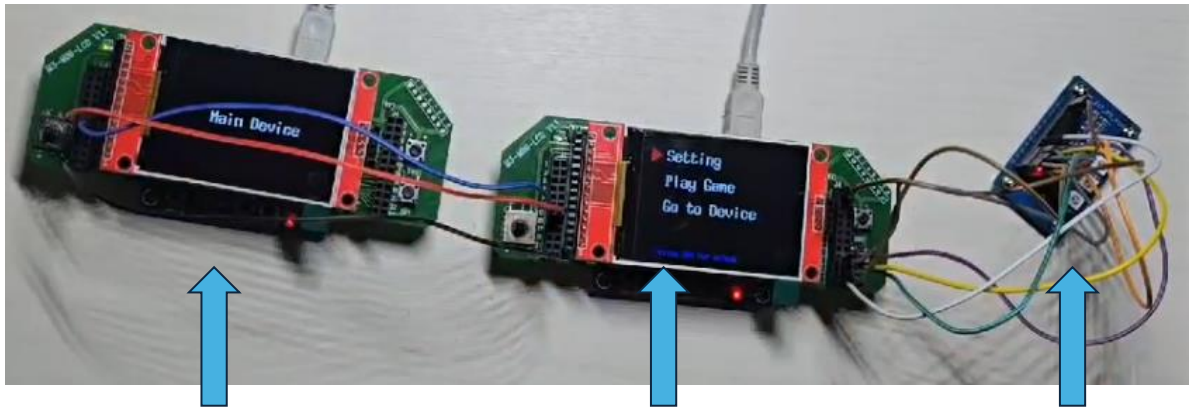
대한상공회의소
서울기술교육센터

Project Z :
연구소 탈환 작전

ARM Architecture project 발표

AI 시스템 반도체 설계(2기) 윤종민

개요



보드2(단말기) 보드1(메인 기기) M3-EXT(체력 display)

게임 설명

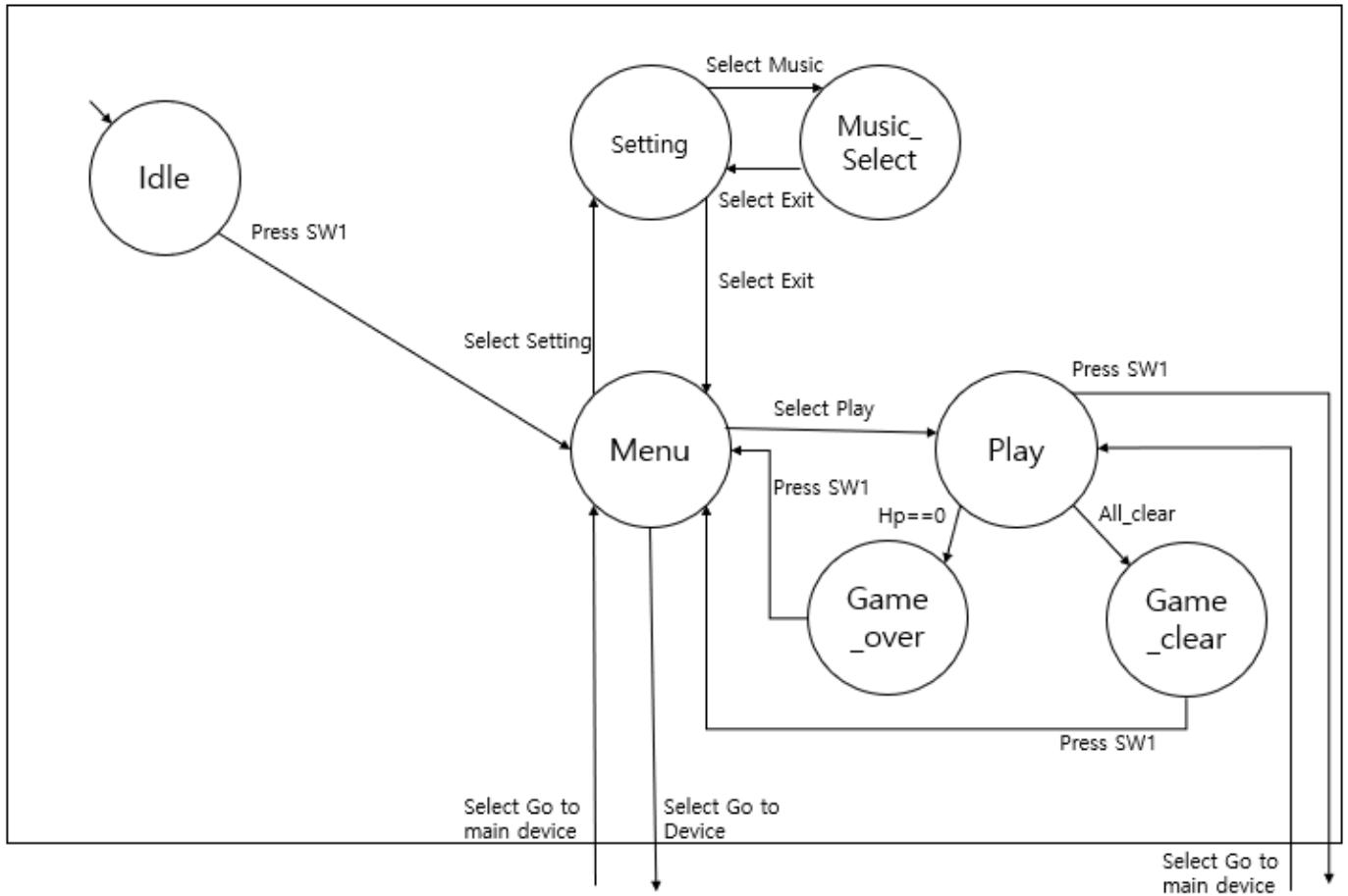
- 게임 제목: "Project Z: 연구소 탈환 작전"
- 게임 목적 : 좀비와 경비 로봇을 피하면서 연구소 탈환
- 게임 플레이: 메인 기기에서 진행 -> 단말기로 key 획득 후 stage 이동

개발 일정

날짜	개발 계획	실제 진행 사항
4.28 (월)	오브젝트 개발 시작	각 오브젝트 구조체 정의 및 상태 천이 구조 뼈대, 움직임 구현
4.29 (화)	오브젝트 개발 완료 및 충돌 판정 개발	이동 + 플레이어 vs 좀비, 충돌 로직 구현
4.30 (수)	단말기 시스템 개발	이동 + 플레이어 vs 좀비, 충돌 로직 구현
5.1 (목)	해킹 미니게임 개발	총알, 독액 움직임 충돌 로직 구현
5.2 (금)	체력 시스템 및 클리어 처리 시스템 개발	총알, 독액 움직임 충돌 로직 구현
5.3 (토)	디버깅 및 난이도 조정	총알, 독액 움직임 충돌 로직 구현 + 단말기 구조 뼈대 구현
5.4 (일)	추가 기능 업데이트 및 최종 점검	단말기 시스템 완성, stage별 오브젝트 생성 수 조절, UART2 통신과 체력 시스템 구현
5.5(월)	영상 촬영 및 문서 작성	UART2 통신 구현, 체력 시스템+SPI 연결, 음악 추가, 상태 이동 마무리 시연 영상, 게임 설명서, 소스파일, bin파일, 발표자료 제작
5.6(화)	문서 작성 마무리 및 제출	마지막 확인 및 제출

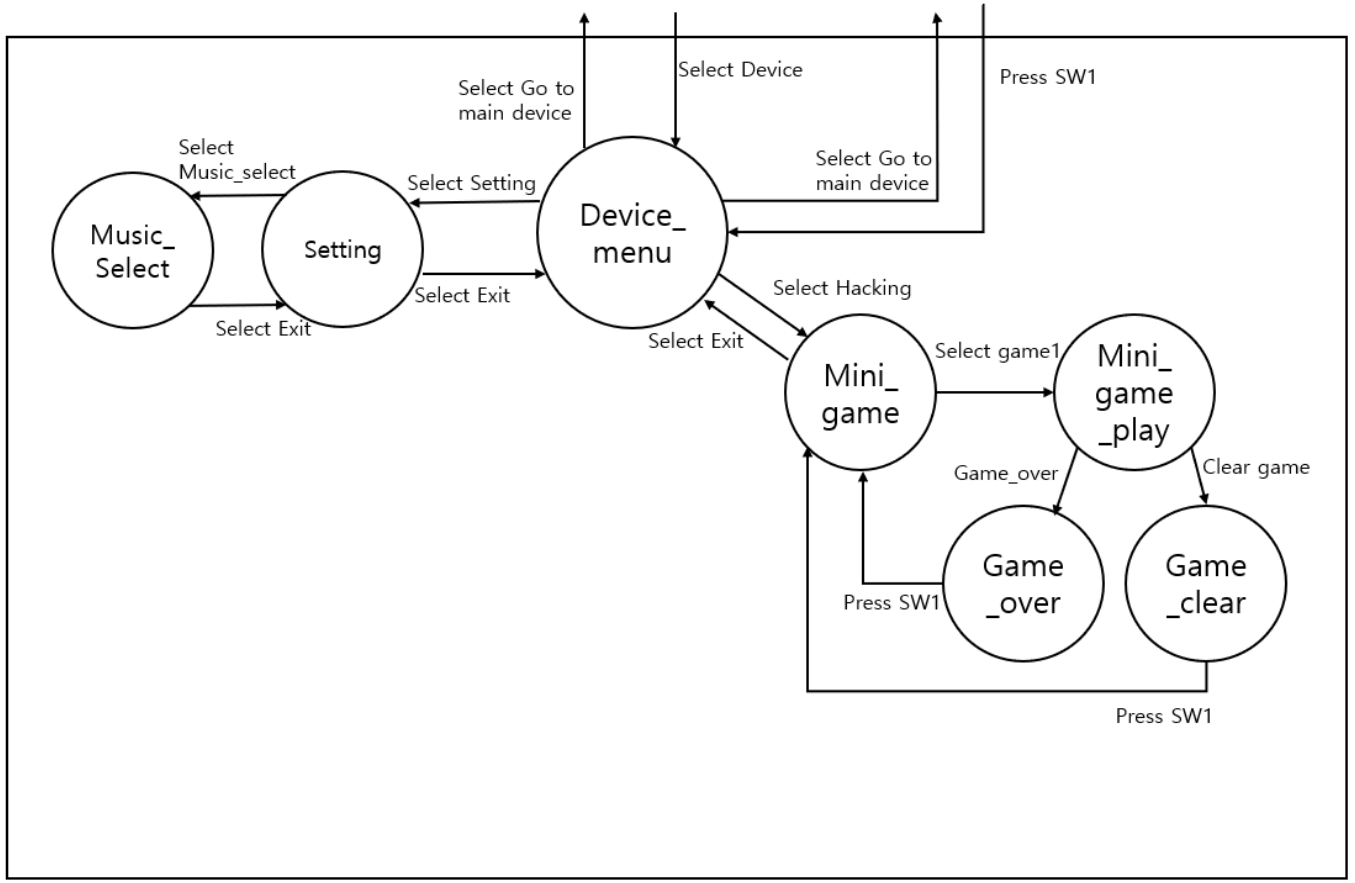
개발 결과

메인 기기 상태 천이도



개발 결과

단말기 상태 천이도

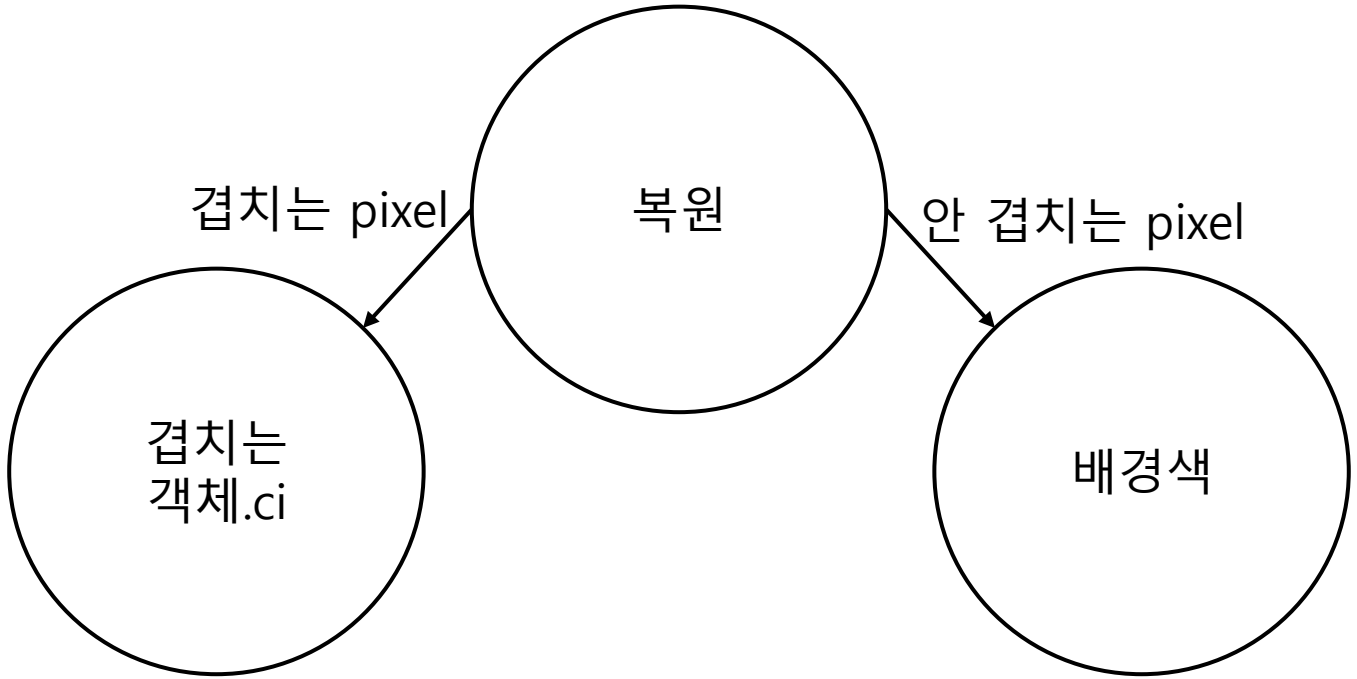


핵심 기술 및 코드

발사체 이동

기본적인 이동: 복원->이동->충돌(검사)->그리기

➡ 그럼 발사체는? (플레이어, 좀비 등을 지나감)



핵심 기술 및 코드

발사체 이동

```
r = color[BACK_COLOR];
```

➡ 처음에 배경색으로 지정

```
for(k=0;k<zmax;k++)
    if(zs[k].alive &&check_collision_pixel(px,py,zs[k].base.x,zs[k].base.y,zs[k].base.w,zs[k].base.h)){
        r=color[zs[k].base.ci]; break;
    }
```

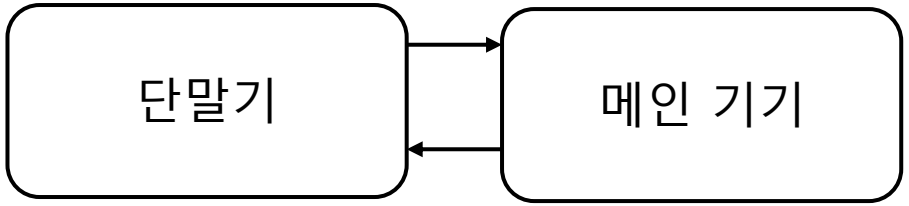
➡ Pixel마다 각각의 오브젝트와 겹치는지 확인
-> 이 경우 살아있는 좀비와 해당 픽셀이 겹치는지 확인

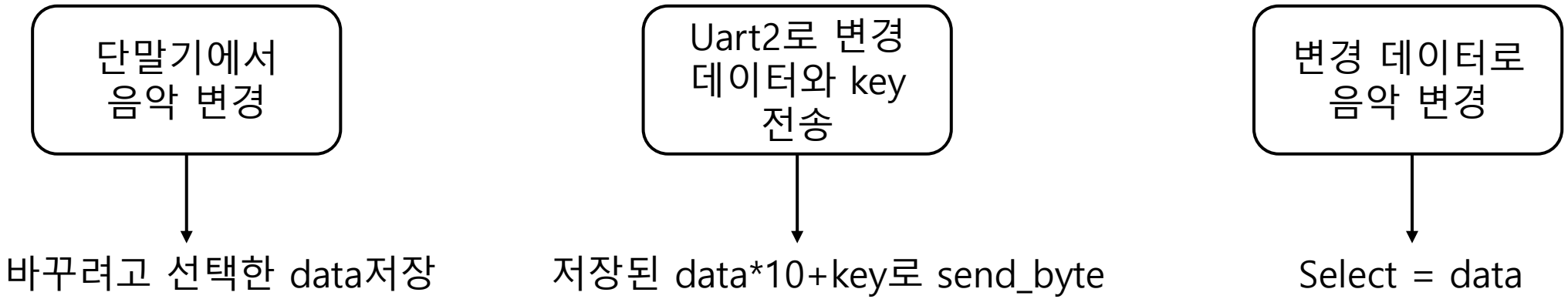
```
Lcd_Put_Pixel(px,py,r);
```

➡ 정해진 color값으로 복원

핵심 기술 및 코드

다른 기기에서 설정 바꾸기





핵심 기술 및 코드

다른 기기에서 설정 바꾸기

```
music_select=3;
case 0: music_select=0; break; //song1 선택
case 1: music_select=1; break; //song2 선택
case 2: music_select=2; break; //mute
```

➡ 단말기에서 음악 변경(안 바꾸면 3)

```
Uart2_Send_Byte(music_select*10+key);
```

➡ 선택한 값*10+key를 byte로 전송

```
if(USART2_rx_ready)
{
    Uart1_Printf("rx_ready = 1, data = %d₩n", USART2_rx_data);
    state = before_device;
    if(((USART2_rx_data/10) != music_select)&&((USART2_rx_data/10) != 3)) {music_select = (USART2_rx_data/10); TIM3_Out_Stop(); song_idx=0;}
```

➡ 음악 변경시 selelct를 바꿈

➡ Rx 받을 때까지
Device state대기

결론

구현: 조종하는 오브젝트 1개, 자율이동 오브젝트 4종류, 보드간
uart통신, SPI 통신, buzzer 등등

아쉬운 사항, 업그레이드 아이디어:

- Setting 요소 추가(난이도 등)
- 이미지 추가
- 코드 업그레이드(Lookup Table 등)
- 메인 기기에서 보스 좀비, 함정, 아이템 박스 보상 추가
- 단말기에서 미니게임 종류 추가

개발 후기

얻은 점:
- 처음으로 ARM코어가 올라간 보드로 Low Level에서 코딩 경험
- LCD 이용한 프로젝트 경험

아쉬운 점:
- 고급 포인터 부분 활용을 충분히 못해본 것
- 추가 기능과 오브젝트 추가를 못한 것
- 충분하지 못한 디버깅 시간

개발 후기

삽질의 추억

Uart2 통신할 때 값을 잘못 보내서 값이 잘못 찍히는 것이었는데 값을 받는 쪽만 확인하고 있었음

➔ 모든 부분을 찍어보면서 넘어오는 값이 문제 있음을 발견

➔ 이후 값을 보내는 것이 아니고 보낼 값을 만드는 과정만 확인

➔ 보내기 직전에 찍어보고 나서야 보내는 부분이 문제 있음을 확인 후 해결

아이템 박스가 하나만 있을 때는 문제없는데, 두 개 이상이면 하나 제외한 나머지가 먹어도 안 없어 지는 등 문제 발생

➔ Main함수 내에서 아이템 박스를 움직이는 부분만 집중

➔ 아이템 박스를 움직이고, 플레이어가 아이템 박스를 획득하는 함수에서 일정 부분이 item_boxes[i]가 아니고 item_boxes[0]인 것을 발견하여 해결

TIM2를 buzzer delay를 위해 사용하는데 ISR을 작성하였는데도 TIM2의 인터럽트 발생 안함

➔ Mian 함수에서 TIM2 인터럽트 발생시 처리 코드만 집중

➔ 나중에 uart로 찍어본 결과 인터럽트 발생 안함

➔ 결국 ISR의 내용이 TIM4것을 복사하고 TIM2로 안 바꾼 것을 발견하여 해결