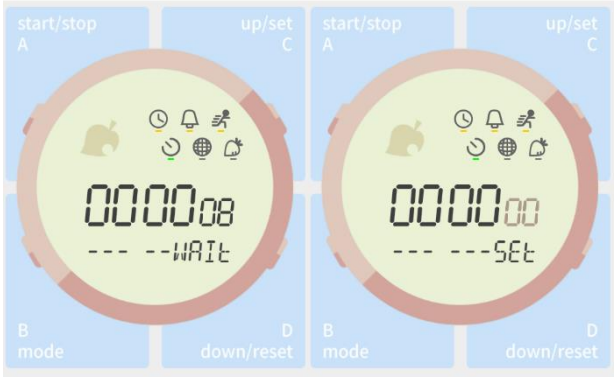
```java
@Test
public void initialize() {
    modeSwitch.nextMode();

    modeSwitch.nextMode();
    //when
    modeSwitch.initialize();
    assertThat(modeSwitch.getMode(),is( value: 0));
}

@Test
public void nextMode() {
    modeSwitch.initialize();

    for (int i = 0; i<10;i++){
        modeSwitch.nextMode();
    }
    assertThat( actual: true, is( value: this.modeSwitch.getMode()<6));
}

@Test
public void setMode() {
    int[] enabledMode= new int[]{0, 0, 1, 1, 1, 1};
    modeSwitch.setMode(enabledMode);
    assertThat(enabledMode,is(modeSwitch.getEnabledMode()));
}
```

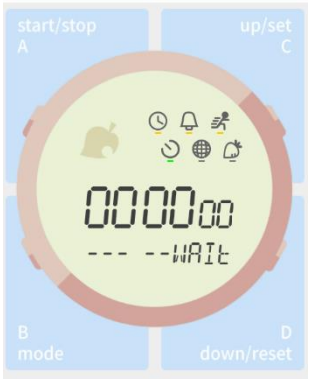
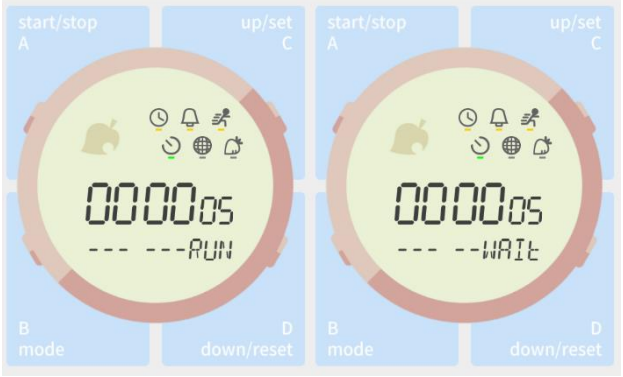
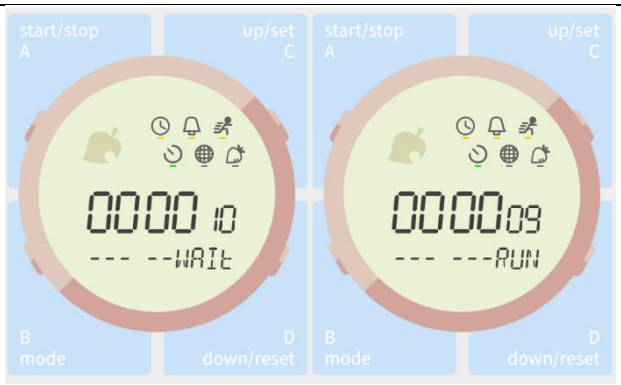
Activity 2061. Unit Testing

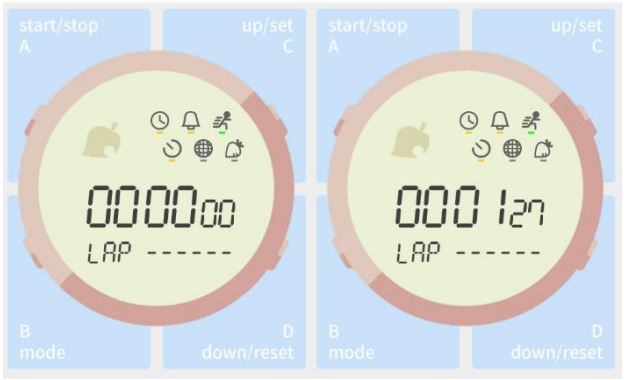
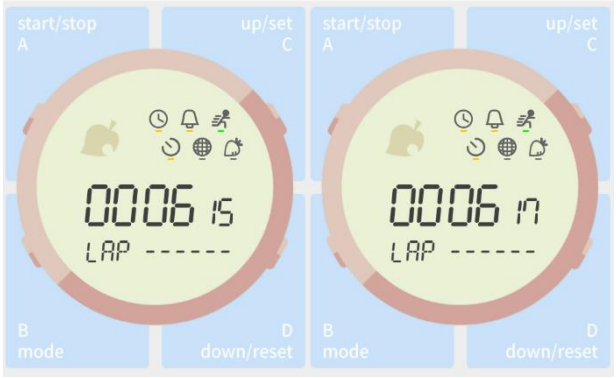
Activity 2063. System Testing

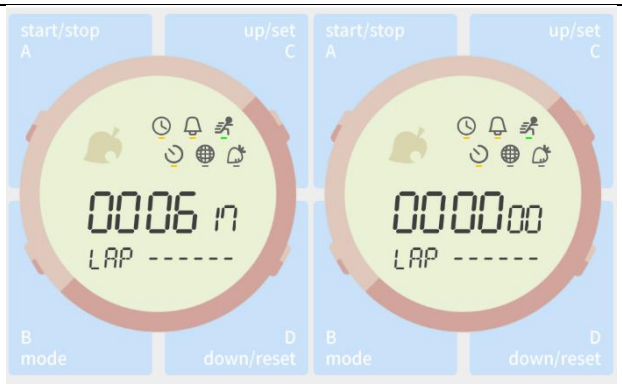
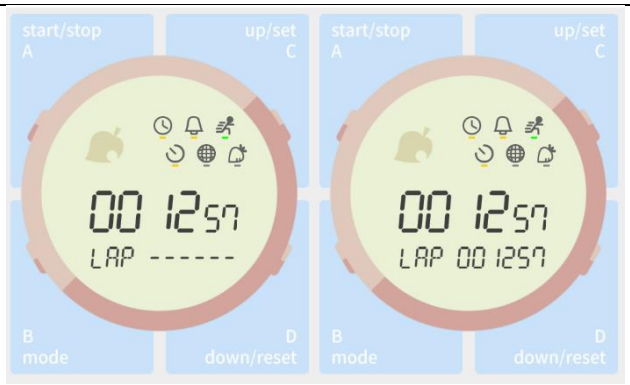
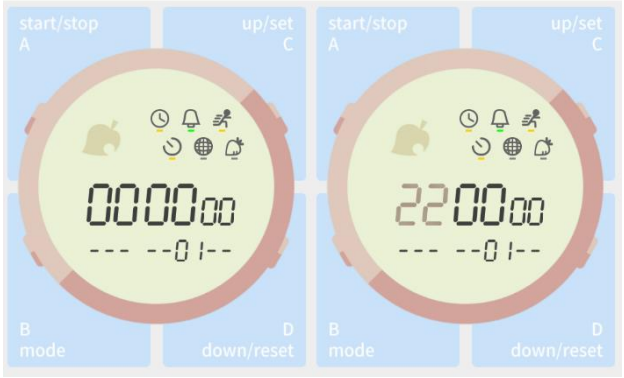
#	System Test Case	Description	Result
1	Set Hour Format	사용자가 시간 표기법을 12H(24H)에서 24H(12H)으로 변경	 A버튼 눌러 시간 표기법 변경 A버튼 누를 때마다 PM10시 28분, 22시 28분으로 변경되는 시간 표기 확인
2	Set Time	사용자가 시간 값 변경	 C버튼으로 설정 진입 C,D버튼으로 값 변경  A버튼으로 유닛 이동, C,D버튼으로 값 변경

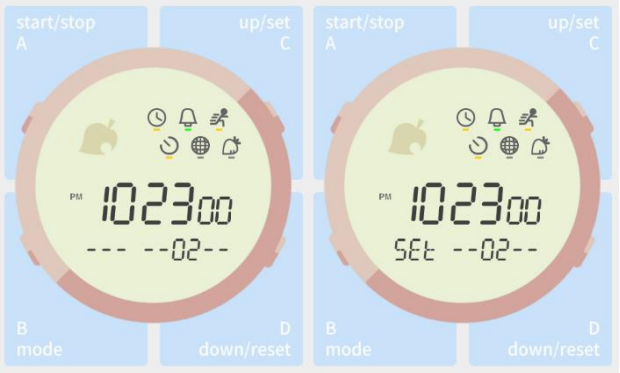
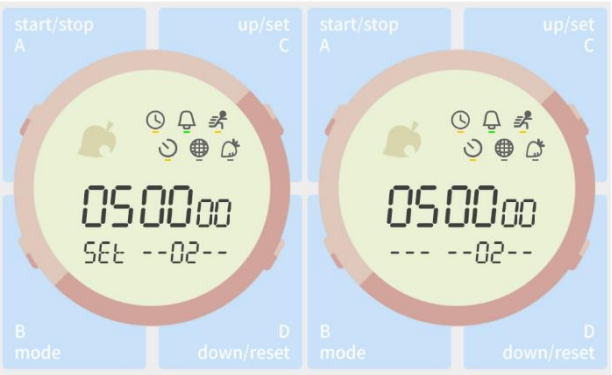
			 A버튼으로 유닛 이동, C,D버튼으로 값 변경  B버튼으로 설정 완료 2018년 3월 4일 20시 29분으로 변경된 시간 확인
3	Start Timer	사용자가 Timer 시작	 B버튼으로 Timer진입 B버튼으로 타이머 시작  타이머 시간 확인

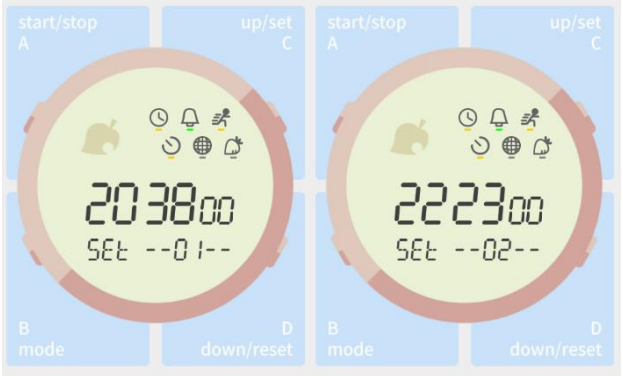
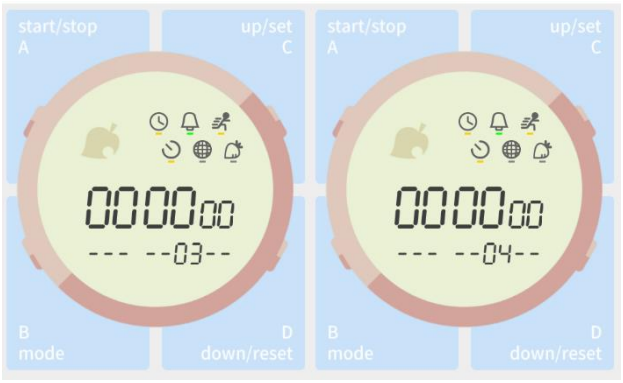
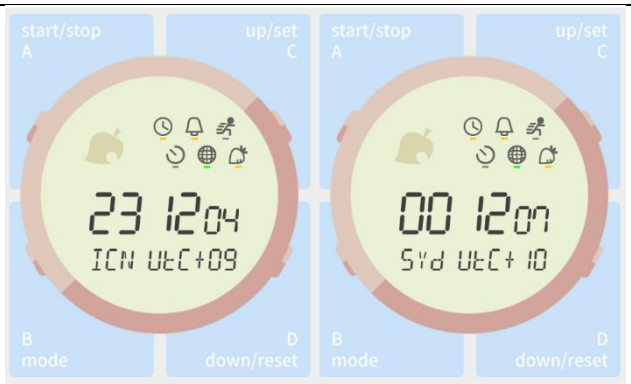
4	Set Timer	사용자가 Timer 시간 설정	 C버튼으로 타이머 설정 모드 진입 C,D버튼으로 시간 값 변경  A버튼으로 시간 요소 선택 C,D버튼으로 시간 값 변경  B버튼으로 설정 완료 설정된 타이머 시간 확인
5	Set Timer	사용자가 Timer 시간을 0초로 설정	

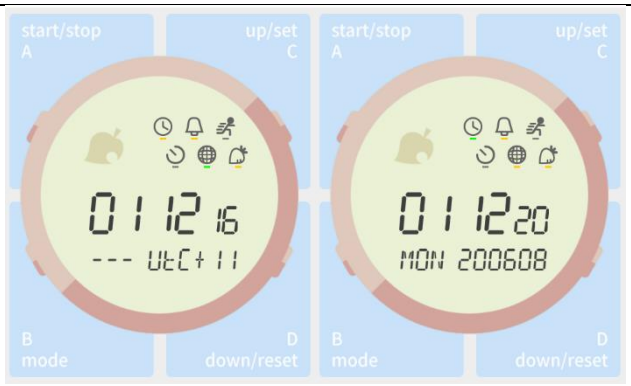
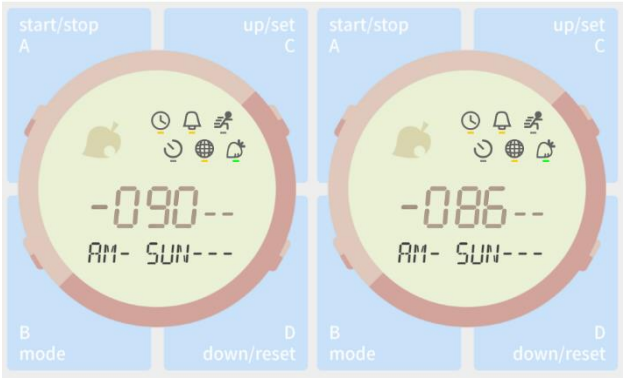
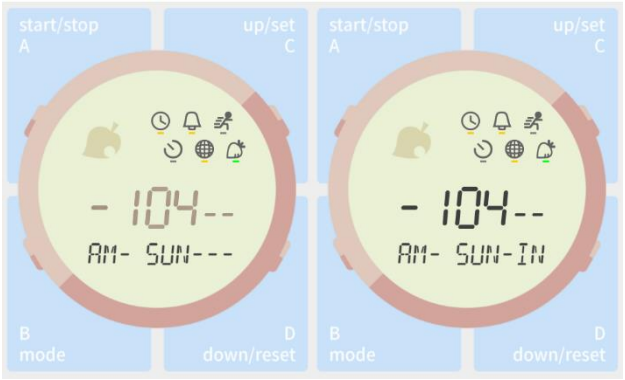
			C버튼으로 타이머 설정 모드 진입 C,D버튼으로 시간 값 0으로 변경  B버튼으로 설정 완료
6	Pause Timer	사용자가 Timer 일시정지	 A버튼으로 Timer 일시정지 감소하지 않는 타이머 시간 확인
7	Reset Timer	사용자가 Timer 초기화	 A버튼으로 타이머 시작 

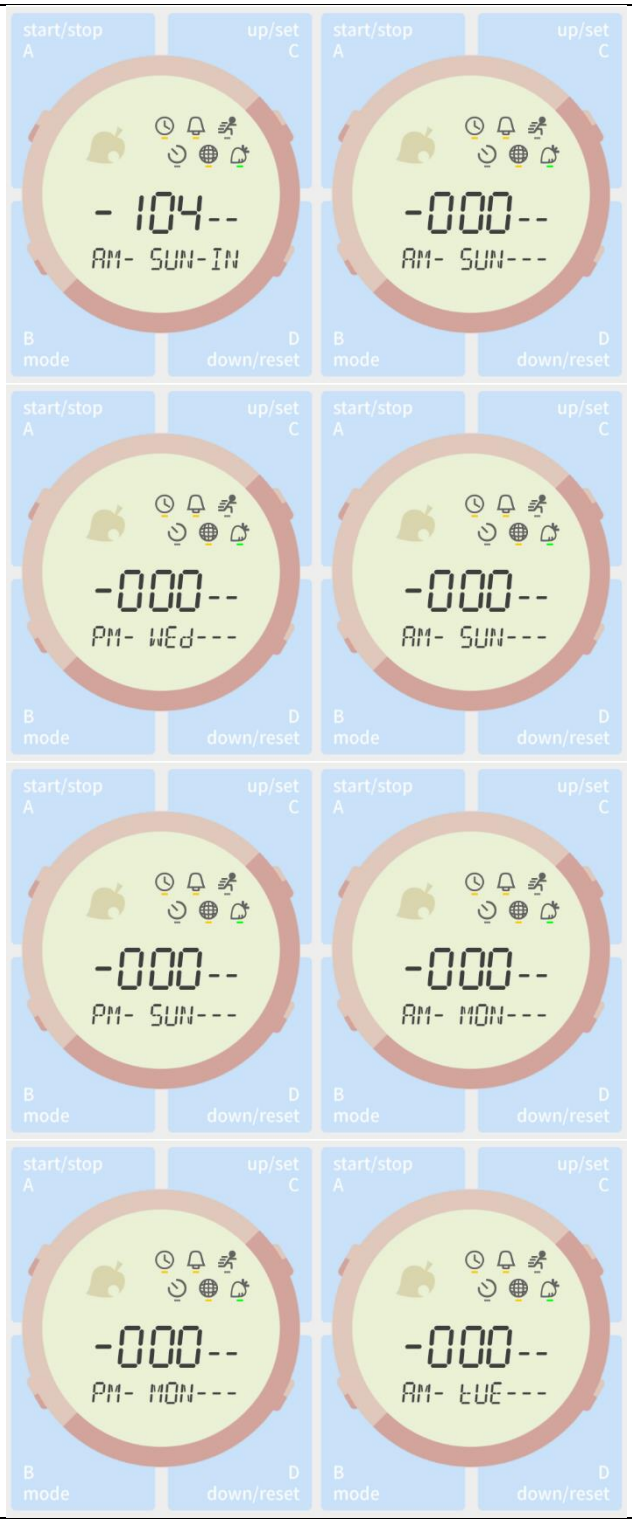
			A버튼으로 타이머 일시정지  D버튼으로 타이머 초기화. 초기화된 타이머 값 확인
8	Start Stopwatch	사용자가 Stopwatch 시작	 A버튼으로 Stopwatch 시작 증가하기 시작한 Stopwatch 시간 확인
9	Pause Stopwatch	사용자가 Stopwatch 일시정지	 A버튼으로 Stopwatch 일시정지 증가하지 않는 Stopwatch 시간 확인

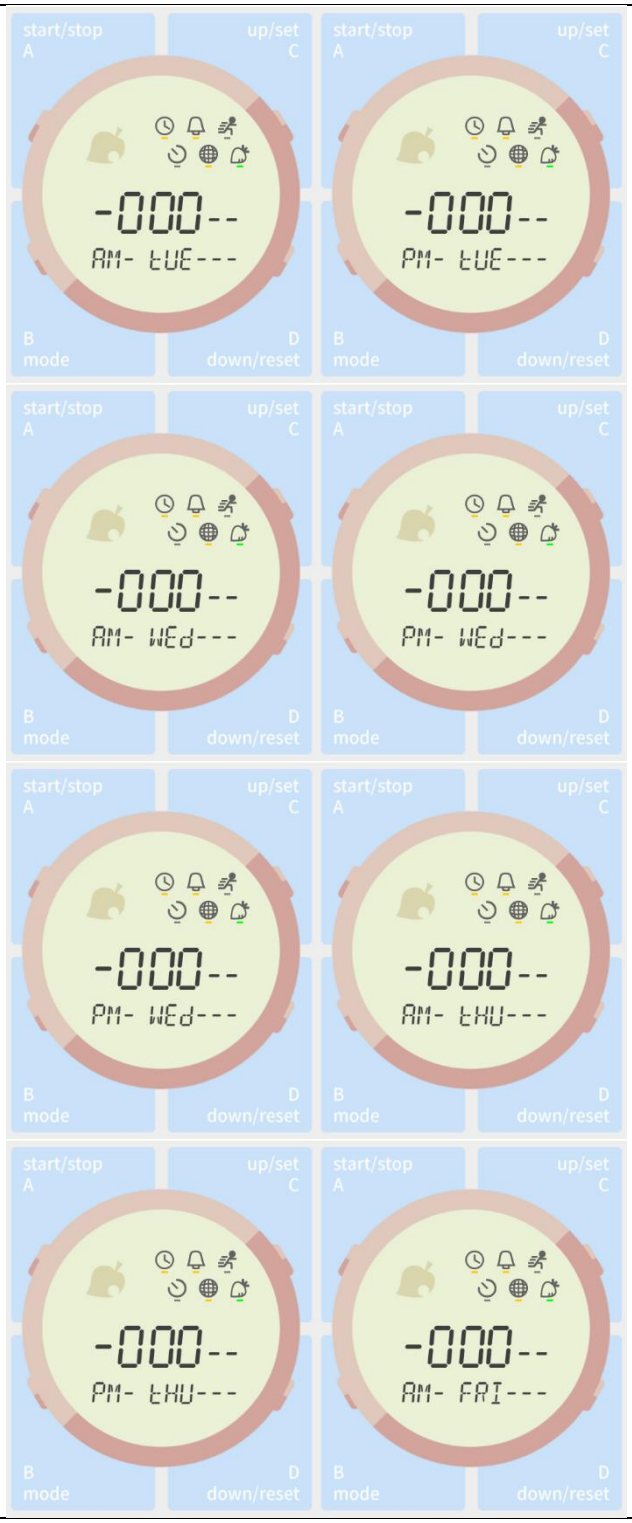
10	Reset Stopwatch	사용자가 Stopwatch 초기화	 D버튼으로 Stopwatch 초기화 0으로 설정된 Stopwatch 확인
11	Record Lap Time	사용자가 Lap Time을 기록	 실행중인 Stopwatch에서 C버튼으로 Lap Time 기록 subsegment에 표시된 Lap Time 확인
12	Set Alarm	사용자가 알람을 설정	 C버튼으로 시간 설정 모드 진입 C,D버튼으로 시간 요소 값 설정

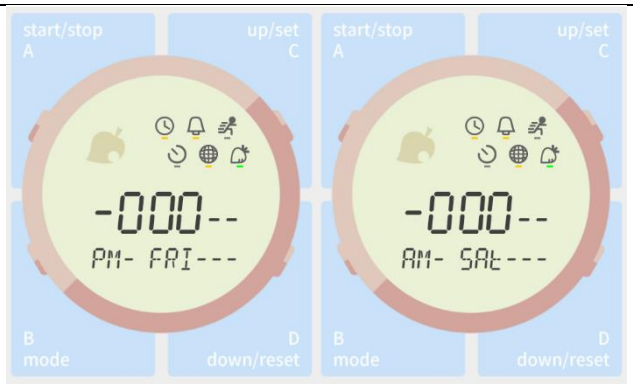
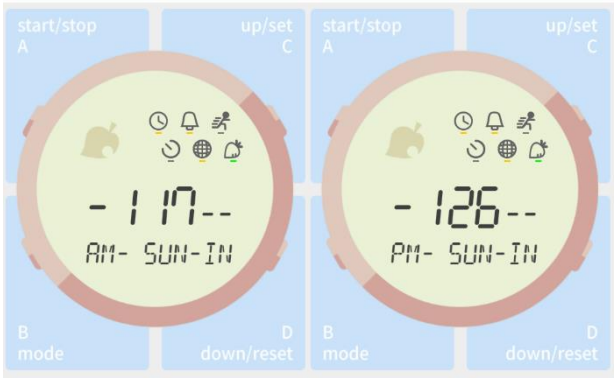
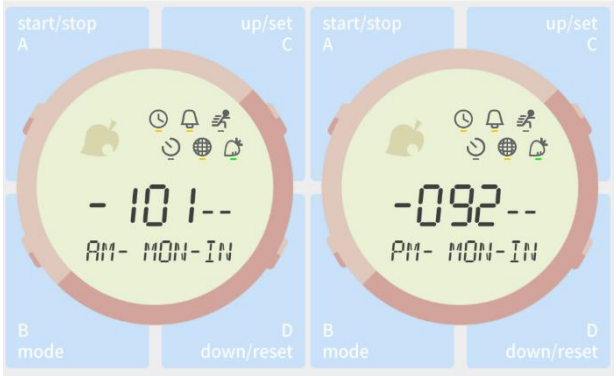
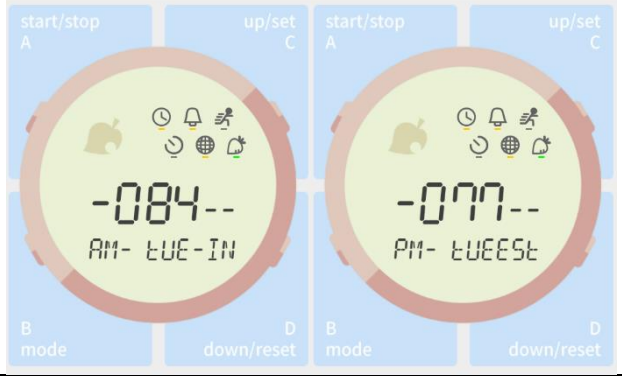
			 A버튼으로 시간 요소 선택 C,D버튼으로 시간 요소 값 설정  B버튼으로 설정 완료 설정된 알람 시간 확인
13	Activate Alarm	사용자가 비활성화 되어있는 알람을 활성화	 D버튼으로 알람 활성화 SET이 추가된 subsegment 확인
14	Deactivate Alarm	사용자가 활성화 되어있는 알람을 비활성화	

			D버튼으로 알람 비활성화 SET이 사라진 subsegment 확인
15	Indicate Another Alarm	사용자가 알람 4개중 표시되지 않은 알람을 확인	 A버튼으로 표시할 알람 변경 변경된 알람 시간과 subsegment 확인  A버튼으로 표시할 알람 변경 변경된 알람 시간과 subsegment 확인
16	Change World Time	사용자가 다른 나라의 시간으로 변경	 B버튼으로 세계시간 모드 진입 A버튼으로 시간 표시할 나라 시간 선택 변경된 시간과 subsegment 확인

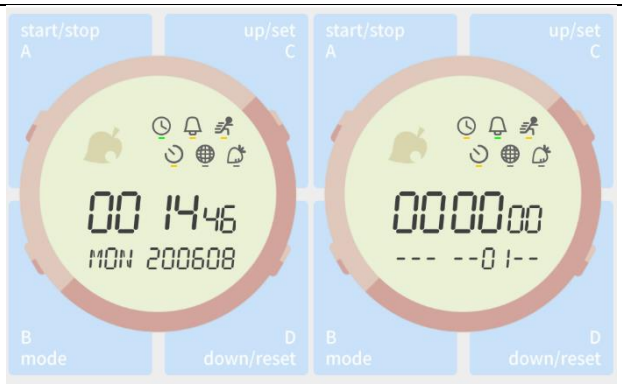
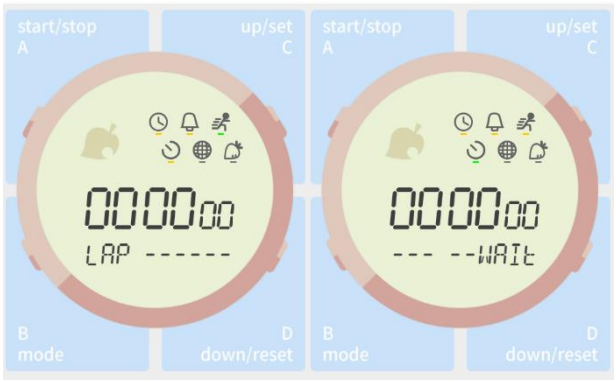
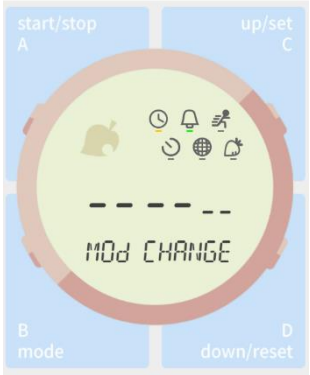
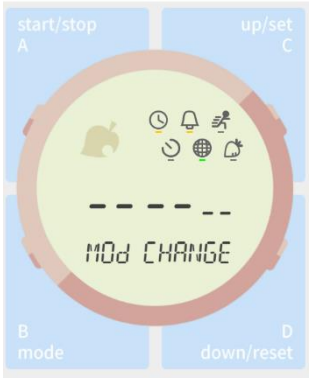
17	Set Time Zone	사용자가 현재 시간을 World Time의 시간으로 설정	 C버튼으로 시간대 변경 B버튼으로 TimeKeeping 모드 진입하여 변경된 시간 확인
18	Set Price	사용자가 무 값을 입력	 C버튼으로 무 값 설정 모드 진입 C, D버튼으로 값 변경  C, D버튼으로 값 변경 B버튼으로 변경 완료 변경된 무 값 확인

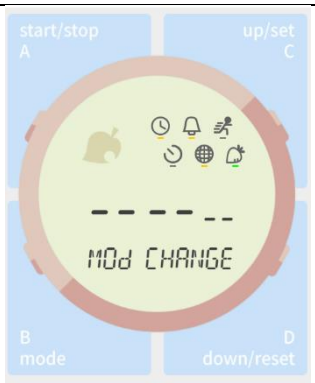
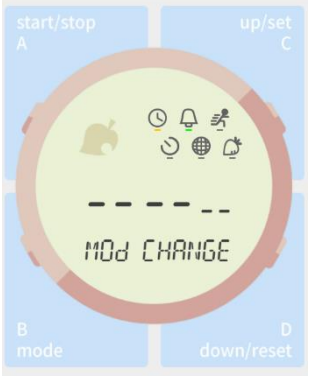
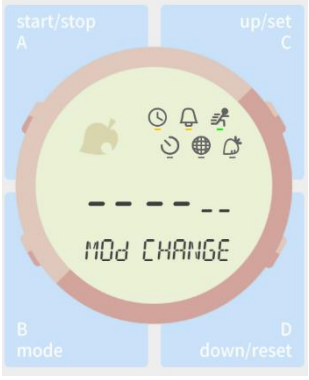
19	Reset Price	사용자가 무 값을 초기화	

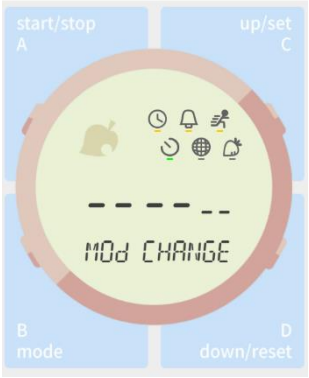
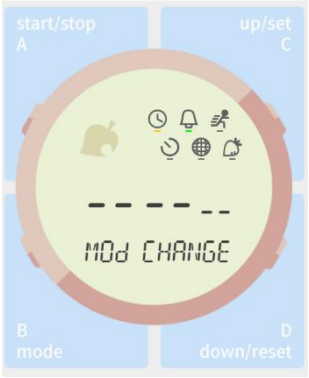
			 start/stop A up/set C -000-- AM- tUE--- PM- tUE--- PM- WEd--- AM- tHU--- PM- tHU--- AM- FRI--- B mode D down/reset

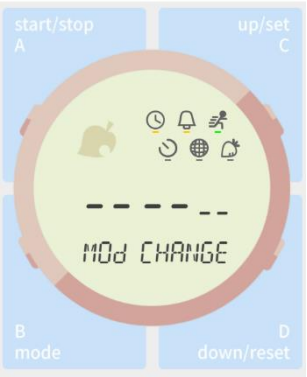
			 D버튼으로 무 값 초기화 A버튼으로 표시 날짜(시간) 변경하여, 다른 날짜(시간) 무 값도 초기화됨을 확인
20	Change Date	사용자가 무 값을 볼 날짜를 변경, 무 값을 입력할 날짜를 변경	  

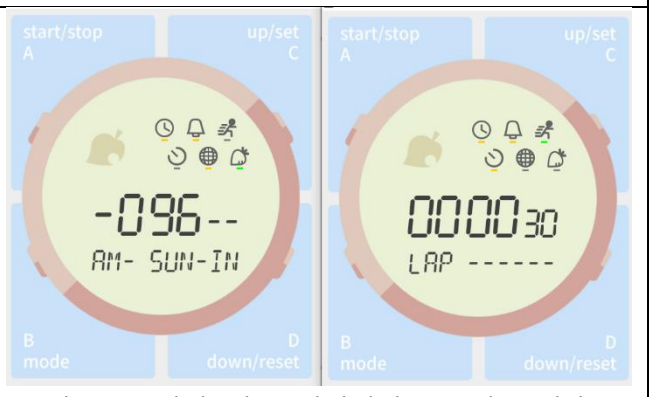
			 A버튼으로 표시할 무 값 날짜(시간) 변경 변경 및 표시 확인

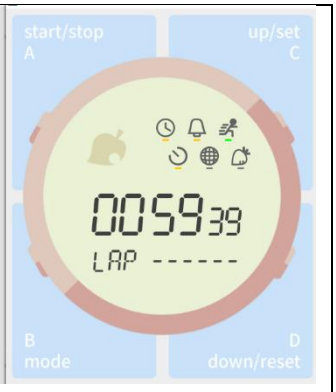
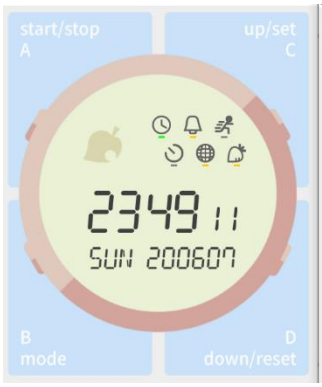
21	Change Mode	사용자가 시계의 모드를 변경	 B버튼으로 표시 모드 변경  전환된 화면 확인
22	Set Indicate Mode	사용자가 3가지의 모드를 선택	 C버튼으로 알람 모드 선택  A버튼으로 선택할 모드 변경 C버튼으로 세계시간 모드 선택

			 A버튼으로 선택할 모드 변경 C버튼으로 무 값 모드 선택  3개 모드 선택 시 설정 완료 후 TimeKeeping모드로 자동 진입 확인
23	Set Indicate Mode	사용자가 3개를 초과하여 모드 선택	 C버튼으로 알람 모드 선택  A버튼으로 선택할 모드 변경

			C버튼으로 스톱워치 모드 선택  A버튼으로 선택할 모드 변경 C버튼으로 타이머 모드 선택  4번째 모드 선택 전에 TimeKeeping으로 자동 복귀 확인
24	Set Indicate Mode	사용자가 3개 미만의 모드 선택	 B버튼으로 모드 설정 화면 진입. C버튼으로 알람 모드 선택

			 A버튼으로 선택 모드 변경 C버튼으로 스톱워치 모드 선택 B버튼으로 설정 완료  모드 변경 없이 B버튼으로 설정 완료 선택사항 변경 없이 TimeKeeping으로 자동 복귀 확인
25	Stop Beep	사용자가 울리는 알람을 종료	 23시 45분으로 알람 설정

			 TimeKeeping모드로 변경 후 현재 시간 확인  23시 45분에 알람이 울림을 확인  D버튼 클릭, 알람 종료 확인
26	Check Timeout	사용자의 입력이 60초간 없을 경우, Time Keeping Mode로 변경	 무 값 모드 진입. 다른 시계에선 스톱워치 시작

			  입력 없이 경과된 시간 확인   60초 경과 시 TimeKeeping 모드로 자동 복귀함을 확인

Activity 2066. Testing Traceability Analysis

No.	Operations in sequence diagram
1	reqChangeTimeFormat()
2	reqSetting()
3	nextUnit()
4	changeUnitValue()
5	reqCompleteSetting()
6	reqStartTimer()
7	reqPauseTimer()
8	reqResetTimer()
9	reqStartStopwatch()
10	reqPauseStopwatch()
11	reqResetStopwatch()
12	reqLapTime()
13	reqActivateAlarm()
14	reqDectivateAlarm()
15	reqChangeIndicatedAlarm()
16	reqChangeWorldTime()
17	reqChangeTimeZone()
18	reqChangePriceValue()
19	reqResetPrice()
20	reqChangeDate()
21	reqModeSwitch()
22	reqSetIndicateMode()
23	reqNextIndicator()
24	reqSelectMode()
25	reqUnselectMode()
26	reqCancelSetIndicateMode()
27	reqStopBeep()

Connectivity	method	class
2	getAlarmTime() : LocalTime	Alarm
5	setAlarmTime() : void	
13	activateAlarm() : void	
14	deactivateAlarm() : void	
	reqBeep() : void	Buzzer
27	stopBeep() : void	
1	reqChangeTimeFormat() : void	Controller
2	reqSetting() : void	
3	nextUnit() : void	
3	increaseUnit() : void	
3	initUnit() : void	
4	changeUnitValue(changeValue : int) : void	
5	reqCompleteSetting() : void	
6	reqStartTimer() : void	
7	reqPauseTimer() : void	
8	reqResetTimer() : void	
9	reqStartStopwatch() : void	
10	reqPauseStopwatch() : void	
11	reqResetStopwatch() : void	
12	reqLapTime() : void	

No.	System Test	class
1	Set Hour Format Test	TimeKeeping
2	Set Time Test	
3	Start Timer Test	Timer
4	Set Timer Test1	
5	Set Timer Test2	
6	Pause Timer Test	
7	Reset Timer Test	
8	Start Stopwatch Test	Stopwatch
9	Pause Stopwatch Test	
10	Reset Stopwatch Test	
11	Record Lap Time Test	
12	Set Alarm Test	Alarm
13	Activate Alarm Test	
14	Deactivate Alarm Test	
15	Indicate Another Alarm Test	
16	Change World Time Test	WorldTime
17	Set Time Zone Test	
18	Set Price Test	TurnipPrice
19	Reset Price Test	
20	Change Date Test	
21	Change Mode Test	ModeSwitch
22	Set Indicate Mode1	
23	Set Indicate Mode2	
24	Set Indicate Mode3	
25	Stop Beep	Buzzer
26	Check Timeout	Timeout

Connectivity	method	class
13	reqActivateAlarm() : void	Controller
14	reqDeactivateAlarm() : void	
15	reqChangeIndicatedAlarm() : void	
16	reqChangeWorldTime() : void	
17	reqChangeTimeZone() : void	
18	reqChangePriceValue(changeValue : int) : void	
19	reqResetPrice() : void	
20	reqChangeDate() : void	
21	reqModeSwitch() : void	
22	reqSetIndicateMode() : void	
23	reqNextIndicator() : void	
24	reqSelectMode() : void	
25	reqUnselectMode() : void	
26	reqCancelSetIndicateMode() : void	
27	reqStopBeep() : void	
9	startStopwatch() : void	Stopwatch
10	pauseStopwatch() : void	
11	resetStopwatch() : void	
12	lapTime() : void	
	initialize() : void	ModeSwitch
23	nextMode() : void	
24	setMode() : void	
2	getCurrentTime() : ZonedDateTime	TimeKeeping
5	setTime(Time : ZonedDateTime) : void	
17	setTimeZone(timeZoneToChange : ZonedId) : void	

Connectivity	method	class
	setWaitTime(time : LocalTime) : void	Timeout
6	startTimer(runTime : LocalTime) : void	Timer
2	getTimerTime() : LocalTime	
5	setTimerTime(time : LocalTime) : void	
7	pauseTimer() : void	
8	resetTimer() : void	
5	calcPrice(inputPrice : int[], isInputted : boolean[]) : int	TurnipCalc
5	setTurnipPrice(priceValue : int) : void	TurnipPrice
5	setHighst(date) : void	
19	resetPrice() : void	
20	nextPrice() : void	
16	nextWorldTime() : void	WorldTime
17	changeTimeZone() : void	