



- LEGO
- ГЕГО

201011331 박세동
201360120 이정현
200911414 이준영

목차 - 1

1. Statement of Purpose
2. System Context Diagram
3. Event List
4. DFD
5. Process Specification
6. Data Dictionary

STATEMENT OF PURPOSE

버튼에 의한 동작을 처리하고 처리결과를 화면, 스피커 등으로 출력

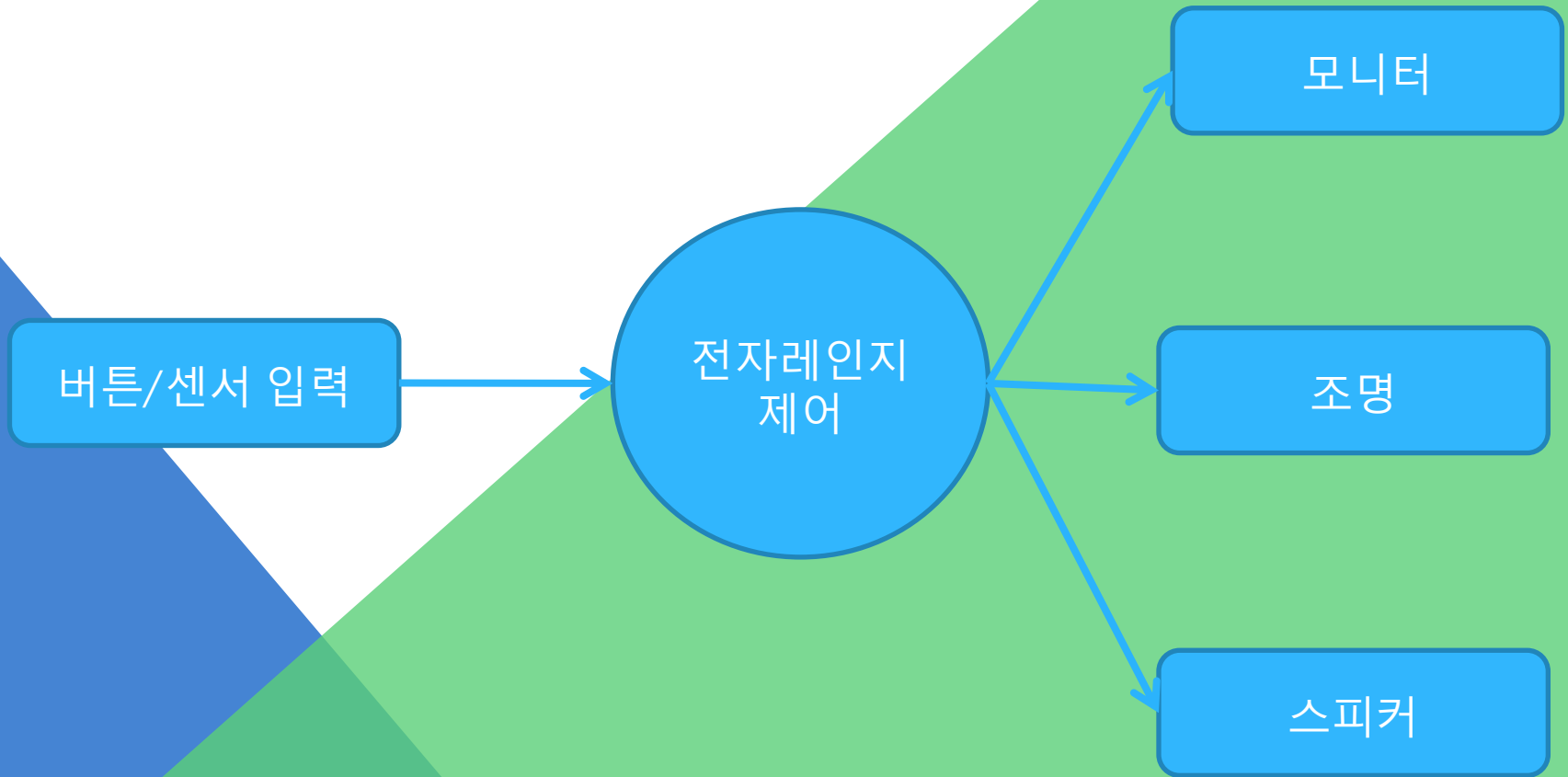
Display에 시간, 온도, 메뉴를 출력

버튼 입력으로 시간과 온도를 설정하고 기기를 동작

동작 시간이 0이 되면 3초간 beep음 발생

문이 열렸을 때와 조리 중에는 backlight가 발생

SYSTEM CONTEXT DIAGRAM



EVENT LIST

입력	설명
A_BUTTON_INPUT	A버튼 클릭 시 마다 10초/10도 증가
B_BUTTON_INPUT	B버튼 클릭 시 마다 30초/20도 증가
C_BUTTON_INPUT	C버튼 클릭 시 온도/시간 설정
D_BUTTON_INPUT	D버튼 클릭 시 모드 선택
E_BUTTON_INPUT	E버튼 클릭 시 실행/취소
F_SENSOR_INPUT	문 열림 감지 센서 입력 시 문 개폐에 따라 작동한다.
G_SENSOR_INPUT	온도 감지 센서

EVENT LIST

출력	설명
Display	현재온도, 설정온도 / 남은시간, 설정시간 표시 현재 조리모드를 보여줌 (00: Manual 모드)
Light	Back Light 문을 열거나 조리 중에 불이 꺼짐 끝나면 꺼짐
Speaker	Beep speaker 조리 완료 후 beep음 3초간 울림

SYSTEM CONTEXT DIAGRAM

-EVENT LIST-

A BUTTON INPUT
B BUTTON INPUT
C BUTTON INPUT
D BUTTON INPUT
E BUTTON INPUT
F SENSOR INPUT
G SENSOR INPUT

버튼/센서 입력

전자레인지
제어

Display

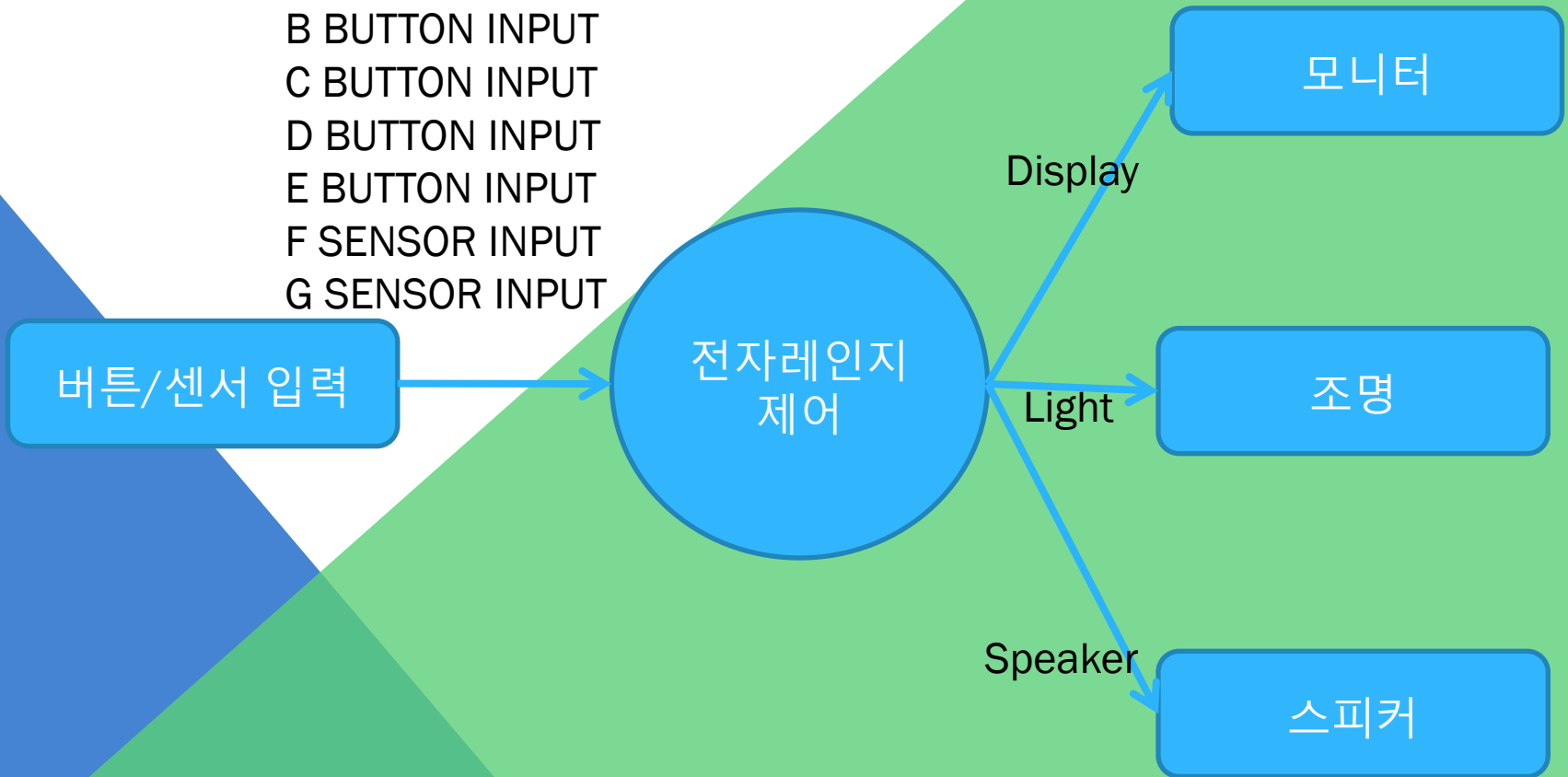
모니터

Light

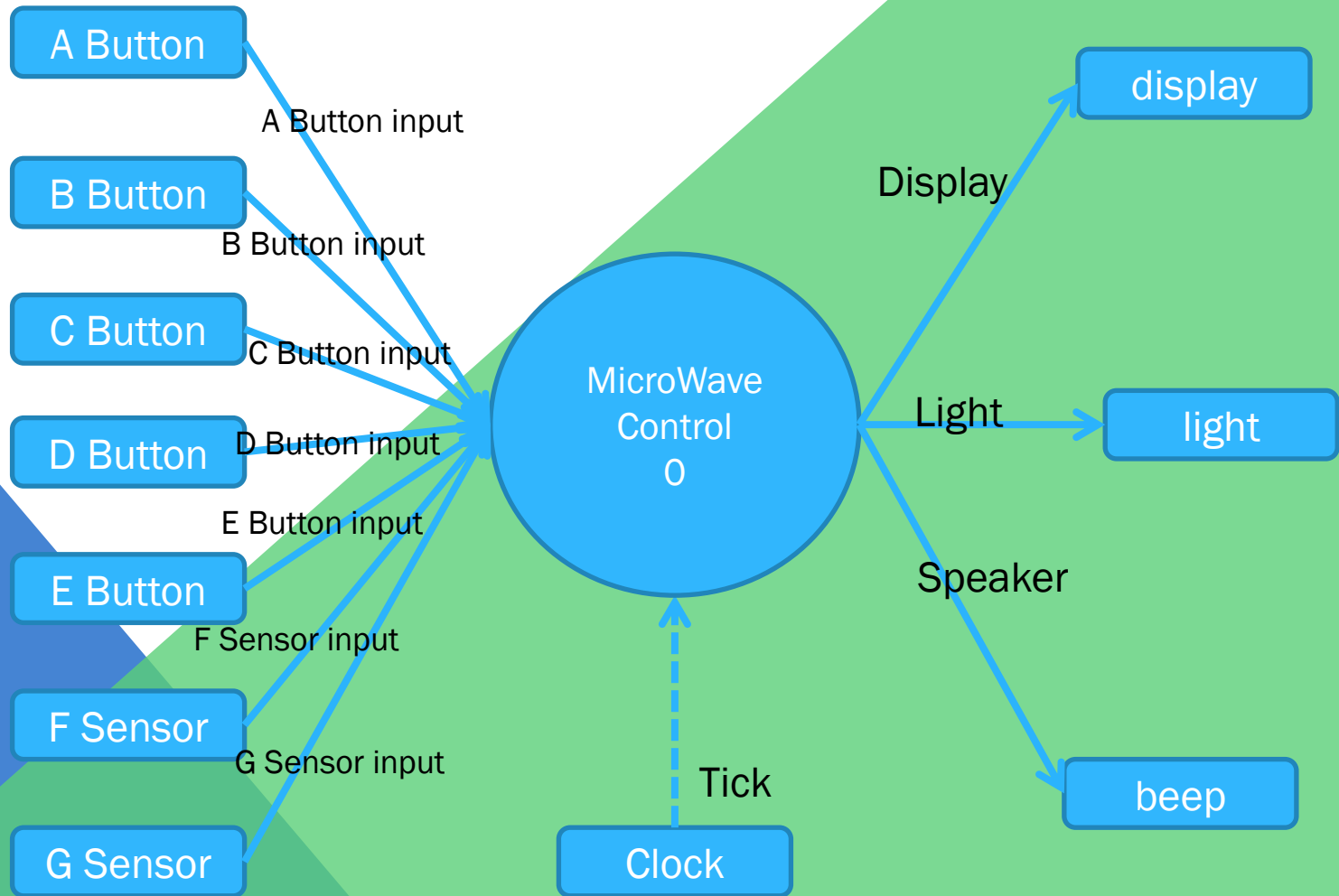
조명

Speaker

스피커



DFD LEVEL 0

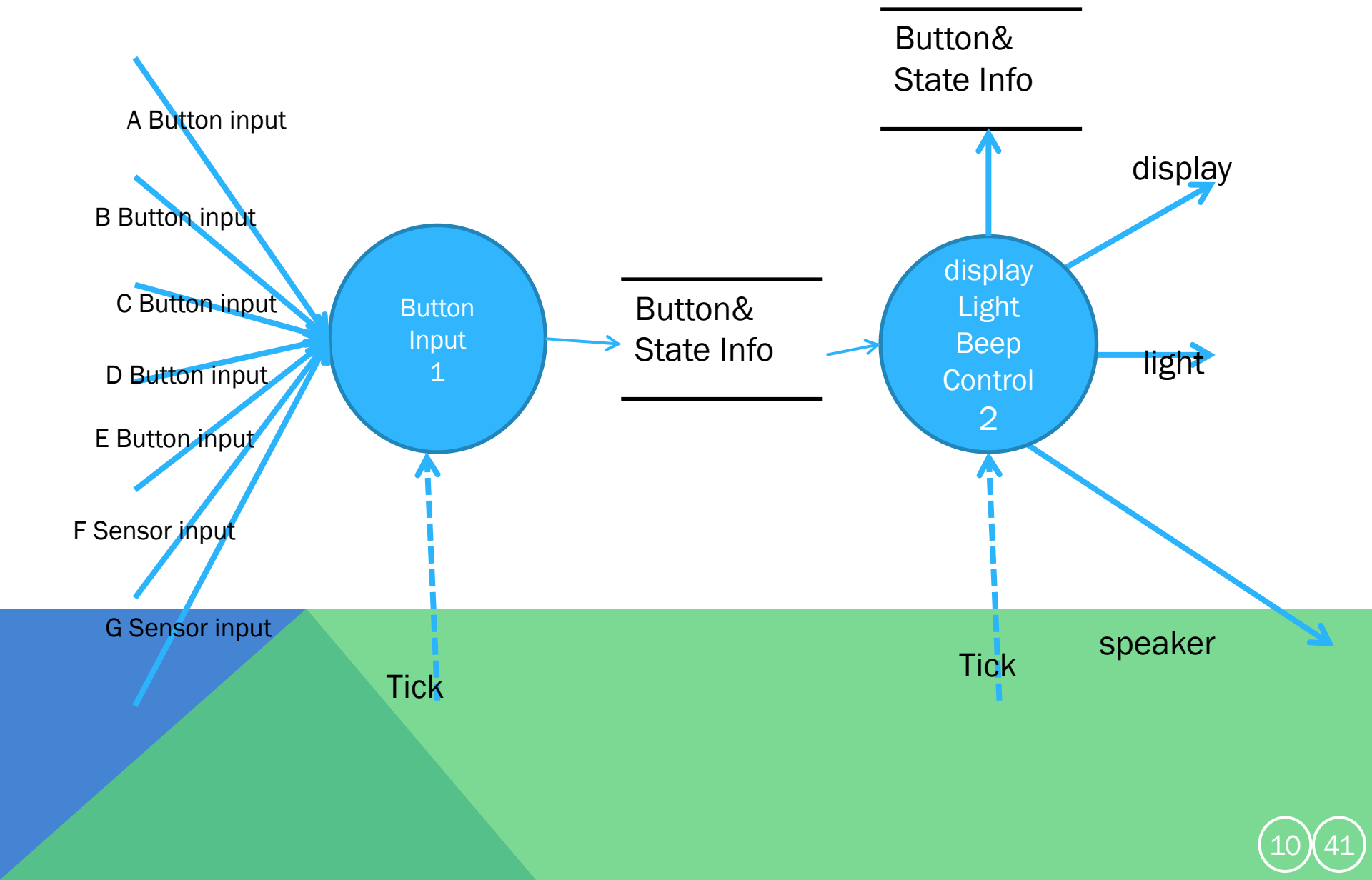


DFD LEVEL 0

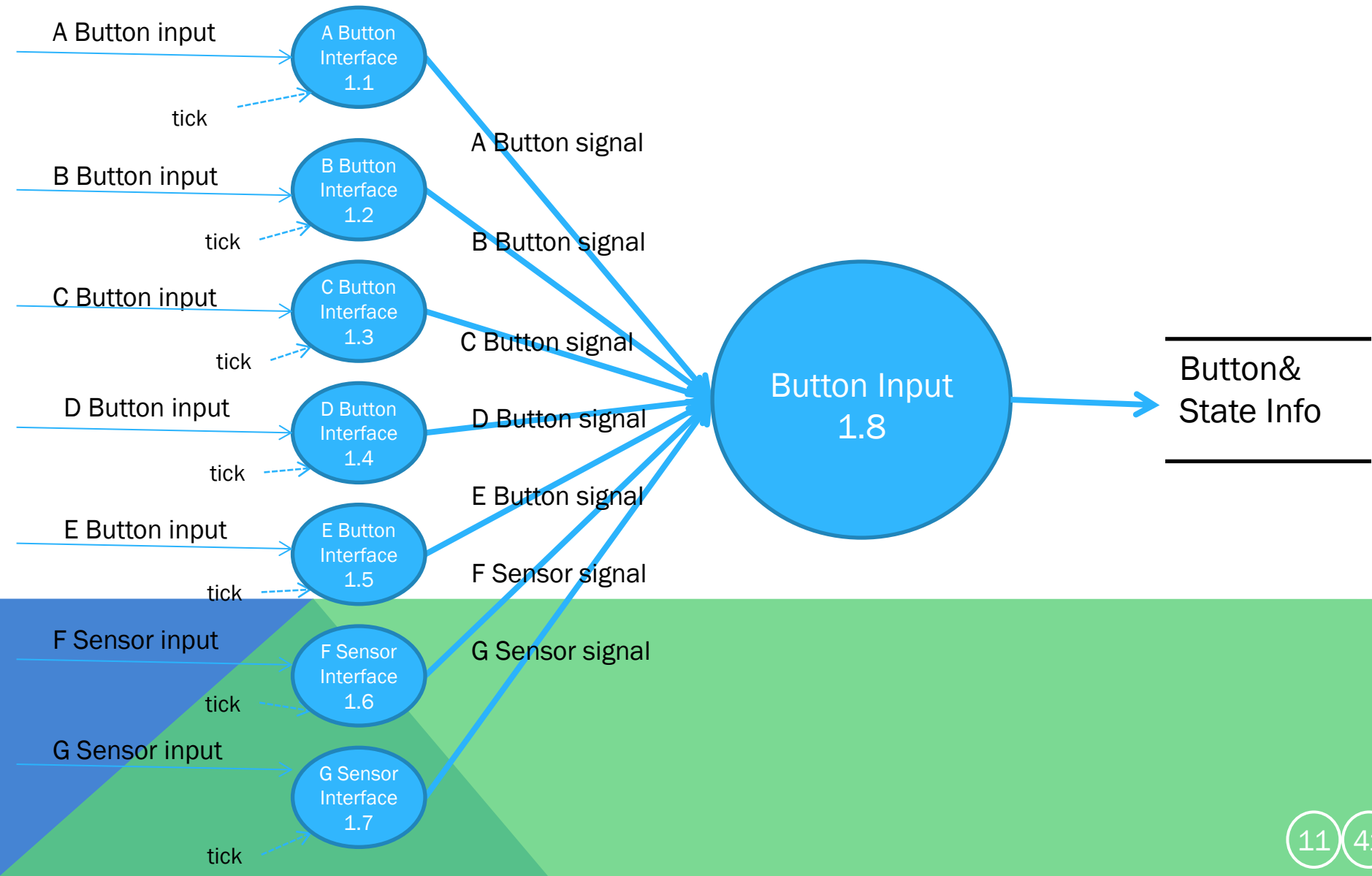
DATA DICTIONARY

입력	설명	입력값 / 타입
A_BUTTON_INPUT	A버튼 클릭 시 마다 10초/10도 증가	BOOL/ INTERRUPT
B_BUTTON_INPUT	B버튼 클릭 시 마다 30초/20도 증가	BOOL/ INTERRUPT
C_BUTTON_INPUT	C버튼 클릭 시 온도/시간 설정	BOOL/ INTERRUPT
D_BUTTON_INPUT	D버튼 클릭 시 모드 선택	BOOL/ INTERRUPT
E_BUTTON_INPUT	E버튼 클릭 시 실행/취소	BOOL/ INTERRUPT
F_SENSOR_INPUT	문 열림 감지 센서	BOOL/ INTERRUPT
G_SENSOR_INPUT	온도 감지 센서	INT/periodic

DFD LEVEL 1



DFD LEVEL 2



PROCESS SPECIFICATION

Reference No.	1.1
name	A Button input
Input	A Button input(int)
output	A Button signal(Boolean)
Process Description	A Button input의 아날로그 값을 받아드려, True/False와 같은 디지털신호로 변환 후 A Button signal로 할당한다. 그 후 Button input & Temp Sense 프로세스로 보내준다.

Reference No.	1.2
name	B Button input
Input	B Button input(int)
output	B Button signal(Boolean)
Process Description	B Button input의 아날로그 값을 받아드려, True/False와 같은 디지털신호로 변환 후 B Button signal로 할당한다. 그 후 Button input & Temp Sense 프로세스로 보내준다.

PROCESS SPECIFICATION

Reference No.	1.3
name	C Button input
Input	C Button input(int)
output	C Button signal(Boolean)
Process Description	C Button input의 아날로그 값을 받아드려, True/False와 같은 디지털신호로 변환 후 C Button signal로 할당한다. 그 후 Button input & Temp Sense 프로세스로 보내준다.

Reference No.	1.4
name	D Button input
Input	D Button input(char)
output	D Button signal(Boolean)
Process Description	D Button input의 아날로그 값을 받아드려, True/False와 같은 디지털신호로 변환 후 D Button signal로 할당한다. 그 후 Button input & Temp Sense 프로세스로 보내준다.

PROCESS SPECIFICATION

Reference No.	1.5
name	E Button input
Input	E Button input(Boolean)
output	E Button signal(Boolean)
Process Description	E Button input의 아날로그 값을 받아드려, True/False와 같은 디지털신호로 변환 후 E Button signal로 할당한다. 그 후 Button input & Temp Sense 프로세스로 보내준다.

Reference No.	1.6
name	F Sensor input
Input	F Sensor input(Boolean)
output	F Sensor signal(Boolean)
Process Description	F sensor의 아날로그 값을 받아드려, True/False와 같은 디지털신호로 변환 후 F sensor signal로 할당한다. 그 후 Button input & Temp Sense 프로세스로 보내준다.

PROCESS SPECIFICATION

Reference No.	1.7
name	G Sensor input
Input	G Sensor input(int)
output	G Sensor signal(Bool)
Process Description	G sensor의 아날로그 값을 받아드려, True/False와 같은 디지털신호로 변환 후 G sensor signal로 할당한다. 그 후 Button input & Temp Sense 프로세스로 보내준다.

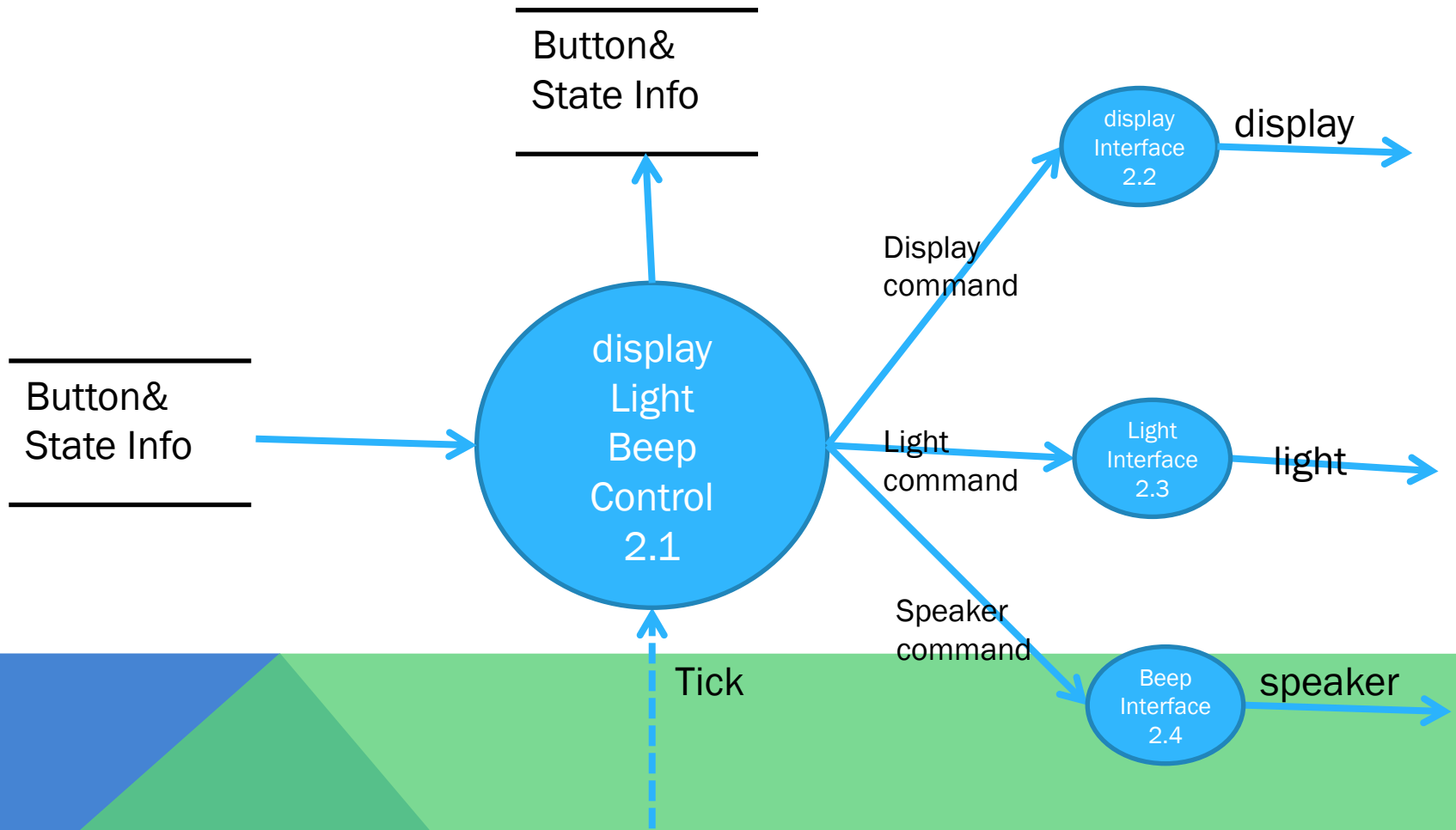
Reference No.	1.8
name	Button input & Temp Sense
Input	A~E Button input G,F Sensor input
output	
Process Description	Button signal 혹은 Sensor signal이 data process 로 오면 Button state data store로 전달한다.

DFD LEVEL 2

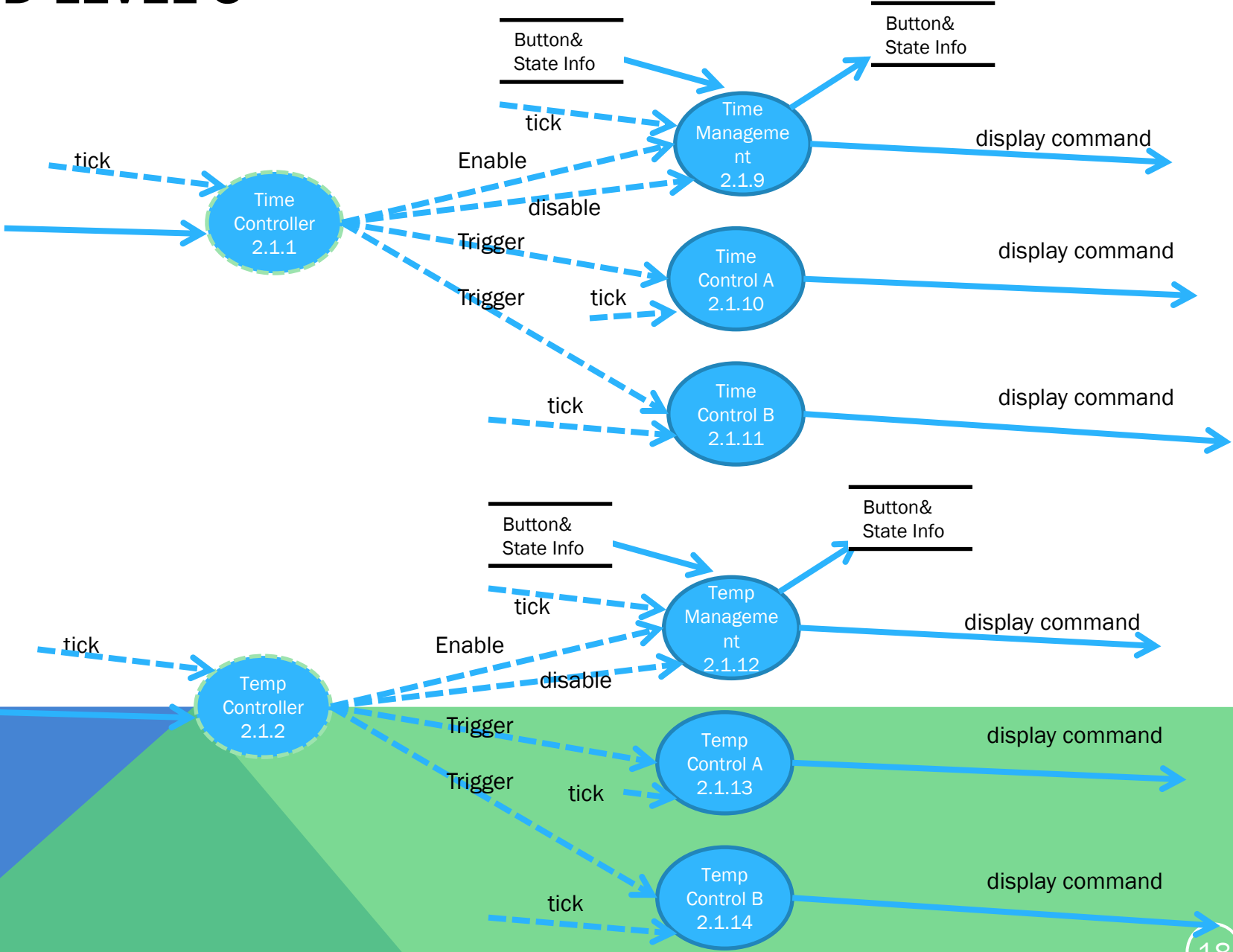
DATA DICTIONARY

입력	설명	타입
A_BUTTON_SIGNAL	A버튼 클릭 유무	BOOL
B_BUTTON_SIGNAL	B버튼 클릭 시 유무	BOOL
C_BUTTON_SIGNAL	C버튼 클릭 시 유무	BOOL
D_BUTTON_SIGNAL	D버튼 클릭 시 유무	BOOL
E_BUTTON_SIGNAL	E버튼 클릭 시 유무	BOOL
F_SENSOR_SIGNAL	F버튼 클릭 시 유무	BOOL
G_SENSOR_SIGNAL	온도 변화 유무	BOOL

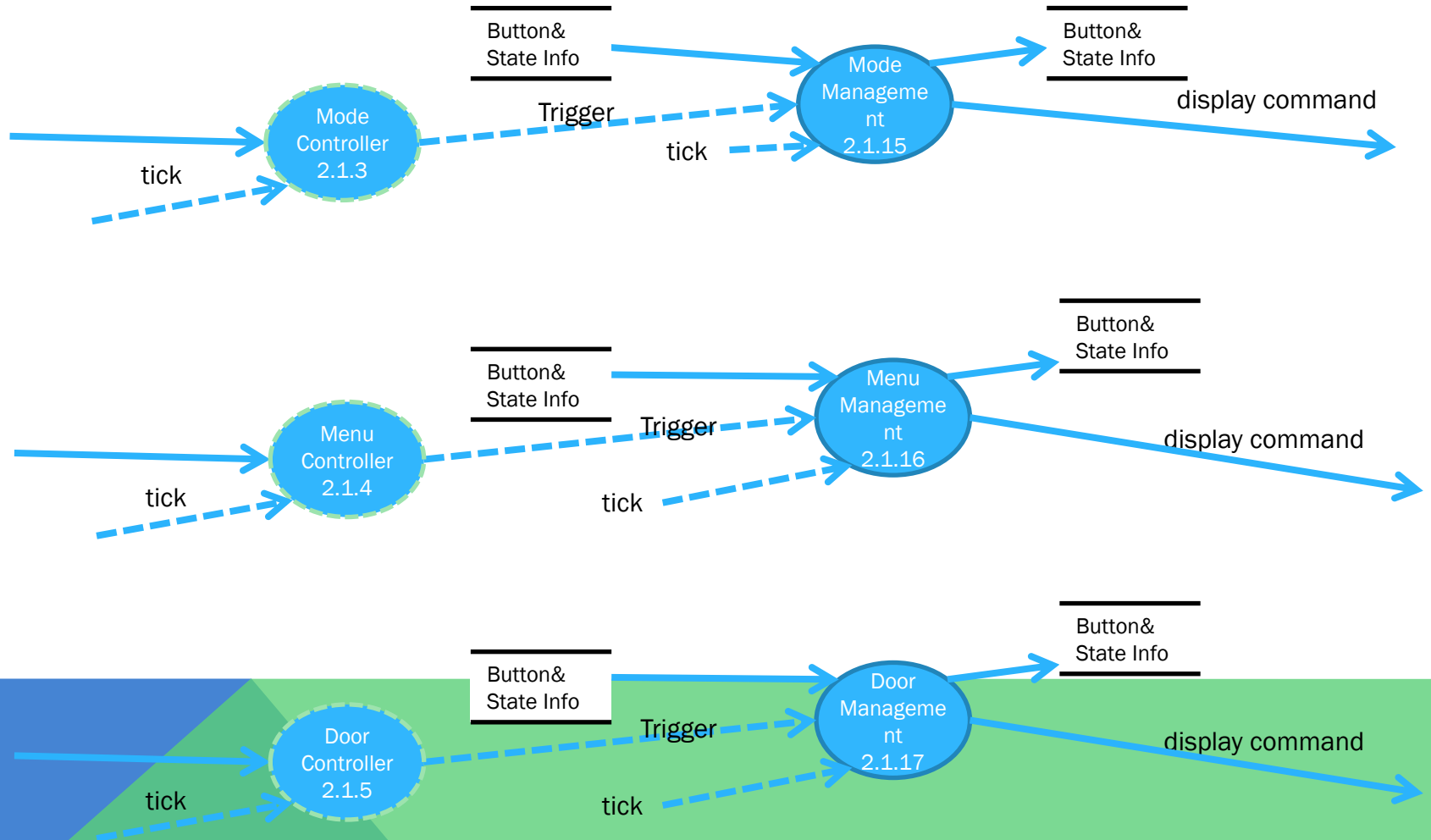
DFD LEVEL2



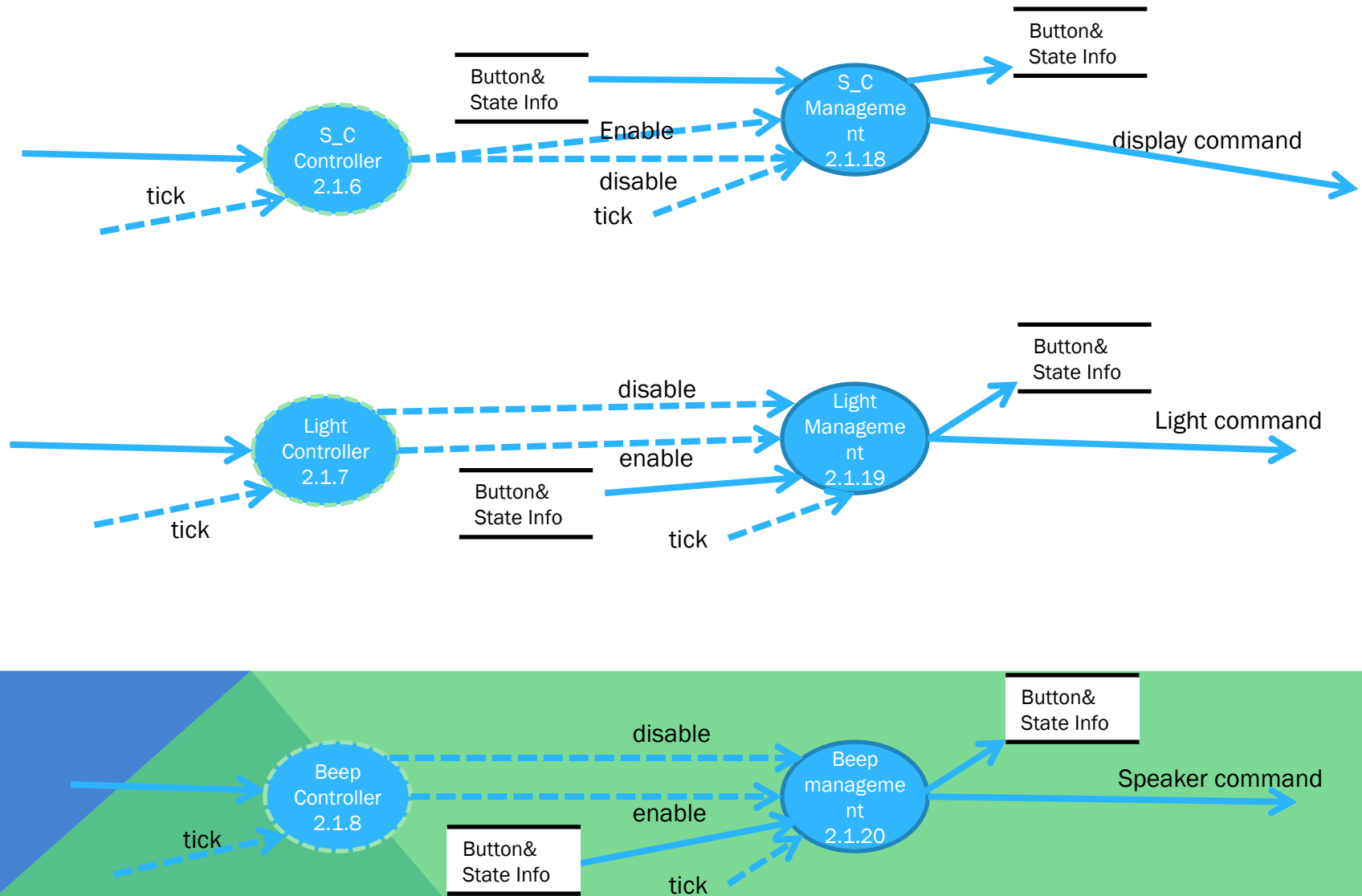
DFD LEVEL 3



DFD LEVEL 3

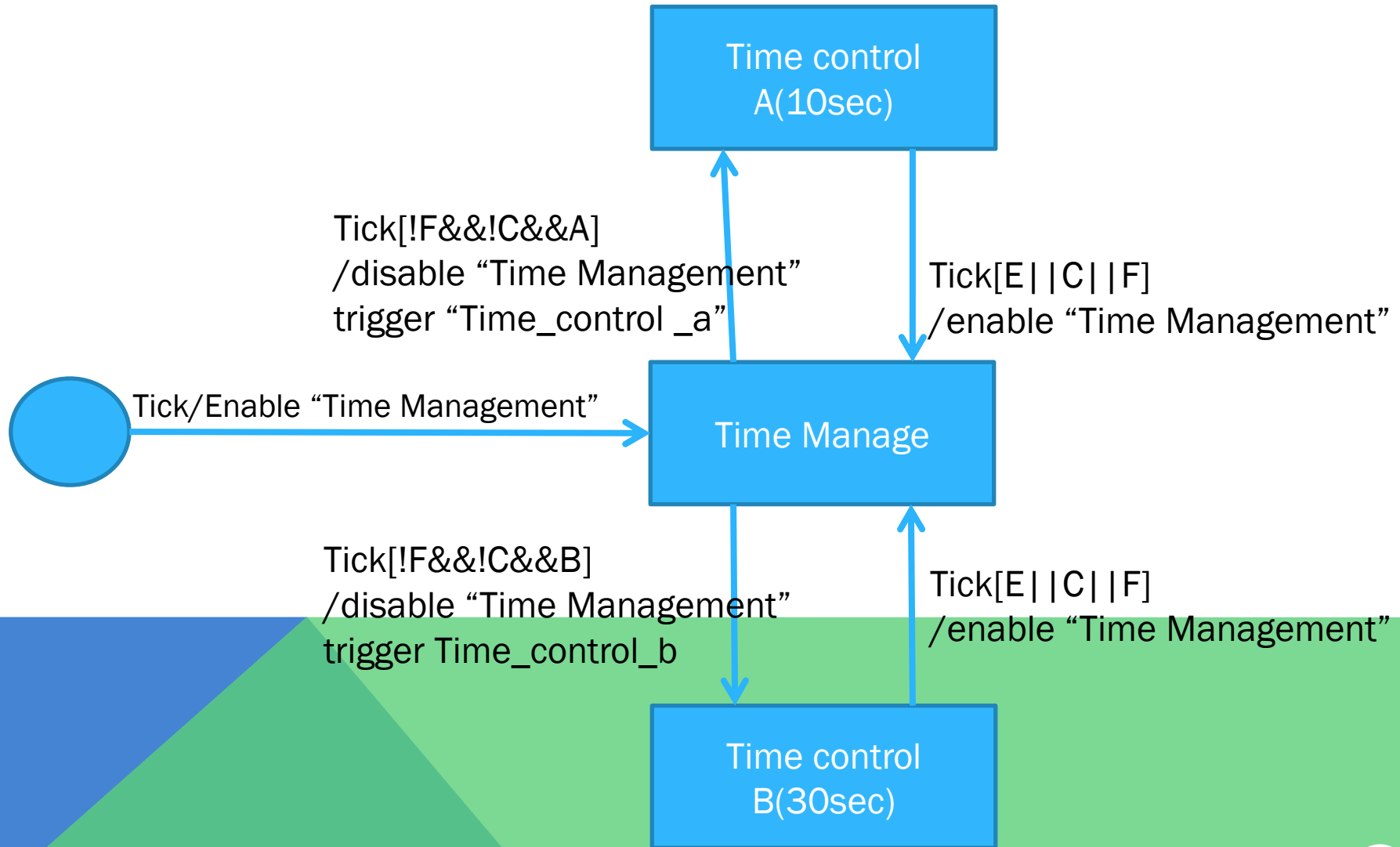


DFD LEVEL 3



DFD LEVEL 4

STATE MACHINE FOR TIME CONTROLLER 2.1.1



PROCESS SPECIFICATION

Reference No.	2.1.1
name	Time Controller
Input	Tick (int)
output	Trigger (int)
Process Description	tick에 따라 시간을 제어해준다. Trigger를 Time management에 보내준다.
Reference No.	2.1.9
name	Time management
Input	Tick(int), enable, disable (bool)
output	display command (char)
Process Description	Enable일 경우 Time Management를 실행시키고 Disable일 경우 다른 trigger를 작동시킨다.

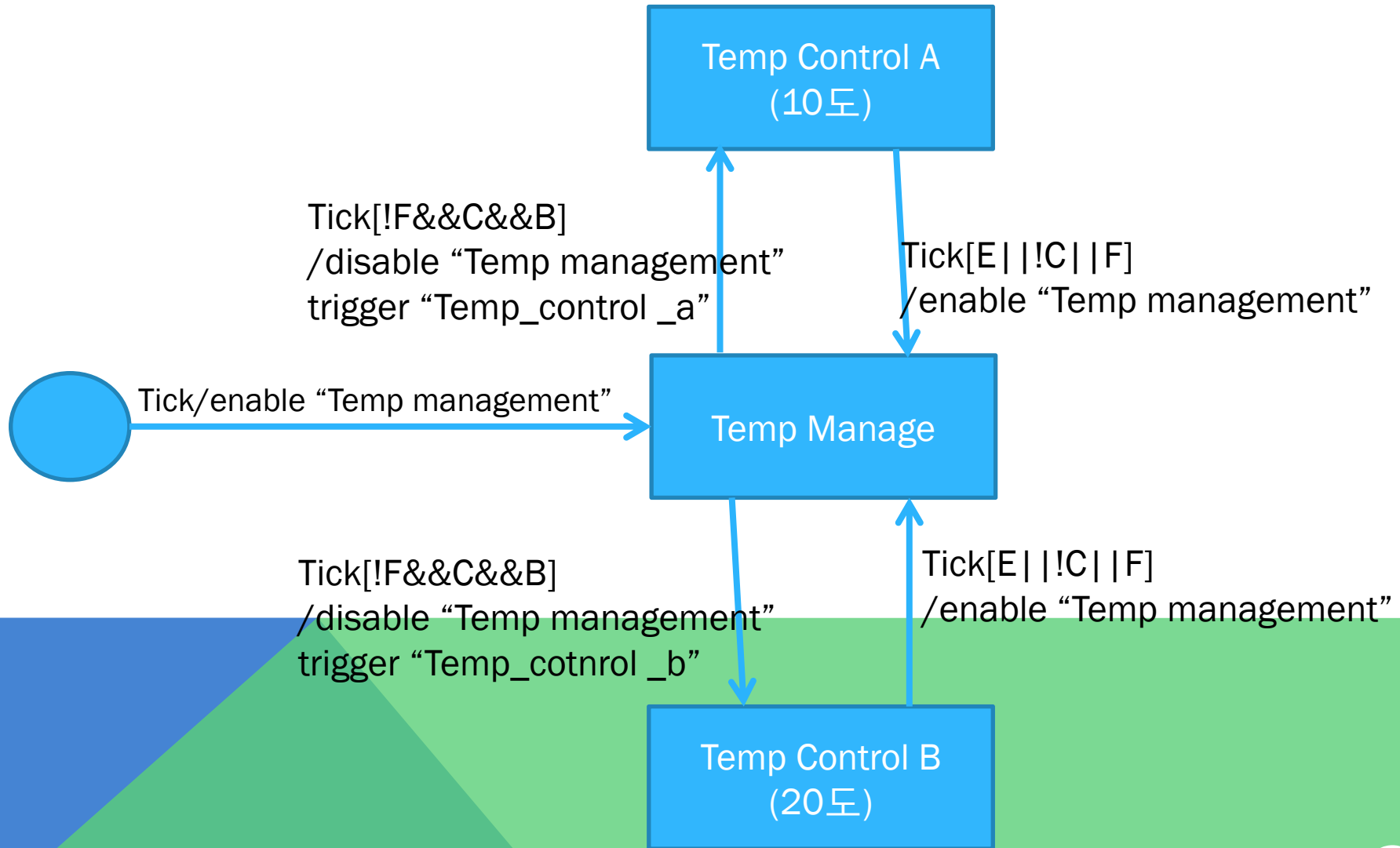
PROCESS SPECIFICATION

Reference No.	2.1.11
name	Time Control B
Input	Tick, trigger (int)
output	display command (char)
Process Description	Time Controller로부터 Trigger를 받아 조건에 맞춰 변경된 시간 값을 저장하고 변경된 시간 값을 출력하기 위해 Display Command에 정보를 보내준다.

Reference No.	2.1.10
name	Time Control A
Input	Trigger, Tick (int)
Output	display command (char)
Process Description	Time Controller로부터 Trigger를 받아 조건에 맞춰 변경된 시간 값을 저장하고 변경된 시간 값을 출력하기 위해 Display Command에 정보를 보내준다.

DFD LEVEL 4

STATE MACHINE FOR TEMP CONTROLLER 2.1.2



PROCESS SPECIFICATION

Reference No.	2.1.2
name	Temp Controller
Input	Tick (int)
output	Trigger (int)
Process Description	tick에 따라 온도 메뉴를 제어해준다. Trigger를 Temp management에 보내준다.

Reference No.	2.1.12
name	Temp management
Input	Enable, disable (bool), tick (int)
output	Display Command (char)
Process Description	Enable일 경우 Temp Management를 실행시키고 Disable일 경우 다른 trigger를 작동시킨다.

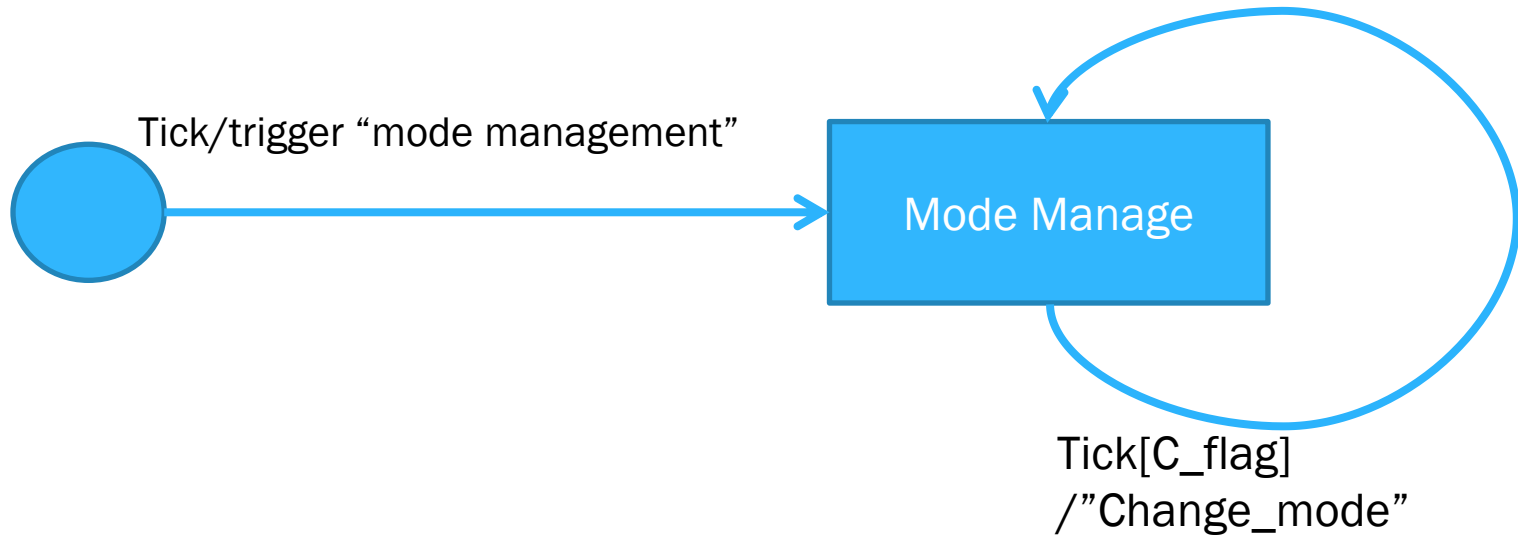
PROCESS SPECIFICATION

Reference No.	2.1.13
name	Temp Control A
Input	Tick, trigger (int)
output	display command (char)
Process Description	Temp Controller로부터 Trigger를 받아 조건에 맞춰 변경된 온도 값을 저장하고 변경된 온도 값을 출력하기 위해 Display Command에 정보를 보내준다.

Reference No.	2.1.14
name	Temp Control B
Input	Trigger, Tick (int)
Output	Display Command (char)
Process Description	Temp Controller로부터 Trigger를 받아 조건에 맞춰 변경된 온도 값을 저장하고 변경된 온도 값을 출력하기 위해 Display Command에 정보를 보내준다.

DFD LEVEL 4

STATE MACHINE FOR MODE CONTROLLER 2.1.3



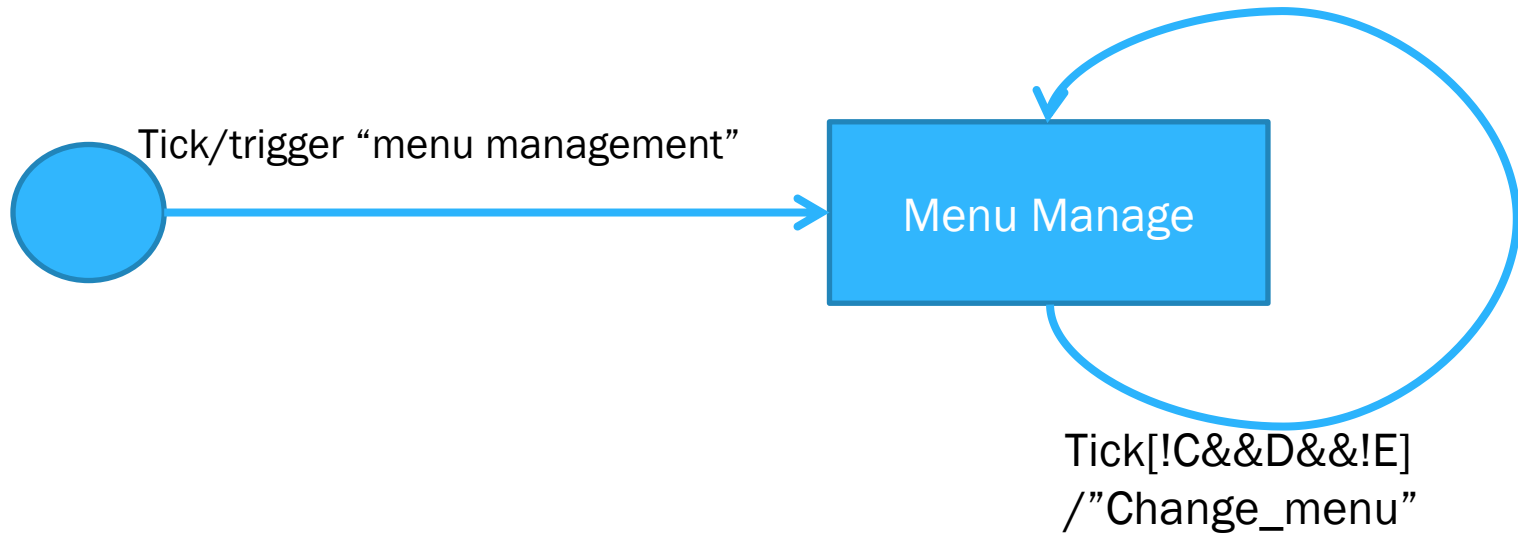
PROCESS SPECIFICATION

Reference No.	2.1.3
name	Mode Controller
Input	Tick (int)
output	Trigger (int)
Process Description	Tick에 따라 모드를 제어해준다. Trigger를 mode management로 보내준다.

Reference No.	2.1.15
name	Mode management
Input	Trigger, Tick (int)
output	Display Command (char)
Process Description	Mode Controller로부터 Trigger를 받아 조건에 맞춰 변경된 시간/온도 선택지를 저장하고 변경된 시간/온도 선택지를 출력하기 위해 Display Command에 정보를 보내준다.

DFD LEVEL 4

STATE MACHINE FOR MENU CONTROLLER 2.1.4



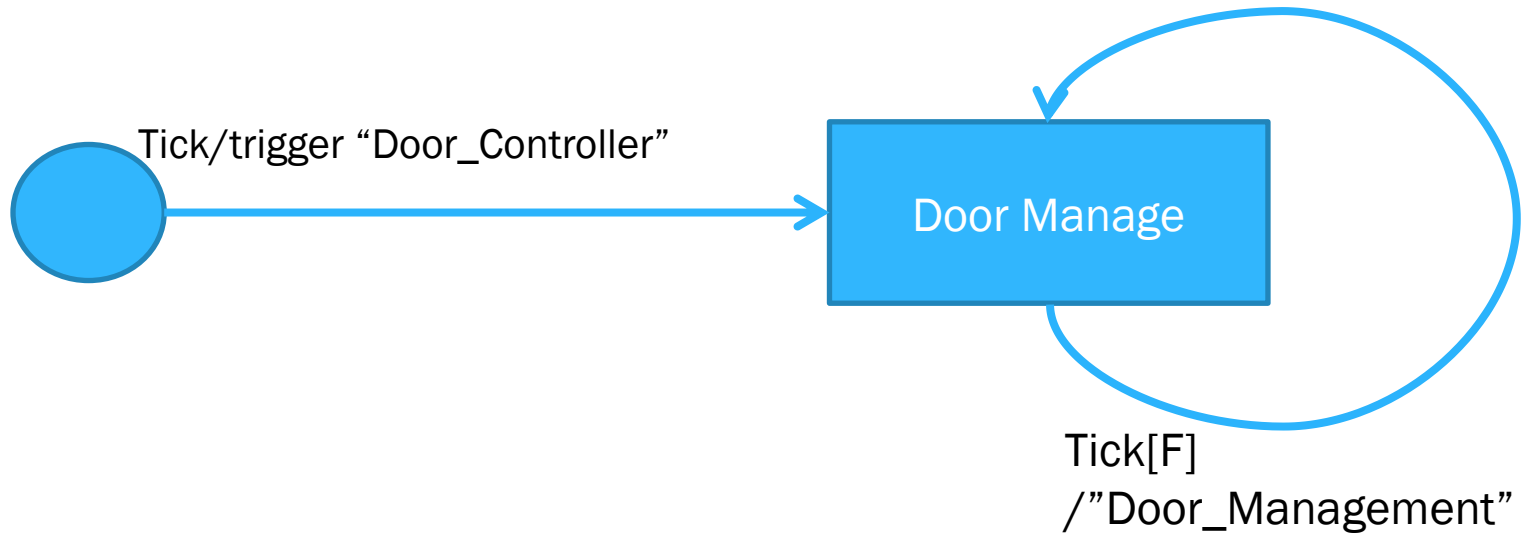
PROCESS SPECIFICATION

Reference No.	2.1.4
name	Menu Controller
Input	Tick (int)
output	Trigger (int)
Process Description	Tick에 따라 메뉴를 제어해준다. Trigger를 Menu management로 보내준다.

Reference No.	2.1.16
name	Menu management
Input	Trigger, Tick (int)
output	Display Command (char)
Process Description	Menu Controller로부터 Trigger를 받아 조건에 맞춰 변경된 메뉴를 저장하고 변경된 메뉴를 출력하기 위해 Display Command에 정보를 보내준다.

DFD LEVEL 4

STATE MACHINE FOR DOOR CONTROLLER 2.1.5



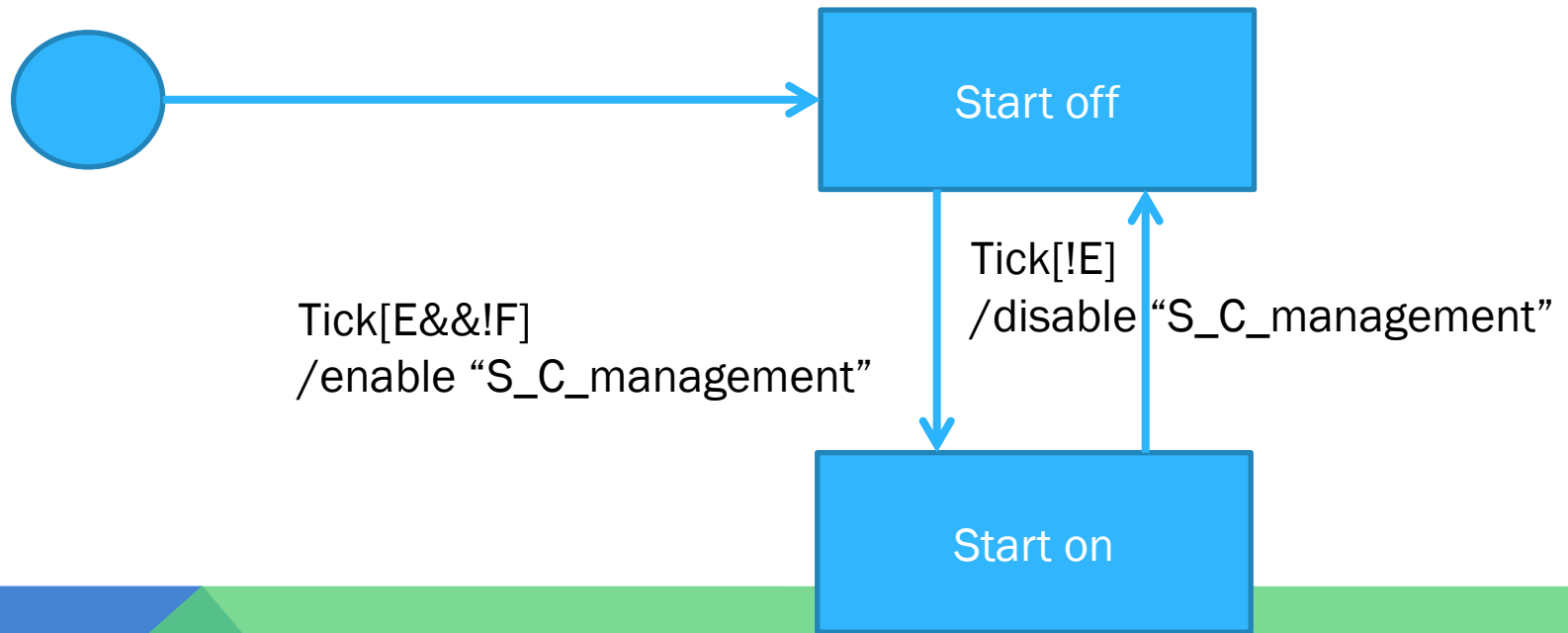
PROCESS SPECIFICATION

Reference No.	2.1.5
name	Door Controller
Input	Tick (int)
output	Trigger(Boolean)
Process Description	Tick에 따라 문 개폐를 제어해준다. Trigger를 Door management로 보내준다.

Reference No.	2.1.17
name	Door management
Input	Trigger (boolean), tick (int)
output	Display command (char)
Process Description	enable상태의 조건이 맞으면 전차레인을 가동한다. disable상태의 조건이 맞으면 전차레인지 가동을 중지한다.

DFD LEVEL 4

STATE MACHINE FOR S_C CONTROLLER 2.1.5



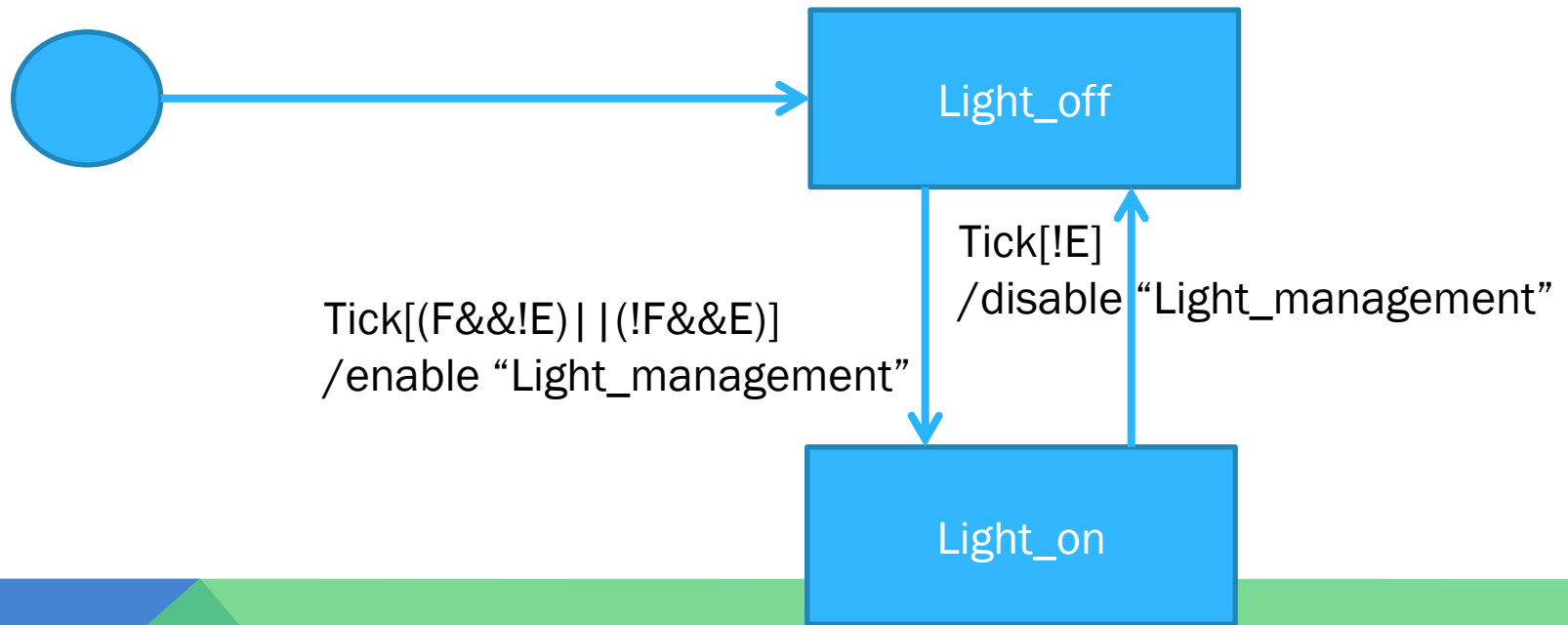
PROCESS SPECIFICATION

Reference No.	2.1.6
name	S_C Controller
Input	Tick (int)
output	Disable, enable (boolean)
Process Description	Tick에 따라 전자레인지 가동을 제어해준다. disable, enable를 S_C management로 보내준다.

Reference No.	2.1.18
name	S_C management
Input	enable, disable (boolean), tick (int)
output	Display command (char)
Process Description	enable상태의 조건이 맞으면 전자레인지를 가동한다. disable상태의 조건이 맞으면 전자레인지 가동을 중지한다.

DFD LEVEL 4

STATE MACHINE FOR LIGHT CONTROLLER 2.1.6



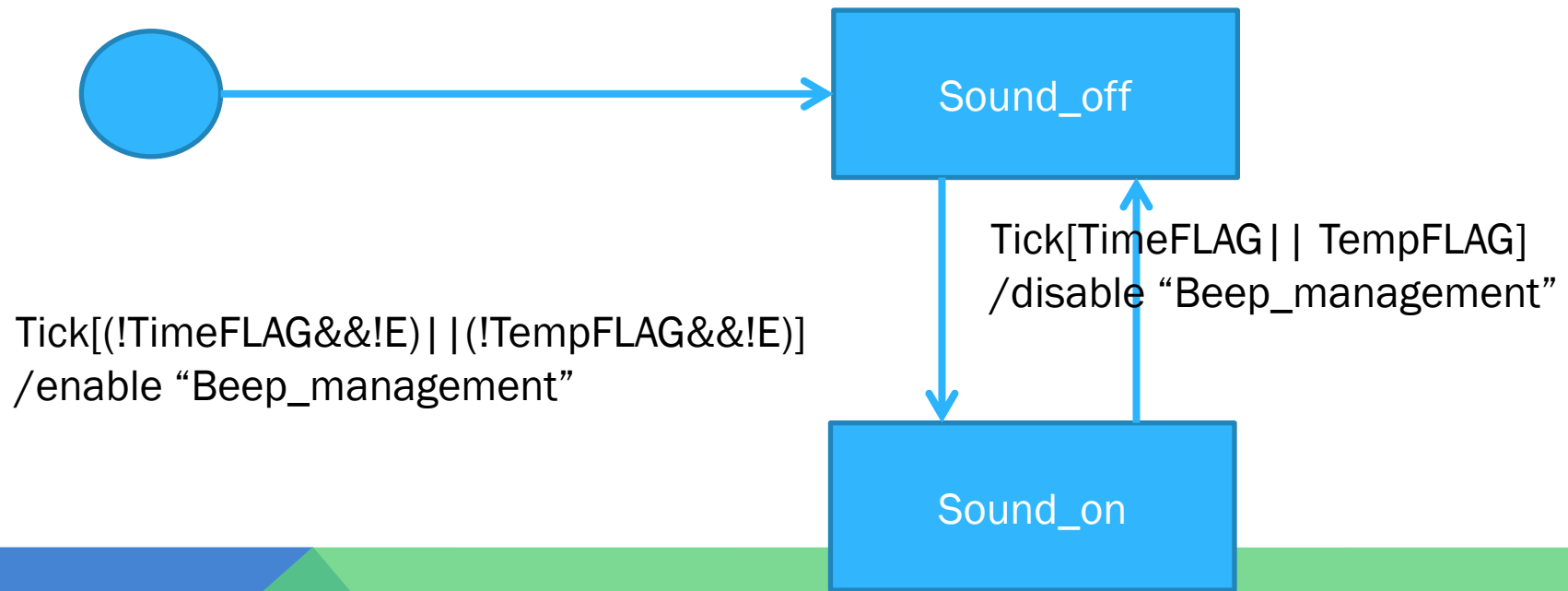
PROCESS SPECIFICATION

Reference No.	2.1.7
name	Light Controller
Input	Tick (int)
output	Disable, enable (bool)
Process Description	Tick에 따라 조명을 제어해준다. Disable, enable를 light management로 보내준다.

Reference No.	2.1.19
name	Light Management
Input	Tick(int), enable, disable(bool)
output	Light command (bool)
Process Description	Enable, disable에 따라 light_off에 light command를 보낸다.

DFD LEVEL 4

STATE MACHINE FOR BEEP CONTROLLER 2.1.7



PROCESS SPECIFICATION

Reference No.	2.1.8
name	Beep Controller
Input	Tick (int)
output	Disable, enable (bool)
Process Description	Tick에 따라 소리를 제어해준다. Disable, enable를 beep management로 보내준다.

Reference No.	2.1.20
name	Beep Management
Input	Tick(int), enable, disable (bool)
output	Speaker command (bool)
Process Description	Enable, disable에 따라 sound_off에 speaker command를 보낸다.

DFD LEVEL 4

DATA DICTIONARY

Data name	Description
A	A Button 입력이 됐을 때의 Flag. A면 놀림 리턴될때 !A.
B	B Button 입력이 됐을 때의 Flag. B면 놀림 리턴될때 !B.
C	C Button 입력이 됐을 때의 상태 값. C이면 온도 !C면 시간.
C_Flag	C Button 입력에 대한 Flag. 눌렀는지 안 눌렀는지에 대한 것.
D	D Button 입력에 대한 Flag. D면 눌린 상태 !D면 안 놀림.
E	E Button 입력에 대한 Flag. E Start !E Stop.
F	F Button 입력에 대한 Flag. F 문 열림 !F 문 닫힘.

DFD LEVEL 4

DATA DICTIONARY

Data name	Description
Button&State Info	각각의 정보저장소
Display command	데이터를 display interface에 전달.
Light command	데이터를 light interface에 전달.
Sound command	데이터를 sound interface에 전달.
TimeFlag	전자레인지 작동 중 정지버튼이 눌린 경우 1 전자레인지 작동 중 시작버튼이 눌린 경우 0
TempFlag	전자레인지 작동 중 정지버튼이 눌린 경우 1 전자레인지 작동 중 시작버튼이 눌린 경우 0

목차 - 2

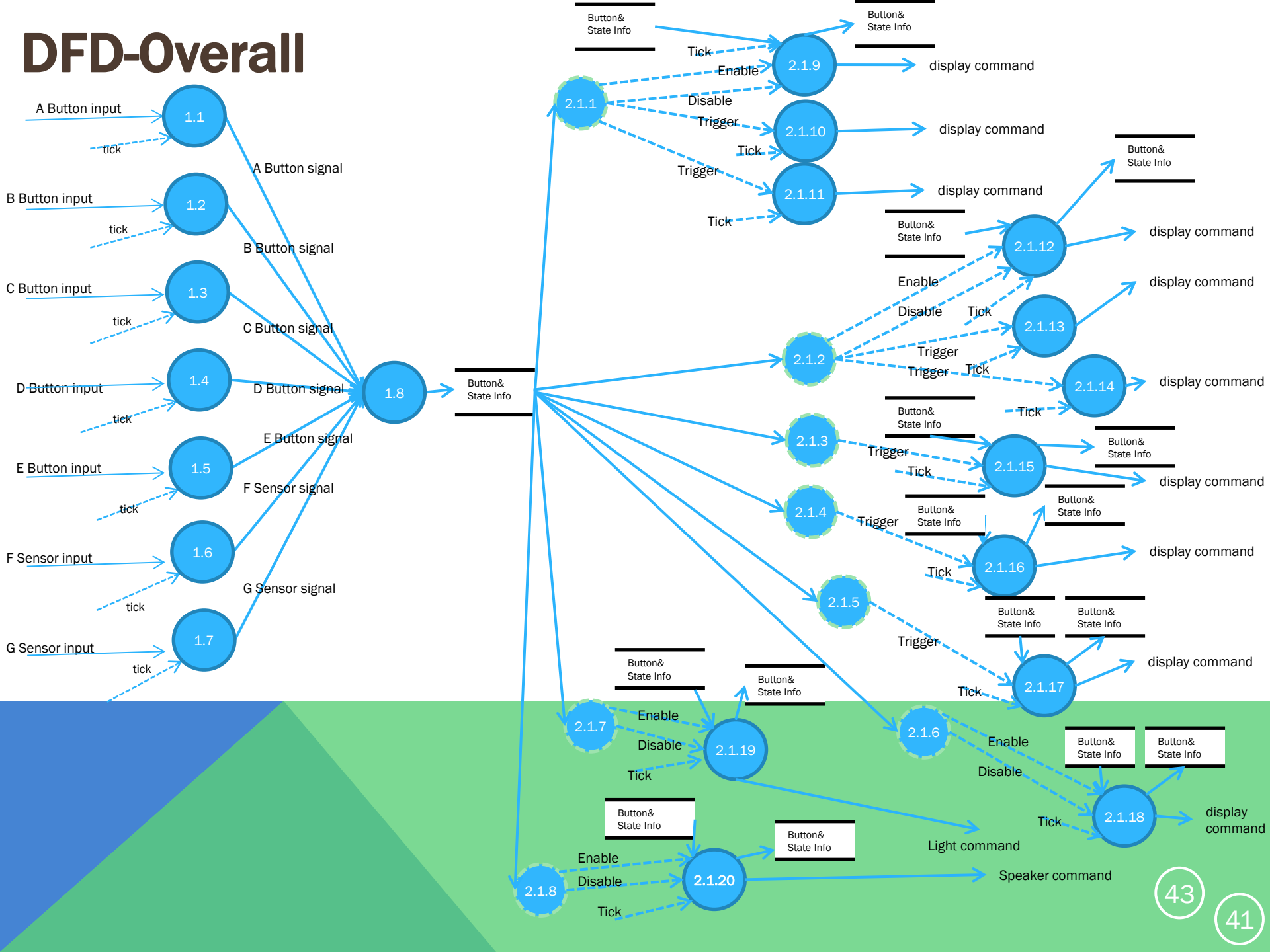
1. Purpose of Structured Chart
2. Structured Chart – Transform Analysis
3. Structured Chart – MOS(Basic)
4. Structured Chart – MOS(Advanced)

Purpose of Structured Chart

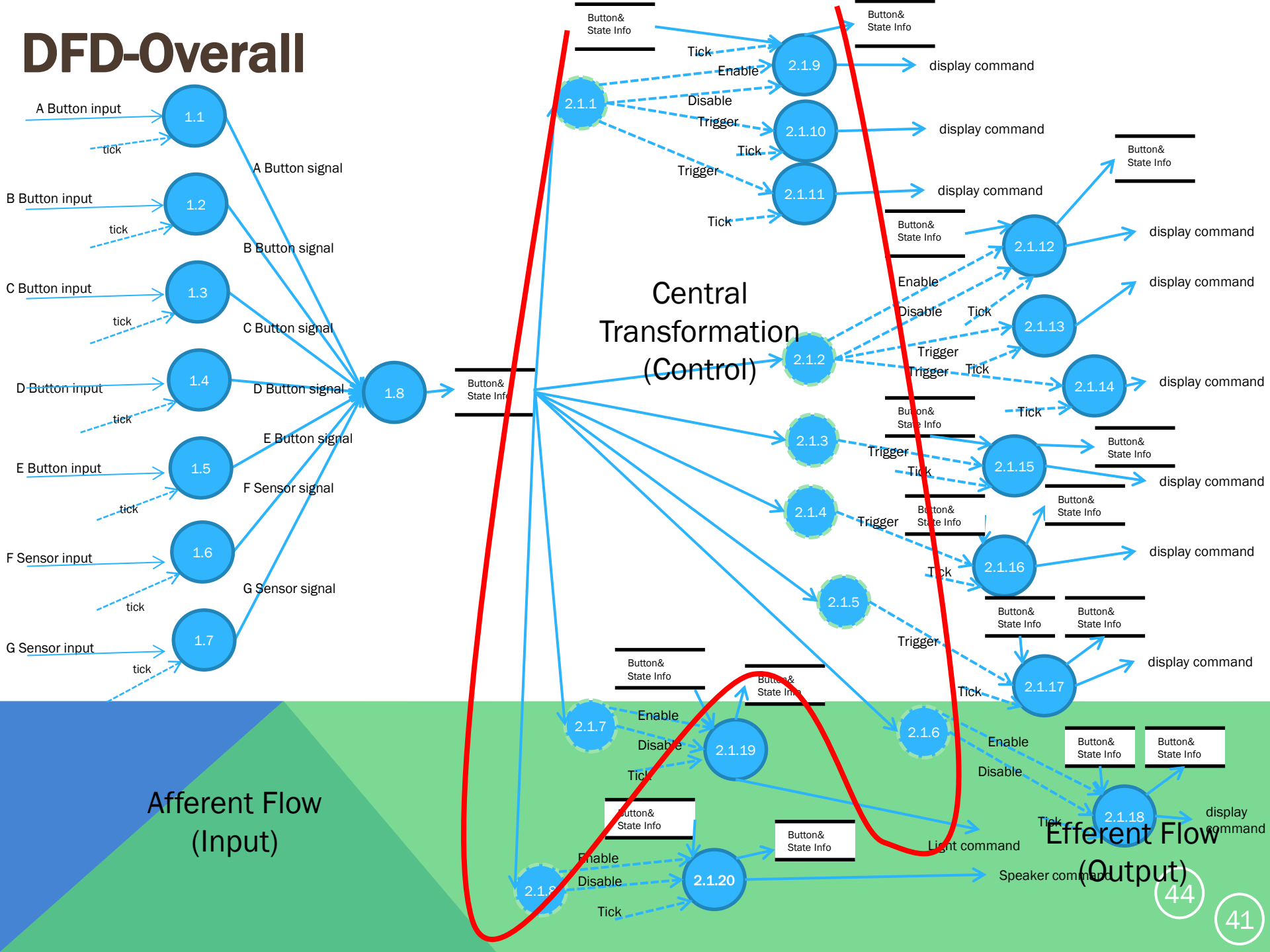
전자레인지 시스템의 전체 구조를 시각화.

각각의 명령어 수행에 대한 세분화.

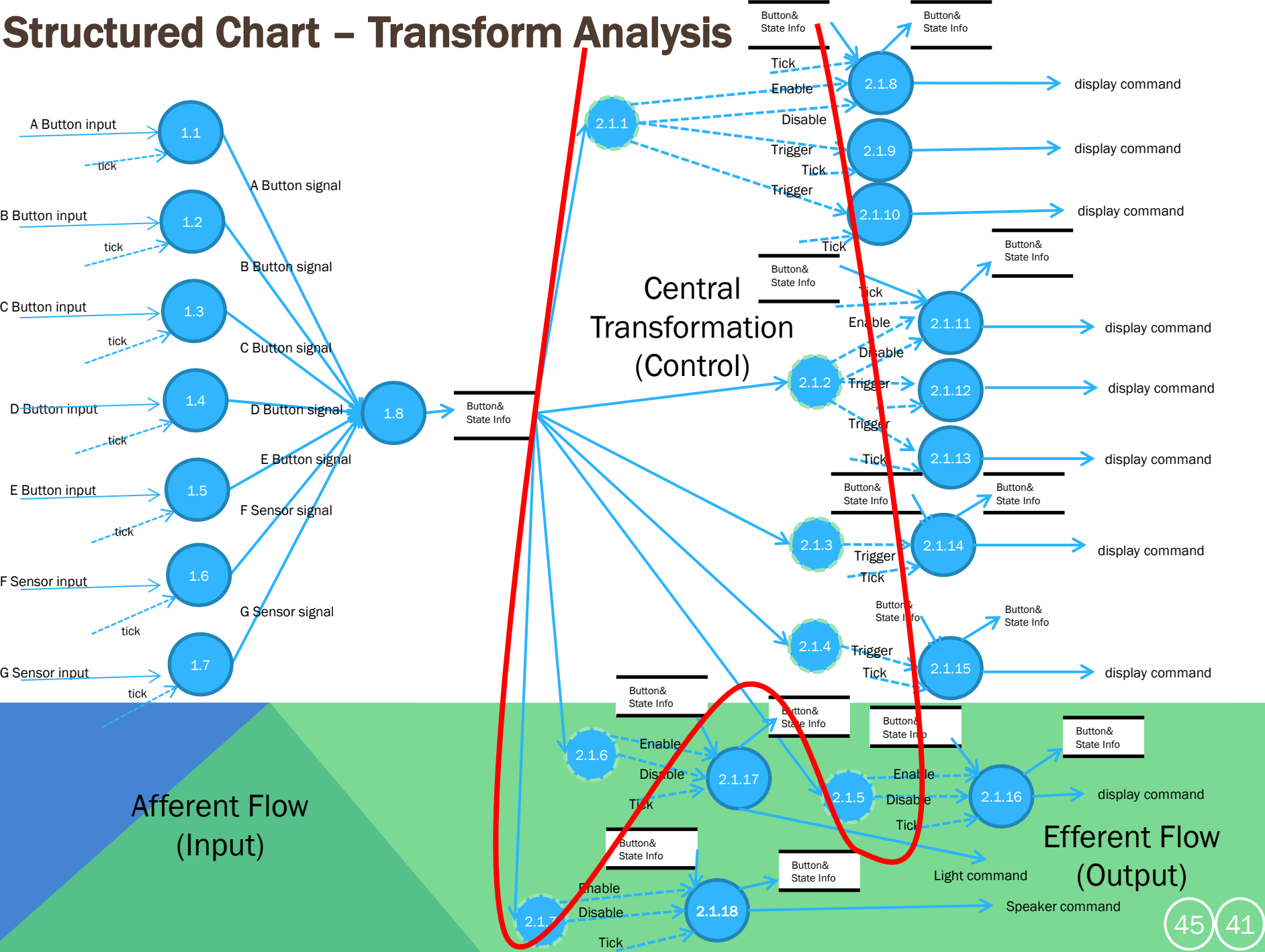
DFD-Overall



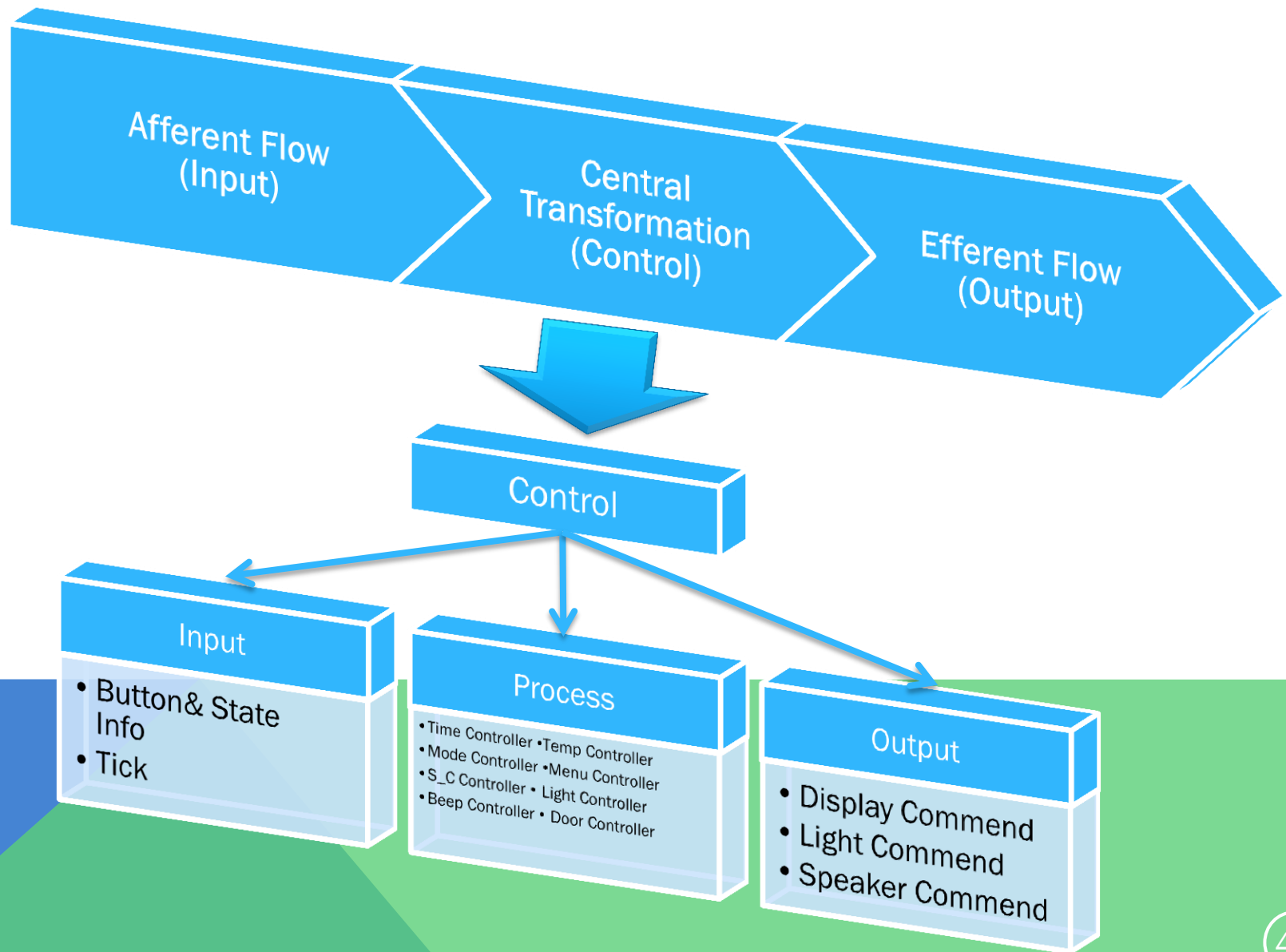
DFD-Overall



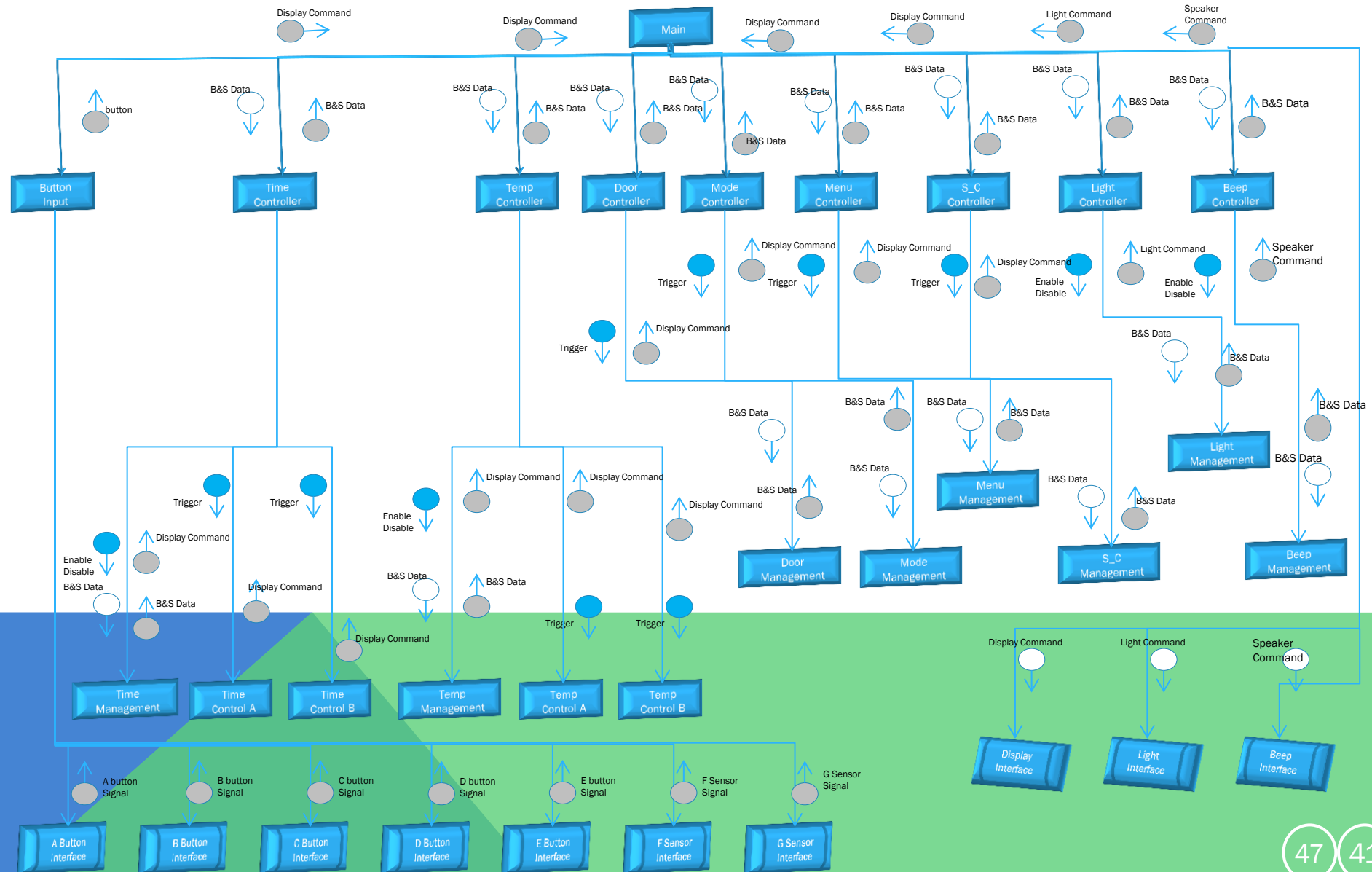
Structured Chart – Transform Analysis



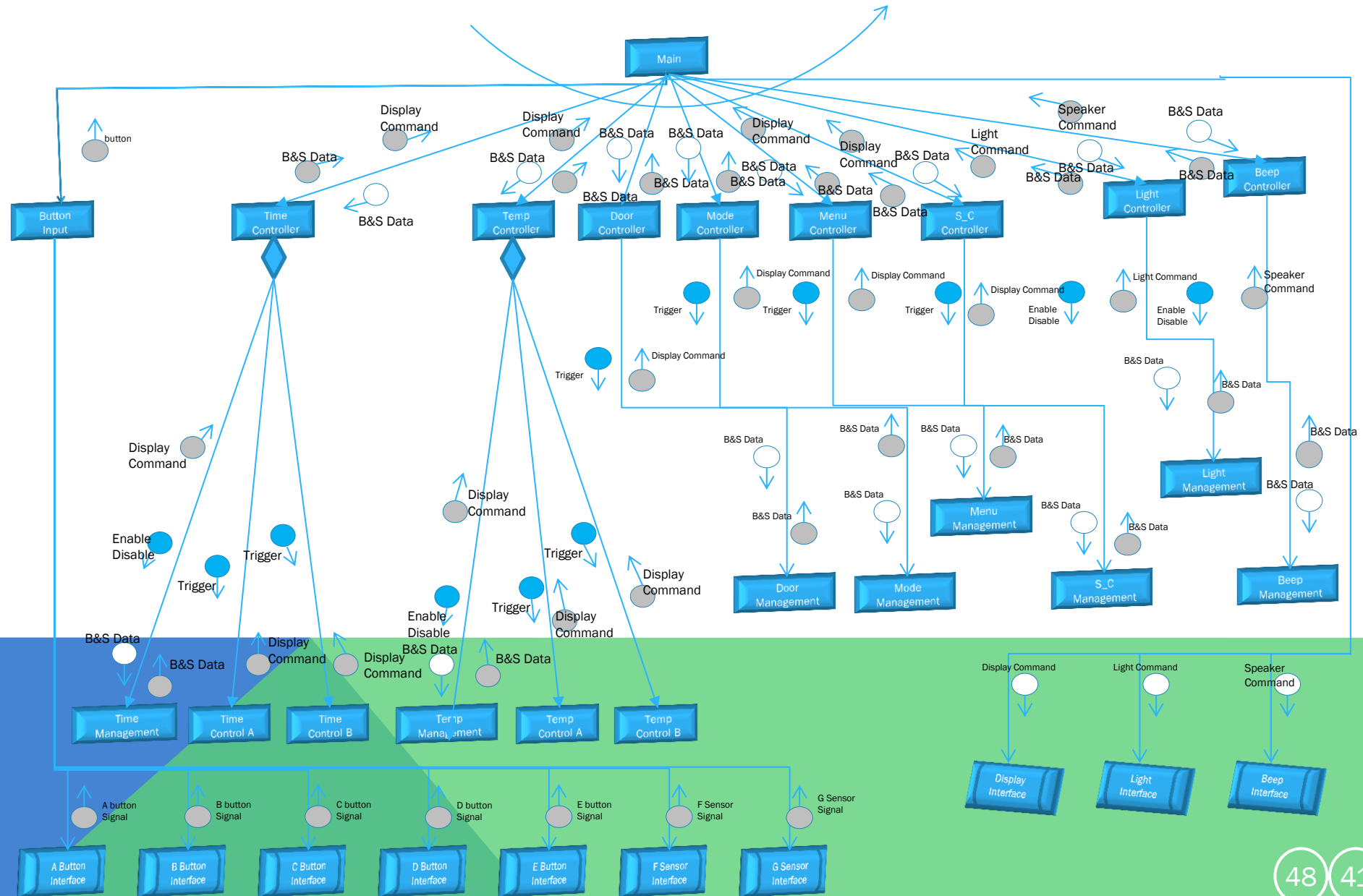
Structured Chart – Transform Analysis



Structured Chart – MOS(Basic)



Structured Chart – MOS(Advanced)



목차 - 3

1. Header

2. Function

HEADER – DATA STORE

```
typedef struct Button_Data{  
    boolean A_BUTTON_INPUT;  
    boolean B_BUTTON_INPUT;  
    boolean C_BUTTON_INPUT;  
    boolean D_BUTTON_INPUT;  
    boolean E_BUTTON_INPUT;  
    boolean F_SENSOR_INPUT;  
}Button_Data;  
  
typedef struct Button_State_Info{  
    Time_Data time_d;  
    Temp_Data temp_d;  
    Mode_Data mode_d;  
    Menu_Data menu_d;  
    SBCB_Data sbcb;  
    Door_Data dd;  
    Light_Data ld;  
    Beep_Data beep_d;  
    Button_Data bd;  
}Button_State_Info;  
  
//: typedef struct Time_Data{  
    int time;  
    int TimeFLAG;  
} Time_Data;  
  
typedef struct Temp_Data{  
    int temp;  
    int TempFLAG;  
} Temp_Data;  
  
typedef struct Mode_Data{  
    int mode;  
} Mode_Data;  
  
typedef struct Menu_Data{  
    int menu;  
} Menu_Data;  
  
typedef struct SBCB_Data{  
    int start_cancel;  
} SBCB_Data;  
  
typedef struct Door_Data{  
    int door_state;  
} Door_Data;  
  
typedef struct Light_Data{  
    int light_state;  
} Light_Data;  
  
typedef struct Beep_Data{  
    int beep_state;  
    int beep_count;  
} Beep_Data;
```

TRUE로 변환

FUNCTION - INPUT

```
Button_State_Info Button_Input(Button_State_Info* bsi){
```

```
    char key = 0;
    int tmp = 'a'-'A';

    if(kbhit()){
        key = getch();
        key = key - tmp;
    }

    switch(key)
    {
        case 'A':
            if(A_Button_Interface (bsi, key))
                return *bsi;
        case 'B':
            if(B_Button_Interface (bsi, key))
                return *bsi;
        case 'C':
            if(C_Button_Interface (bsi, key))
                return *bsi;
        case 'D':
            if(D_Button_Interface (bsi, key))
                return *bsi;
        case 'E':
            if(E_Button_Interface (bsi, key))
                return *bsi;
        case 'F':
            if(F_Sensor_Interface (bsi, key)) |
                return *bsi;
    }

    return *bsi;
}
```

FUNCTION - INPUT

Reference No.	1.8
name	Button input & Temp Sense
Input	A~E Button input G,F Sensor input
output	
Process Description	Button signal 혹은 Sensor signal이 data process 로 오면 Button state data store로 전달한다.

FUNCTION - CONTROLLER

```
int Time_Control_B(Button_State_Info* bsi){
```

```
    if((*bsi).time_d.time<=570)
        (*bsi).time_d.time = (*bsi).time_d.time+30;
    else if((*bsi).time_d.time==600)
        (*bsi).time_d.time = 30;
    else if((*bsi).time_d.time>570)
        (*bsi).time_d.time = 600;

    return (*bsi).time_d.time;
```

```
}
```

```
void Time_Management(Button_State_Info* bsi){
```

```
    if( (*bsi).sbc_b.start_cancel==0 ){
        if((*bsi).bd.A_BUTTON_INPUT==TRUE)
            (*bsi).time_d.time = Time_Control_A(bsi);

        else if((*bsi).bd.B_BUTTON_INPUT==TRUE)
            (*bsi).time_d.time = Time_Control_B(bsi);
    }
    else if( (*bsi).sbc_b.start_cancel==1 ){
        if((*bsi).time_d.time!=0){
            (*bsi).time_d.time--;
        }
    }
}
```

```
int Time_Controller(Button_State_Info* bsi){
```

```
    if((*bsi).mode_d.mode==0){
        Time_Management(bsi);
    }
    return (*bsi).time_d.time;
```

```
}
```

```
int Time_Control_A(Button_State_Info* bsi){
```

```
    if((*bsi).time_d.time<=590)
        (*bsi).time_d.time = (*bsi).time_d.time+1; // 10으로 바꿔야됨
    else if((*bsi).time_d.time==600)
        (*bsi).time_d.time = 10;
    else if((*bsi).time_d.time>590)
        (*bsi).time_d.time = 600;

    return (*bsi).time_d.time;
```

```
}
```

//시작된 상태일 때 시간 감소

PROCESS SPECIFICATION

Reference No.	2.1.1
name	Time Controller
Input	Tick (int)
output	Trigger (int)
Process Description	tick에 따라 시간을 제어해준다. Trigger를 Time management에 보내준다.
Reference No.	2.1.9
name	Time management
Input	Tick(int), enable, disable (bool)
output	display command (char)
Process Description	Enable일 경우 Time Management를 실행시키고 Disable일 경우 다른 trigger를 작동시킨다.

PROCESS SPECIFICATION

Reference No.	2.1.11
name	Time Control B
Input	Tick, trigger (int)
output	display command (char)
Process Description	Time Controller로부터 Trigger를 받아 조건에 맞춰 변경된 시간 값을 저장하고 변경된 시간 값을 출력하기 위해 Display Command에 정보를 보내준다.

Reference No.	2.1.10
name	Time Control A
Input	Trigger, Tick (int)
Output	display command (char)
Process Description	Time Controller로부터 Trigger를 받아 조건에 맞춰 변경된 시간 값을 저장하고 변경된 시간 값을 출력하기 위해 Display Command에 정보를 보내준다.

FUNCTION - CONTROLLER

```
int Temp_Control_A(Button_State_Info* bsi){  
    if((*bsi).temp_d.temp<=80)  
        (*bsi).temp_d.temp = (*bsi).temp_d.temp+10;  
    else if((*bsi).temp_d.temp==90)  
        (*bsi).temp_d.temp = 30;  
    else if((*bsi).temp_d.temp>90)  
        (*bsi).temp_d.temp = 90;  
  
    return (*bsi).temp_d.temp;  
}
```

```
int Temp_Control_B(Button_State_Info* bsi){  
  
    if((*bsi).temp_d.temp<=70)  
        (*bsi).temp_d.temp = (*bsi).temp_d.temp+20;  
    else if((*bsi).temp_d.temp==90)  
        (*bsi).temp_d.temp = 40;  
    else if((*bsi).temp_d.temp>70)  
        (*bsi).temp_d.temp = 90;  
  
    return (*bsi).temp_d.temp;  
}
```

```
int Temp_Controller(Button_State_Info* bsi){  
    if((*bsi).mode_d.mode==1){  
        Temp_Management(bsi);  
    }  
    return (*bsi).temp_d.temp;  
}
```

```
void Temp_Management(Button_State_Info* bsi){  
    if((*bsi).sbc_b.start_cancel==0){  
        if((*bsi).bd.A_BUTTON_INPUT==TRUE)  
            (*bsi).temp_d.temp = Temp_Control_A(bsi);  
  
        else if((*bsi).bd.B_BUTTON_INPUT==TRUE)  
            (*bsi).temp_d.temp = Temp_Control_B(bsi);  
    }  
    else if((*bsi).sbc_b.start_cancel==1){ // 시지  
        (*bsi).temp_d.temp--;  
    }  
}
```


PROCESS SPECIFICATION

Reference No.	2.1.2
name	Temp Controller
Input	Tick (int)
output	Trigger (int)
Process Description	tick에 따라 온도 메뉴를 제어해준다. Trigger를 Temp management에 보내준다.

Reference No.	2.1.12
name	Temp management
Input	Enable, disable (bool), tick (int)
output	Display Command (char)
Process Description	Enable일 경우 Temp Management를 실행시키고 Disable일 경우 다른 trigger를 작동시킨다.

PROCESS SPECIFICATION

Reference No.	2.1.13
name	Temp Control A
Input	Tick, trigger (int)
output	display command (char)
Process Description	Temp Controller로부터 Trigger를 받아 조건에 맞춰 변경된 온도 값을 저장하고 변경된 온도 값을 출력하기 위해 Display Command에 정보를 보내준다.

Reference No.	2.1.14
name	Temp Control B
Input	Trigger, Tick (int)
Output	Display Command (char)
Process Description	Temp Controller로부터 Trigger를 받아 조건에 맞춰 변경된 온도 값을 저장하고 변경된 온도 값을 출력하기 위해 Display Command에 정보를 보내준다.

FUNCTION - CONTROLLER

```
int Mode_Controller(Button_State_Info* bsi){  
    if((*bsi).bd.C_BUTTON_INPUT==TRUE){  
        Mode_Management(bsi);  
    }  
    return (*bsi).mode_d.mode;  
}
```

```
void Mode_Management(Button_State_Info* bsi){  
    if((*bsi).mode_d.mode==0){  
        (*bsi).mode_d.mode=1;  
    }  
    else if((*bsi).mode_d.mode==1){  
        (*bsi).mode_d.mode=0;  
    }  
    else printf("오류"); //예외처리  
}
```

PROCESS SPECIFICATION

Reference No.	2.1.3
name	Mode Controller
Input	Tick (int)
output	Trigger (int)
Process Description	Tick에 따라 모드를 제어해준다. Trigger를 mode management로 보내준다.

Reference No.	2.1.15
name	Mode management
Input	Trigger, Tick (int)
output	Display Command (char)
Process Description	Mode Controller로부터 Trigger를 받아 조건에 맞춰 변경된 시간/온도 선택지를 저장하고 변경된 시간/온도 선택지를 출력하기 위해 Display Command에 정보를 보내준다.

FUNCTION - CONTROLLER

```
└─
└─ int Menu_Controller(Button_State_Info* bsi){
    |
    |   if((*bsi).bd.D_BUTTON_INPUT==TRUE){
    |       Menu_Management(bsi);
    |   }
    |   return (*bsi).menu_d.menu;
    |
    }

└─ void Menu_Management(Button_State_Info* bsi){
    |
    |   (*bsi).menu_d.menu =(*bsi).menu_d.menu+1;
    |
    |   if((*bsi).menu_d.menu==6){
    |       (*bsi).menu_d.menu=0;
    |   }
    |
    }
```

PROCESS SPECIFICATION

Reference No.	2.1.4
name	Menu Controller
Input	Tick (int)
output	Trigger (int)
Process Description	Tick에 따라 메뉴를 제어해준다. Trigger를 Menu management로 보내준다.

Reference No.	2.1.16
name	Menu management
Input	Trigger, Tick (int)
output	Display Command (char)
Process Description	Menu Controller로부터 Trigger를 받아 조건에 맞춰 변경된 메뉴를 저장하고 변경된 메뉴를 출력하기 위해 Display Command에 정보를 보내준다.

FUNCTION - CONTROLLER

```
int Door_Controller(Button_State_Info *bsi){  
    if((*bsi).bd.F_SENSOR_INPUT==TRUE){  
        Door_Management(bsi);  
    }  
  
    return (*bsi).dd.door_state;  
}  
  
void Door_Management(Button_State_Info *bsi){  
    if((*bsi).dd.door_state==0){  
        (*bsi).dd.door_state=1;  
    }  
  
    else if((*bsi).dd.door_state==1){  
        (*bsi).dd.door_state=0;  
    }  
  
    else printf("오류"); //예외처리  
}
```

PROCESS SPECIFICATION

Reference No.	2.1.5
name	Door Controller
Input	Tick (int)
output	Trigger(Boolean)
Process Description	Tick에 따라 문 개폐를 제어해준다. Trigger를 Door management로 보내준다.

Reference No.	2.1.17
name	Door management
Input	Trigger (boolean), tick (int)
output	Display command (char)
Process Description	enable상태의 조건이 맞으면 전자레인을 가동한다. disable상태의 조건이 맞으면 전자레인지를 가동을 중지한다.

FUNCTION - CONTROLLER

```
void S_C_Management(Button_State_Info* bsi){
```

```
    if((*bsi).sbc_b.start_cancel==0){
```

```
        if( ((*bsi).time_d.time != 0) && (*bsi).bd.E_BUTTON_INPUT == 1 ){
            (*bsi).time_d.TimeFLAG = 1;      //TimeFLAG 시간을 지정했을 경우 시작된 상태
            (*bsi).sbc_b.start_cancel=1;
            basic = 1;
        }
```

```
        else if( (*bsi).temp_d.temp != 20 && (*bsi).bd.E_BUTTON_INPUT == 1 ){
            (*bsi).temp_d.TempFLAG = 1;      //TempFLAG 온도를 지정해서 시작된 상태
            (*bsi).sbc_b.start_cancel=1;
            basic = 1;
        }
    }
```

```
    else if((*bsi).sbc_b.start_cancel==1){
```

```
        if( (*bsi).time_d.time == 0 && (*bsi).mode_d.mode == 0 ){
            (*bsi).time_d.TimeFLAG = 0;      //TimeFLAG 시간을 지정했을 경우 중지 상태
            (*bsi).sbc_b.start_cancel=0;
        }
```

```
        else if( (*bsi).temp_d.temp == 20 && (*bsi).mode_d.mode ==1 ){
            (*bsi).temp_d.TempFLAG = 0;      //TempFLAG 온도를 지정해서 중지 상태
            (*bsi).sbc_b.start_cancel=0;
        }
    }
```

```
    else printf("오류"); //예외처리
```

```
int S_C_Controller(Button_State_Info* bsi){
```

```
    if( ((*bsi).bd.E_BUTTON_INPUT==TRUE) && ((*bsi).bd.F_SENSOR_INPUT==FALSE) ){
        S_C_Management(bsi);
    }
```

```
    else{
        S_C_Management(bsi);
    }
```

```
    return (*bsi).sbc_b.start_cancel;
```

```
}
```

PROCESS SPECIFICATION

Reference No.	2.1.6
name	S_C Controller
Input	Tick (int)
output	Disable, enable (boolean)
Process Description	Tick에 따라 전자레인지 가동을 제어해준다. disable, enable를 S_C management로 보내준다.

Reference No.	2.1.18
name	S_C management
Input	enable, disable (boolean), tick (int)
output	Display command (char)
Process Description	enable상태의 조건이 맞으면 전자레인지를 가동한다. disable상태의 조건이 맞으면 전자레인지 가동을 중지한다.

FUNCTION - CONTROLLER

```
int Light_Controller(Button_State_Info* bsi){  
    if( ( (*bsi).sbc.b.start_cancel==1 && (*bsi).dd.door_state==0 ) || ( (*bsi).dd.door_state==1 ) ){ //  
        Light_Management(bsi);  
    }  
  
    else if( ( (*bsi).sbc.b.start_cancel==0) && ((*bsi).dd.door_state==0) ) ){  
        Light_Management(bsi);  
    }  
  
    return (*bsi).ld.light_state;  
}  
  
void Light_Management(Button_State_Info* bsi){  
    if((*bsi).ld.light_state==0){  
        if( ((*bsi).sbc.b.start_cancel==1 && (*bsi).dd.door_state==0) || ( (*bsi).dd.door_state==1 ) )  
            (*bsi).ld.light_state=1;  
    }  
  
    else if((*bsi).ld.light_state==1){  
        (*bsi).ld.light_state=0;  
    }  
  
    else printf("오류"); //예외처리  
}
```

PROCESS SPECIFICATION

Reference No.	2.1.7
name	Light Controller
Input	Tick (int)
output	Disable, enable (bool)
Process Description	Tick에 따라 조명을 제어해준다. Disable, enable를 light management로 보내준다.

Reference No.	2.1.19
name	Light Management
Input	Tick(int), enable, disable(bool)
output	Light command (bool)
Process Description	Enable, disable에 따라 light_off에 light command를 보낸다.

FUNCTION - CONTROLLER

```
int Beep_Controller(Button_State_Info* bsi){  
    if( (((*bsi).time_d.TimeFLAG==0) || (*bsi).temp_d.TempFLAG==0) && (*bsi).sbcg.start_cancel==0 && basic == 1){  
        Beep_Management(bsi);  
    }  
  
    else if( (*bsi).temp_d.TempFLAG==1 || (*bsi).time_d.TimeFLAG==1 ){           // 안나는 조건  
        Beep_Management(bsi);  
    }  
  
    else{                               // 상황 유지 부분  
        Beep_Management(bsi);  
    }  
  
    return (*bsi).beep_d.beep_state;  
}  
  
void Beep_Management(Button_State_Info* bsi){  
    if((*bsi).beep_d.beep_state==0)           //beep_state=0 소리 안들림 beep_state=1 소리 들릴 준비  
    {  
        if( ((*bsi).time_d.time == 0) && bsi->sbcg.start_cancel == 0 && basic == 1 ){  
            (*bsi).beep_d.beep_state=1;  
        }  
    }  
    /* ... */  
}
```

PROCESS SPECIFICATION

Reference No.	2.1.8
name	Beep Controller
Input	Tick (int)
output	Disable, enable (bool)
Process Description	Tick에 따라 소리를 제어해준다. Disable, enable를 beep management로 보내준다.

Reference No.	2.1.20
name	Beep Management
Input	Tick(int), enable, disable (bool)
output	Speaker command (bool)
Process Description	Enable, disable에 따라 sound_off에 speaker command를 보낸다.

FUNCTION - CONTROLLER

```
void Display_Interface(Button_State_Info* bsi){  
    printf("온도 %d // 시간 %d : %d \n",(*bsi).temp_d.temp, ((*bsi).time_d.time)/60, ((*bsi).time_d.time)%60);  
    printf("메뉴 상태 : %d \n", (*bsi).menu_d.menu);  
}
```

```
void Light_Interface(Button_State_Info* bsi){  
    if((*bsi).dd.door_state==1 || (*bsi).sbc.d.start_cancel==1){  
        SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),RED);  
    }  
    else SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),WHITE);  
}
```

```
void Beep_Interface(Button_State_Info* bsi){  
    if( ((*bsi).beep_d.beep_state == 1) && ((*bsi).time_d.time == 0)){  
        printf("ㄹㄹ");  
        (*bsi).beep_d.beep_count--;  
    }  
    if( (*bsi).beep_d.beep_count == -1){  
        (*bsi).beep_d.beep_state=0;  
        (*bsi).beep_d.beep_count=2;  
        bsi->time_d.TimeFLAG = 0;  
        basic = 0;  
    }  
}
```

PROCESS SPECIFICATION

Reference No.	2.2
name	Display Interface
Input	Display Command(bool)
output	Display
Process Description	입력받은 커맨드를 바탕으로 결과값을 화면에 출력해준다.

Reference No.	2.3
name	Light Interface
Input	Light Command(bool)
output	Light
Process Description	입력받은 커맨드를 바탕으로 결과값을 화면에 출력해준다.

PROCESS SPECIFICATION

Reference No.	2.3
name	Beep Interface
Input	Speaker Command(bool)
output	Speaker
Process Description	입력받은 커맨드를 바탕으로 결과값을 beep로 출력해준다.

FUNCTION - MAIN

```
void main (void){  
  
    Button_State_Info bsi;  
  
    init(&bsi);           //정보 초기화  
    init_button(&bsi);    //입력키 초기화  
  
    while(1){  
  
        bsi = Button_Input(&bsi);  
  
        bsi.time_d.time = Time_Controller(&bsi);  
        bsi.temp_d.temp = Temp_Controller(&bsi);  
        bsi.mode_d.mode = Mode_Controller(&bsi);  
        bsi.menu_d.menu = Menu_Controller(&bsi);  
        bsi.scb.start_cancel = S_C_Controller(&bsi);  
        bsi.dd.door_state = Door_Controller(&bsi);  
        bsi.ld.light_state = Light_Controller(&bsi);  
        bsi.beep_d.beep_state = Beep_Controller(&bsi);  
  
        Light_Interface(&bsi);  
        Display_Interface(&bsi);  
        Beep_Interface(&bsi);  
  
        init_button(&bsi);  
        Sleep(100);  
        system("cls");  
    }  
}
```

감사합니다.