

추억의 게임 모음집

2팀

목차

프로젝트 개발 이유

프로젝트 기능

활용 디바이스 종류 및 목적

프로젝트 시나리오

시스템 구성도

프로젝트 개발 이유

과거 추억의 회상

엔터테인먼트
부가기능

임베디드
시스템에 대한
이해 증진

프로젝트 기능

A. 게임을 최대 4개 까지 설치하여 고르고 실행할 수 있는 메뉴 기능.

- 설치하기 위해서는, 소스 코드에 메뉴에 맞는 슬롯에 코드를 객체화 하여 삽입 혹은 연결만 하면 끝.
- 설치된 게임을 메뉴에서 선택하여 실행.
- 현재 설치된 게임은 손수 구현한: 갤러그.

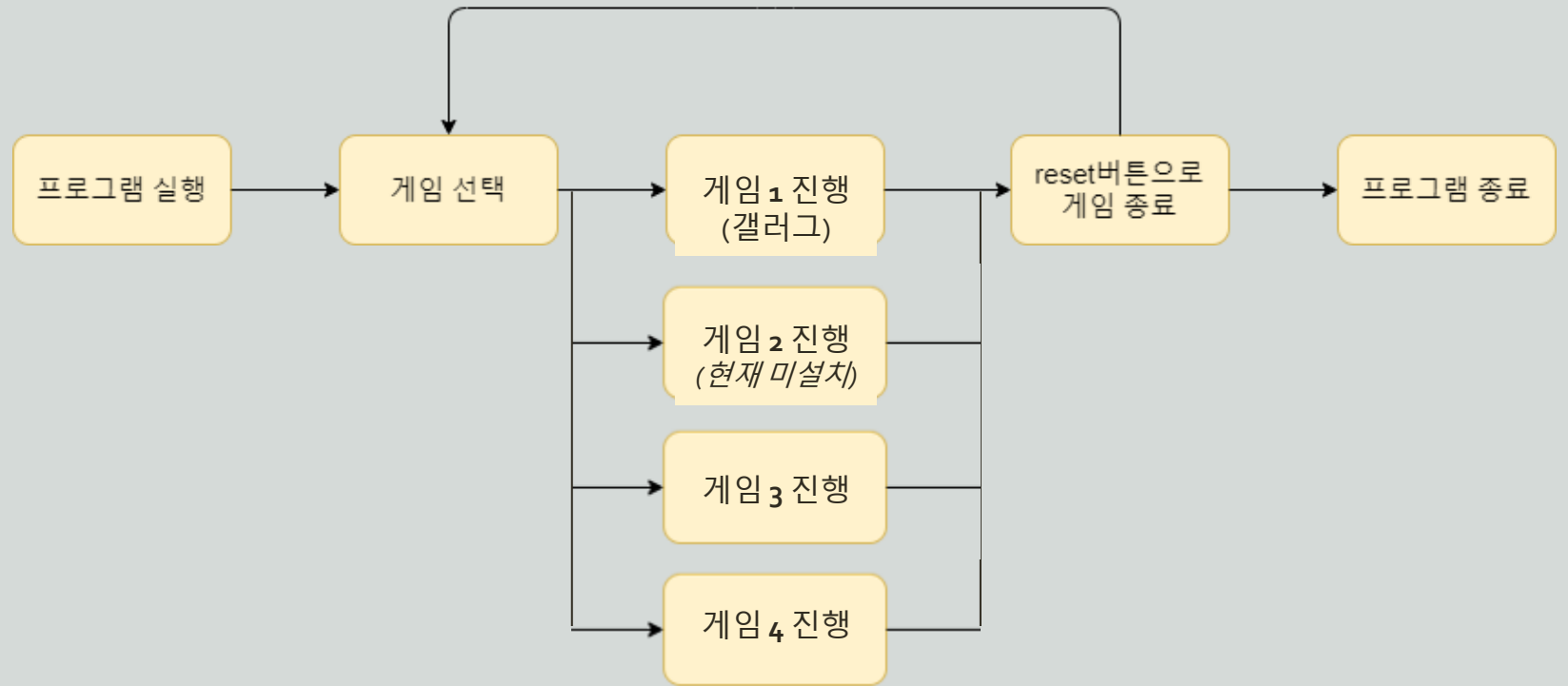
B. 갤러그 (Achro-EM 포트)

- 1981년에 출시된 아케이드 게임인 갤러그를 Achro-EM 보드에 실음.
- Push Switch 디바이스로 조작하며, 게임 화면은 Dot Matrix, 추가 정보는 Text LCD 디바이스와 LED 디바이스를 사용.

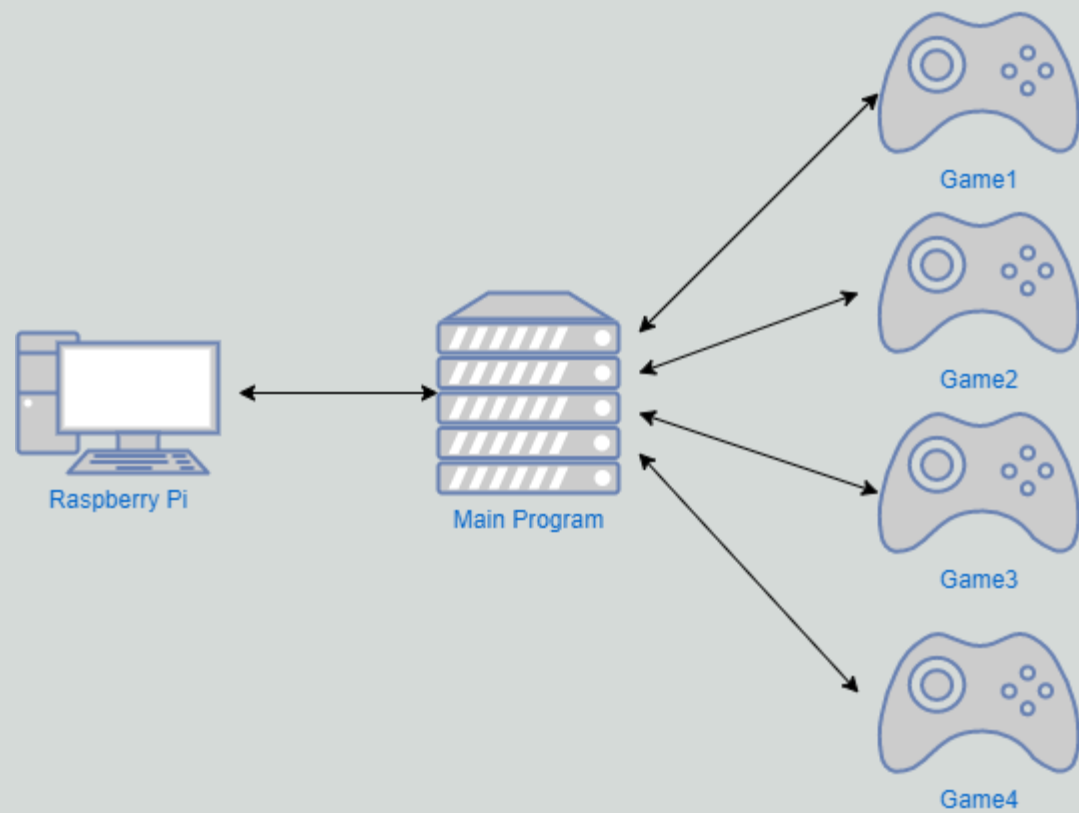
활용 디바이스 종류 및 목적

- **A. 메뉴 기능**
 - Push Switch 디바이스: 메뉴 순회
 - Text LCD 디바이스: 선택된 메뉴의 번호를 텍스트 형식으로 출력
 - LED 디바이스: 선택된 메뉴 표시
- **B. 갤러그**
 - Push Switch 디바이스: 플레이어 조작
 - Dot Matrix 디바이스: 게임 화면(GUI)
 - Text LCD 디바이스: 점수 등의 추가 정보 표시
 - LED 디바이스: 선택된 메뉴 표시

프로젝트 시나리오



시스템 구성도



기능 분석

A. 메뉴 기능

메뉴를 순회하는 기능

- 키패드를 생각할 때, 4번을 누르면 왼쪽으로 순회.
- 6번을 누르면 오른쪽으로 순회
- 5번을 누르면 지정한 게임을 선택하고 실행

```
void* thread_inputUserKeymain()
{
    unsigned char push_sw_buff[MAX_BUTTON]; //push_switch buffer
    int buff_size; //buffer size
    buff_size = sizeof(push_sw_buff);
    usleep(50000);
    read(push_dev, &push_sw_buff, buff_size);
    if (push_sw_buff[3] == 1)
    {
        gamenmber = gamenmber - 1;
        if (gamenmber == 0)
            gamenmber = 4;
        pushswitch = 0;
        thread_main();
    }
    else if (push_sw_buff[5] == 1)
    {
        gamenmber = gamenmber + 1;
        if (gamenmber == 5)
            gamenmber = 1;
        pushswitch = 0;
        thread_main();
    }
    else if (push_sw_buff[4] == 1)
    {
        selectgame = 1;
        choice_game();
        selectgame = 0;
    }
    else if (push_sw_buff[0] == 1)
    {
        soul--;
        if (soul < 1)
        {
```


기능 분석

A. 메뉴 기능

- **Text LCD**

어떠한 게임을
선택하였는지를 출력

- **LED**

선택한 게임 번호를 표시

- 이 모든 것이 쓰레드로
나뉘어 작동되고 있음.

```
void print_main()
{
    int len;
    char buf1[16] = "game 1";
    char buf2[16] = "game 2";
    char buf3[16] = "game 3";
    char buf4[16] = "game 4";
    unsigned char string[32];
    memset(string, 0, sizeof(string));
    if (pushswitch == 0)
    {
        switch (gamenmber)
        {
            case 1:
                len = strlen(buf1);
                if (len > 0)
                {
                    strncat(string, buf1, len);
                    memset(string + len, ' ', LINE_BUFF - len);
                }

                break;
            case 2:
                len = strlen(buf2);
                if (len > 0)
                {
```

기능 분석

B. 갤러그

게임 논리상 다루어야 할 개체들

1. 플레이어 캐릭터
 2. 플레이어 발사체
 3. 적 개체들
 4. 적 발사체
- 이들을 그들의 위치에 맞게 이동시키고, Dot Matrix에 출력

```
void game() { // 게임 실행
    while (1) {
        dev = open(FPGA_DOT_DEVICE, O_WRONLY);
        if (dev < 0) {
            printf("Device open error : %s\n", FPGA_DOT_DEVICE);
            exit(1);
        }
        str_size = sizeof(player[m]);
        write(dev, player[m], str_size);
        playermove();
        str_size = sizeof(player[m]);
        write(dev, player[m], str_size);
        bulletmove();
        str_size = sizeof(player[m]);
        write(dev, player[m], str_size);
        enemybulletready();
        playermove();
        enemybullet();
        str_size = sizeof(player[m]);
        write(dev, player[m], str_size);
        bulletmove();
        str_size = sizeof(player[m]);
        write(dev, player[m], str_size);
        if (b == 0) {
            break;
        }
        else if (a == 0) {
            break;
        }
        sleep(1);
    }
}
```

기능 분석

B. 갤러그

플레이어 조종

- 플레이어의 기본 위치에 배치하고,
- Push Switch의 왼쪽 이나 오른쪽 버튼을 누르면 그에 맞게 이동

```
unsigned char player[5][10] = { // 플레이어 출력
    { 0x3e, 0x3e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x08, 0x1c }, //기본
    { 0x3e, 0x3e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x04, 0x0e }, //우1
    { 0x3e, 0x3e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x02, 0x07 }, //우2
    { 0x3e, 0x3e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x10, 0x38 }, //좌1
    { 0x3e, 0x3e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x30, 0x70 } //좌2
};

void playermove() {
    unsigned char push_sw_buff[MAX_BUTTON]; //push_switch buffer
    int buff_size; //buffer size
    buff_size = sizeof(push_sw_buff);
    read(push_dev, &push_sw_buff, buff_size);
    if (m == 0 && push_sw_buff[5] == 1) { //우버튼 누름
        m = 1;
    }
    else if (m == 1 && push_sw_buff[5] == 1) { //우버튼 누름
        m = 2;
    }
}
```

기능 분석

B. 갤러그

발사체의 이동 성질 설정

- 발사체가 어떻게 이동하는지 프로그래밍.
- 플레이어의 위치와 다음으로 이동할 위치에 따라 발사체의 위치가 정해짐.

```
void bulletmove() {
    unsigned char push_sw_buff[MAX_BUTTON]; //push_switch buffer
    int buff_size; //buffer size
    buff_size = sizeof(push_sw_buff);
    read(push_dev, &push_sw_buff, buff_size);
    if (m = 0 && push_sw_buff[4] == 1) {
        unsigned char bull1[6][10] = {
            { 0x3e, 0x3e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x08, 0x08, 0x1c },
            { 0x3e, 0x3e, 0x00, 0x00, 0x00, 0x00, 0x08, 0x00, 0x08, 0x1c },
            { 0x3e, 0x3e, 0x00, 0x00, 0x00, 0x08, 0x00, 0x00, 0x08, 0x1c },
            { 0x3e, 0x3e, 0x00, 0x00, 0x08, 0x00, 0x00, 0x00, 0x08, 0x1c },
            { 0x3e, 0x3e, 0x00, 0x08, 0x00, 0x00, 0x00, 0x00, 0x08, 0x1c },
            { 0x3e, 0x3e, 0x08, 0x00, 0x00, 0x00, 0x00, 0x00, 0x08, 0x1c }
        };
        for (i = 0; i < 6; i++) {
            str_size = sizeof(bull1[i]);
            write(dev, bull1[i], str_size);
            sleep(1);
        }
        a = a - 1;
    }
    else if (m = 1 && push_sw_buff[4] == 1) {
        unsigned char bull2[6][10] = {
            { 0x3e, 0x3e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x04, 0x04, 0x0e },
            { 0x3e, 0x3e, 0x00, 0x00, 0x00, 0x00, 0x04, 0x00, 0x04, 0x0e },
            { 0x3e, 0x3e, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00, 0x04, 0x0e },
            { 0x3e, 0x3e, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00, 0x04, 0x0e },
            { 0x3e, 0x3e, 0x00, 0x04, 0x00, 0x00, 0x00, 0x00, 0x04, 0x0e },
            { 0x3e, 0x3e, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x04, 0x0e }
        };
        for (i = 0; i < 6; i++) {
            str_size = sizeof(bull2[i]);
            write(dev, bull2[i], str_size);
            sleep(1);
        }
        a = a - 1;
    }
    else if (m = 2 && push_sw_buff[4] == 1) {
        unsigned char bull3[6][10] = {
```

기능 분석

B. 갤러그

적의 위치와 발사체 계산

- 적들을 움직이게 한 후,
- 랜덤 값을 주어, 불규칙적인 성질을 줌.
- 그로 인해, 정기적으로 발사체를 쏘지 않고, 무작위로 쏴.
 - 게임의 난이도를 위하여 예측성을 줄이기 위함.

```
void enemybullet() {
    if (bull = 1) {
        unsigned char bull6[4][10] = { //양쪽 끝 2,6
            {0x3e, 0x3e, 0x00, 0x22, 0x22, 0x00, 0x00, 0x00, 0x00, 0x00},
            {0x3e, 0x3e, 0x00, 0x00, 0x22, 0x22, 0x00, 0x00, 0x00, 0x00},
            {0x3e, 0x3e, 0x00, 0x00, 0x00, 0x22, 0x22, 0x00, 0x00, 0x00},
            {0x3e, 0x3e, 0x00, 0x00, 0x00, 0x00, 0x22, 0x22, 0x00, 0x00}
        };
        for (i = 0; i < 4; i++) {
            str_size = sizeof(bull6[i]);
            write(dev, bull6[i], str_size);
        }
        if (m == !0) {
            b = b - 1;
        }
    }
    else if (bull = 2) {
        unsigned char bull6[4][10] = { //중간
            {0x3e, 0x3e, 0x00, 0x08, 0x08, 0x00, 0x00, 0x00, 0x00, 0x00},
            {0x3e, 0x3e, 0x00, 0x00, 0x08, 0x08, 0x00, 0x00, 0x00, 0x00},
            {0x3e, 0x3e, 0x00, 0x00, 0x00, 0x08, 0x08, 0x00, 0x00, 0x00},
            {0x3e, 0x3e, 0x00, 0x00, 0x00, 0x00, 0x08, 0x08, 0x00, 0x00}
        };
        for (i = 0; i < 4; i++) {
            str_size = sizeof(bull6[i]);
            write(dev, bull6[i], str_size);
        }
        if (m == 2 || m == 4) {
            b = b - 1;
        }
    }
}
```

```
void enemybulletready() {
    srand((unsigned int)time(NULL));
    bull = rand() % 2;
    if (bull = 1) {
        unsigned char bull6ready[10] = { //양쪽 끝 2,6
            0x3e, 0x3e, 0x22, 0x22, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
        };
        str_size = sizeof(bull6ready);
        write(dev, bull6ready, str_size);
    }
    else if (bull = 2) {
        unsigned char bull7ready[10] = {
            0x3e, 0x3e, 0x08, 0x08, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
        };
        str_size = sizeof(bull7ready);
        write(dev, bull7ready, str_size);
    }
    //중간 4
    sleep(1);
}
```

Thank you