



Unit Test 결과 보고

about Coffee Machine System

2016.11.7

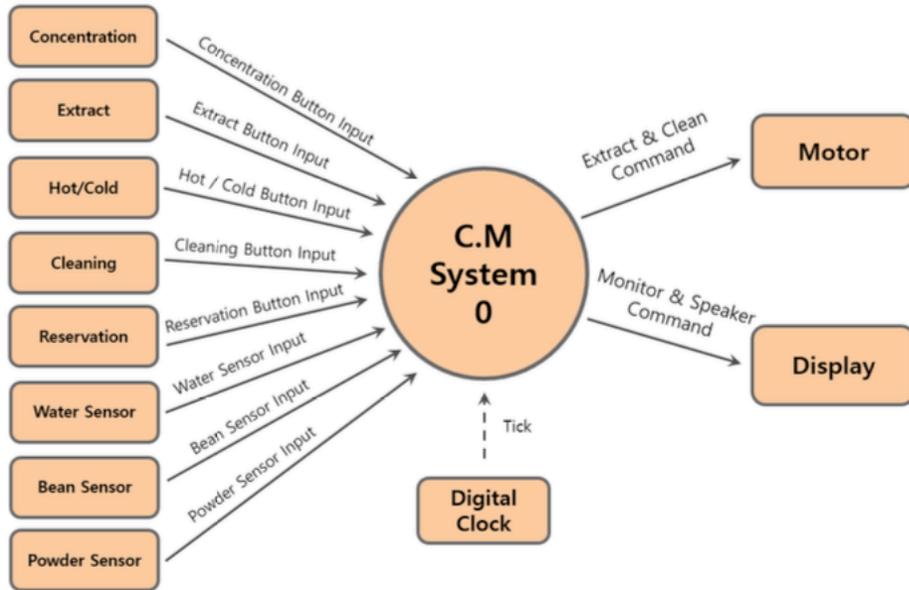
by T5



피드백 반영 사항

피드백 반영 사항 - SRA (2.0 -> 2.3)

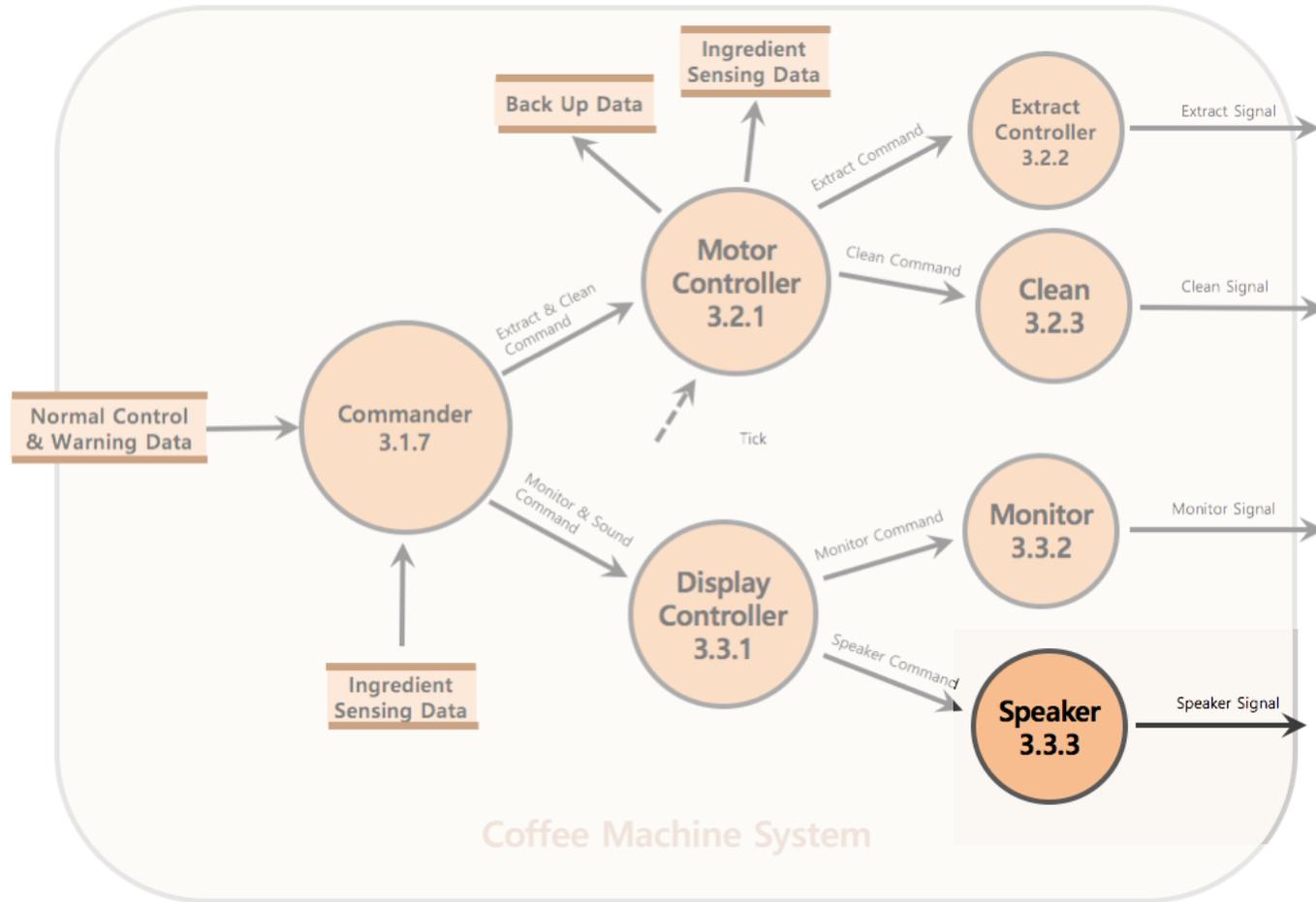
✓ Process 0의 Output을 **Work & Notify**로 명세한 것을 수정



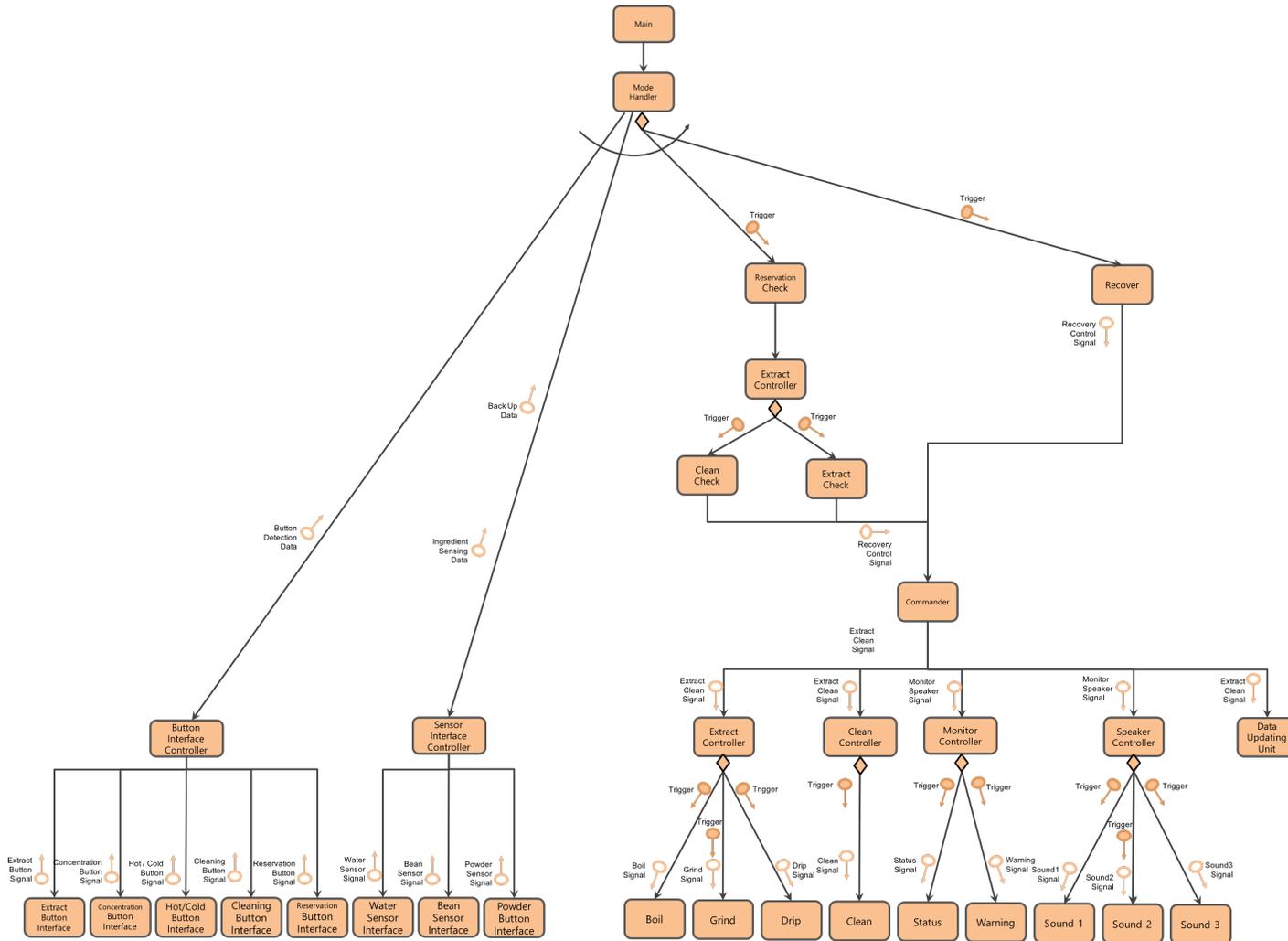
Reference No.	0
Name	C.M System
Input	Concentration Button Input, Hot/Cold Button Input, Extract Coffee Button Input, Cleaning Button Input, Reservation Button Input, Water Sense Input, Bean Sense Input, Powder Sense Input
Output	Extract & Clean Command, Monitor & Speaker Command
Process Description	사용자가 입력한 버튼과 Sensor 로부터 받은 정보를 input 으로 하여 C.M System 내에서 사용자가 요구한 동작에 가능 여부와 어떻게 동작을 실행 시킬지에 대한 중추적인 판단을 한다.

피드백 반영 사항 - SRA (2.0 -> 2.3)

- ✓ Speaker Process의 Output을 명세에 맞게 변경



피드백 반영 사항 - SDS (1.0 -> 1.5)



- ✓ Commander에 달린 다이아몬드 제거
- ✓ Commander 하위 4개 프로세스에 다이아몬드 추가
- ✓ Data Updating Unit으로 나가는 data 추가

SA 문서 수정 사항 - SRA (2.0 -> 2.3)

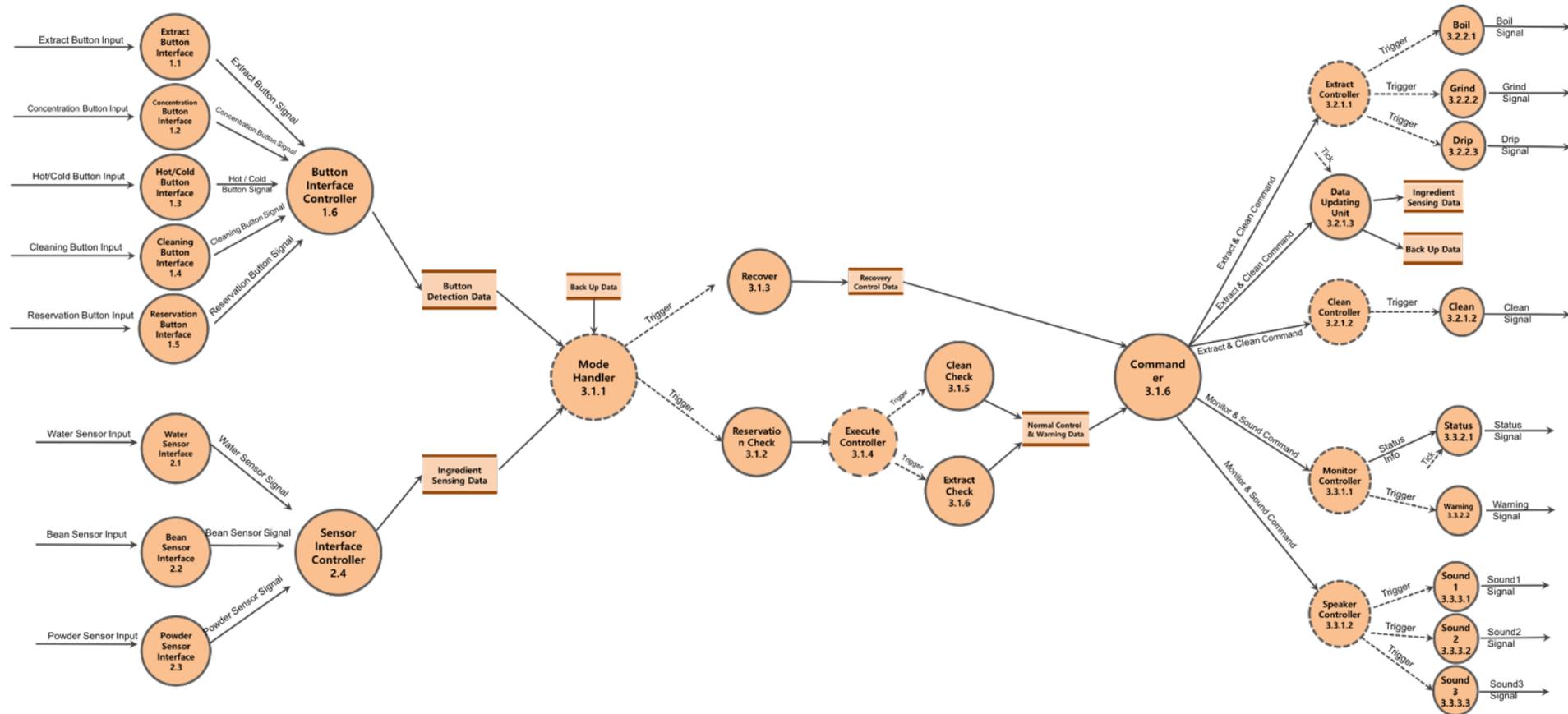
- (좌)-예약 수행 도중 강제 종료 이후에 작업 복원 후 예약 다시 수행을 방지하는 내용 추가
- (우)-Extract Controller(3.1.4)에서 Extract Check(3.1.6)으로 인자를 보낼 때 재료값 뿐만 아니라 온도 설정과 물의 양도 인자로 전달해야 함.

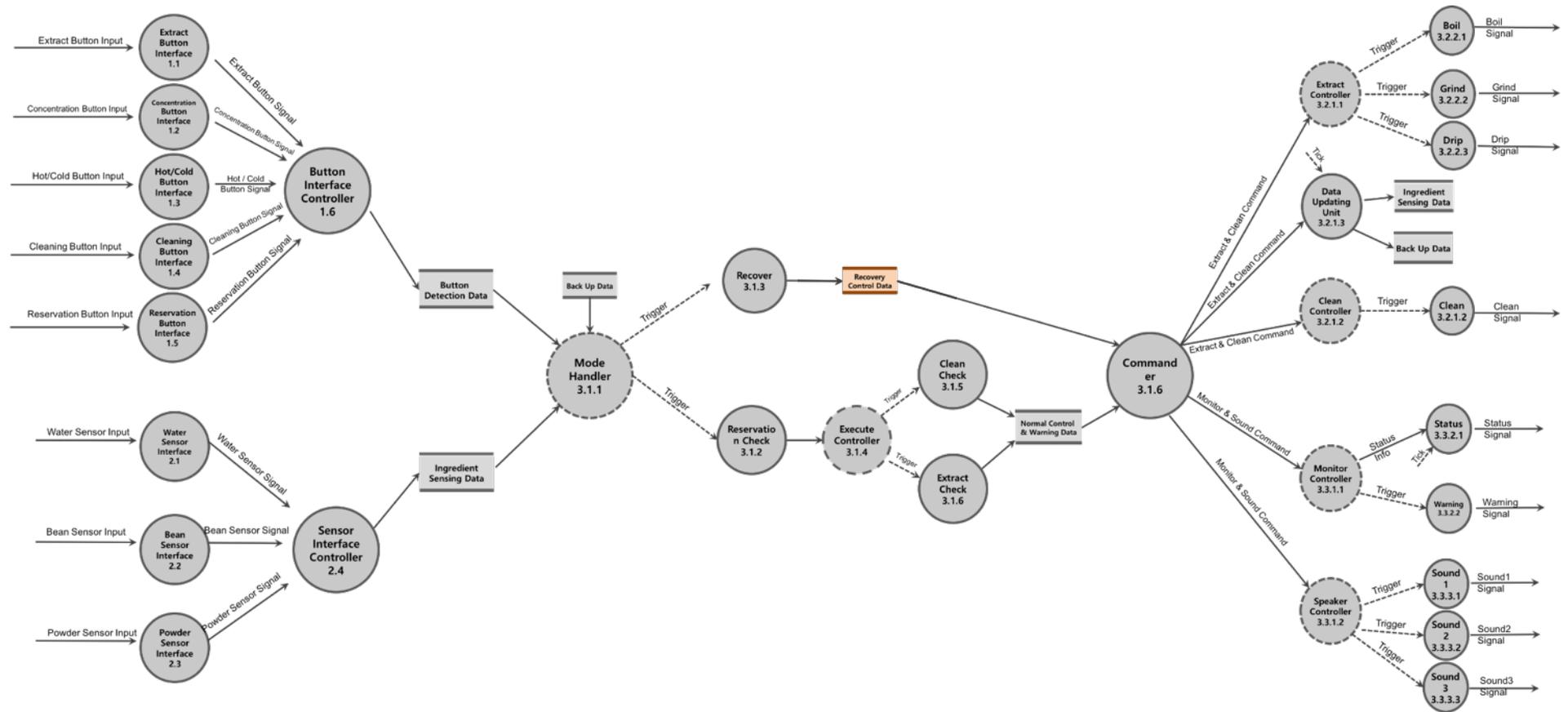
Reference No.	3.1.2
Name	Reservation Check
Input	Trigger Reservation Check
Output	Execute Info
Process Description	<p>Button Detection Data & Ingredient Sensing Data 에 저장된 예약시간과 현재시간을 비교하여 현재 시간이 예약 시간과 동일한지 판단한다. 만약 현재 시간이 예약시간과 같고 예약 정보를 Execute Controller에 전송하지 않았다면 Button Detection 데이터의 예약정보를 가지고 청소 예약이라면 Execute Controller에 Execute Info를 전송한다. 추출 예약이라면 Button Detection Data & Ingredient Sensing Data를 Execute Controller에 전송한다.</p> <p>현재시간이 예약시간이 아니라면 예약정보를 포함하지 않은 Button Detection Data & Ingredient Sensing Data 를 가지고 청소명령이 들어왔다면 Execute Controller에 Button Detection Data & Ingredient Sensing Data를 전송하고, 추출명령이 들어왔다면 Execute Info를 Execute Controller에 전송한다.</p>

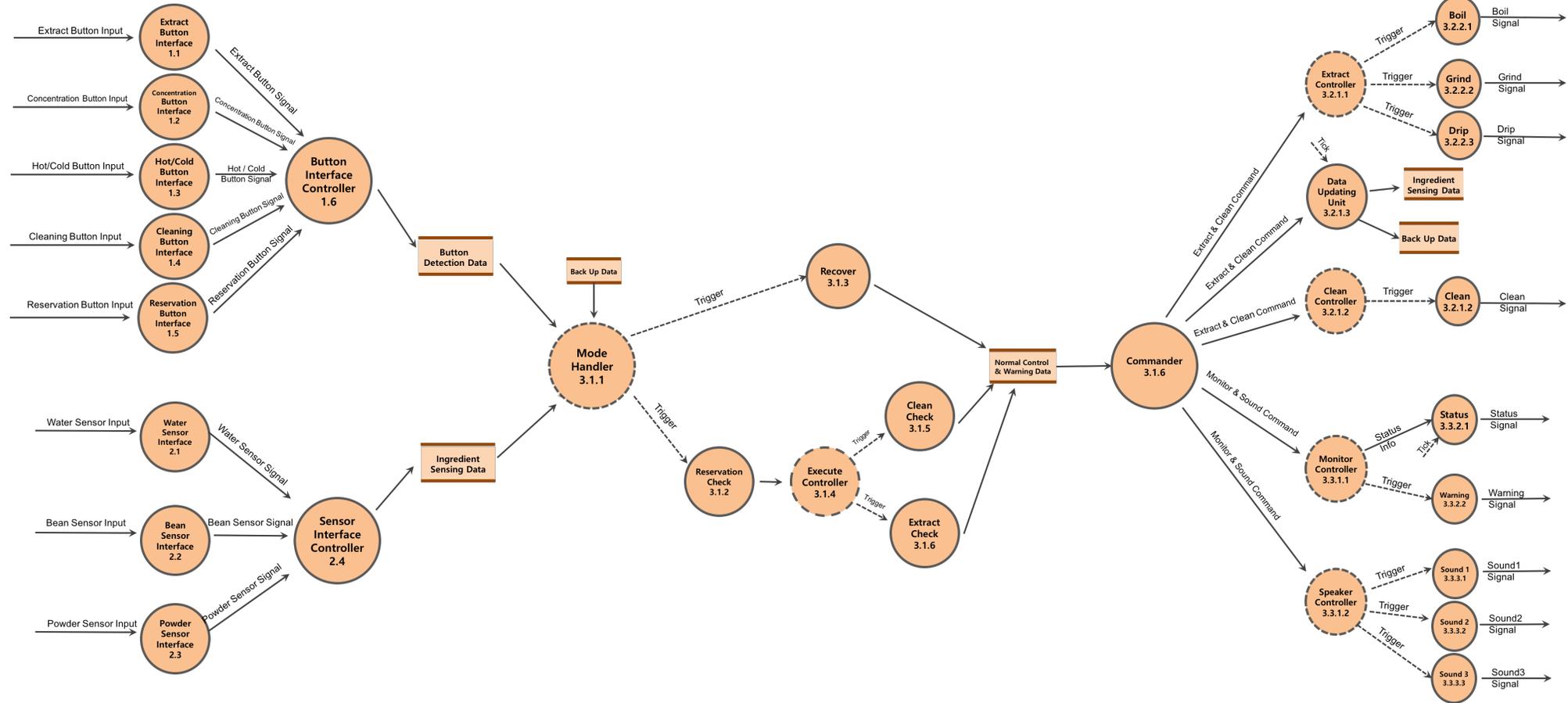
Reference No.	3.1.4
Name	Execute Controller
Input	Execute Info
Output	Trigger Clean Check, Trigger Extract Check
Process Description	<p>Execute Info의 명령정보를 받아서 청소명령일 경우 Clean Check에 물의 양을 전송하고 Trigger한다. 추출명령일 경우 Extract Check에 각각 재료의 양과 온도 설정 그리고 물의양을 전송하고 Trigger한다.</p>

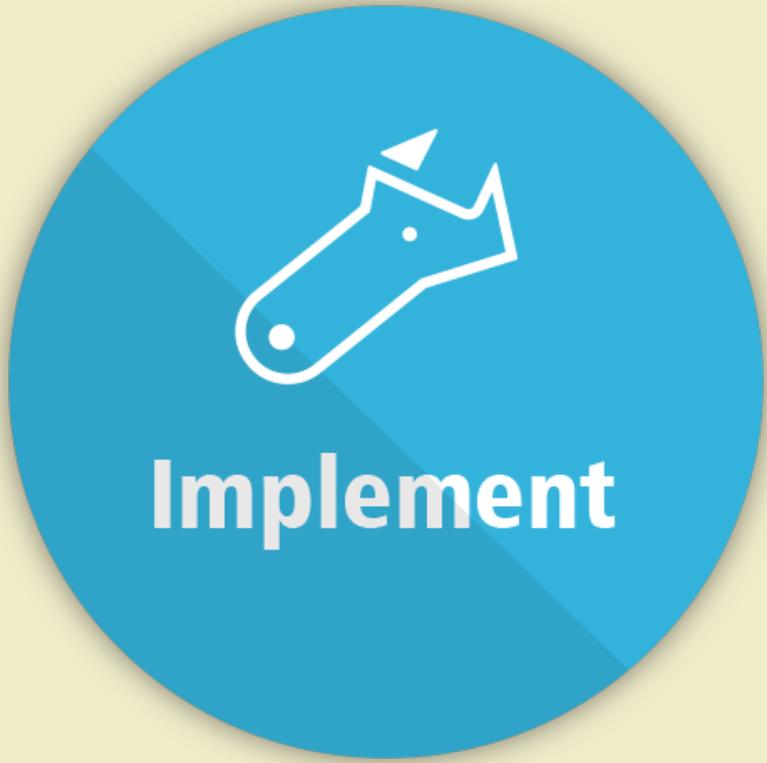
SA 문서 수정 사항 - SRA (2.0 -> 2.3)

- Commander(3.1.6)에서 참조하는 데이터가 Recovery Control Data와 Normal Control & Warning Data가 동일하여 두 data store를 하나로 통합



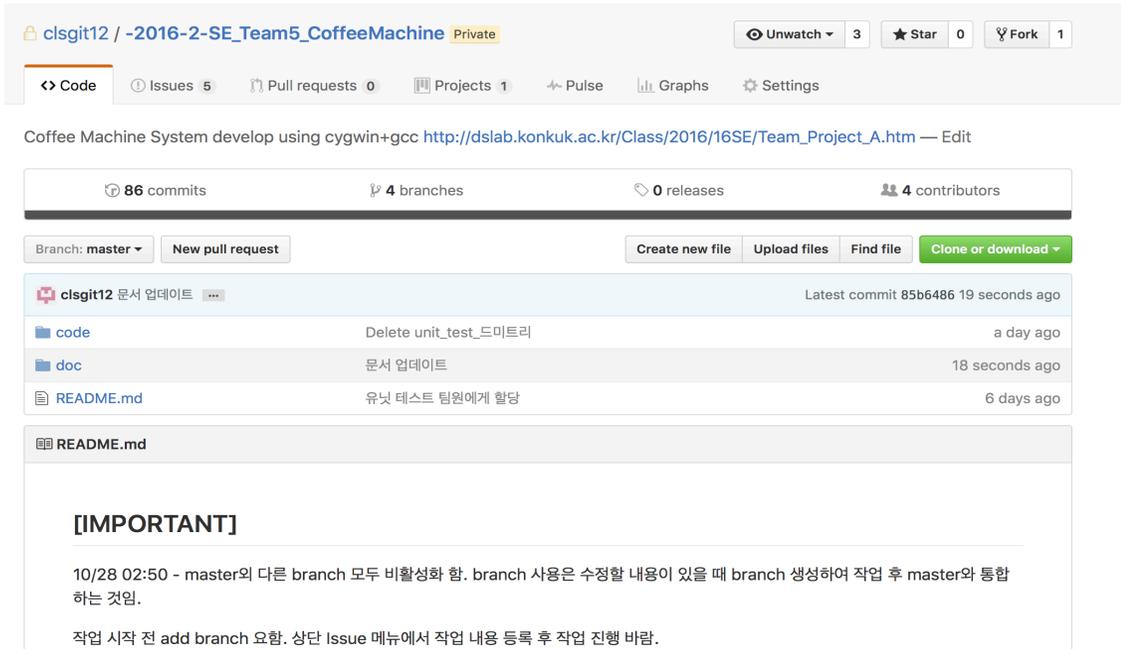




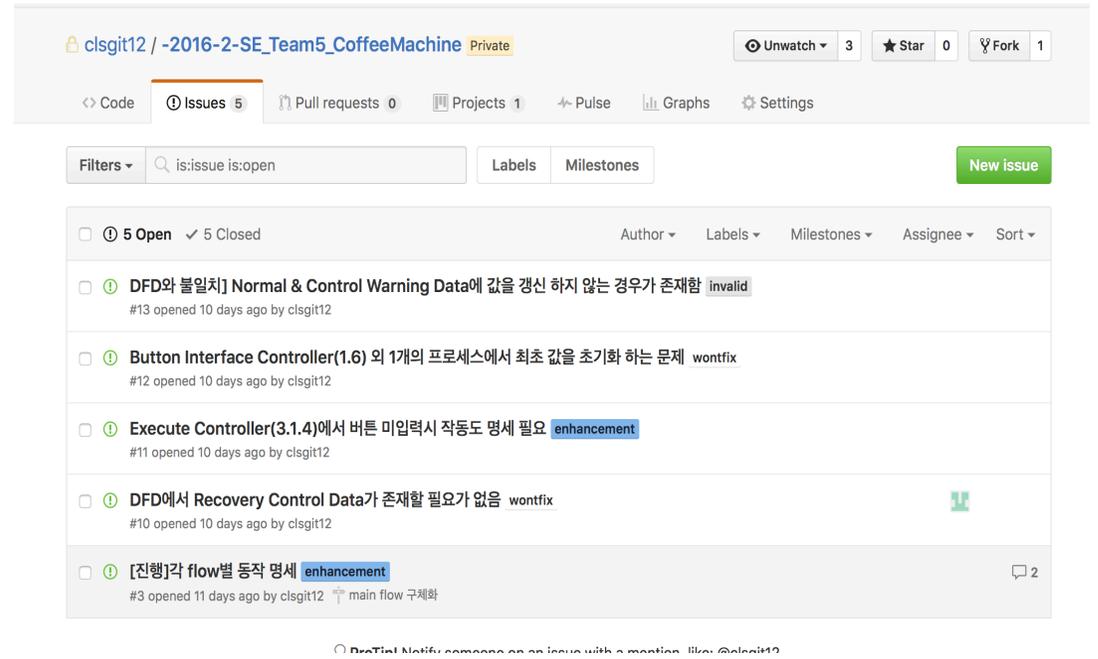


구현

- For issue and change tracking



The screenshot shows the repository page for 'clsgit12 / -2016-2-SE_Team5_CoffeeMachine'. The repository is private and has 3 unwatchers, 0 stars, and 1 fork. The navigation bar includes links for Code, Issues (5), Pull requests (0), Projects (1), Pulse, Graphs, and Settings. The repository description is 'Coffee Machine System develop using cygwin+gcc' with a link to a class page. Below the description, there are statistics: 86 commits, 4 branches, 0 releases, and 4 contributors. A 'Branch: master' dropdown and a 'New pull request' button are visible. There are also buttons for 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. The file list shows 'code' (Delete unit_test_드미트리, a day ago), 'doc' (문서 업데이트, 18 seconds ago), and 'README.md' (유닛 테스트 팀원에게 할당, 6 days ago). The README content includes an important notice: '10/28 02:50 - master의 다른 branch 모두 비활성화 함. branch 사용은 수정할 내용이 있을 때 branch 생성하여 작업 후 master와 통합 하는 것임. 작업 시작 전 add branch 요함. 상단 Issue 메뉴에서 작업 내용 등록 후 작업 진행 바람.'



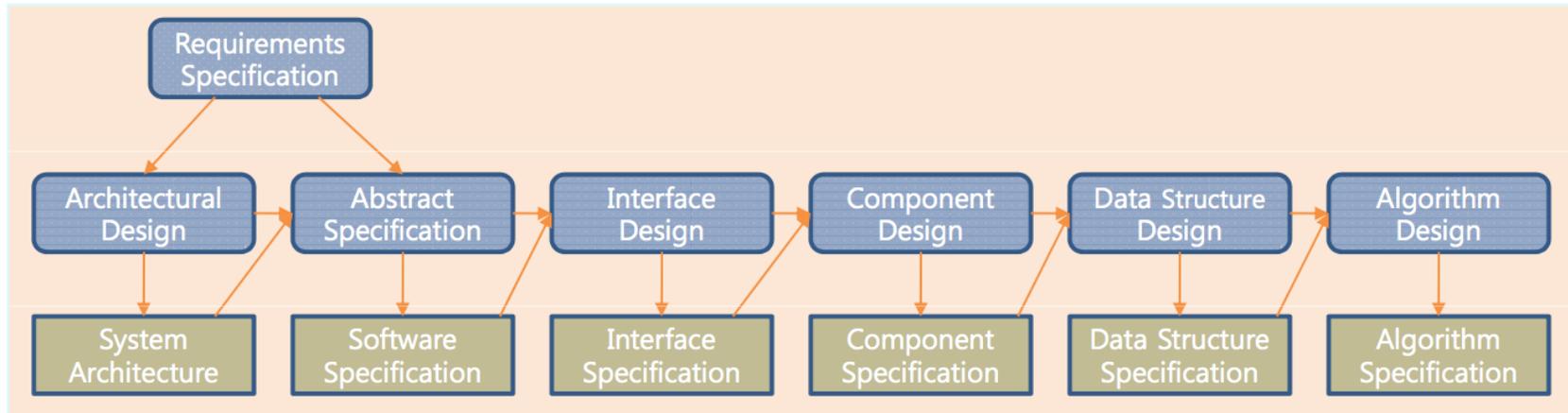
The screenshot shows the Issues page for the same repository. The navigation bar is the same as the repository page. The 'Issues' tab is selected, showing 5 issues. There are filters for 'is:issue is:open', 'Labels', and 'Milestones'. A 'New issue' button is in the top right. The issues list shows 5 open issues and 5 closed issues. The issues are:

- #13 [DFD와 불일치] Normal & Control Warning Data에 값을 갱신 하지 않는 경우가 존재함 **invalid** (opened 10 days ago by clsgit12)
- #12 Button Interface Controller(1.6) 의 1개의 프로세스에서 최초 값을 초기화 하는 문제 **wontfix** (opened 10 days ago by clsgit12)
- #11 Execute Controller(3.1.4)에서 버튼 미입력시 작동도 명세 필요 **enhancement** (opened 10 days ago by clsgit12)
- #10 DFD에서 Recovery Control Data가 존재할 필요가 없음 **wontfix** (opened 10 days ago by clsgit12)
- #3 [진행]각 flow별 동작 명세 **enhancement** (opened 11 days ago by clsgit12, main flow 구체화, 2 comments)

At the bottom, there is a notification: 'DoTial Notify someone on an issue with a mention: @clsgit12'

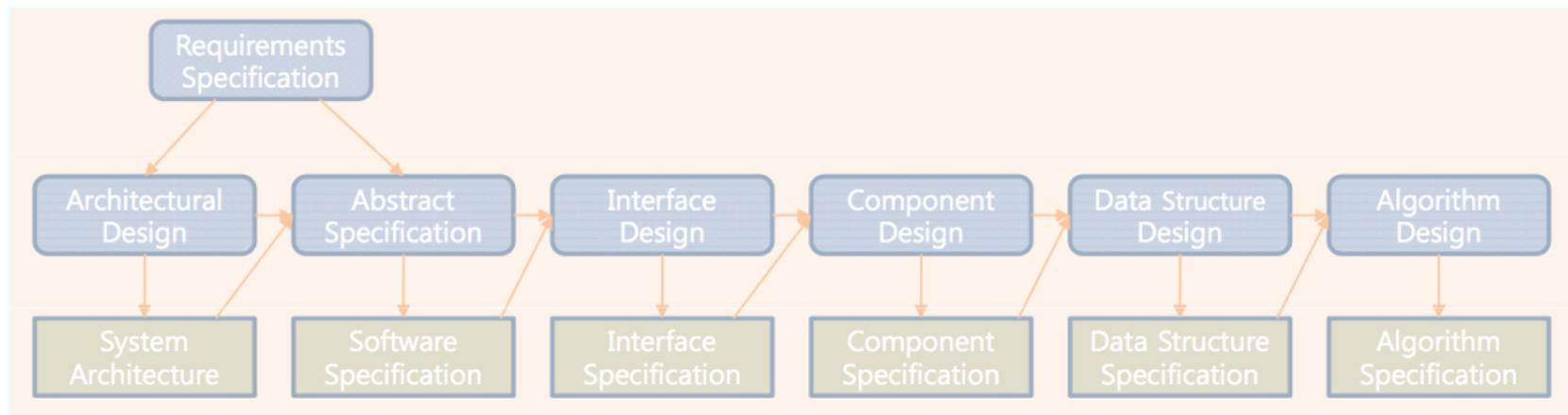
2. Software Design and Implementation

- Process of converting system specification into executable system.
- Software design
 - Design a software structure to realize the specification
- Implementation
 - Translate the design structure into an executable program.
 - Programming is a personal activity. No generic programming process.
- Software design process



2. Software Design and Implementation

- Process of converting system specification into executable system.
- Software design
 - Design a software structure to realize the specification
- Implementation
 - Translate the design structure into an executable program.
 - Programming is a personal activity. No generic programming process.
- Software design process



2. Software Design and Implementation

1. Process of converting system specification into executable system.

2. Software design

1. Design a software structure to realize the specification.

Programming is a personal activity.

3. Software design process

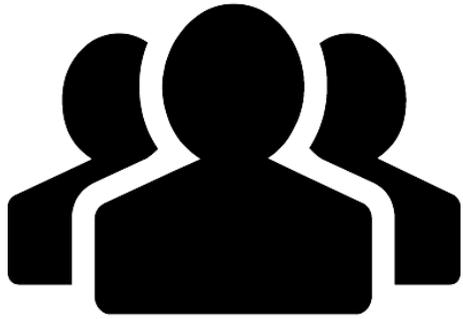




유닛 테스트

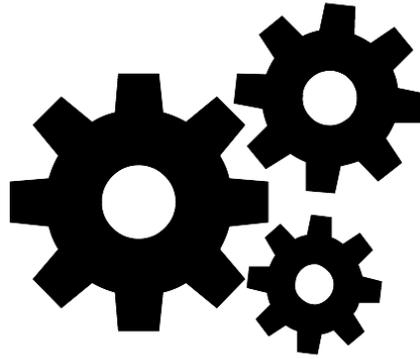
(using CuTest)

Unit Test - 개요



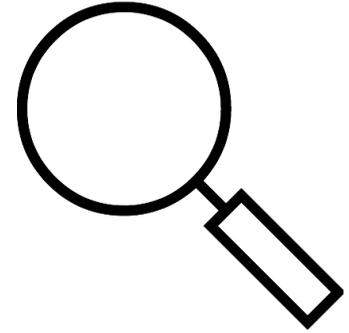
4

테스트 인원



18

테스트 대상 모듈



95

테스트 케이스

테스트 케이스는 독립적이어야 할 것

- 다른 모듈이나 시스템을 사용하는 연결부 배제
- 서드 파티 라이브러리와 종속성 제거

테스트 더블을 사용

- 원래의 로직을 간소화된 코드로 대체

Unit Test – 테스트 더블 예시

```
1  /*
2  * Reference No.
3  * 3.3.3.1
4  * Name
5  * Monitor Controller
6  * Input
7  * Monitor & Speaker Command
8  * Output
9  * Status Info, Trigger Warning
10 * Process Description
11 * Monitor & Speaker Command에 담긴 경보음 발생 여부 값과 화면에 출력할 정보를 받는다.
12 * Status 프로세스에는 화면에 출력할 정보를 Status Info로 보낸다.
13 * 화면에 출력할 정보는 다음과 같다.
14 * [현재 시간, 현재 상태, 선택 농도, 선택 온도, 물 잔량, 커피 잔량(원두), 가루 커피 유무, 커피 가루 정소 필요, 커피 예약 시간, 청소 예약 시간]
15 * 다음 상황일 경우에 한해 Warning 프로세스가 알람을 화면에 출력하도록 정보를 보내고 트리거한다.
16 * 상황 : [물부족, 원두부족, 커피가루제거필요]
17 */
18
19
20 void monitor_controller(struct tm* currentTime, struct ButtonDetectionData* buttonDetectionData, struct IngredientSensingData* ingredientSensingData, struct BackUpData* backUpData)
21 {
22     if (normalControlWarningData->warning == WARNING_WATER_SHORT ||
23         normalControlWarningData->warning == WARNING_BEAN_SHORT ||
24         normalControlWarningData->warning == WARNING_POWDER_CLEAN) {
25         Warning(normalControlWarningData->warning);
26         normalControlWarningData->warning = 0;
27     }
28     Status(currentTime, buttonDetectionData, ingredientSensingData, backUpData);
29 }
```

(3.3.3.1) Monitor Controller의 원 함수

Unit Test – 테스트 더블 예시

```
30
31
32  int output = -1;
33  int is_warning = -1;
34
35  void monitor_controller(int need_water, int need_bean, int must_clean_powder, int warning) {
36
37      is_warning = warning;
38
39      if ( need_water == 1 ||
40          need_bean == 1 ||
41          must_clean_powder == 1) {
42          output = 1;
43          is_warning = 0;
44      } else {
45          output = 0;
46      }
47  }
48
```

(3.3.3.1) Monitor Controller의 테스트 더블

Unit Test – Test case 예시 (Recover)

Identifier	명령 정보	온도	농도	결과
CMS.UTC.3130.003	청소	X	X	CLEAN
CMS.UTC.3130.004	청소	온	200	CLEAN
CMS.UTC.3130.005	예약 청소	X	X	CLEAN
CMS.UTC.3130.006	예약 청소	온	200	CLEAN
CMS.UTC.3130.007	추출	온	100	HOT_100
CMS.UTC.3130.008	추출	온	200	HOT_200
CMS.UTC.3130.010	추출	냉	100	COLD_100
CMS.UTC.3130.011	추출	냉	200	COLD_200
CMS.UTC.3130.013	예약 추출	온	100	HOT_100
CMS.UTC.3130.014	예약 추출	온	200	HOT_200
CMS.UTC.3130.016	예약 추출	냉	100	COLD_100

(3.1.3) Recover의 Test case

프로그램 실행 화면

```
-----MENU-----
E Extract      추 출
C Clean        청 소
R Reservation  예 약
S Supplement   재 료 보 충
V Volume       농 도 설 정
T Temperature 온 도 토 글
D Dump Powder 커피 가 루 청 소
Q Quit        종 료

-----STATUS DISPLAY-----
현재 시간 03:38:43
현재 상태 가 열 중
선택 농도 200ml
선택 온도 온
물 잔량 410/1000ml
원두 잔량 7/100g
커피 가루 유무 0
커피 가루 청소 필요 X
추출 예약 시간 ---:--
청소 예약 시간 ---:--
커피 예약 농도 --
커피 예약 온도 --
```

작동 예 - 커피 추출 중 화면



작동만 하게끔
만들어진 커피머신은
사용자를 불편하게 한다.