

SASD

# Microwave Oven System

[T1]

201211325 권세미

201211357 송인희

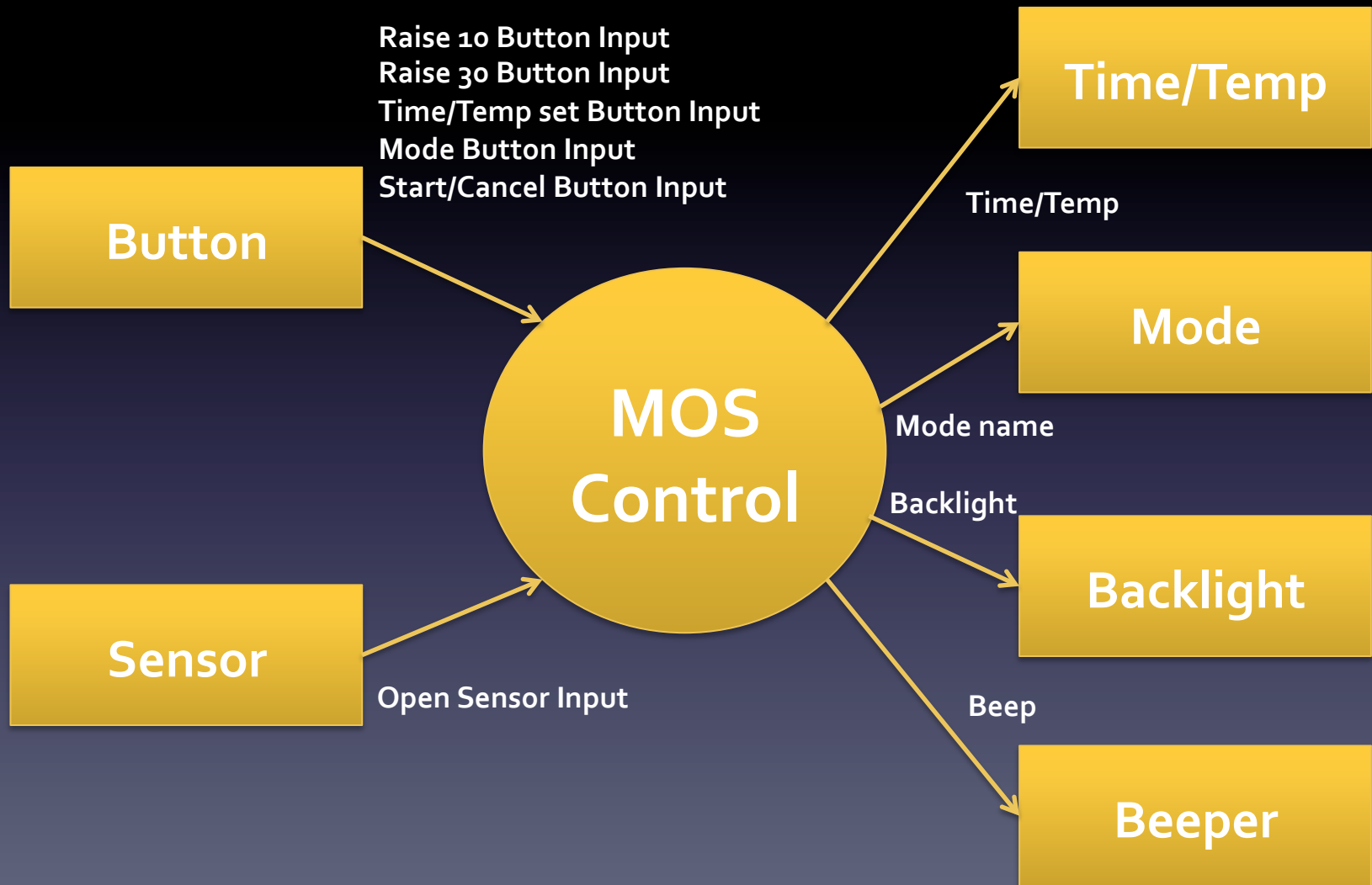
201213618 이진진

201211377 임하림

# Statement of Purpose

- MOS(Microwave oven System)는 지정된 시간 또는 온도를 기준으로 하여 설정된 시간 및 목표 온도에 도달할 동안 음식물을 데워준다.
- 사용자가 버튼을 눌러 시간 및 온도를 설정할 수 있으며, 조리할 음식을 선택하면 자동으로 시간이 설정된다.
- 설정된 시간 및 현재 온도는 화면에 표시되며 조리 시간이 경과함에 따라 점차 줄어들고, 높아진다. 시간이 00:00이 되거나 온도가 목표온도에 도달할 경우 조리가 완료된다.
- 문을 열거나 조리중일 때 Backlight가 켜지며, 조리가 끝나면 backlight가 꺼지고 3초간 beep음이 나서 조리의 완료를 알 수 있다.
- HW(MOS)와 연동까지는 고려하지 않으며, SW로만 구성하여 시뮬레이션 용도로 사용한다.

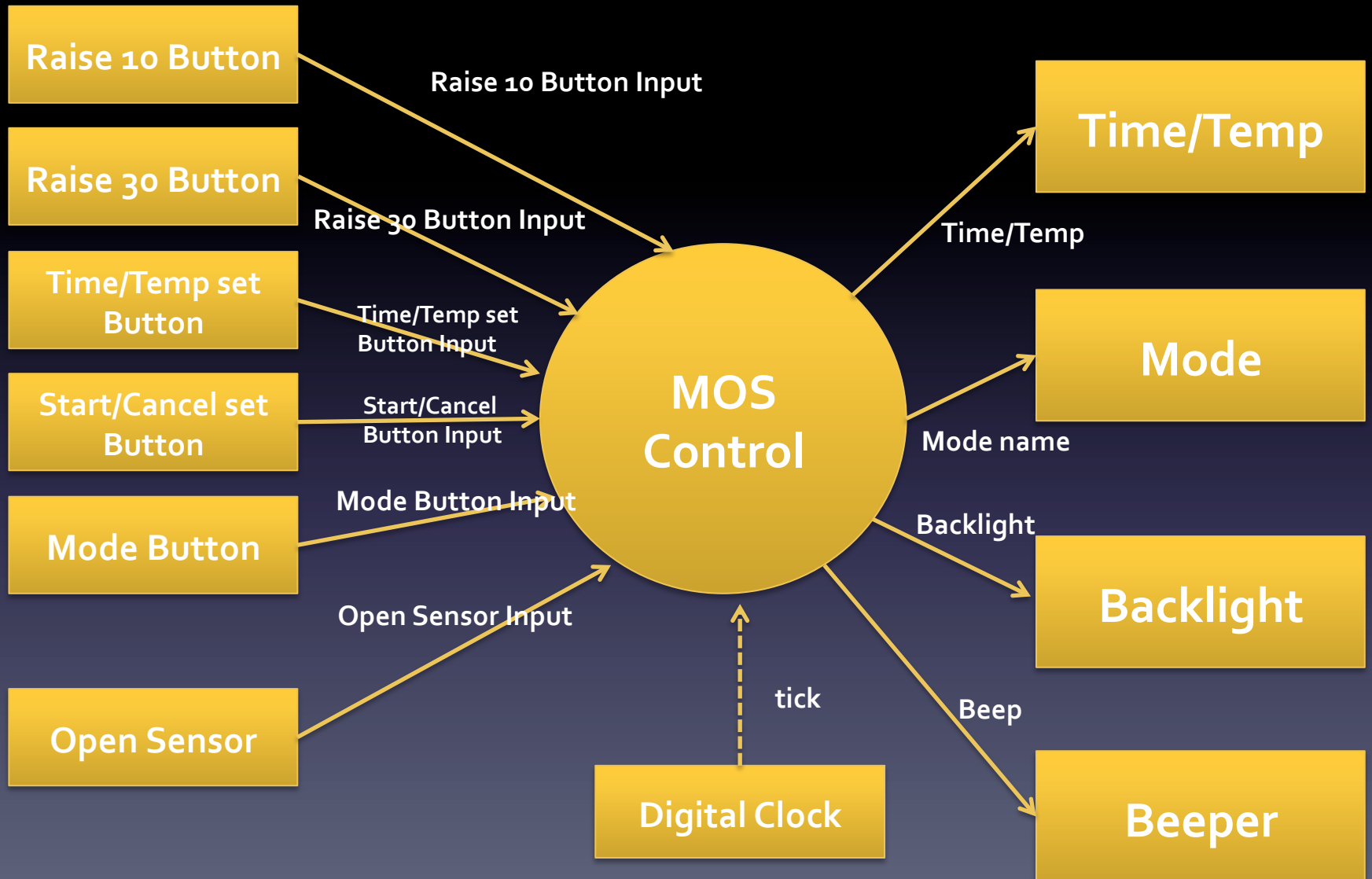
# System Context Diagram



# Event list

Input / Output Event	Description	Format/Type
Raise 10 Button Input	한번 누를 때마다 설정 시간·온도가 10씩 증가	Char / Interrupt
Raise 30 Button Input	한번 누를 때마다 설정 시간이 30초, 온도가 20도씩 증가	Char / Interrupt
Time/Temp set Button Input	시간과 온도 중 무엇을 조절할 것인지 입력 받음 누를 때마다 설정이 바뀜 (초기값: 시간)	Char / Interrupt
Mode Button Input	선택한 모드에 따라 정해진 시간만큼 설정됨	Char / Interrupt
Start/Cancel Button Input	조리 전: 시작 / 조리 중: 취소	Char / Interrupt
Open Sensor Input	사용자의 입력에 따른 문의 열림 상태 감지	Char / Interrupt
Time/Temp	조리 전: Time/Temp set Button의 상태에 따라 display 되는 설정된 시간이나 온도 조리 중: 시작했을때 Time/Temp set Button 의 상태에 따라 display 되는 현재 남은 시간이나 온도	Char[]
Mode name	선택된 조리모드의 번호와 명칭	Char[]
Backlight	문 열림·조리 중: 켜짐 / 문 닫힘·조리 끝: 꺼짐	On/Off
Beep	조리 끝: 3초간 켜짐 / 그외: 꺼짐	On/Off

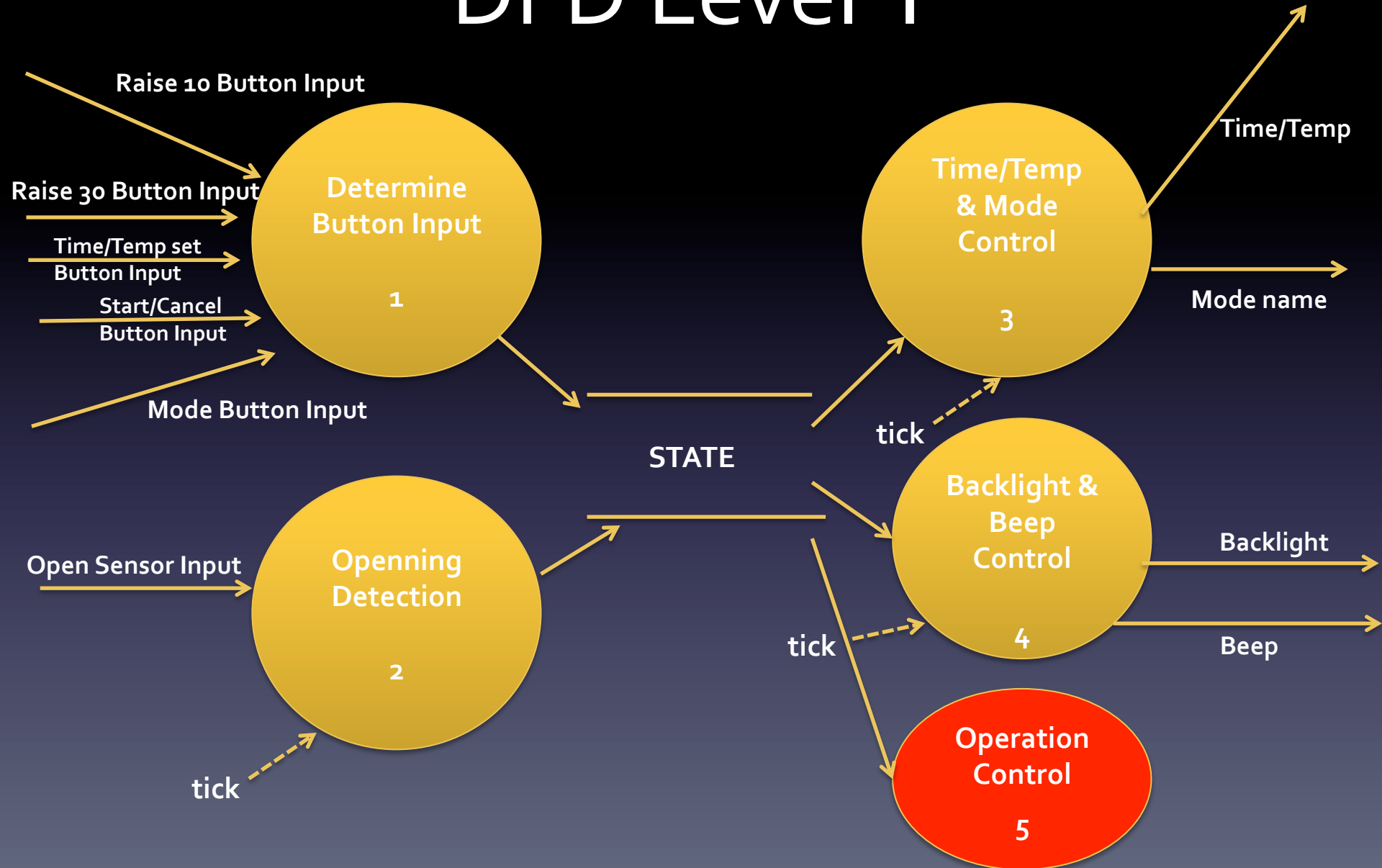
# DFD Level 0



# DFD Level 0 Data Dictionary

Input / Output Event	Description
tick	1 tick당 0.1초 증가

# DFD Level 1

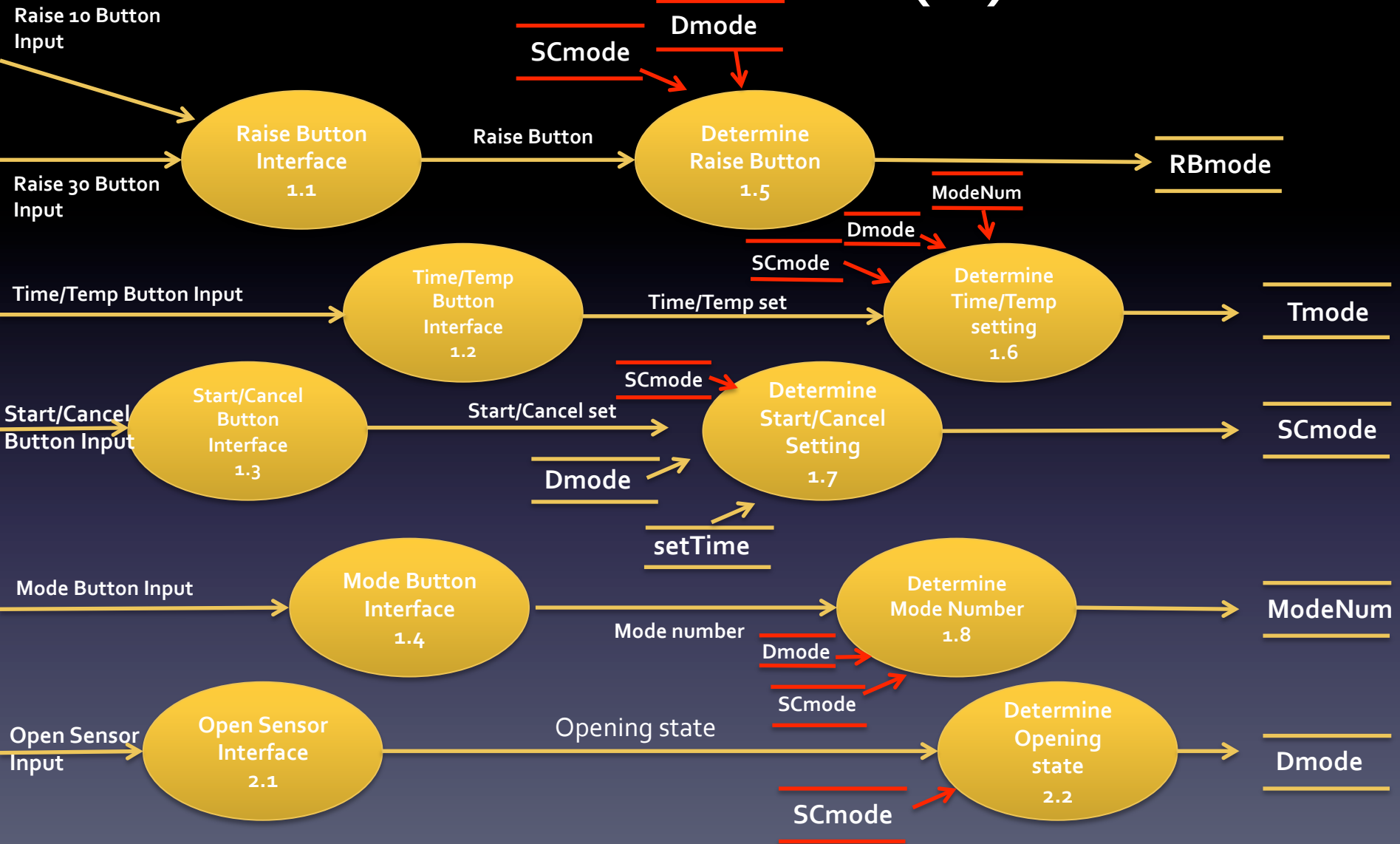


# DFD Level 1 Data Dictionary

Input / Output Event	Description	Format/Type
STATE	설정된 시간/온도 그리고 모드의 저장 값 (RBMode, Tmode, SCmode, ModeNum,Dmode)	struct



# DFD Level 2 (1)



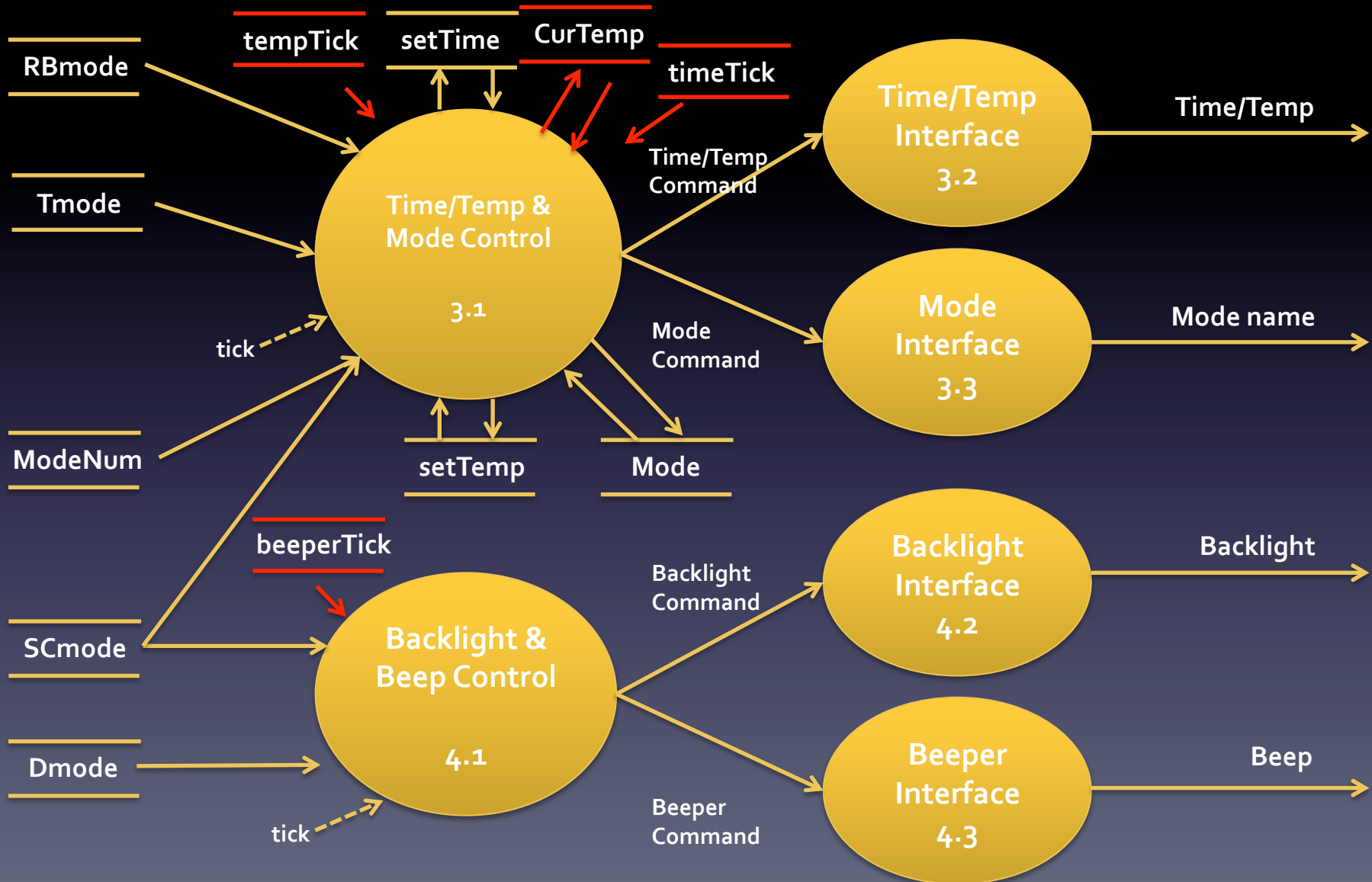
```
switch(key)
{
    case 'a' : Raise_Button = Raise_Button_Interface(key);
        state->RBmode = Determine_Raise_Button(Raise_Button, state->SCmode, state->Dmode, state->RBmode);
        break;
    case 'b' : Raise_Button = Raise_Button_Interface(key);
        state->RBmode = Determine_Raise_Button(Raise_Button, state->SCmode, state->Dmode, state->RBmode);
        break;

    case 'c' : Time_Temp_set = Time_Temp_Button_Interface(key);
        state->Tmode = Determine_Time_Temp_setting(*state, time_temp, Time_Temp_set, state->Tmode, state->SCmode, state->Dmode);
        break;
    case 'd' : Mode_Number = Mode_Button_Interface(key);
        state->ModeNum = Determine_Mode_Number(Mode_Number, state->ModeNum, state->SCmode, setTime, state->Dmode);
        break;
    case 'e' : Start_Cancel_set = Start_Cancel_Button_Interface(key);
        state->SCmode = Determine_Start_Cancel_Setting(Start_Cancel_set, state->SCmode, state->Dmode, setTime);
        break;
    case 'f' : Opening_state = Open_Sensor_Interface(key);
        state->Dmode = Determine_Opening_state(Opening_state, state->SCmode, state->Dmode);
        break;
}
```

# DFD Level 2(1) Data Dictionary

Input / Output Event	Description	Format/Type
Raise Button	사용자가 누른 Raise Button(키보드 A, 키보드 B)이 char 형으로 변환 된 것	char
Time/Temp set	Time/Temp Button(키보드 C)이 char형으로 변화 된 것	Char
Start/Cancel set	사용자가 누른 Start/Cancel Button(키보드E )이 char 형으로 변환된 것	Char
setTime	조리 전 : RBmode와 Tmode / setTemp 또는 mode에 의해 설정된 시간 저장 값 조리 중 : tick에 따라 감소하는 시간	Integer
Mode Number	Mode 버튼(키보드D)이 char형으로 변화 된 것	char
Opening state	사용자가 누른 Open Button(키보드 F)이 char 형으로 변환된 것	char
RBmode	선택된 Raise Button의 결과 값 (0 : Raise 10/10(Time/Temp), 1 : Raise 30/20(Time/Temp), 3 : 놀리지 않은 상태)	integer
Tmode	선택된 Time/Temp 모드의 저장 값 (0 : Time, 1 : Temp)	integer
SCmode	선택된 Start/Cancel의 저장 값 (0 : Cancel, 1 : Start, 2 : Finish, 4 : 조리중에 취소로 바꾼 상태 )	integer
ModeNum	선택할 모드의 번호 (0: 메뉴얼, 1: 떡, 2: 죽, 3: 밥, 4: 국/찌개, 5: 피자)	integer
Dmode	문의 열림/닫힘 상태 저장 값 (0:문 닫힘, 1:문 열림)	integer

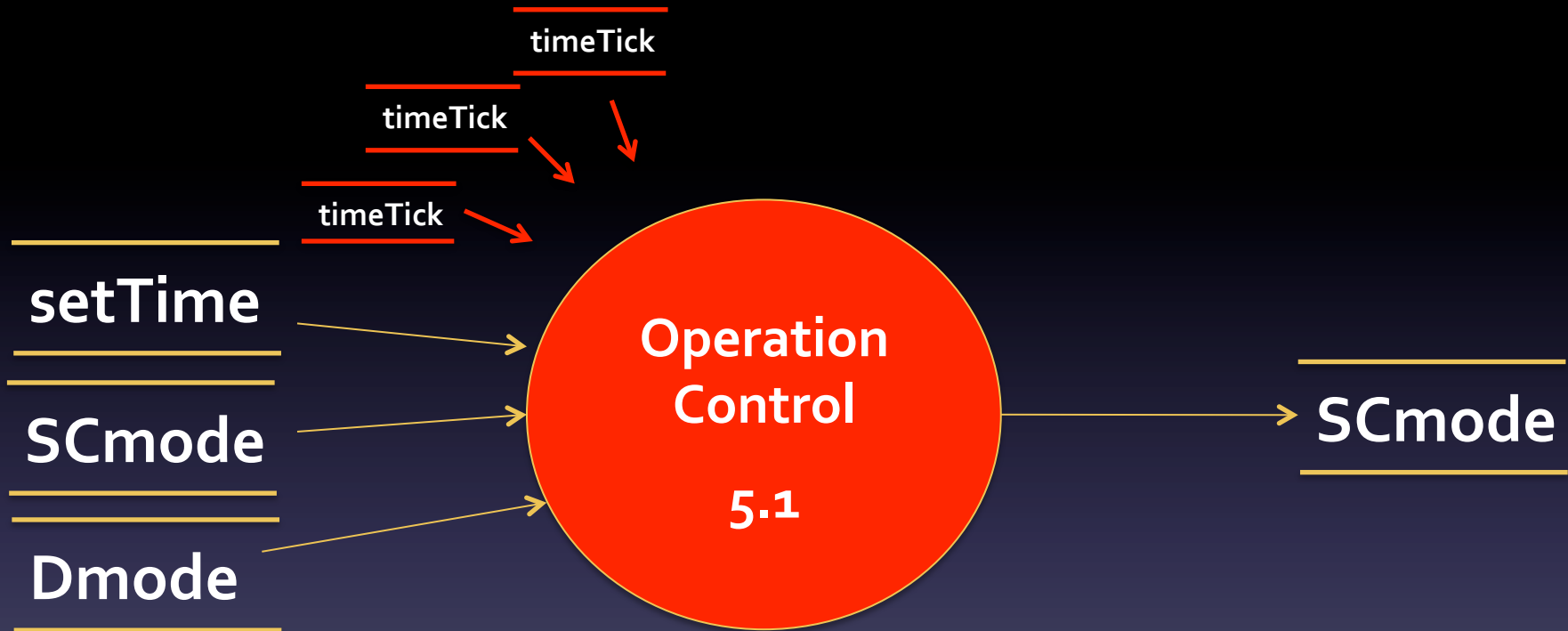
# DFD Level 2 (2)

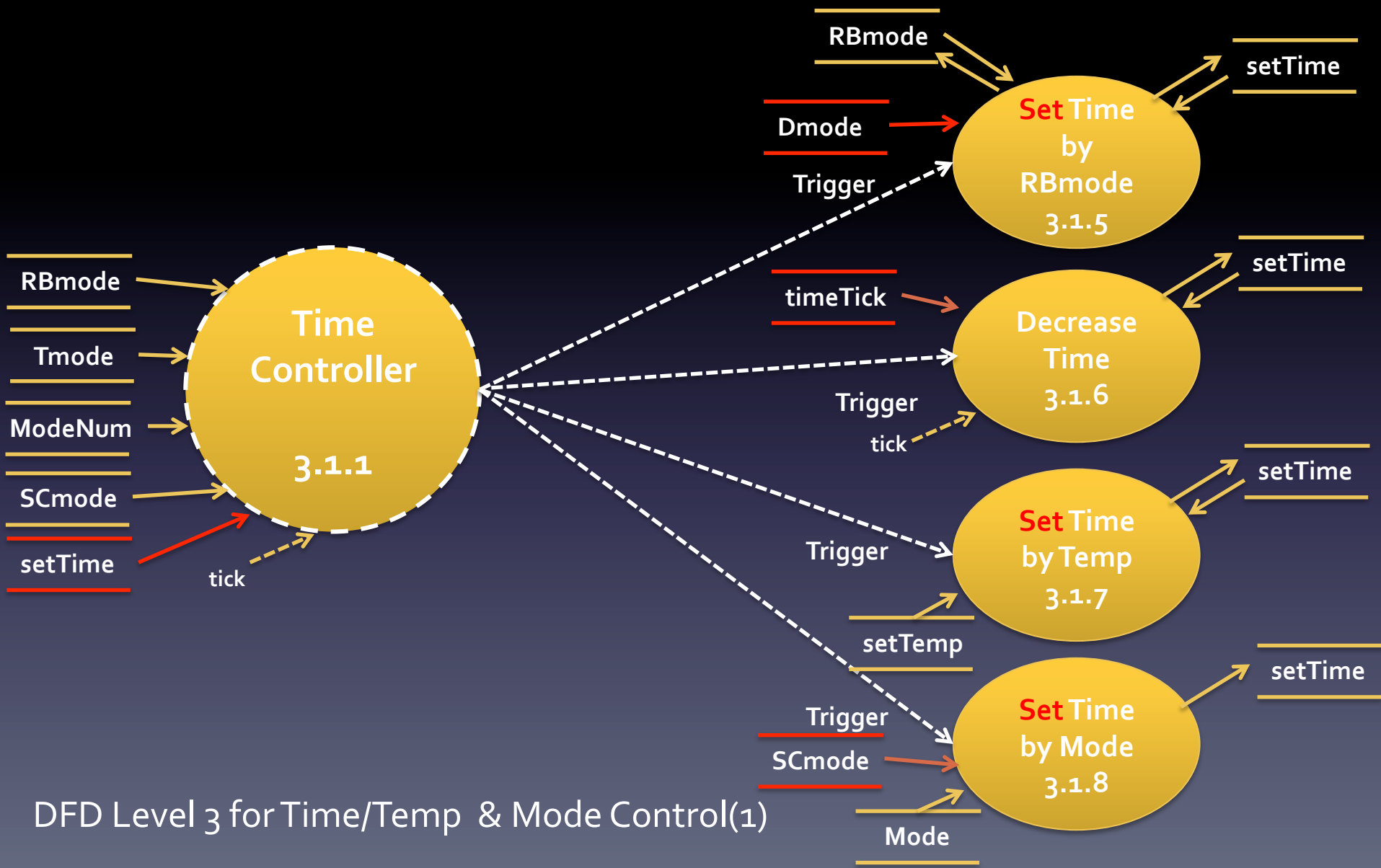


# DFD Level 2(2) Data Dictionary

Input / Output Event	Description	Format/Type
Mode	ModeNum에 따라 선택된 모드의 정보를 담는 구조체 공간 (모드 번호(0~5:integer), 모드 이름(char[]), 조리시간(integer))	struct
CurTemp	감지된 현재 온도의 저장 값	integer
setTemp	RBmode와 Tmode / setTemp 또는 mode에 의해 설정된 시간 저장 값	integer
Time/Temp Command	취합된 시간과 온도 정보를 Display하기 위해 Interface로 보내는 Command (시그윈: char[], visual: int[])	Integer[]/char[]
Mode Command	Mode 정보를 Display하기 위해 Interface로 보내는 Command	char[]
Backlight Command	Backlight interface에 Backlight의 on/off를 명령해주는 변수 (0: off/ 1: on)	integer
Beeper Command	Beeper interface에 Beep의 on을 명령해주는 변수 ( 1: on)	integer

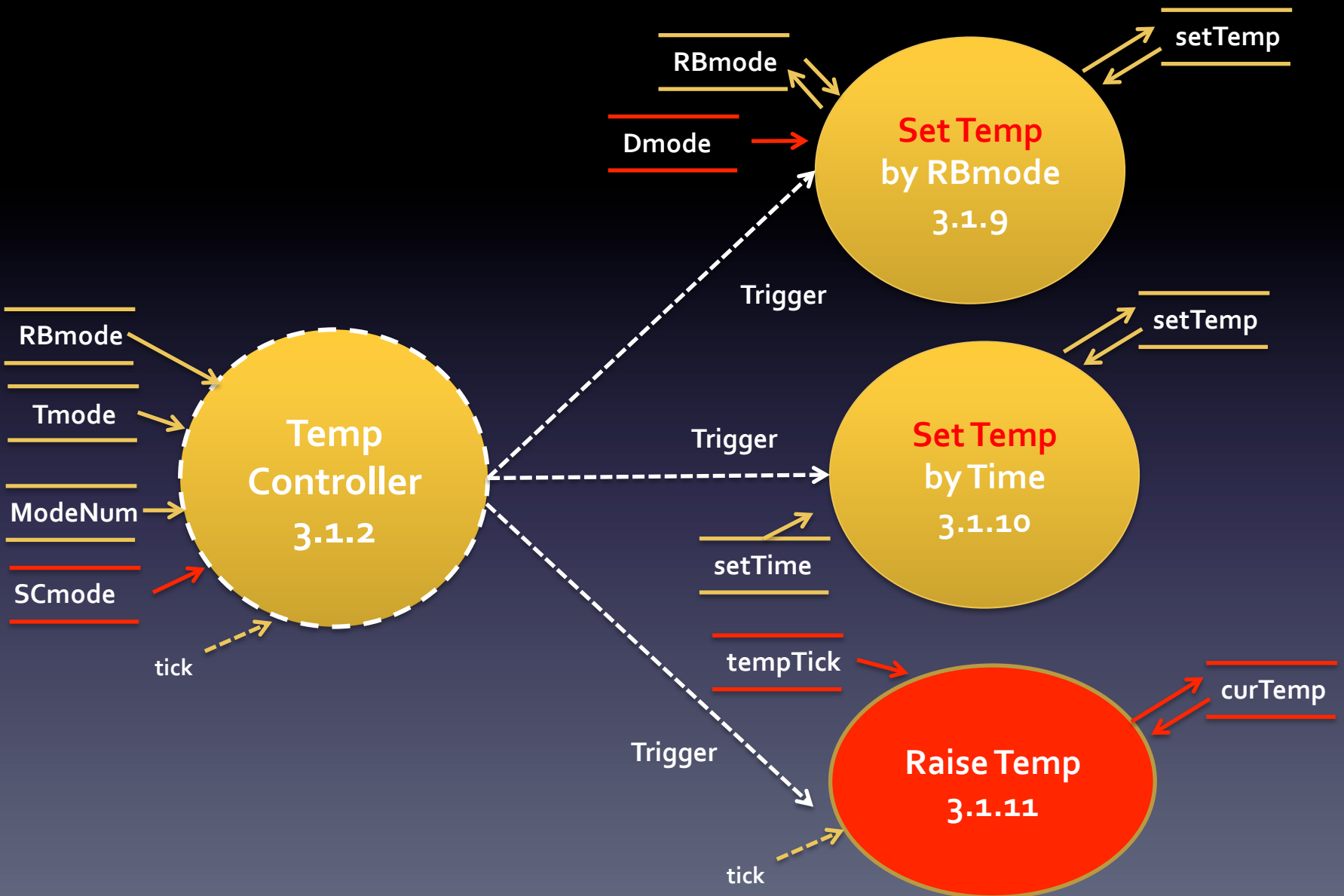
# DFD Level 2(3)





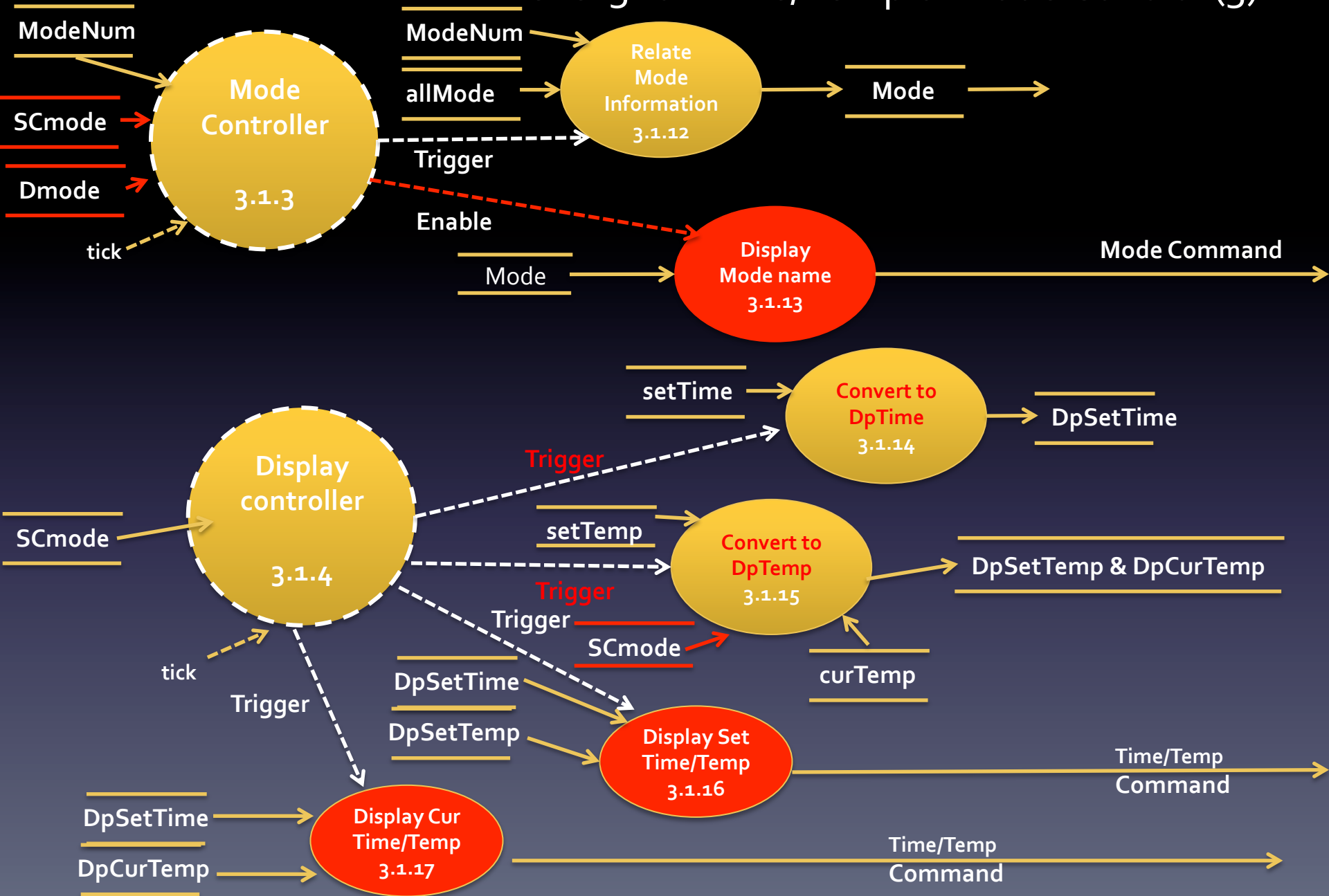
DFD Level 3 for Time/Temp & Mode Control(1)

# DFD Level 3 for Time/Temp & Mode Control(2)





# DFD Level 3 for Time/Temp & Mode Control (3)

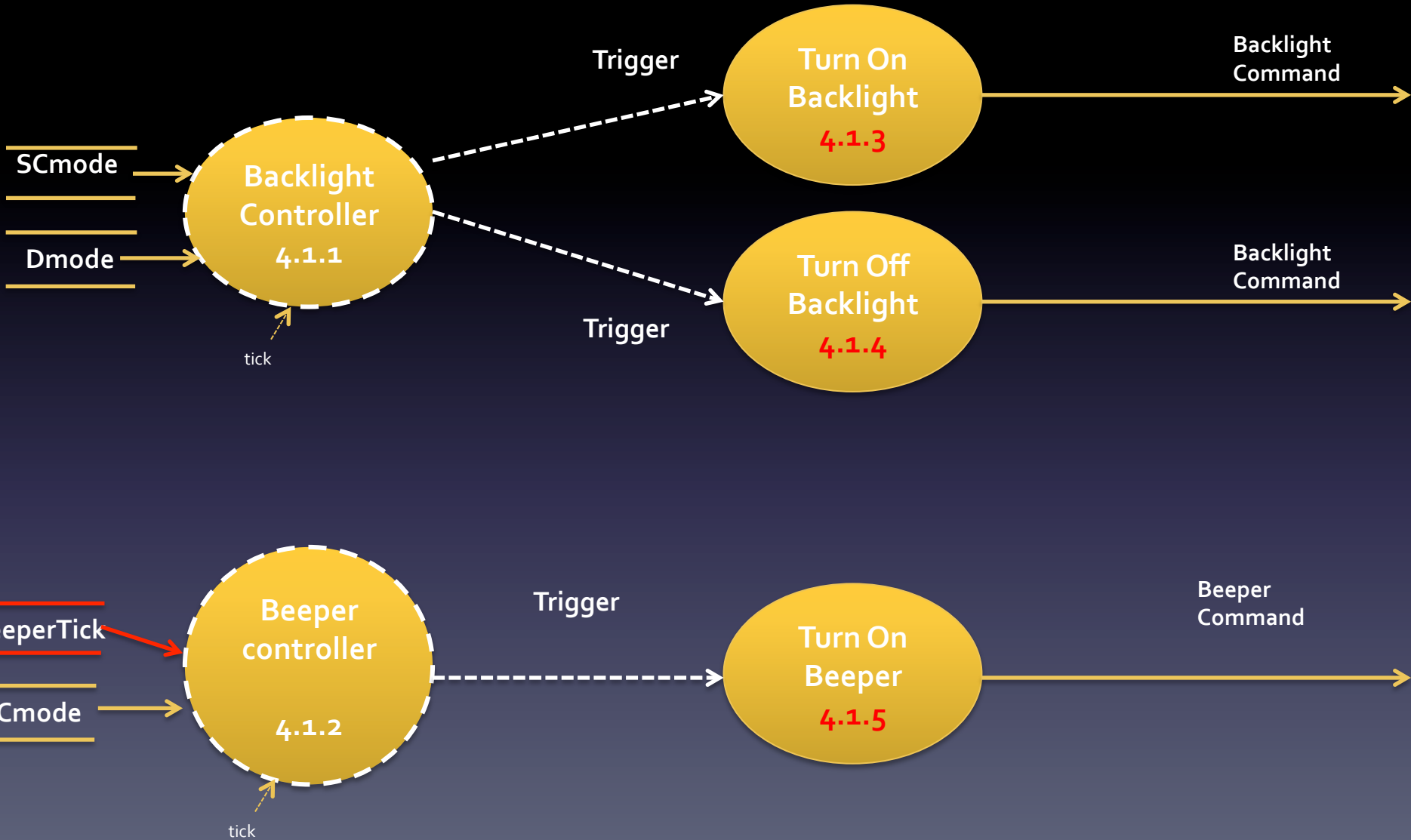


# DFD Level 3 for Time/Temp & Mode Control (3)

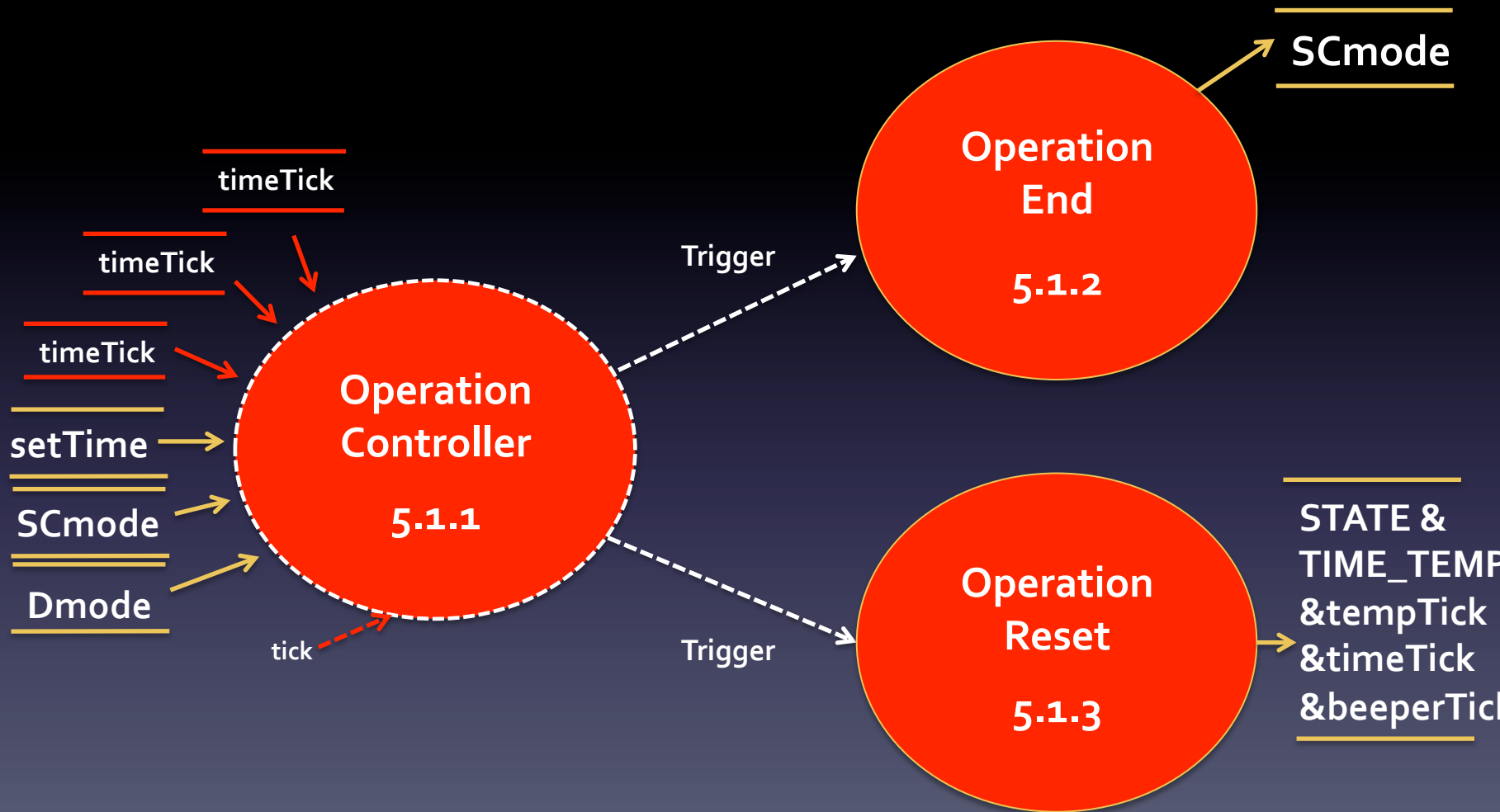
## Data Dictionary

Input / Output Event	Description	Format/Type
allMode	<p>모든 모드의 정보가 담겨있는 구조체 배열</p> <p>allMode[0]: Manual Mode 구조체  allMode[1]: 떡 Mode 구조체  allMode[2]: 죽 Mode 구조체  allMode[3]: 밥 Mode 구조체  allMode[4]: 국/찌개 Mode 구조체  allMode[5]: 피자 Mode 구조체</p>	Struct[]
DpSetTime	Integer형 setTime이 배열로 display될 수 있도록 배열로 변경하여 저장된 값 (시그윈: char[] / visual : int[])	Integer[]/char[]
DpSetTemp & DpCurTemp	Integer형 setTemp와 CurTemp가 배열로 display될 수 있도록 배열로 변경하여 저장된 값(시그윈: char[] / visual : int[])	Integer[]/char[]
timeTick	Time controller에 작용하는 tick의 연산에 쓰이는 변수	Integer
tempTick	Temp controller에 작용하는 tick의 연산에 쓰이는 변수	Integer
beepTick	Beeper controller에 작용하는 tick의 연산에 쓰이는 변수	Integer

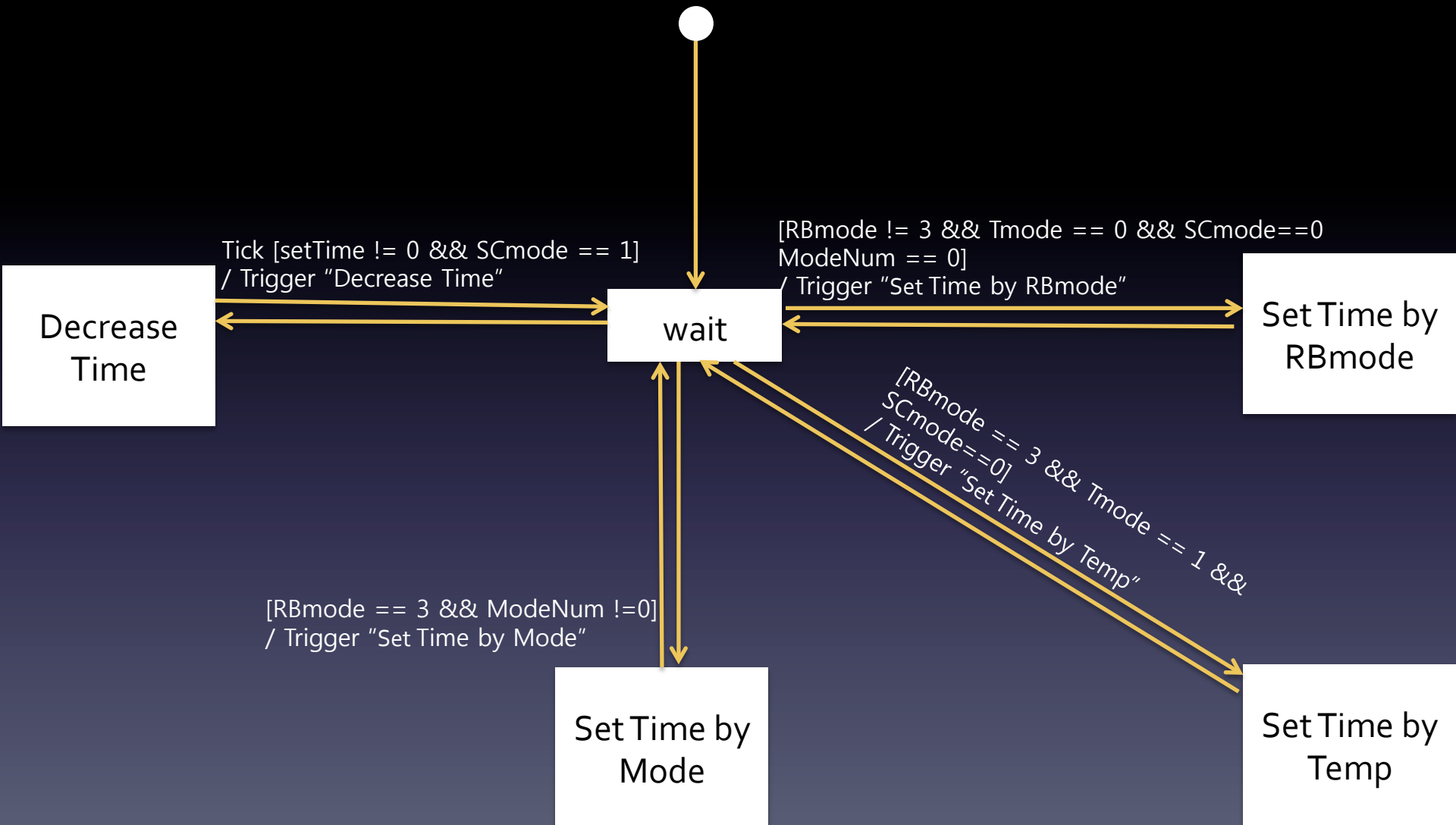
# DFD Level 3 for Backlight & Beep Control



# DFD Level 3 for Operation Control



# State Transition Diagram for Time Controller(3.1.1)



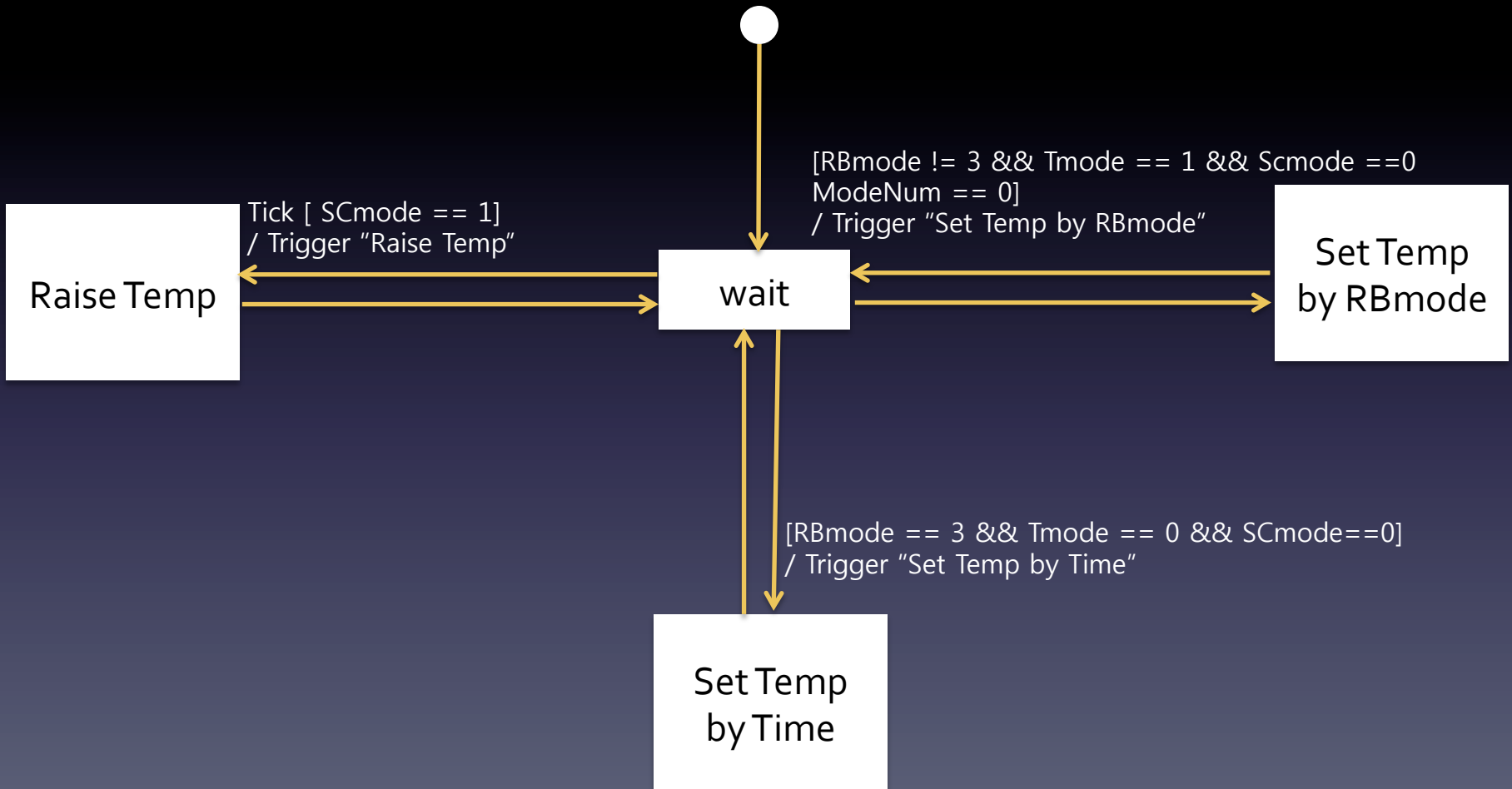
```
if(time_temp->setTime != 0 && state->SCmode == 1)
    time_temp->setTime = Decrease_Time(*time_temp, timeTick);

else if(state->RBmode != 3 && state->Tmode == 0 && state->SCmode == 0 && Mode.modeNum == 0)
    time_temp->setTime = Set_Time_by_RBmode(state, *time_temp);

else if(state->RBmode == 3 && state->Tmode == 1 && state->SCmode == 0)
{
    int STATE::RBmode
    if(state->M
        time_temp->setTime = Set_Time_by_Temp(*time_temp);
        //Tmode == 0일 때 Mode값 넘겨주는 부분
    else
        time_temp->setTime = Set_Time_by_Mode(state, *time_temp, Mode);
}

//Tmode == 1일때 Mode값 넘겨주는부분
else if((state->RBmode == 3 && state->ModeNum != 0))
    time_temp->setTime = Set_Time_by_Mode(state, *time_temp, Mode);
```

# State Transition Diagram for Temp Controller(3.1.2)



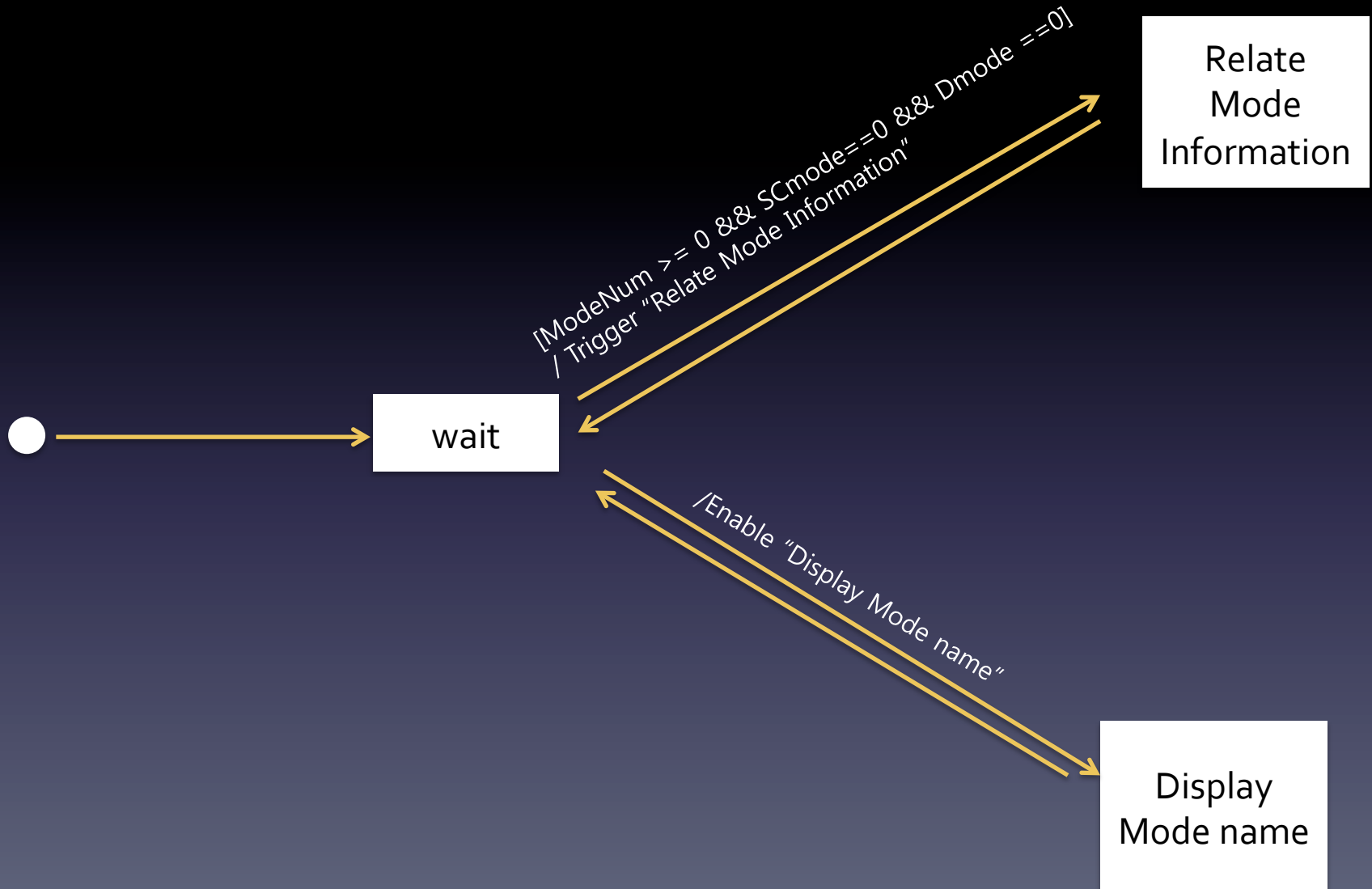
```
if(state->SCmode == 1)
{
    time_temp->curTemp= Raise_Temp(time_temp, tempTick);
}

else if(state->RBmode != 3 && state->Tmode == 1 && state->SCmode == 0 && Mode.modeNum == 0)
{
    time_temp->setTemp= Set_Temp_by_RBmode(state,time_temp);
}

else if(state->RBmode == 3 && state->Tmode == 0 && state->SCmode == 0)
{
    time_temp->setTemp= Set_Temp_by_Time(time_temp);
}
```



# State Transition Diagram for Mode Controller(3.1.3)



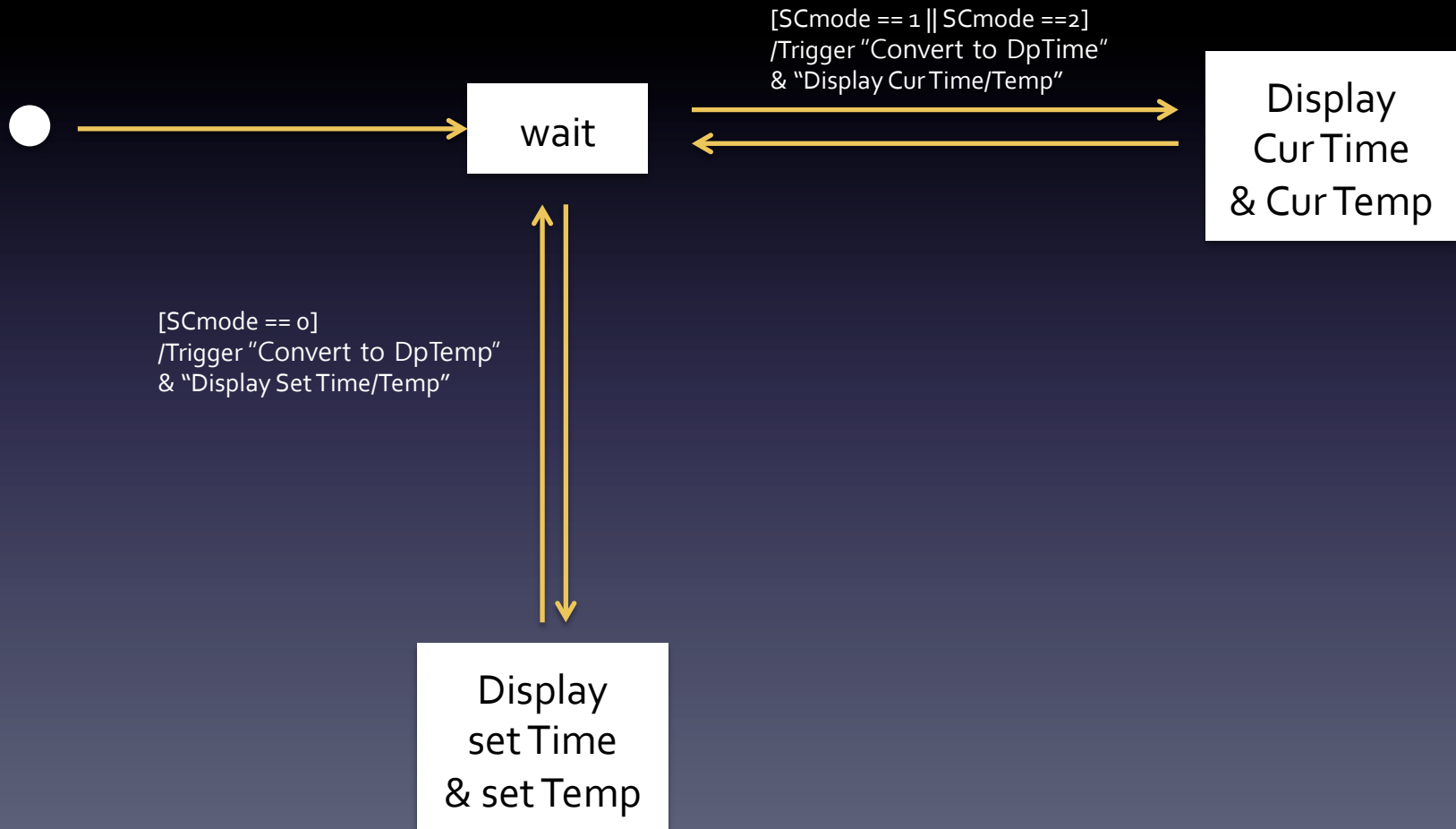
```
char* mode_command;
```

```
if((*Mode)->modeNum >= 0 && SCmode == 0 && Dmode == 0)  
    (*Mode) = Relate_Mode_Information(ModeNum, allMode);
```

```
mode_command = Display_Mode_Name((Mode));
```

```
return mode_command;
```

# State Transition Diagram for Display Controller(3.1.4)



```
display->DpSetTime = Convert_to_DpTime(time_temp);
if(SCmode == 0){
    display->DpSetTemp = Convert_to_DpTemp(SCmode, time_temp);/*
    command->Time_Temp_Command[4] = display->DpSetTemp[0];
    command->Time_Temp_Command[5] = display->DpSetTemp[1];*/
}

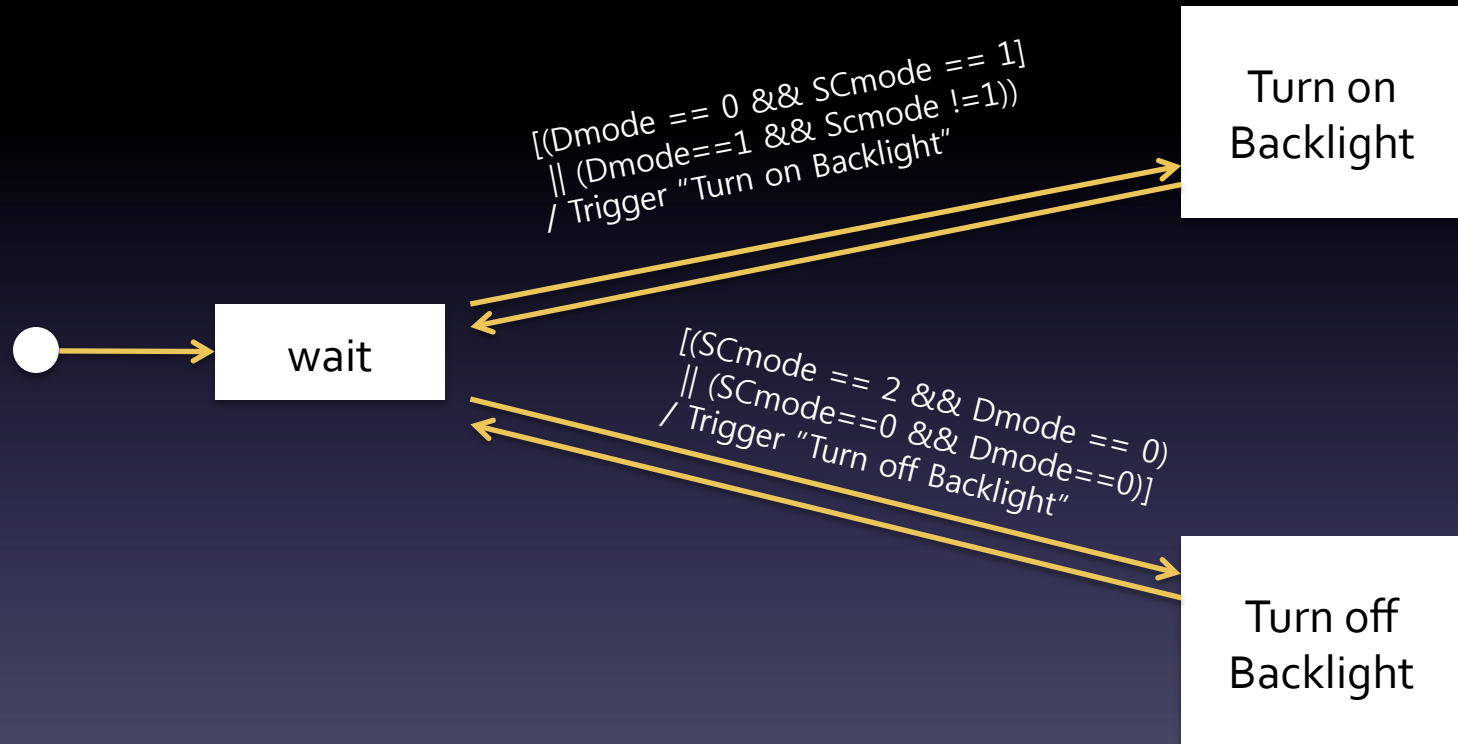
command->Time_Temp_Command = Display_Set_Time_Temp(display);

}

else if(SCmode == 1 || SCmode == 2){
    display->DpCurTemp = Convert_to_DpTemp(SCmode, time_temp);
    command->Time_Temp_Command = Display_Cur_Time_Temp(display);
}

return command->Time_Temp_Command;
```

# State Transition Diagram for Backlight Controller(4.1.1)

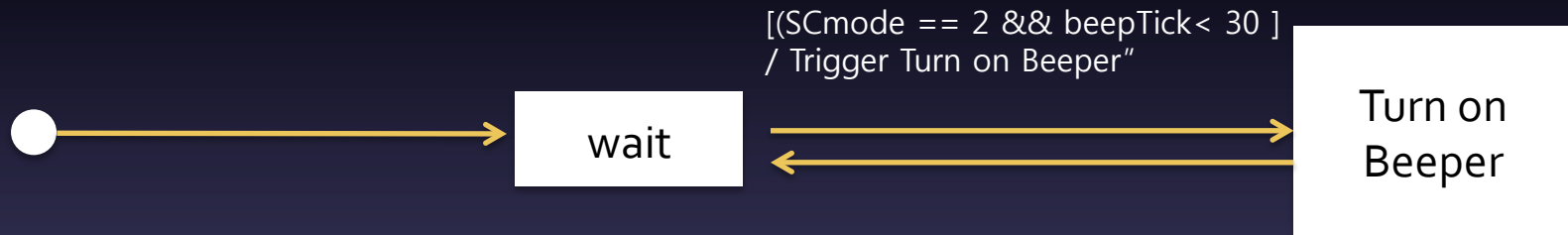


```
int backlight_command = 0;

if((Dmode == 0 && SCmode == 1) || (Dmode == 1 && SCmode != 1))
    backlight_command = Turn_On_Backlight();
else if((SCmode == 2 && Dmode == 0) || (SCmode == 0 && Dmode == 0))
    backlight_command = Turn_Off_Backlight();

return backlight_command;
```

# State Transition Diagram for Beeper Controller(4.1.2)



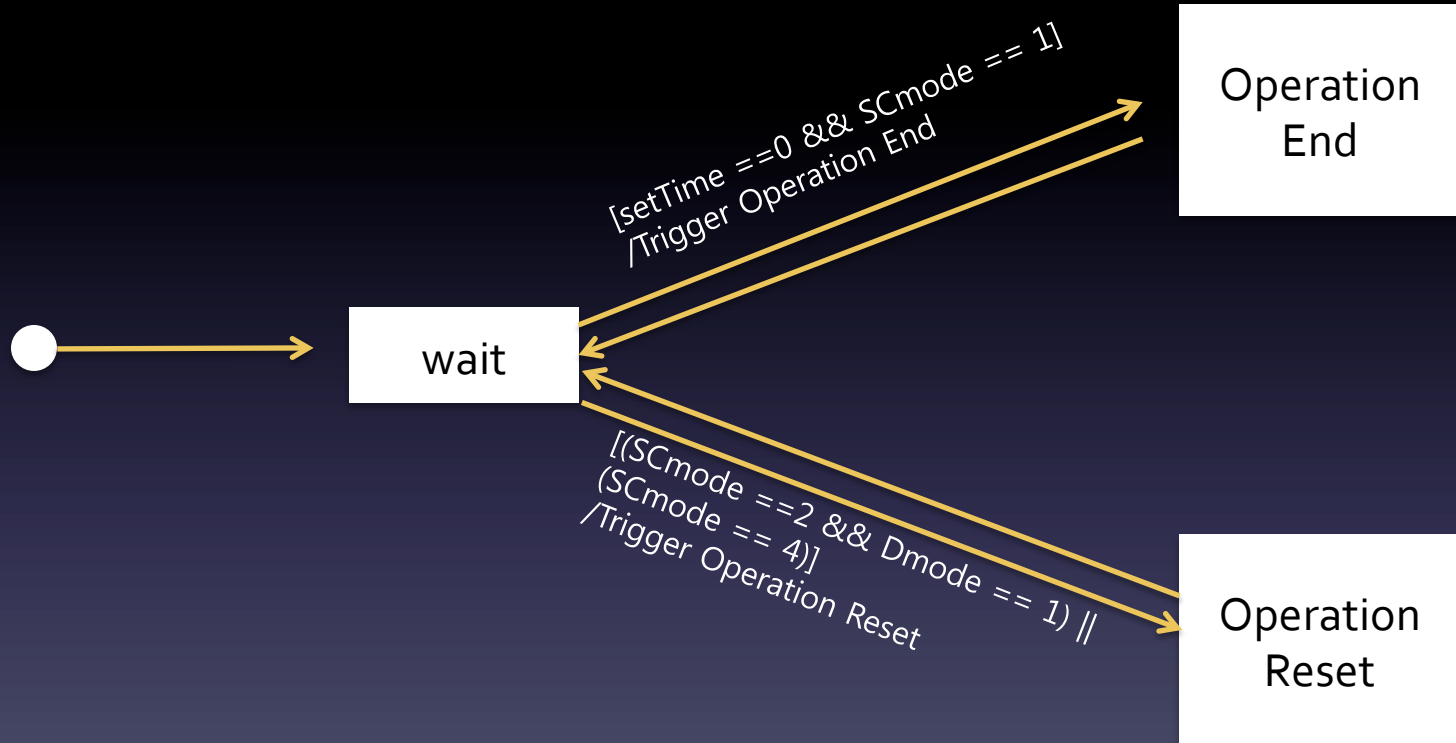
```
int beeper_command = 0;

if(SCmode == 2 && *beepTick <30)
    beeper_command = Turn_On_Beeper();

return beeper_command;
```

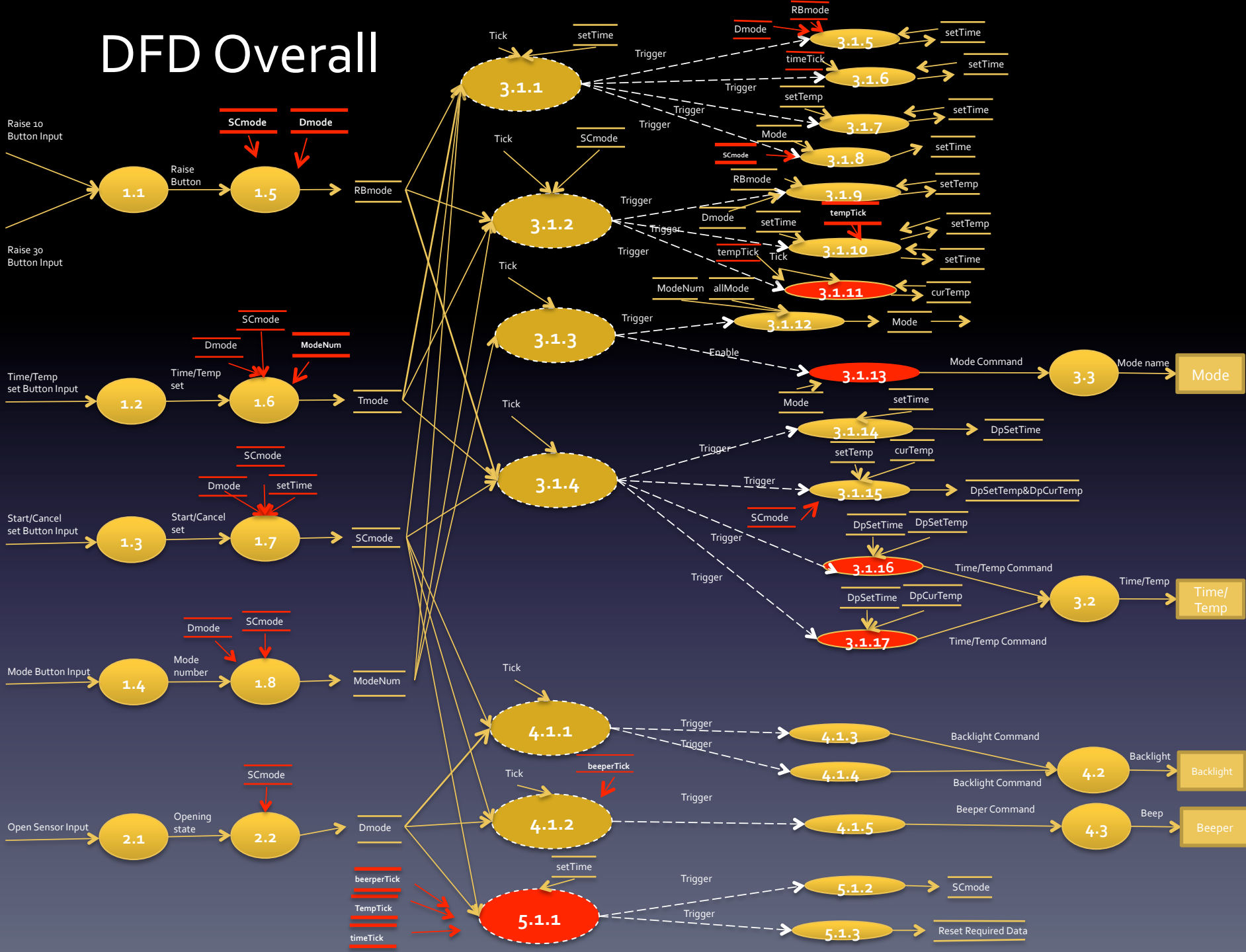


# State Transition Diagram for Operation Controller(5.1.1)

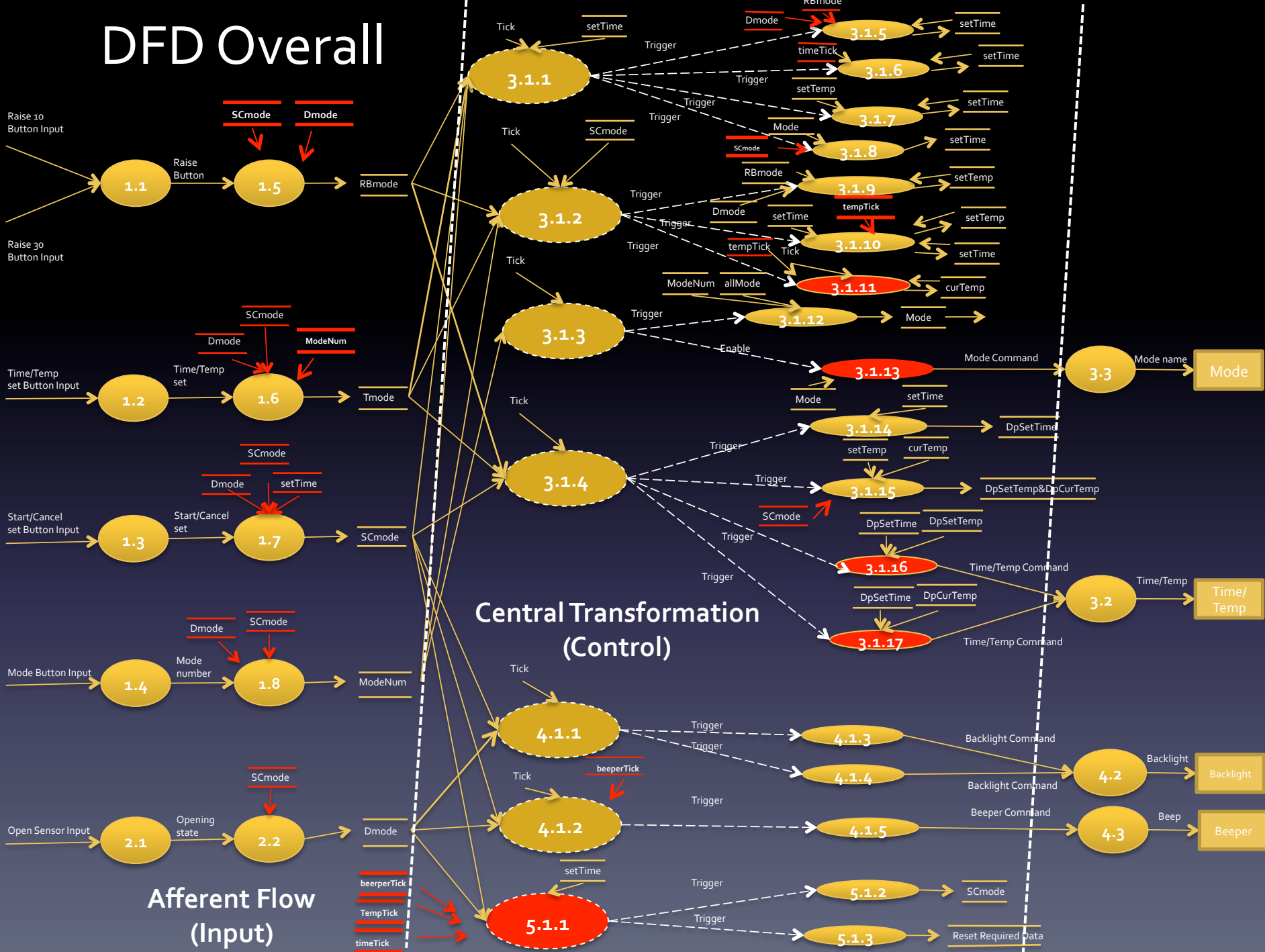


```
if(state->SCmode ==1 && time_temp->setTime == 0)
    Operation_End(state, time_temp);
else if((state->SCmode == 2 && state->Dmode == 1) || (state->SCmode == 4))
    Operation_Reset(state, time_temp, tempTick, beepTick, timeTick );
```

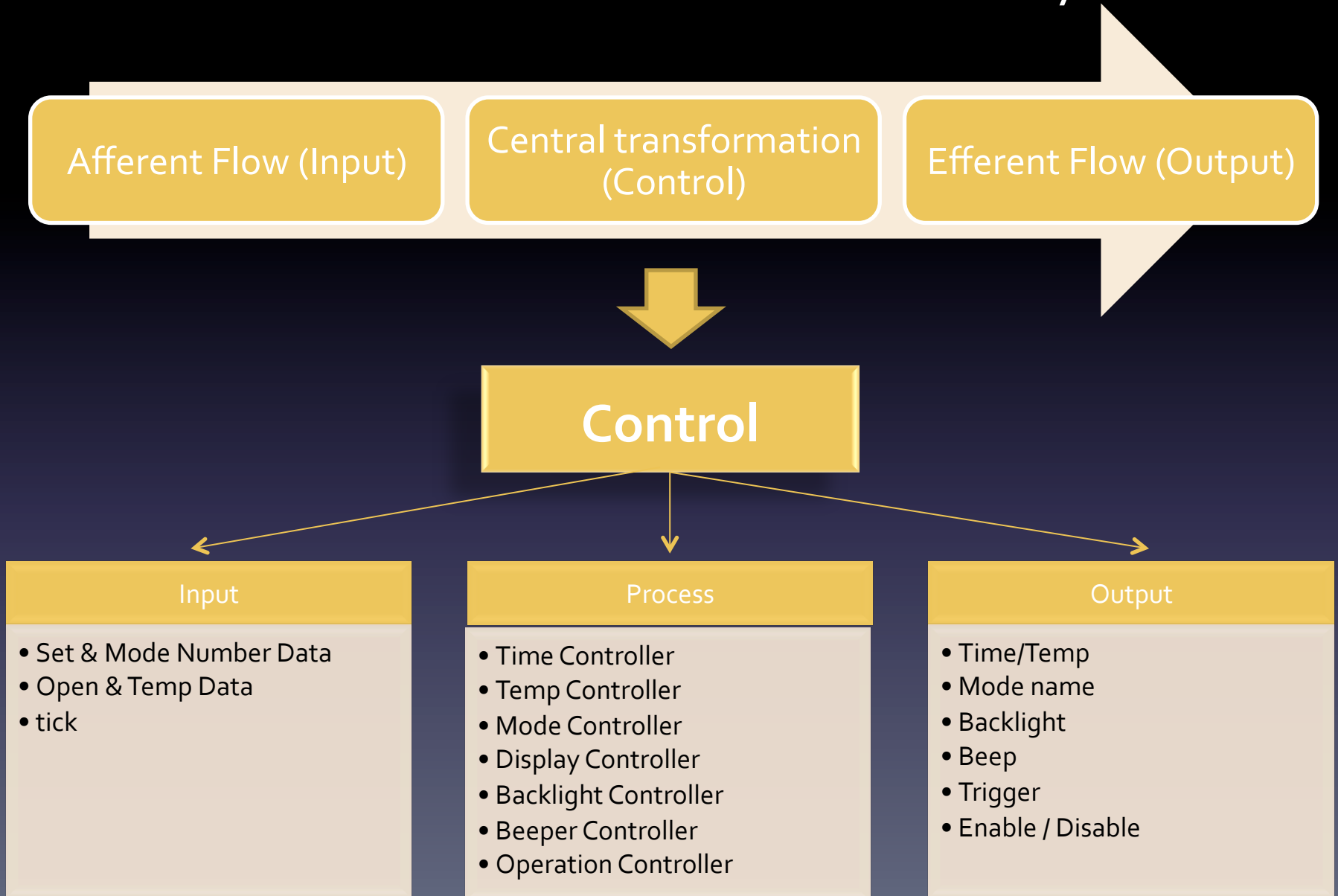
# DFD Overall



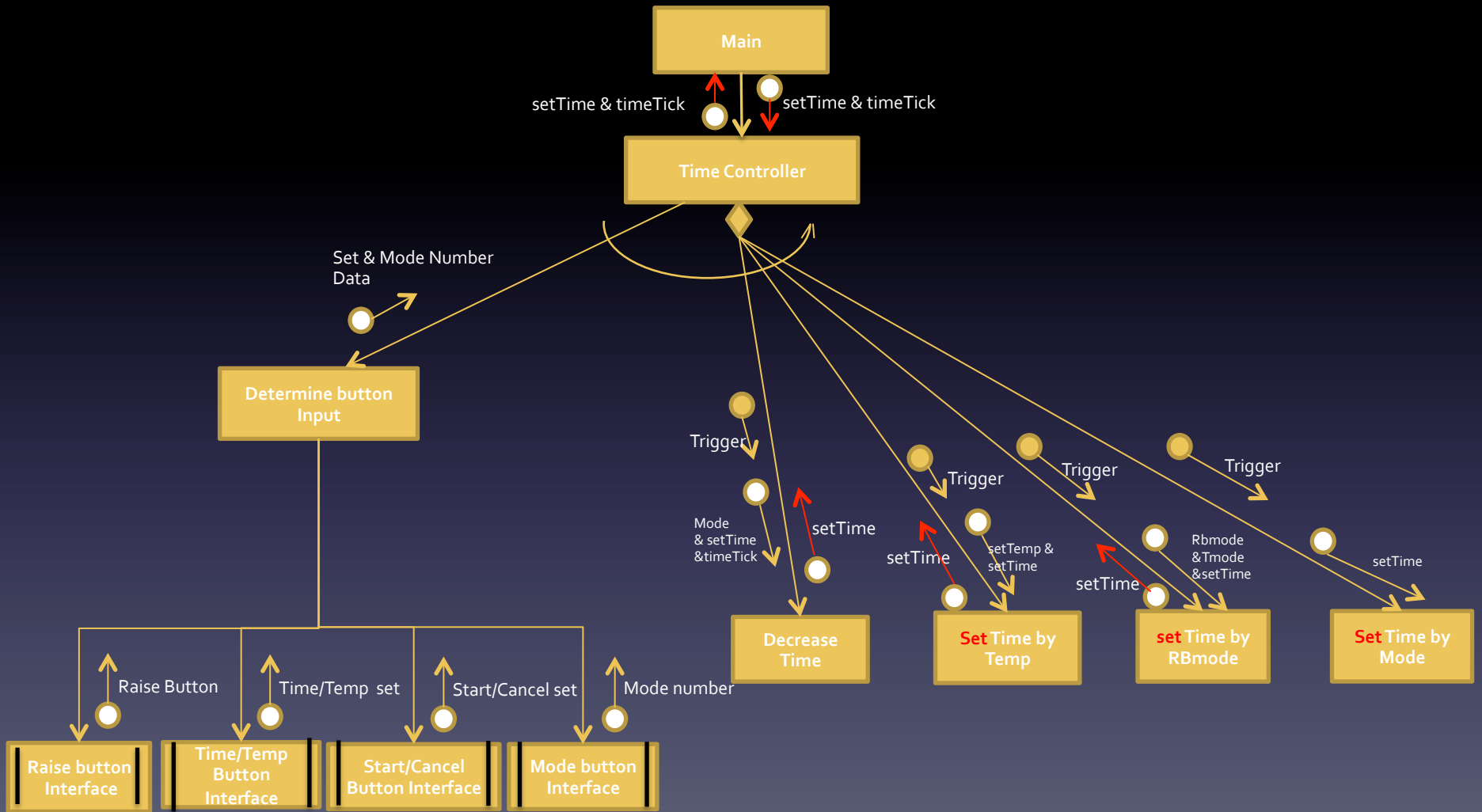
# DFD Overall



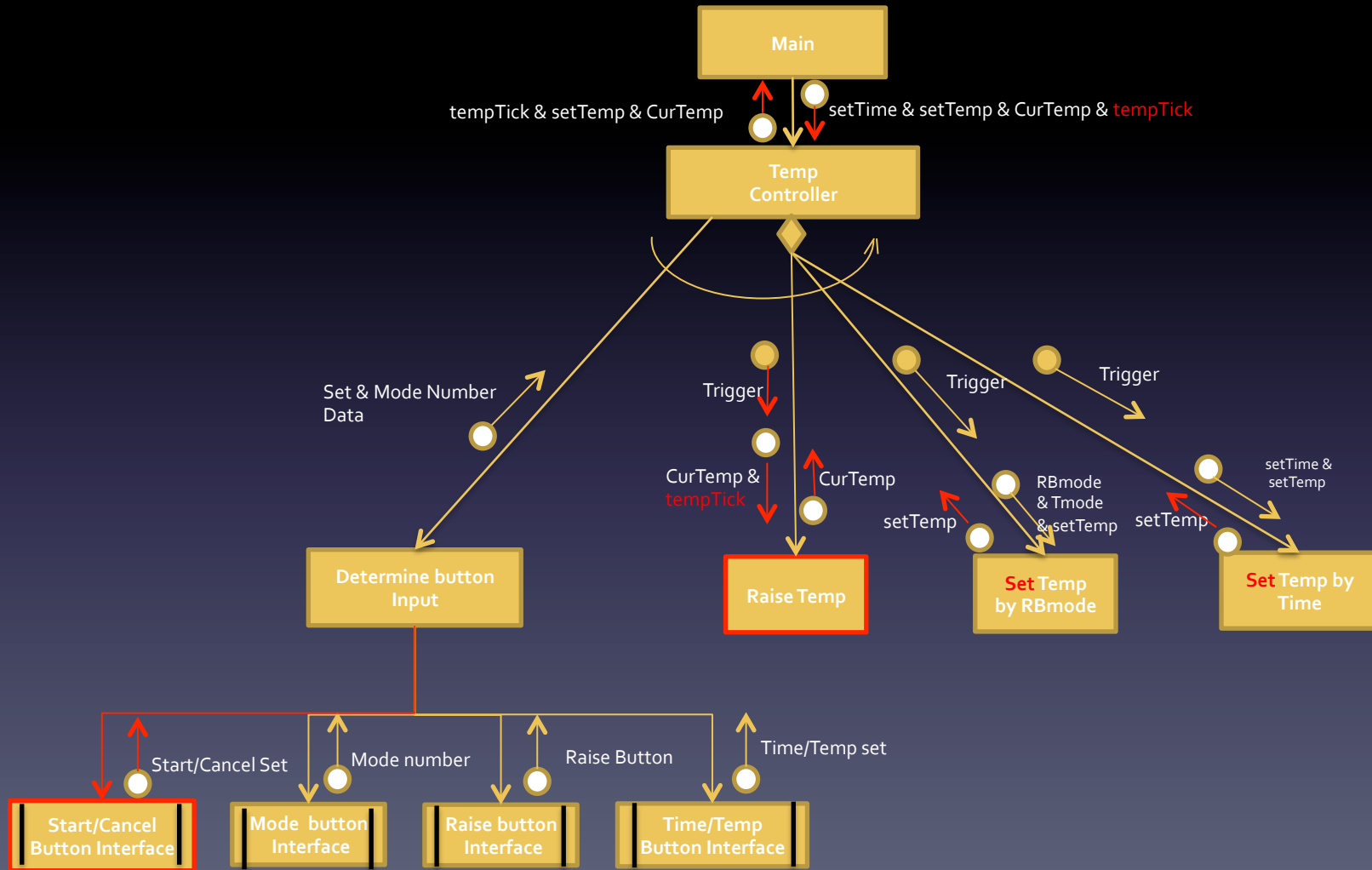
# Structured Chart – Transfrom Analysis



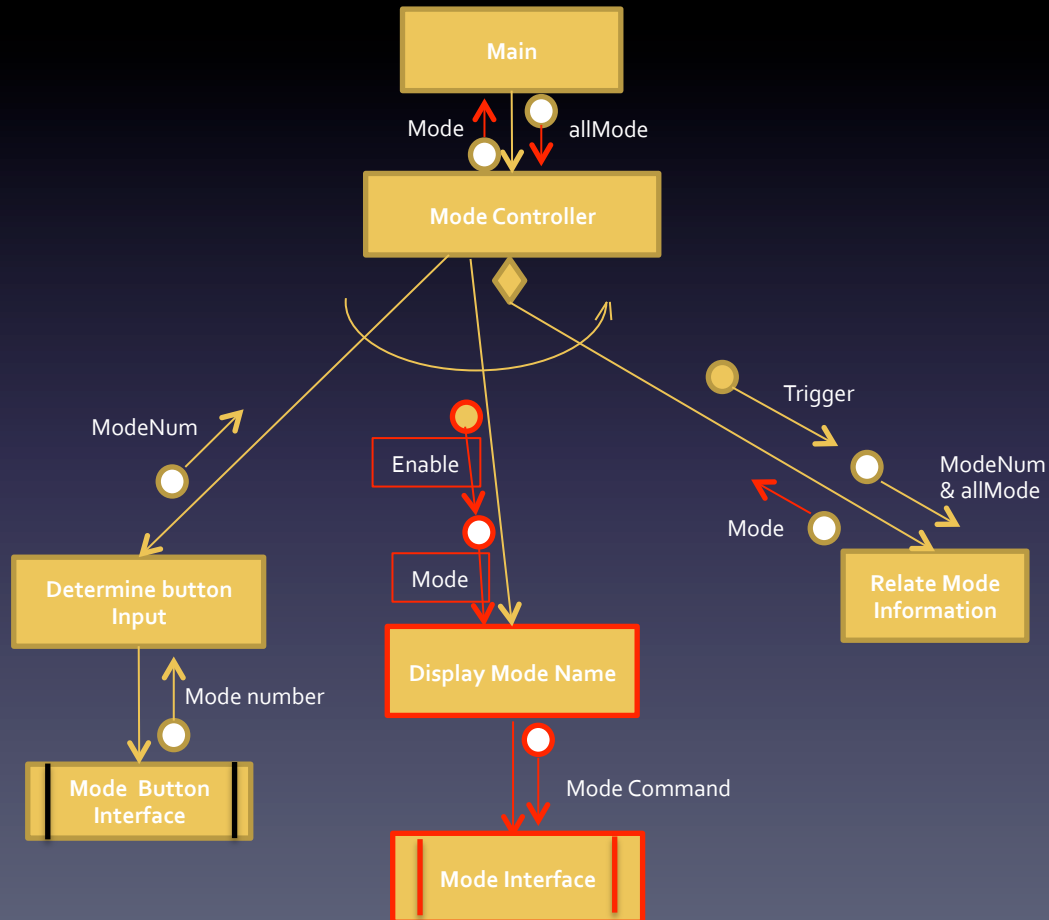
# Structured Charts - Time Controller



# Structured Charts - Temp Controller

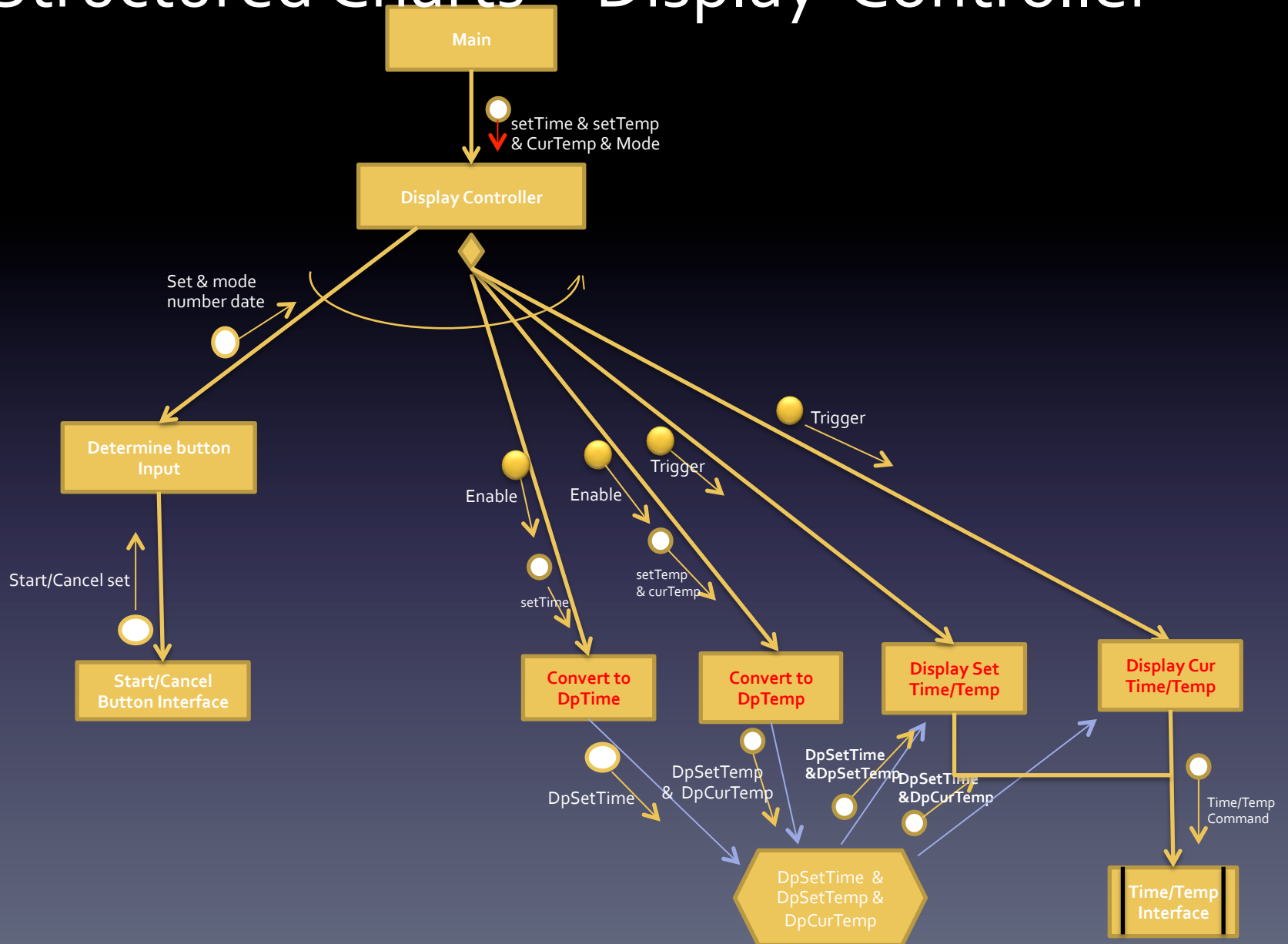


# Structured Charts - Mode Controller

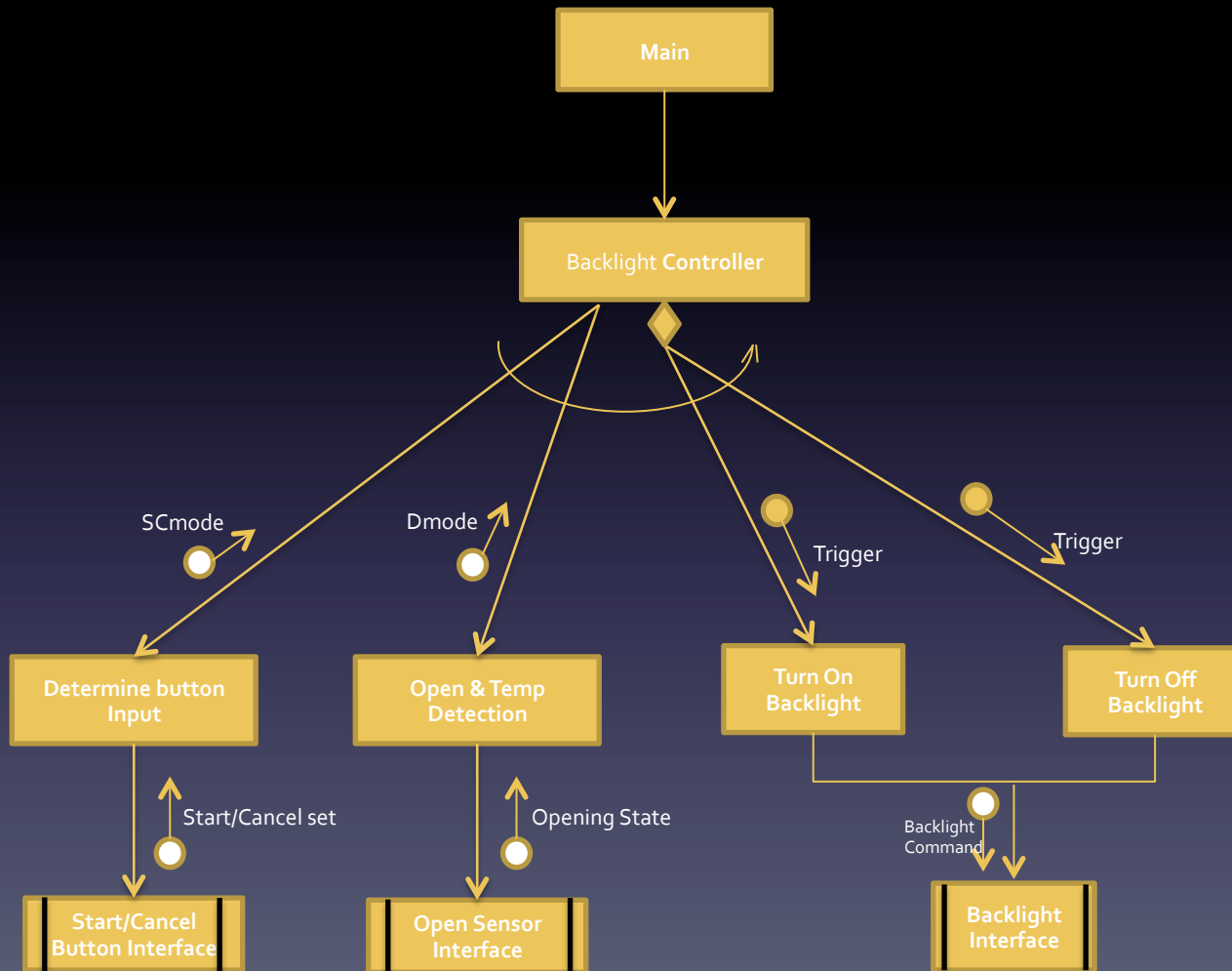




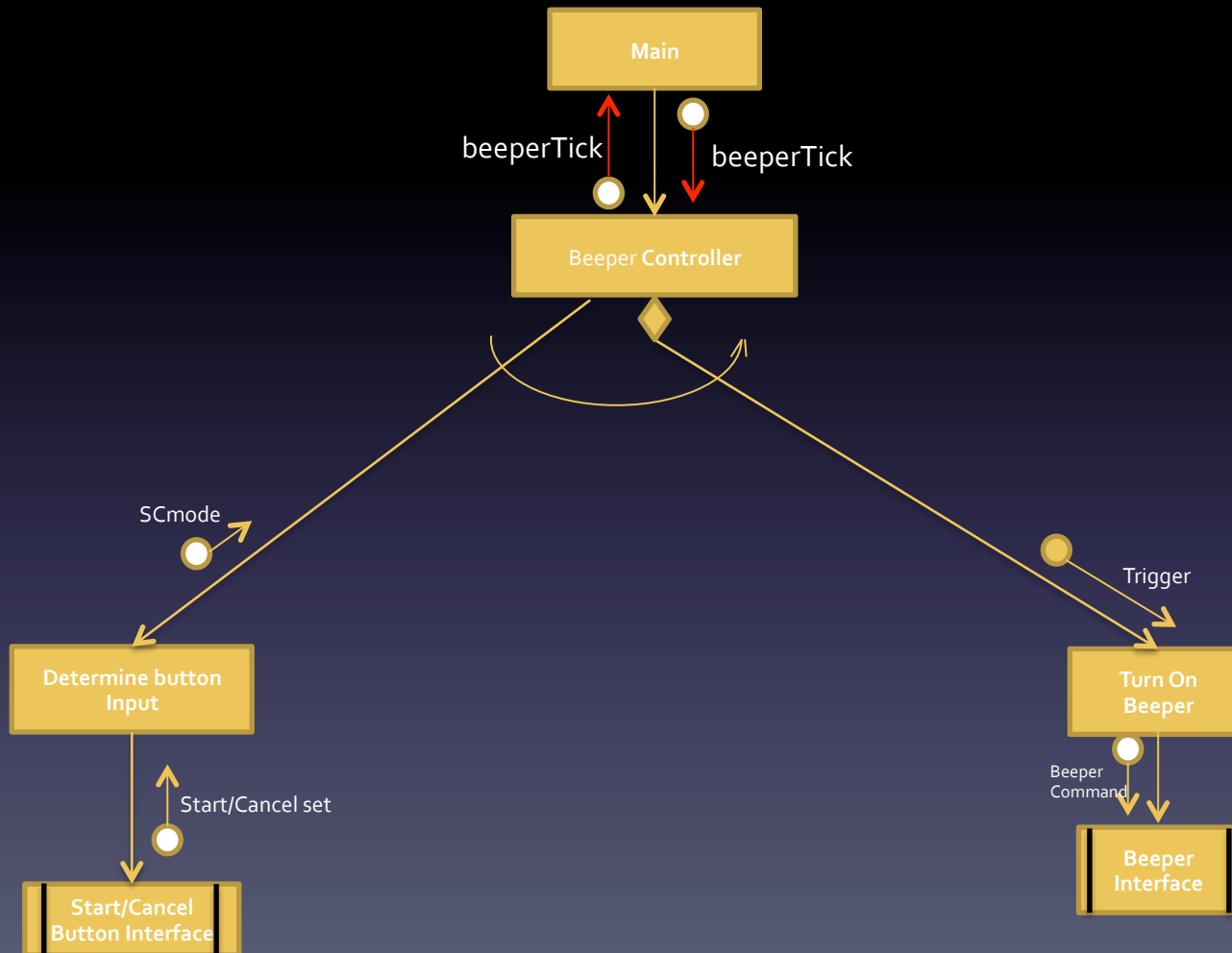
# Structured Charts - Display Controller



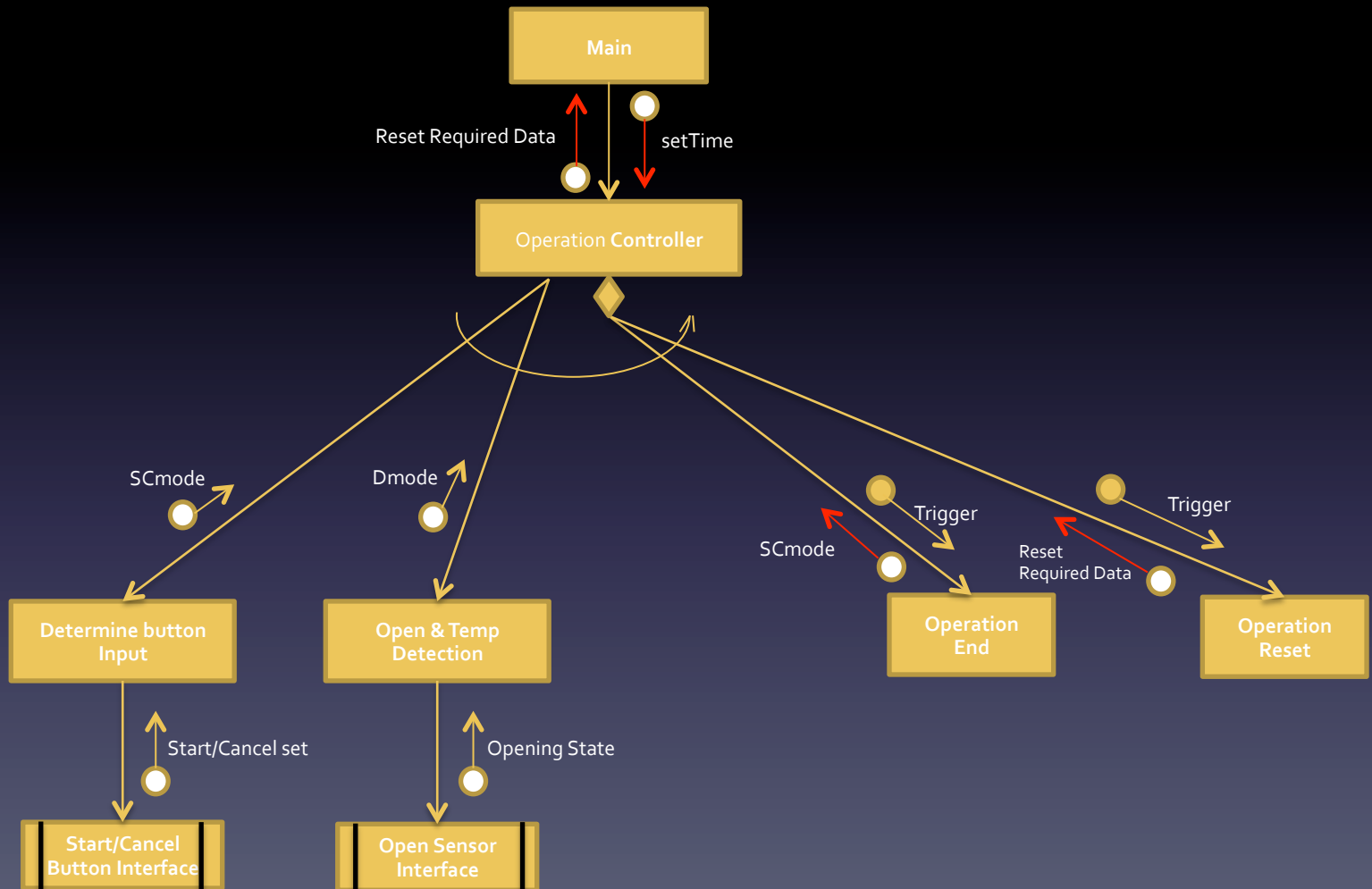
# Structured Charts - Backlight Controller



# Structured Charts - Beeper Controller



# Structured Charts - Operation Controller



# Process Specification - MOS

Reference No.	1.1
Name	<b>Raise Button Interface</b>
Input	Raise 10 Button Input(char), Raise 30 Button Input(char)
Output	Raise Button(char)
Process Description	키보드를 통해 받은 전기적 입력을 char로 변환 후 Determine Raise Button 프로세스로 전달한다.

Reference No.	1.2
Name	<b>Time/Temp Button Interface</b>
Input	Time/Temp Button Input(char)
Output	Time/Temp set(char)
Process Description	키보드를 통해 받은 전기적 입력을 char로 변환 후 Determine Time/Temp setting 프로세스로 전달한다.

# Process Specification - MOS

Reference No.	1.3
Name	<b>Start/Cancel Button Interface</b>
Input	Start/Cancel Button Input(char)
Output	Start/Cancel set(char)
Process Description	키보드를 통해 받은 전기적 입력을 char로 변환 후 Determine Start/Cancel Setting프로세스로 전달한다.

Reference No.	1.4
Name	<b>Mode Button Interface</b>
Input	Mode Button Input(char)
Output	Mode number(char)
Process Description	키보드를 통해 받은 Mode Button Input(char)의 전기적 입력을 char로 변환 후 Determine Mode Setting으로 전달한다.

# Process Specification - MOS

Reference No.	1.5
Name	<b>Determine Raise Button</b>
Input	Raise Button (char), SCmode(int),Dmode(int),RBmode(int)
Output	<b>RBmode(integer)</b>
Process Description	Scmode와 Dmode를 판단해 Scmode가 0이고 Dmode가 0이면(문이 닫히고 조리 전 이면) Raise Button을 확인후 a버튼이 눌렀다면 Rbmode에 0을, b버튼이 눌렀다면 Rbmode에 1을 저장해준다.

Reference No.	1.6
Name	<b>Determine Time/Temp setting</b>
Input	Time/Tempset(char),Tmode(int),Scmode(int),Dmode(int),ModeNum(int)
Output	Tmode(integer)
Process Description	Input의 값이 짝수이면 Time으로, 홀수이면 Temp로 Tmode에 적절한 값을 저장해준다.

# Process Specification - MOS

Reference No.	1.7
Name	<b>Determine Start/Cancel setting</b>
Input	SCmode(integer), Start/Cancel set(integer), Dmode(integer), setTime(integer)
Output	SCmode(integer)
Process Description	setTime이 0이 아니고 Dmode가 0(문닫힘), SCmode가 0(cancel)이라면 SCmode를 1(start)로 바꾸고 setTime이 0이 아니고 SCmode가 1(start)이면 SCmode를 0(cancel)로 바꾼다.
Reference No.	1.8
Name	<b>Determine Mode Number</b>
Input	Mode number(integer), SCmode(integer),Dmode(integer)
Output	ModeNum(integer)
Process Description	SCmode가 Mode number input을 받아 총 놀림 횟수를 % 연산 후 ModeNum으로 내보낸다.



# Process Specification - MOS

Reference No.	2.1
Name	<b>Open Sensor Interface</b>
Input	Open Sensor Input(char)
Output	Opening state(integer)
Process Description	Open Sensor Input이 감지되었을 때 Opening state를 내보낸다

Reference No.	2.2
Name	<b>Determine Opening state</b>
Input	Opening state(integer), <b>SCmode(integer)</b>
Output	Dmode(integer)
Process Description	Opening state(char)를 받아서 Dmode를 통하여 Close(0)상태라면 Open(1)을 반환하고 Open(1)상태라면 Close(0)를 반환한다.

# Process Specification - MOS

Reference No.	3.1.1
Name	<b>Time Controller</b>
Input	RBmode( <b>integer</b> ), Tmode(integer), ModeNum(Integer), SCmode( <b>integer</b> ), setTime(integer)
Output	Trigger,
Process Description	<p>시간에 관련된 input(RBmode, Tmode, ModeNum, SCmode)를 받아 상태를 판단하여 상응하는 프로세스들에게 output을 넘겨주는 Controller</p> <p>(RBmode !=3 &amp;&amp; Tmode == 0 &amp;&amp; SCmode == 0 &amp;&amp; ModeNum == 0 → Trigger Set Time by RBmode)</p> <p>(RBmode ==3 &amp;&amp; Tmode == 1 &amp;&amp; SCmode == 0 → Trigger Set Time by Temp)</p> <p>(RBmode ==3 &amp;&amp; ModeNum != 0 → Trigger Set Time by Mode)</p> <p>(setTime !=0 &amp;&amp; SCmode == 1 → Trigger Decrease Time)</p>

Reference No.	3.1.2
Name	<b>Temp Controller</b>
Input	RBmode( <b>integer</b> ), Tmode(integer), ModeNum(Integer), Scmode(integer), tick
Output	Trigger
Process Description	<p>온도에 관련된 input(RBmode, Tmode, ModeNum)를 받아 상태를 판단하여 상응하는 프로세스들에게 output을 넘겨주는 Controller</p> <p>(SCmode == 1 → Trigger Raise Temp )</p> <p>(RBmode != 3 &amp;&amp; Tmode == 1 &amp;&amp; SCmode == 0 ModeNum == 0 → Trigger Set Temp by RBmode)</p> <p>(RBmode == 3 &amp;&amp; Tmode == 0 &amp;&amp; SCmode == 0 → Trigger Set Temp by Time)</p>

# Process Specification - MOS

Reference No.	3.1.3
Name	<b>Mode Controller</b>
Input	ModeNum(Integer),Scmode(integer),Dmode(integer),tick
Output	Trigger, Enable
Process Description	Mode에 관련된 input(ModeNum)를 받아 상태를 판단하여 상응하는 프로세스에게 output을 넘겨주고 Display Mode Name를겨주는 Controller (ModeNum >=0 && SCmode == 0 && Dmode == 0 → Trigger Relate Mode Information)

Reference No.	3.1.4
Name	<b>Display Controller</b>
Input	tick,Scmode(integer),TIME_TEMP(struct)
Output	Trigger, tick
Process Description	출력을 위한 input(setTemp, curTemp, setTime)을 받아와 출력 command를 내리기까지 상응하는 프로세스들에게 output을 넘겨주는 Controller

# Process Specification - MOS

Reference No.	3.1.5
Name	<b>Set Time by RBmode</b>
Input	RBmode(integer), setTime(integer), Dmode(integer), trigger
Output	setTime(integer)
Process Description	Tmode(시간/온도 모드)가 0(시간)인 경우 RBmode의 값에 해당하는 시간만큼 setTime을 증가시킨 후 RBmode에 3을 대입한다

# Process Specification - MOS

Reference No.	3.1.6
Name	<b>Decrease Time</b>
Input	setTime(Integer), tick,timeTick(integer),trigger
Output	setTime(Integer)
Process Description	조리가 시작되면 Trigger되어 1초에 1씩 setTime을 감소시킨다.

Reference No.	3.1.7
Name	<b>Set Time by Temp</b>
Input	setTime(Integer), setTemp(Integer),trigger
Output	setTime(Integer)
Process Description	setTemp에 설정된 온도에 상응하는 시간만큼 setTime을 올려준다.

# Process Specification - MOS

Reference No.	3.1.8
Name	<b>Set Time by Mode</b>
Input	setTime(Integer), Mode(Struct),trigger,Scmode(integer),
Output	setTime(Integer)
Process Description	Mode를 통해 관련된 Struct 정보에서 해당하는 시간만큼 setTime을 올려준다.

Reference No.	3.1.9
Name	<b>Set Temp by RBmode</b>
Input	setTemp(Integer), RBmode(integer),Dmode(integer),trigger
Output	setTemp(Integer)
Process Description	RBmode가 눌렀을 때(!= 3일 때) trigger되어 Tmode(시간/온도 모드)가 1(온도)인 경우 RBmode의 값에 해당하는 온도만큼 setTemp를 증가시킨 후 RBmode에 3을 대입한다

# Process Specification - MOS

Reference No.	3.1.10
Name	<b>Set Temp by Time</b>
Input	setTime(Integer), setTemp(Integer),trigger
Output	setTemp(Integer)
Process Description	setTime에서 설정된 시간에 상응하는 온도만큼 setTemp를 올려준다.

Reference No.	<b>3.1.11</b>
Name	<b>Raise Temp</b>
Input	Trigger, Tick, CurTemp(integer),timeTick(integer)
Output	CurTemp(integer)
Process Description	30 tick(3초) 마다 CurTemp를 10°C 씩 올려준다.

# Process Specification - MOS

Reference No.	3.1.12
Name	<b>Relate Mode Information</b>
Input	ModeNum(integer, allMode(Struct[]))
Output	Mode(Struct)
Process Description	allMode배열 중에 ModeNum에 해당하는 정보를 Mode에저장한다.

Reference No.	3.1.13
Name	<b>Display Mode name</b>
Input	Mode(Struct),Enable,
Output	Mode command(int[] /char[])
Process Description	Mode를 받아서 그에 해당하는 정보들을 출력할 수 있도록 Mode Interface에 명령을 전달한다.



# Process Specification - MOS

Reference No.	3.1.14
Name	<b>Convert to DpTime</b>
Input	Trigger, setTime(Integer)
Output	DpSetTime(Integer[]/char[])
Process Description	Integer형 setTime을 화면에 나타내기 위하여 Integer 배열 형태인 DpSetTime으로 변환하여 내보낸다.

Reference No.	3.1.15
Name	<b>Convert to DpTemp</b>
Input	Trigger, setTemp(Integer), curTemp(Integer) ,Scmode(integer)
Output	DpSetTemp&DpCurTemp(int[]/char[])
Process Description	setTemp, curTemp를 받아 Display하기 위해 integer 배열 형으로 바꾸어 DpSetTemp & DpCurTemp에 저장 후 내보낸다.

# Process Specification - MOS

Reference No.	3.1.16
Name	<b>Display Set Time/Temp</b>
Input	Trigger, DpSetTime(char[]/int[]), DpSetTemp(char[]/int[])
Output	Time/Temp Command(int[]/char[])
Process Description	조리가 시작되기 전에 DpSetTime과 DpSetTemp를 취합한 정보를 Time/Temp Interface가 출력할 수 있도록 Time/Temp Command를 보내준다.

Reference No.	3.1.17
Name	<b>Display Cur Time/Temp</b>
Input	Trigger, DpSetTime(char[]/int[]), DpCurTemp(char[]/int[])
Output	Time/Temp Command(int[]/char[])
Process Description	조리가 시작된 후 DpSetTime과 DpCurTemp를 취합한 정보를 Time/Temp Interface가 출력할 수 있도록 Time/Temp Command를 보내준다.

# Process Specification - MOS

Reference No.	3.2
Name	<b>Time/Temp Interface</b>
Input	Time/Temp Command(integer[]/char[])
Output	Time/Temp(Integer[]/char[])
Process Description	MWO가 작동할 때 시간과 온도에 관한 정보를 출력하는 interface로 Time/Temp command를 받아 Time/Temp으로 내보낸다.

# Process Specification - MOS

Reference No.	3.3
Name	<b>Mode Interface</b>
Input	Mode Command(char[])
Output	Mode name(char[])
Process Description	MWO가 작동할 때 모드에 관한 정보를 출력하는 interface로 Mode Command를 받아 Mode name으로 내보낸다.

Reference No.	4.1.1
Name	<b>Backlight Controller</b>
Input	SCmode(integer), Dmode(integer)
Output	Trigger
Process Description	Backlight를 작동/중지 시키는데 고려해야 할 input(SCmode, Dmode)를 받아 상태를 판단하여 상응하는 프로세스에게 output을 넘겨주는 Controller ((Dmode == 0 && SCmode == 1)    (Dmode == 1 && SCmode != 1) → Turn On Backlight) ((Dmode == 0 && SCmode == 2)    (Dmode == 0 && SCmode == 0) → Turn Off Backlight)

# Process Specification - MOS

Reference No.	4.1.2
Name	<b>Beeper Controller</b>
Input	SCmode(integer), tick, <b>beeperTick(integer)</b>
Output	Trigger
Process Description	Beeper를 작동시키는데 고려해야 할 input(SCmode, Dmode <b>beeperTick</b> )를 받아 상태를 판단하여 상응하는 프로세스에게 output을 넘겨주는 Controller (SCmode ==2 && beeperTick <30 → Trigger Turn on Beeper)

Reference No.	<b>4.1.3</b>
Name	<b>Turn On Backlight</b>
Input	Trigger
Output	Backlight Command(integer)
Process Description	Backlight Controller로 부터 trigger되어 Backlight command를 Backlight Interface로 보내 Backlight를 키는 기능을 한다.

# Process Specification - MOS

Reference No.	<b>4.1.4</b>
Name	<b>Turn Off Backlight</b>
Input	Trigger
Output	Backlight Command(integer)
Process Description	Backlight Controller로 부터 trigger되어 Backlight command를 Backlight Interface로 보내 Backlight를 끄는 기능을 한다.

Reference No.	<b>4.1.5</b>
Name	<b>Turn On Beeper</b>
Input	Trigger
Output	Beeper Command(integer)
Process Description	Beeper Controller로 부터 trigger되어 Beeper command를 Beeper Interface로 보내 Beep을 3초간 키는 기능을 한다.

# Process Specification - MOS

Reference No.	4.2
Name	<b>Backlight Interface</b>
Input	Backlight Command(integer)
Output	Backlight
Process Description	MWO가 작동할 때 Backlight를 켜고 끄기 위해 Backlight Command를 받아 Backlight를 작동/중지시킨다.

Reference No.	4.3
Name	<b>Beeper Interface</b>
Input	Beeper Command(integer)
Output	Beep
Process Description	MWO가 작동할 때 Backlight를 켜고 끄기 위해 Beeper Command를 받아 Beep을 작동/중지시킨다.

# Process Specification - MOS

Reference No.	<b>5.1.1</b>
Name	<b>Operation Controller</b>
Input	SetTime(integer), SCmode(integer), tick,Dmode(integer),tempTick(integer),beepTick(integer),timeTick(integer)
Output	Trigger
Process Description	SetTime, SCmode, Dmode를 받아 상태를 판단하여 상응하는 프로세스들에게 Trigger를 보내는 controller. (setTime ==0 && SCmode == 1 → Trigger Operation End) (SCmode ==2 && Dmode == 1 → Trigger Operation Reset)
Reference No.	<b>5.1.2</b>
Name	<b>Operation End</b>
Input	Trigger
Output	SCmode(integer)
Process Description	SCmode의 값을 1(start)에서 2(finish)로 바꿔준다.



# Process Specification - MOS

Reference No.	<b>5.1.3</b>
Name	<b>Operation Reset</b>
Input	Trigger
Output	STATE(struct),TIME_TEMP(struct),tempTick(integer),timeTick(integer),beeperTick(integer)
Process Description	Reset Required Data를 모두 초기 상태로 바꾸어 준다.