

Coffee Machine System

Team Presentation 4

TI

201211364 이경민

201211333 김영호

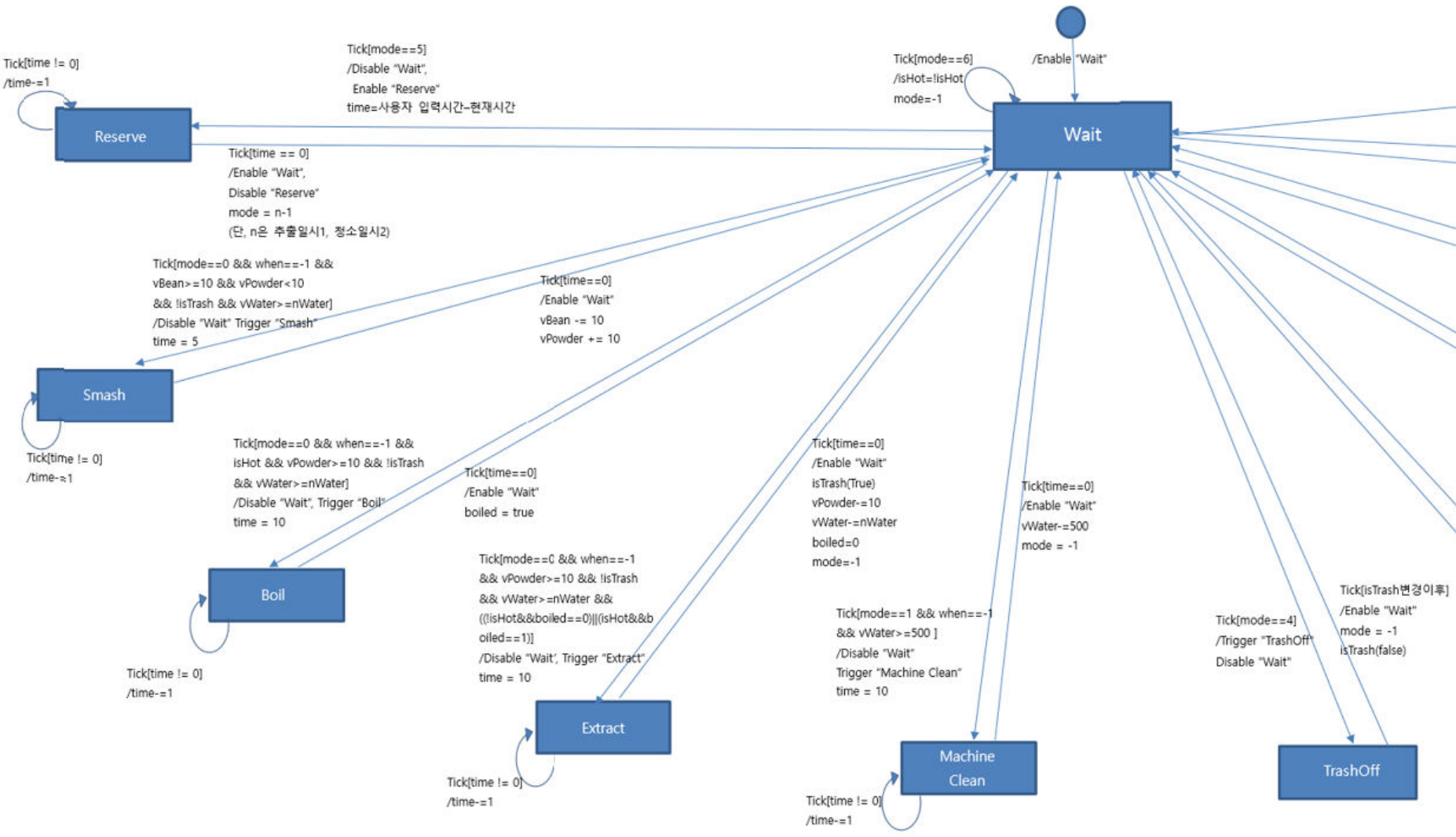
201211347 박성근

201211376 임제현

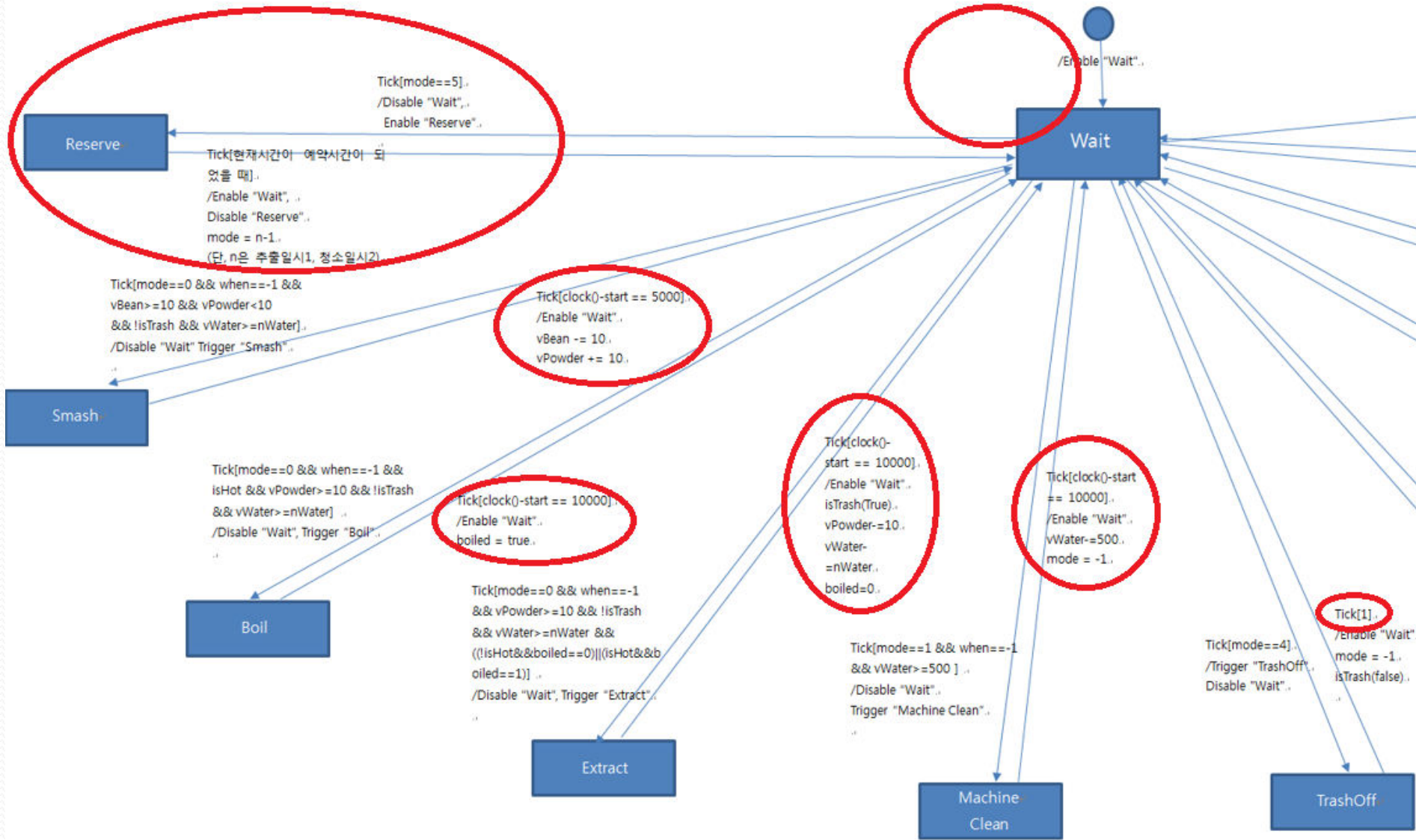
Index

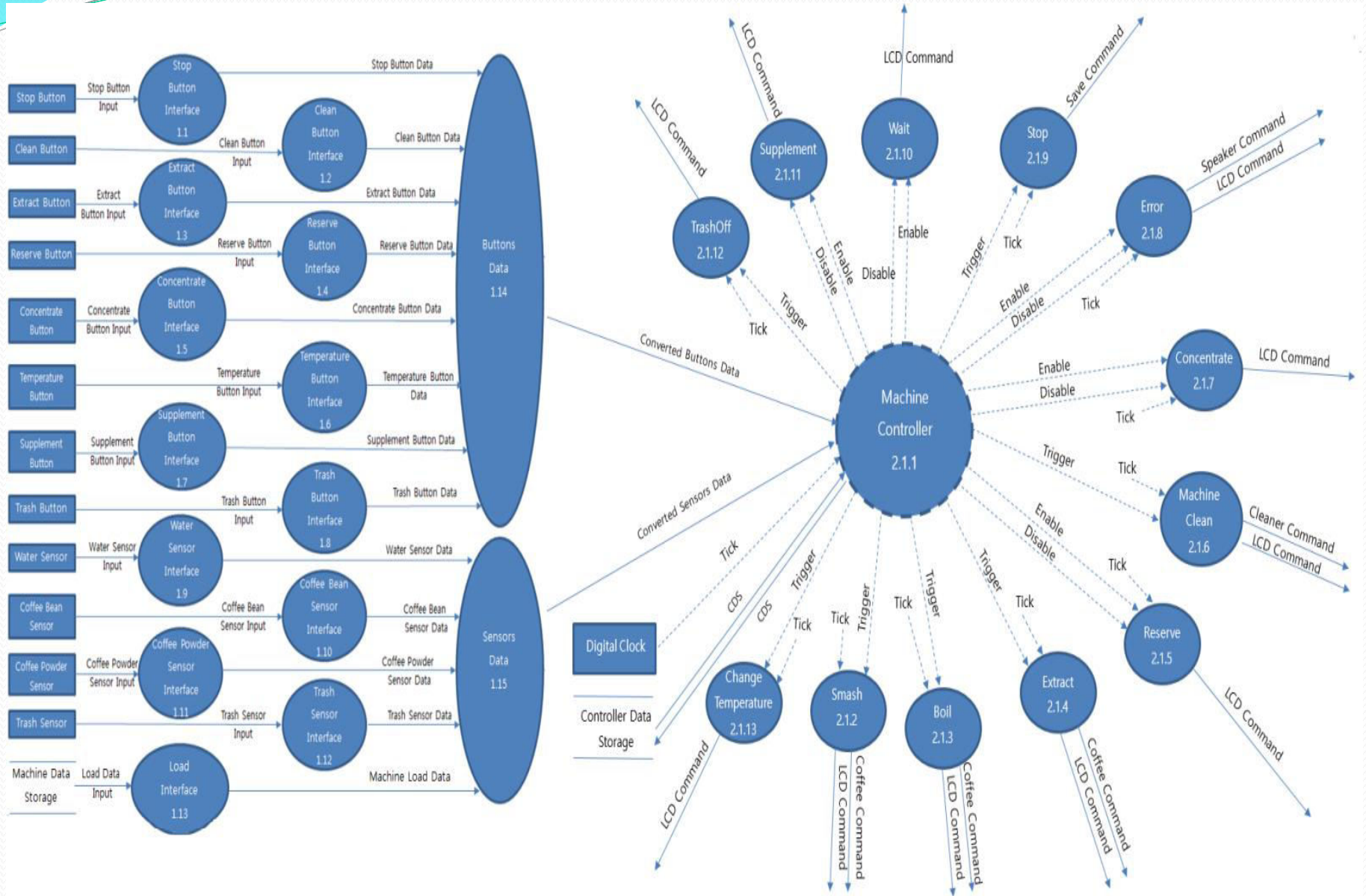
- Coffee Machine System
- 구현
- System Test List
- System Test Result

Coffee Machine System



Coffee Machine System





구현

```
int main(void){
    while(1){
        Machine_Controller(cBD, Sensors_Data(&vWater, &vBean, &vPowder, &isTrash, Load_Interface()));
    }
    return 0;
}
```

```
void Machine_Controller(struct converted_Buttons_Data cBD, struct converted_Sensors_Data cSD){
    digital_clock = time(NULL);
    t = localtime(&digital_clock);
    int hour = t->tm_hour;
    int min = t->tm_min;

    printf("#####\n");
    switch(cDS.mode){
        case -1 :
            wait(cSD, hour, min);
            cBD = Buttons_Data();
            cDS.mode = cBD.mode;
            break;
        case 0 :
            if(cDS.nWater>(*cSD.cSD_vWater)){
                Error(cSD, hour, min, 7);
                break;
            }
            if((*cSD.cSD_isTrash ==true)){
                Error(cSD, hour, min, 13);
                break;
            }
            if((*cSD.cSD_vBean)<10 && (*cSD.cSD_vPowder)<10){
                Error(cSD, hour, min, 12);
                break;
            }
            if(cDS.when== -1 && (*cSD.cSD_vBean)>=10 && (*cSD.cSD_vPowder)<10 && (*cSD.cSD_isTrash)==false && (*cSD.cSD_vWater)>=cDS.nWater){
                Smash(cSD, hour, min);
                break;
            }
            if(cDS.when== -1 && cDS.isHot == true && cDS.boiled == false && (*cSD.cSD_vPowder)>=10 && (*cSD.cSD_isTrash)==false && (*cSD.cSD_vWater)>=cDS.nWater){
                Boil(cSD, hour, min);
                break;
            }
            if(cDS.when== -1 && (*cSD.cSD_vPowder)>=10 && (*cSD.cSD_isTrash)==false && (*cSD.cSD_vWater)>=cDS.nWater && ((cDS.isHot==false && cDS.boiled==false) || (cDS.isHot==true && cDS.boiled==true))){
                Extract(cSD, hour, min);
                break;
            }
            break;
        case 1 :
            if((*cSD.cSD_vWater)<500){
                Error(cSD, hour, min, 7);
                break;
            }
    }
}
```

구현

CDS	커피 머신의 상태 데이터	Structure
	mode (int) : 입력 받은 버튼의 종류(mode 의 초기화 값은 -1로 설정한다.)	
	when (int) : 예약 시간 값	
	isHot (boolean) : 추출해야 하는 커피의 온도	
	nWater (int) : 현재 농도 설정 값	
	boiled (boolean) : 현재 추출하는 커피의 온도	

```
void wait(struct converted_Sensors_Data D, int h, int m){
    struct lcd_command lc = {CDS.when, CDS.mode, CDS.isHot, CDS.nWater, D.CSD_vWater, D.CSD_vBean, D.CSD_vPowder, D.CSD_isTrash, 8, h, m};
    LCD_Interface(lc);
}
```

```
struct converted_Buttons_Data Buttons_Data(){
    struct converted_Buttons_Data ret_cBD;
    int input;
    scanf("%d", &input);

    if(input==0){
        ret_cBD.mode = Extract_Button_Interface();
    }

    return ret_cBD;
}
```

```
int Extract_Button_Interface(){
    int mode = 0;
    return mode;
}
```

```
        case 6 :
            Change_Temperature();
            break;
    }

    if(cBD.isStop==true){
        Stop(cSD);
    }

    printf("#####\n");
}
```

```
void Error(struct converted_Sensors_Data D, int h, int m, int a){
    struct lcd_Command lc = {cDS.when, cDS.mode, cDS.isHot, cDS.nWater, D.cSD_vWater, D.cSD_vBean, D.cSD_vPowder, D.cSD_isTrash, a, h, m};

    LCD_Interface(lc);
    Speaker_Interface(a);
    cDS.mode = -1;
}
```


구현

```
struct coffee_Command Smash(struct converted_Sensors_Data D, int h, int m){
    struct lcd_Command lc = {cDS.when, cDS.mode, cDS.isHot, cDS.nWater, D.cSD_vWater, D.cSD_vBean, D.cSD_vPowder, D.cSD_isTrash, 1, h, m};
    LCD_Interface(lc);

    clock_t start = clock();
    while(clock() - start < 5000){}

    struct coffee_Command cc = {0, cDS.nWater, D.cSD_vWater, D.cSD_vBean, D.cSD_vPowder, D.cSD_isTrash, cDS.boiled};

    *D.cSD_vBean -= 10;
    *D.cSD_vPowder += 10;

    Coffee_Interface(cc);
    return cc;
}

boolean Boil(struct converted_Sensors_Data D, int h, int m){
    struct lcd_Command lc = {cDS.when, cDS.mode, cDS.isHot, cDS.nWater, D.cSD_vWater, D.cSD_vBean, D.cSD_vPowder, D.cSD_isTrash, 2, h, m};
    LCD_Interface(lc);

    clock_t start = clock();
    while(clock() - start < 10000){}

    struct coffee_Command cc = {1, cDS.nWater, D.cSD_vWater, D.cSD_vBean, D.cSD_vPowder, D.cSD_isTrash, cDS.boiled};

    Coffee_Interface(cc);
    cDS.boiled = true;
    return cDS.boiled;
}

struct coffee_Command Extract(struct converted_Sensors_Data D, int h, int m){
    struct lcd_Command lc = {cDS.when, cDS.mode, cDS.isHot, cDS.nWater, D.cSD_vWater, D.cSD_vBean, D.cSD_vPowder, D.cSD_isTrash, 3, h, m};
    LCD_Interface(lc);

    clock_t start = clock();
    while(clock() - start < 10000){}

    struct coffee_Command cc = {2, cDS.nWater, D.cSD_vWater, D.cSD_vBean, D.cSD_vPowder, D.cSD_isTrash, cDS.boiled};

    *D.cSD_vPowder -= 10;
    *D.cSD_vWater -= cDS.nWater;
    *D.cSD_isTrash = true;

    Coffee_Interface(cc);
    cDS.mode = -1;
    cDS.boiled = false;
    return cc;
}
```

구현

```
void Coffee_Interface(struct coffee_Command cC){
    struct coffee_Command Drop = cC;
    Coffee(Drop);
}
```

```
void Speaker(int Sound){
    if(Sound == 7){
        printf("뿅!\n");
    }
    else if(Sound == 12){
        printf("뿅-익 뿅-익 뿅-익\n");
    }
    else if(Sound ==13){
        printf("뿅 뿅!\n");
    }
}

void Coffee(struct coffee_Command Drop){
    if(Drop.m==2){
        printf("##### 커피가 나왔습니다 #####\n");
        if(Drop.boiled ==true){
            printf("뜨거운 ");
        }
        else if(Drop.boiled ==false){
            printf("차가운 ");
        }
        if(Drop.nWater==100){
            printf("진한 커피\n");
        }
        else if(Drop.nWater==200){
            printf("중간 커피\n");
        }
        else if(Drop.nWater==300){
            printf("연한 커피\n");
        }
        printf("#####\n");
    }
}
```

System Test List

Test Case Identification	Input Specification	Output Specification
CMS_STC_000_000	프로그램 시작	
CMS_STC_001_000	Menu Input = 0 농도 = 진하게 온도 = 냉 물 = 1000ml 커피 잔량 = 100g 가루 커피 유무 = o 찌꺼기 유무 = x	재료가 부족하지 않아 정상적으로 추출을 실행하고 차가운 진한 커피가 나온다. 물 = 900ml 가루 커피 유무 = x 찌꺼기 유무 = o
CMS_STC_001_001	Menu input = 0 농도 = 진하게 온도 = 냉 물 = 1000ml 커피 잔량 = 100g 가루 커피 유무 = x 찌꺼기 유무 = x	가루 커피가 없으므로 분쇄를 실행하고 추출을 진행한다. 물 = 900ml 커피 잔량 = 90g 가루 커피 유무 = x 찌꺼기 유무 = o
CMS_STC_001_002	Menu input = 0 농도 = 진하게 온도 = 온 물 = 1000ml 커피 잔량 = 100g 가루 커피 유무 = o 찌꺼기 유무 = x	온도가 온으로 설정되어 있으므로 가열 후 추출한다. 물 = 900ml 가루커피 유무 = x 찌꺼기 유무 = o
CMS_STC_001_003	Menu Input = 0 농도 = 진하게 온도 = 냉 물 = 1000ml 커피 잔량 = 0g 가루 커피 유무 = x 찌꺼기 유무 = x	분쇄를 하기에 충분한 커피가 없으므로 에러메시지 출력. 원두 부족 (10g) 빠 -익 빠 -익 빠 -익
CMS_STC_001_004	Menu input = 0 농도 = 진하게 온도 = 냉 물 = 0ml 커피 잔량 = 100g 가루 커피 유무 = x 찌꺼기 유무 = x	추출하기에 충분한 물이 없으므로 에러메시지 출력. 물 부족 (잔량 0ml) 빽!
CMS_STC_001_005	Menu input = 0 농도 = 진하게 온도 = 냉 물 = 1000ml 커피 잔량 = 100g 가루 커피 유무 = x 찌꺼기 유무 = o	커피 찌꺼기가 있으므로 에러메시지 출력. 커피 가루 제거 빽 빽!
CMS_STC_001_006	Menu input = 0 농도 = 진하게 온도 = 냉 물 = 0ml 커피 잔량 = 0g 가루 커피 유무 = x 찌꺼기 유무 = o	물, 커피 잔량, 찌꺼기 세개다 문제가 있지만 물부족 에러메시지만 출력 물 부족(잔량 0ml) 빽!

System Test List

CMS_STC_002_000	Menu input = 1 물 = 1000ml	청소를 한다. 물 = 500ml	CMS_STC_004_001	Menu input = 3 가루 커피 유무 = x 2 '커피 가루' 선택 후 1입력	정상 범위 내의 보충은 정상적으로 작동 한다. 가루 커피 유무 = o
CMS_STC_002_001	Menu input = 1 물 = 0ml	청소하기에 충분한 물이 없으므로 에러메 시지 출력. 물 부족 (잔량 0ml) 뵙!	CMS_STC_004_002	Menu input = 3 물 = 500ml 3 '물' 선택후 정상범위(1000ml이 내)로 365ml보충	정상 범위 내의 보충은 정상적으로 작동 한다. 물 = 865ml
CMS_STC_003_000	Menu input = 2 선택농도 = 진하게 1 '진하게' 로 설정.	선택된 농도로 선택 농도를 변경한다. 선택 농도 = 진하게	CMS_STC_004_003	Menu input = 3 물 = 500ml 커피 잔량 = 50g 가루 커피 유무 = x 1,2,3 이외의 값.	변경없음.
CMS_STC_003_001	Menu input = 2 선택농도 = 진하게 2 '중간' 으로 설정.	선택된 농도로 선택 농도를 변경한다. 선택 농도 = 중간	CMS_STC_004_004	Menu input = 3 커피 잔량 = 50g 1 '원두' 선택후 최대용량(100g)초과 하게 80g보충	초과하게 되면 최대용량으로 보충하게 된 다. 커피 잔량 = 100g
CMS_STC_003_002	Menu input = 2 선택농도 = 진하게 3 '연하게' 로 설정.	선택된 농도로 선택 농도를 변경한다. 선택 농도 = 연하게	CMS_STC_004_005	Menu input = 3 가루 커피 유무 = o 2 '커피 가루' 선택 후 1 입력	변경없음.
CMS_STC_003_003	Menu input = 2 선택농도 = 진하게 1,2,3 이외의 값	변경없음.	CMS_STC_004_006	Menu input = 3 물 = 500ml 3 '물' 선택후 최대용량(1000ml)초 과하게 800ml보충	초과하게 되면 최대용량으로 보충하게 된 다. 물 = 1000ml
CMS_STC_004_000	Menu input = 3 커피 잔량 = 50g 1 '원두' 선택후 정상범위(100g이내) 로 30g보충	정상 범위 내의 보충은 정상적으로 작동 한다. 커피 잔량 = 80g			

System Test List

CMS_STC_004_007	Menu input = 3 물 = 500ml 커피 잔량 = 50g 가루 커피 유무 = o 1 '원두' 선택후 음수 -10입력.	변경 없음.
CMS_STC_004_008	Menu input = 3 물 = 500ml 커피 잔량 = 50g 가루 커피 유무 = x 2 '커피 가루' 선택 후 1이외의 값 입력.	변경 없음.
CMS_STC_004_009	Menu input = 3 물 = 500ml 커피 잔량 = 50g 가루 커피 유무 = o 3 '물' 선택후 음수 -100 입력.	변경 없음.

CMS_STC_005_000	Menu input = 4 찌꺼기 유무 = o	찌꺼기를 제거한다. 찌꺼기 유무 = x
CMS_STC_005_001	Menu input = 4 찌꺼기 유무 = x	변경없음.

CMS_STC_006_000	Menu Input = 5 농도 = 진하게 온도 = 냉 물 = 1000ml 커피 잔량 = 100g 가루 커피 유무 = o 찌꺼기 유무 = x 1 '커피 추출'선택 후 2분뒤로 설정.	대기하면서 현재 시간이 변경되면 변경된 정보를 띄워주면서 예약을 기다린다. 시 간이 되면 예약해둔 대로 작동한다. 물 = 900ml 가루 커피 유무 = x 찌꺼기 유무 = o
CMS_STC_006_001	Menu Input = 5 물 = 1000ml 2 '청소 예약'선택 후 2분뒤로 설정.	대기하면서 현재 시간이 변경되면 변경된 정보를 띄워주면서 예약을 기다린다. 시 간이 되면 예약해둔 대로 작동한다. 물 = 500ml
CMS_STC_006_002	Menu Input = 5 농도 = 진하게 온도 = 냉 물 = 1000ml 커피 잔량 = 100g 가루 커피 유무 = o 찌꺼기 유무 = x 1 '커피 추출'선택 후 현재 시간으로 설정.	바로 작동한다. 물 = 900ml 가루 커피 유무 = x 찌꺼기 유무 = o
CMS_STC_006_003	Menu Input = 5 물 = 1000ml 2 '청소 예약'선택 후 현재 시간으로 설정.	바로 작동한다. 물 = 500ml

System Test List

CMS_STC_006_004	Menu Input = 5 농도 = 진하게 온도 = 냉 물 = 1000ml 커피 잔량 = 100g 가루 커피 유무 = o 찌꺼기 유무 = x 1 '커피 추출'선택 후 지난 시간으로 설정.	24시간 표기법을 사용하기 때문에 다음날 작동한다. 물 = 900ml 가루 커피 유무 = x 찌꺼기 유무 = o	CMS_STC_006_008	Menu Input = 5 농도 = 진하게 온도 = 냉 물 = 0ml 커피 잔량 = 0g 가루 커피 유무 = x 찌꺼기 유무 = x 1 '커피 추출'선택 후 2분뒤로 설정.	예약을 기다리다 작동할 때 오류 메시지를 출력한다. 물 부족 (잔량 0ml) 빽!
CMS_STC_006_005	Menu Input = 5 물 = 1000ml 2 '청소 예약'선택 후 지난 시간으로 설정.	24시간 표기법을 사용하기 때문에 다음날 작동한다. 물 = 500ml	CMS_STC_006_009	Menu Input = 5 물 = 0ml 2 '청소 예약'선택 후 2분뒤로 설정.	예약을 기다리다 작동할 때 오류 메시지를 출력한다. 물 부족 (잔량 0ml) 빽!
CMS_STC_006_006	Menu Input = 5 농도 = 진하게 온도 = 냉 물 = 1000ml 커피 잔량 = 100g 가루 커피 유무 = o 찌꺼기 유무 = x 1 '커피 추출'선택 후 24시간 표기법을 지 키지 않는 174시 80분 입력.	계속 예약중 상태이며 끝나지 않는다.	CMS_STC_007_000	Menu Input = 6 온도 = 냉	토글 입력 이므로 반대되는 값으로 변경 된다. 온도 = 온
CMS_STC_006_007	Menu Input = 5 물 = 1000ml 2 '청소 예약'선택 후 24시간 표기법을 지 키지 않는 174시 80분 입력.	계속 예약중 상태이며 끝나지 않는다.	CMS_STC_007_001	Menu Input = 6 온도 = 온	토글 입력 이므로 반대되는 값으로 변경 된다. 온도 = 냉
			CMS_STC_008_000	Menu Input = 9 물 = 0ml 커피 잔량 = 0g 가루 커피 유무 = x 찌꺼기 유무 = o	종료후 재실행시키면 종료하기 직전의 값 그대로 불러온다. (기계 초기값 물 = 1000ml 커피 잔량 = 100g 가루 커피 유무 = o 찌꺼기 유무 = x)

System Test Result

ALL PASS!!!

Team 1

QnA?



Team 1

Thanks