

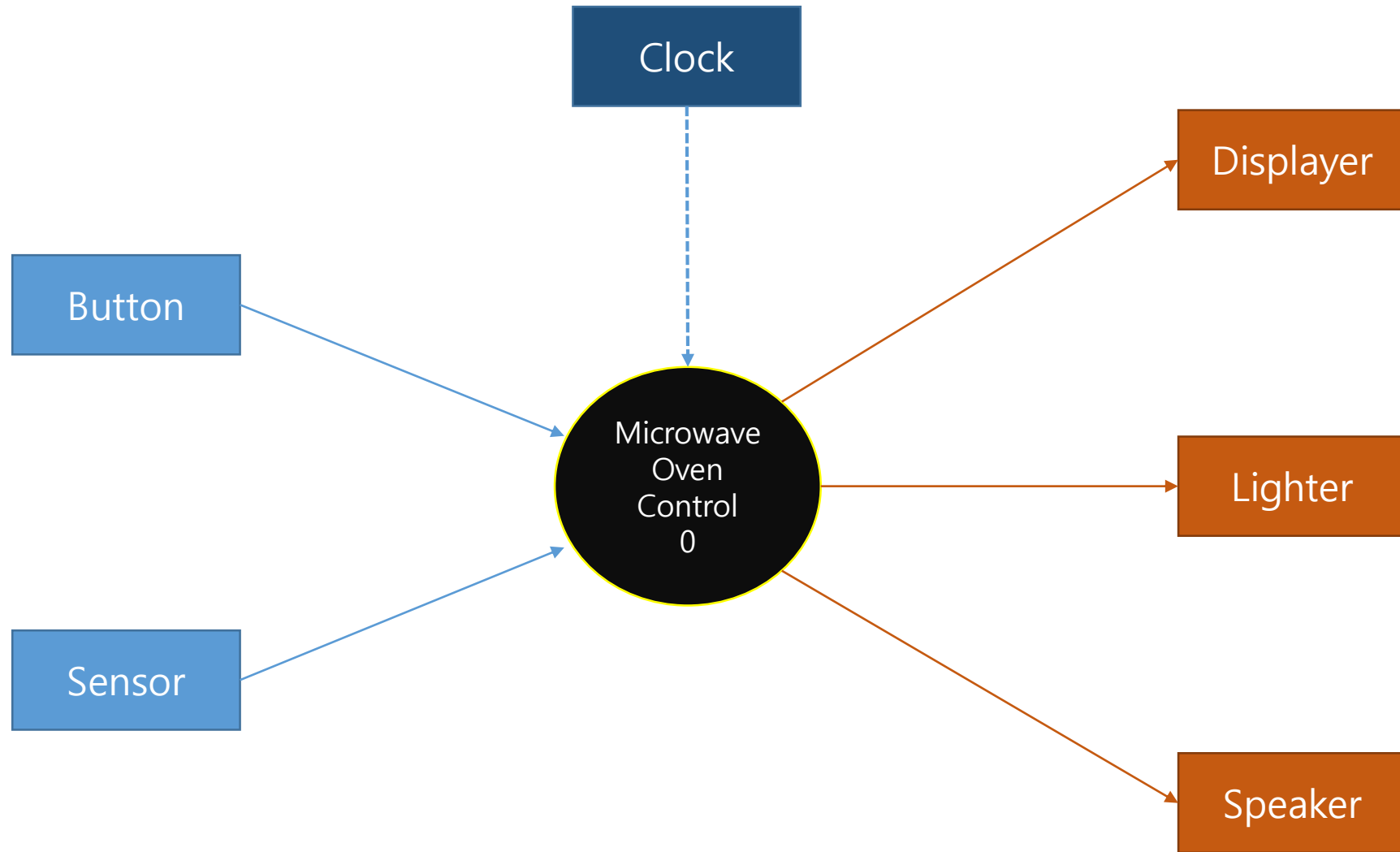
Microwave Oven System

T3

이경수,한득환,김대희,신민용

Structured Analysis

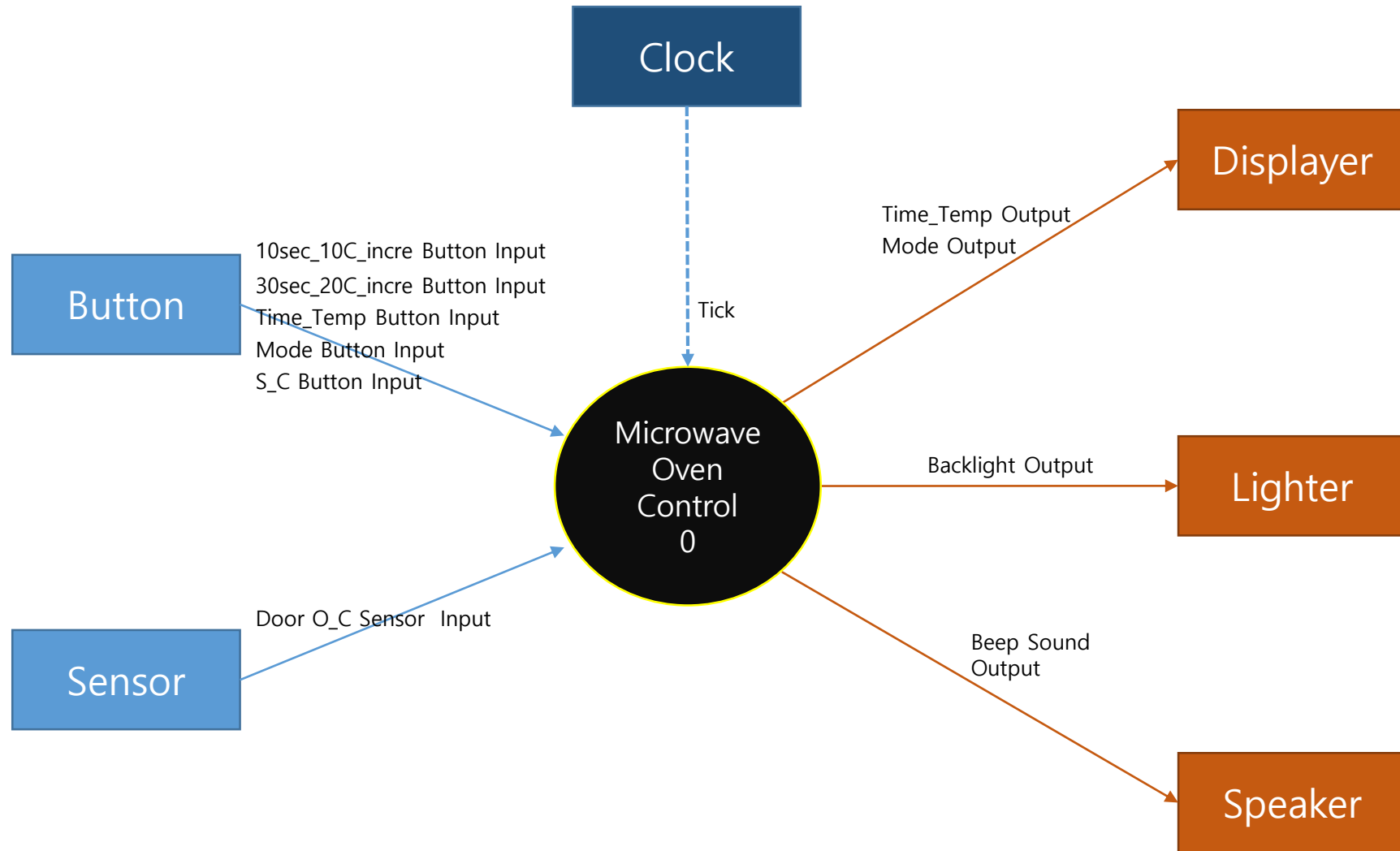
System Context Diagram



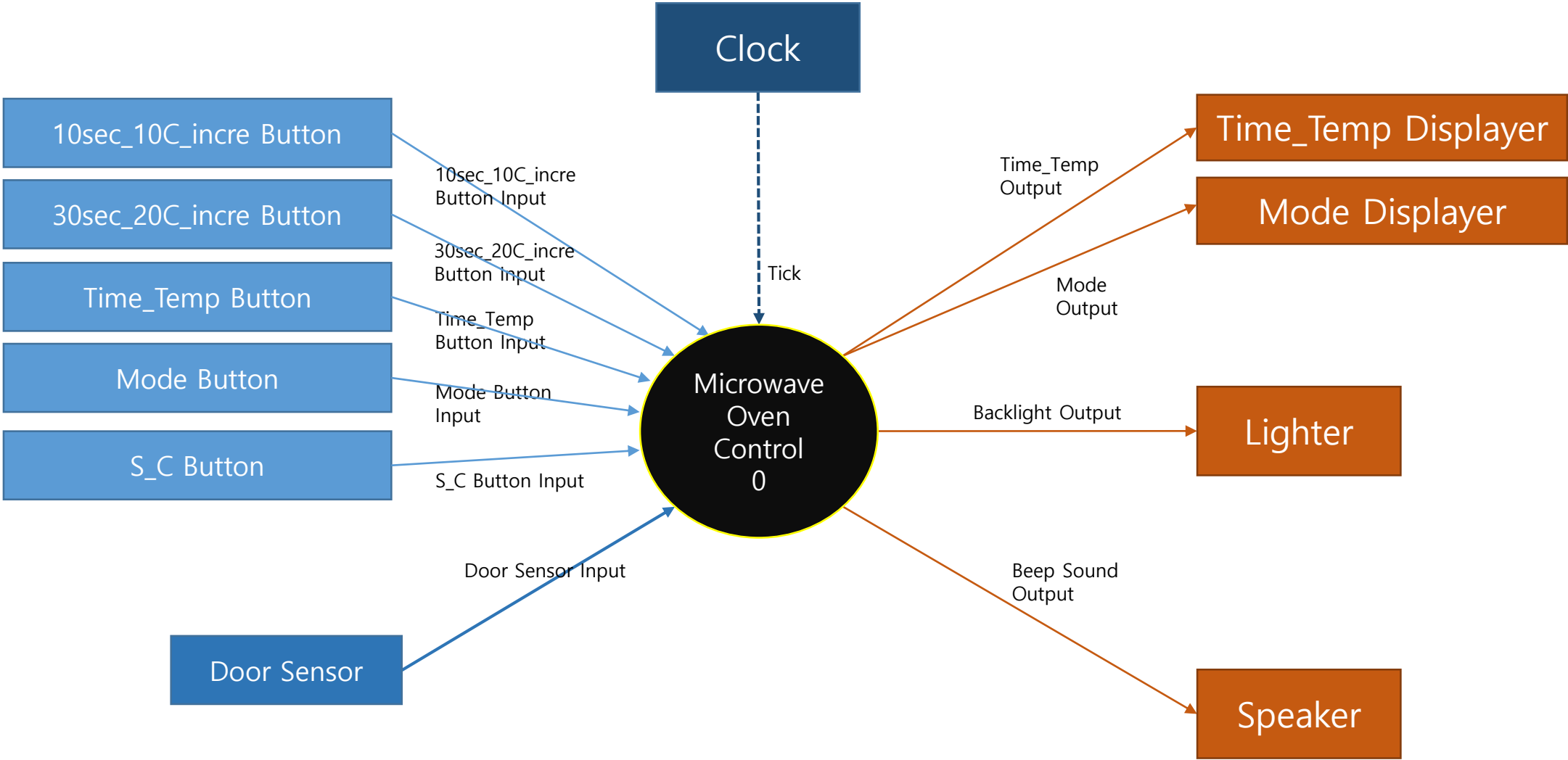
Event List

Event	Description
10sec_10C_incre Button Input	한번 누를 때 마다 시간/온도가 10sec/10C 씩 증가한다.
30sec_20C_incre Button Input	한번 누를 때 마다 시간/온도가 30sec/20C씩 증가한다.
Time_Temp Button Input	초기에는 Time으로 설정, 누를 때 마다 Time/Temp가 번갈아 변경된다.
Mode Button Input	미리 설정된 조리방법에 따라 정해진 시간으로 설정된다.
S_C Button Input	조리를 시작/취소 한다.
Door O_C Sensor Input	문을 열고 닫는다.
Temp Sensor Input	현재의 온도를 감지한다.
Tick	시간을 측정 해준다. Tick 하나당 1초로 가정한다.
Time_Temp Output	현재 시간 / 온도 를 보여준다.
Mode Output	현재 모드를 보여준다.
Backlight Output	Lighter을 통해 빛을 방출한다.
Beep Sound Output	Speaker를 통해 비프음을 낸다.

System Context Diagram



DFD Level 0



DFD Level 0 - Data Dictionary (1/2)

7

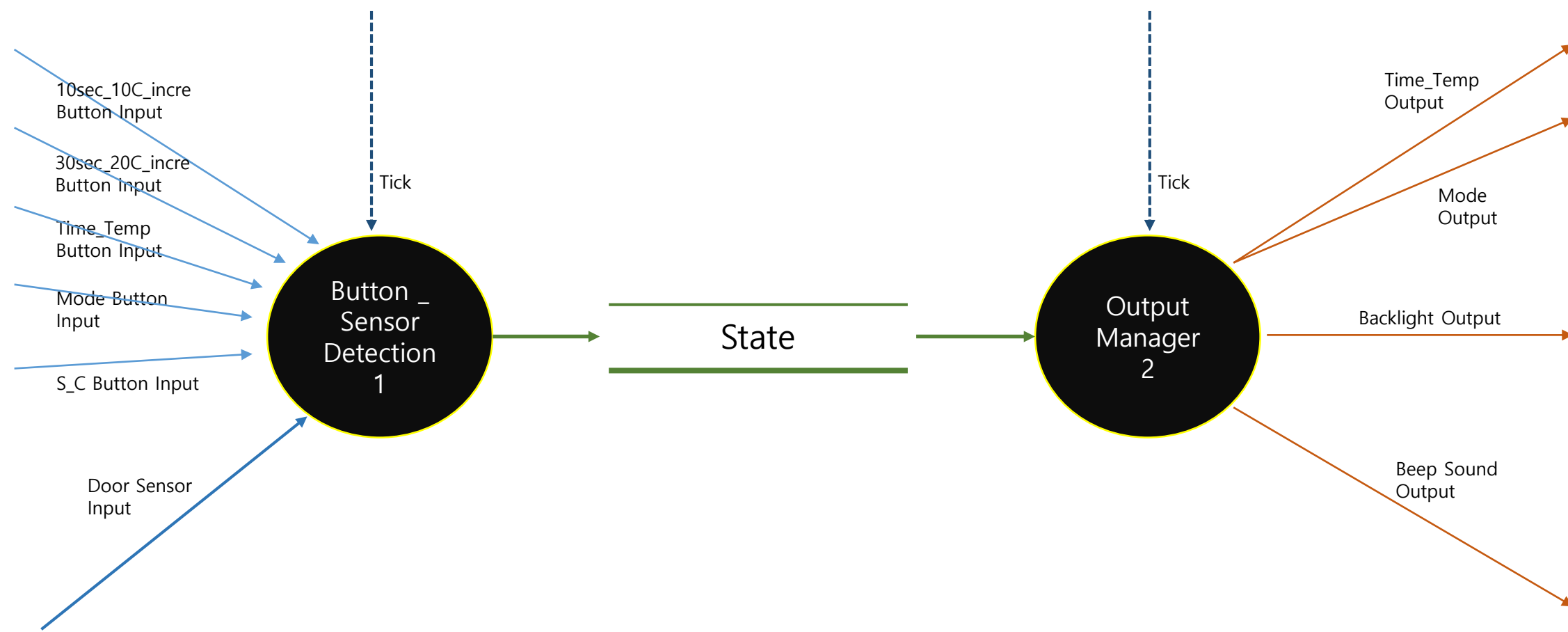
Event	Description	Format	Type
10sec_10C_incre Button Input	한번 누를 때 마다 시간/온도가 10sec/10C 씩 증가한다. Default는 False이고, 버튼의 입력이 들어 올 때 만 True가 된다.	False / True (Bool)	Interrupt
30sec_20C_incre Button Input	한번 누를 때 마다 시간/온도가 30sec/20C씩 증가한다. Default는 False이고, 버튼의 입력이 들어 올 때 만 True가 된다.	False / True (Bool)	Interrupt
Time_Temp Button Input	초기에는 Time으로 설정되어있고, 누를 때 마다 Temp -> Time 으로 번갈아 가며 변경된다. Default는 False이고, 버튼의 입력이 들어 올 때만 True가 된다.	False / True (Bool)	Interrupt
Mode Button Input	미리 설정된 조리방법에 따라 정해진 시간으로 설정된다. 처음에는 Manual로 되어있다. 버튼을 누를 때마다 "떡 -> 죽 -> 밥 -> 국 -> 피자 -> Manual -> 떡 ..."으로 진행된다.	Manual(0) / Dduck(1) / Juk(2) / Bob(3) / Kuk / Jjigae(4) / Pizza(5) (Integer)	Interrupt
S_C Button Input	조리를 시작/취소 한다. 조리 전이면 "시작" 하고, 조리 중이면 "취소 " 한다. Default는 False이고, 버튼의 입력이 들어 올 때 만 True가 되어 상황에 맞게 동작 한다.	False / True (Bool)	Interrupt
Door O_C Sensor Input	문을 열고 닫는다. 조리 중에 문을 열면 조리가 멈추고 Backlight가 켜진다. Default는 False이고, 버튼의 입력이 들어 올 때 만 True가 된다.	False / True (Bool)	Interrupt

DFD Level 0 - Data Dictionary (2/2)

8

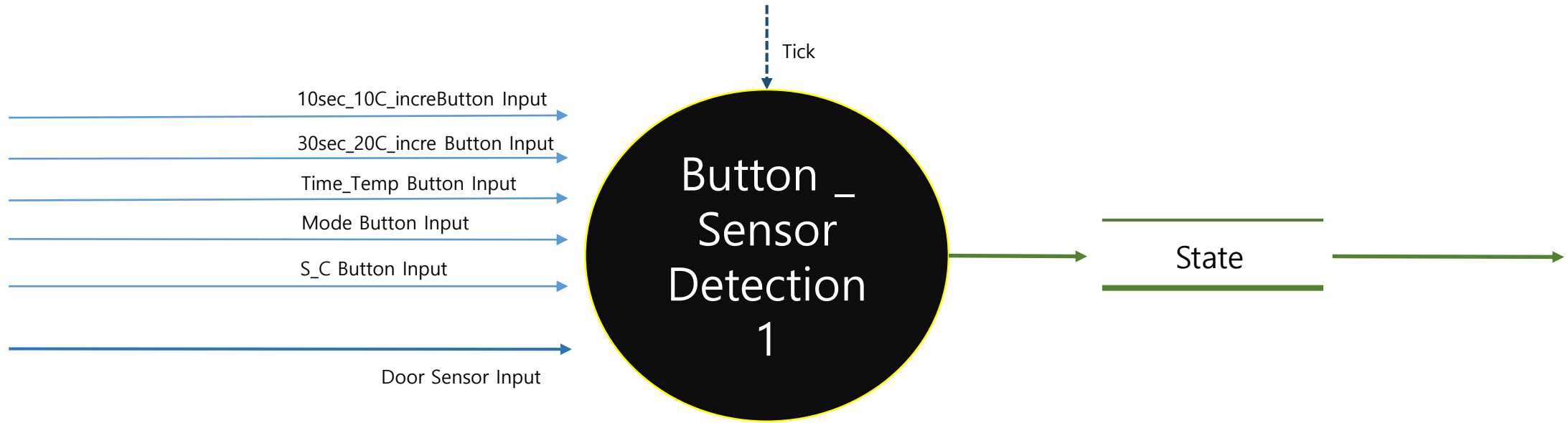
Event	Description	Format	Type
Tick	시간을 측정 해준다. Tick 하나당 1초로 가정한다. Default는 False이고 1초마다 Tick이 True가 되어 들어간다.	False / True (Bool)	Temporal
Time_Temp Output	현재 시간 / 온도 를 보여준다.	00 : 00 / 00 C (Integer / Char)	Periodic
Mode Output	현재 모드를 보여준다. "00 : manual", "01 : 떡", "02 : 죽", "03 : 밥", "04 : 국_찌개", "05 : 피자" 이렇게 보여준다.	00 : ModeName (Integer / Char)	Periodic
Backlight Output	Lighter를 통해 온도와 시간, 모드를 노란색으로 표시해준다. Default는 False이고 문이 열렸을 때나 조리 중에만 켜진다.	False / True (Bool)	Periodic
Beep Sound Output	Speaker를 통해 비프음을 낸다. Default는 False이고 조리가 끝났을때 만 3초간 비프음을 낸다.	False / True (Bool)	Periodic

DFD Level 1



DFD Level 1 – Process Specification

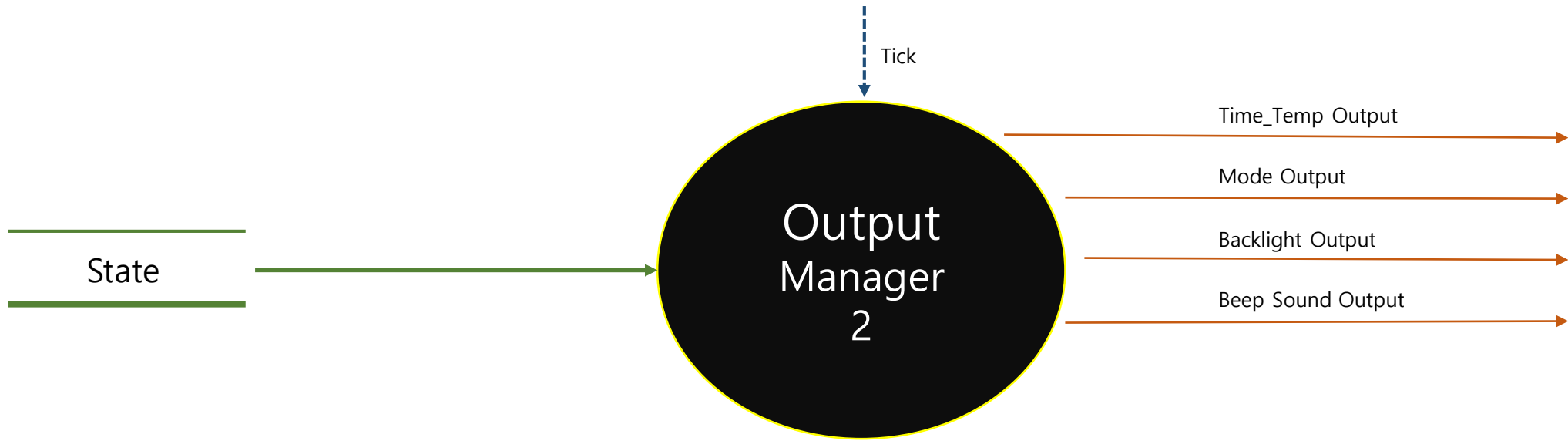
10



Reference No.	1
Name	Button _ Sensor Detection
Input	10sec_10C_incre Button Input, 30sec_20C_incre Button Input, Time_Temp Button Input, Mode Button Input, S_C Button Input, Door Sensor Input, Tick
Output	Input States / Door State
Process Description	여러 인풋들의 스테이트들을 모아 다음 프로세스에 전달한다.

DFD Level 1 – Process Specification

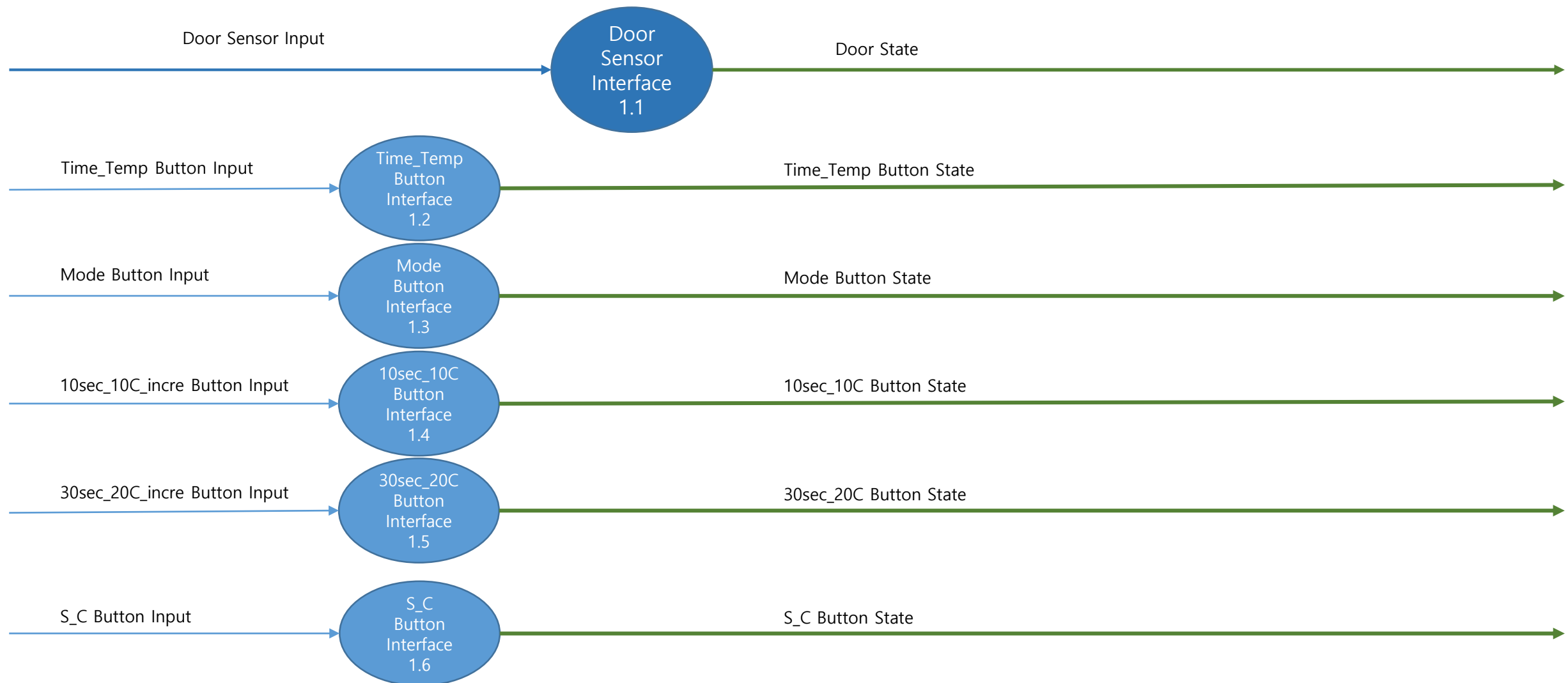
11



Reference No.	2
Name	Output Manager
Input	Input States / Door State, Tick
Output	Time_Temp Output, Mode Output, Backlight Output, Beep Sound Output
Process Description	여러 인풋들의 스테이트들 가지고 출력을 해준다.

DFD Level 2 (1/2)

12

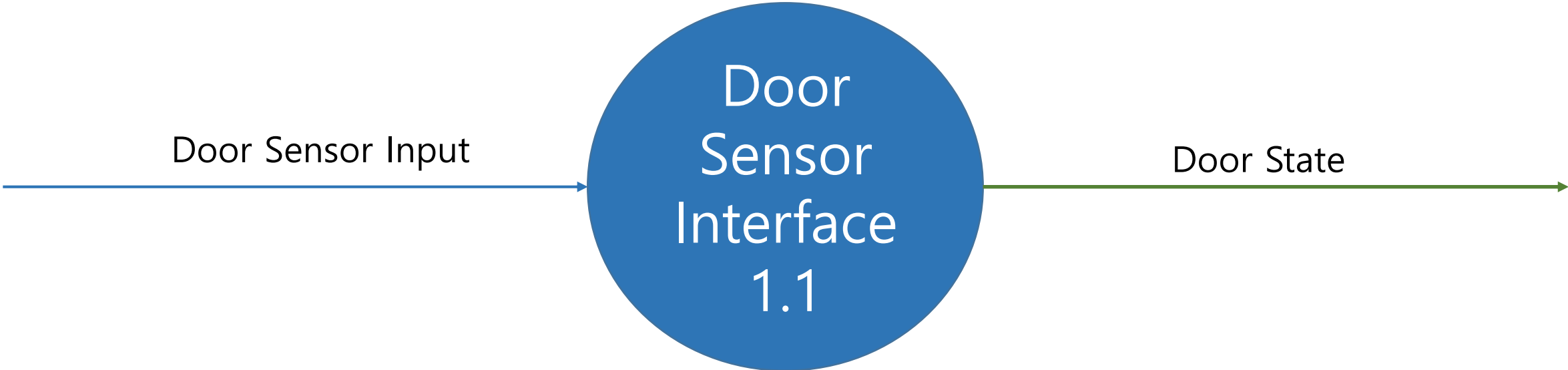


DFD Level 2 - Data Dictionary (1/2)

13

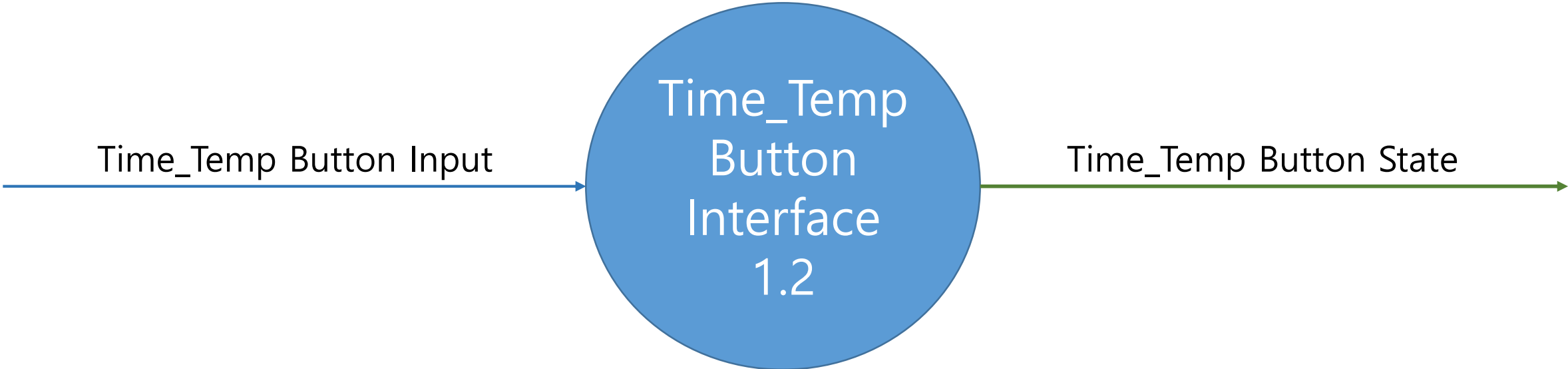
Event	Description	Format	Type
10sec_10C Button State	10sec_10C_incre Button의 입력이 들어왔는지 안 왔는지에 대한 상태를 나타낸다. (후에 1010B는 10sec_10C Button State 가 True면 1, False면 0 이다.)	False / True (Bool)	Interrupt
30sec_20C Button State	30sec_20C_incre Button의 입력이 들어왔는지 안 왔는지에 대한 상태를 나타낸다. (후에 3020B는 30sec_20C Button State 가 True면 1, False면 0 이다.)	False / True (Bool)	Interrupt
Time_Temp Button State	Time_Temp Button의 입력이 들어왔는지 안 왔는지에 대한 상태를 나타낸다. (후에 TT는 Time_Temp Button State가 True면 1, False면 0 이다.)	False / True (Bool)	Interrupt
Mode Button State	Mode Button의 입력이 들어왔는지 안 왔는지에 대한 상태를 나타낸다. (후에 M는 Mode Button State가 True면 1, False면 0 이다.)	False / True (Bool)	Interrupt
S_C Button State	S_C Button의 입력이 들어왔는지 안 왔는지에 대한 상태를 나타낸다. (후에 S는 S_C Button State가 True면 1, False면 0 이다.)	False / True (Bool)	Interrupt
Door State	문이 열려있는지 닫혀있는지에 대한 상태를 나타낸다. (후에 D는 Door State가 True면 1, False면 0 이다.)	False / True (Bool)	Interrupt

DFD Level 2 (1/2) – Process Specification



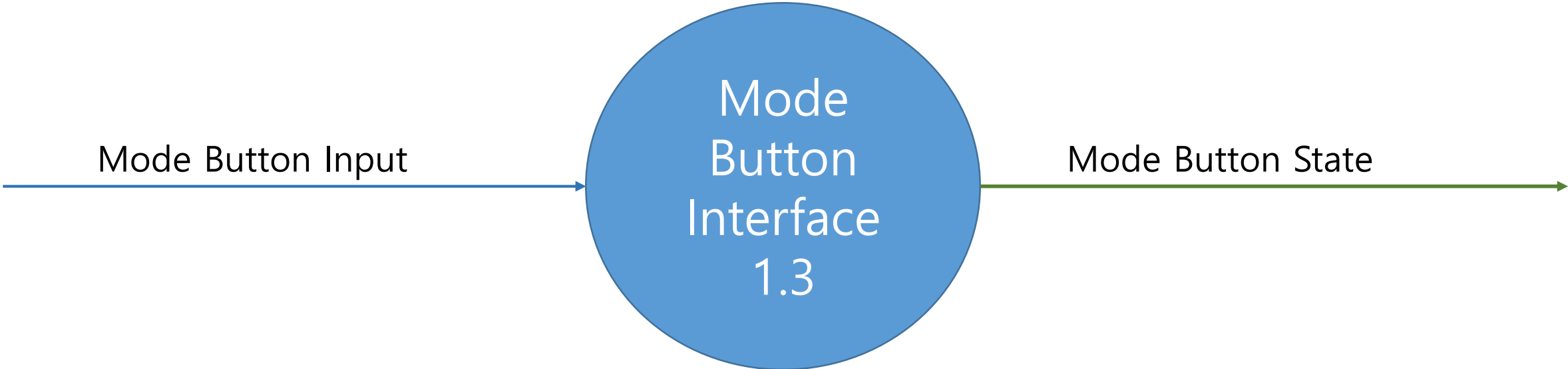
Reference No.	1.1
Name	Door Sensor Interface
Input	Door Sensor Input
Output	Door State (Bool)
Process Description	Door Sensor Input이 눌렸는지 안 눌렸는지 판단하여 Door State 를 True/False 로 내보낸다.

DFD Level 2 (1/2) – Process Specification



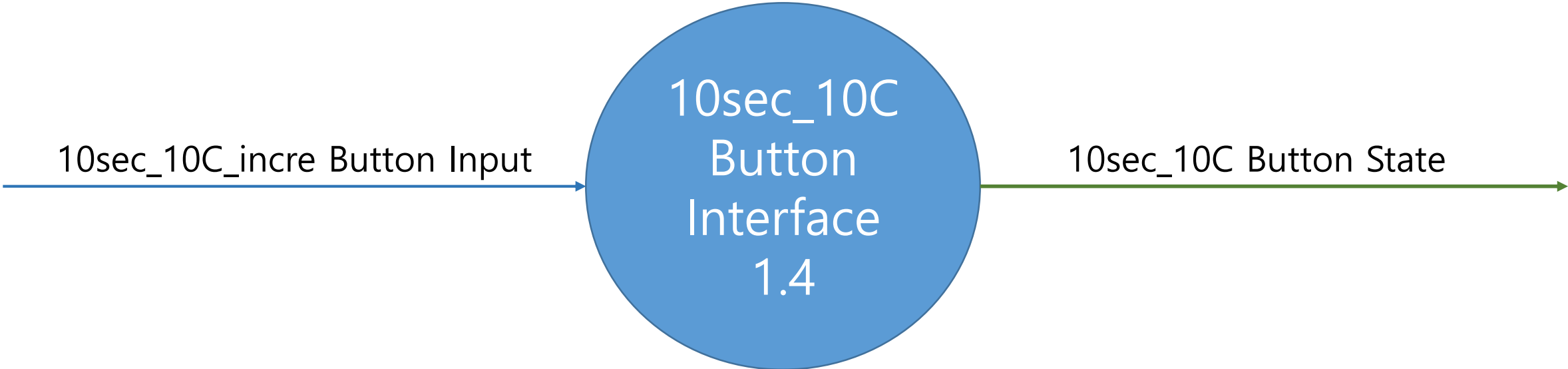
Reference No.	1.2
Name	Time_Temp Button Interface
Input	Time_Temp Button Input
Output	Time_Temp Button State (Bool)
Process Description	Time_Temp Button Input이 눌렸는지 안 눌렸는지 판단하여 Time_Temp State 를 True/False 로 내보낸다.

DFD Level 2 (1/2) – Process Specification



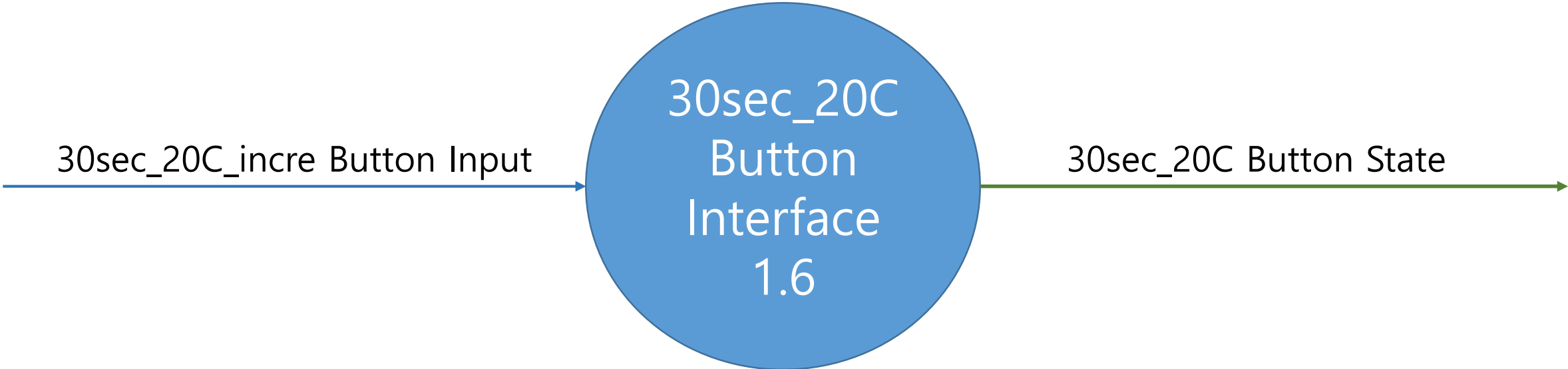
Reference No.	1.3
Name	Mode Button Interface
Input	Mode Button Input
Output	Mode Button State (Bool)
Process Description	Mode Button Input이 눌렸는지 안 눌렸는지 판단하여 Mode Button State 를 True/False 로 내보낸다.

DFD Level 2 (1/2) – Process Specification



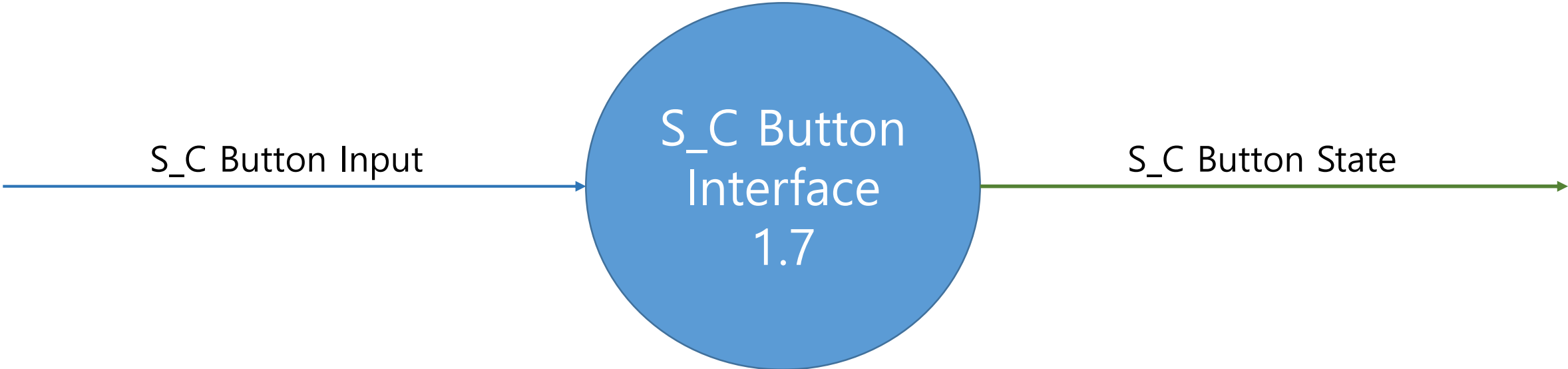
Reference No.	1.4
Name	10sec_10C Button Interface
Input	10sec_10C_incre Button Input
Output	10sec_10C Button State (Bool)
Process Description	10sec_10C_incre Button Input이 눌렸는지 안 눌렸는지 판단하여 10sec_10C Button State 를 True/False 로 내보낸다.

DFD Level 2 (1/2) – Process Specification



Reference No.	1.6
Name	30sec_20C Button Interface
Input	30sec_20C_incre Button Input
Output	30sec_20C Button State (Bool)
Process Description	30sec_20C_incre Button Input이 눌렸는지 안 눌렸는지 판단하여 30sec_20C Button State 를 True/False 로 내보낸다.

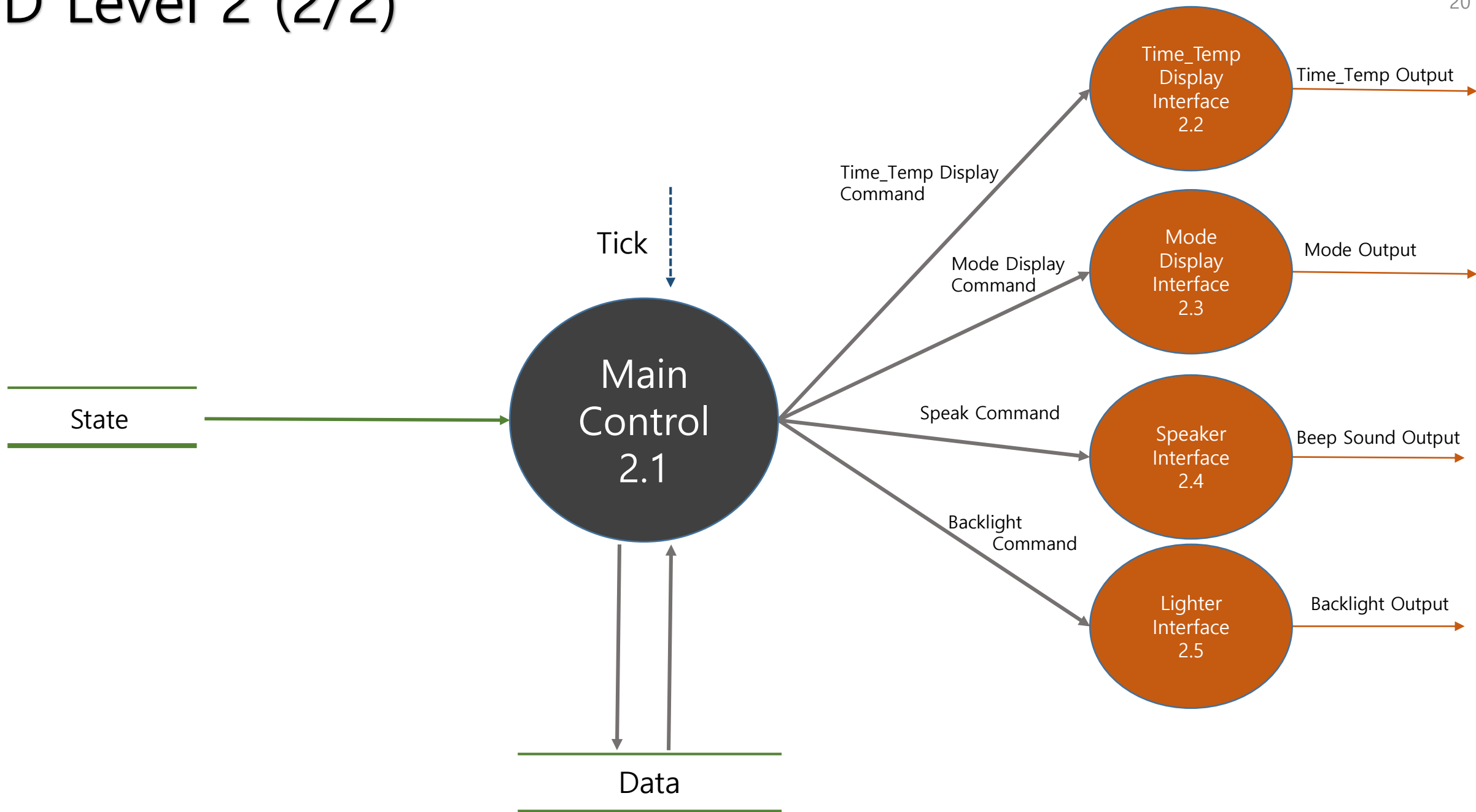
DFD Level 2 (1/2) – Process Specification



Reference No.	1.7
Name	S_C Button Interface
Input	S_C Button Input
Output	S_C Button State (Bool)
Process Description	S_C Button Input이 눌렸는지 안 눌렸는지 판단하여 S_C Button State 를 True/False 로 내보낸다.

DFD Level 2 (2/2)

20

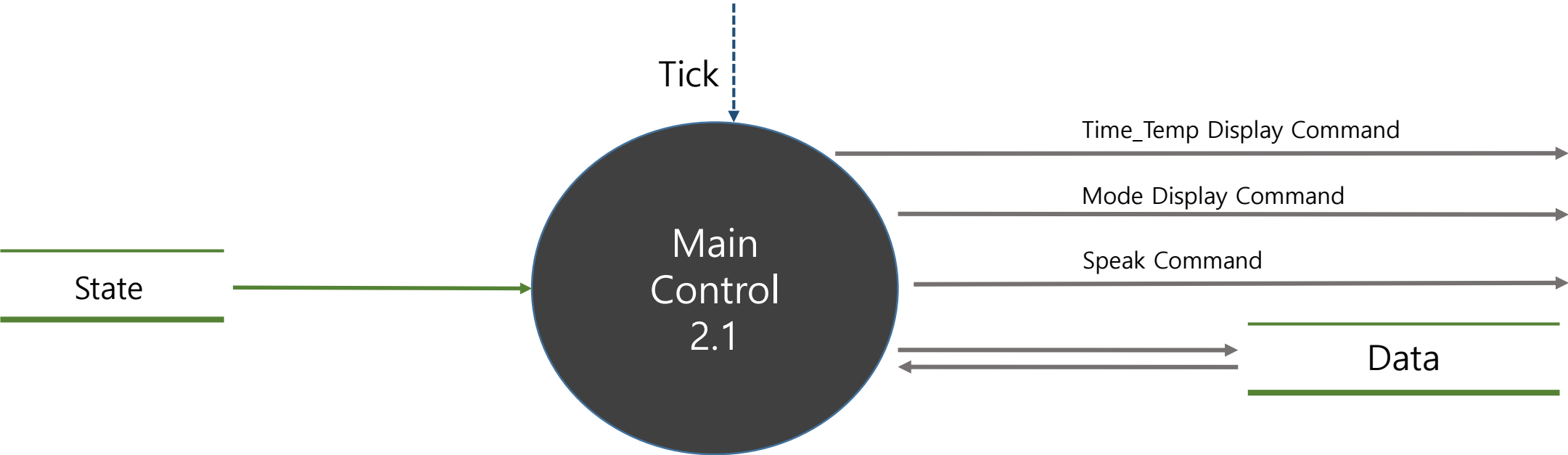


DFD Level 2 - Data Dictionary (2/2)

21

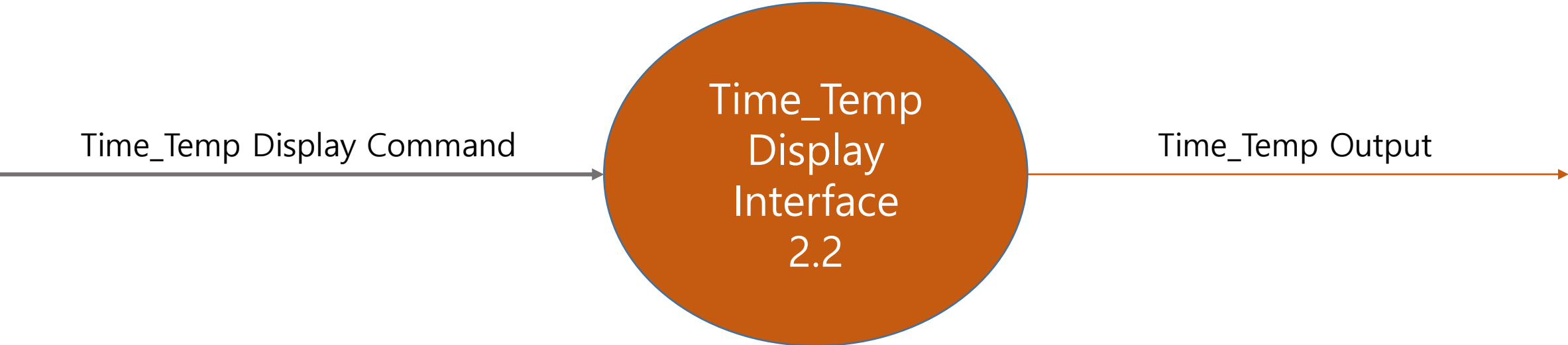
Event	Description	Format	Type
Time_Temp Display Command	Time_Temp Display Interface 를 통해 현재 시간과 온도를 출력하도록 명령 한다.	False / True (Bool)	Interrupt
Mode Display Command	Mode Display Interface 를 통해 현재 모드를 출력하도록 명령 한다.	False / True (Bool)	Interrupt
Speak Command	Speaker Interface 를 소리를 방출 하도록 명령 한다.	False / True (Bool)	Interrupt
Backlight Command	Lighter Interface를 통해 빛을 방출 하도록 명령 한다.	00 (Integer)	Periodic
Data	입력에 따른 시간을 stime 에 저장하고, 입력에 따른 온도를 stemp 에 저장한다. 그리고 현재 시간을 ctime, 현재 온도를 ctemp에 저장한다. 그리고 cmode에는 현재 모드가 저장된다.	00 "AB" (Integer, Char)	Periodic

DFD Level 2 (2/2) – Process Specification



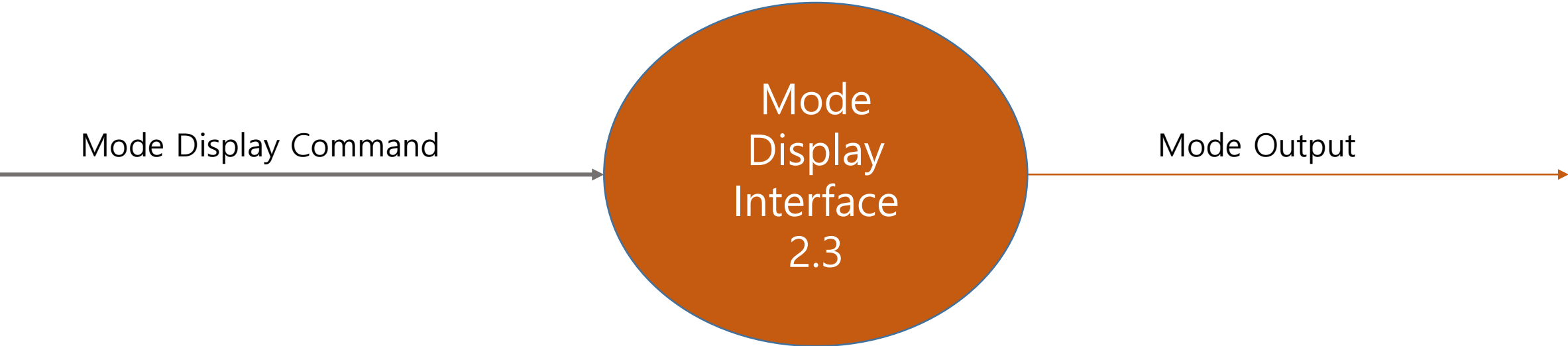
Reference No.	2.1
Name	Main Control
Input	State
Output	Time_Temp Display Command, Mode Display Command, Speak Command, Data
Process Description	버튼들의 인풋 스테이트를 받아와서 Data에 저장하고 끌어와 출력을 제어 한다.

DFD Level 2 (2/2) – Process Specification



Reference No.	2.2
Name	Time_Temp Display Interface
Input	Time_Temp Display Command (Bool)
Output	Time_Temp Output
Process Description	컨트롤에서 받아온 Time_Temp Display Command 가 True면 시간 또는 온도를 출력해 준다.

DFD Level 2 (2/2) – Process Specification



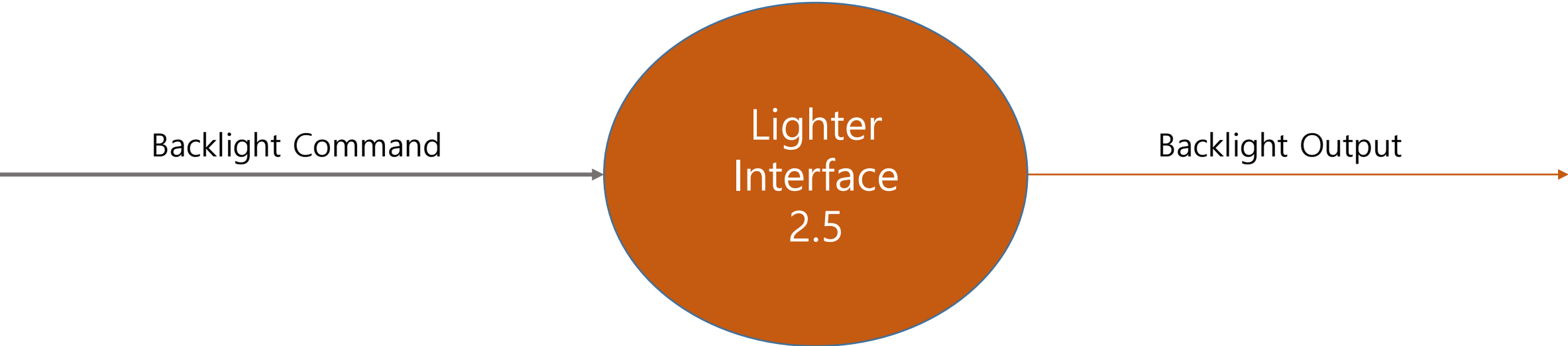
Reference No.	2.3
Name	Mode Display Interface
Input	Mode Display Command (Bool)
Output	Mode Output
Process Description	컨트롤에서 받아온 Mode Display Command 가 True면 모드를 출력해 준다.

DFD Level 2 (2/2) – Process Specification



Reference No.	2.4
Name	Speaker Interface
Input	Speak Command (Bool)
Output	Beep Sound Output
Process Description	컨트롤에서 받아온 Speak Command 가 True면 비프음을 3초간 출력해 준다.

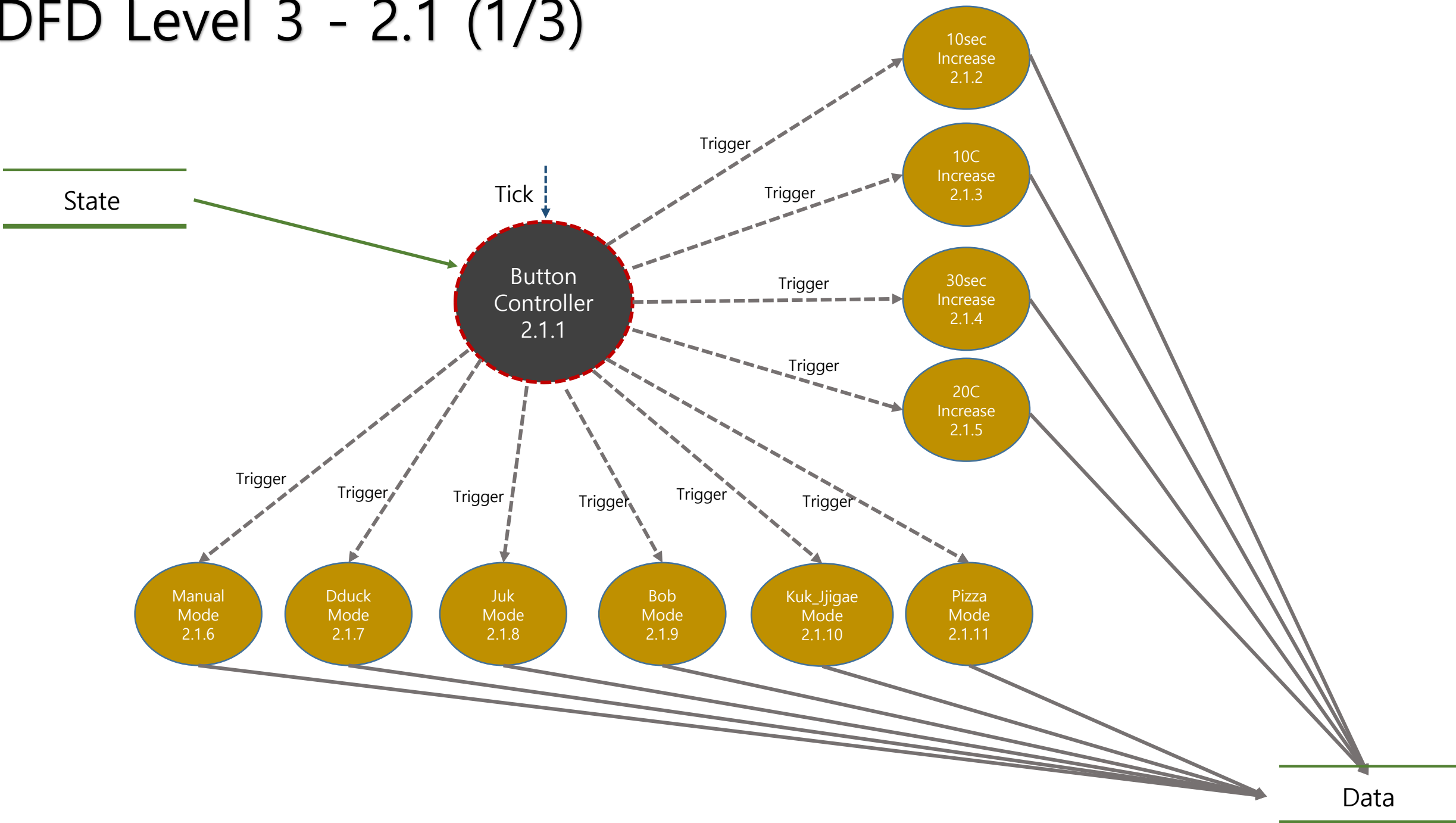
DFD Level 2 (2/2) – Process Specification



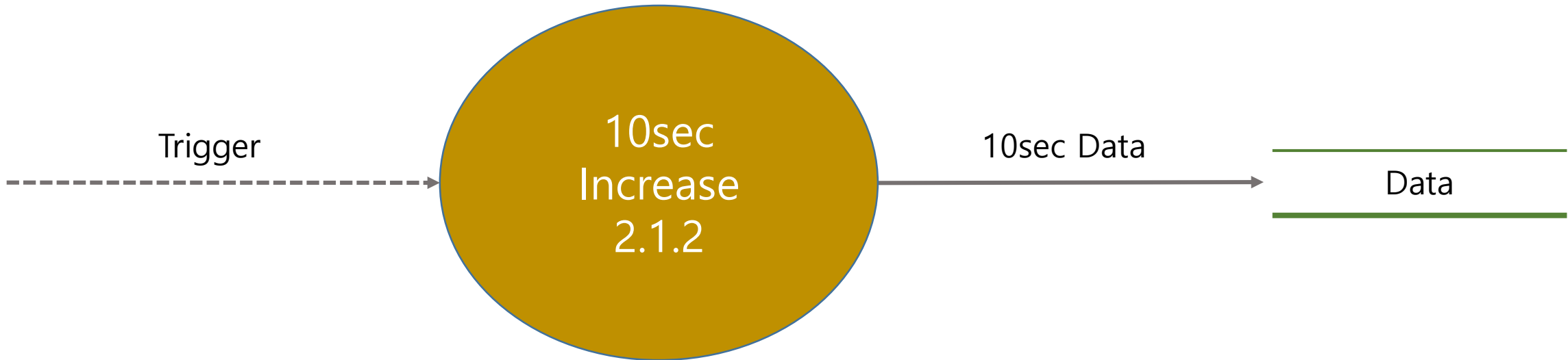
Reference No.	2.5
Name	Lighter Interface
Input	Backlight Command (Bool)
Output	Backlight Output
Process Description	컨트롤에서 받아온 Backlight Command 가 True면 빛을 방출해 준다.

DFD Level 3 - 2.1 (1/3)

27

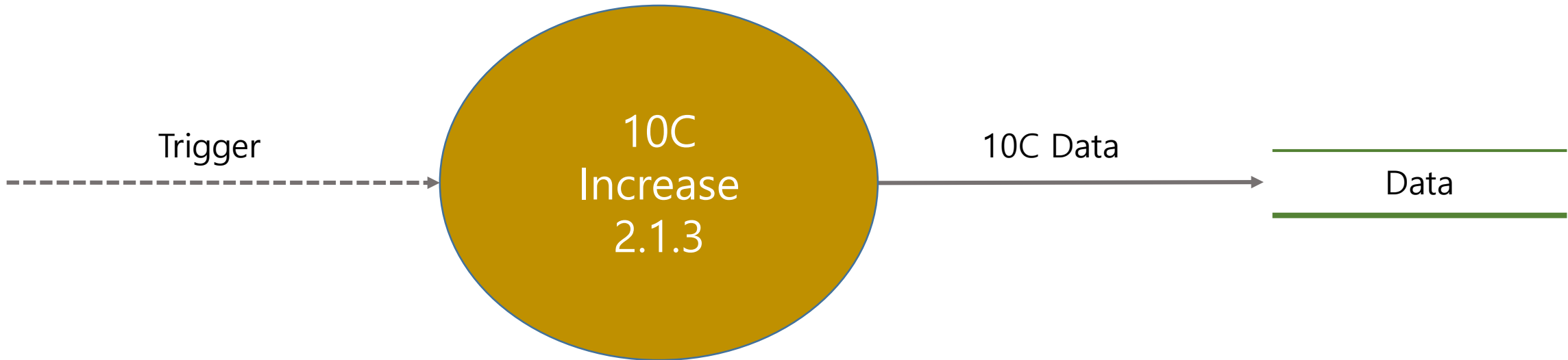


DFD Level 3 – 2.1 Process Specification



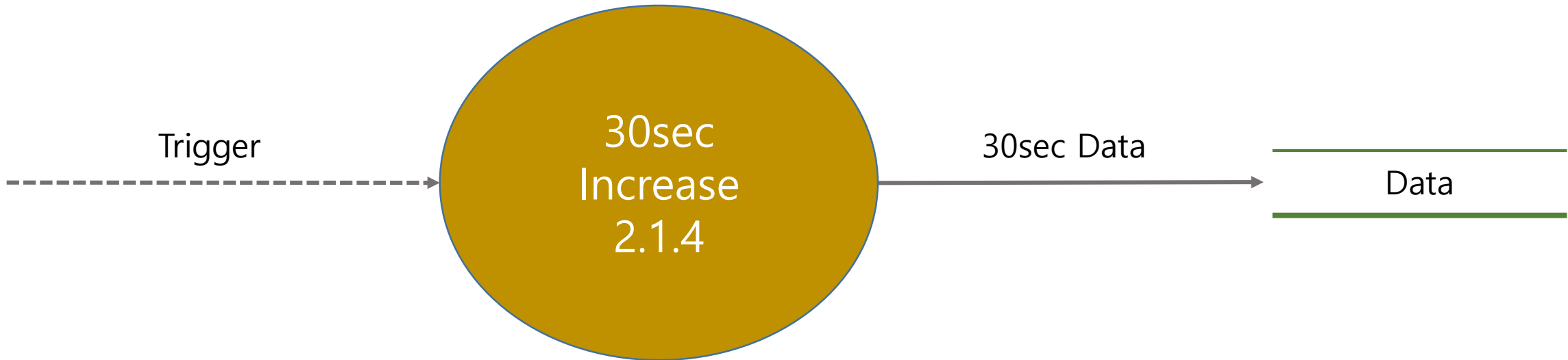
Reference No.	2.1.2
Name	10sec Increase
Input	Trigger
Output	10sec Data (Integer)
Process Description	10sec Increase 에 Trigger가 들어 오면 10초 라는 시간 데이터를 Data에 있는 stime에 누적 시킨다.

DFD Level 3 – 2.1 Process Specification



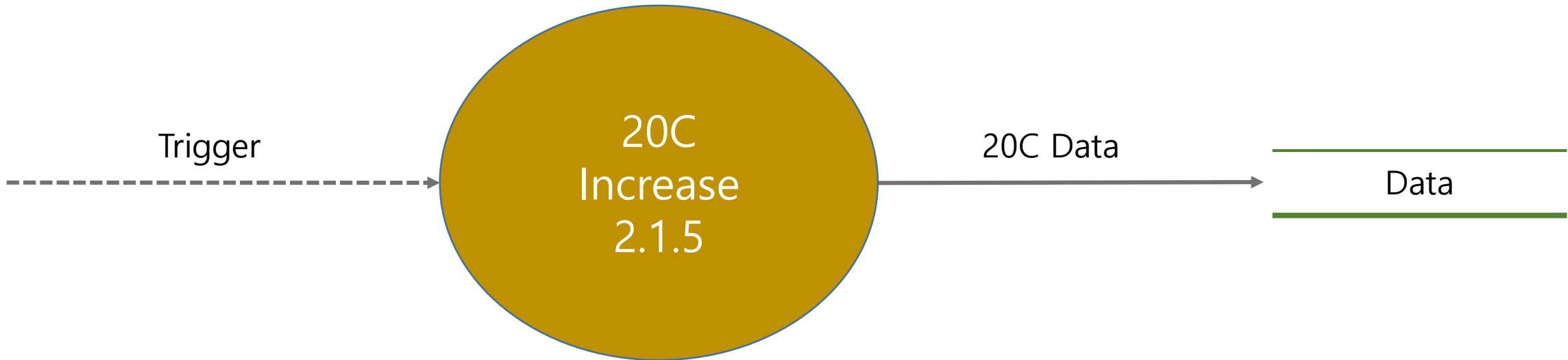
Reference No.	2.1.3
Name	10C Increase
Input	Trigger
Output	10C Data (Integer)
Process Description	10C Increase 에 Trigger가 들어 오면 10도 라는 온도 데이터를 Data에 있는 stemp에 누적 시킨다.

DFD Level 3 – 2.1 Process Specification



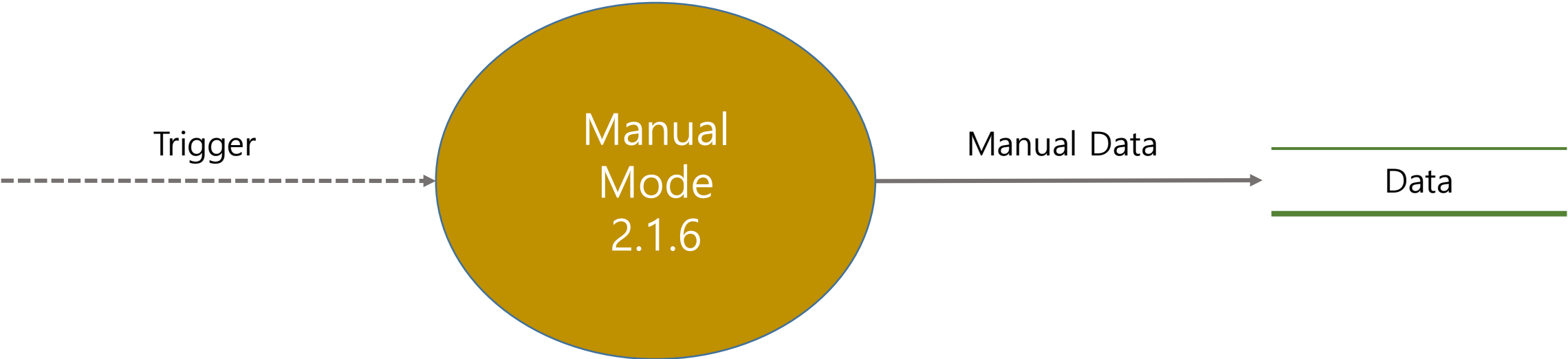
Reference No.	2.1.4
Name	30sec Increase
Input	Trigger
Output	30sec Data (Integer)
Process Description	30sec Increase 에 Trigger가 들어 오면 30초 라는 시간 데이터를 Data에 있는 stime에 누적 시킨다.

DFD Level 3 – 2.1 Process Specification

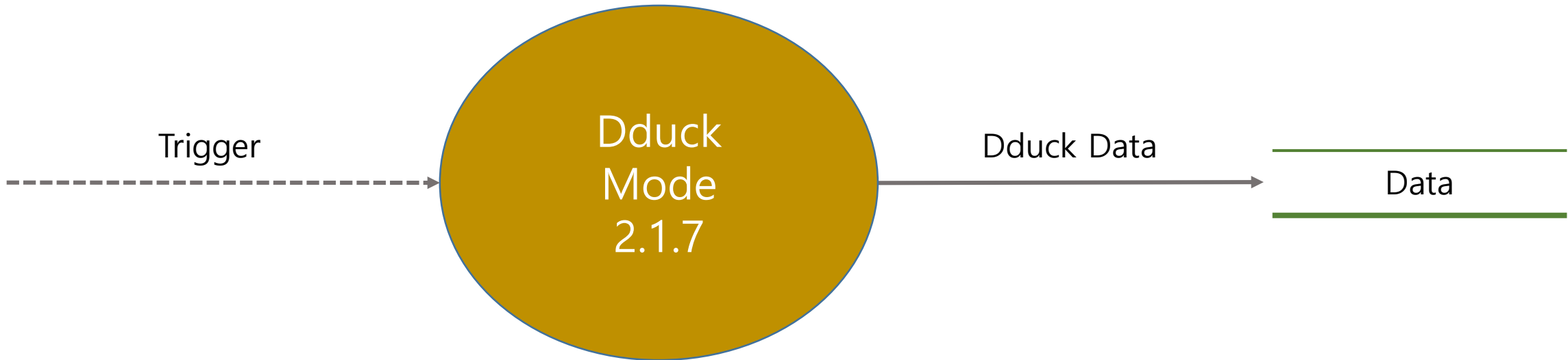


Reference No.	2.1.5
Name	20C Increase
Input	Trigger
Output	20C Data (Integer)
Process Description	20C Increase 에 Trigger가 들어 오면 20도 라는 온도 데이터를 Data에 있는 stemp에 누적 시킨다.

DFD Level 3 – 2.1 Process Specification

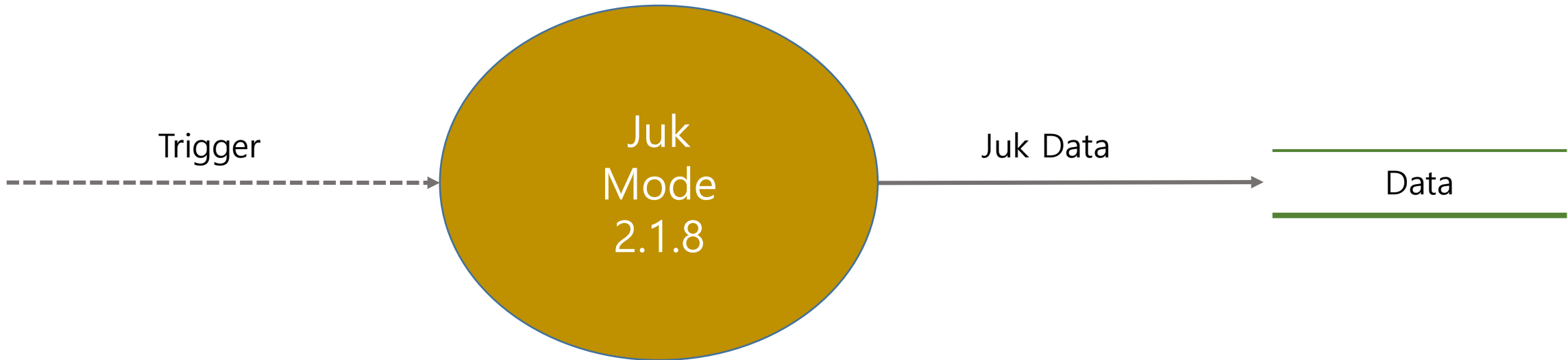


Reference No.	2.1.6
Name	Manual Mode
Input	Trigger
Output	Manual Data (Char)
Process Description	Manual Mode에 Trigger가 들어 오면 Manual Mode 라는 문자 데이터를 Data에 있는 cmode에 변경 시킨다.



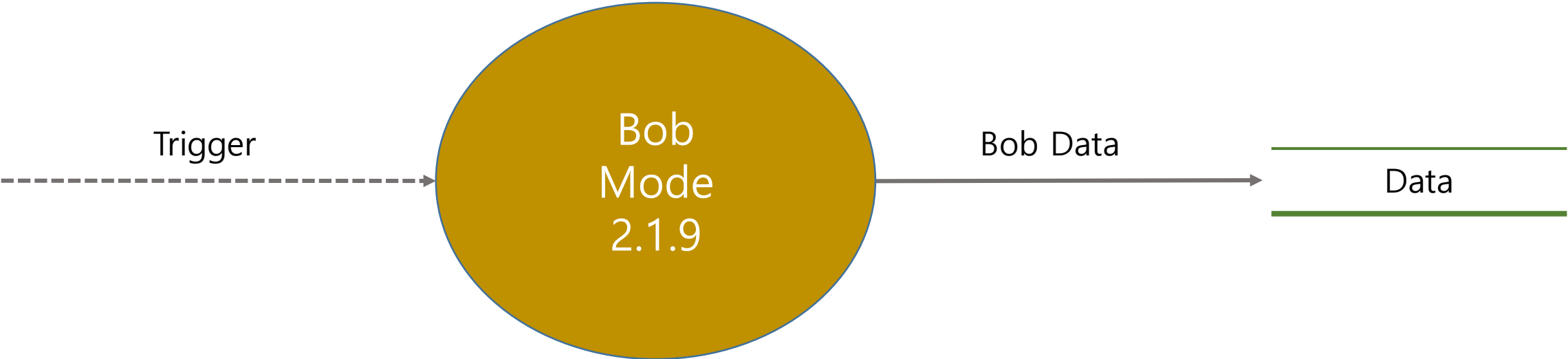
Reference No.	2.1.7
Name	Dduck Mode
Input	Trigger
Output	Dduck Data
Process Description	Dduck Mode에 Trigger가 들어 오면 Dduck Mode 라는 시간 데이터와 문자 데이터를 Data에 있는 stime과 cmode에 변경 시킨다.

DFD Level 3 – 2.1 Process Specification



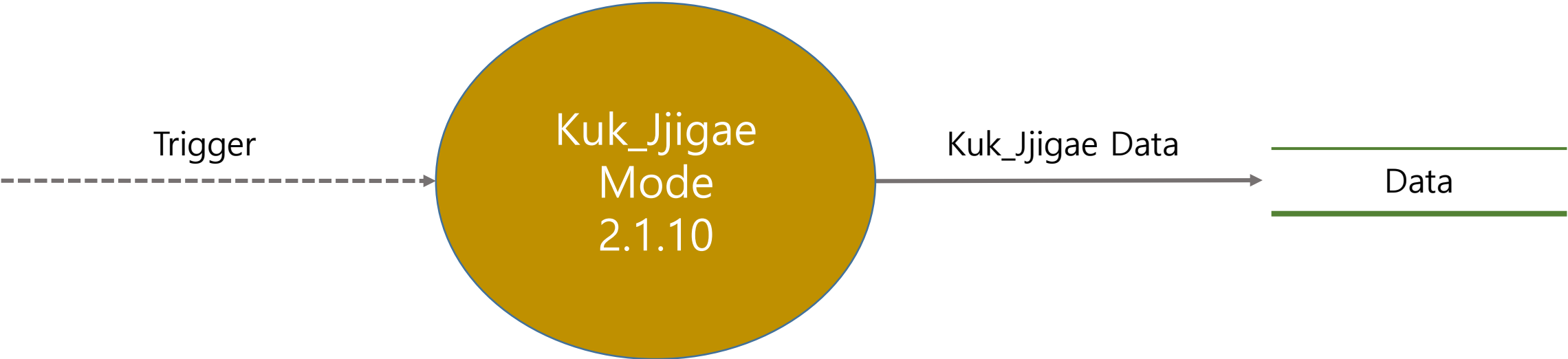
Reference No.	2.1.8
Name	Juk Mode
Input	Trigger
Output	Juk Data
Process Description	Juk Mode에 Trigger가 들어 오면 Juk Mode 라는 시간 데이터와 문자 데이터를 Data에 있는 stime과 cmode에 변경 시킨다.

DFD Level 3 – 2.1 Process Specification



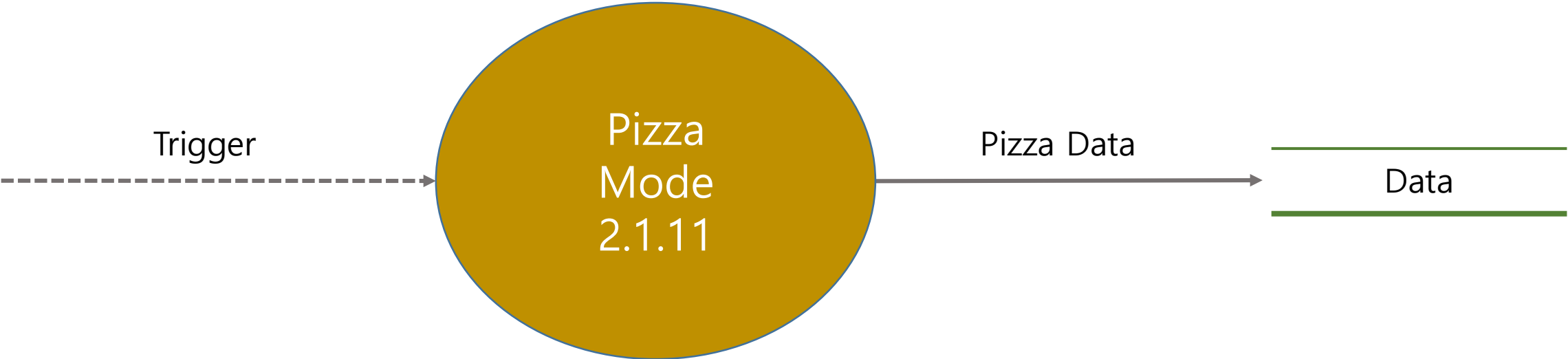
Reference No.	2.1.9
Name	Bob Mode
Input	Trigger
Output	Bob Data
Process Description	Bob Mode에 Trigger가 들어 오면 Bob Mode 라는 시간 데이터와 문자 데이터를 Data 에 있는 stime과 cmode에 변경 시킨다.

DFD Level 3 – 2.1 Process Specification



Reference No.	2.1.10
Name	Kuk_Jjigae Mode
Input	Trigger
Output	Kuk_Jjigae Data
Process Description	Kuk_Jjigae Mode에 Trigger가 들어 오면 Kuk_Jjigae Mode 라는 시간 데이터와 문자 데이터를 Data에 있는 stime과 cmode에 변경 시킨다.

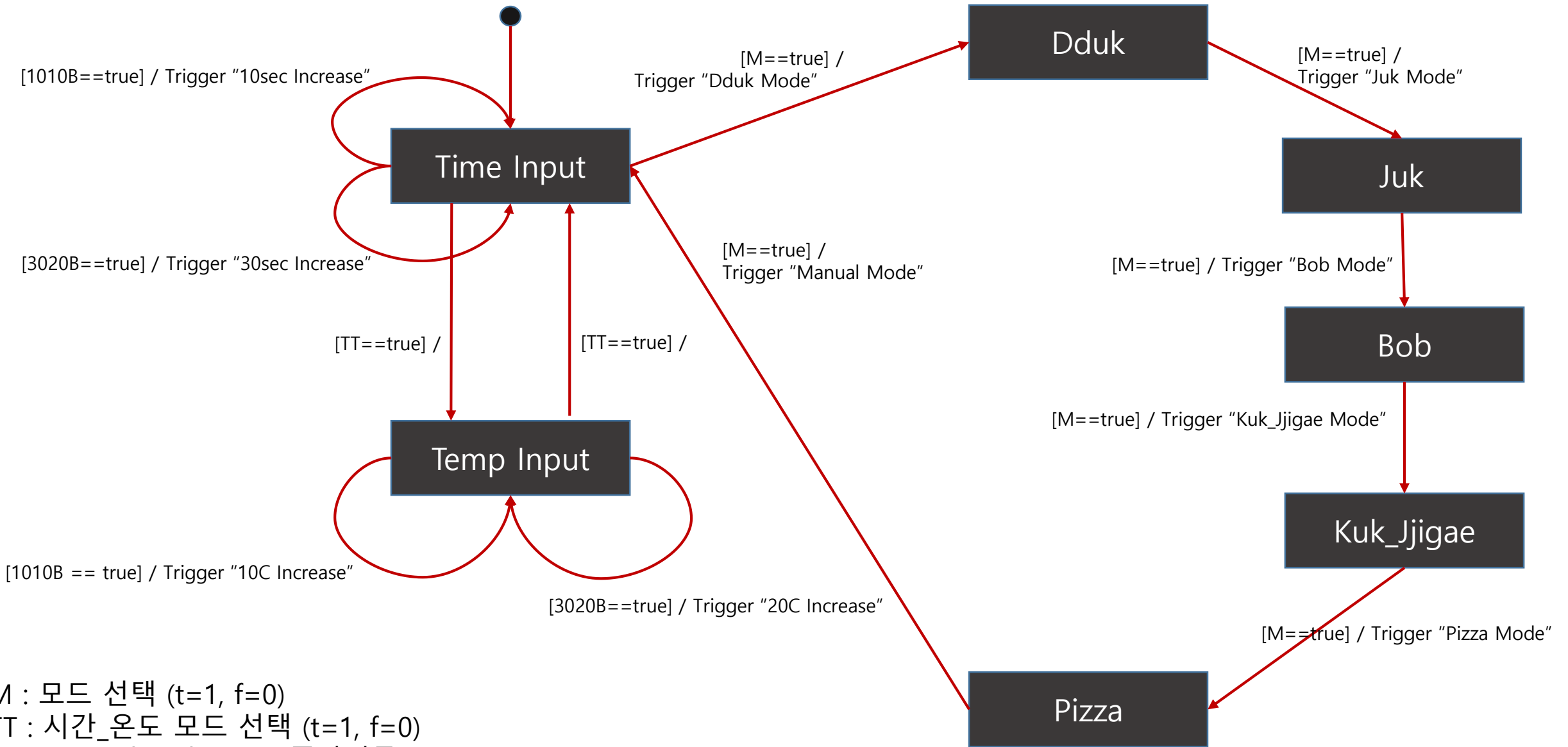
DFD Level 3 – 2.1 Process Specification



Reference No.	2.1.11
Name	Pizza Mode
Input	Trigger
Output	Pizza Data
Process Description	Pizza Mode에 Trigger가 들어 오면 Pizza Mode 라는 시간 데이터와 문자 데이터를 Data에 있는 stime과 cmode에 변경 시킨다.

DFD Level 3 - 2.1 (1/3) [State Transition Diagram for Button Controller 2.1.1]

38



M : 모드 선택 (t=1, f=0)

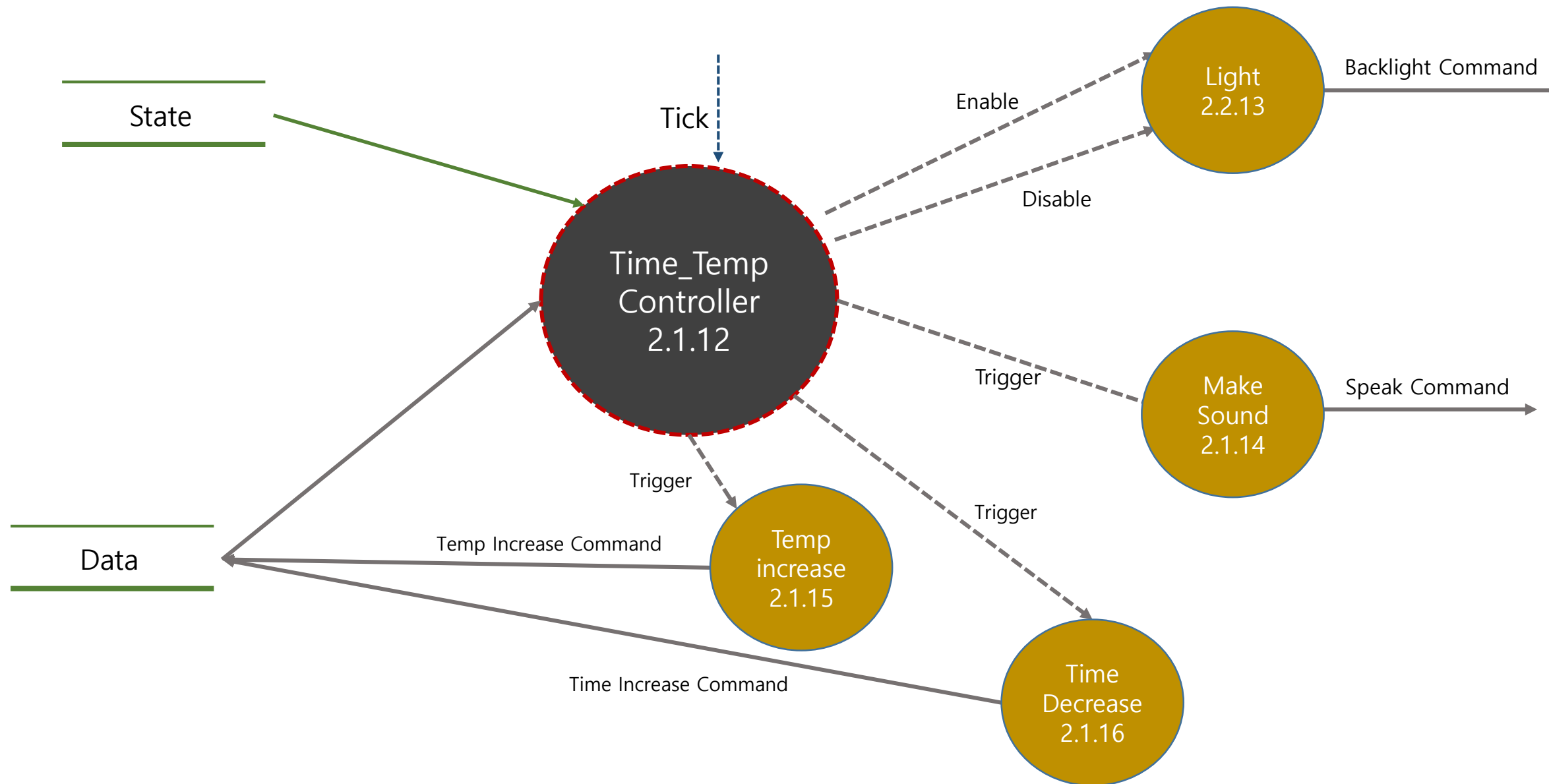
TT : 시간_온도 모드 선택 (t=1, f=0)

1010B : 10초 또는 10도 증가버튼 (t=1, f=0)

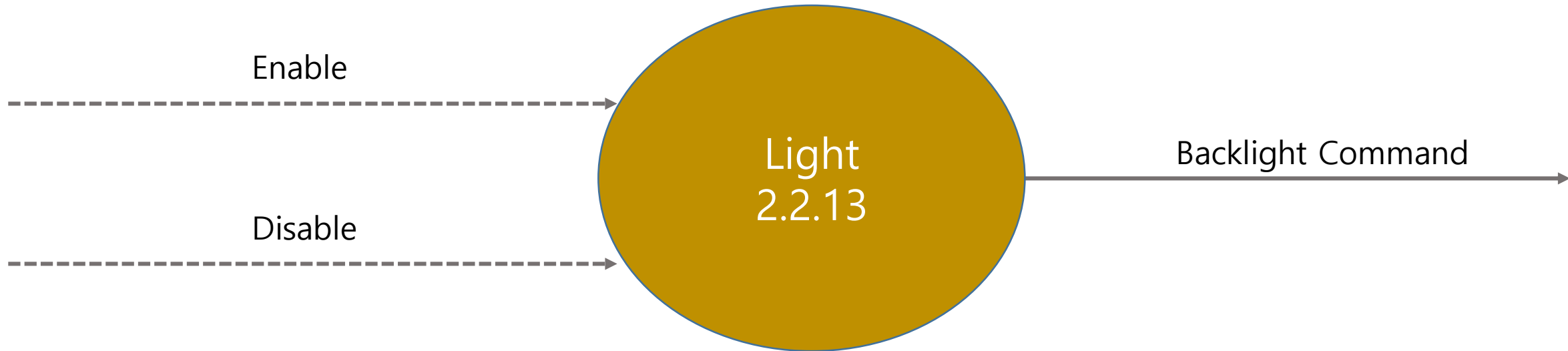
3020B : 30초 또는 20도 증가버튼 (t=1, f=0)

DFD Level 3 - 2.1 (2/3)

39



DFD Level 3 – 2.1 Process Specification

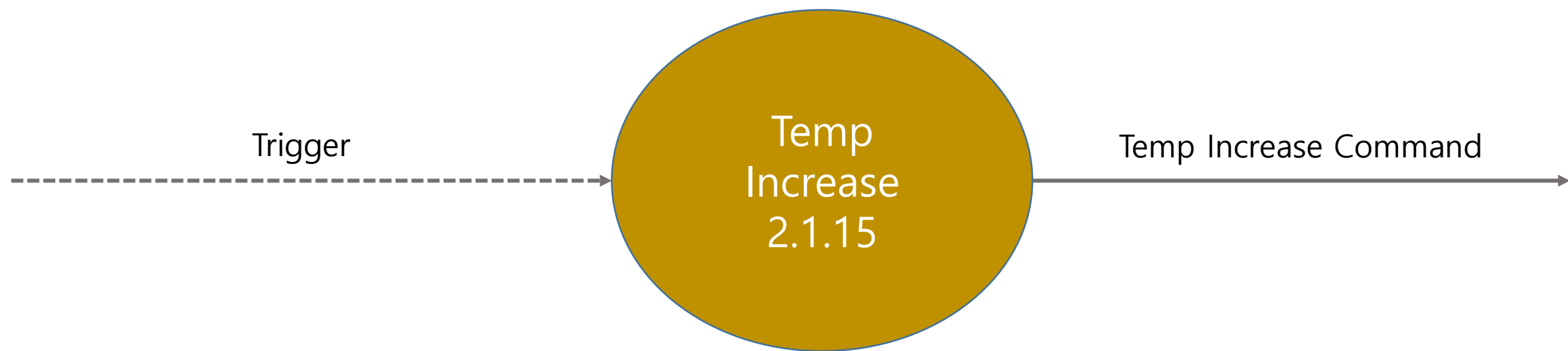


Reference No.	2.1.13
Name	Light
Input	Enable, Disable
Output	Backlight Command (Bool)
Process Description	Light에 Enable이 들어오면 Lighter Interface에게 Backlight를 켜라는 Backlight Command를 보낸다. Disable이 들어오면 끄라는 Backlight Command를 보낸다.

DFD Level 3 – 2.1 Process Specification



Reference No.	2.1.14
Name	Make Sound
Input	Trigger
Output	Speak Command (Bool)
Process Description	Make Sound에 Trigger가 들어오면 Speaker Interface에게 Speak하라는 Speak Command를 보낸다. Disable이 들어오면 끄라는 Speak Command를 보낸다.

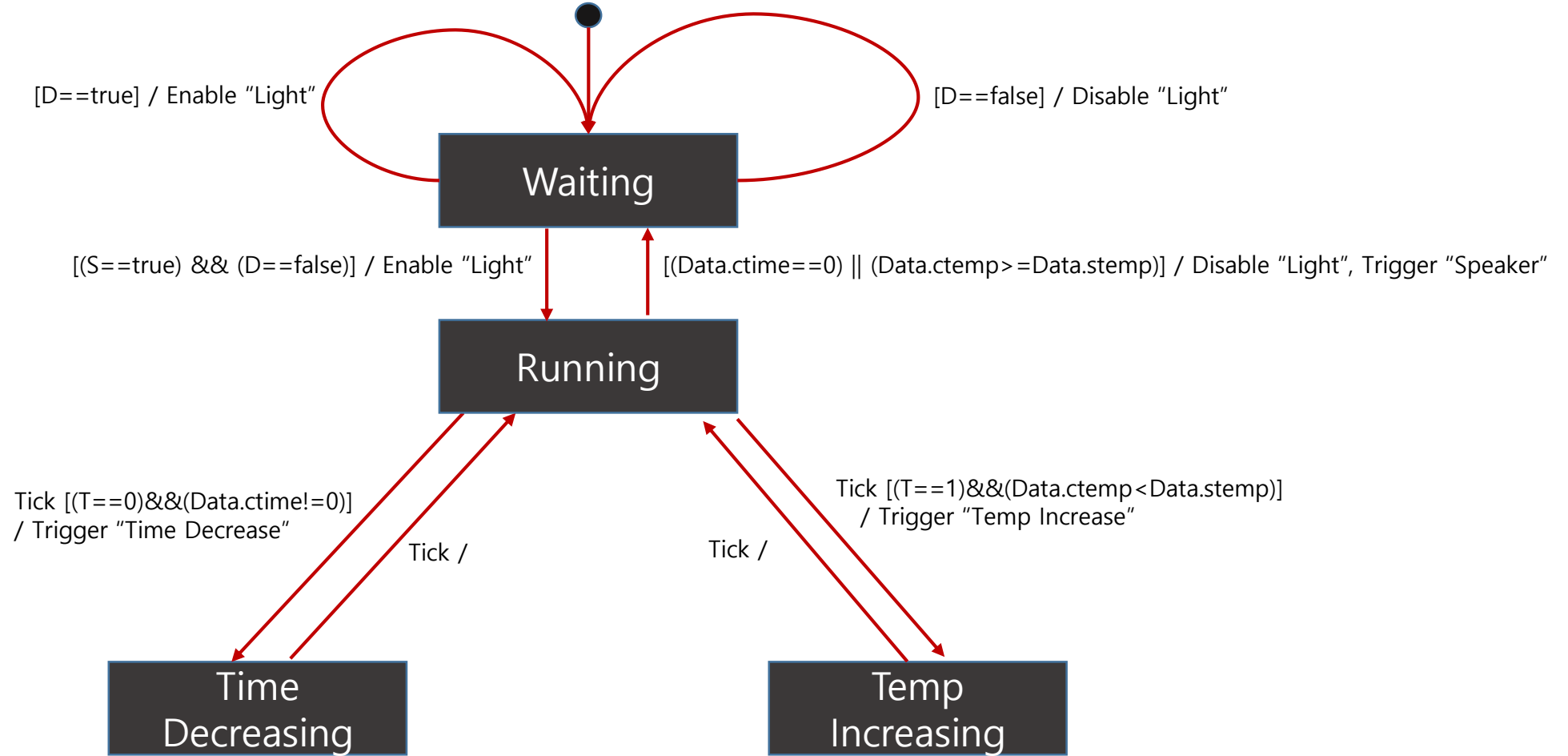


Reference No.	2.1.15
Name	Temp Increase
Input	Trigger
Output	Temp Increase Command
Process Description	Temp Increase에 Trigger가 들어오면 Data에 ctemp를 증가시킨다.



Reference No.	2.1.16
Name	Time Decrease
Input	Trigger
Output	Time Increase Command
Process Description	Time Increase에 Trigger가 들어오면 Data에 ctime을 증가시킨다.

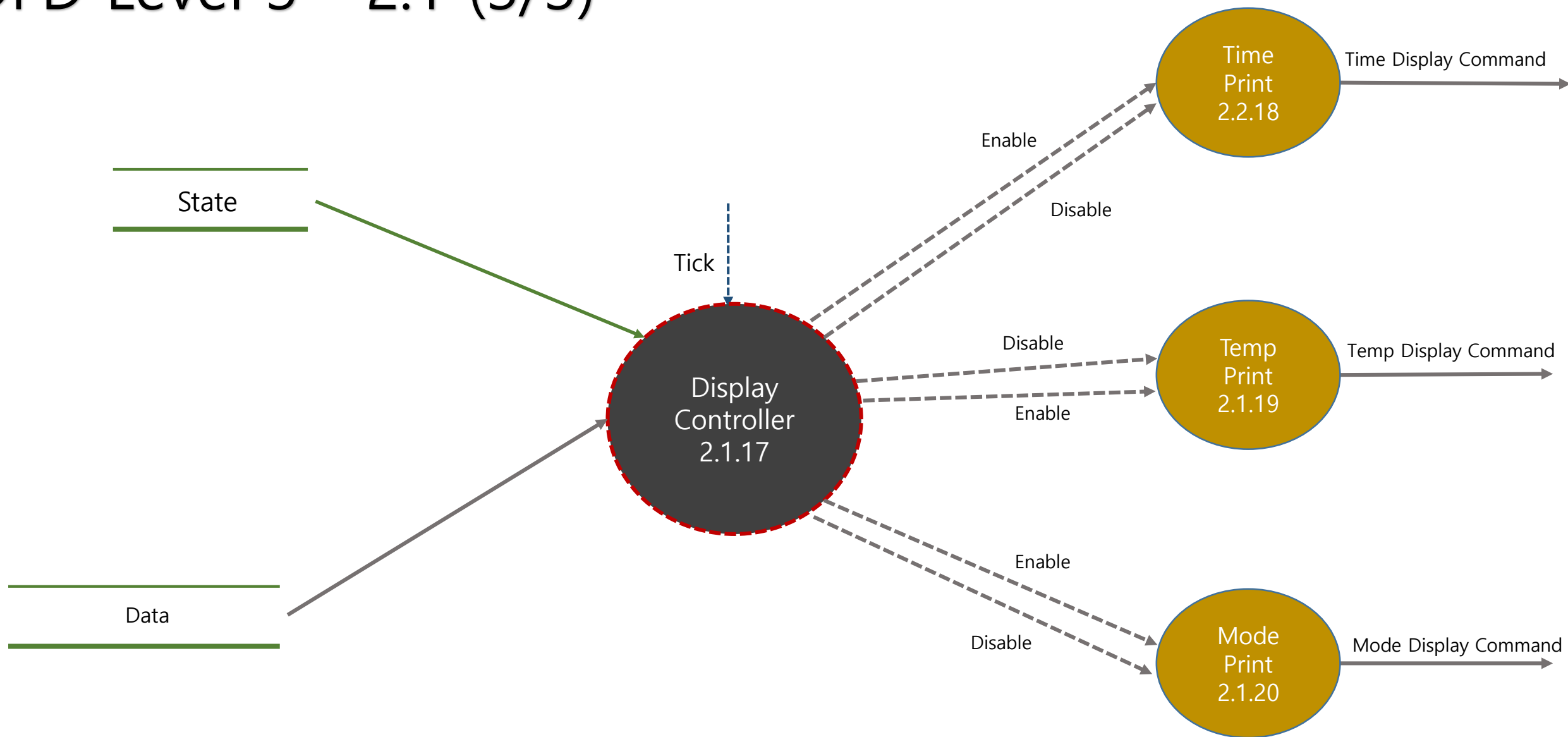
DFD Level 3 - 2.1 (2/3) [State Transition Diagram for Time_Temp Controller 2.1.11] ⁴⁴



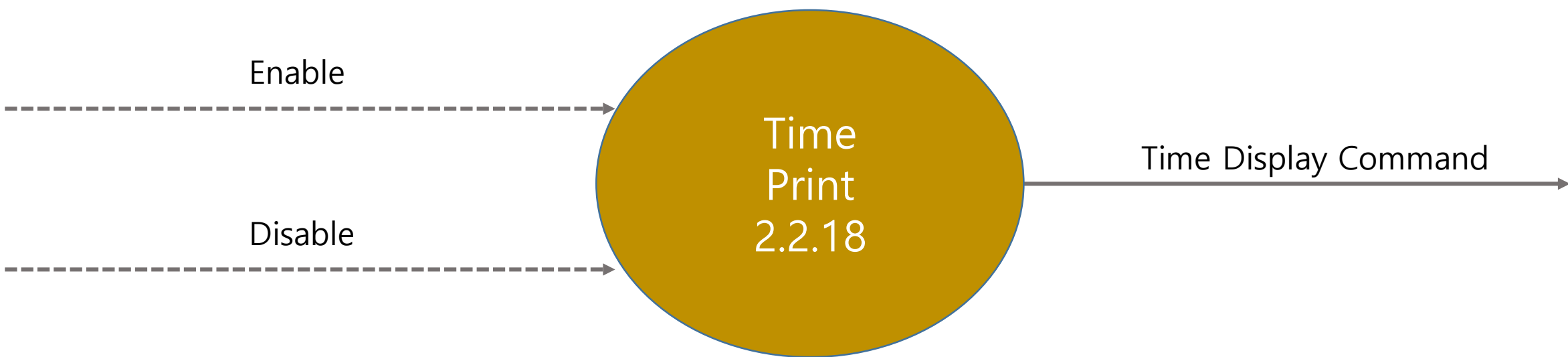
S : 조리 시작_취소 (t = 1, f = 0), D : 문상태 (t = 1, f = 0), T : Time / Temp (0 = 시간, 1 = 온도)
Data.ctime : Data에 저장되어있는 현재 시간, Data.ctemp : Data에 저장되어있는 현재 온도
Data.stime : Data에 저장되어있는 설정된 시간, Data.stemp : Data에 저장되어있는 설정된 온도

DFD Level 3 - 2.1 (3/3)

45

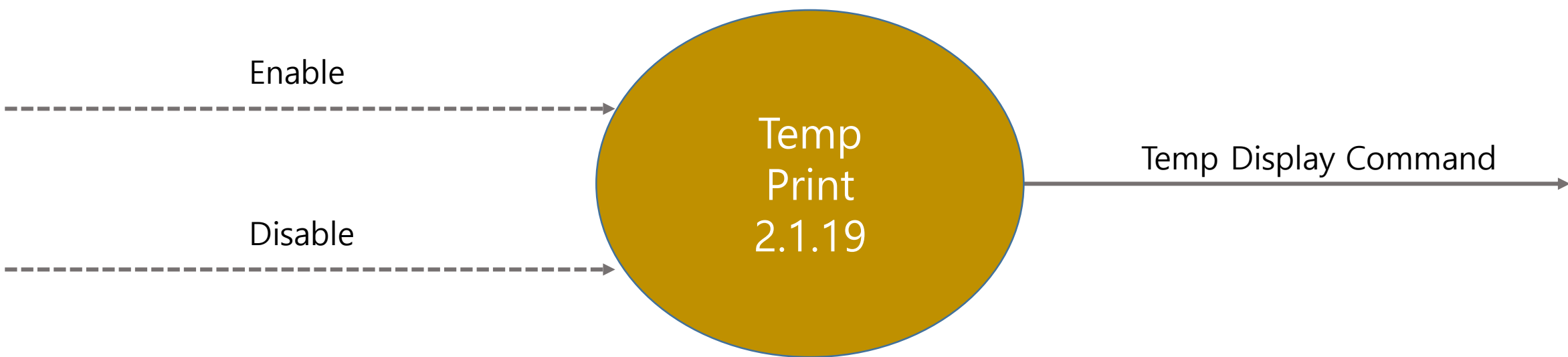


DFD Level 3 – 2.1 Process Specification



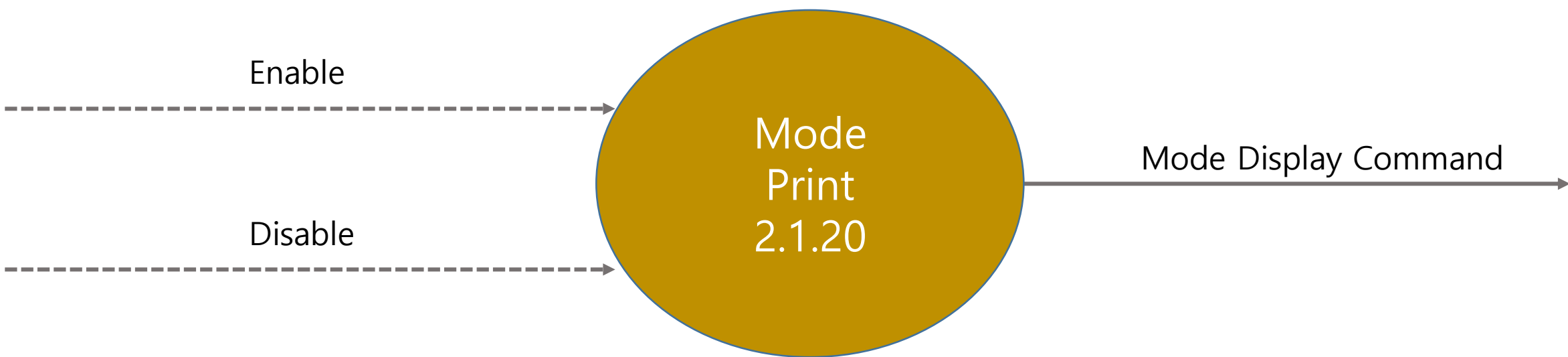
Reference No.	2.1.18
Name	Time Print
Input	Enable, Disable
Output	Time Display Command (Bool)
Process Description	Time Print에 Enable이 들어오면 Time_Temp Display Interface에게 Time을 Display하라는 Time Display Command를 보낸다. Disable이 들어오면 끄라는 Time Display Command를 보낸다.

DFD Level 3 – 2.1 Process Specification

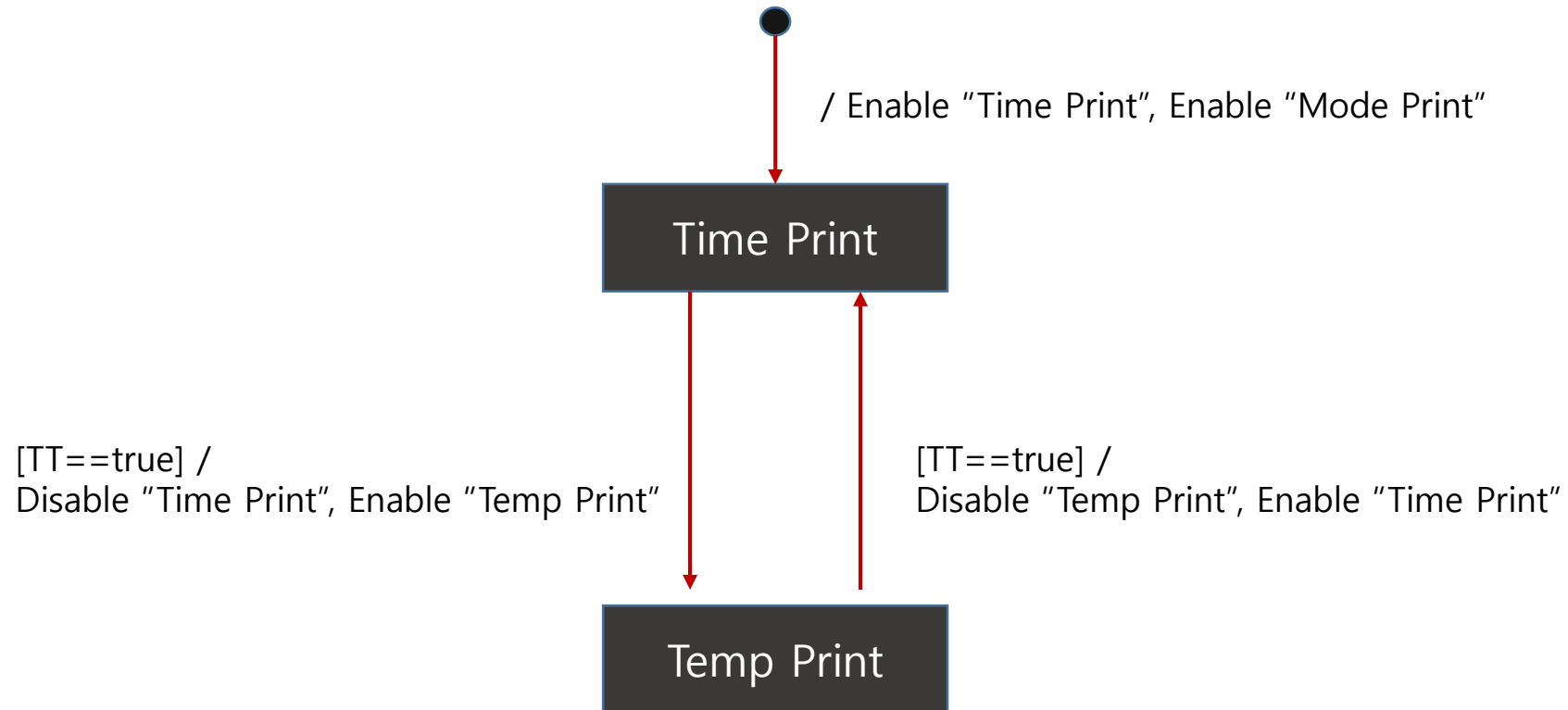


Reference No.	2.1.19
Name	Temp Print
Input	Enable, Disable
Output	Time Display Command (Bool)
Process Description	Temp Print에 Enable이 들어오면 Time_Temp Display Interface에게 Temp을 Display하라는 Temp Display Command를 보낸다. Disable이 들어오면 끄라는 Temp Display Command를 보낸다.

DFD Level 3 – 2.1 Process Specification

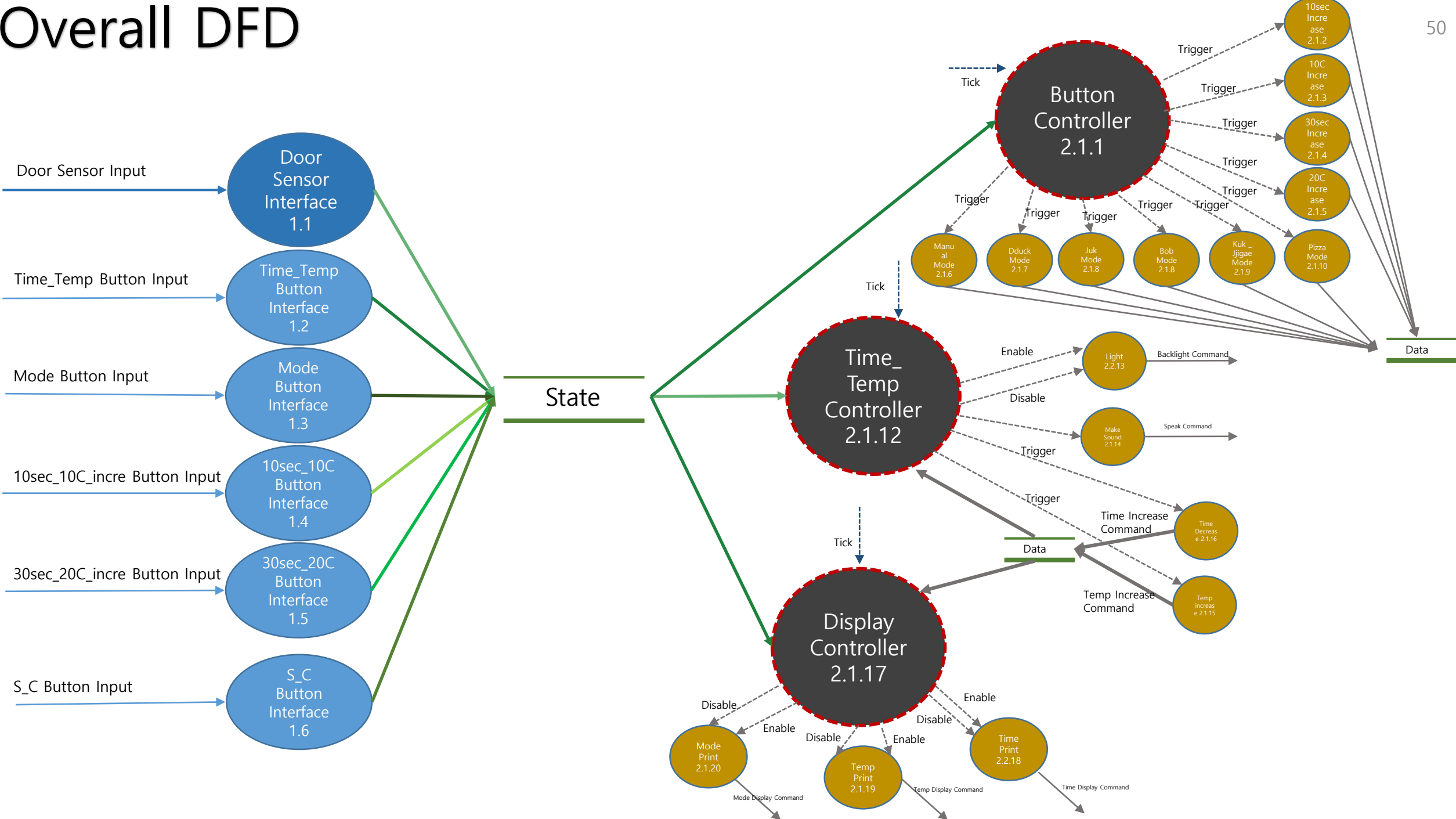


Reference No.	2.1.20
Name	Mode Print
Input	Enable, Disable
Output	Mode Display Command (Bool)
Process Description	Time Print에 Enable이 들어오면 Mode Display Interface에게 Mode를 Display하라는 Mode Display Command를 보낸다. Disable이 들어오면 끄라는 Mode Display Command를 보낸다.



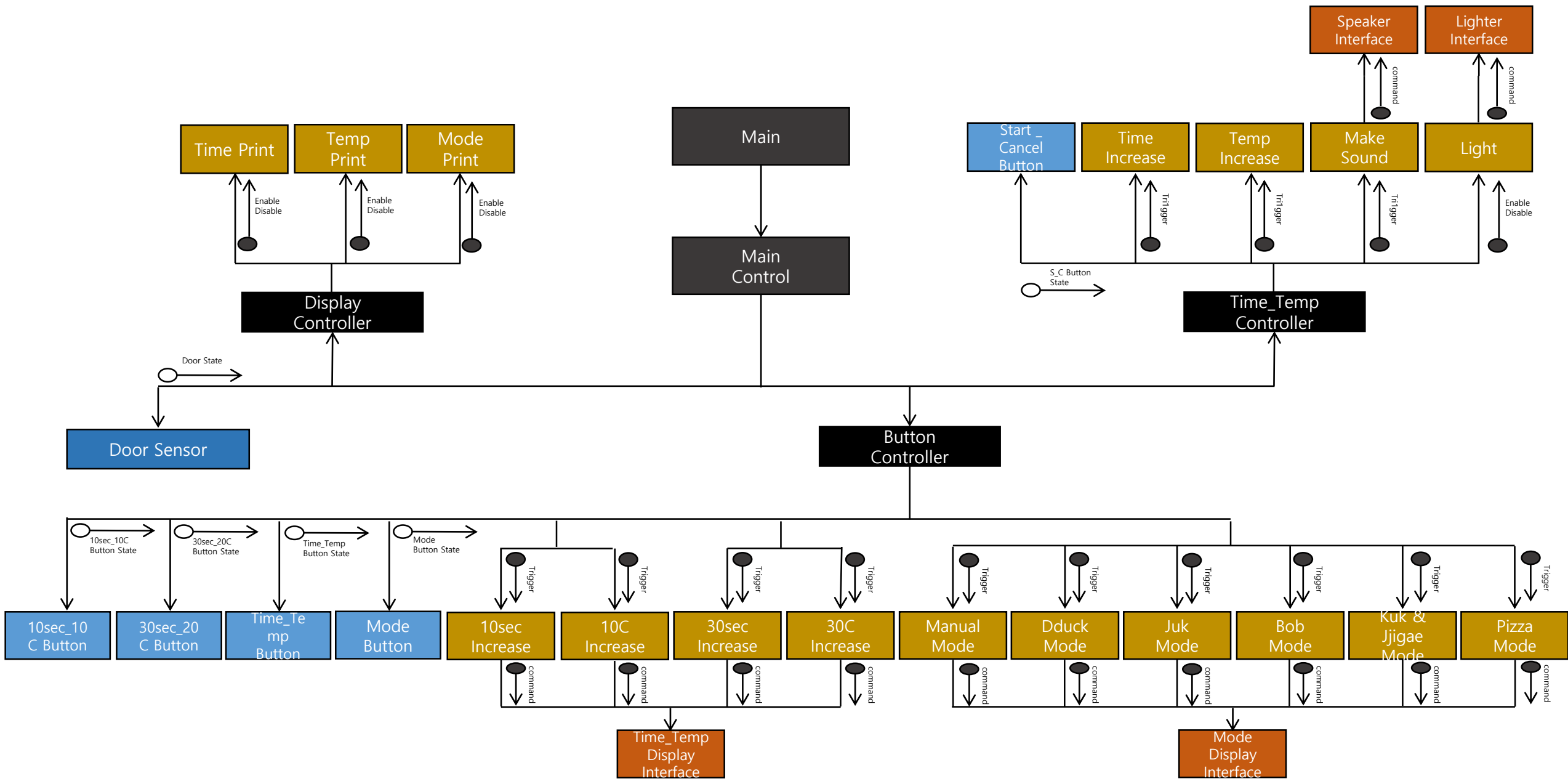
TT : 시간온도모드 (t = 1, f = 0)

Overall DFD



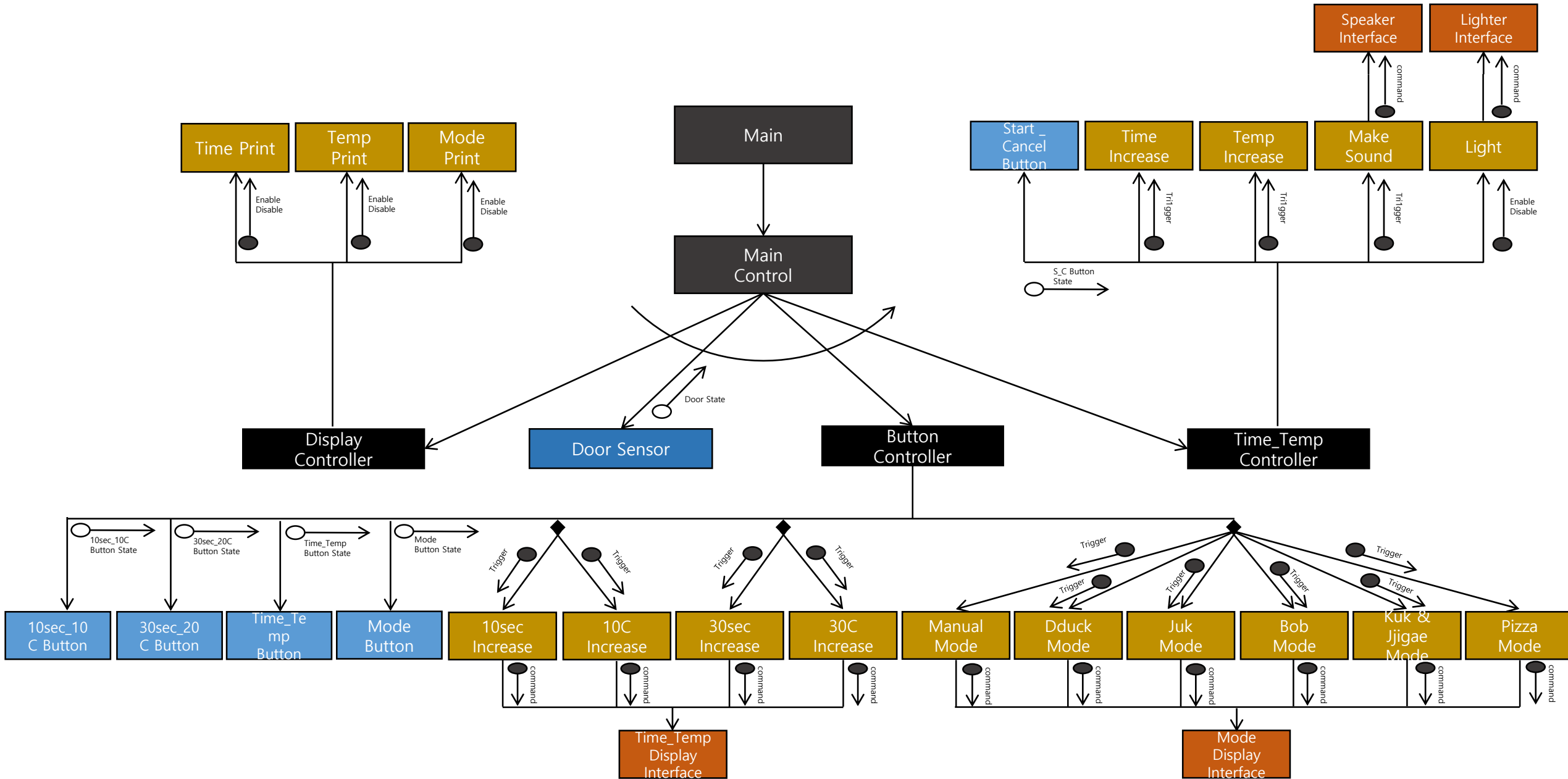
Structured Design

Structured Charts – Microwave Oven System (Basic)



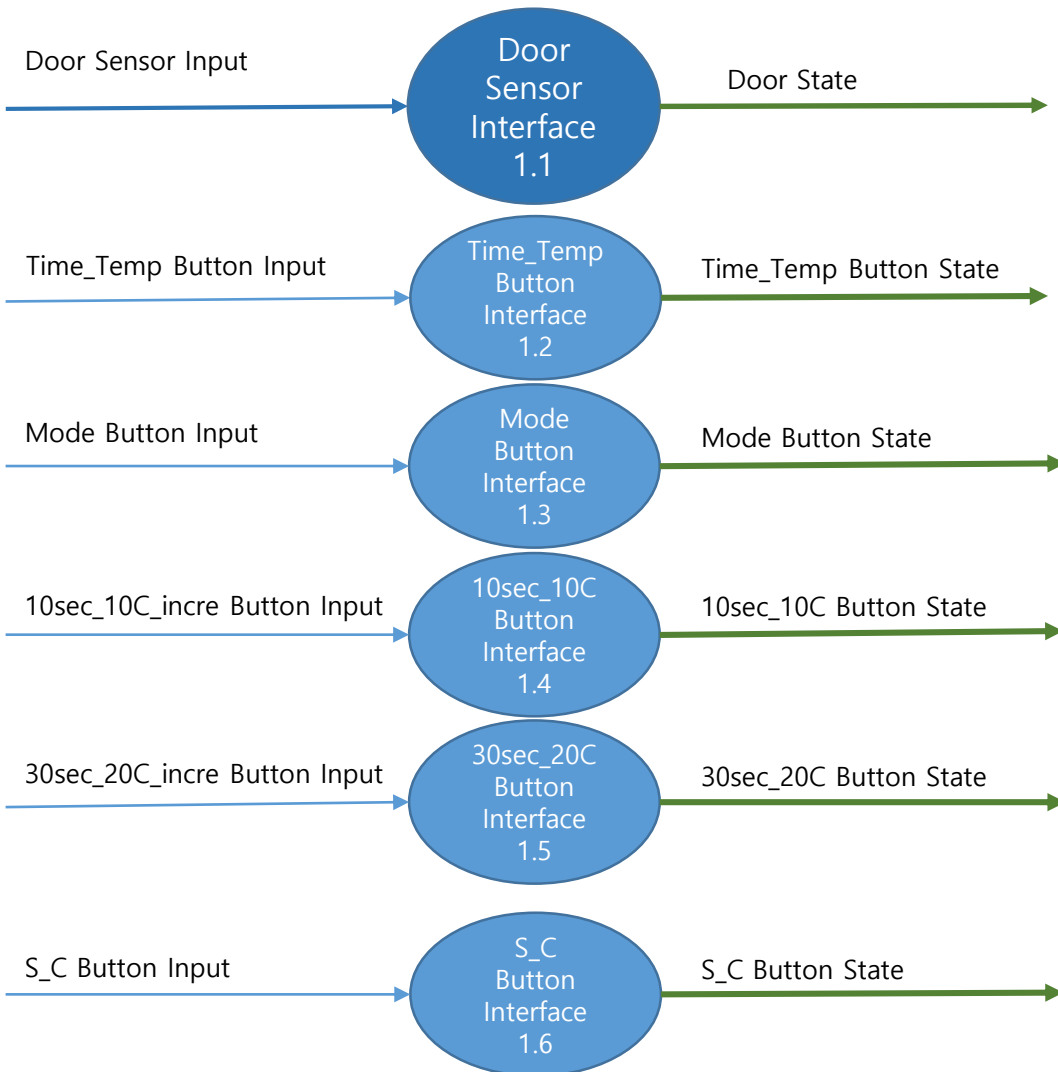
Structured Charts – Microwave Oven System (Advanced)

53



Process & Code

Prime Process – Microwave Oven System



```
void DoorSenSorInterface (Bool DoorSensorInput){  
s.DoorState = DoorSensorInput;  
}
```

```
void Time_TempInterface (Bool Time_TempInput){  
s.Time_TempButtonState = Time_TempInput;  
}
```

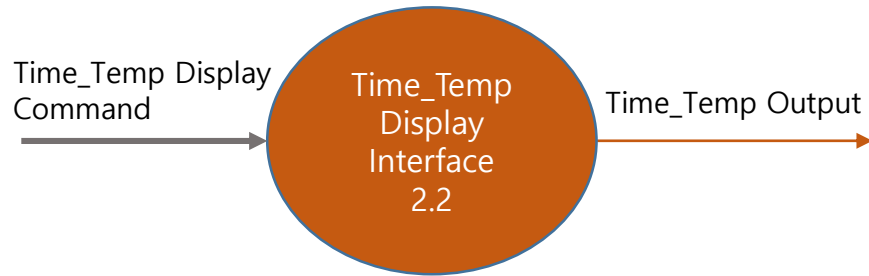
```
void ModeButtonInterface (Bool ModeButtonInput){  
s.ModeButtonState = ModeButtonInput;  
}
```

```
void _10sec_10CButtonInterface (Bool  
_10sec_10C_increButtonInput){  
s._10sec_10CButtonState = _10sec_10C_increButtonInput;  
}
```

```
void _30sec_20CButtonInterface (Bool  
_30sec_20C_increButtonInput){  
s._30sec_20CButtonState = _30sec_20C_increButtonInput;  
}
```

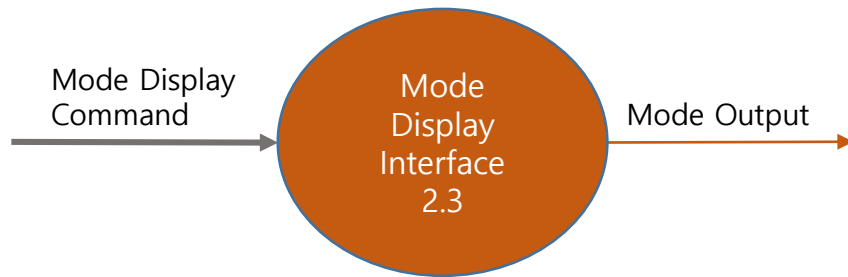
```
void S_CButtonInterface (Bool S_CButtonInput){  
s.S_CButtonState = S_CButtonInput;  
}
```

Prime Process – Microwave Oven System



```
void Time_TempDisplayInterface(Bool TimeDisplayCommand,
Bool TempDisplayCommand) {
    system("clear");
    if ( (T == TimeDisplayCommand) && (F ==
TempDisplayCommand)) {
        if ( d.ctime == -1 ) {
            printf("%d:%02dWt", (d.stime)/60, (d.stime)%60);
        }
        else {
            printf("%d:%02dWt", (d.ctime)/60, (d.ctime)%60);
        }
    }
    else {
        if ( d.ctemp == -1 ) {
            printf("%dCWt", d.stemp);
        }
        else {
            printf("%dCWt", d.ctemp);
        }
    }
}
```


Prime Process – Microwave Oven System



```
void ModeDisplayInterface(Bool ModeDisplayCommand) {
```

```
    if (ModeDisplayCommand == T) {  
        printf("Mode : ");
```

```
        switch (d.i_cmode) {
```

```
        case 0:
```

```
            printf("Manual\n");
```

```
            break;
```

```
        case 1:
```

```
            printf("Dduck\n");
```

```
            break;
```

```
        case 2:
```

```
            printf("Juk\n");
```

```
            break;
```

```
        case 3:
```

```
            printf("Bob\n");
```

```
            break;
```

```
        case 4:
```

```
            printf("Kuk_Jjigae\n");
```

```
            break;
```

```
        case 5:
```

```
            printf("Pizza\n");
```

```
            break;
```

```
        default :
```

```
            printf("Error, in switch of ModeDisplayInterface.\n");
```

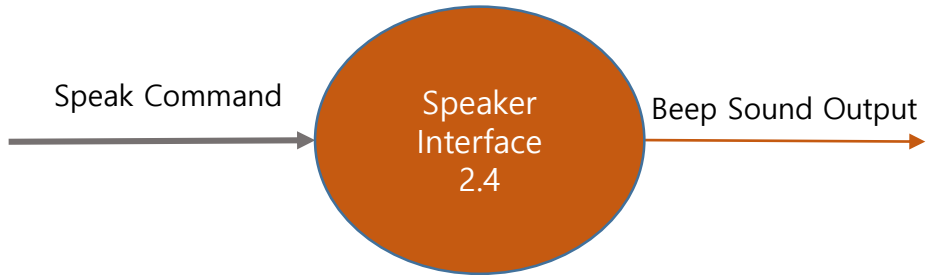
```
    }
```

```
}
```

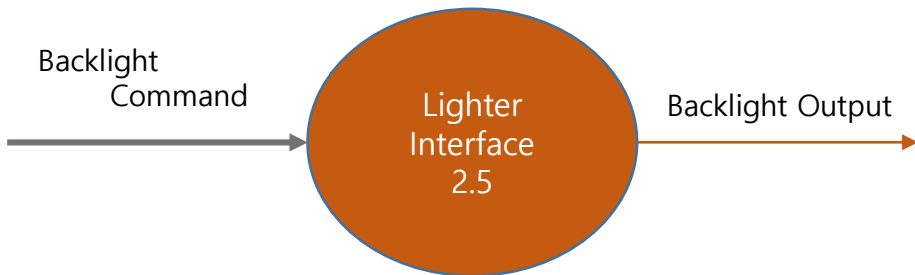
```
    printf("\n");
```

```
}
```

Prime Process – Microwave Oven System



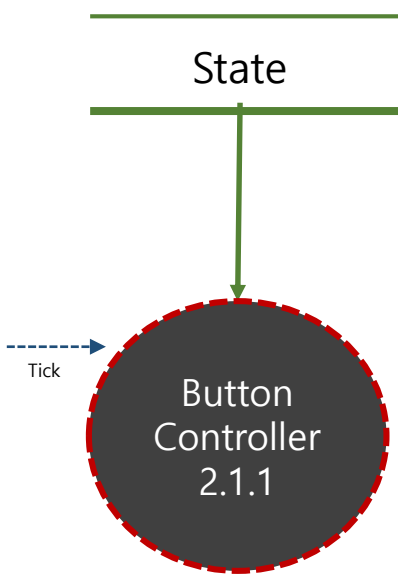
```
void LighterInterface(Bool BacklightCommand) {  
    if ( (T == BacklightCommand) ) {  
        printf("%c[1;33m",27);  
    }  
    else {  
        printf("%c[0m", 27);  
    }  
}
```



```
void SpeakerInterface (Bool SpeakCommand) {  
    int i;  
  
    if (T == SpeakCommand) {  
        printf("\Wa");  
        sleep(1);  
        printf("\Wa");  
        sleep(1);  
        printf("\Wa");  
        sleep(1);  
        SC=1;  
    }  
}
```

Process & Code (Button Controller)

59



```
void buttoncontroller (struct State *){
```

```
static int state = 1;// odd : Time, even : Temp
```

```
static int mode = 0;// Mode
```

```
Bool trigger = F;
```

```
if(s->Time_TempButtonState == T){
```

```
state++;
```

```
ABLE++; // sum
```

```
s->Time_TempButtonState = F;
```

```
if ( (state % 2) != 0 ) {
```

```
d.stemp = 20;
```

```
}
```

```
else {
```

```
d.stime = 0;
```

```
}
```

```
}
```

```
///
```

```
if( ( s->ModeButtonState == T ) && ( (state % 2) != 0 ) ){
```

```
mode++;
```

```
mode %= 6;
```

```
if ( mode == 0 ) {
```

```
d.stime = 0;
```

```
}
```

```
s->ModeButtonState = F;
```

```
}
```

```
///
```

```
if (mode == 0) {
```

```
trigger = T;
```

```
ManualMode(trigger);
```

```
if( ( s->_10sec_10CButtonState == T ) && ( (state % 2) != 0 ) ) {
```

```
s->_10sec_10CButtonState = F;
```

```
trigger = T;
```

```
_10secIncrease(trigger);
```

```
}
```

```
if( ( s->_30sec_20CButtonState == T ) && ( (state % 2) != 0 ) ){
```

```
s->_30sec_20CButtonState = F;
```

```
trigger = T;
```

```
_30secIncrease(trigger);
```

```
}
```

```
////
```

```
if( ( s->_10sec_10CButtonState == T ) && ( (state % 2) == 0 ) ) {
```

```
s->_10sec_10CButtonState = F;
```

```
trigger = T;
```

```
_10CIncrease(trigger);
```

```
}
```

```
if( ( s->_30sec_20CButtonState == T ) && ( (state % 2) == 0 ) ){
```

```
s->_30sec_20CButtonState = F;
```

```
trigger = T;
```

```
_20CIncrease(trigger);
```

```
}
```

```
}
```

```
else {
```

```
switch(mode){
```

```
case 1: // Dduck Mode
```

```
trigger = T;
```

```
DduckMode(trigger);
```

```
break;
```

```
case 2: // Juk Mode
```

```
trigger = T;
```

```
JukMode(trigger);
```

```
break;
```

```
case 3: // Bob Mode
```

```
trigger = T;
```

```
BobMode(trigger);
```

```
break;
```

```
case 4: // Kuk_Jjigae Mode
```

```
trigger = T;
```

```
Kuk_JjigaeMode(trigger);
```

```
break;
```

```
case 5: // Pizza Mode
```

```
trigger = T;
```

```
PizzaMode(trigger);
```

```
break;
```

```
default :
```

```
printf("Error, in switch of
```

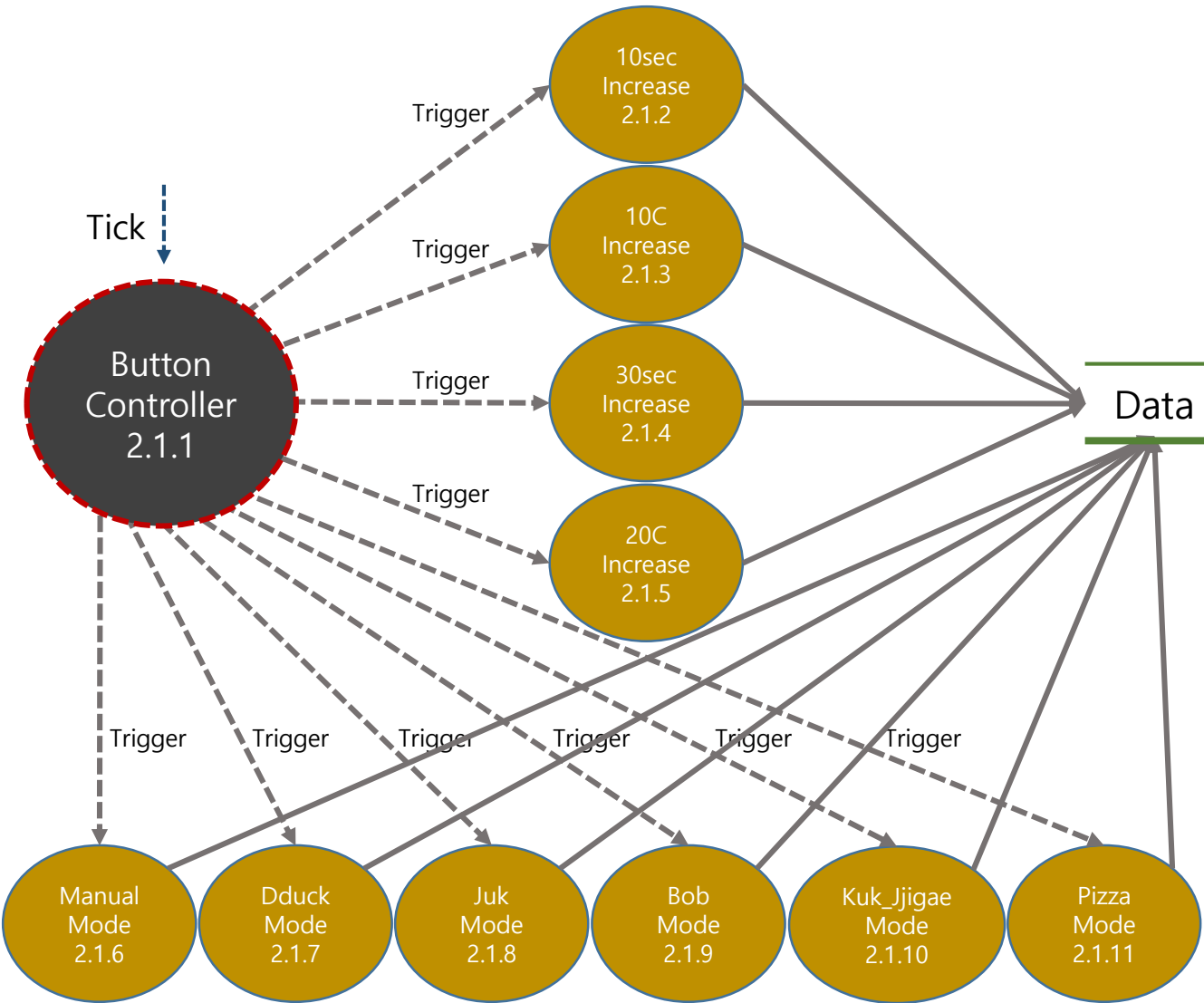
```
burroncontroller.\n");
```

```
break;
```

```
} // switch
```

```
}
```

Process & Code

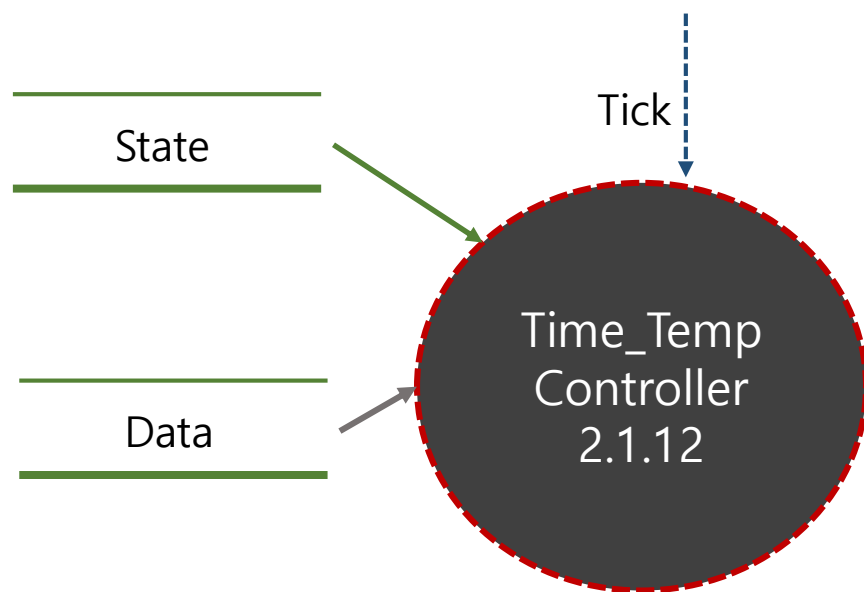


```
void _10secIncrease (Bool trigger) {  
    if(T == trigger) {  
        d.stime += 10;  
        d.stime %= 600;  
    }  
}  
  
void _30secIncrease (Bool trigger) {  
    if(T == trigger) {  
        d.stime += 30;  
        d.stime %= 600;  
    }  
}  
  
void _10CIncrease (Bool trigger) {  
    if(T == trigger) {  
        d.stemp += 10;  
        if(d.stemp > 90) {  
            d.stemp = (d.stemp % 70);  
        }  
    }  
}  
  
void _20CIncrease (Bool trigger) {  
    if(T == trigger) {  
        d.stemp += 20;  
        if(d.stemp > 90) {  
            d.stemp = (d.stemp % 70);  
        }  
    }  
}
```

```
void ManualMode (Bool trigger) {  
    if(T == trigger) {  
        d.i_cmode = 0;  
        d.c_cmode = "Manual";  
    }  
}  
  
void DduckMode (Bool trigger) {  
    if(T == trigger) {  
        d.stime = 60;  
        d.i_cmode = 1;  
        d.c_cmode = "Dduck";  
    }  
}  
  
void JukMode (Bool trigger) {  
    if(T == trigger) {  
        d.stime = 90;  
        d.i_cmode = 2;  
        d.c_cmode = "Juk";  
    }  
}  
  
void BobMode (Bool trigger) {  
    if(T == trigger) {  
        d.stime = 120;  
        d.i_cmode = 3;  
        d.c_cmode = "Bob";  
    }  
}  
  
void Kuk_JjigaeMode (Bool trigger) {  
    if(T == trigger) {  
        d.stime = 300;  
        d.i_cmode = 4;  
        d.c_cmode = "Kuk_Jjigae";  
    }  
}  
  
void PizzaMode (Bool trigger) {  
    if(T == trigger) {  
        d.stime = 120;  
        d.i_cmode = 5;  
        d.c_cmode = "Pizza";  
    }  
}
```

Process & Code (Time_Temp Controller)

61



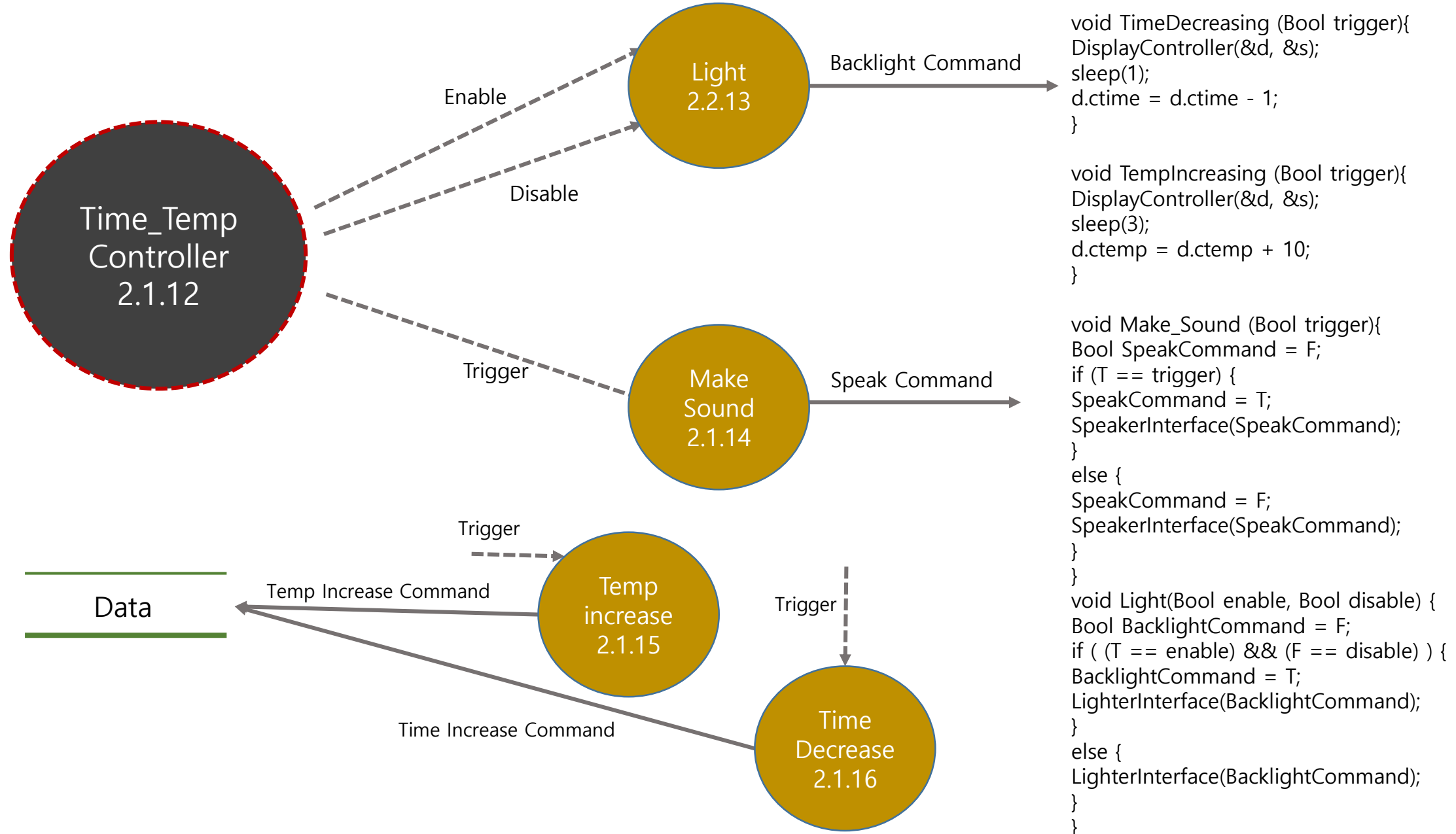
```
void Time_TempController(struct Data *d, struct State *s){
```

```
    Bool trigger = F;  
    Bool enable = F;  
    Bool disable = F;
```

```
    if ( s->DoorState == T ) {  
        s->DoorState = F;  
        DOOR++;  
        if ( (DOOR % 2) == 0 ) {  
            enable = T;  
        }  
        else {  
            disable = T;  
        }  
        Light(enable, disable);  
    }  
    else {
```

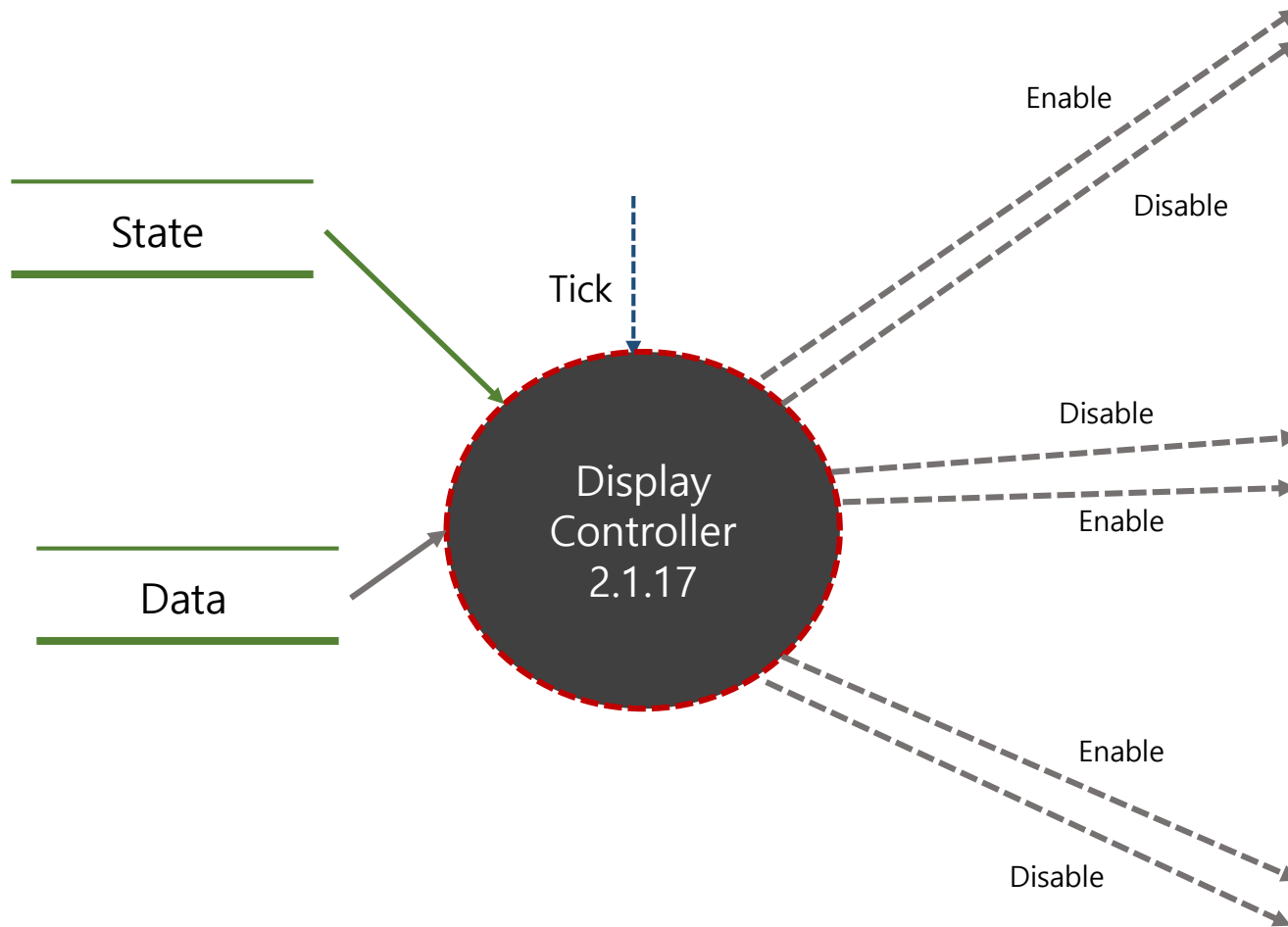
```
        if ( (s->S_CButtonState == T) &&  
            ((DOOR%2) != 0) ) {  
            if( (ABLE % 2) != 0 ){ // When ModeState is  
                Time.  
                if (d->ctime == 0 ) {  
                    trigger = T;  
                    disable = T;  
                    Light(enable, disable);  
                    Make_Sound(trigger);  
                }  
                else {  
                    trigger = T;  
                    enable = T;  
                    Light(enable, disable);  
                    TimeDecreasing(trigger);  
                }  
            }  
            else if( (ABLE % 2) == 0 ){// When  
                ModeState is Temp.  
                if (d->ctemp >= d->stemp ) {  
                    trigger = T;  
                    disable = T;  
                    Light(enable, disable);  
                    Make_Sound(trigger);  
                }  
                else {  
                    trigger = T;  
                    enable = T;  
                    Light(enable, disable);  
                    TempIncreasing(trigger);  
                }  
            }  
        } // if  
        else {  
            s->S_CButtonState = F;  
        }  
    } // else  
}
```

Process & Code



Process & Code (Display Controller)

63



```
void DisplayController(struct Data *d, struct State *s) {  
    Bool enable = F;  
    Bool disable = F;  
  
    if ( (ABLE % 2) != 0 ) {  
        enable = T;  
        TimePrint(enable, disable);  
        ModePrint(enable, disable);  
    }  
    else {  
        enable = T;  
        TempPrint(enable, disable);  
        ModePrint(enable, disable);  
    }  
}
```

State

```
typedef struct State {  
    Bool DoorState;  
    Bool Time_TempButtonState;  
    Bool ModeButtonState;  
    Bool _10sec_10CButtonState;  
    Bool _30sec_20CButtonState;  
    Bool S_CButtonState;  
};
```

Data

```
typedef struct Data {  
    int stime;  
    int stemp;  
    int ctime;  
    int ctemp;  
    //  
    int i_cmode;  
    char *c_cmode;  
};
```

Microwave Oven System System & Unit Testing #1

T3

이경수,한득환,김대희,신민용

System Test Feature

Identifier	Feature	valid / Invalid value
MOW.STC.000	시간확인	시작과 동시에 화면 표시, 최대값 넘어 갈 때 시간 연산, A, B 버튼 입력 시 시간 변경 확인
MOW.STC.001	온도확인	시간/온도 전환 확인, 최대값 넘어 갈 때 온도 연산, A, B 버튼 입력 시 온도 변경 확인
MOW.STC.002	모드확인	D 버튼 입력 시 설정된 모드와 시간 확인 모드변경 루프 정상작동 확인
MOW.STC.003	각 상황에서 Backlight 실행	F 버튼 입력 시 점/소등 확인 F 버튼 입력 후 다른 버튼 입력 시 점/소등 유지여부 확인
MOW.STC.004	조리시작	초기 미 설정 상태에서의 시작 가능여부 확인 문의 개폐에 따른 시작 확인 및 조리 시작 시 조명 점등 확인
MOW.STC.005	조리중	조리 중 조명 상태 점등 확인 Tick에 따른 시간 감소와 온도 증가 확인
MOW.STC.006	조리중 버튼입력	조리 중 다른 버튼 입력 시 동작불가 확인 조리 중 Cancel 동작 확인
MOW.STC.007	조리종료	조리 이후 종료 시 비프음 출력 확인 조명상태 확인
MOW.STC.008	RESET	조리 종료 시 초기화 모드 변경 확인 모드변경 시 이전 설정 값 초기화 확인

Test Case / Result

Test case identifier	Input specification	Output specification
MOW.STC.000.000	프로그램 실행	00:00 Mode : Manual
MOW.STC.000.001	00:00 Mode : Manual, a	00:10 Mode : Manual
MOW.STC.000.002	00:10 Mode : Manual, b	00:40 Mode : Manual
MOW.STC.000.003	09:50 Mode : Manual, a	00:00 Mode : Manual
MOW.STC.000.004	09:50 Mode : Manual, b	00:20 Mode : Manual
MOW.STC.000.005	20C Mode : Manual, c	00:00 Mode : Manual
MOW.STC.001.000	00:00 Mode : Manual, c	20C Mode : Manual
MOW.STC.001.001	20C Mode : Manual, a	30C Mode : Manual
MOW.STC.001.002	30C Mode : Manual, b	50C Mode : Manual
MOW.STC.001.003	90C Mode : Manual, a	30C Mode : Manual
MOW.STC.001.004	80C Mode : Manual, b	30C Mode : Manual

Identifier	PASS/FAIL
MOW.STC.000.000	PASS
MOW.STC.000.001	PASS
MOW.STC.000.002	PASS
MOW.STC.000.003	PASS
MOW.STC.000.004	PASS
MOW.STC.000.005	PASS
MOW.STC.001.000	PASS
MOW.STC.001.001	PASS
MOW.STC.001.002	PASS
MOW.STC.001.003	PASS
MOW.STC.001.004	PASS

Test case identifier	Input specification	Output specification
MOW.STC.002.000	00:00 Mode : Manual, d	01:00 Mode : Dduck
MOW.STC.002.001	01:00 Mode : Dduck, d	01:30 Mode : Juk
MOW.STC.002.002	01:30 Mode : Juk, d	02:00 Mode : Bob
MOW.STC.002.003	02:00 Mode : Bob, d	05:00 Mode : Kuk_Jigae
MOW.STC.002.004	05:00 Mode : Kuk_Jigae, d	02:00 Mode : Pizza
MOW.STC.002.005	02:00 Mode : Pizza, d	00:00 Mode : Manual
MOW.STC.002.006	20C Mode : Manual, d	20C Mode : Manual
MOW.STC.002.007	01:30 Mode : Juk, a	01:30 Mode : Juk
MOW.STC.002.008	01:30 Mode : Juk, b	01:30 Mode : Juk
MOW.STC.002.009	01:30 Mode : Juk, c	01:30 Mode : Juk

Identifier	PASS/FAIL
MOW.STC.002.000	PASS
MOW.STC.002.001	PASS
MOW.STC.002.002	PASS
MOW.STC.002.003	PASS
MOW.STC.002.004	PASS
MOW.STC.002.005	PASS
MOW.STC.002.006	PASS
MOW.STC.002.007	PASS
MOW.STC.002.008	PASS
MOW.STC.002.009	PASS

Test case identifier	Input specification	Output specification
MOW.STC.003.000	00:00 Mode : Manual, f	00:00 Mode : Manual
MOW.STC.003.001	20C Mode : Manual, f	20C Mode : Manual
MOW.STC.003.002	00:00 Mode : Manual, f	00:00 Mode : Manual
MOW.STC.003.003	20C Mode : Manual, f	20C Mode : Manual
MOW.STC.003.004	02:00 Mode : Bob, f	02:00 Mode : Bob
MOW.STC.003.005	02:00 Mode : Bob, f	02:00 Mode : Bob
MOW.STC.003.006	00:00 Mode : Manual, a	00:10 Mode : Manual
MOW.STC.003.007	00:00 Mode : Manual, b	00:30 Mode : Manual
MOW.STC.003.008	20C Mode : Manual, a	30C Mode : Manual
MOW.STC.003.009	20C Mode : Manual, b	40C Mode : Manual
MOW.STC.003.010	00:00 Mode : Manual, c	20C Mode : Manual
MOW.STC.003.011	20C Mode : Manual, c	00:00 Mode : Manual
MOW.STC.003.012	00:00 Mode : Manual, d	01:00 Mode : Dduck

Identifier	PASS/FAIL
MOW.STC.003.000	PASS
MOW.STC.003.001	PASS
MOW.STC.003.002	PASS
MOW.STC.003.003	PASS
MOW.STC.003.004	PASS
MOW.STC.003.005	PASS
MOW.STC.003.006	PASS
MOW.STC.003.007	PASS
MOW.STC.003.008	PASS
MOW.STC.003.009	PASS
MOW.STC.003.010	PASS
MOW.STC.003.011	PASS
MOW.STC.003.012	PASS

Test case identifier	Input specification	Output specification
MOW.STC.004.000	00:00 Mode : Manual, e	00:00 Mode : Manual
MOW.STC.004.001	00:10 Mode : Manual, e, (문열림), 1초후	00:10 Mode : Manual
MOW.STC.004.002	40C Mode : Manual, e, (문열림), 3초후	40C Mode : Manual
MOW.STC.004.003	02:00 Mode : Bob, e, (문열림), 1초후	02:00 Mode : Bob
MOW.STC.004.004	00:10 Mode : Manual, e	00:10 Mode : Manual
MOW.STC.004.005	40C Mode : Manual, e	20C Mode : Manual
MOW.STC.004.006	02:00 Mode : Bob, e	02:00 Mode : Bob
MOW.STC.005.000	00:10 Mode : Manual , (작동중), 1초후	00:09 Mode : Manual
MOW.STC.005.001	00:10 Mode : Manual , (작동중), 10초후	00:00 Mode : Manual
MOW.STC.005.002	20C Mode : Manual, (작동중), 3초후, 80C 설정	30C Mode : Manual
MOW.STC.005.003	20C Mode : Manual, (작동중), 18초후	80C Mode : Manual
MOW.STC.005.004	02:00 Mode : Bob, (작동중), 1초후	01:59 Mode : Bob
MOW.STC.005.005	02:00 Mode : Bob, (작동중), 2분후	00:00 Mode : Bob

Identifier	PASS/FAIL
MOW.STC.004.000	FAIL (바로 비프음)
MOW.STC.004.001	FAIL (00:00 Mode : Manual)
MOW.STC.004.002	FAIL (20C Mode : Manual)
MOW.STC.004.003	FAIL (00:00 Mode : Bob)
MOW.STC.004.004	PASS
MOW.STC.004.005	PASS
MOW.STC.004.006	PASS
MOW.STC.005.000	PASS
MOW.STC.005.001	PASS
MOW.STC.005.002	PASS
MOW.STC.005.003	PASS
MOW.STC.005.004	PASS
MOW.STC.005.005	PASS

Test case identifier	Input specification	Output specification
MOW.STC.006.000	00:09 Mode : Manual (작동중), a	00:09 Mode : Manual (작동중)
MOW.STC.006.001	00:09 Mode : Manual (작동중), b	00:09 Mode : Manual (작동중)
MOW.STC.006.002	00:09 Mode : Manual (작동중), c	00:09 Mode : Manual (작동중)
MOW.STC.006.003	00:09 Mode : Manual (작동중), d	00:09 Mode : Manual (작동중)
MOW.STC.006.004	00:09 Mode : Manual (작동중), e	00:00 Mode : Manual
MOW.STC.006.005	00:09 Mode : Manual (작동중), f	00:09 Mode : Manual (작동중)
MOW.STC.006.006	30C Mode : Manual (작동중), a	30C Mode : Manual (작동중)
MOW.STC.006.007	30C Mode : Manual (작동중), b	30C Mode : Manual (작동중)
MOW.STC.006.008	30C Mode : Manual (작동중), c	30C Mode : Manual (작동중)
MOW.STC.006.009	30C Mode : Manual (작동중), d	30C Mode : Manual (작동중)
MOW.STC.006.010	30C Mode : Manual (작동중), e	20C Mode : Manual
MOW.STC.006.011	30C Mode : Manual (작동중), f	30C Mode : Manual (작동중)
MOW.STC.006.012	01:59 Mode : Bob (작동중), a	01:59 Mode : Bob (작동중)
MOW.STC.006.013	01:59 Mode : Bob (작동중), b	01:59 Mode : Bob (작동중)
MOW.STC.006.014	01:59 Mode : Bob (작동중), c	01:59 Mode : Bob (작동중)
MOW.STC.006.015	01:59 Mode : Bob (작동중), d	01:59 Mode : Bob (작동중)
MOW.STC.006.016	01:59 Mode : Bob (작동중), e	00:00 Mode : Manual
MOW.STC.006.017	01:59 Mode : Bob (작동중), f	01:59 Mode : Bob (작동중)

Identifier	PASS/FAIL
MOW.STC.006.000	PASS
MOW.STC.006.001	PASS
MOW.STC.006.002	PASS
MOW.STC.006.003	PASS
MOW.STC.006.004	PASS
MOW.STC.006.005	PASS
MOW.STC.006.006	PASS
MOW.STC.006.007	PASS
MOW.STC.006.008	PASS
MOW.STC.006.009	PASS
MOW.STC.006.010	PASS
MOW.STC.006.011	PASS
MOW.STC.006.012	PASS
MOW.STC.006.013	PASS
MOW.STC.006.014	PASS
MOW.STC.006.015	PASS
MOW.STC.006.016	FAIL (00:00 Mode : Bob)
MOW.STC.006.017	PASS

Test case identifier	Input specification	Output specification
MOW.STC.007.000	00:01 Mode : Manual , (작동중), 1초후	00:00 Mode : Manual (3초간 비프음)
MOW.STC.007.001	00:00 Mode : Manual (3초간 비프음 후)	00:00 Mode : Manual
MOW.STC.007.002	30C Mode : Manual, (작동중), 3초후, 40C 설정	40C Mode : Manual (3초간 비프음)
MOW.STC.007.003	40C Mode : Manual (3초간 비프음 후)	20C Mode : Manual
MOW.STC.007.004	00:01 Mode : Bob, (작동중), 1초후	00:00 Mode : Bob (3초간 비프음)
MOW.STC.007.005	00:00 Mode : Bob (3초간 비프음 후)	00:00 Mode : Manual
MOW.STC.008.000	00:00 Mode : Manual (3초간 비프음 후)	00:00 Mode : Manual
MOW.STC.008.001	40C Mode : Manual (3초간 비프음 후)	20C Mode : Manual
MOW.STC.008.002	00:00 Mode : Bob (3초간 비프음 후)	00:00 Mode : Manual
MOW.STC.008.003	00:10 Mode : Manual, c 입력 후 c	00:00 Mode : Manual
MOW.STC.008.004	70C Mode : Manual, c 입력 후 c	20C Mode : Manual
MOW.STC.008.005	02:00 Mode : Pizza, d	00:00 Mode : Manual
MOW.STC.008.006	02:00 Mode : Pizza, a 입력 후 d	00:00 Mode : Manual
MOW.STC.008.007	00:10 Mode : Manual, d 버튼 6회 입력 (모드 한 루프 회전)	00:00 Mode : Manual

Identifier	PASS/FAIL
MOW.STC.007.000	PASS
MOW.STC.007.001	PASS
MOW.STC.007.002	PASS
MOW.STC.007.003	PASS
MOW.STC.007.004	PASS
MOW.STC.007.005	FAIL (00:00 Mode : Bob)
MOW.STC.008.000	PASS
MOW.STC.008.001	PASS
MOW.STC.008.002	FAIL (00:00 Mode : Bob)
MOW.STC.008.003	PASS
MOW.STC.008.004	PASS
MOW.STC.008.005	PASS
MOW.STC.008.006	FAIL (00:10 Mode Manual)
MOW.STC.008.007	PASS

Comment

- Manual이 아닌 다른 모드 (Bob 등) 에서 a, b 버튼 등 시간/온도 조절 버튼 입력 시 내부적으로 증가하는 현상이 발생

(ex 02:00 Mode : Pizza , a,d Input -> 00:10 Mode : Manual)

- 사전 설정된 모드로 한번 조리 시 제대로 된 Mode 초기화가 되지 않고 지속적 오동작하는 현상이 발생

(ex 00:00 Mode : Bob , (조리 후) -> 00:00 Mode : Bob 유지)

Unit Test Table 1 / Test 대상 Process

Test Id	Name	Description
2.1.1	buttoncontroller	State를 받아서 그 State에 따른 Button Interface를 Trigger 시킨다
2.1.12	Time_TempController	State와 Data를 받아서 그에 따른 기능을 Trigger시키거나 Enable/Disable 시킨다.
2.1.13	Light	Light에 Enable이 들어오면 Lighter Interface에게 Backlight를 켜라는 Backlight Command를 보낸다. Disable이 들어오면 끄라는 Backlight Command를 보낸다.
2.1.14	Make_Sound	Make Sound에 Trigger가 들어오면 Speaker Interface에게 Speak하라는 Speak Command를 보낸다. Disable이 들어오면 끄라는 Speak Command를 보낸다.
2.1.15	TempIncreasing	Temp Increase에 Trigger가 들어오면 Data에 ctemp를 증가시킨다.
2.1.16	TimeDecreasing	Time Increase에 Trigger가 들어오면 Data에 ctime을 증가시킨다.
2.1.17	DisplayController	State와 Data를 받아서 그에 따른 기능을 Enable/Disable 시킨다.
2.1.2	10secIncrease	10sec Increase 에 Trigger가 들어 오면 10초 라는 시간 데이터를 Data에 있는 stime에 누적 시킨다.
2.1.3	10CIncrease	10C Increase 에 Trigger가 들어 오면 10도 라는 온도 데이터를 Data에 있는 stemp에 누적 시킨다.
2.1.4	30secIncrease	30sec Increase 에 Trigger가 들어 오면 30초 라는 시간 데이터를 Data에 있는 stime에 누적 시킨다.
2.1.5	20CIncrease	20C Increase 에 Trigger가 들어 오면 20도 라는 온도 데이터를 Data에 있는 stemp에 누적 시킨다.
2.2	Time_TempDisplayInterface	컨트롤에서 받아온 Time_Temp Display Command 가 True면 시간 또는 온도를 출력해 준다.
2.3	ModeDisplayInterface	컨트롤에서 받아온 Mode Display Command 가 True면 모드를 출력해 준다.
2.4	SpeakerInterface	컨트롤에서 받아온 Speak Command 가 True면 비프음을 3초간 출력해 준다.
2.5	LighterInterface	컨트롤에서 받아온 Backlight Command 가 True면 빛을 방출해 준다.

Unit Test Table 2 / Test 비대상 Process

Test Id	Name	Description
1.1	DoorSenSorInterface	Door Sensor Input의 입력 여부를 판단하여 Door State 를 True/False 로 내보낸다.
1.2	Time_TempButtonInterface	Time_Temp Button Input의 입력 여부를 판단하여 Time_Temp State 를 True/False 로 내보낸다.
1.3	ModeButtonInterface	Mode Button Input의 입력 여부를 판단하여 Mode Button State 를 True/False 로 내보낸다.
1.4	10sec_10CButtonInterface	10sec_10C_incre Button Input이 입력 여부를 판단하여 10sec_10C Button State 를 True/False 로 내보낸다.
1.5	30sec_20CButtonInterface	30sec_20C_incre Button Input의 입력 여부를 판단하여 30sec_20C Button State 를 True/False 로 내보낸다.
1.6	S_CButtonInterface	S_C Button Input의 입력 여부를 판단하여 S_C Button State 를 True/False 로 내보낸다.
2.1.6	ManualMode	Manual Mode에 Trigger가 들어 오면 Manual Mode 라는 문자 데이터를 Data에 있는 cmode에 변경 시킨다.
2.1.7	DduckMode	Dduck Mode에 Trigger가 들어 오면 Dduck Mode 라는 시간 데이터와 문자 데이터를 Data에 있는 stime과 cmode에 변경 시킨다.
2.1.8	JukMode	Juk Mode에 Trigger가 들어 오면 Juk Mode 라는 시간 데이터와 문자 데이터를 Data에 있는 stime과 cmode에 변경 시킨다.

Test Id	Name	Description
2.1.9	BobMode	Bob Mode에 Trigger가 들어 오면 Bob Mode 라는 시간 데이터와 문자 데이터를 Data에 있는 stime과 cmode에 변경 시킨다.
2.1.10	Kuk_JjigaeMode	Kuk_Jjigae Mode에 Trigger가 들어 오면 Kuk_Jjigae Mode 라는 시간 데이터와 문자 데이터를 Data에 있는 stime과 cmode에 변경 시킨다.
2.1.11	PizzaMode	Pizza Mode에 Trigger가 들어 오면 Pizza Mode 라는 시간 데이터와 문자 데이터를 Data에 있는 stime과 cmode에 변경 시킨다.
2.1.18	TimePrint	Time Print에 Enable이 들어오면 Time_Temp Display Interface에게 Time을 Display하라는 Time Display Command를 보낸다. Disable이 들어오면 끄라는 Time Display Command를 보낸다.
2.1.19	TempPrint	Temp Print에 Enable이 들어오면 Time_Temp Display Interface에게 Temp을 Display하라는 Temp Display Command를 보낸다. Disable이 들어오면 끄라는 Temp Display Command를 보낸다.
2.1.20	ModePrint	Time Print에 Enable이 들어오면 Mode Display Interface에게 Mode를 Display하라는 Mode Display Command를 보낸다. Disable이 들어오면 끄라는 Mode Display Command를 보낸다.

테스트 비대상 선정의 공통 이유는 단순히 다음 Process에 Command 인자 값과 Enable/Disable의 조건 값만을 전달하는 역할을 하기 때문에 Test List에 선정하지 않았다.

Unit Test Feature

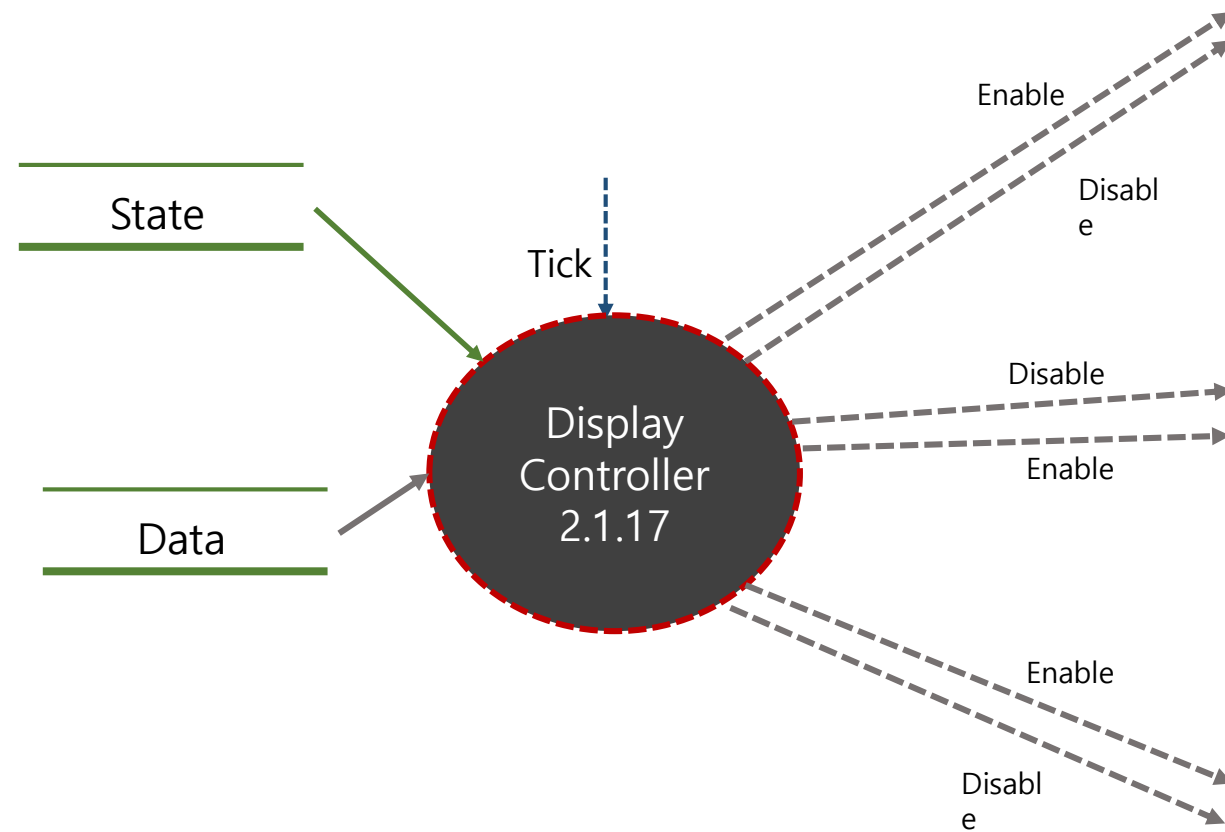
Identifier	Feature (Process ID in DFD)	Valid / Invalid value
MOW.UTC.00.000	2.1.1 buttoncontroller	State를 받고, Time_TempButtonState가 True이다.
MOW.UTC.00.001	2.1.1 buttoncontroller	State를 받고, ModeButtonState가 True이다.
MOW.UTC.00.002	2.1.1 buttoncontroller	State를 받고, DoorState가 True이다.
MOW.UTC.00.003	2.1.1 buttoncontroller	State를 받고, _10sec_10CButtonState가 True이다.
MOW.UTC.00.004	2.1.1 buttoncontroller	State를 받고, _30sec_20CButtonState가 True이다.
MOW.UTC.00.005	2.1.1 buttoncontroller	State를 받고, S_CButtonState가 True이다.
MOW.UTC.01.000	2.1.13 Time_TempController	Data와 State를 받고, DoorState가 True이고, DOOR가 짝수이다.
MOW.UTC.01.001	2.1.13 Time_TempController	Data와 State를 받고, DoorState가 True이고, DOOR가 홀수이다.
MOW.UTC.01.002	2.1.13 Time_TempController	Data와 State를 받고, DoorState가 False이고, S_CButtonState가 True이고, ABLE이 홀수이고, d.ctime이 40이다.
MOW.UTC.01.003	2.1.13 Time_TempController	Data와 State를 받고, DoorState가 False이고, S_CButtonState가 True이고, ABLE이 홀수이고, d.ctime이 0이다.
MOW.UTC.01.004	2.1.13 Time_TempController	Data와 State를 받고, DoorState가 False이고, S_CButtonState가 True이고, ABLE이 짝수이고, d.ctemp이 40이다.
MOW.UTC.01.005	2.1.13 Time_TempController	Data와 State를 받고, DoorState가 False이고, S_CButtonState가 True이고, ABLE이 짝수이고, d.ctime이 0이다.

Identifier	Feature (Process ID in DFD)	Valid / Invalid value
MOW.UTC.02.000	2.1.17 DisplayController	Data와 State를 받고, ABLE이 홀수 이다.
MOW.UTC.02.001	2.1.17 DisplayController	Data와 State를 받고, ABLE이 짝수 이다.
MOW.UTC.03.000	2.2 Time_TempDisplayInterface	TimeDisplayCommand가 True고 d.ctime이 -1이고, d.stime이 10이다.
MOW.UTC.03.001	2.2 Time_TempDisplayInterface	TimeDisplayCommand가 True고 d.ctime이 100이다.
MOW.UTC.03.002	2.2 Time_TempDisplayInterface	TempDisplayCommand가 True고 d.ctemp이 -1이고, d.stemp가 40이다.
MOW.UTC.03.003	2.2 Time_TempDisplayInterface	TempDisplayCommand가 True고 d.ctemp이 50이다.
MOW.UTC.04.000	2.3 ModeDisplayInterface	ModeDisplayCommand가 True고 d.i_cmode가 0 이다.
MOW.UTC.04.001	2.3 ModeDisplayInterface	ModeDisplayCommand가 True고 d.i_cmode가 1 이다.
MOW.UTC.04.002	2.3 ModeDisplayInterface	ModeDisplayCommand가 True고 d.i_cmode가 2 이다.
MOW.UTC.04.003	2.3 ModeDisplayInterface	ModeDisplayCommand가 True고 d.i_cmode가 3 이다.
MOW.UTC.04.004	2.3 ModeDisplayInterface	ModeDisplayCommand가 True고 d.i_cmode가 4 이다.
MOW.UTC.04.005	2.3 ModeDisplayInterface	ModeDisplayCommand가 True고 d.i_cmode가 5 이다.

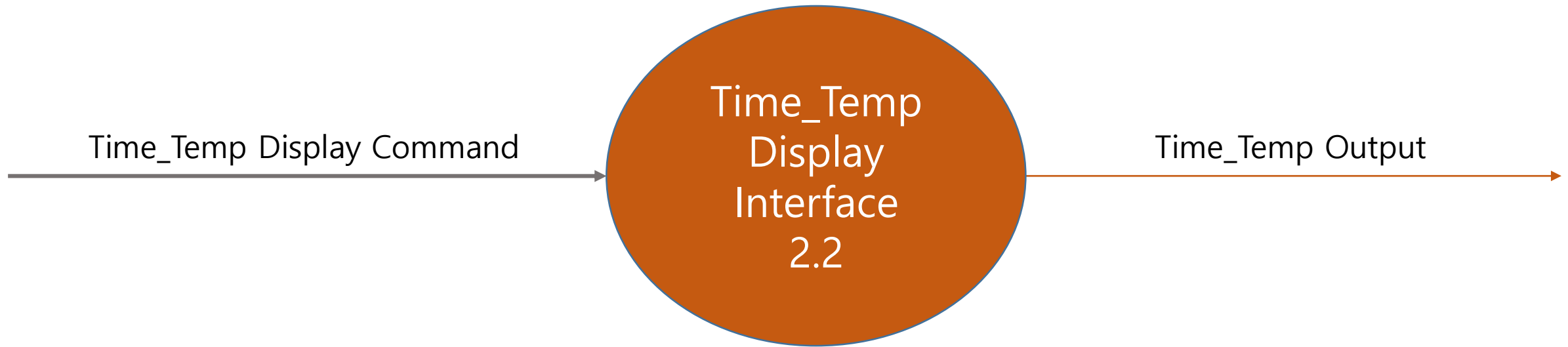
Identifier	Feature (Process ID in DFD)	Valid / Invalid value
MOW.UTC.05.000	2.4 SpeakerInterface	SpeakCommand가 True다.
MOW.UTC.06.000	2.5 LighterInterface	BacklightCommand가 True이다.
MOW.UTC.06.001	2.5 LighterInterface	BacklightCommand가 False이다.
MOW.UTC.07.000	2.1.2 10secIncrease	Trigger 받고, d.stime이 30이다.
MOW.UTC.08.000	2.1.3 10CIncrease	Trigger 받고, d.stemp이 40이다.
MOW.UTC.09.000	2.1.4 30secIncrease	Trigger 받고, d.stime이 90이다.
MOW.UTC.10.000	2.1.5 20CIncrease	Trigger 받고, d.stemp이 50이다.
MOW.UTC.11.000	2.1.13 Light	Enable 이 True고, Disable 이 False 이다.
MOW.UTC.11.001	2.1.13 Light	Enable 이 False고, Disable 이 True 이다.
MOW.UTC.12.000	2.1.14 Make_Sound	Trigger 받는다.
MOW.UTC.13.000	2.1.15 TempIncreasing	Trigger 받고, d.ctemp는 50이다.
MOW.UTC.14.000	2.1.16 TimeDecreasing	Trigger 받고, d.ctime는 80이다.

Unit Test Case

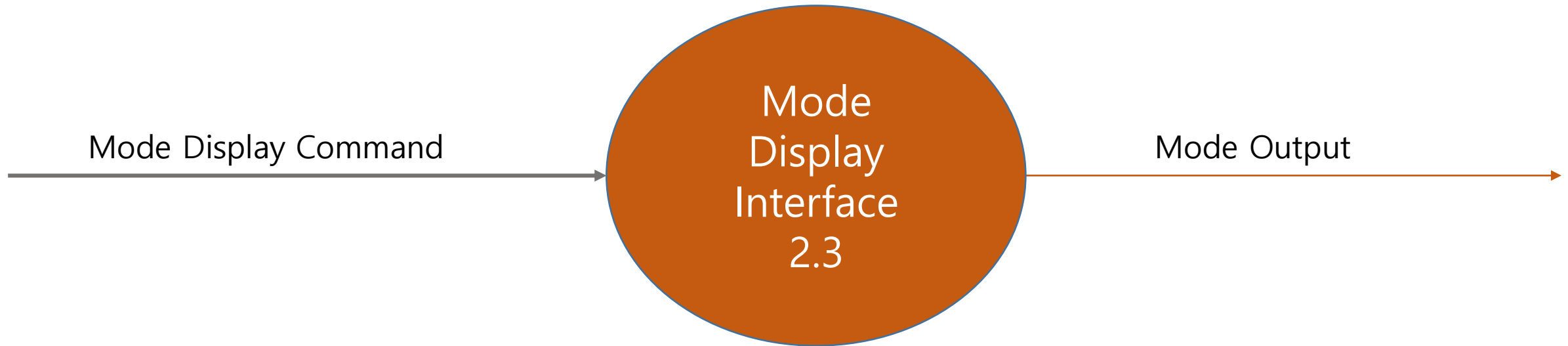
Identifier	Input specification	Output specification
MOW.UTC.00.000	State in / Time_TempButtonState == True / state%2=0	d.stime = 0
MOW.UTC.00.001	State in / Time_TempButtonState == True / state%2=1	d.stemp = 20
MOW.UTC.00.002	State in / ModeButtonState == True / (state % 2) != 0 / mode = 5	d.stime = 0
MOW.UTC.00.003	State in / _10sec_10CButtonState == True / state%2=1	_10secIncrease(T)
MOW.UTC.00.004	State in / _10sec_10CButtonState == True / state%2=0	_10CIncrease(T)
MOW.UTC.00.005	State in / _30sec_20CButtonState == True / state%2=1	_30secIncrease(T)
MOW.UTC.00.006	State in / _30sec_20CButtonState == True / state%2=0	_10CIncrease(T)
MOW.UTC.01.000	Data in / State in / DoorState == True / DOOR%2==0	enable = T
MOW.UTC.01.001	Data in / State in / DoorState == True / DOOR%2==1	disable = T
MOW.UTC.01.002	Data in / State in / DoorState == False / S_CButtonState == True / ABLE%2=1 / d.ctime = 40	Light(T, F) / TimeDecreasing(T)
MOW.UTC.01.003	Data in / State in / DoorState == False / S_CButtonState == True / ABLE%2=1 / d.ctime = 0	Light(F, T) / Make_Sound(T)
MOW.UTC.01.004	Data in / State in / DoorState == False / S_CButtonState == True / ABLE%2=0 / d.ctemp = 40 / d.stemp = 40	Light(F, T) / Make_Sound(T)
MOW.UTC.01.005	Data in / State in / DoorState == False / S_CButtonState == True / ABLE%2=0 / d.ctemp = 20 / d.stemp = 50	Light(T, F) / TempIncreasing(T)



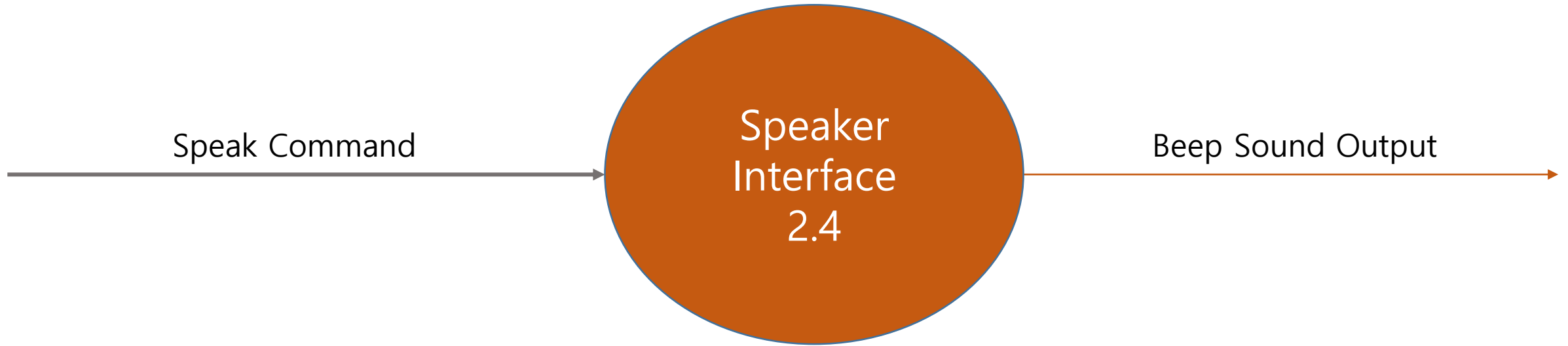
Identifier	Input specification	Output specification
MOW.UTC.02.000	Data in / State in / ABLE%2==1	TimePrint(T, F) / ModePrint(T, F)
MOW.UTC.02.001	Data in / State in / ABLE%2==0	TempPrint(T, F) / ModePrint(T, F)



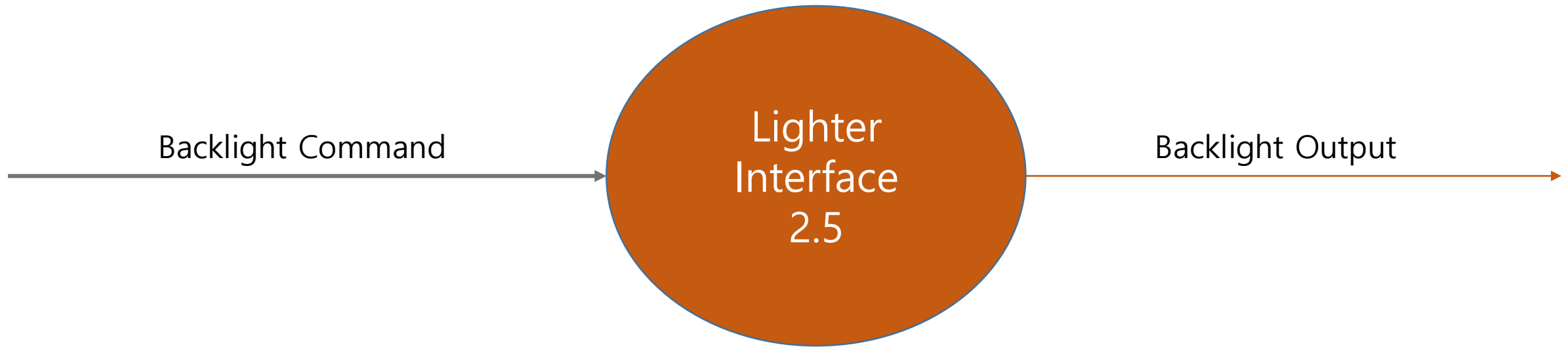
Identifier	Input specification	Output specification
MOW.UTC.03.000	TimeDisplayCommand == True / d.ctime = -1 / d.stime = 10	printf("0:10");
MOW.UTC.03.001	TimeDisplayCommand == True / d.ctime = 100	printf("1:40");
MOW.UTC.03.002	TimeDisplayCommand == True / d.ctemp = -1 / d.stemp = 40	printf("40C");
MOW.UTC.03.003	TimeDisplayCommand == True / d.ctemp = 50	printf("50C");



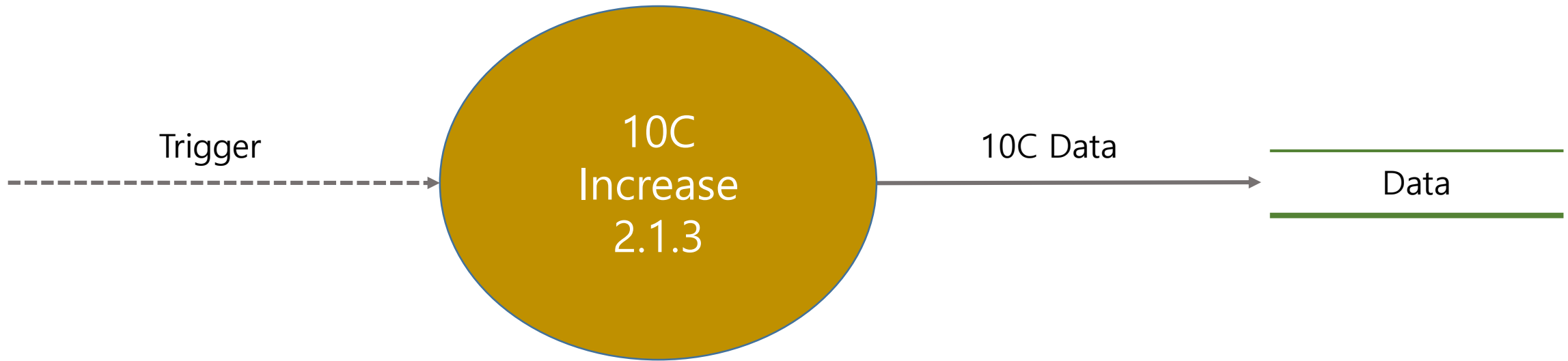
Identifier	Input specification	Output specification
MOW.UTC.04.000	ModeDisplayCommand == True / d.i_cmode = 0	printf("Manual\\n");
MOW.UTC.04.001	ModeDisplayCommand == True / d.i_cmode = 1	printf("Dduck\\n");
MOW.UTC.04.002	ModeDisplayCommand == True / d.i_cmode = 2	printf("Juk\\n");
MOW.UTC.04.003	ModeDisplayCommand == True / d.i_cmode = 3	printf("Bob\\n");
MOW.UTC.04.004	ModeDisplayCommand == True / d.i_cmode = 4	printf("Kuk_Jjigae\\n");
MOW.UTC.04.005	ModeDisplayCommand == True / d.i_cmode = 5	printf("Pizza\\n");



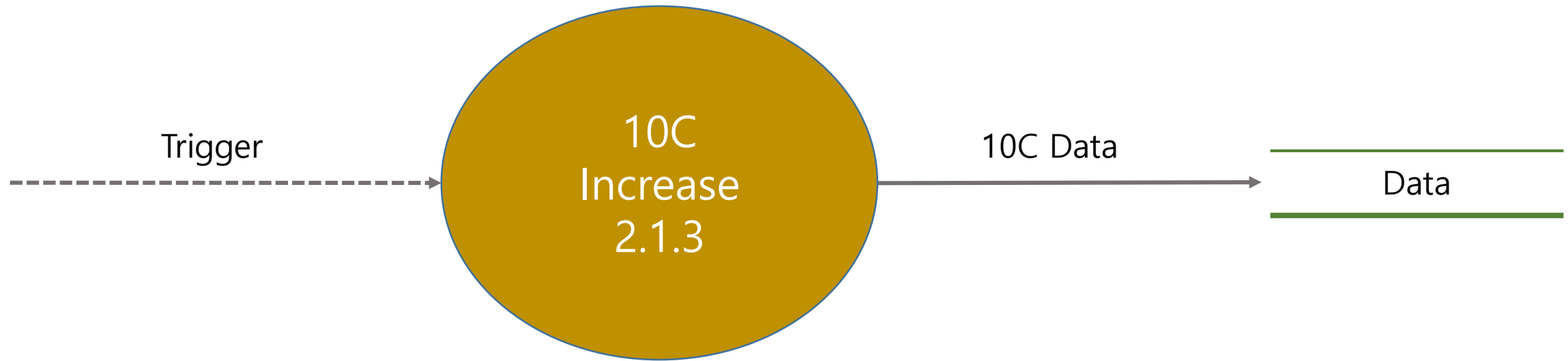
Identifier	Input specification	Output specification
MOW.UTC.05.000	SpeakCommand == True	printf("Wa"); * 3



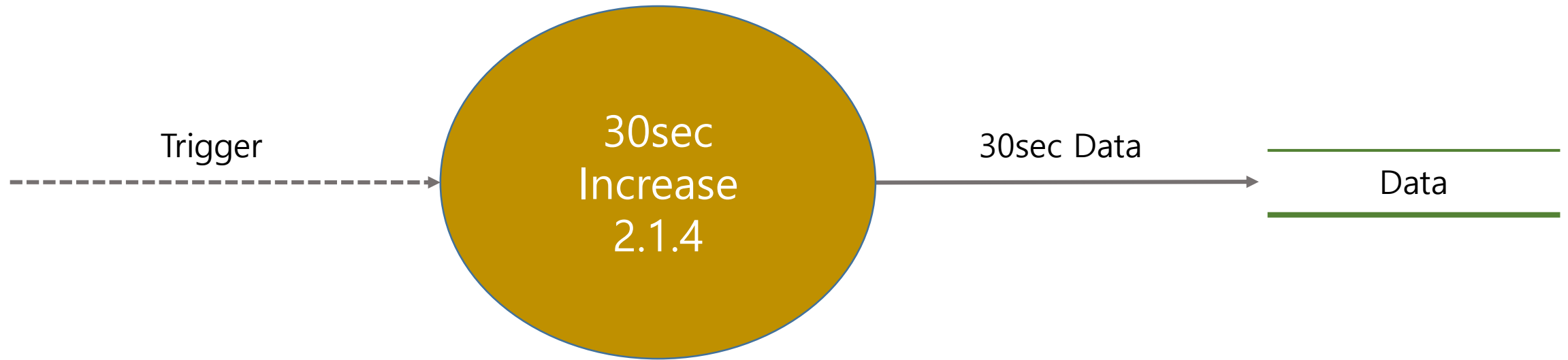
Identifier	Input specification	Output specification
MOW.UTC.06.000	BacklightCommand == True	printf("%c[1;33m",27);
MOW.UTC.06.001	BacklightCommand == False	printf("%c[0m", 27);



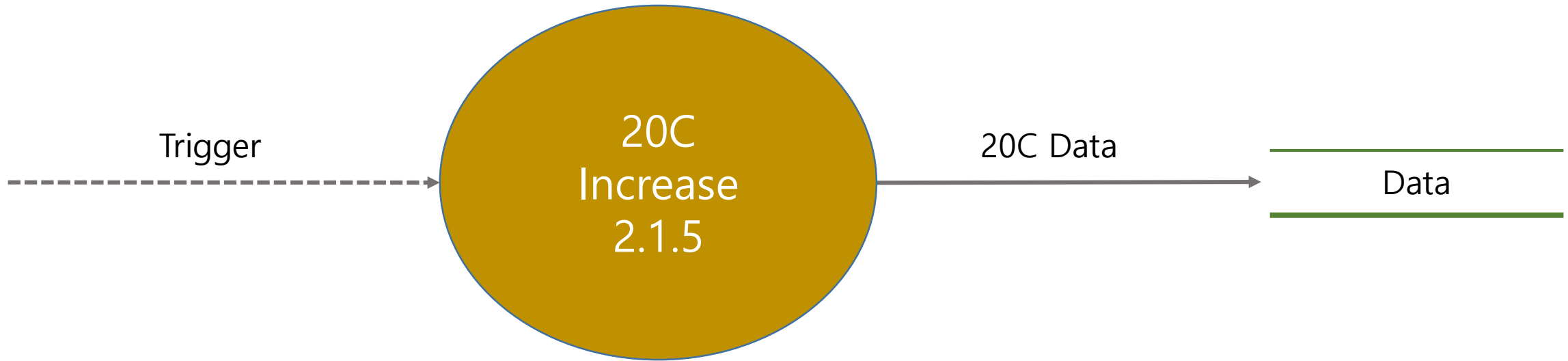
Identifier	Input specification	Output specification
MOW.UTC.07.000	Trigger in / d.stime = 30	d.stime = 40



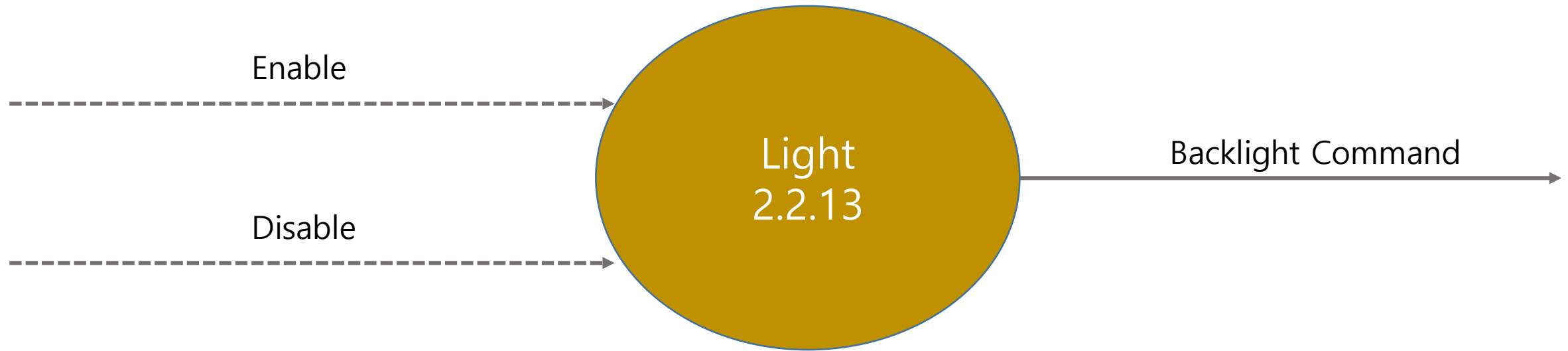
Identifier	Input specification	Output specification
MOW.UTC.08.000	Trigger in / d.stemp = 40	d.stemp = 50



Identifier	Input specification	Output specification
MOW.UTC.09.000	Trigger in / d.stime = 90	d.stime = 120



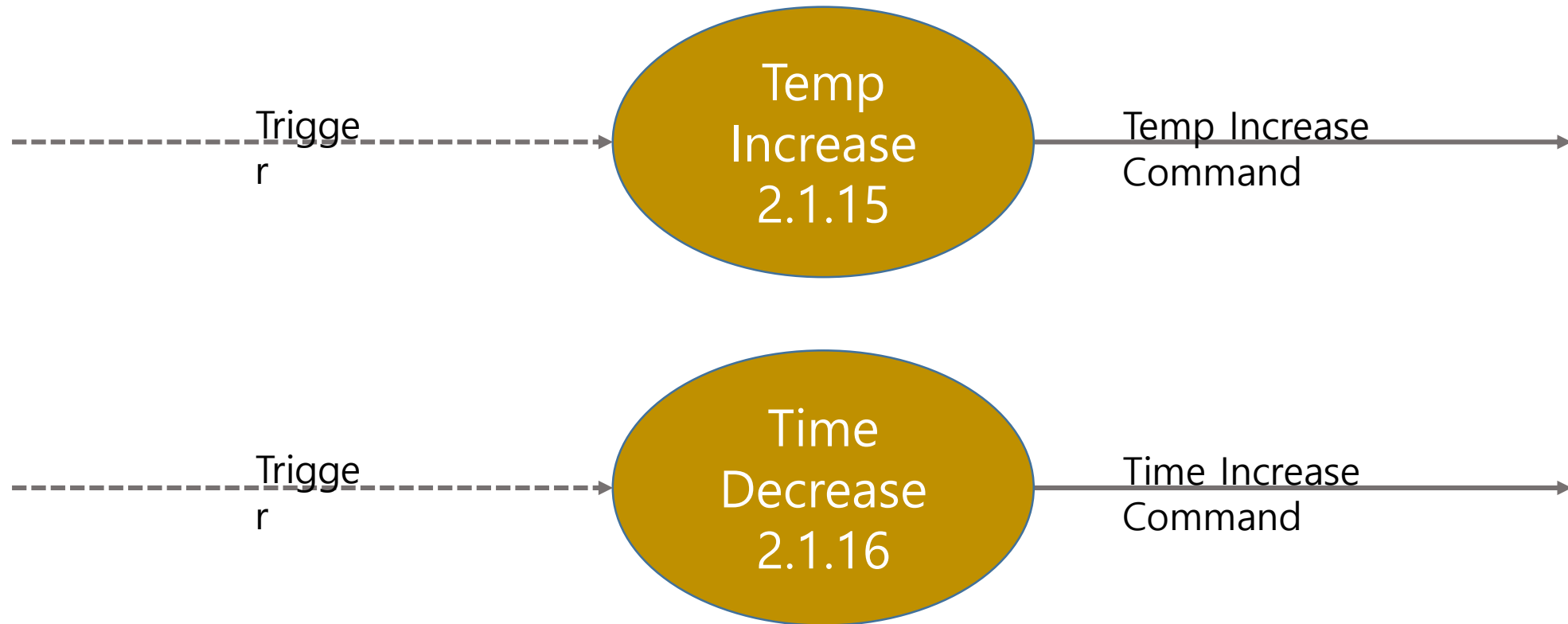
Identifier	Input specification	Output specification
MOW.UTC.10.000	Trigger in / d.stemp = 50	d.stemp = 70



Identifier	Input specification	Output specification
MOW.UTC.11.000	Enable == True / Disable == False	LighterInterface(T)
MOW.UTC.11.001	Enable == False / Disable == True	LighterInterface(F)



Identifier	Input specification	Output specification
MOW.UTC.12.000	Trigger in	SpeakerInterface(T)



Identifier	Input specification	Output specification
MOW.UTC.13.000	Trigger in / d.ctemp = 50	DisplayController / d.ctemp = 60
MOW.UTC.14.000	Trigger in / d.ctime = 80	DisplayController / d.ctime = 79

Identifier	PASS/FAIL
MOW.UTC.00.000	PASS
MOW.UTC.00.001	PASS
MOW.UTC.00.002	PASS
MOW.UTC.00.003	PASS
MOW.UTC.00.004	PASS
MOW.UTC.00.005	PASS
MOW.UTC.00.006	PASS
MOW.UTC.01.000	PASS
MOW.UTC.01.001	PASS
MOW.UTC.01.002	PASS
MOW.UTC.01.003	PASS
MOW.UTC.01.004	PASS
MOW.UTC.01.005	PASS
MOW.UTC.02.000	PASS
MOW.UTC.02.001	PASS

Identifier	PASS/FAIL
MOW.UTC.03.000	PASS
MOW.UTC.03.001	PASS
MOW.UTC.03.002	PASS
MOW.UTC.03.003	PASS
MOW.UTC.04.000	PASS
MOW.UTC.04.001	PASS
MOW.UTC.04.002	PASS
MOW.UTC.04.003	PASS
MOW.UTC.04.004	PASS
MOW.UTC.04.005	PASS
MOW.UTC.05.000	PASS
MOW.UTC.06.000	PASS
MOW.UTC.06.001	PASS
MOW.UTC.07.000	PASS

Identifier	PASS/FAIL
MOW.UTC.08.000	PASS
MOW.UTC.09.000	PASS
MOW.UTC.10.000	PASS
MOW.UTC.11.000	PASS
MOW.UTC.11.001	PASS
MOW.UTC.12.000	PASS
MOW.UTC.13.000	PASS
MOW.UTC.14.000	PASS

Comment

- 프로세스를 독립적으로 봤을 때 각각의 기능은 제대로 수행한다.
- 그러나 초기값이 조건에 명시된 값과 맞지 않는 경우가 있고, 같은 기능을 수행하는 변수들이 각기 다른 프로세스에서 사용되었다.
- 즉, 구현된 소스의 각 프로세스의 기능적인 면에서 볼 때는 문제가 없지만 코딩상의 문제로 데이터의 흐름이 엮여있어 구조상 오류가 발생했다.

Microwave Oven System System & Unit Testing #2

T3

이경수,한득환,김대희,신민용

Failed System Test List

Test case identifier	Input specification	Output specification	Result(Pass/Fail)
MOW.STC.004.000	00:00 Mode : Manual, e	00:00 Mode : Manual	FAIL (바로 비프음)
MOW.STC.004.001	00:10 Mode : Manual, e, (문열림), 1초후	00:10 Mode : Manual	FAIL (00:00 Mode : Manual)
MOW.STC.004.002	40C Mode : Manual, e, (문열림), 3초후	40C Mode : Manual	FAIL (20C Mode : Manual)
MOW.STC.004.003	02:00 Mode : Bob, e, (문열림), 1초후	02:00 Mode : Bob	FAIL (00:00 Mode : Bob)
MOW.STC.006.016	01:59 Mode : Bob (작동중), e	00:00 Mode : Manual	FAIL (00:00 Mode : Bob)
MOW.STC.007.005	00:00 Mode : Bob (3초간 비프음 후)	00:00 Mode : Manual	FAIL (00:00 Mode : Bob)
MOW.STC.008.002	00:00 Mode : Bob (3초간 비프음 후)	00:00 Mode : Manual	FAIL (00:00 Mode : Bob)
MOW.STC.008.006	02:00 Mode : Pizza, a 입력 후 d	00:00 Mode : Manual	FAIL (00:10 Mode Manual)

What's a Problem???

- 초기상태/무설정/무조건 상태에서 시작 및 종료되는 현상
- 각 설정상태(시간, 온도, 사용자설정)에서의 시작 불가 예외처리 중 자동 초기화하는 현상
- 사용자설정(Mode) 상태에서 Oven을 이용 후 초기 Manual Mode로 초기화되지 않는 현상

-초기상태에서의 시작 예외 처리와, 각 프로세스에서 구조체의 초기화 과정에서의 오류로 판단되어 초기화에 중점을 두고 분석 및 수정에 들어감

초기상태/무설정/무조건 상태에서 시작이 불가하도록 예외 처리

```
int main(){
    ...
    switch(userinput){
        ...
        case 'e':
            if((d.stime == 0) && (d.stemp == 20)) break;
            if((DOOR%2) == 0) break;
            // e버튼으로 시작과 동시에 초기상태인지, 문열림인지 확인
        ...
    }
    ...
    return 0;
}
```

각 프로세스에서의 무분별한 초기화로 인한 구조적 오류를 예방하기 위해 초기화 함수를 직접 구현하여 사용

```
void DataClear()
```

```
{  
    d.stime = 0;  
    d.stemp = 20;  
    d.ctime = -1;  
    d.ctemp = -1;  
    d.i_cmode = 0;  
}
```

```
void StateClear()
```

```
{  
    s.DoorState = F;  
    s.Time_TempButtonState = F;  
    s.ModeButtonState = F;  
    s._10sec_10CButtonState = F;  
    s._30sec_20CButtonState = F;  
    s.S_CButtonState = F;  
}
```

```

int main(){
    ...
    switch(userinput){
    ...
        case 'e':
            ...
            d.ctime = d.stime;
            d.ctemp = 20;
            S_CButtonInput = T;
            S_CButtonInterface(S_CButtonInput);           // 1.6
            buttoncontroller(&s);                          // 2.1.1
            while ( SC != 1 ) {
                if(kbhit()){
                    a = getch();
                    if(a == 'e'){
                        s.S_CButtonState = F;
                        DataClear();
                        Light(F, T);                      // 2.1.13
                        DisplayController(&d, &s);          // 2.1.17
                        break;
                    }
                }
            }
            ...
            DataClear();

```

```

void SpeakerInterface (Bool SpeakCommand) {                                     // 2.4
    int i;
    char a;
    if (T == SpeakCommand) {
        for(i=0; i<3; i++) {
            printf("Wa");
            sleep(1);
            if(kbhit()){
                a = getch();
                if(a == 'e'){
                    s.S_CButtonState = F;
                    DataClear();
                    Light(F, T);
                    DisplayController(&d, &s);
                    break;
                }
            }
        }
    }
    SC=1;
}
}

```

```
void buttoncontroller (struct State *s){  
    ...  
    if ( d.i_cmode == 0 ) {  
        DataClear();  
        StateClear();  
    }  
    ...  
}
```

Restart System Test List

Test case identifier	Input specification	Output specification	Result(Pass/Fail)
MOW.STC.004.000	00:00 Mode : Manual, e	00:00 Mode : Manual	PASS
MOW.STC.004.001	00:10 Mode : Manual, e, (문열림), 1초후	00:10 Mode : Manual	PASS
MOW.STC.004.002	40C Mode : Manual, e, (문열림), 3초후	40C Mode : Manual	PASS
MOW.STC.004.003	02:00 Mode : Bob, e, (문열림), 1초후	02:00 Mode : Bob	PASS
MOW.STC.006.016	01:59 Mode : Bob (작동중), e	00:00 Mode : Manual	PASS
MOW.STC.007.005	00:00 Mode : Bob (3초간 비프음 후)	00:00 Mode : Manual	PASS
MOW.STC.008.002	00:00 Mode : Bob (3초간 비프음 후)	00:00 Mode : Manual	PASS
MOW.STC.008.006	02:00 Mode : Pizza, a 입력 후 d	00:00 Mode : Manual	PASS

Microwave Oven System System & CUnit Testing #3

T3

이경수,한득환,김대희,신민용

System Test 변동사항

- 기존 1회차 System Test의 Fail 관련한 사항은 모두 2회차 테스트에서 수정되고, 결과가 모두 PASS로 나옴
- Source Code의 변경사항은 System적인 측면에서 변화를 주는 것이 아닌 단순 초기화 함수의 추가이므로, 기존에 수행했던 System Test 결과에 변동사항으로 반영될 수 없으며, 실제 기존의 Case에 따라 System Test 결과 모두 변동이 없음
- 따라서, 예전의 Test Feature와, Case에 추가하여 조금 더 동작 과정에서 심화된 과정이나 누락된 예외사항을 시험해보고자 하였음

Lasted System Test Feature

Identifier	Feature	valid / Invalid value
MOW.STC.000	시간확인	시작과 동시에 화면 표시, 최대값 넘어 갈 때 시간 연산, A, B 버튼 입력 시 시간 변경 확인
MOW.STC.001	온도확인	시간/온도 전환 확인, 최대값 넘어 갈 때 온도 연산, A, B 버튼 입력 시 온도 변경 확인
MOW.STC.002	모드확인	D 버튼 입력 시 설정된 모드와 시간 확인 모드변경 루프 정상작동 확인
MOW.STC.003	각 상황에서 Backlight 실행	F 버튼 입력 시 점/소등 확인 F 버튼 입력 후 다른 버튼 입력 시 점/소등 유지여부 확인
MOW.STC.004	조리시작	초기 미 설정 상태에서의 시작 가능여부 확인 문의 개폐에 따른 시작 확인 및 조리 시작 시 조명 점등 확인
MOW.STC.005	조리중	조리 중 조명 상태 점등 확인 Tick에 따른 시간 감소와 온도 증가 확인
MOW.STC.006	조리중 버튼입력	조리 중 다른 버튼 입력 시 동작불가 확인 조리 중 Cancel 동작 확인
MOW.STC.007	조리종료	조리 이후 종료 시 비프음 출력 확인 조명상태 확인
MOW.STC.008	RESET	조리 종료 시 초기화 모드 변경 확인 모드변경 시 이전 설정 값 초기화 확인

Lasted System Test Case / Result

Test case identifier	Input specification	Output specification
MOW.STC.000.000	프로그램 실행	00:00 Mode : Manual
MOW.STC.000.001	00:00 Mode : Manual, a	00:10 Mode : Manual
MOW.STC.000.002	00:10 Mode : Manual, b	00:40 Mode : Manual
MOW.STC.000.003	09:50 Mode : Manual, a	00:00 Mode : Manual
MOW.STC.000.004	09:50 Mode : Manual, b	00:20 Mode : Manual
MOW.STC.000.005	20C Mode : Manual, c	00:00 Mode : Manual
MOW.STC.001.000	00:00 Mode : Manual, c	20C Mode : Manual
MOW.STC.001.001	20C Mode : Manual, a	30C Mode : Manual
MOW.STC.001.002	30C Mode : Manual, b	50C Mode : Manual
MOW.STC.001.003	90C Mode : Manual, a	30C Mode : Manual
MOW.STC.001.004	80C Mode : Manual, b	30C Mode : Manual

Identifier	PASS/FAIL
MOW.STC.000.000	PASS
MOW.STC.000.001	PASS
MOW.STC.000.002	PASS
MOW.STC.000.003	PASS
MOW.STC.000.004	PASS
MOW.STC.000.005	PASS
MOW.STC.001.000	PASS
MOW.STC.001.001	PASS
MOW.STC.001.002	PASS
MOW.STC.001.003	PASS
MOW.STC.001.004	PASS

Test case identifier	Input specification	Output specification
MOW.STC.002.000	00:00 Mode : Manual, d	01:00 Mode : Dduck
MOW.STC.002.001	01:00 Mode : Dduck, d	01:30 Mode : Juk
MOW.STC.002.002	01:30 Mode : Juk, d	02:00 Mode : Bob
MOW.STC.002.003	02:00 Mode : Bob, d	05:00 Mode : Kuk_Jigae
MOW.STC.002.004	05:00 Mode : Kuk_Jigae, d	02:00 Mode : Pizza
MOW.STC.002.005	02:00 Mode : Pizza, d	00:00 Mode : Manual
MOW.STC.002.006	20C Mode : Manual, d	20C Mode : Manual
MOW.STC.002.007	01:30 Mode : Juk, a	01:30 Mode : Juk
MOW.STC.002.008	01:30 Mode : Juk, b	01:30 Mode : Juk
MOW.STC.002.009	01:30 Mode : Juk, c	01:30 Mode : Juk

Identifier	PASS/FAIL
MOW.STC.002.000	PASS
MOW.STC.002.001	PASS
MOW.STC.002.002	PASS
MOW.STC.002.003	PASS
MOW.STC.002.004	PASS
MOW.STC.002.005	PASS
MOW.STC.002.006	PASS
MOW.STC.002.007	PASS
MOW.STC.002.008	PASS
MOW.STC.002.009	PASS

Test case identifier	Input specification	Output specification
MOW.STC.003.000	00:00 Mode : Manual, f	00:00 Mode : Manual
MOW.STC.003.001	20C Mode : Manual, f	20C Mode : Manual
MOW.STC.003.002	00:00 Mode : Manual, f	00:00 Mode : Manual
MOW.STC.003.003	20C Mode : Manual, f	20C Mode : Manual
MOW.STC.003.004	02:00 Mode : Bob, f	02:00 Mode : Bob
MOW.STC.003.005	02:00 Mode : Bob, f	02:00 Mode : Bob
MOW.STC.003.006	00:00 Mode : Manual, a	00:10 Mode : Manual
MOW.STC.003.007	00:00 Mode : Manual, b	00:30 Mode : Manual
MOW.STC.003.008	20C Mode : Manual, a	30C Mode : Manual
MOW.STC.003.009	20C Mode : Manual, b	40C Mode : Manual
MOW.STC.003.010	00:00 Mode : Manual, c	20C Mode : Manual
MOW.STC.003.011	20C Mode : Manual, c	00:00 Mode : Manual
MOW.STC.003.012	00:00 Mode : Manual, d	01:00 Mode : Dduck

Identifier	PASS/FAIL
MOW.STC.003.000	PASS
MOW.STC.003.001	PASS
MOW.STC.003.002	PASS
MOW.STC.003.003	PASS
MOW.STC.003.004	PASS
MOW.STC.003.005	PASS
MOW.STC.003.006	PASS
MOW.STC.003.007	PASS
MOW.STC.003.008	PASS
MOW.STC.003.009	PASS
MOW.STC.003.010	PASS
MOW.STC.003.011	PASS
MOW.STC.003.012	PASS

Test case identifier	Input specification	Output specification
MOW.STC.004.000	00:00 Mode : Manual, e	00:00 Mode : Manual
MOW.STC.004.001	00:10 Mode : Manual, e, (문열림), 1초후	00:10 Mode : Manual
MOW.STC.004.002	40C Mode : Manual, e, (문열림), 3초후	40C Mode : Manual
MOW.STC.004.003	02:00 Mode : Bob, e, (문열림), 1초후	02:00 Mode : Bob
MOW.STC.004.004	00:10 Mode : Manual, e	00:10 Mode : Manual
MOW.STC.004.005	40C Mode : Manual, e	20C Mode : Manual
MOW.STC.004.006	02:00 Mode : Bob, e	02:00 Mode : Bob
MOW.STC.005.000	00:10 Mode : Manual , (작동중), 1초후	00:09 Mode : Manual
MOW.STC.005.001	00:10 Mode : Manual , (작동중), 10초후	00:00 Mode : Manual
MOW.STC.005.002	20C Mode : Manual, (작동중), 3초후, 80C 설정	30C Mode : Manual
MOW.STC.005.003	20C Mode : Manual, (작동중), 18초후	80C Mode : Manual
MOW.STC.005.004	02:00 Mode : Bob, (작동중), 1초후	01:59 Mode : Bob
MOW.STC.005.005	02:00 Mode : Bob, (작동중), 2분후	00:00 Mode : Bob

Identifier	PASS/FAIL
MOW.STC.004.000	PASS
MOW.STC.004.001	PASS
MOW.STC.004.002	PASS
MOW.STC.004.003	PASS
MOW.STC.004.004	PASS
MOW.STC.004.005	PASS
MOW.STC.004.006	PASS
MOW.STC.005.000	PASS
MOW.STC.005.001	PASS
MOW.STC.005.002	PASS
MOW.STC.005.003	PASS
MOW.STC.005.004	PASS
MOW.STC.005.005	PASS

Test case identifier	Input specification	Output specification
MOW.STC.006.000	00:09 Mode : Manual (작동중), a	00:09 Mode : Manual (작동중)
MOW.STC.006.001	00:09 Mode : Manual (작동중), b	00:09 Mode : Manual (작동중)
MOW.STC.006.002	00:09 Mode : Manual (작동중), c	00:09 Mode : Manual (작동중)
MOW.STC.006.003	00:09 Mode : Manual (작동중), d	00:09 Mode : Manual (작동중)
MOW.STC.006.004	00:09 Mode : Manual (작동중), e	00:00 Mode : Manual
MOW.STC.006.005	00:09 Mode : Manual (작동중), f	00:09 Mode : Manual (작동중)
MOW.STC.006.006	30C Mode : Manual (작동중), a	30C Mode : Manual (작동중)
MOW.STC.006.007	30C Mode : Manual (작동중), b	30C Mode : Manual (작동중)
MOW.STC.006.008	30C Mode : Manual (작동중), c	30C Mode : Manual (작동중)
MOW.STC.006.009	30C Mode : Manual (작동중), d	30C Mode : Manual (작동중)
MOW.STC.006.010	30C Mode : Manual (작동중), e	20C Mode : Manual
MOW.STC.006.011	30C Mode : Manual (작동중), f	30C Mode : Manual (작동중)
MOW.STC.006.012	01:59 Mode : Bob (작동중), a	01:59 Mode : Bob (작동중)
MOW.STC.006.013	01:59 Mode : Bob (작동중), b	01:59 Mode : Bob (작동중)
MOW.STC.006.014	01:59 Mode : Bob (작동중), c	01:59 Mode : Bob (작동중)
MOW.STC.006.015	01:59 Mode : Bob (작동중), d	01:59 Mode : Bob (작동중)
MOW.STC.006.016	01:59 Mode : Bob (작동중), e	00:00 Mode : Manual
MOW.STC.006.017	01:59 Mode : Bob (작동중), f	01:59 Mode : Bob (작동중)

Identifier	PASS/FAIL
MOW.STC.006.000	PASS
MOW.STC.006.001	PASS
MOW.STC.006.002	PASS
MOW.STC.006.003	PASS
MOW.STC.006.004	PASS
MOW.STC.006.005	PASS
MOW.STC.006.006	PASS
MOW.STC.006.007	PASS
MOW.STC.006.008	PASS
MOW.STC.006.009	PASS
MOW.STC.006.010	PASS
MOW.STC.006.011	PASS
MOW.STC.006.012	PASS
MOW.STC.006.013	PASS
MOW.STC.006.014	PASS
MOW.STC.006.015	PASS
MOW.STC.006.016	PASS
MOW.STC.006.017	PASS

Test case identifier	Input specification	Output specification
MOW.STC.007.000	00:01 Mode : Manual , (작동중), 1초후	00:00 Mode : Manual (3초간 비프음)
MOW.STC.007.001	00:00 Mode : Manual (3초간 비프음 후)	00:00 Mode : Manual
MOW.STC.007.002	30C Mode : Manual, (작동중), 3초후, 40C 설정	40C Mode : Manual (3초간 비프음)
MOW.STC.007.003	40C Mode : Manual (3초간 비프음 후)	20C Mode : Manual
MOW.STC.007.004	00:01 Mode : Bob, (작동중), 1초후	00:00 Mode : Bob (3초간 비프음)
MOW.STC.007.005	00:00 Mode : Bob (3초간 비프음 후)	00:00 Mode : Manual
MOW.STC.008.000	00:00 Mode : Manual (3초간 비프음 후)	00:00 Mode : Manual
MOW.STC.008.001	40C Mode : Manual (3초간 비프음 후)	20C Mode : Manual
MOW.STC.008.002	00:00 Mode : Bob (3초간 비프음 후)	00:00 Mode : Manual
MOW.STC.008.003	00:10 Mode : Manual, c 입력 후 c	00:00 Mode : Manual
MOW.STC.008.004	70C Mode : Manual, c 입력 후 c	20C Mode : Manual
MOW.STC.008.005	02:00 Mode : Pizza, d	00:00 Mode : Manual
MOW.STC.008.006	02:00 Mode : Pizza, a 입력 후 d	00:00 Mode : Manual
MOW.STC.008.007	00:10 Mode : Manual, d 버튼 6회 입력 (모드 한 루프 회전)	00:00 Mode : Manual

Identifier	PASS/FAIL
MOW.STC.007.000	PASS
MOW.STC.007.001	PASS
MOW.STC.007.002	PASS
MOW.STC.007.003	PASS
MOW.STC.007.004	PASS
MOW.STC.007.005	PASS
MOW.STC.008.000	PASS
MOW.STC.008.001	PASS
MOW.STC.008.002	PASS
MOW.STC.008.003	PASS
MOW.STC.008.004	PASS
MOW.STC.008.005	PASS
MOW.STC.008.006	PASS
MOW.STC.008.007	PASS

Additional System Test Feature

Identifier	Feature	valid / Invalid value
MOW.STC.009	예외처리	Start/Cancel 버튼 Input 전, 후 State와 Data를 설정하는 단계에서 원활한 동작 확인 설정 도중 SRS 요구의 제어 예외처리 확인
MOW.STC.010	조리 이후 재시작	1회 조리 후 재설정 조리가능여부 점검으로, 시스템의 지속성 여부 확인
MOW.STC.011	연속된 버튼입력	조리도중의 무작위 연속된 버튼입력으로 오작동 발생여부 확인

Additional System Test Case / Result

Test case identifier	Input specification	Output specification
MOW.STC.009.000	00:00 Mode : Manual, a, b, a, b	01:20 Mode : Manual
MOW.STC.009.001	00:00 Mode : Bob, c	00:00 Mode : Bob
MOW.STC.009.002	00:00 Mode : Manual, a, b, a, b (문열림)	01:20 Mode : Manual
MOW.STC.009.003	00:00 Mode : Bob, c (문열림)	00:00 Mode : Bob
MOW.STC.009.004	20C Mode : Manual, a, b, a, b	80C Mode : Manual
MOW.STC.009.005	20C Mode : Manual, a, b, a, b (문열림)	80C Mode : Manual
MOW.STC.009.006	00:00 Mode : Manual, a, b, a, b, c	20C Mode : Manual
MOW.STC.009.007	20C Mode : Manual, a, b, a, b, c	00:00 Mode : Manual
MOW.STC.009.008	00:00 Mode : Manual, a, b, a, b, c (문열림)	80C Mode : Manual
MOW.STC.009.009	20C Mode : Manual, a, b, a, b, c (문열림)	00:00 Mode : Manual

Identifier	PASS/FAIL
MOW.STC.009.000	PASS
MOW.STC.009.001	PASS
MOW.STC.009.002	PASS
MOW.STC.009.003	PASS
MOW.STC.009.004	PASS
MOW.STC.009.005	PASS
MOW.STC.009.006	PASS
MOW.STC.009.007	PASS
MOW.STC.009.008	PASS
MOW.STC.009.009	PASS

Test case identifier	Input specification	Output specification
MOW.STC.010.001	00:01 Mode : Manual (작동완료 후), a, b, a	00:50 Mode : Manual
MOW.STC.010.002	80C Mode : Manual (작동완료 후), c, b, a	50C Mode : Manual
MOW.STC.010.003	00:01 Mode : Pizza (작동완료 후), d, d, d	02:00 Mode : Bob
MOW.STC.010.004	00:01 Mode : Manual (작동완료 후), a, a, e	00:19 Mode : Manual (작동 중)
MOW.STC.010.005	80C Mode : Manual (작동완료 후), c, b, e	20C Mode : Manual (작동 중)
MOW.STC.010.006	00:01 Mode : Pizza (작동완료 후), d, e	00:59 Mode : Dduck (작동 중)
MOW.STC.011.001	00:19 Mode : Manual (작동 중), a, b, c, a, f	00:14 Mode : Manual (작동 중)
MOW.STC.011.002	20C Mode : Manual (작동 중), a, b, f, c, d	40C Mode : Manual (작동 중)
MOW.STC.011.003	00:59 Mode : Dduck (작동 중), d, f, a, c, b	00:57 Mode : Dduck (작동 중)

Identifier	PASS/FAIL
MOW.STC.010.001	PASS
MOW.STC.010.002	PASS
MOW.STC.010.003	PASS
MOW.STC.010.004	PASS
MOW.STC.010.005	PASS
MOW.STC.010.006	PASS
MOW.STC.011.001	PASS
MOW.STC.011.002	PASS
MOW.STC.011.003	PASS

Cunit Test 변동사항

- Cunit Header 내부에는 printf 등의 함수 동작가능 여부를 Testing할 수 있는 함수가 존재하지 않아, 기존에 Debugging Unit Test Case List의 3,4,5,6은 Test할 수 없었음
- Unit Test Case List의 11은 printf함수를 활용해 Light On/Off를 폰트의 컬러 변환으로 확인하는 것이었으므로 역시 Test가 불가능했음
- State변수와 ABLE변수 모두 기능은 같으나 다른 함수에서 사용하여 혼동의 여지가 있어 ABLE로 통일 변경함
- Bool type인 Trigger, Enable, Disable 변수들이 사용되는 함수 내부에 지역변수로 선언되어 있어, Assert함수로 내부에 변동사항을 비교할 수 없어, 세 변수를 모두 Bool type extern 변수로 변경함

CUnit Test Feature

Identifier	Feature (Process ID in DFD)	Valid / Invalid value
MOW.UTC.00.000	2.1.1 buttoncontroller	State를 받고, Time_TempButtonState가 True이다.
MOW.UTC.00.001	2.1.1 buttoncontroller	State를 받고, ModeButtonState가 True이다.
MOW.UTC.00.002	2.1.1 buttoncontroller	State를 받고, DoorState가 True이다.
MOW.UTC.00.003	2.1.1 buttoncontroller	State를 받고, _10sec_10CButtonState가 True이다.
MOW.UTC.00.004	2.1.1 buttoncontroller	State를 받고, _30sec_20CButtonState가 True이다.
MOW.UTC.00.005	2.1.1 buttoncontroller	State를 받고, S_CButtonState가 True이다.
MOW.UTC.01.000	2.1.13 Time_TempController	Data와 State를 받고, DoorState가 True이고, DOOR가 짝수이다.
MOW.UTC.01.001	2.1.13 Time_TempController	Data와 State를 받고, DoorState가 True이고, DOOR가 홀수이다.
MOW.UTC.01.002	2.1.13 Time_TempController	Data와 State를 받고, DoorState가 False이고, S_CButtonState가 True이고, ABLE이 홀수이고, d.ctime이 40이다.
MOW.UTC.01.003	2.1.13 Time_TempController	Data와 State를 받고, DoorState가 False이고, S_CButtonState가 True이고, ABLE이 홀수이고, d.ctime이 0이다.
MOW.UTC.01.004	2.1.13 Time_TempController	Data와 State를 받고, DoorState가 False이고, S_CButtonState가 True이고, ABLE이 짝수이고, d.ctemp이 40이다.
MOW.UTC.01.005	2.1.13 Time_TempController	Data와 State를 받고, DoorState가 False이고, S_CButtonState가 True이고, ABLE이 짝수이고, d.ctime이 0이다.

Identifier	Feature (Process ID in DFD)	Valid / Invalid value
MOW.UTC.02.000	2.1.17 DisplayController	Data와 State를 받고, ABLE이 홀수 이다.
MOW.UTC.02.001	2.1.17 DisplayController	Data와 State를 받고, ABLE이 짝수 이다.
MOW.UTC.07.000	2.1.2 10secIncrease	Trigger 받고, d.stime이 30이다.
MOW.UTC.08.000	2.1.3 10CIncrease	Trigger 받고, d.stemp이 40이다.
MOW.UTC.09.000	2.1.4 30secIncrease	Trigger 받고, d.stime이 90이다.
MOW.UTC.10.000	2.1.5 20CIncrease	Trigger 받고, d.stemp이 50이다.
MOW.UTC.11.000	2.1.13 Light	Enable 이 True고, Disable 이 False 이다.
MOW.UTC.11.001	2.1.13 Light	Enable 이 False고, Disable 이 True 이다.
MOW.UTC.12.000	2.1.14 Make_Sound	Trigger 받는다.
MOW.UTC.13.000	2.1.15 TempIncreasing	Trigger 받고, d.ctemp는 50이다.
MOW.UTC.14.000	2.1.16 TimeDecreasing	Trigger 받고, d.ctime는 80이다.

Cunit Test Case / Code

```
#include "CUnit/Basic.h"
#include "T3_main_for_gcc.h"

void MOW.UTC_00_000() {
    s.Time_TempButtonState = T;
    ABLE = 1;

    buttoncontroller (&s);

    CU_ASSERT_EQUAL(d.stime, 0);
}
```

Identifier	Input specification	Output specification
MOW.UTC.00.000	State in / Time_TempButtonState == True / state%2=0	d.stime = 0


```

void MOW.UTC_00_001() {
    s.Time_TempButtonState = T;
    ABLE = 2;

    buttoncontroller (&s);

    CU_ASSERT_EQUAL(d.stemp, 20);
}

```

Identifier	Input specification	Output specification
MOW.UTC.00.001	State in / Time_TempButtonState == True / state%2=1	d.stemp = 20

```

void MOW.UTC_00_002() {
    s.ModeButtonState = T;
    ABLE = 1;
    d.i_cmode = 5;

    buttoncontroller (&s);

    CU_ASSERT_EQUAL(d.stime, 0);
}

```

Identifier	Input specification	Output specification
MOW.UTC.00.002	State in / ModeButtonState == True / (state % 2) != 0 / mode = 5	d.stime = 0

```

void MOW.UTC_00_003() {
    s._10sec_10CButtonState = T;
    ABLE = 1;

    buttoncontroller (&s);

    CU_ASSERT_EQUAL(trigger, T);
}

```

Identifier	Input specification	Output specification
MOW.UTC.00.003	State in / _10sec_10CButtonState == True / state%2=1	_10secIncrease(T)

```

void MOW.UTC_00_004() {
    s._10sec_10CButtonState = T;
    ABLE = 2;

    buttoncontroller (&s);

    CU_ASSERT_EQUAL(trigger, T);
}

```

Identifier	Input specification	Output specification
MOW.UTC.00.004	State in / _10sec_10CButtonState == True / state%2=0	_10CIncrease(T)

```

void MOW.UTC_00_005() {
    s._30sec_20CButtonState = T;
    ABLE = 1;

    buttoncontroller (&s);

    CU_ASSERT_EQUAL(trigger, T);
}
void MOW.UTC_00_006() {
    s._30sec_20CButtonState = T;
    ABLE = 2;

    buttoncontroller (&s);

    CU_ASSERT_EQUAL(trigger, T);
}

```

Identifier	Input specification	Output specification
MOW.UTC.00.005	State in / _30sec_20CButtonState == True / state%2=1	_30secIncrease(T)
MOW.UTC.00.006	State in / _30sec_20CButtonState == True / state%2=0	_10CIncrease(T)

```

void MOW.UTC_01_000() {
    s.DoorState = T;
    DOOR = 1;

    Time_TempController(&d, &s);

    CU_ASSERT_EQUAL(enable, T);
}
void MOW.UTC_01_001() {
    s.DoorState = T;
    DOOR = 2;

    Time_TempController(&d, &s);

    CU_ASSERT_EQUAL(disable, T);
}

```

Identifier	Input specification	Output specification
MOW.UTC.01.000	Data in / State in / DoorState == True / DOOR%2==0	enable = T
MOW.UTC.01.001	Data in / State in / DoorState == True / DOOR%2==1	disable = T

```

void MOW.UTC_01_002() {
    s.DoorState = F;
    s.S_CButtonState = T;
    ABLE = 1;
    d.ctime = 40;

    Time_TempController(&d, &s);

    CU_ASSERT_EQUAL(d.ctime, 39);
}
void MOW.UTC_01_003() {
    s.DoorState = F;
    s.S_CButtonState = T;
    ABLE = 1;
    d.ctime = 0;

    Time_TempController(&d, &s);

    CU_ASSERT_EQUAL(trigger, T);
}

```

Identifier	Input specification	Output specification
MOW.UTC.01.002	Data in / State in / DoorState == False / S_CButtonState == True / ABLE%2==1 / d.ctime = 40	Light(T, F) / TimeDecreasing(T)
MOW.UTC.01.003	Data in / State in / DoorState == False / S_CButtonState == True / ABLE%2==1 / d.ctime = 0	Light(F, T) / Make_Sound(T)

```

void MOW.UTC_01_004() {
    s.DoorState = F;
    s.S_CButtonState = T;
    ABLE = 2;
    d.ctemp = 20;
    d.stemp = 50;

    Time_TempController(&d, &s);

    CU_ASSERT_EQUAL(d.ctemp, 30);
}
void MOW.UTC_01_005() {
    s.DoorState = F;
    s.S_CButtonState = T;
    ABLE = 2;
    d.ctemp = 40;
    d.stemp = 40;

    Time_TempController(&d, &s);

    CU_ASSERT_EQUAL(trigger, T);
}

```

Identifier	Input specification	Output specification
MOW.UTC.01.004	Data in / State in / DoorState == False / S_CButtonState == True / ABLE%2==0 / d.ctemp = 40 / d.stemp = 40	Light(F, T) / Make_Sound(T)
MOW.UTC.01.005	Data in / State in / DoorState == False / S_CButtonState == True / ABLE%2==0 / d.ctemp = 20 / d.stemp = 50	Light(T, F) / TempIncreasing(T)


```

void MOW.UTC_02_000(void) {
    ABLE = 1;

    DisplayController(&d, &s);

    CU_ASSERT_EQUAL(enable, T);
}

void MOW.UTC_02_001(void) {
    ABLE = 2;

    DisplayController(&d, &s);

    CU_ASSERT_EQUAL(enable, T);
}

```

Identifier	Input specification	Output specification
MOW.UTC.02.000	Data in / State in / ABLE%2==1	TimePrint(T, F) / ModePrint(T, F)
MOW.UTC.02.001	Data in / State in / ABLE%2==0	TempPrint(T, F) / ModePrint(T, F)

```

void MOW.UTC_07_000(void) {
    trigger = T;
    d.stime = 30;

    _10secIncrease(T);

    CU_ASSERT_EQUAL(d.stime, 40);
}
void MOW.UTC_08_000(void) {
    trigger = T;
    d.stemp = 40;

    _10CIncrease(T);

    CU_ASSERT_EQUAL(d.stemp, 50);
}

```

Identifier	Input specification	Output specification
MOW.UTC.07.000	Trigger in / d.stime = 30	d.stime = 40
MOW.UTC.08.000	Trigger in / d.stemp = 40	d.stemp = 50

```

void MOW.UTC_09_000(void) {
    trigger = T;
    d.stime = 90;

    _30secIncrease(T);

    CU_ASSERT_EQUAL(d.stime, 120);
}
void MOW.UTC_10_000(void) {
    trigger = T;
    d.stemp = 50;

    _20CIncrease(T);

    CU_ASSERT_EQUAL(d.stemp, 70);
}

```

Identifier	Input specification	Output specification
MOW.UTC.09.000	Trigger in / d.stime = 90	d.stime = 120
MOW.UTC.10.000	Trigger in / d.stemp = 50	d.stemp = 70

```

void MOW.UTC_13_000(void) {
    trigger = T;
    d.ctemp = 50;

    TempIncreasing(T);

    CU_ASSERT_EQUAL(d.ctemp, 60);
}
void MOW.UTC_14_000(void) {
    trigger = T;
    d.ctime = 80;

    TimeDecreasing(T);

    CU_ASSERT_EQUAL(d.ctime, 79);
}

```

Identifier	Input specification	Output specification
MOW.UTC.13.000	Trigger in / d.ctemp = 50	DisplayController / d.ctemp = 60
MOW.UTC.14.000	Trigger in / d.ctime = 80	DisplayController / d.ctime = 79

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

Suite: testing a suite

Test: MDW.UTC_00_000 ... passed
Test: MDW.UTC_00_001 ... passed
Test: MDW.UTC_00_002 ... passed
Test: MDW.UTC_00_003 ... passed
Test: MDW.UTC_00_004 ... passed
Test: MDW.UTC_00_005 ... passed
Test: MDW.UTC_00_006 ... passed
Test: MDW.UTC_01_000 ... **passed**
Test: MDW.UTC_01_001 ... passed
Test: MDW.UTC_01_002 ... **passed**
Test: MDW.UTC_01_003 ... passed
Test: MDW.UTC_01_004 ... **passed**
Test: MDW.UTC_01_005 ... passed
Test: MDW.UTC_02_000 ... passed
Test: MDW.UTC_02_001 ... passed
Test: MDW.UTC_07_000 ... passed
Test: MDW.UTC_08_000 ... passed
Test: MDW.UTC_09_000 ... passed
Test: MDW.UTC_10_000 ... passed
Test: MDW.UTC_13_000 ... passed
Test: MDW.UTC_14_000 ... passed

Run Summary:

Type	Total	Ran	Passed	Failed	Inactive
suites	1	1	n/a	0	0
tests	21	21	21	0	0
asserts	21	21	21	0	n/a

Elapsed time = 0.000 seconds

Result

Thank you😊