

Team Presentation #4

Implementation

Team 1

201211338	김주호
220111352	박소은
201211346	박선민
201211371	이유민

INDEX

SRA & SDS

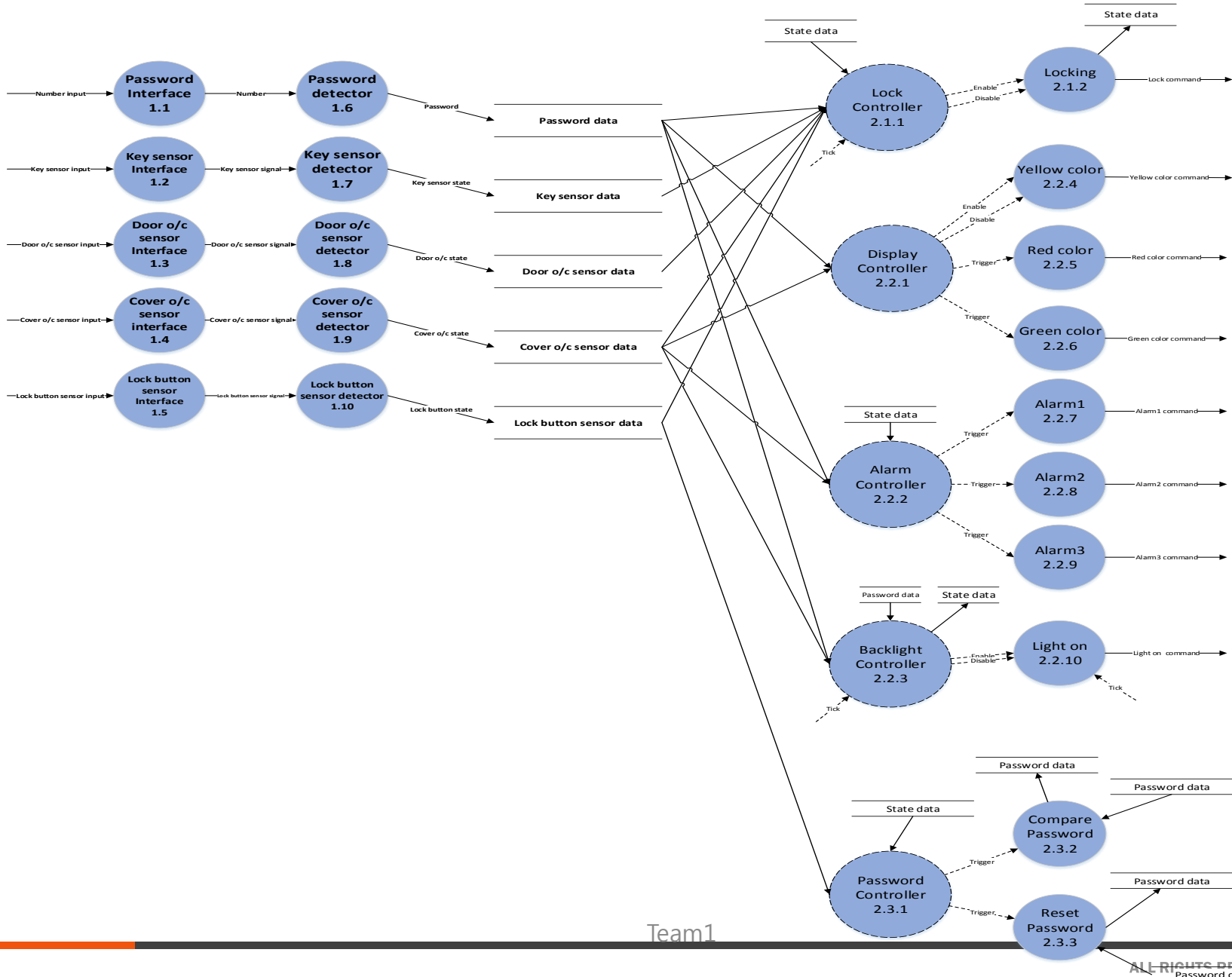
Development Environment

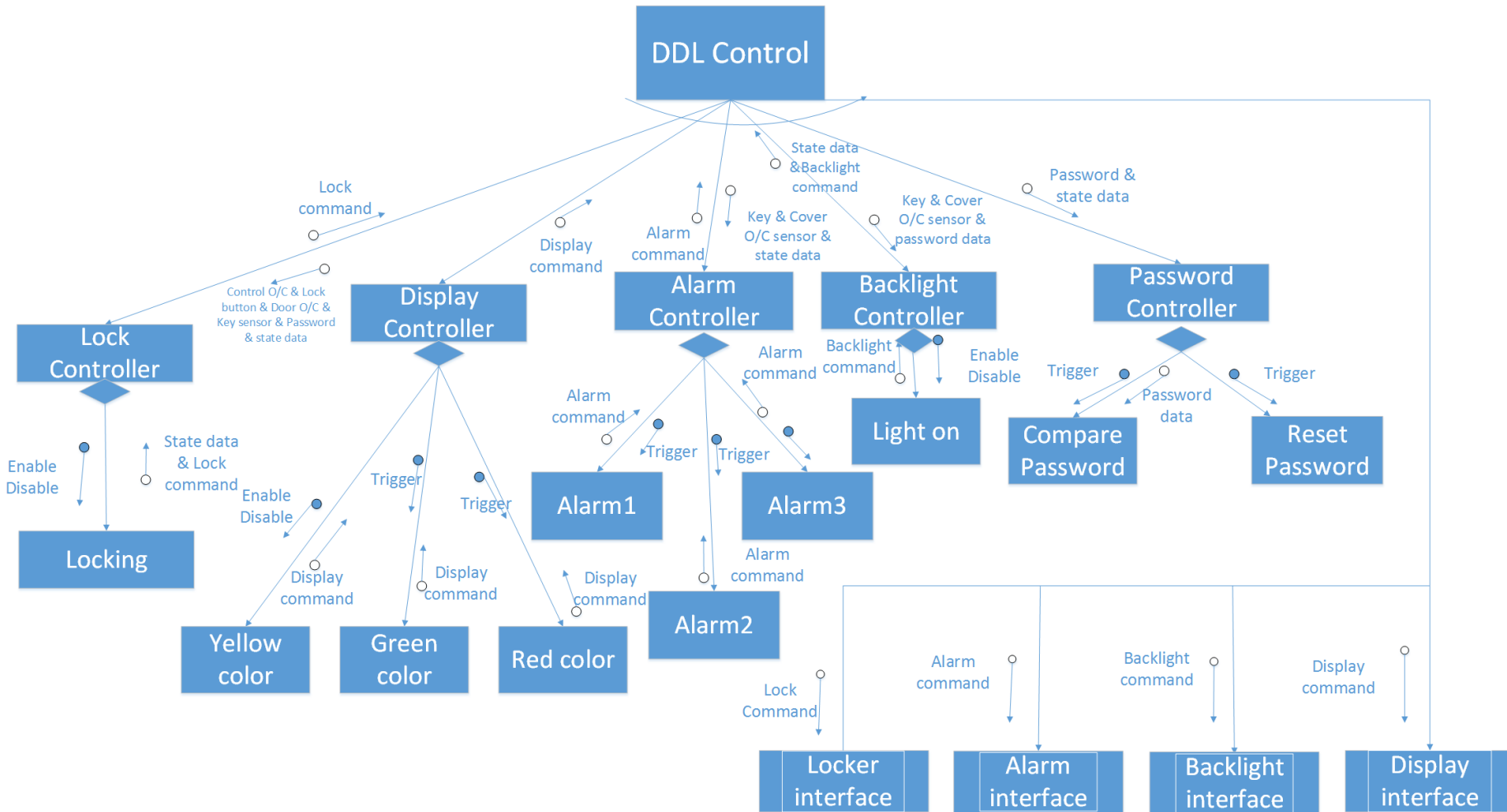
Source Code

Demo

1

SRA & SDS





2

Development Environment

- Eclipse IDE for C/C++ Developers
- Windows 7 Pro
- MinGW 4.7.0

3

Source Code


```

typedef struct state_data{
    int L; //L=Lock Mode
    int BO; //Backlight Off
    int AC; //Auto Close
    int INPUT_END; //비밀번호 네 자리를 모두 입력했는지
    int C_flag; //Cover가 막 열린 순간인지
}STATE_DATA;

typedef struct data{
    int D; //Door open&close
    int C; //Cover open&close
    int LB; //Lock Button
    int K; //key
    int N; //password button이 눌렸는지 안눌렸는지
    int P; //password가 맞는지 틀린지
}DATA;

typedef struct password{
    char* password; //설정된 비밀번호
    char* input_password; //사용자가 입력하는 비밀번호
    int count ; // 몇번째 글자에다 넣어야하는지
}PASSWORD_DATA;

typedef enum _keyState{
    RIGHT = 1, //key값이 맞을때,
    WAIT , //대기상태일때
    WRONG
}KEY_STATE; // the color of backlight for each state

typedef enum _sound{
    ALARM1 = 1,
    ALARM2 ,
    ALARM3
}SOUND; // the alram sound for each state

```

```

int init(STATE_DATA* state,DATA* data ,PASSWORD_DATA* password) ; //데이터 초기화
int gotoxy(int x,int y); // 좌표 옮기는 함수
void TC(int font); //텍스트 칼라 바꾸는 함수
void Backlight(); // 백라이트 부분 디스플레이
void OutLiner(int mode); // 바깥라인 부분 디스플레이
char Cover(); // 커버 부분 디스플레이
void CoverOpen(char color); //key값에 따른 디스플레이 컨트롤러 부분

int play(STATE_DATA* state, DATA* data, PASSWORD_DATA* password_data); //전체 컨트롤러
void get_Alarm(int mode); // 알람 사운드 가져오기
void init_sound(); // 사운드 초기화
void password_setting(PASSWORD_DATA* password); //패스워드 셋팅
void input(STATE_DATA* state, DATA* data, PASSWORD_DATA* password); // input 받아오기
void lock_controller(DATA* data, STATE_DATA* state,unsigned int tick); // Lock controller
void locking(int condition, STATE_DATA* state); //상태에 따른 lock
void password_controller(DATA* data, STATE_DATA* state, unsigned int tick, PASSWORD_DATA* password); //패스워드 컨트롤러
void compare_password(PASSWORD_DATA* password,STATE_DATA* state ,DATA* data);
void reset_password(PASSWORD_DATA* password, STATE_DATA* state, DATA* data);
void display_controller(DATA* data);
void backlight_controller(STATE_DATA* state, DATA* data);
void alarm_controller(STATE_DATA* state, DATA* data);
void cover_open(); //덮개가 닫혀있을때 디스플레이
void cover_close(); // 덮개가 닫혀있을때 디스플레이
void ui(STATE_DATA*,DATA*); //전체 유아미

```

Input Function

```

if(input == 'l')(data->LB = TRUE;)
else if(input == 'c')
{
    if(data->C == FALSE){
        data->C = TRUE;
        state->C_flag = 0;
    }
    else
    {
        data->C = FALSE ;
        //state->C_flag = 1;
    }
}
}
else if(input == 'k'){data->K = RIGHT;} // 어디로 가야할까 : 해결
else if (input == 'j'){data->K = WRONG;} // 틀린 키값일때의 상태를 보이기 위해서
else if('0' <= input <= '9' && state->BO ==FALSE && data->C == TRUE)
{
    data ->N = TRUE ; //
    if(password->count<4)
    {
        password->input_password[password->count] = input;
        (password->count)++ ;

        if(password->count == 3)
        {
            state->INPUT_END = TRUE;
        }
    }
    //else{password->count = 0 ;}

    //if(password->count==4){ state->INPUT_END = TRUE;}
}
}

```

Lock Function

```
void lock_controller(DATA* data, STATE_DATA* state, unsigned int tick) //수정필요
```

```
{
    if(state->L==FALSE && data->D == TRUE){ //0값에 대한 설명은?
        if(l_flag == 1){
            l_tick = tick;
            l_flag = 0 ;
        } // 차이점을 알아보기 위해서 //
        if(tick - l_tick>=3)
        {
            state->AC = TRUE;
            l_flag = 1 ;
        }
    }

    /*if(state->L==TRUE)
    {
        tick = 0;
    }*/

    if((state->L==TRUE&&data->LB==TRUE)|| (data->C==TRUE&&state->L==TRUE&&data->K==RIGHT)|| (data->C==TRUE&&state->L==TRUE&&data->P==TRUE))
    {
        locking(DISABLE, state);
    }
    else if((state->L==FALSE&&data->LB==TRUE&&data->D==FALSE)|| (data->D==FALSE&&state->L==TRUE&&state->AC==TRUE))
    {
        locking(ENABLE, state);
    }
}
```

```
void locking(int condition, STATE_DATA* state) //
{
    if(condition==ENABLE){state->L = TRUE;}
    else if(condition==DISABLE){state->L = FALSE;}
}
```

Password Function

```

void password_controller(DATA* data, STATE_DATA* state, PASSWORD_DATA* password)
{
    if(state->INPUT_END==TRUE&&data->C==FALSE)
    {
        compare_password(password,state,data);
    }
    else if(state->BO==TRUE)
    {
        reset_password(password,state,data);
    }
    else if(state->INPUT_END == FALSE && data->C ==FALSE)
        reset_password(password,state,data);
}

void compare_password(PASSWORD_DATA* password,STATE_DATA* state ,DATA* data)
{
    if(strcmp(password->input_password,password->password)==TRUE)
    {
        data->P = TRUE;
        reset_password(password,state,data); // P값이 의미가 없을듯 바로 바로 바뀌면
    }
    else if(strcmp(password->input_password,password->password)==FALSE)
    {
        reset_password(password,state,data);
    }
}

void reset_password(PASSWORD_DATA* password, STATE_DATA* state, DATA* data)
{
    password->input_password = (char*)calloc(5,sizeof(char));
    state->INPUT_END = FALSE;
    data->P = FALSE;
}

```

Display Function

```

void display_controller(DATA* data)
{
    if(data->C==FALSE&&data->K==WRONG)
    {
        CoverLamp('r');
        Sleep(100);
        CoverLamp('y');
        data ->K = WAIT ;
    }

    else if(data->C==FALSE&&data->K==RIGH
    {
        CoverLamp('g');
        Sleep(100);
        CoverLamp('y');
        data -> K = WAIT ;
    }
}

```

```

void CoverLamp(char color) //도어락 램프 - 색상 신호를 받아 램프의 색을 바꾼다
{
    if(color=='y') //yellow
        TC(FOREGROUND_INTENSITY | FOREGROUND_RED | FOREGROUND_GREEN);
    else if(color=='g') //green
        TC(FOREGROUND_INTENSITY | FOREGROUND_GREEN);
    else //red
        TC(FOREGROUND_INTENSITY | FOREGROUND_RED);

    gotoxy(39,23);
    printf("■ ■ ■ ■ ");
}

```

BackLight Function

```
void backlight_controller(STATE_DATA* state, DATA* data, unsigned int tick)
{
    if(state->C_flag == 0)
    {
        b_tick = tick ; //커버가 열릴때 틱 저장
        state->C_flag = 1 ;
        state->BO = FALSE ;
    }
    if(tick-b_tick>=10){ //틱 차이 구하기
        state->BO=TRUE;
        state->C_flag = 0 ;
    } //틱구현?

    if(state->BO==TRUE||data->C==FALSE)
    {
        OutLiner(DISABLE); // 백라이트 끄기
    }

    else if(data->C==TRUE&&(data->N==TRUE||state->C_flag==0))
    {
        OutLiner(ENABLE); // 백라이트 켜기

        if(state->C_flag==0){state->C_flag=1;}
        else{data->N=FALSE;} //????
    }
}
```

4

Demo

4

Conclusion



Q&A



THANK YOU