

Final Report For Digital Watch System

Team 4

김선우 이지우 김도연 신호혜

Contents



1 Revision

1.1 DFD

1.2 specification

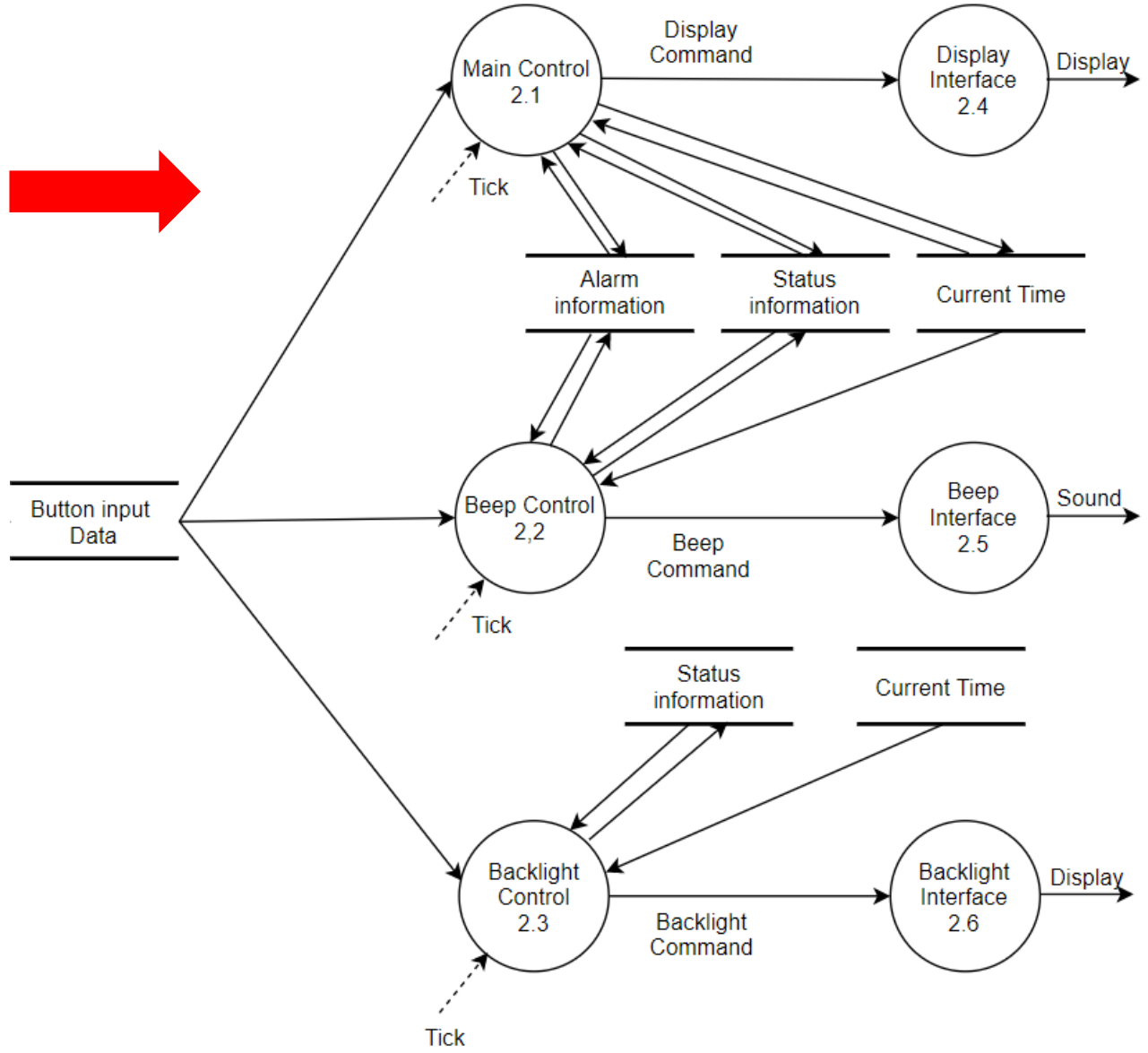
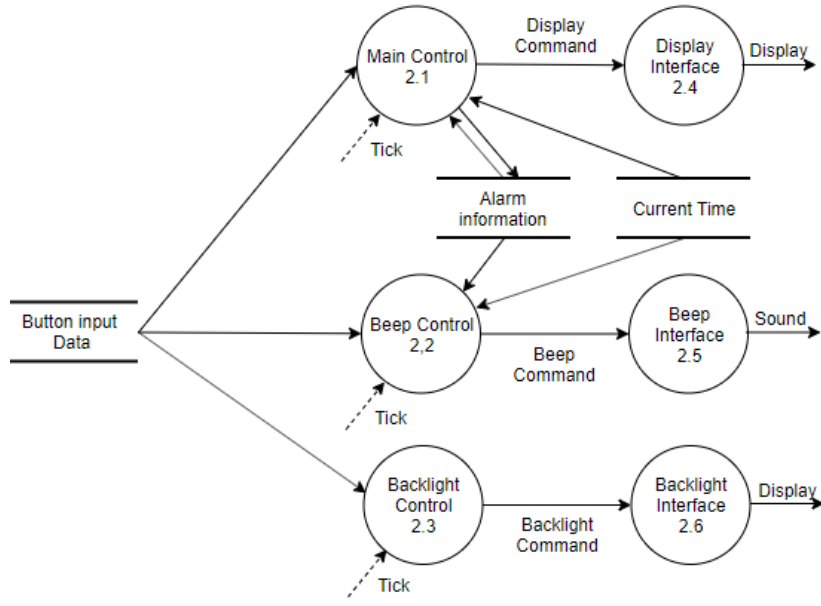
2 Implementation

3 DEMO

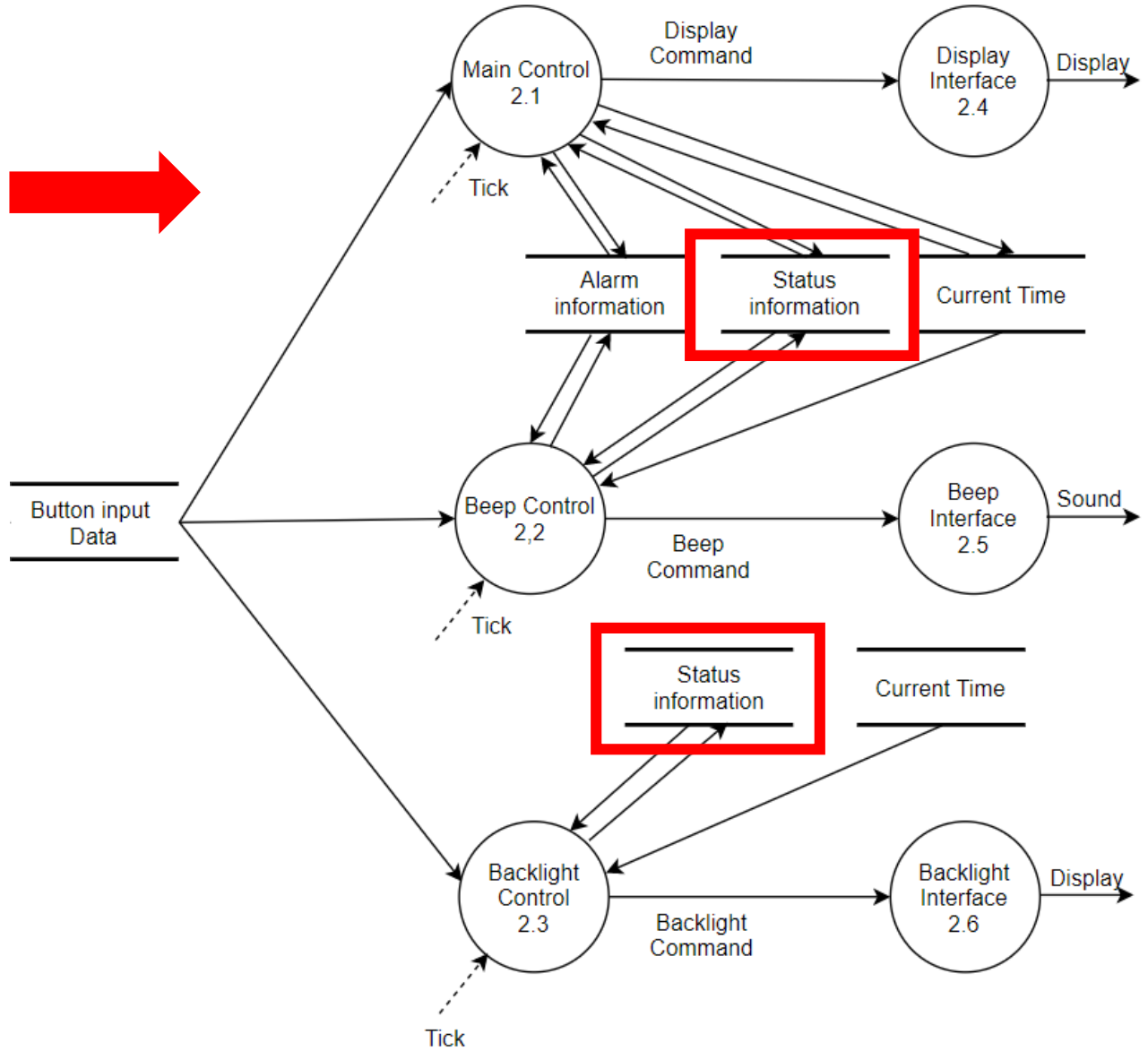
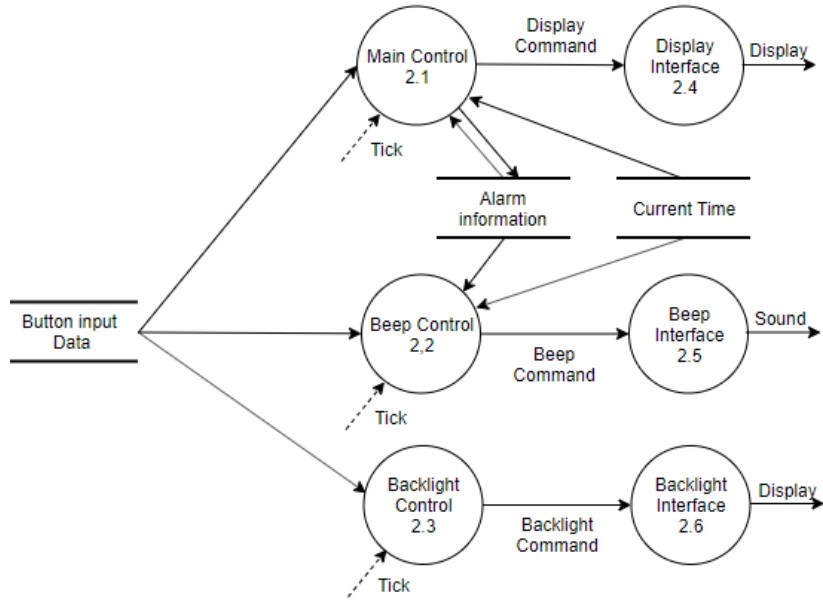
4 Q & A

Revision

Revision



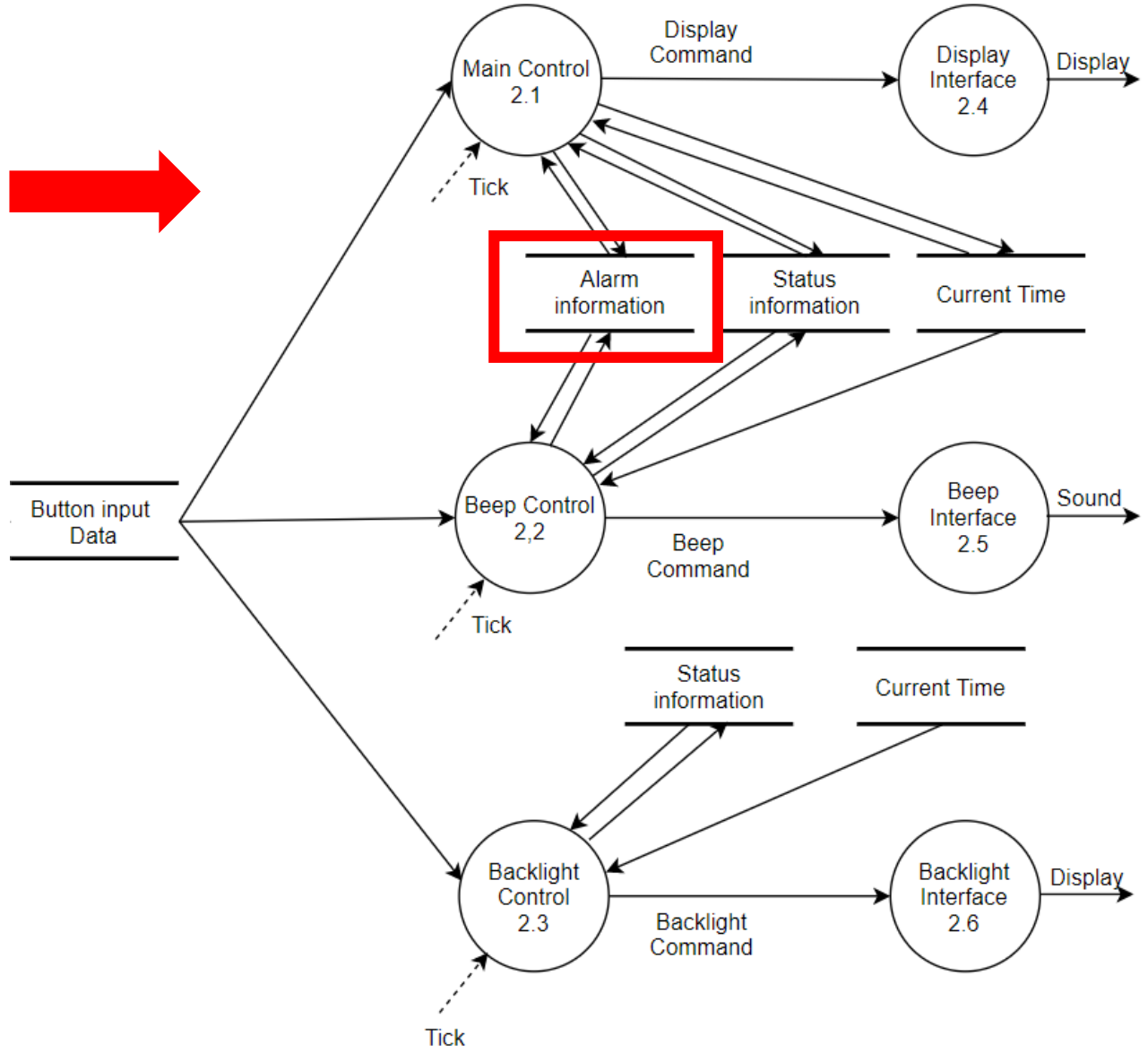
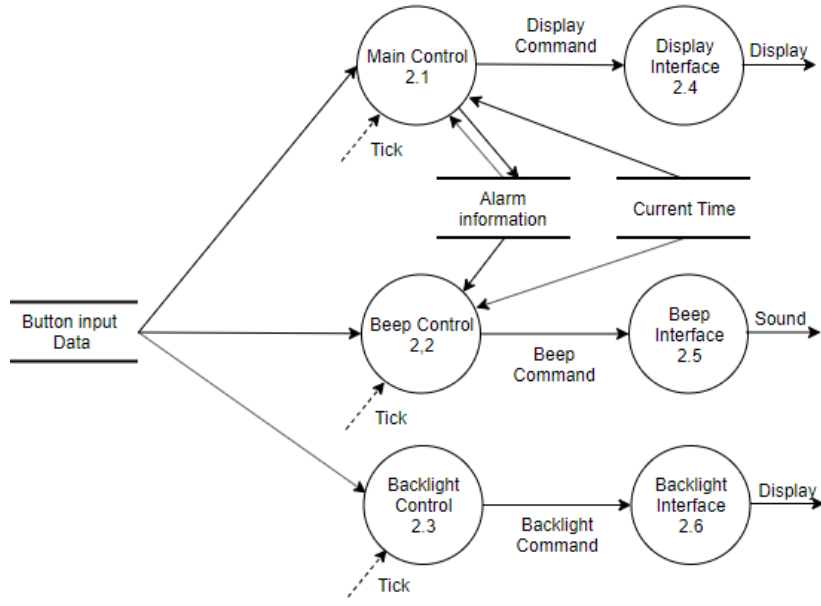
Revision



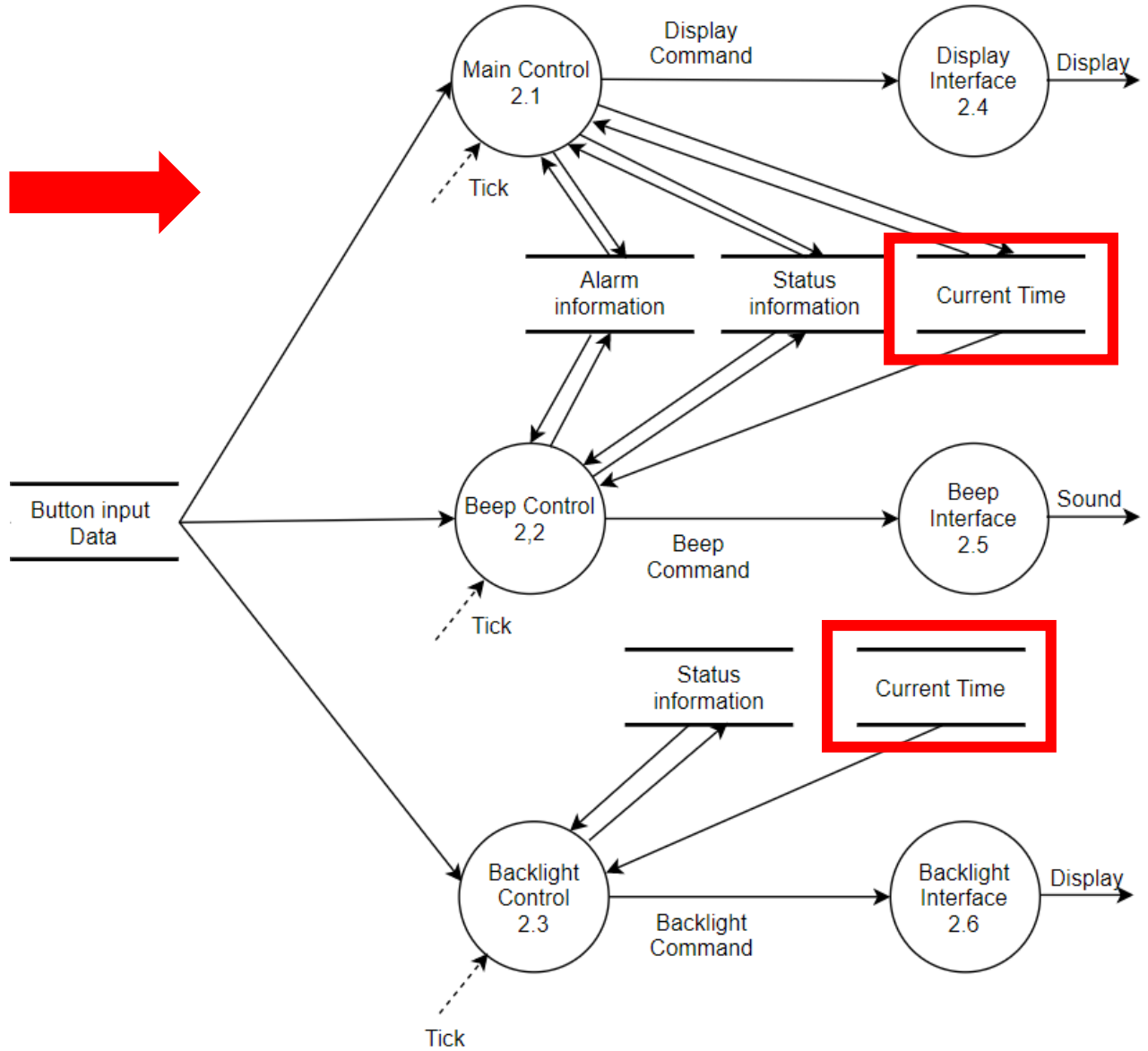
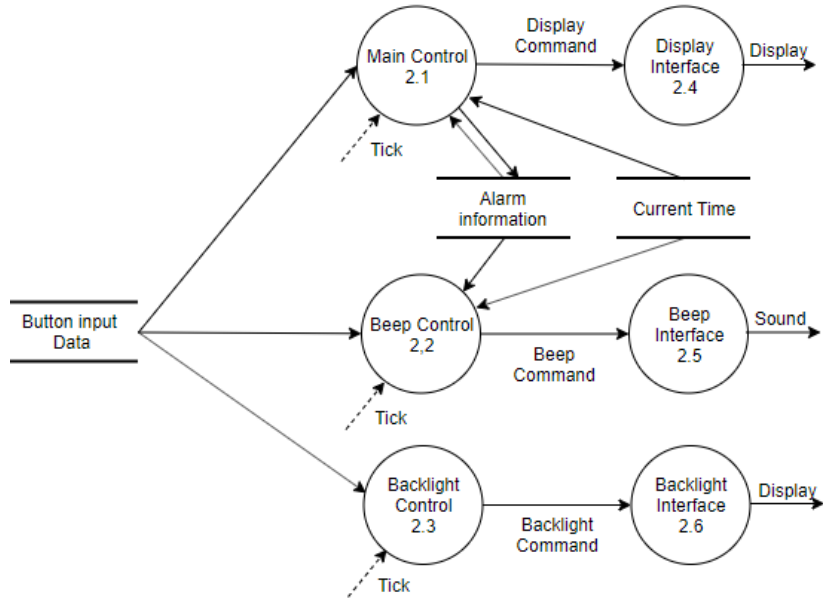
Revision

Input / Output Event	Description	Format / Type
Status information	<p>각 모드 (Time keeping, Alarm, Stopwatch) 저장하는 변수</p> <p>각 상태 (sec set, hour set, minute set, year set, month set, day set, alarm hour set, alarm min set, start, stop, lap time) 저장하는 변수</p> <p>알람 상태 (turn on, turn off) 저장하는 변수</p> <p>Backlight 상태 (turn yellow) 저장하는 변수</p>	Integer, periodic

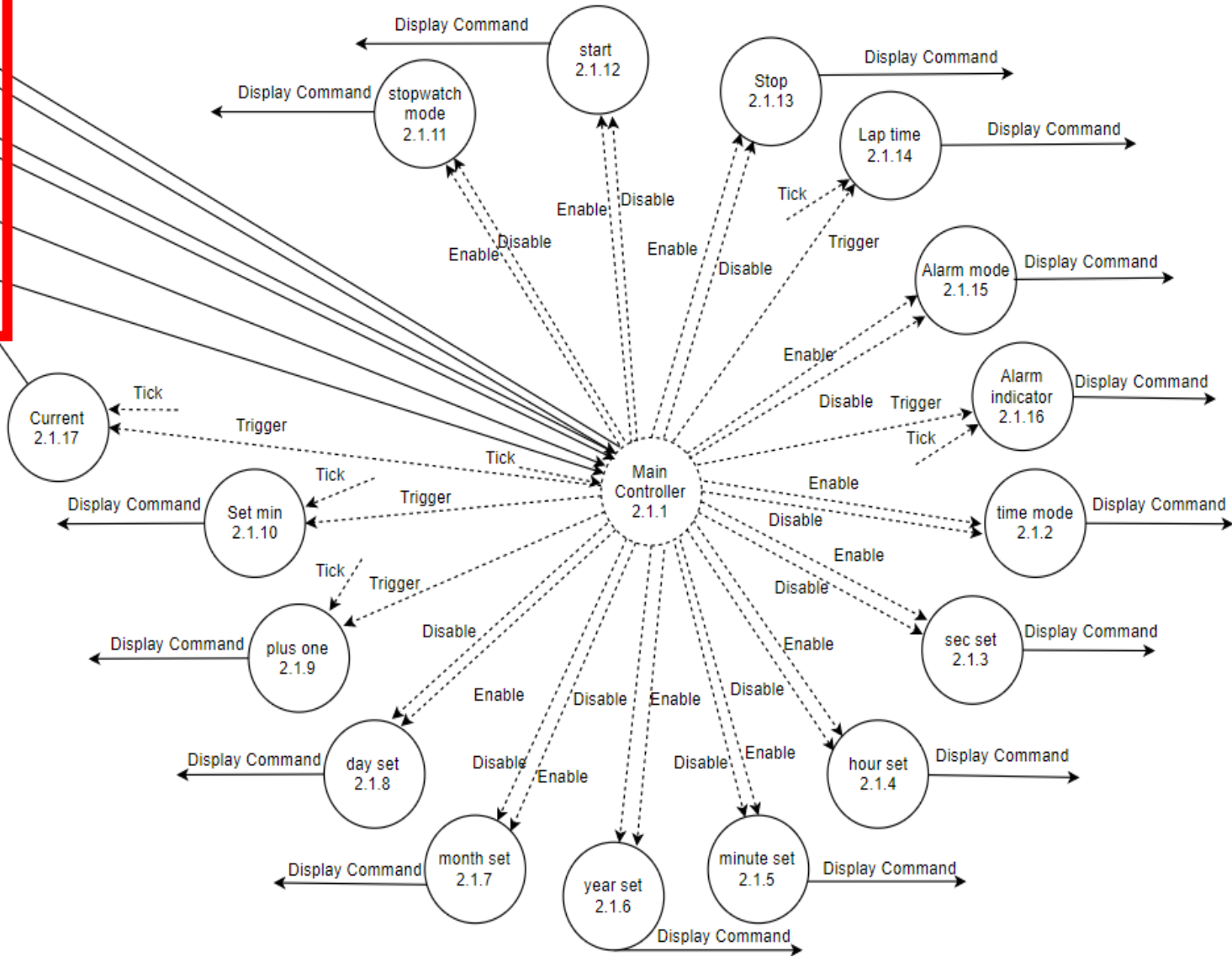
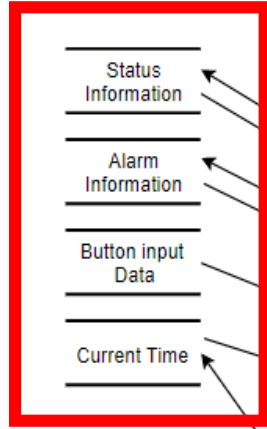
Revision



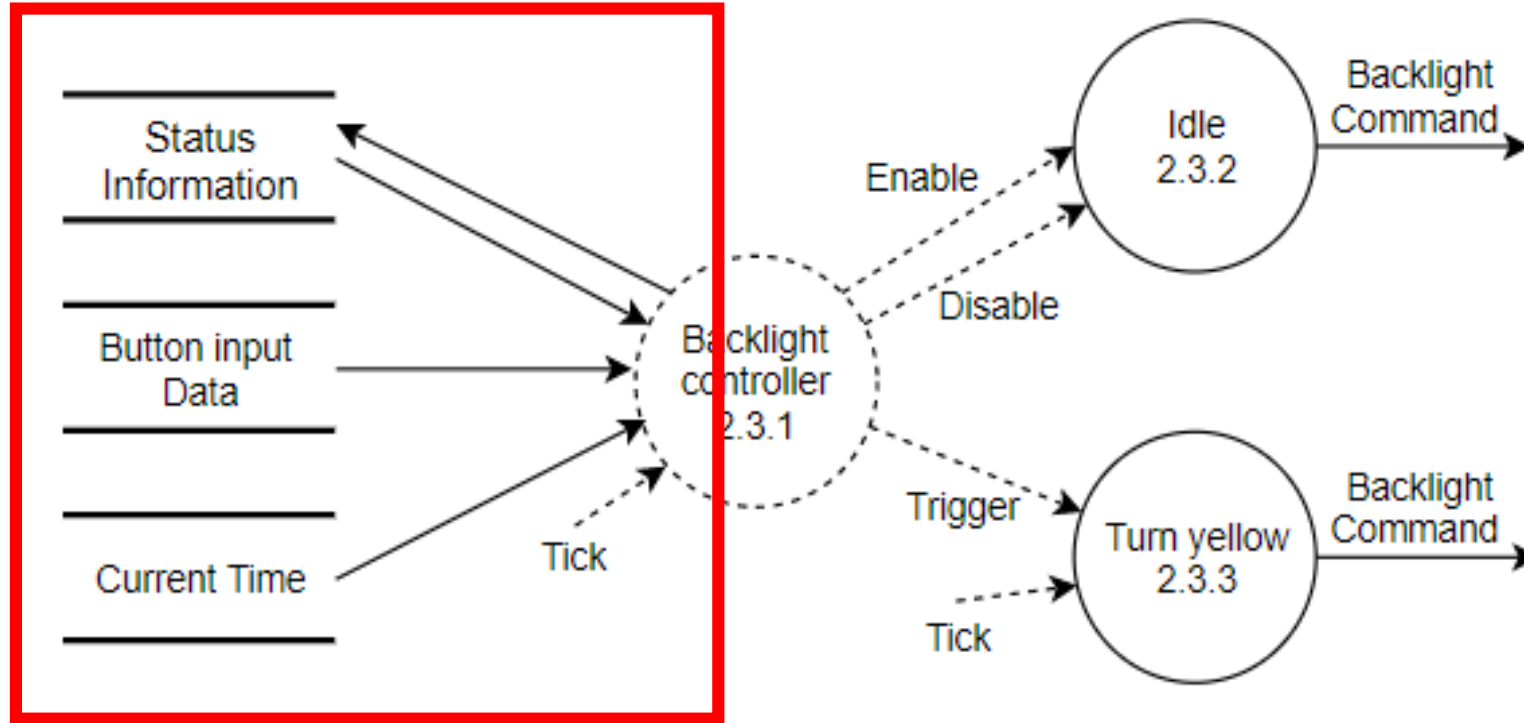
Revision



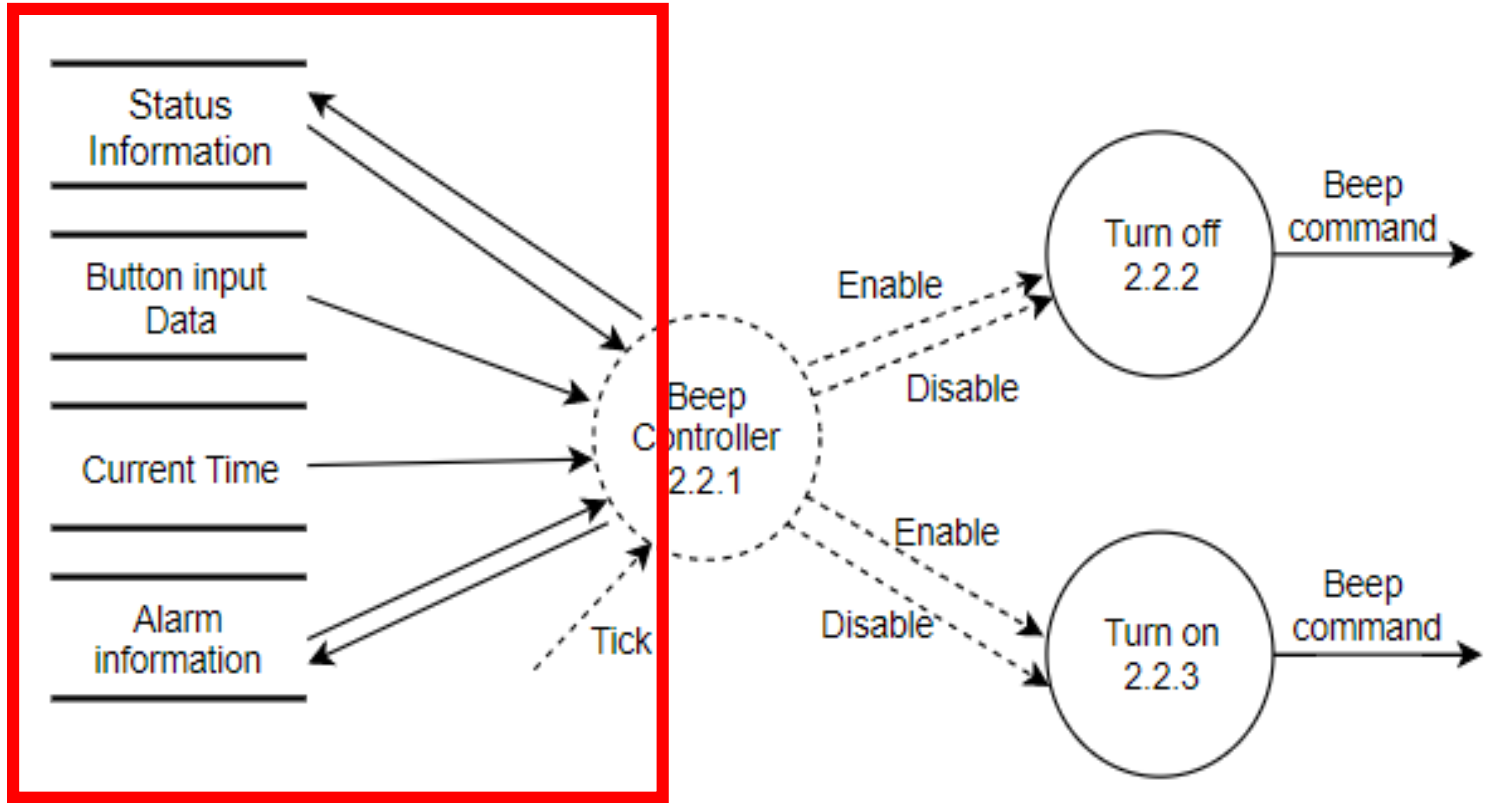
Revision



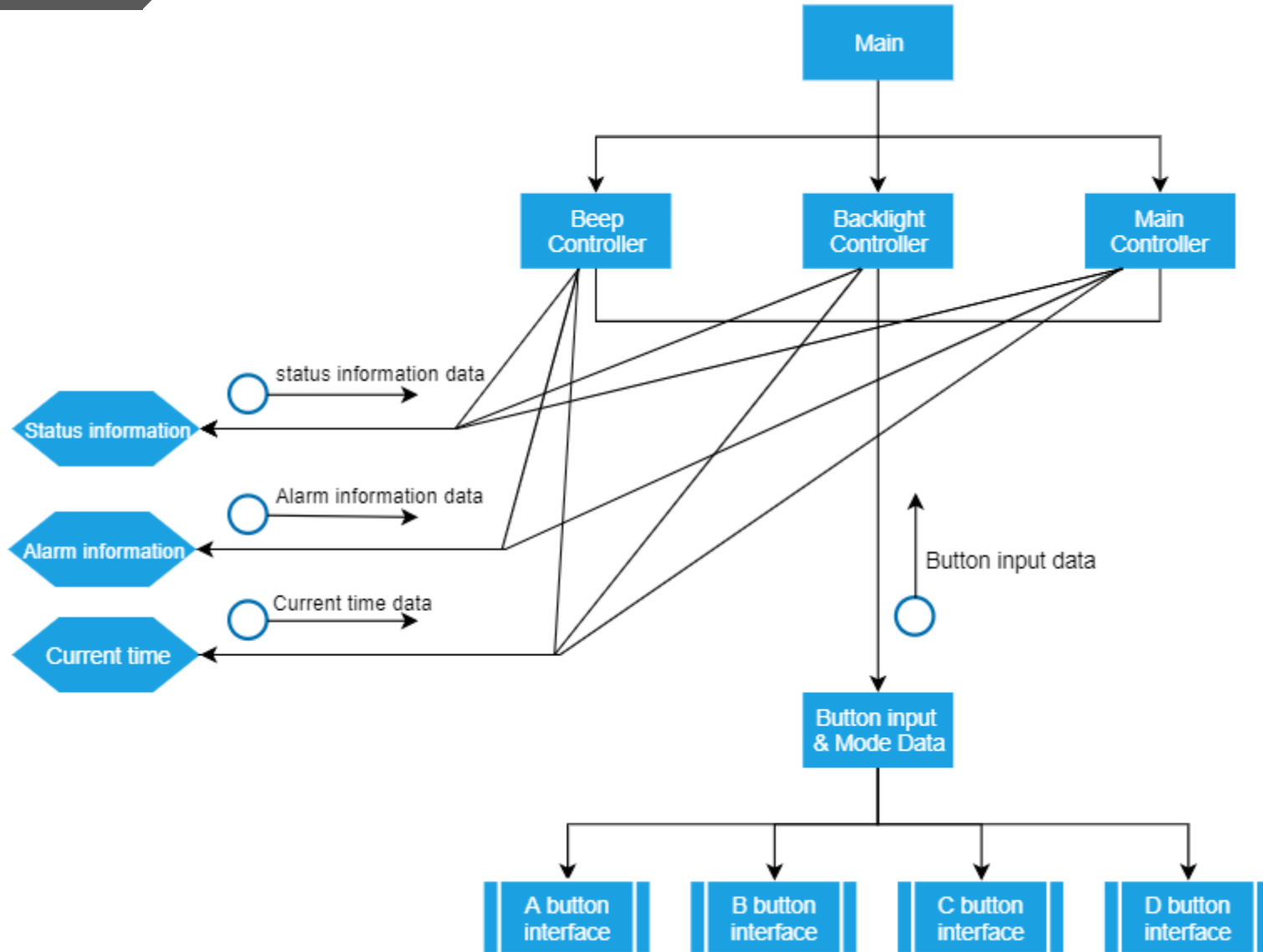
Revision



Revision



Revision



Implementation

Implementation

※ Main.c

```
#include "DWS.h"

void main() {

    // Data Storage 및 변수 초기화
    st_info = 0;
    al_info = TURN_OFF;
    mo_info = TIME_MODE;
    bl_info = 0;

    display_command = 0;

    // Alarm information 관련 변수
    alarm_information alarm_info;
    alarm_info.alarm_power = false;
    alarm_info.display_alarm_indicator = ' ';
    alarm_info.beep_time = 0;
    alarm_time = 1546268400;
    alarm_tm = *localtime(&alarm_time);

    // Stopwatch 관련 변수
    stop_min=0;
    stop_sec=0;
    stop_milisec=0;

    // Backlight 관련 변수
    backlight_time = 0;

    // 초기 시간 설정 바꾸기
    current_time = 1546268400;
    current_tm = *localtime(&current_time);

    timer_t timerID;
    createCurrent(&timerID,1,0);
}
```

Implementation

※ DWS.h

```
//mo_info
#define TIME_MODE 1
#define ALARM_MODE 8
#define STOPWATCH_MODE 11

//st_info
#define SEC_SET 2
#define HOUR_SET 3
#define MINUTE_SET 4
#define YEAR_SET 5
#define MONTH_SET 6
#define DAY_SET 7
#define AL_HOUR_SET 9
#define AL_MINUTE_SET 10
#define START 12
#define STOP 13
#define LAP_TIME 14

//al_info
#define TURN_ON 15
#define TURN_OFF 16

//bl_info
#define TURN_YELLOW 17
```

```
// Data storage
int st_info; //Status information
int al_info; //Alarm information
int mo_info; //mode information
int bl_info; //backlight information
```

Implementation

※ Main.c

```
#include "DWS.h"

void main() {

    // Data Storage 및 변수 초기화
    st_info = 0;
    al_info = TURN_OFF;
    mo_info = TIME_MODE;
    bl_info = 0;

    display_command = 0;

    // Alarm information 관련 변수
    alarm_information alarm_info;
    alarm_info.alarm_power = false;
    alarm_info.display_alarm_indicator = ' ';
    alarm_info.beep_time = 0;
    alarm_time = 1546268400;
    alarm_tm = *localtime(&alarm_time);

    // Stopwatch 관련 변수
    stop_min=0;
    stop_sec=0;
    stop_milisec=0;

    // Backlight 관련 변수
    backlight_time = 0;

    // 초기 시간 설정 바꾸기
    current_time = 1546268400;
    current_tm = *localtime(&current_time);

    timer_t timerID;
    createCurrent(&timerID,1,0);
```


Implementation

※ DWS.c

```
// Current 타이머를 만드는 함수
int createCurrent(timer_t *timerID, int sec, int msec){
    struct sigevent te;
    struct itimerspec its;
    struct sigaction sa;
    int sigNo=SIGRTMIN;

    sa.sa_flags=SA_SIGINFO;
    sa.sa_sigaction = current;
    sigemptyset(&sa.sa_mask);

    if(sigaction(sigNo, &sa, NULL)== -1 ){
        printf("sigaction error\n");
        return -1;
    }
    te.sigev_notify = SIGEV_SIGNAL;
    te.sigev_signo = sigNo;
    te.sigev_value.sival_ptr = timerID;
    timer_create(CLOCK_REALTIME, &te, timerID);

    its.it_interval.tv_sec = sec;
    its.it_interval.tv_nsec = msec * 1000000;
    its.it_value.tv_sec = sec;

    its.it_value.tv_nsec = msec * 1000000;
    timer_settime(*timerID, 0, &its, NULL);

    return 0;
}
```

```
// 월마다 날짜를 구하는 함수
int date(int year, int month)
{
    int mdays[12] = { 31,28,31,30,31,30,31,31,30,31,30,31 };
    if ((year % 4 == 0) && ((year % 100 != 0) || (year % 400 == 0)))//윤년 판단
    {
        mdays[1] = 29;//2월달의 날 수를 29로 설정
    }

    //month는 mdays의 인덱스로 사용해서 실제 출력할 때는 1을더해서 출력합니다.
    int day = mdays[month];
    return day;
}
```

```
// 현재 시간을 매초 업데이트하는 프로세스
static void current()
{
    current_tm.tm_sec++;

    if(current_tm.tm_sec==60){
        current_tm.tm_sec=0;
        current_tm.tm_min++;
    }
    else if(current_tm.tm_min==60){
        current_tm.tm_min=0;
        current_tm.tm_hour++;
    }
    else if(current_tm.tm_hour==24){
        current_tm.tm_hour=0;
        if(current_tm.tm_wday==6){
            current_tm.tm_wday=0;
        }else{
            current_tm.tm_mday++;
            current_tm.tm_wday++;
        }
    }
    else if(current_tm.tm_mday==date(current_tm.tm_year, current_tm.tm_mon)){
        current_tm.tm_mday=0;
        current_tm.tm_mon++;
    }
    else if(current_tm.tm_mon==12){
        current_tm.tm_mon=1;
        current_tm.tm_year++;
    }
    else if(current_tm.tm_year==2099){
        current_tm.tm_year=2019;
    }
}
```

Implementation - Beep Controller

※ Main.c

```
while (1) {  
  
    if(kbhit()==1){  
    }  
    // Beep Controller  
    if (al_info == TURN_OFF) turn_off();  
    if (al_info == TURN_ON) turn_on();  
  
    // Backlight Controller  
    if (btn == 'd' && bl_info == 0) idle();  
    if (bl_info == TURN_YELLOW) turn_yellow();  
  
    // Main Controller  
    if (mo_info == TIME_MODE)  
    {  
        if (st_info == 0) time_mode();  
        if (st_info == SEC_SET) sec_set();  
        if (st_info == HOUR_SET) hour_set();  
        if (st_info == MINUTE_SET) minute_set();  
        if (st_info == YEAR_SET) year_set();  
        if (st_info == MONTH_SET) month_set();  
        if (st_info == DAY_SET) day_set();  
    }  
    else if (mo_info == ALARM_MODE) {  
        if (st_info == 0) alarm_mode();  
        if (st_info == AL_HOUR_SET) hour_set();  
        if (st_info == AL_MINUTE_SET ) minute_set();  
    }  
    else if (mo_info == STOPWATCH_MODE){  
        if(st_info == 0) stopwatch_mode();  
        if(st_info == START) start();  
        if(st_info == LAP_TIME) lap_time();  
        if(st_info == STOP) stop();  
    }  
  
    display(display_command);  
}
```

Implementation

- 기타 함수

※ DWS.c

```
int getch()
{
    int c;
    struct termios oldattr, newattr;

    tcgetattr(STDIN_FILENO, &oldattr);
    newattr = oldattr;
    newattr.c_lflag &= ~(ICANON | ECHO);
    newattr.c_cc[VMIN] = 1;
    newattr.c_cc[VTIME] = 0;
    tcsetattr(STDIN_FILENO, TCSANOW, &newattr);
    c = getchar();
    tcsetattr(STDIN_FILENO, TCSANOW, &oldattr);
    return c;
}

int kbhit()
{
    struct termios oldt, newt;
    int oldf;

    tcgetattr(STDIN_FILENO, &oldt);
    newt = oldt;
    newt.c_lflag &= ~(ICANON | ECHO);
    tcsetattr(STDIN_FILENO, TCSANOW, &newt);
    oldf = fcntl(STDIN_FILENO, F_GETFL, 0);
    fcntl(STDIN_FILENO, F_SETFL, oldf | O_NONBLOCK);

    btn = getch();

    tcsetattr(STDIN_FILENO, TCSANOW, &oldt);
    fcntl(STDIN_FILENO, F_SETFL, oldf);

    if (btn != EOF){return 1;}

    return 0;
}

void gotoxy(int x, int y) {
    printf("\033[%d;%df", y, x);

    fflush(stdout);
}
```

Implementation - Beep Controller

※ Main.c

```
while (1) {  
  
    if(kbhit()==1){  
  
        }  
        // Beep Controller  
        if (al_info == TURN_OFF) turn_off();  
        if (al_info == TURN_ON) turn_on();  
  
        // Backlight Controller  
        if (btn == 'd' && bl_info == 0) idle();  
        if (bl_info == TURN_YELLOW) turn_yellow();  
  
        // Main Controller  
        if (mo_info == TIME_MODE)  
        {  
            if (st_info == 0) time_mode();  
            if (st_info == SEC_SET) sec_set();  
            if (st_info == HOUR_SET) hour_set();  
            if (st_info == MINUTE_SET) minute_set();  
            if (st_info == YEAR_SET) year_set();  
            if (st_info == MONTH_SET) month_set();  
            if (st_info == DAY_SET) day_set();  
  
        }  
        else if (mo_info == ALARM_MODE) {  
  
            if (st_info == 0) alarm_mode();  
            if (st_info == AL_HOUR_SET) hour_set();  
            if (st_info == AL_MINUTE_SET ) minute_set();  
  
        }  
        else if (mo_info == STOPWATCH_MODE){  
  
            if(st_info == 0) stopwatch_mode();  
            if(st_info == START) start();  
            if(st_info == LAP_TIME) lap_time();  
            if(st_info == STOP) stop();  
  
        }  
  
        display(display_command);  
  
    }  
}
```

Implementation - Beep Controller

※ DWS.c

```
void turn_off()
{
    if((alarm_info.alarm_power == true) && (alarm_tm.tm_hour == current_tm.tm_hour) && (alarm_tm.tm_min == current_tm.tm_min)) al_info = TURN_ON;
}

void turn_on()
{
    if (btn == 'a' || btn == 'b' || btn == 'c' || btn == 'd' ) {
        alarm_tm = *localtime(&current_time);
        alarm_info.beep_time=0;
        alarm_info.alarm_power=false;
        alarm_info.display_alarm_indicator = ' ';
        al_info = TURN_OFF;
    }

    //display
    if(alarm_info.beep_time<5){
        alarm_info.beep_time++;
        printf("\a");
        sleep(1);
    }

    if(alarm_info.beep_time==5) {
        alarm_tm = *localtime(&current_time);
        alarm_info.beep_time=0;
        alarm_info.alarm_power=false;
        alarm_info.display_alarm_indicator = ' ';
        al_info = TURN_OFF;
    }
    btn = 0;
}
```

Implementation - Backlight Controller

※ Main.c

```
while (1) {

    if(kbhit()==1){

    }
    // Beep Controller
    if (al_info == TURN_OFF) turn_off();
    if (al_info == TURN_ON) turn_on();

    // Backlight Controller
    if (btn == 'd' && bl_info == 0) idle();
    if (bl_info == TURN_YELLOW) turn_yellow();

    // Main Controller
    if (mo_info == TIME_MODE)
    {
        if (st_info == 0) time_mode();
        if (st_info == SEC_SET) sec_set();
        if (st_info == HOUR_SET) hour_set();
        if (st_info == MINUTE_SET) minute_set();
        if (st_info == YEAR_SET) year_set();
        if (st_info == MONTH_SET) month_set();
        if (st_info == DAY_SET) day_set();

    }
    else if (mo_info == ALARM_MODE) {

        if (st_info == 0) alarm_mode();
        if (st_info == AL_HOUR_SET) hour_set();
        if (st_info == AL_MINUTE_SET ) minute_set();

    }
    else if (mo_info == STOPWATCH_MODE){

        if(st_info == 0) stopwatch_mode();
        if(st_info == START) start();
        if(st_info == LAP_TIME) lap_time();
        if(st_info == STOP) stop();

    }

    display(display_command);

}
```

Implementation - Backlight Controller

※ DWS.c

```
void idle() {  
    if (btn == 'd') {  
        bl_info=TURN_YELLOW;  
    }  
    btn = 0;  
}  
  
void turn_yellow() {  
    display(PRINT_BACKLIGHT);  
}
```

Implementation - Main Controller

✂ Main.c

```
while (1) {

    if(kbhit()==1){

        }
        // Beep Controller
        if (al_info == TURN_OFF) turn_off();
        if (al_info == TURN_ON) turn_on();

        // Backlight Controller
        if (btn == 'd' && bl_info == 0) idle();
        if (bl_info == TURN_YELLOW) turn_yellow();

        // Main Controller
        if (mo_info == TIME_MODE)
        {
            if (st_info == 0) time_mode();
            if (st_info == SEC_SET) sec_set();
            if (st_info == HOUR_SET) hour_set();
            if (st_info == MINUTE_SET) minute_set();
            if (st_info == YEAR_SET) year_set();
            if (st_info == MONTH_SET) month_set();
            if (st_info == DAY_SET) day_set();
        }

        else if (mo_info == ALARM_MODE) {

            if (st_info == 0) alarm_mode();
            if (st_info == AL_HOUR_SET) hour_set();
            if (st_info == AL_MINUTE_SET ) minute_set();

        }

        else if (mo_info == STOPWATCH_MODE){

            if(st_info == 0) stopwatch_mode();
            if(st_info == START) start();
            if(st_info == LAP_TIME) lap_time();
            if(st_info == STOP) stop();

        }

        display(display_command);

    }
}
```


Implementation - Main Controller

※ DWS.c

```
// 현재 시간을 보여주는 프로세스
void time_mode()
{
    if (btn == 'a') {
        st_info = SEC_SET;
    }
    if (btn == 'c') {
        mo_info = ALARM_MODE;
    }
    display_command = PRINT_TIME_MODE;
    btn = 0;
}
```

Implementation - Main Controller

※ DWS.c

```
void sec_set()
{
    void hour_set()
    {
        //TIME_MODE
        void minute_set()
        {
            void year_set()
            {
                if (btn == 'a') {
                    st_info = 0;
                }
                if (btn == 'b') {
                    if (current_tm.tm_year == 199) set_min(st_info);
                    else plus_one(st_info);
                }
                if (btn == 'c') {
                    st_info = MONTH_SET;
                }
            }
            void day_set()
            {
                if (btn == 'a') {
                    st_info = 0;
                }
                if (btn == 'b') {
                    //달 별로 일 수가 다르기 때문에 date함수를 구현해 맞는 max_day를 넣어주었습니다.
                    if (current_tm.tm_mday == date(current_tm.tm_year, current_tm.tm_mon)) set_min(st_info);
                    else plus_one(st_info);
                }
                if (btn == 'c') {
                    st_info = SEC_SET;
                }
            }
            //display
            display_command = PRINT_DAY_SET;
            btn = 0;
            printf("%일\n");
        }
        void month_set()
        {
            if (btn == 'a') {
                st_info = 0;
            }
            if (btn == 'b') {
                plus_one(st_info);
            }
            if (btn == 'c') {
                st_info = MONTH_SET;
            }
            //display
            display_command = PRINT_MONTH_SET;
            btn = 0;
            printf("%월\n");
        }
        void year_set()
        {
            if (btn == 'a') {
                st_info = 0;
            }
            if (btn == 'b') {
                plus_one(st_info);
            }
            if (btn == 'c') {
                st_info = YEAR_SET;
            }
            //display
            display_command = PRINT_YEAR_SET;
            btn = 0;
            printf("%년\n");
        }
        //display
        display_command = PRINT_HOUR_SET;
        btn = 0;
        printf("%시\n");
    }
    //display
    display_command = PRINT_MINUTE_SET;
    btn = 0;
    printf("%분\n");
}
//display
display_command = PRINT_SECONDS_SET;
btn = 0;
printf("%초\n");
}
```

Implementation - Main Controller

※ DWS.c

```
void plus_one(int st_info)
{
    switch (st_info)
    {
        case SEC_SET:
            current_tm.tm_sec++; break;
        case HOUR_SET:
            current_tm.tm_hour++; break;
        case MINUTE_SET:
            current_tm.tm_min++; break;
        case YEAR_SET:
            current_tm.tm_year++; break;
        case MONTH_SET:
            current_tm.tm_mon++; break;
        case DAY_SET:
            current_tm.tm_mday++;
            if(current_tm.tm_wday==6) c
            else current_tm.tm_wday++;
            break;
        case AL_HOUR_SET:
            alarm_tm.tm_hour++; break;
        case AL_MINUTE_SET:
            alarm_tm.tm_min++; break;
        default: break;
    }
}
```

```
void set_min(int st_info)
{
    switch (st_info)
    {
        case SEC_SET:
            current_tm.tm_sec = 0;
            break;
        case HOUR_SET:
            current_tm.tm_hour = 0;
            break;
        case MINUTE_SET:
            current_tm.tm_min = 0;
            break;
        case YEAR_SET:
            current_tm.tm_year = 2019;
            break;
        case MONTH_SET:
            current_tm.tm_mon = 1;
            break;
        case DAY_SET:
            current_tm.tm_mday = 1;
            break;
        case AL_HOUR_SET:
            alarm_tm.tm_hour = 0;
            break;
        case AL_MINUTE_SET:
            alarm_tm.tm_min = 0;
            break;
        default: break;
    }
}
```

Implementation - Main Controller

※ Main.c

```
while (1) {

    if(kbhit()==1){

        }
        // Beep Controller
        if (al_info == TURN_OFF) turn_off();
        if (al_info == TURN_ON) turn_on();

        // Backlight Controller
        if (btn == 'd' && bl_info == 0) idle();
        if (bl_info == TURN_YELLOW) turn_yellow();

        // Main Controller
        if(mo_info==TIME_MODE)
        {
            if (st_info == 0) time_mode();
            if (st_info == SEC_SET) sec_set();
            if (st_info == HOUR_SET) hour_set();
            if (st_info == MINUTE_SET) minute_set();
            if (st_info == YEAR_SET) year_set();
            if (st_info == MONTH_SET) month_set();
            if (st_info == DAY_SET) day_set();

        }
        else if (mo_info == ALARM_MODE) {

            if (st_info == 0) alarm_mode();
            if (st_info == AL_HOUR_SET) hour_set();
            if (st_info == AL_MINUTE_SET ) minute_set();

        }
        else if (mo_info == STOPWATCH_MODE){

            if(st_info == 0) stopwatch_mode();
            if(st_info == START) start();
            if(st_info == LAP_TIME) lap_time();
            if(st_info == STOP) stop();

        }

        display(display_command);

    }
}
```

Implementation - Main Controller

※ DWS.h / DWS.c

```
typedef struct _alarm_information{
    bool alarm_power; // 알람 켜짐 여부
    char display_alarm_indicator; // Alarm indicator on/off 여부
    int beep_time;
}alarm_information;
```

```
void alarm_mode()
{
    if (btn == 'a') {
        st_info =AL_HOUR_SET;
    }
    if (btn == 'b') {
        alarm_indicator();
    }
    if (btn == 'c') {
        mo_info = STOPWATCH_MODE;
    }

    //display
    display_command = PRINT_ALARM_MODE;
    btn = 0;
}

void alarm_indicator()
{
    if(alarm_info.alarm_power == true){
        alarm_info.alarm_power=false;
        alarm_info.display_alarm_indicator = ' ';
    }else{
        alarm_info.alarm_power=true;
        alarm_info.display_alarm_indicator = 'I';
    }
}
```

Implementation - Main Controller

※ Main.c

```
while (1) {  
  
    if(kbhit()==1){  
  
        }  
        // Beep Controller  
        if (al_info == TURN_OFF) turn_off();  
        if (al_info == TURN_ON) turn_on();  
  
        // Backlight Controller  
        if (btn == 'd' && bl_info == 0) idle();  
        if (bl_info == TURN_YELLOW) turn_yellow();  
  
        // Main Controller  
        if (mo_info == TIME_MODE)  
        {  
            if (st_info == 0) time_mode();  
            if (st_info == SEC_SET) sec_set();  
            if (st_info == HOUR_SET) hour_set();  
            if (st_info == MINUTE_SET) minute_set();  
            if (st_info == YEAR_SET) year_set();  
            if (st_info == MONTH_SET) month_set();  
            if (st_info == DAY_SET) day_set();  
  
        }  
        else if (mo_info == ALARM_MODE) {  
  
            if (st_info == 0) alarm_mode();  
            if (st_info == AL_HOUR_SET) hour_set();  
            if (st_info == AL_MINUTE_SET ) minute_set();  
  
        }  
        else if (mo_info == STOPWATCH_MODE){  
  
            if(st_info == 0) stopwatch_mode();  
            if(st_info == START) start();  
            if(st_info == LAP_TIME) lap_time();  
            if(st_info == STOP) stop();  
  
        }  
  
        display(display_command);  
  
    }  
}
```

Implementation - Main Controller

※ DWS.c

```
void stopwatch_mode()
{
    if (btn == 'b') {
        st_info=START;
    }
    if (btn == 'c') {
        mo_info = TIME_MODE;
    }

    // Stopwatch에 필요한 기준값 초기화
    gettimeofday(&st_start, NULL);
    stop_milisec = 0;
    stop_sec = 0;
    stop_min = 0;
    display_command = PRINT_STOPWATCH_MODE;
    btn = 0;
}

void start()
{
    if (btn == 'a') {
        st_info=LAP_TIME;
    }
    if (btn == 'b') {
        st_info=STOP;
    }

    //display
    gettimeofday(&st_tv, NULL);
    st_tv.tv_sec -= st_start.tv_sec;
    st_stop.tv_sec = st_tv.tv_sec;
    st_tm = *localtime(&st_tv.tv_sec);
    stop_milisec = st_tv.tv_usec/10000;
    stop_sec = st_tm.tm_sec;
    stop_min = st_tm.tm_min;
    display_command = PRINT_START;

    btn = 0;
}
```

```
void lap_time()
{
    if (btn == 'a') {
        lap_time();
    }
    if (btn == 'b') {
        st_info=START;
    }

    //display
    display_command = PRINT_LAP_TIME;
    btn = 0;
}

void stop()
{
    if (btn == 'a') {
        st_info=0;
    }
    if (btn == 'b') {
        st_info=START;
        gettimeofday(&st_tv, NULL);
        st_tv.tv_sec -= st_start.tv_sec;
        st_start.tv_sec += (st_tv.tv_sec-st_stop.tv_sec);
    }

    //display
    display_command = PRINT_STOP;
    btn = 0;
}
```

Implementation

※ Main.c

```
while (1) {  
  
    if(kbhit()==1){  
  
        }  
        // Beep Controller  
        if (al_info == TURN_OFF) turn_off();  
        if (al_info == TURN_ON) turn_on();  
  
        // Backlight Controller  
        if (btn == 'd' && bl_info == 0) idle();  
        if (bl_info == TURN_YELLOW) turn_yellow();  
  
        // Main Controller  
        if (mo_info==TIME_MODE)  
        {  
            if (st_info == 0) time_mode();  
            if (st_info == SEC_SET) sec_set();  
            if (st_info == HOUR_SET) hour_set();  
            if (st_info == MINUTE_SET) minute_set();  
            if (st_info == YEAR_SET) year_set();  
            if (st_info == MONTH_SET) month_set();  
            if (st_info == DAY_SET) day_set();  
  
        }  
        else if (mo_info == ALARM_MODE) {  
  
            if (st_info == 0) alarm_mode();  
            if (st_info == AL_HOUR_SET) hour_set();  
            if (st_info == AL_MINUTE_SET ) minute_set();  
  
        }  
        else if (mo_info == STOPWATCH_MODE){  
  
            if(st_info == 0) stopwatch_mode();  
            if(st_info == START) start();  
            if(st_info == LAP_TIME) lap_time();  
            if(st_info == STOP) stop();  
  
        }  
  
        display(display_command);  
  
    }  
}
```


Implementation

※ DWS.h / DWS.c

```
//display command
#define PRINT_TIME_MODE 20
#define PRINT_SEC_SET 21
#define PRINT_HOUR_SET 22
#define PRINT_MINUTE_SET 23
#define PRINT_YEAR_SET 24
#define PRINT_MONTH_SET 25
#define PRINT_DAY_SET 26
#define PRINT_ALARM_MODE 27
#define PRINT_AL_HOUR_SET 28
#define PRINT_AL_MINUTE_SET 29
#define PRINT_STOPWATCH_MODE 30
#define PRINT_START 31
#define PRINT_STOP 32
#define PRINT_LAP_TIME 33
#define PRINT_BACKLIGHT 34
```

```
void display(int display_command)
{
    //if(bl_info==TURN_YELLOW) printf("%c[1;33m", 27);
    switch(display_command) {

        case PRINT_BACKLIGHT:
            if(backlight_time<5) {
                printf("%c[1;33m", 27);
                backlight_time++;
            }
            else {
                backlight_time = 0;
                bl_info=0;
                printf("%c[0m", 27);
            }
            usleep(100000);
            break;

        case PRINT_TIME_MODE:
            system("clear");
            gotoxy(70,20);
            printf("%s %c %d-%d-%d %d:%d:%d\n", week_day[current_tm.tm_wday], alarm_info.display_alarm_ind
            icator, current_tm.tm_year + 1900, current_tm.tm_mon + 1, current_tm.tm_mday, current_tm.tm_hour, current_tm.tm_min, current_tm.tm_sec);
            usleep(100000);
            break;

        case PRINT_SEC_SET:
            system("clear");
            gotoxy(70,20);
            printf("%s %c %d-%d-%d %d:%d:%c[4m%d%c[0m\n", week_day[current_tm.tm_wday], alarm_info.display_alarm_ind
            icator, current_tm.tm_year + 1900, current_tm.tm_mon + 1, current_tm.tm_mday, current_tm.tm_hour, current_tm.tm_min, 27, current
            _tm.tm_sec, 27);
            usleep(100000);
            break;

        case PRINT_HOUR_SET:
            system("clear");
            gotoxy(70,20);
            printf("%s %c %d-%d-%d %c[4m%d%c[0m:%d:%d\n", week_day[current_tm.tm_wday], alarm_info.display_alarm_ind
            icator, current_tm.tm_year + 1900, current_tm.tm_mon + 1, current_tm.tm_mday, 27, current_tm.tm_hour, 27, current_tm.tm_min, cur
            rent_tm.tm_sec);
            sleep(1);
            break;
    }
}
```

DEMO

Q & A