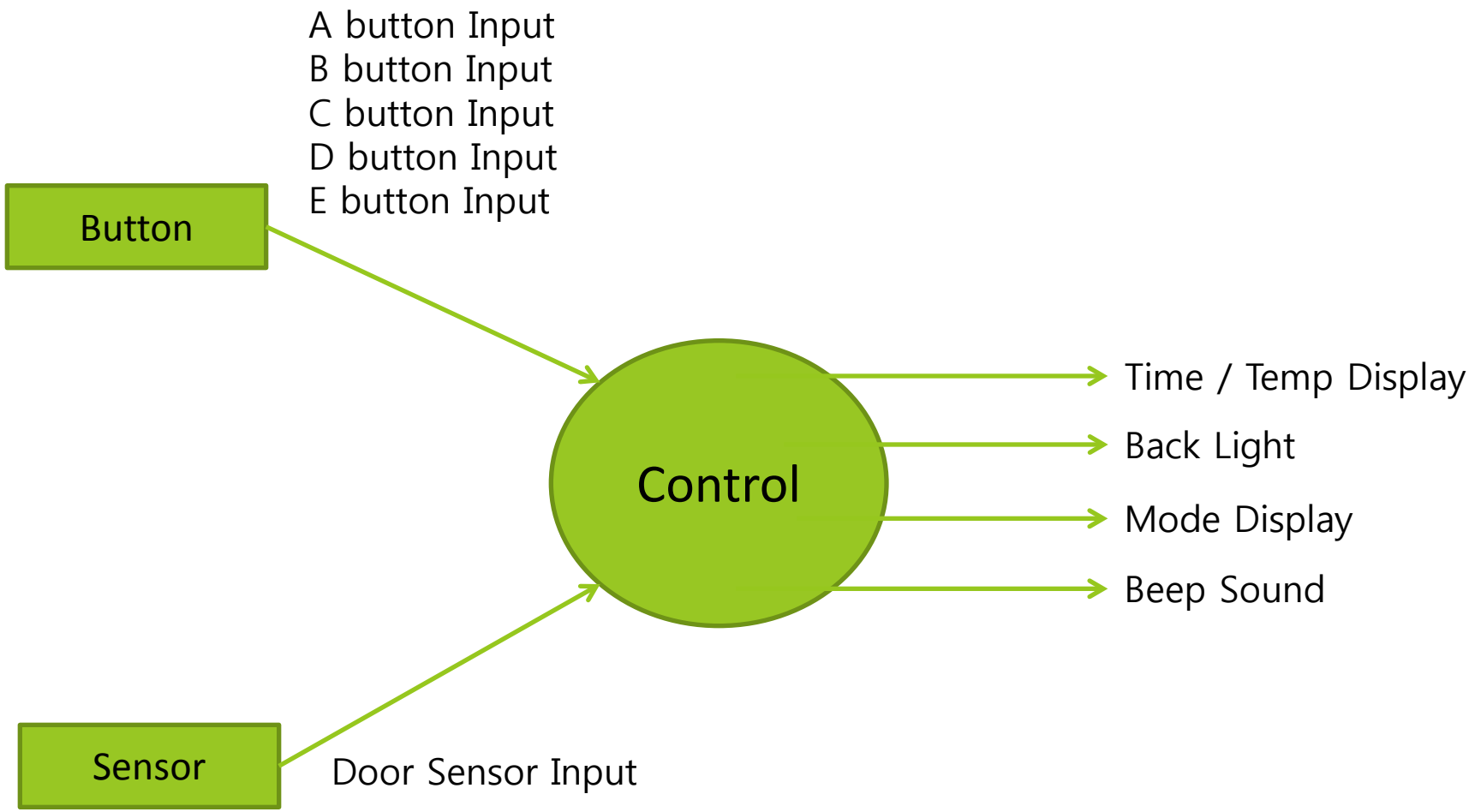
Microwave Oven System

[SE_T2]
200911416 이현호
200911389 박성희
200911375 김선우

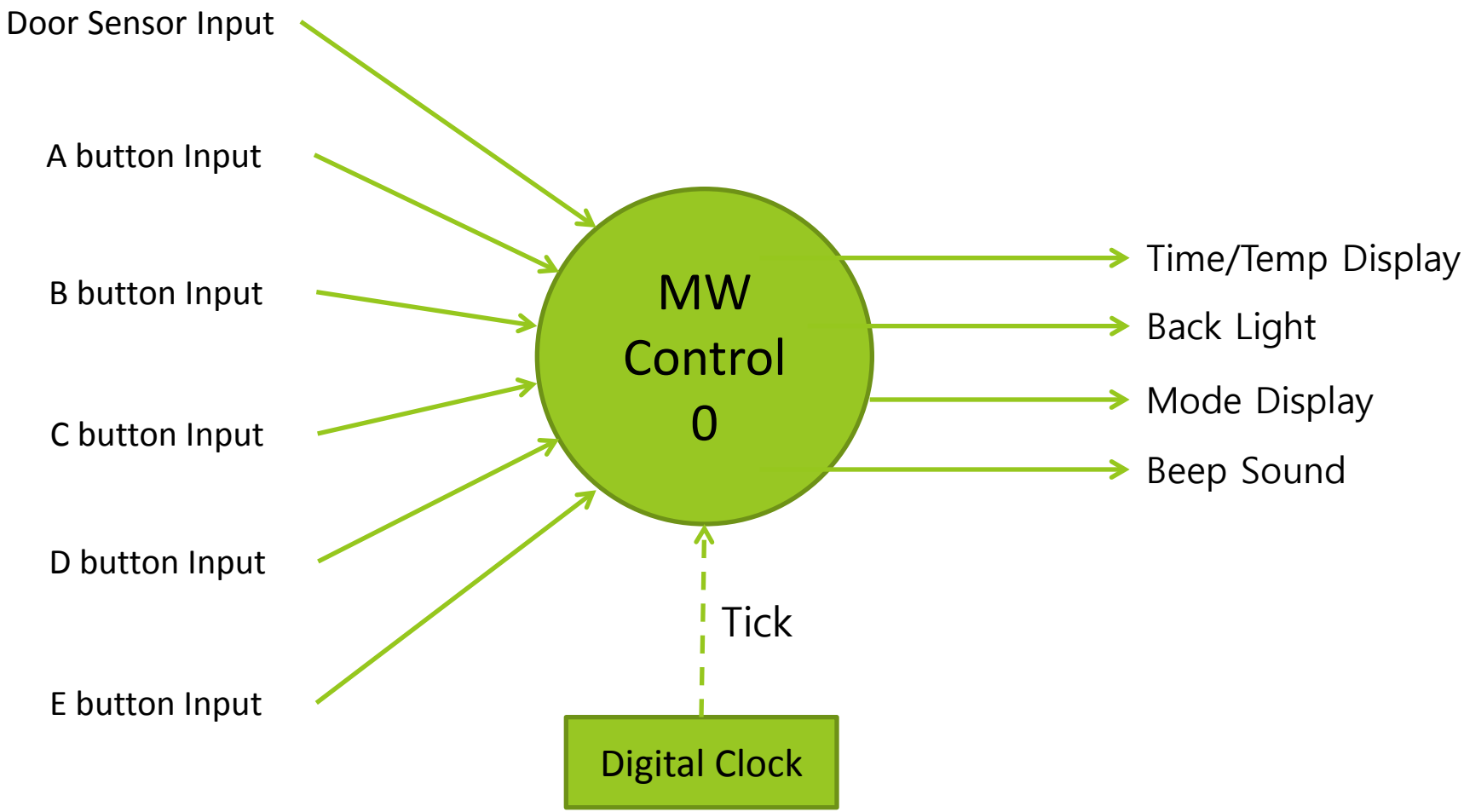
System Context Diagram



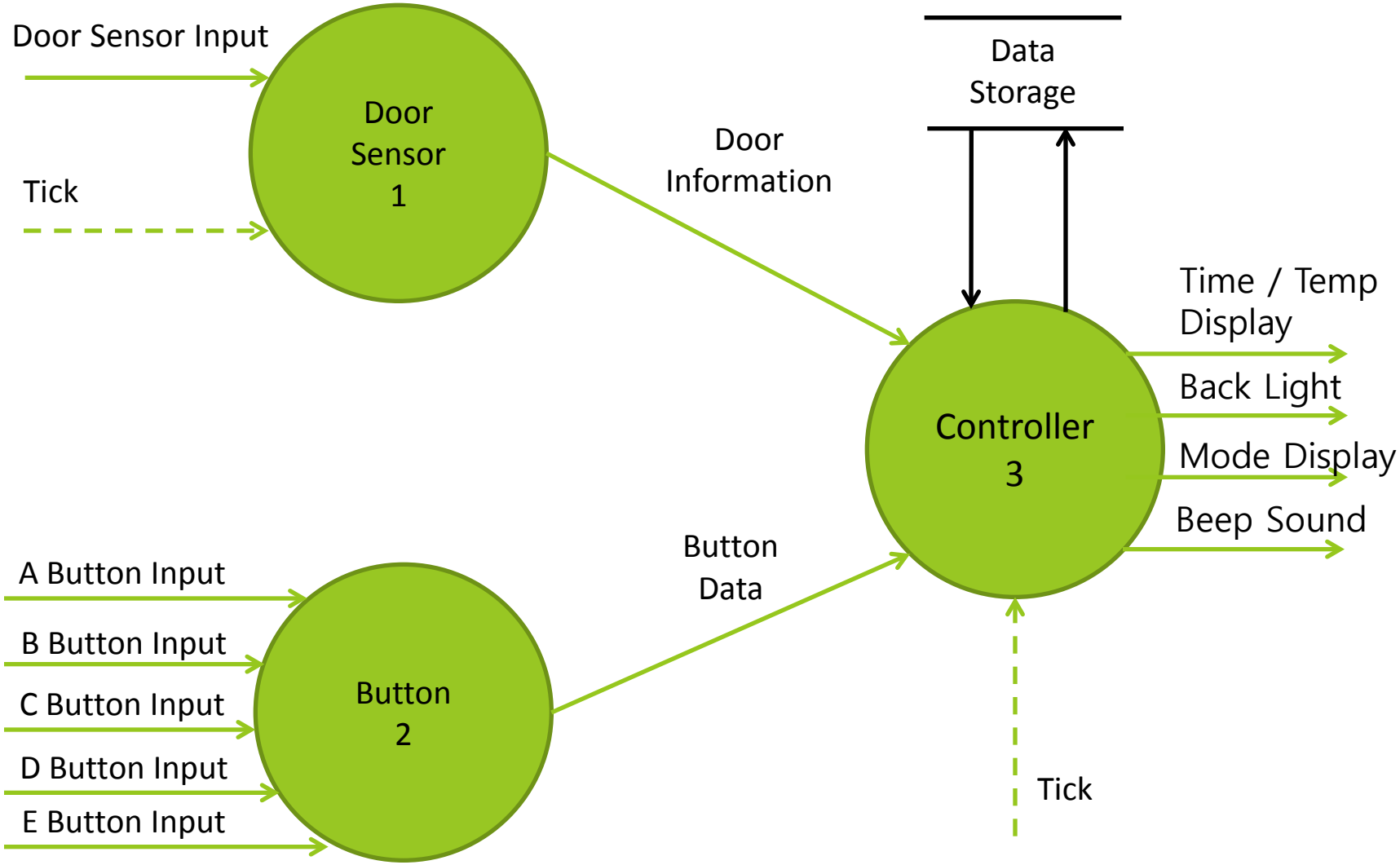
Event List

Input/Output Event	Description	Format / Type
Door Sensor Input	문 열림 / 닫힘 상태	True / False, Periodic
A Button Input	10sec/10℃ 증가	char, Interrupt
B Button Input	30sec/20℃ 증가	char, Interrupt
C Button Input	시간 / 온도 (Default : 시간)	char, Interrupt
D Button Input	조리 모드 선택	char, Interrupt
E Button Input	시작 / 정지	char, Interrupt
Temp/Time Display	현재 온도 / 설정 온도, 남은 시간 / 설정 시간 표시	interrupt
Mode Display	현재 모드 (00 : 모드 사용 안함) 표시	Interrupt
Beep Speaker	Beep음 출력하는 스피커	Interrupt
Back Light	문이 열려있을 시 / 조리 중 Back Light On	On/Off
Tick	Digital Clock	1tick = 0.1seconds

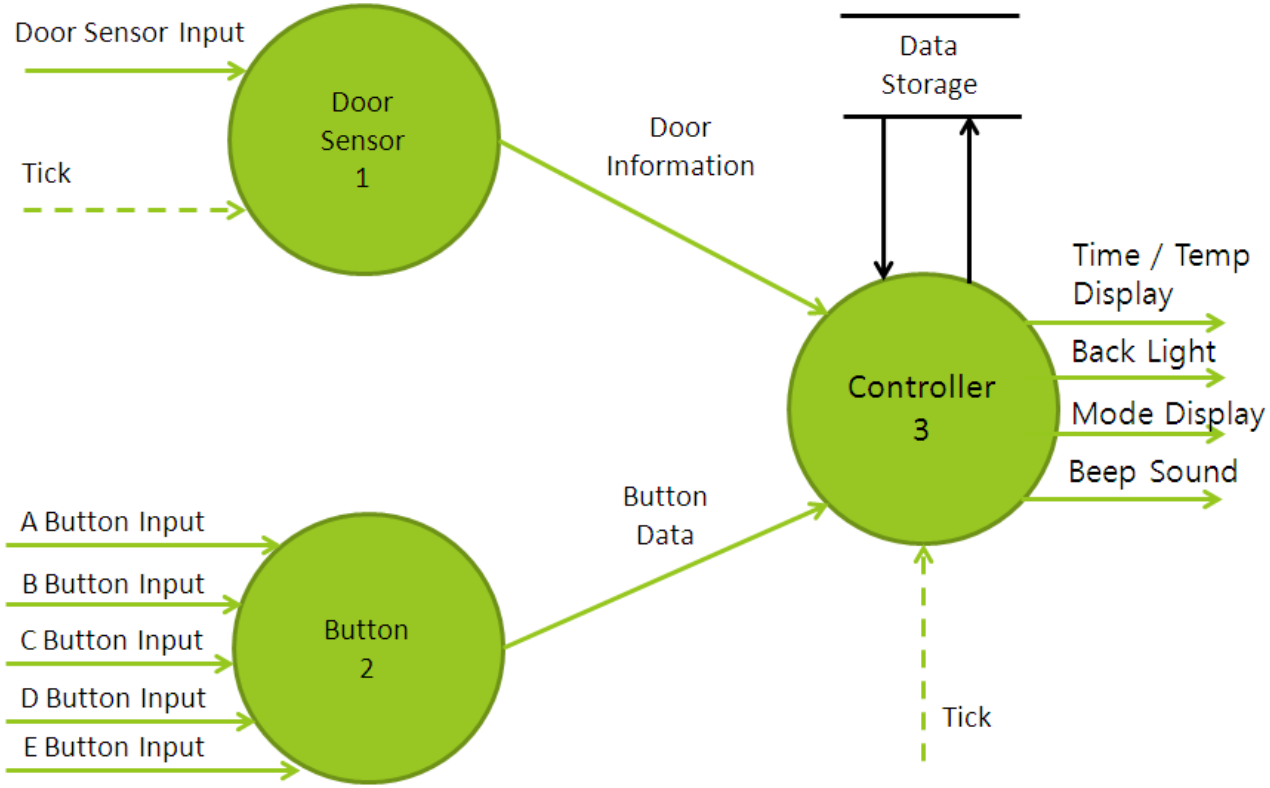
DFD Level 0 – Microwave



DFD Level 1 – Microwave



DFD Level 1 – Microwave




```c
char Button( ) // 1
{
    char button_input;
    return button_input = kgetch( );
}
```

```c
bool Door_Sensor(char button, bool door_info) // 2
{
    if( (button == 'f') && (Data_Storage.s == false) )
        return !door_info;
}
```

Data Dictionary – Level 1

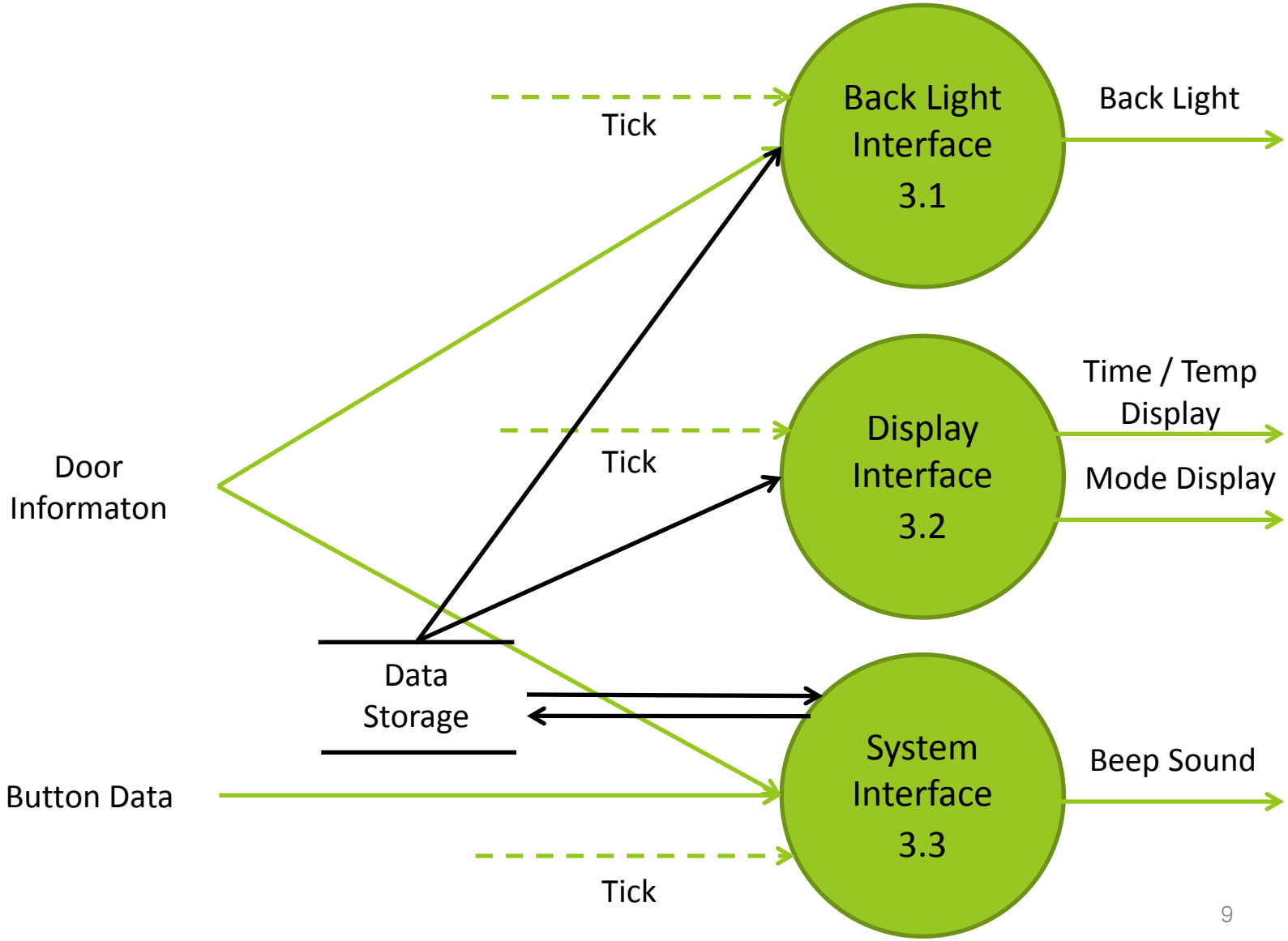
Input/Output Event	Description	Variable ID	Format / Type
Door Information	Door sensor를 통해 받은 문 열림/닫힘 정보 (true : 문 열림 / false : 문 닫힘)	door_info	boolean
Button Data	사용자가 입력한 버튼의 정보	button	char

Process Specification

Reference No.	**1**
Name	Door Sensor
Input	Door Sensor Input
Output	door_info(boolean)
Process Description	Door Sensor를 통해 문 열림 상태를 감지하여 boolean 변수 Door Information에 저장 (true : 문 열림 / false : 문 닫힘)

Reference No.	**2**
Name	Button
Input	A Button Input, B Button Input, C Button Input, D Button Input, E Button Input
Output	button(char)
Process Description	다섯 가지 버튼의 입력을 감지하여 char형 변수 button에 저장

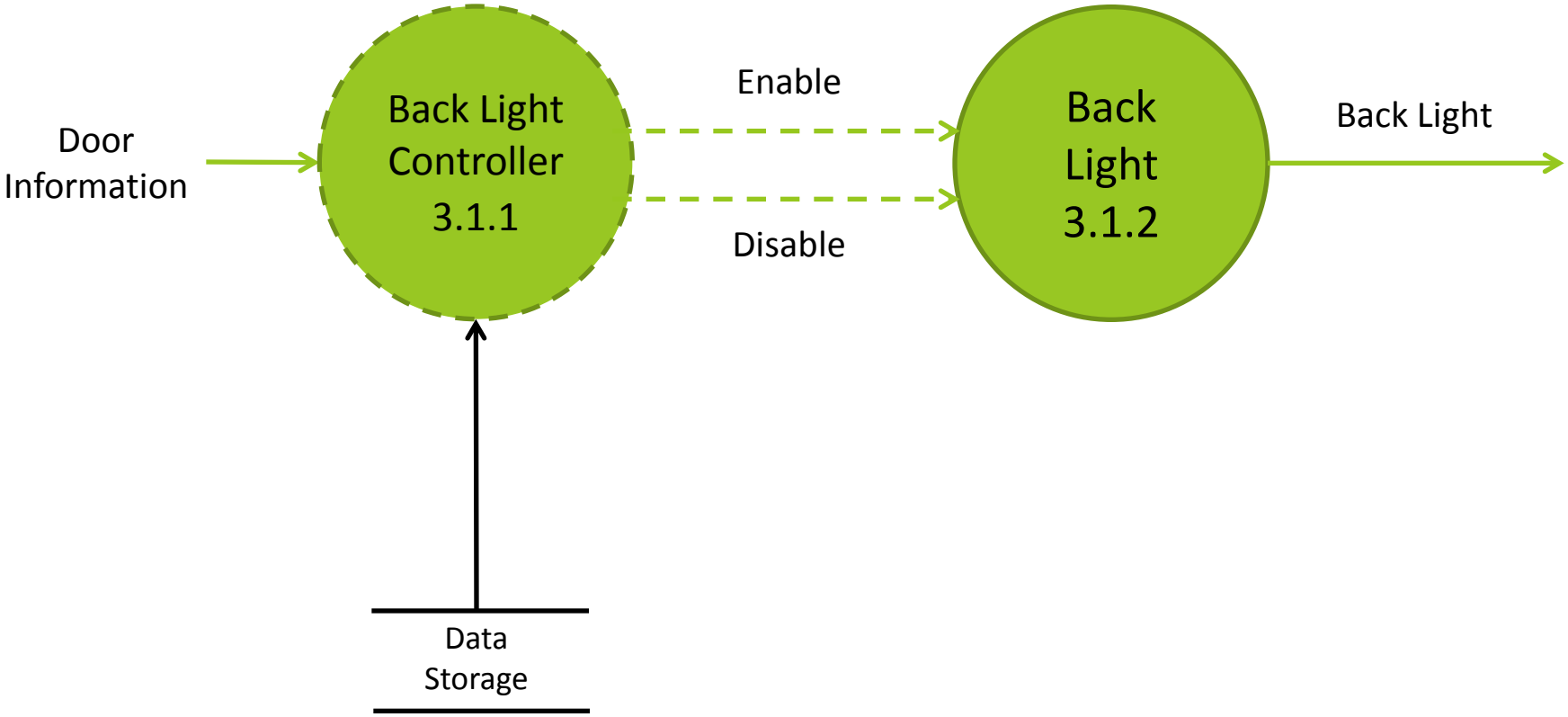
DFD Level 2 – Microwave



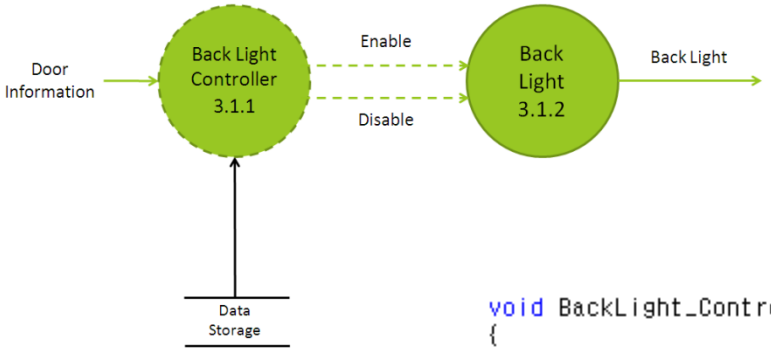
Data Dictionary – Level 2

	Output Interface 에서 활용할 Data들을 담은 구조체		
Data Storage	Variable ID	Description	Type
	setTime	설정 시간	Int
	currTemp	현재 온도	Double
	setTemp	설정 온도	Double
	isTime	시간 / 온도 모드	Boolean / 0 : False(온도 모드) 1: True(시간 모드)
	cookMode	조리 모드	Int / 0: 매뉴얼, 1: 떡, 2: 죽 3: 밥, 4: 국,찌개, 5: 피자
	s	시작 / 정지	Boolean / 0: False(정지) 1: True(시작)

DFD Level 3 – Microwave



DFD Level 3 – Microwave




```c
void Back_Light(bool enable) // 3.1.2
{
    if(enable == true)
        printf("₩033[93m%s₩n", "Display On");
    else
        printf("₩033[0m%s₩n", "Display Off");
}
```

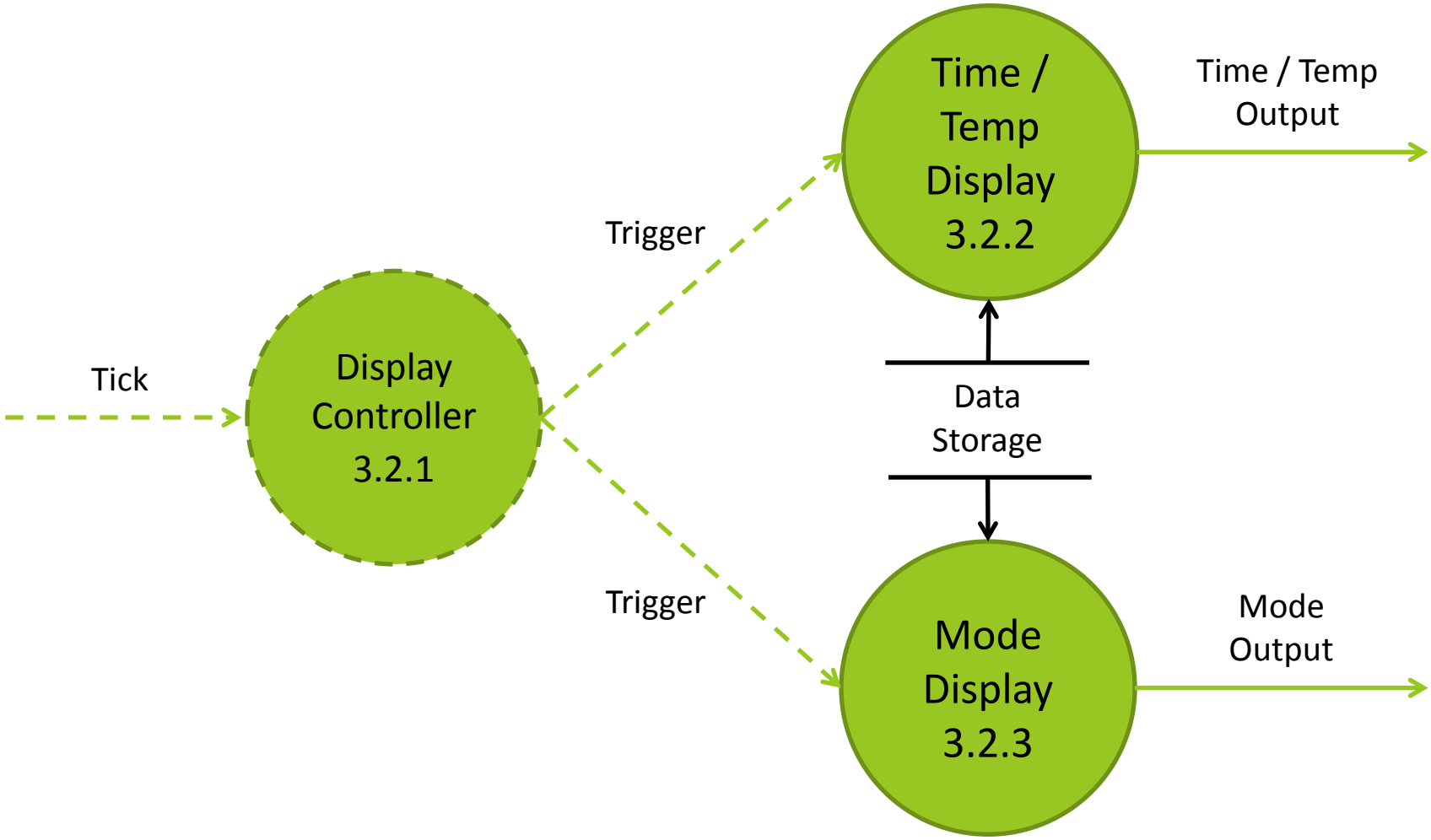
```c
void BackLight_Controller(bool s, bool door_info) // 3.1.1
{
    static int state = 1; // 1은 'BackLight off' state, 2는 'BackLight On' state
    static bool enable = false;

    switch(state)
    {
    case 1:
        if( door_info == true || ((door_info == false) && (s == true)) )
        {
            enable = true;
            state = 2;
        }
        break;
    case 2:
        if( (door_info == false) && (s == false) )
        {
            enable = false;
            state = 1;
        }
        break;
    }

    Back_Light(enable); // 3.1.2
}
```

Process Specification

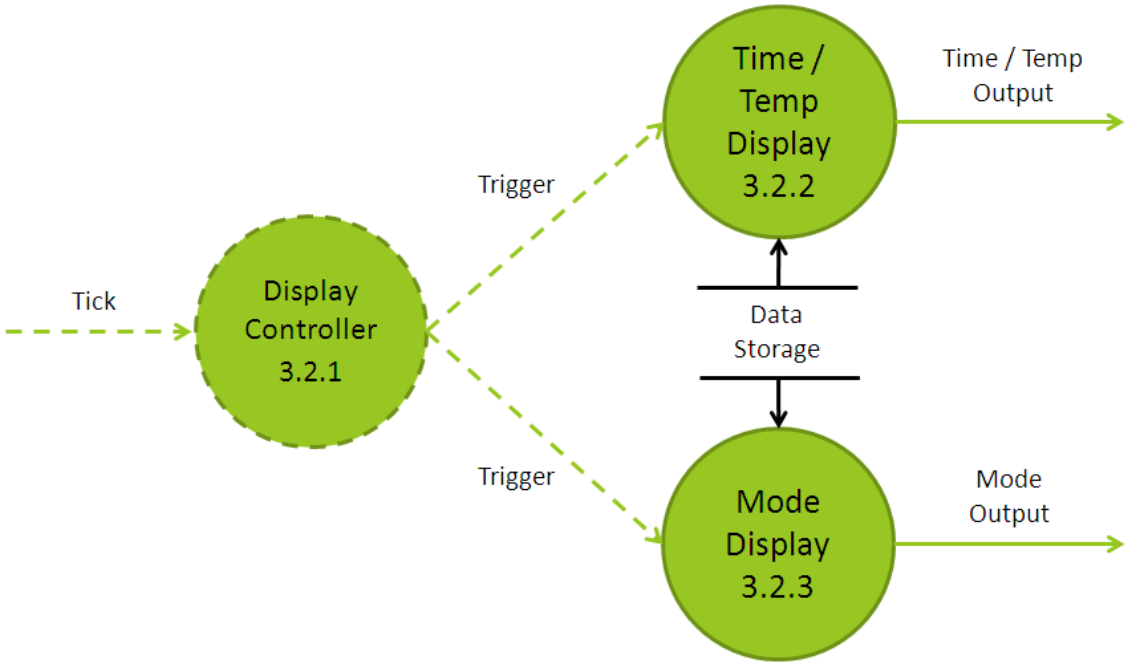
Reference No.	**3.1.1**
Name	Output Controller
Input (type)	door_info(boolean), Data_storage(Structure)
Output	Enable / Disable
Process Description	door_info와 data_storage.s 정보에 따라 Back Light Process를 Enable / Disable (data.storage.s == 1 || door_info == 1) : Back Light On

Reference No.	**3.1.2**
Name	Back Light
Input (type)	Enable, Disable
Output	Back Light
Process Description	Enable / Disable에 따라 Back Light가 On / Off

DFD Level 3 – Microwave

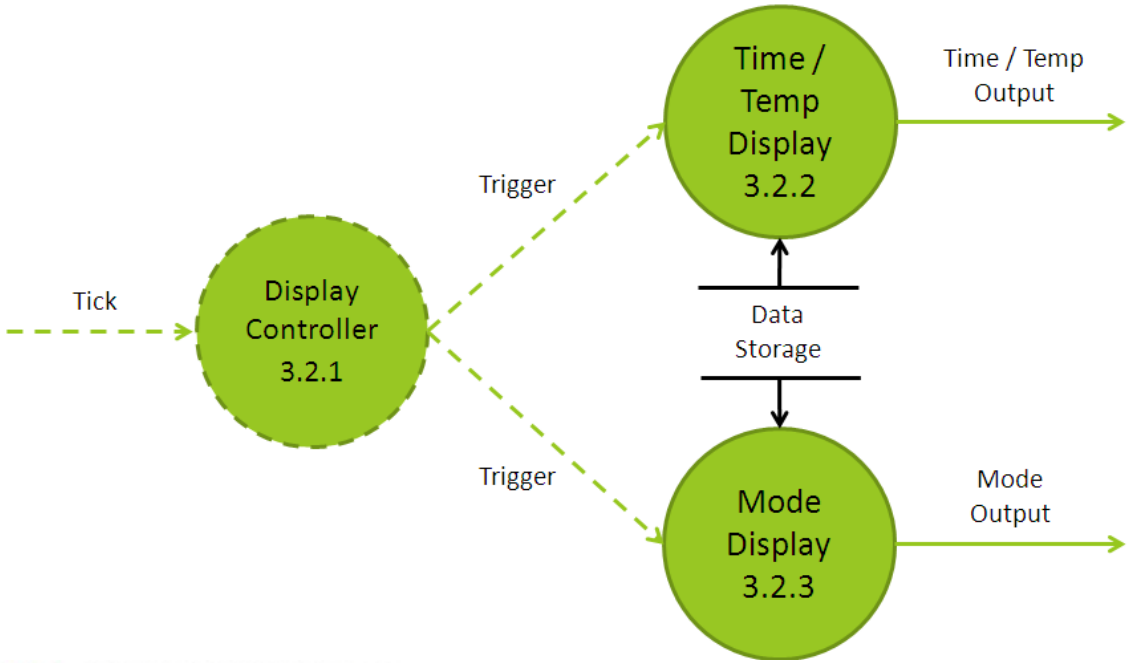


DFD Level 3 – Microwave



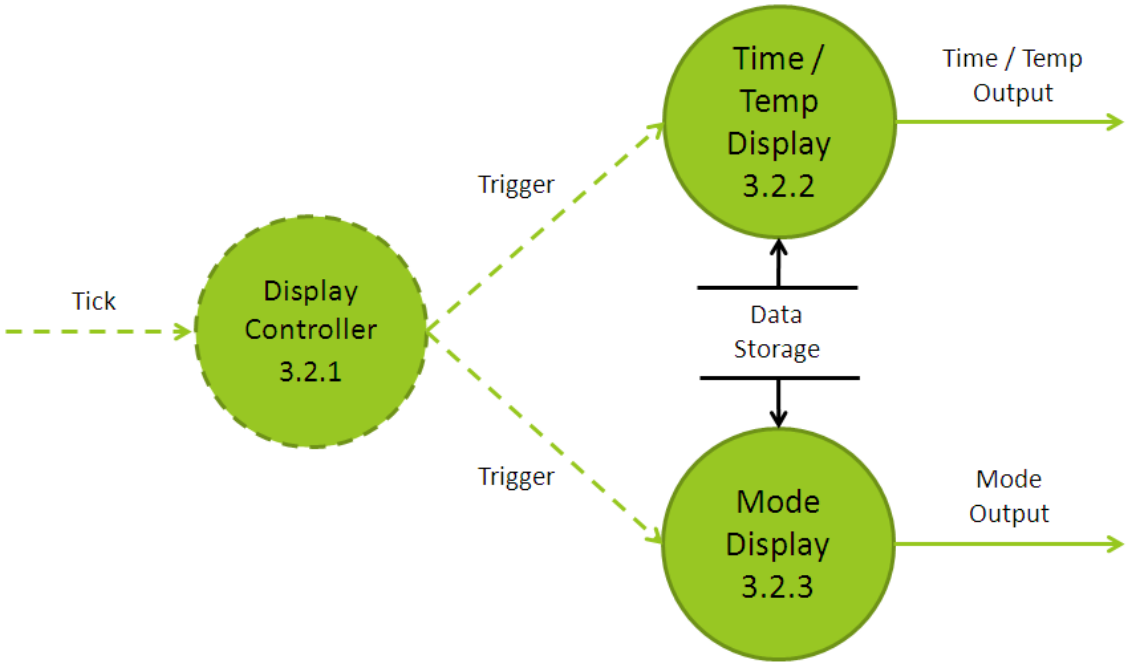

```
void Display_Controller()
{
    static int state = 1; // 1밖에 없음. 1은 Display Output

    Time_Temp_Display(Data_Storage);
    Mode_Display(Data_Storage.cookMode);
}
```

DFD Level 3 – Microwave




```
void Time_Temp_Display( )
{
    int min;
    if(Data_Storage.isTime == true)
    {
        min = (int) Data_Storage.setTime / 60;
        printf("%.2d : %.2d₩n", min, Data_Storage.setTime - (min * 60) );
    }
    else
    {
        printf("%.2d / %.2d º C₩n", (int) Data_Storage.currTemp, (int) Data_Storage.setTemp);
    }
}
```

DFD Level 3 – Microwave




```
void Mode_Display(int cookMode)
{
    static char mode_string[6][15] = {"manual", "떡", "죽", "밥", "국/찌개", "피자"};
    printf("%.2d : %s₩n", cookMode, mode_string[cookMode]);
}
```

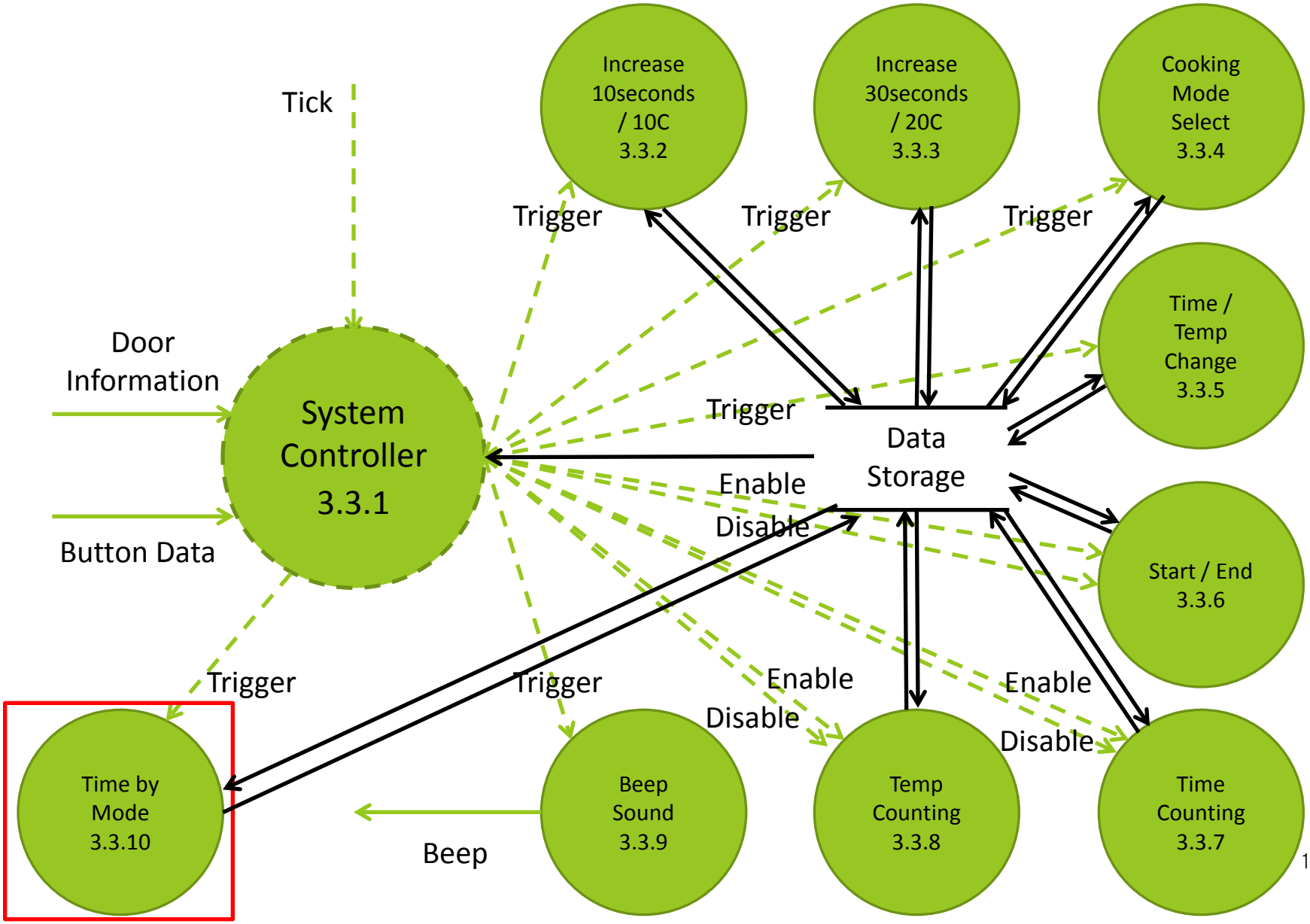
Process Specification

Reference No.	**3.2.1**
Name	Display Contoller
Input (type)	Tick
Output	Trigger
Process Description	Tick(=0.1seconds)마다 Time / Temp Display, Mode Display Process를 Trigger

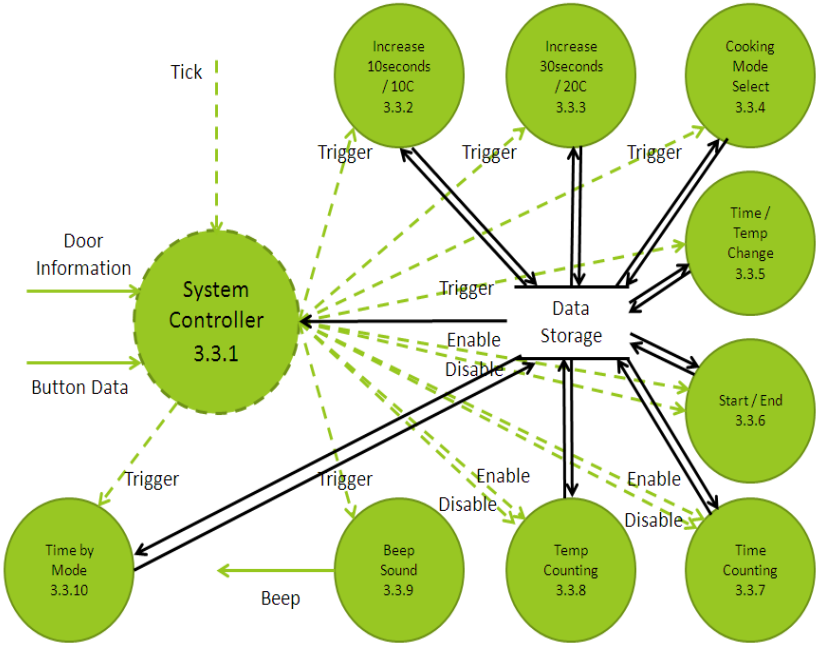
Reference No.	**3.2.2**
Name	Time/Temp Display
Input (type)	Trigger, Data_Storage(Structure)
Output	Time/Temp Output
Process Description	Trigger로 호출되어 if(Data_Storage.isTime==True) 이면 Data_Storage.setTime을 출력 If(Data_Storage.isTime==False)이면 Data_Storage.currTemp/Data_Storage.setTemp 출력

Reference No.	**3.2.3**
Name	Mode Display
Input (type)	Trigger, Data_Storage(Structure)
Output	Mode Output
Process Description	Trigger로 호출되어 Data_Storage.cookMode에 입력되어있는 정수의 값을 배열 참조 값으로 하여 String을 저장한 2차원 배열인 mode_string을 출력

DFD Level 3 – Microwave

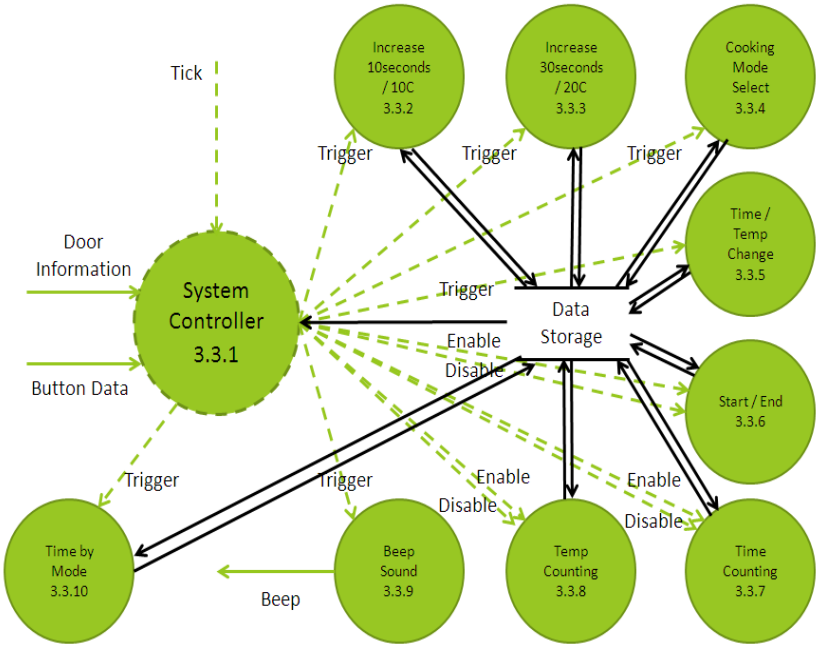


DFD Level 3 – Microwave



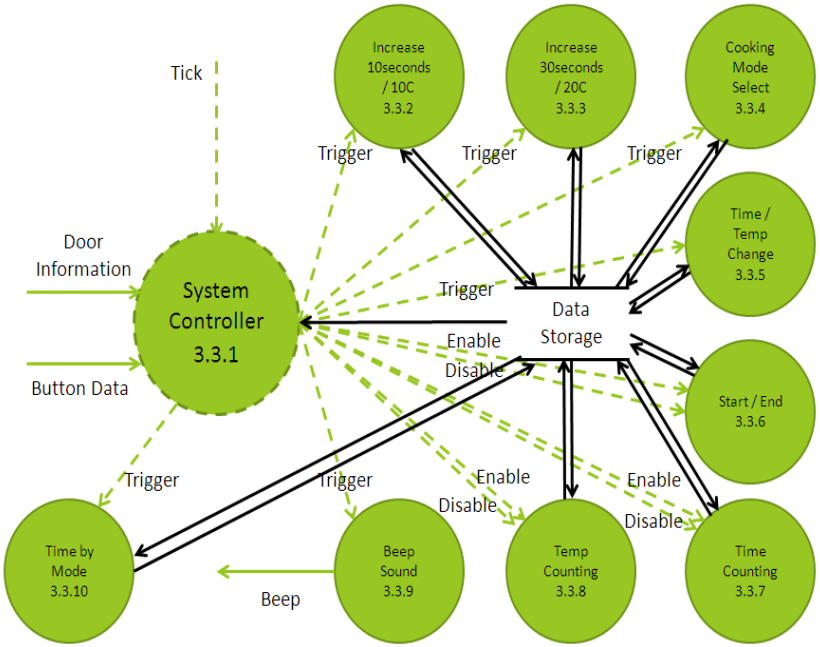

```
void System_Controller(bool door_info, char button) // 3.3.1
{
    static int state = 1; // 1은 'Stop' state
    static bool TimeCount = false; // Time_Counting 3.3.7의 Enable/Disable 상태를 저장
    static bool TempCount = false; // Temp_Counting 3.3.8의 Enable/Disable 상태를 저장

    switch(state)
    {
    case 1:
        if((button == 'a') && (Data_Storage.cookMode == 0))
        {
            state = 2;  //2는 'Increase 10seconds/10c' state
            Increase10seconds_10c();
        }
        else if((button == 'b') && (Data_Storage.cookMode == 0))
        {
            state = 3; //3은 'Increase 30seconds/20c' state
            Increase30seconds_20c();
        }
        else if(button == 'c')
        {
            state = 4;  //4는 'Time / Temp Change' state
            Time_Temp_Change();
        }
        else if((button == 'd') && Data_Storage.isTime)
        {
            state = 5;  //5는 'Mode Change' state
            Mode_Change();
            Time_by_Mode();
        }
        else if( (button == 'e') && !door_info && !(Data_Storage.isTime && (Data_Storage.setTime==0))
                                                 && !(!Data_Storage.isTime && (Data_Storage.setTemp==20)))
        {
            state = 6;  //6은 'Time or Temp' state
            Start_End(ENABLE); // true는 Enable, false은 Disable.
        }
        break;
    case 2:
        state = 1; // 'Stop' state로 돌아감
        break;
    case 3:
        state = 1; // 'Stop' state로 돌아감
        break;
    case 4:
        state = 1; // 'Stop' state로 돌아감
        break;
    case 5:
        state = 1; // 'Stop' state로 돌아감
        break;
```

DFD Level 3 – Microwave



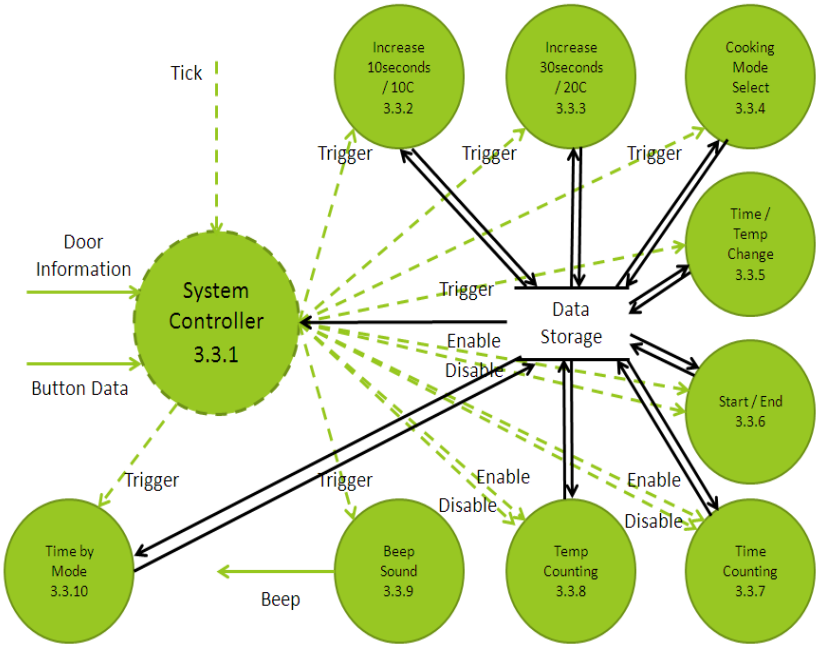

```
    case 6:
        if(Data_Storage.isTime) // Time 이면
        {
            state = 7; // 7은 'Operation by Time' state
            TimeCount = ENABLE;
        }
        else if(!Data_Storage.isTime) // Temp 이면
        {
            state = 8; // 8은 'Operation by Temp' state
            TempCount = ENABLE;
        }
        break;
    case 7:
        if(button == 'e')
        {
            state = 1; // 'Stop' state로 돌아감
            Start_End(DISABLE);
            TimeCount = DISABLE;
            Time_by_Mode();
        }
        else if(Data_Storage.setTime == 0)
        {
            state = 9; // 9는 'Done' state임
            TimeCount = DISABLE;
        }
        break;
    case 8:
        if(button == 'e')
        {
            state = 1; // 'Stop' state로 돌아감
            Start_End(DISABLE);
            TempCount = DISABLE;
            Data_Storage.setTemp = 20;
            Data_Storage.currTemp = 20;
        }
        else if(Data_Storage.setTemp == Data_Storage.currTemp)
        {
            state = 9; // 9는 'Done' state임
            TempCount = DISABLE;
            Data_Storage.currTemp = 20;
        }
        break;
    case 9:
        state = 1; // 'Stop' state로 돌아감
        Beep_Sound(); // Trigger 'Beep Sound'
        Time_by_Mode(); // Trigger 'Time by Mode'
        Start_End(DISABLE); // Disable "Start / End"
        break;
    }
    Time_Counting(TimeCount);
    Temp_Counting(TempCount);
}
```

DFD Level 3 – Microwave



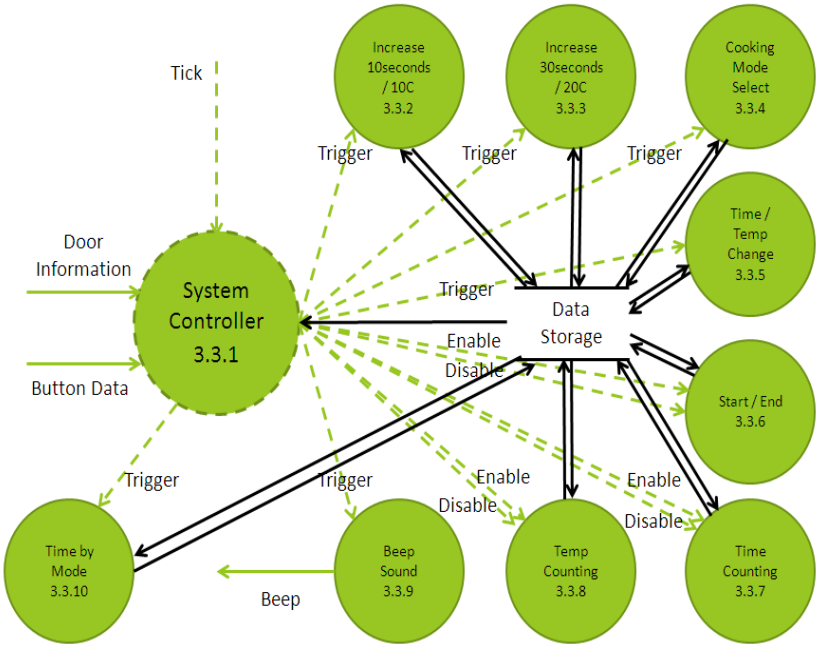

```
void increase10seconds_10c()
{
    if(Data_Storage.isTime == true)
    {
        Data_Storage.setTime +=  10;
        if(Data_Storage.setTime > 600)
            Data_Storage.setTime = 10;
    }
    else
    {
        Data_Storage.setTemp += 10;
        if(Data_Storage.setTemp > 90)
            Data_Storage.setTemp = 30;
    }
}
```

DFD Level 3 – Microwave




```
void Increase30seconds_20c( )
{
    if(Data_Storage.isTime == true)
    {
        if(Data_Storage.setTime == 600)
            Data_Storage.setTime = 30;
        else
        {
            Data_Storage.setTime += 30;
            if(Data_Storage.setTime > 600)
                Data_Storage.setTime = 600;
        }
    }
    else
    {
        if(Data_Storage.setTemp == 90)
            Data_Storage.setTemp = 50;
        else
        {
            Data_Storage.setTemp += 30;
            if(Data_Storage.setTemp > 90)
                Data_Storage.setTemp = 90;
        }
    }
}
```

DFD Level 3 – Microwave

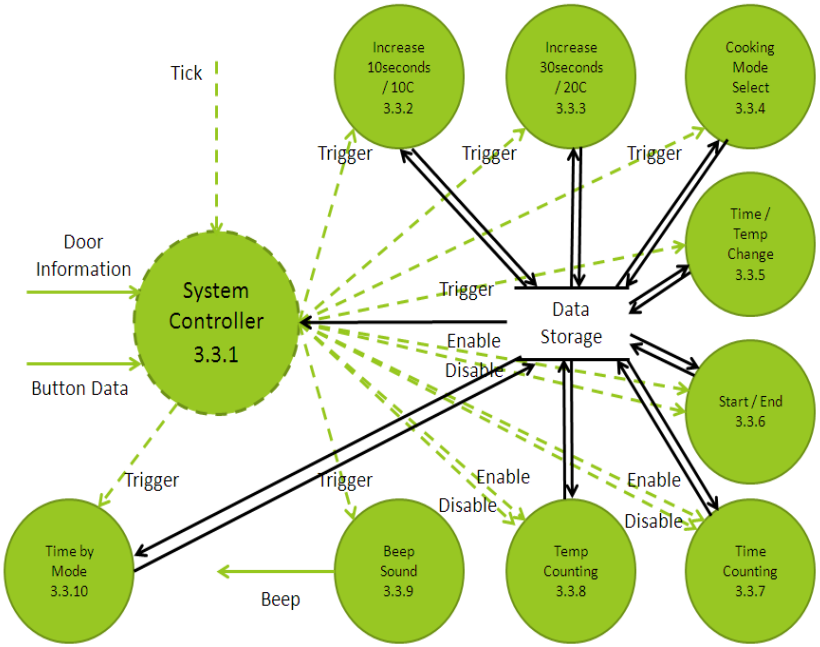



```
void Mode_Change( ) // Cooking Mode Select 3.3.4
{
    Data_Storage.cookMode = (Data_Storage.cookMode+1)%6;
}
```

```
void Time_Temp_Change( ) // 3.3.5
{
    Data_Storage.isTime = !Data_Storage.isTime;
    if(Data_Storage.isTime == false)
    {
        Data_Storage.cookMode = 0;
        Data_Storage.setTemp  = 20;
    }
    else
        Data_Storage.setTime  = 0;
}
```

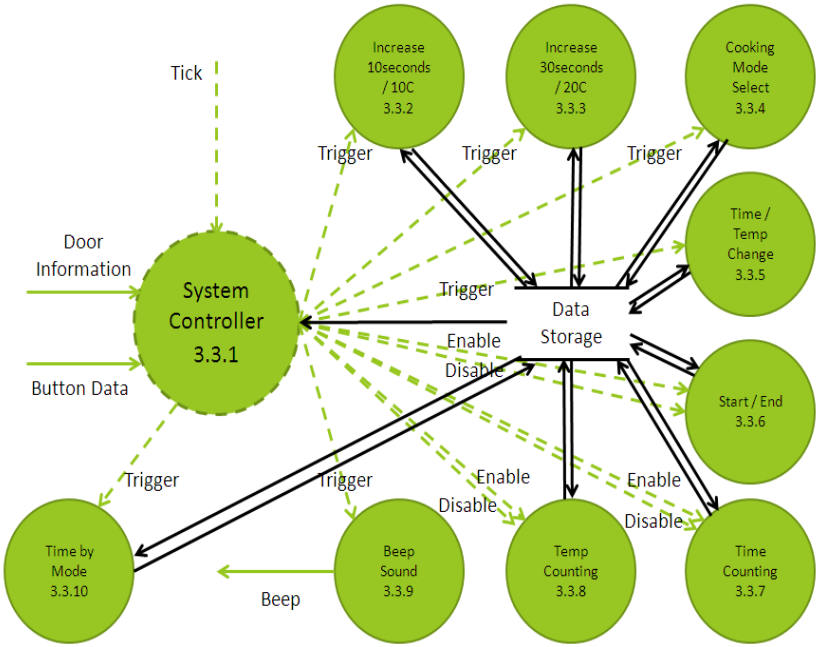
```
void Start_End(bool enable)
{
    Data_Storage.s = enable;
}
```

DFD Level 3 – Microwave




```
void Time_Counting(bool enable)
{
    static int tick = 0;
    if(enable == true)
    {
        tick = (tick + 1)%10;
        if(tick == 0)
            Data_Storage.setTime--;
    }
    else
        tick = 0;
}

void Temp_Counting(bool enable)
{
    static int tick = 0;
    if(enable == true)
    {
        tick = (tick + 1)%30;
        if(tick == 0)
            Data_Storage.currTemp += 10;
    }
    else
        tick = 0;
}

void Beep_Sound() // 3.3.9
{
    Beep(900, 500);
    usleep(TICK * 5000);
    Beep(900, 500);
    usleep(TICK * 5000);
    Beep(900, 500);
}
```

DFD Level 3 – Microwave




```
void Time_by_Mode() // 3.3.10
{
    if(Data_Storage.cookMode == 0)
        Data_Storage.setTime = 0;
    else if(Data_Storage.cookMode == 1)
        Data_Storage.setTime = 60;
    else if(Data_Storage.cookMode == 2)
        Data_Storage.setTime = 90;
    else if(Data_Storage.cookMode == 3)
        Data_Storage.setTime = 120;
    else if(Data_Storage.cookMode == 4)
        Data_Storage.setTime = 300;
    else if(Data_Storage.cookMode == 5)
        Data_Storage.setTime = 120;
}
```

Process Specification

Reference No.	**3.3.1**
Name	System Controller
Input (name / type)	door_info(boolean), button(char), Tick, Data_Storage(sturucture)
Output	Trigger, Enable, Disable

Process Description	if(button == ‘a’)일 때 Increase 10seconds / 10C Process를 Trigger
	if(button == ‘b’) 일 때 Increase 30seconds / 20C Process를 Trigger
	if(button == ‘c’) 일 때 Cooking Mode Select Process를 Trigger
	if(button == ‘d’) 일 때 Time / Temp Change Process를 Trigger
	if(button == ‘e’) 일 때 Start / End Process를 Enable / Disable
	if((Data_Storage.isTime==True)&&(Data_Storage.s==True)) 일 때 Time Counting Process를 Enable / else 일 때 Disable
	if((Data_Storage.isTime==False)&&(Data_Storage.s==True)) 일 때 Temp Counting Process를 Enable / else 일 때 Disable
	if((((Data_Storage.isTime==True)&&(Data_Storage.setTime==0))
	||((Data_Storage.isTime==False)&&(Data_Storage.setTemp==Data.Storage.currTemp)))
	&&(Data_Storage.s==True))일 때 Beep Sound Process를 Trigger

Process Specification

Reference No.	**3.3.2**
Name	Increase 10seconds / 10C
Input	Trigger, Data_Storage(structure)
Output	Data_Storage(structure)
Process Description	if((Data_Storage.isTime==True)&&(Data_Storage.cookMode==0)) 일 때 Data_Storage.setTime을 10 증가, if((Data_Storage.isTime==False)일 때 Data_Storage.setTemp 를 10 증가

Reference No.	**3.3.3**
Name	Increase 30seconds / 20C
Input	Trigger, Data_Storage(structure)
Output	Data_Storage(structure)
Process Description	if((Data_Storage.isTime==True)&&(Data_Storage.cookMode==0)) 일 때 Data_Storage.setTime을 30 증가, if((Data_Storage.isTime==False)일 때 Data_Storage.setTemp 를 20 증가

Process Specification

Reference No.	**3.3.4**
Name	Cooking Mode Select
Input	Trigger, Data_Storage(structure)
Output	Data_Storage(structure)
Process Description	if(Data_Storage.isTime==True) 일 때 Data_Storage.cookMode = (Data_Storage.cookMode+1)%6

Reference No.	**3.3.5**
Name	Time / Temp Change
Input	Trigger, Data_Storage(structure)
Output	Data_Storage(structure)

Process Description	if(Data_Storage.isTime==True) 일 때
	Data_Storage.isTime = False, Data_Storage.cookMode = 0, Data_Storage.setTemp = 20
	if(Data_Storage.isTime==False) 일 때
	Data_Storage.isTime = True, Data_Storage.setTime = 0

Process Specification

Reference No.	**3.3.6**
Name	Start / End
Input	Enable, Disable, Data_Storage(structure)
Output	Data_Storage(structure)
Process Description	Enable, Disable에 따라 Enable이면 Data_Storage.s = True, Disable이면 Data_Storage.s = False

Reference No.	**3.3.7**
Name	Time Counting
Input	Enable, Disable, Data_Storage(structure), Tick
Output	Data_Storage(structure)
Process Description	Enable, Disable에 따라 Enable이면 10 Tick마다 Data_Storage.setTime = Data_Storage.setTime – 1

Process Specification

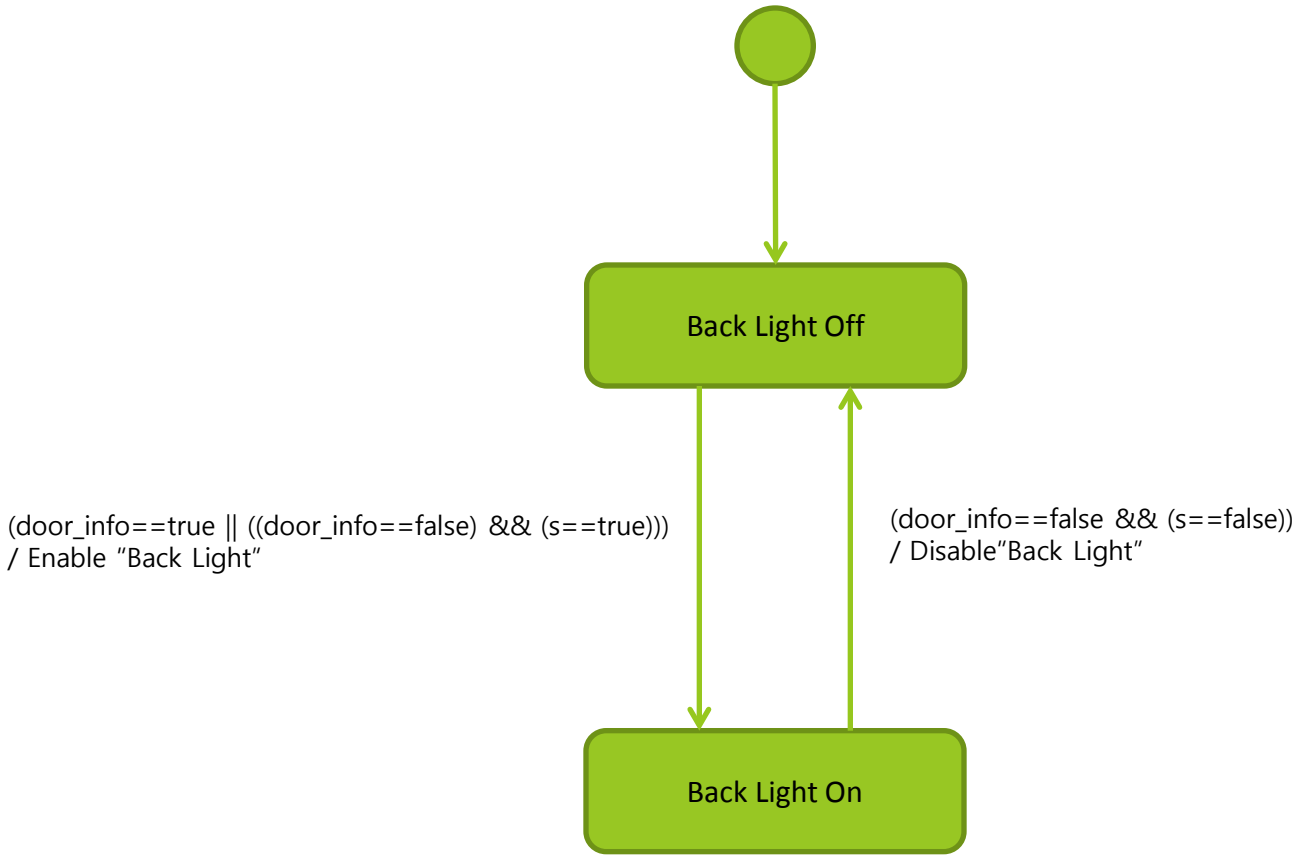
Reference No.	**3.3.8**
Name	Temp Counting
Input	Enable, Disable, Data_Storage(structure)
Output	Data_Storage(structure)
Process Description	Enable, Disable에 따라 Enable이면 30 Tick마다 Data_Storage.setTemp = Data_Storage.setTemp + 10

Reference No.	**3.3.9**
Name	Beep Sound
Input	Trigger
Output	Beep
Process Description	Trigger 신호를 받아 Beep 음 출력(3초)

Reference No.	**3.3.10**
Name	Time by Mode
Input	Trigger, Data_Storage(structure)
Output	Data_Storage(structure)
Process Description	Data_Storage.cookMode에 따라 Data_Storage.setTime을 설정

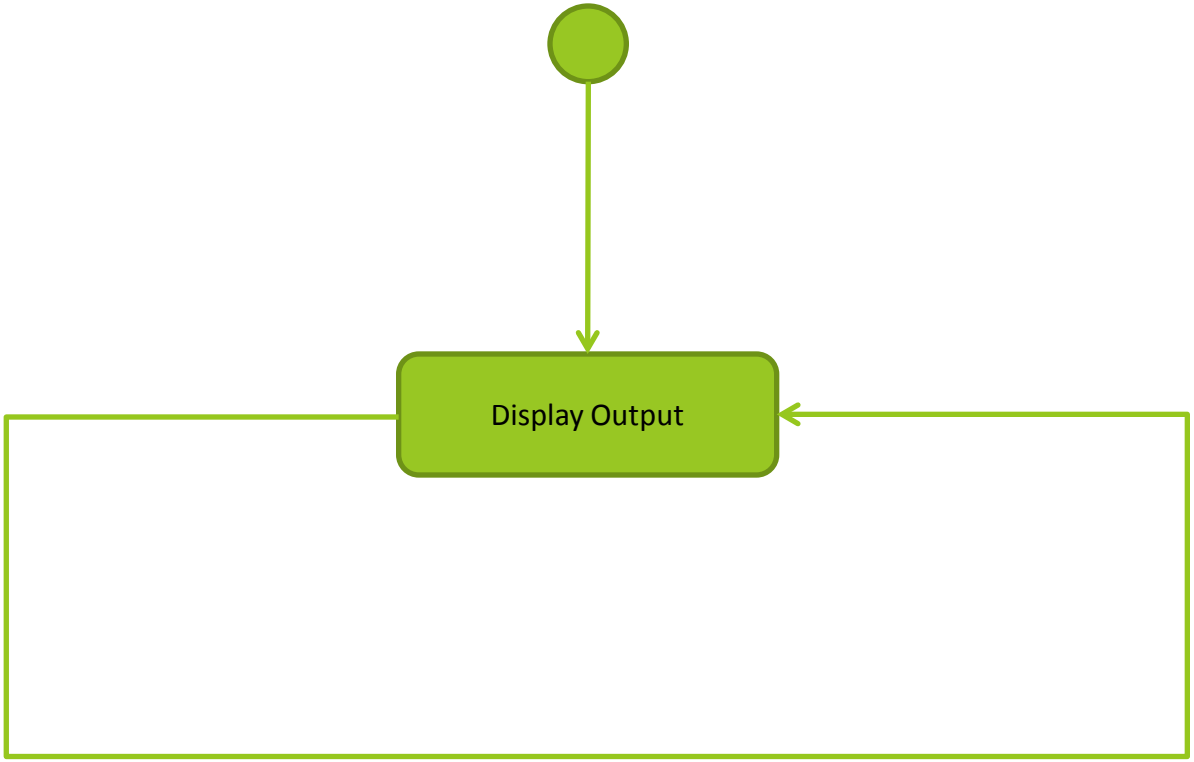
DFD Level 4 – Microwave

- State Transition Diagram 3.1.1



DFD Level 4 – Microwave

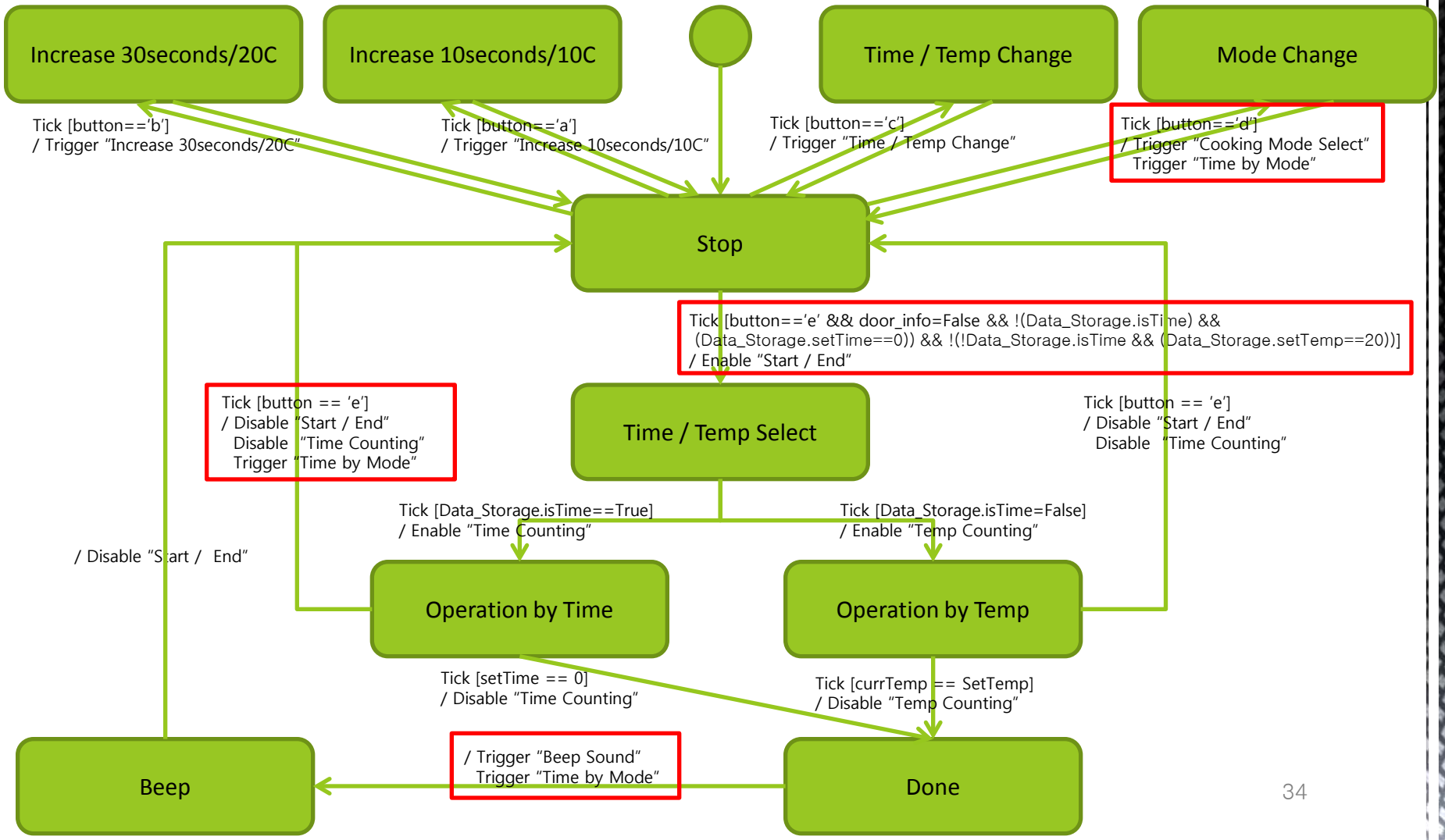
- State Transition Diagram 3.2.1



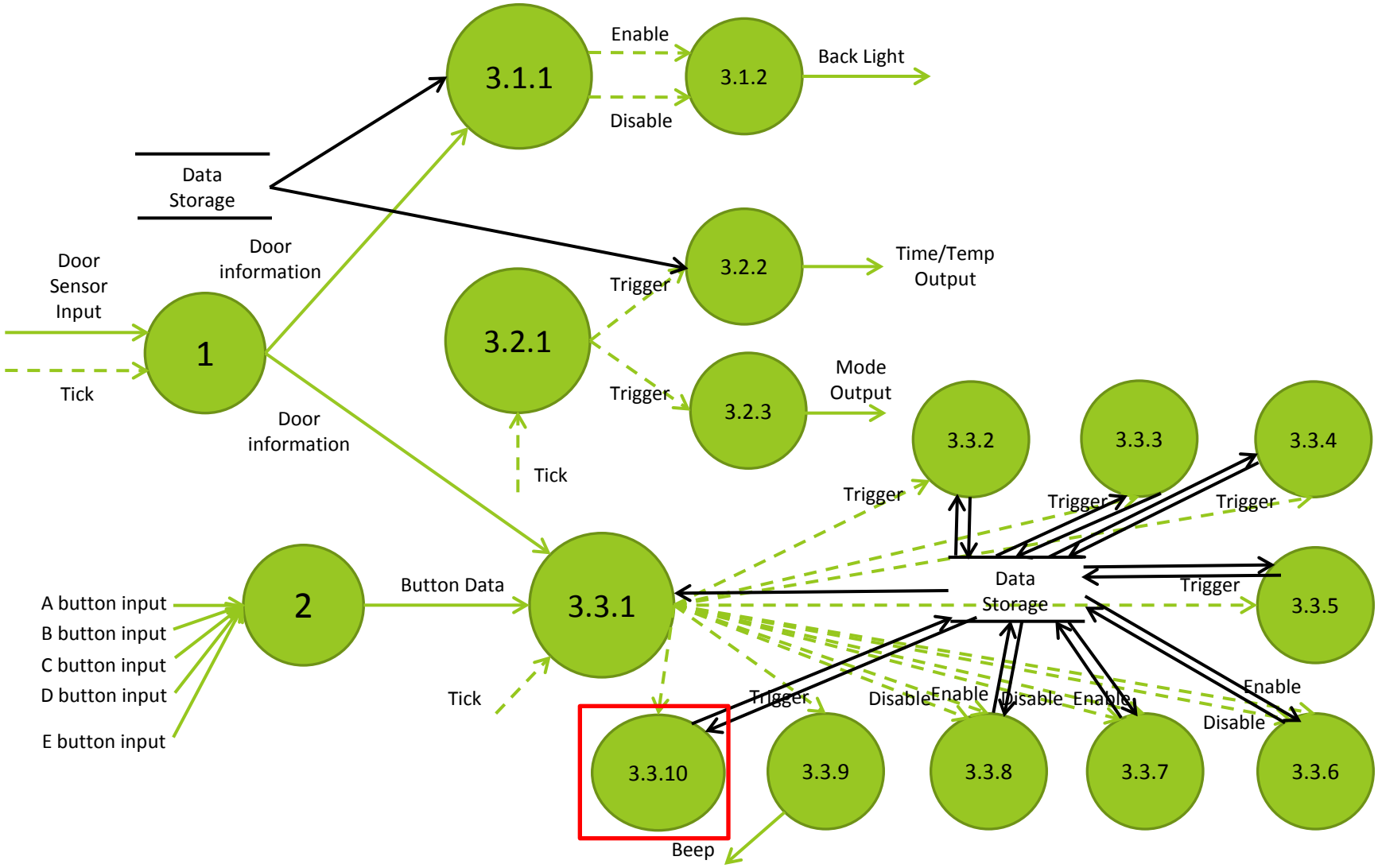
Tick
/Trigger "Time/Temp Display
Trigger "Mode Display"

DFD Level 4 – Microwave

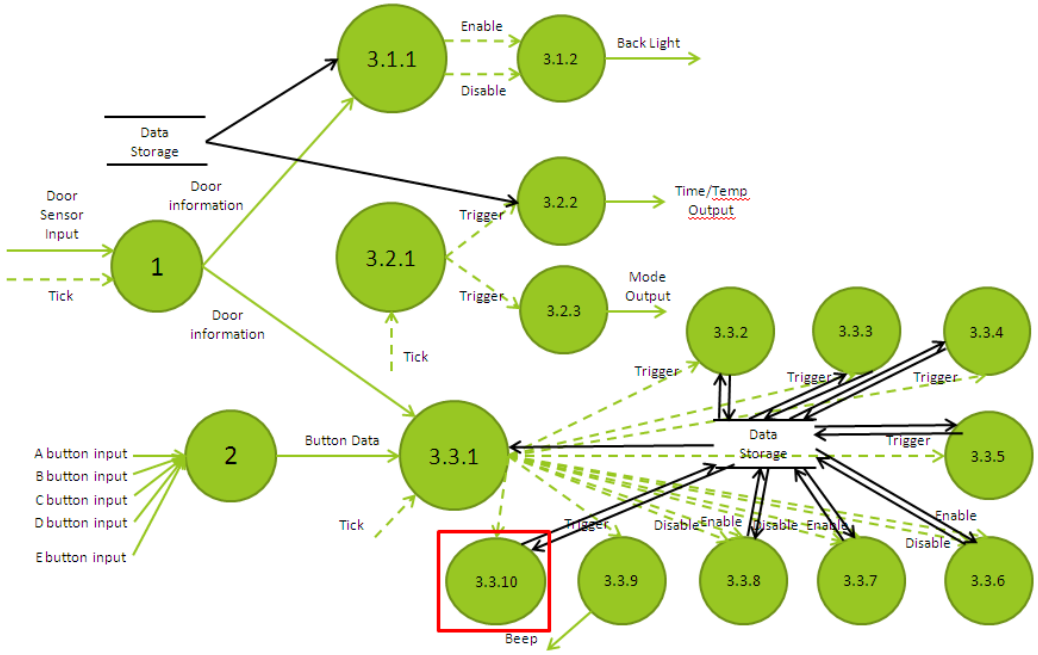
- State Transition Diagram 3.3.1



DFD - Microwave

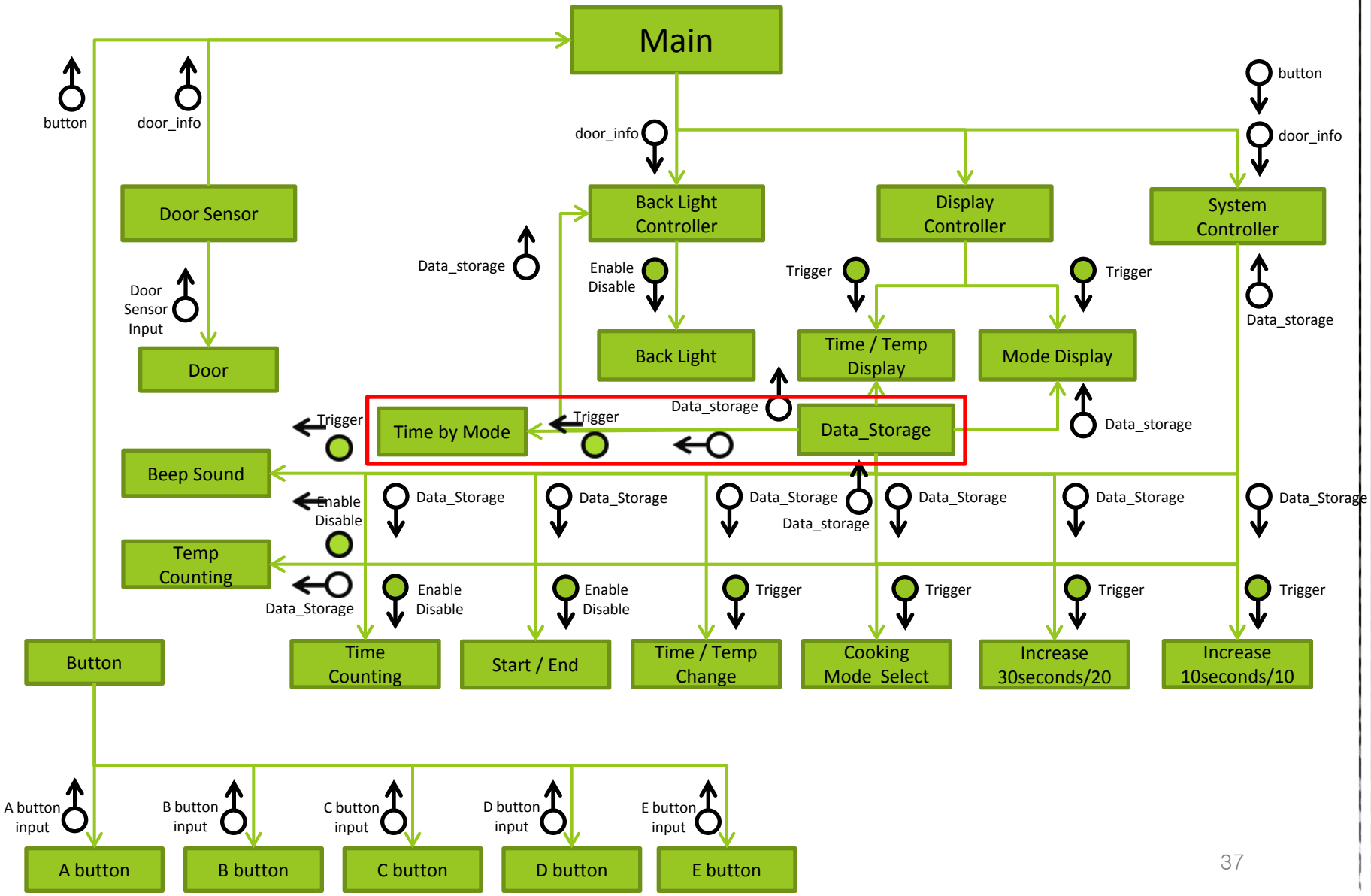


DFD - Microwave

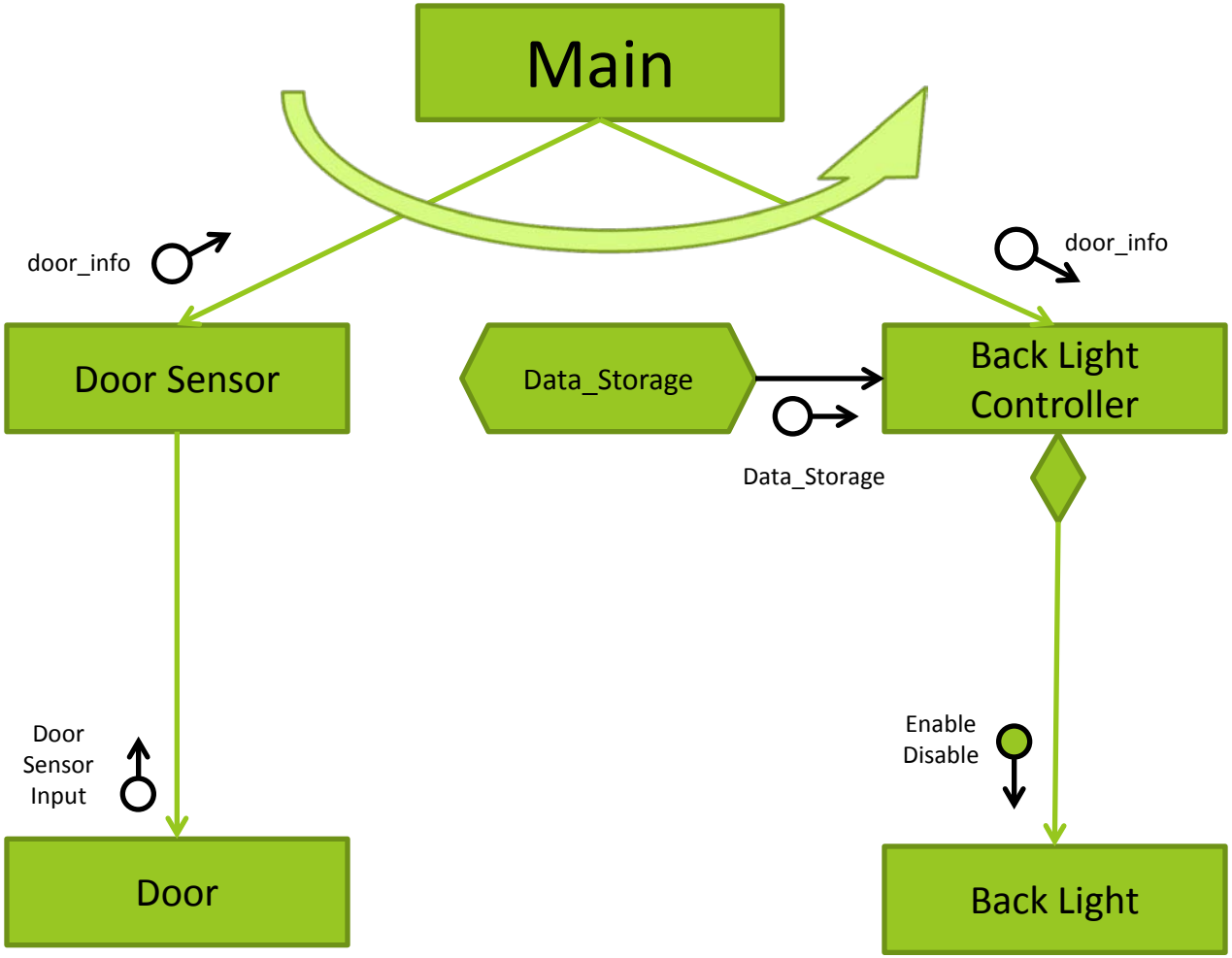



```c
void main()
{
    char button = '₩t'; // 입력된 버튼의 값을 저장하는 변수. Button Data에 해당.
    bool door_info = false; // 문의 개폐 상태를 저장하는 변수. 열림/닫힘(TRUE/FALSE)

    set_stdin(); // 콘솔 입력을 위한 함수

    while(button != 'q')
    {
        button = Button(); // 'Button 2' Data Process에 해당하는 함수.
        door_info = Door_Sensor(button, door_info); // 'Door Sensor 1' Data Process에 해당하는 함수.
        BackLight_Controller(Data_Storage.s, door_info); // 3.1.1
        Display_Controller(); // 3.2.1
        System_Controller(door_info, button); // 3.3.1
        //printf("문 상태 = %s₩n", door_info ? "열림":"닫힘");

        usleep(TICK * 1000); // TICK (0.1초) 만큼 딜레이를 주는 함수
        system("clear");
    }

    reset_stdin(); // 콘솔 입력을 위한 함수
}
```

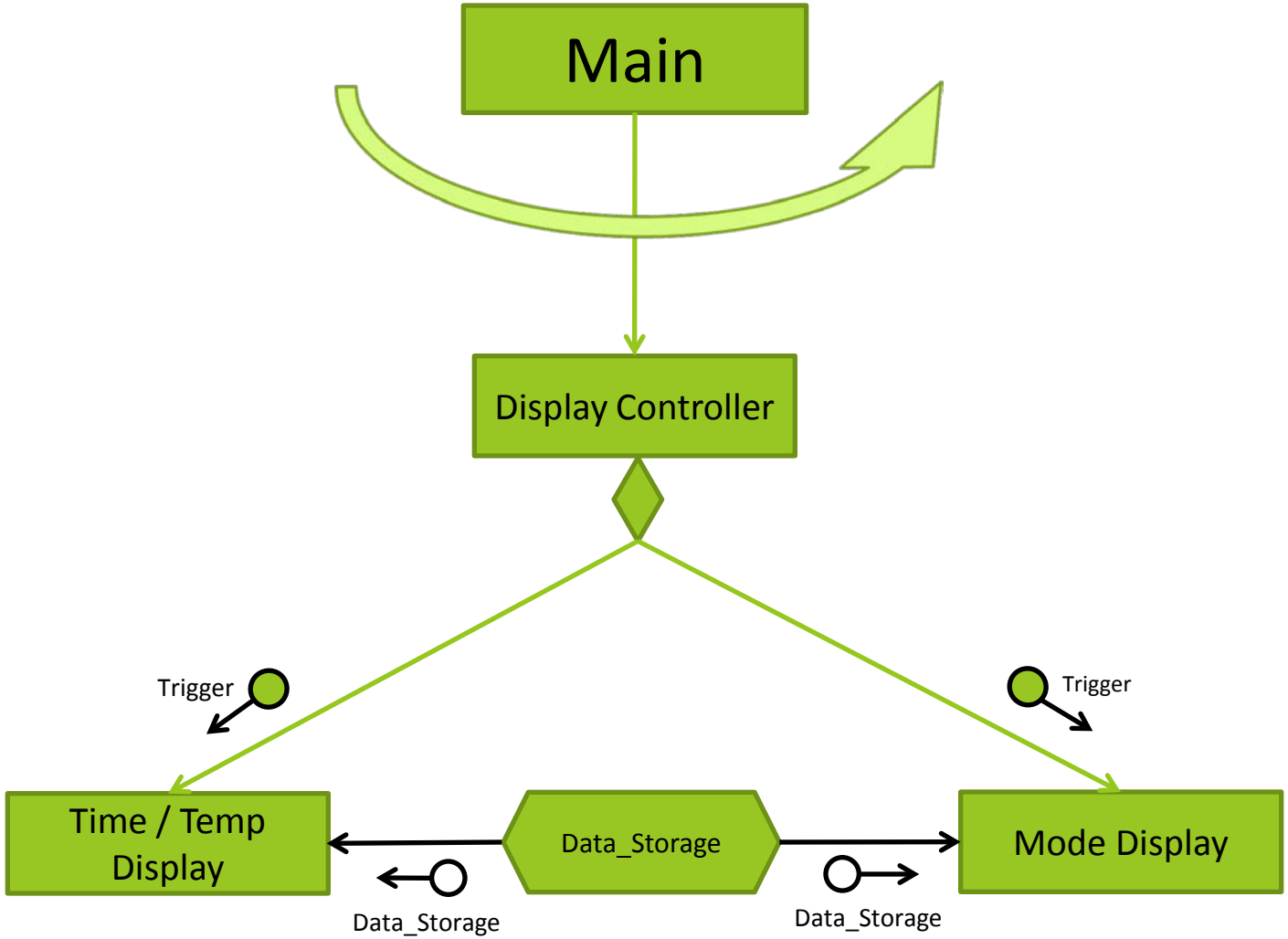
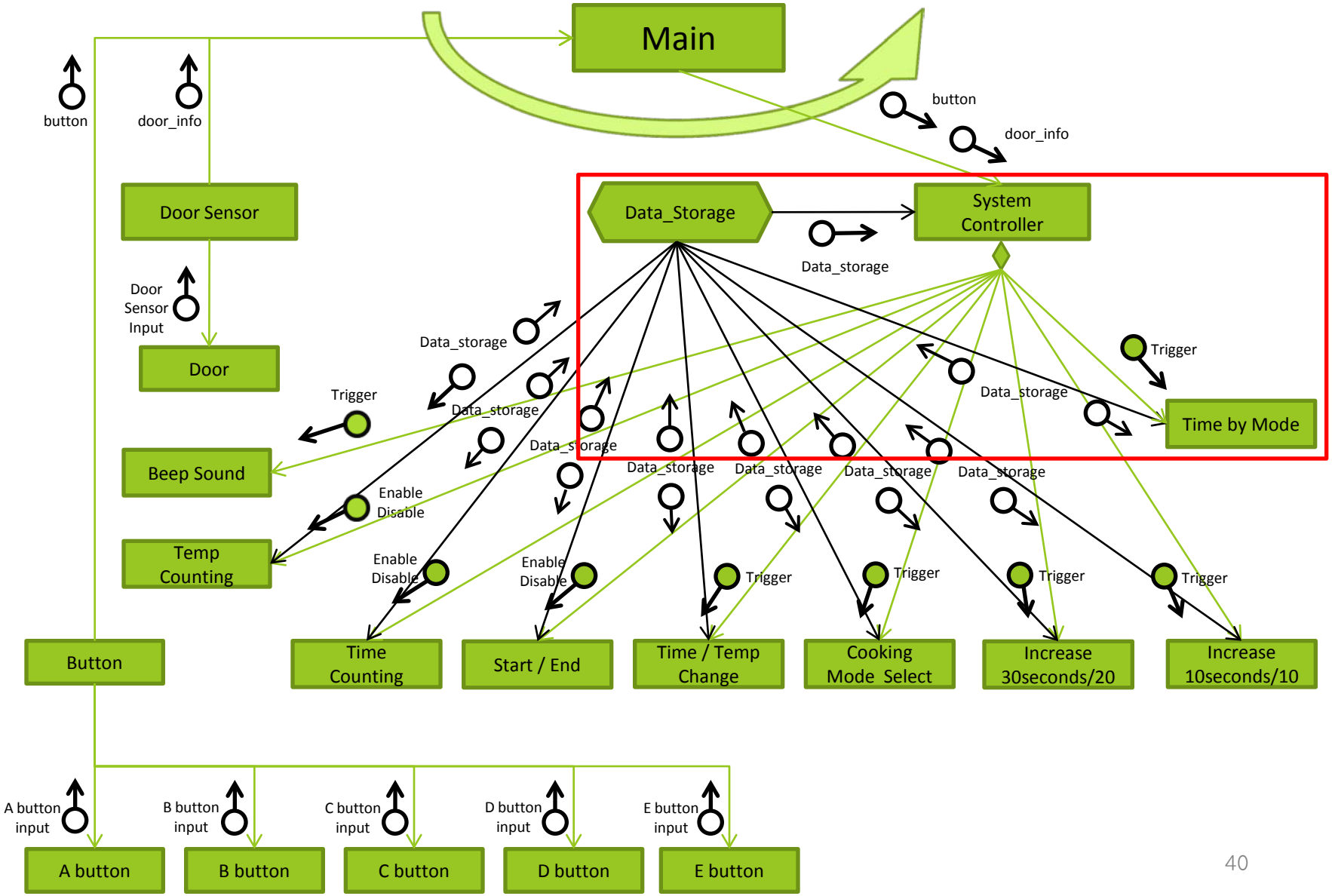
Structured Chart



Structured Charts – Back Light Controller



Structured Charts – Display Controller



Structured Charts – System Controller