

장창용 최다원

허지웅 박상운

IOT 프로그래밍 프로젝트 2팀



목차

- 구상 작품 설명
- 수행 내용
 - 자료조사
 - 알고리즘
 - 하드웨어 활용
 - 임시 코드
- 앞으로의 계획



구상 작품

- H-SMART4412를 이용한 갤러그



수행 내용



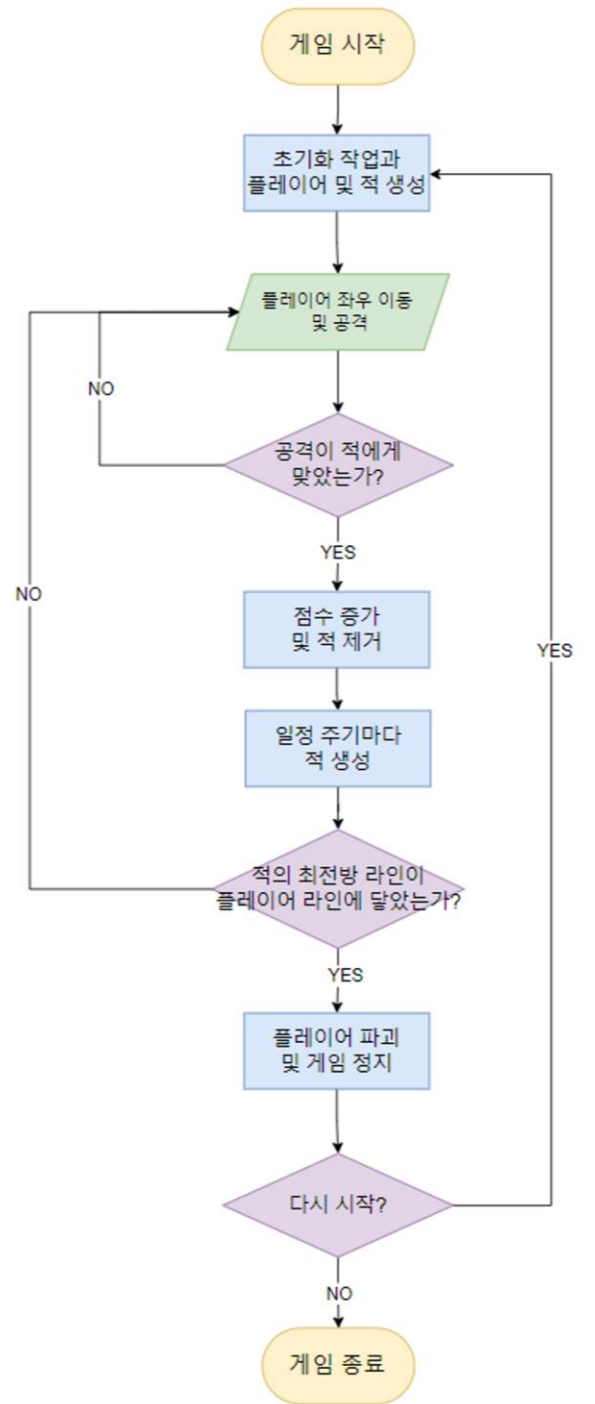


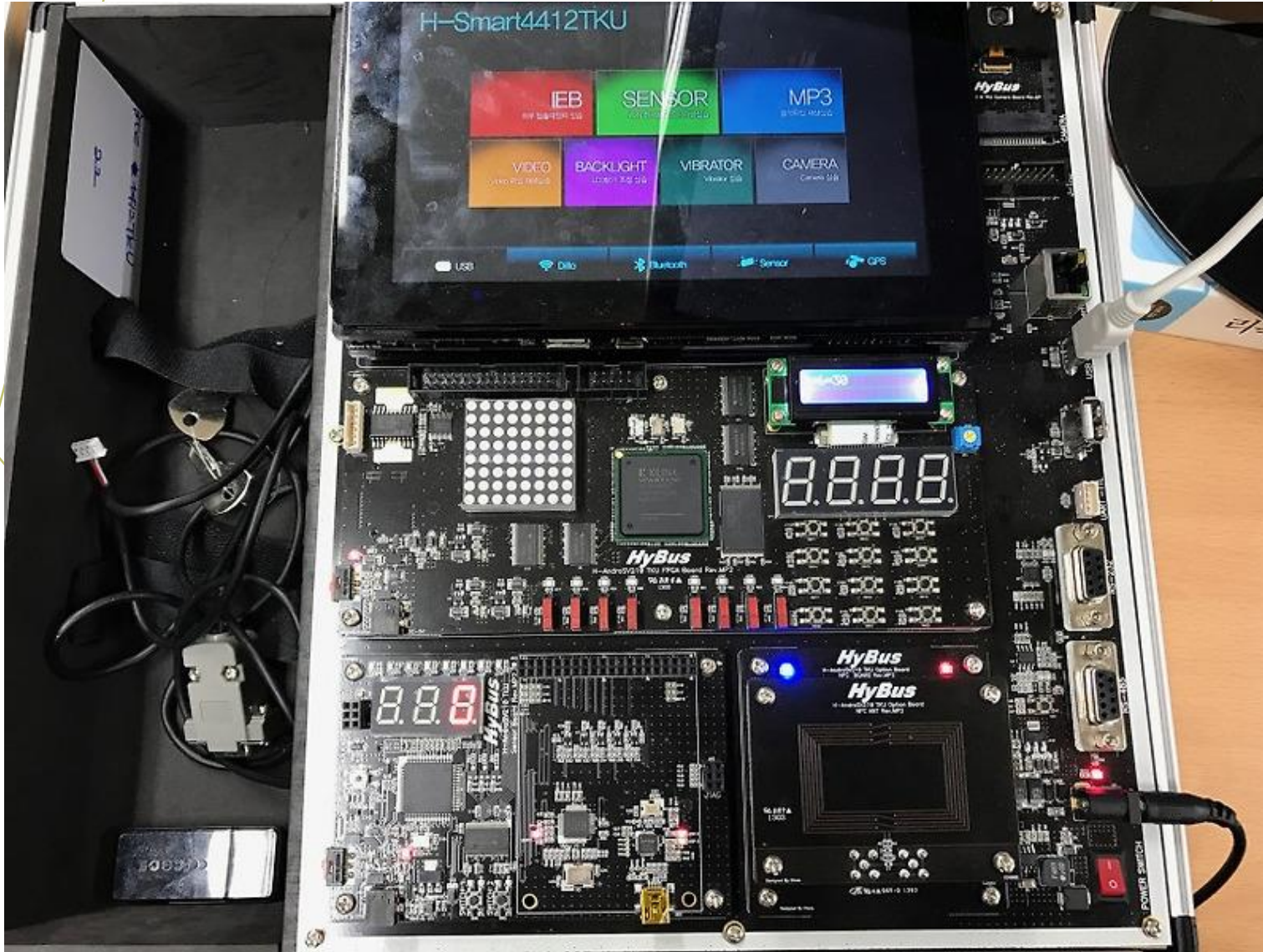
자료 조사

- H-SMART4412 임베디드 시스템 개발
 - C++언어로 시스템 코드 작성
 - 코드 파일 전송 시 TeraTerm 메뉴-전송-ZMODEM-보내기
 - 사용 장치들과 상호작용 코드 분석

알고리즘

- 초기화 및 게임시작
 - 점수 초기화 및 플레이어, 적 재생성
- 게임 진행
 - 적은 플레이어를 향해 내려옴
 - 플레이어는 적에게 총알을 발사해 적중시 적을 제거 점수 증가
- 게임 종료
 - 적이 플레이어에게 닿으면 플레이어 제거 및 게임오버 화면 표시

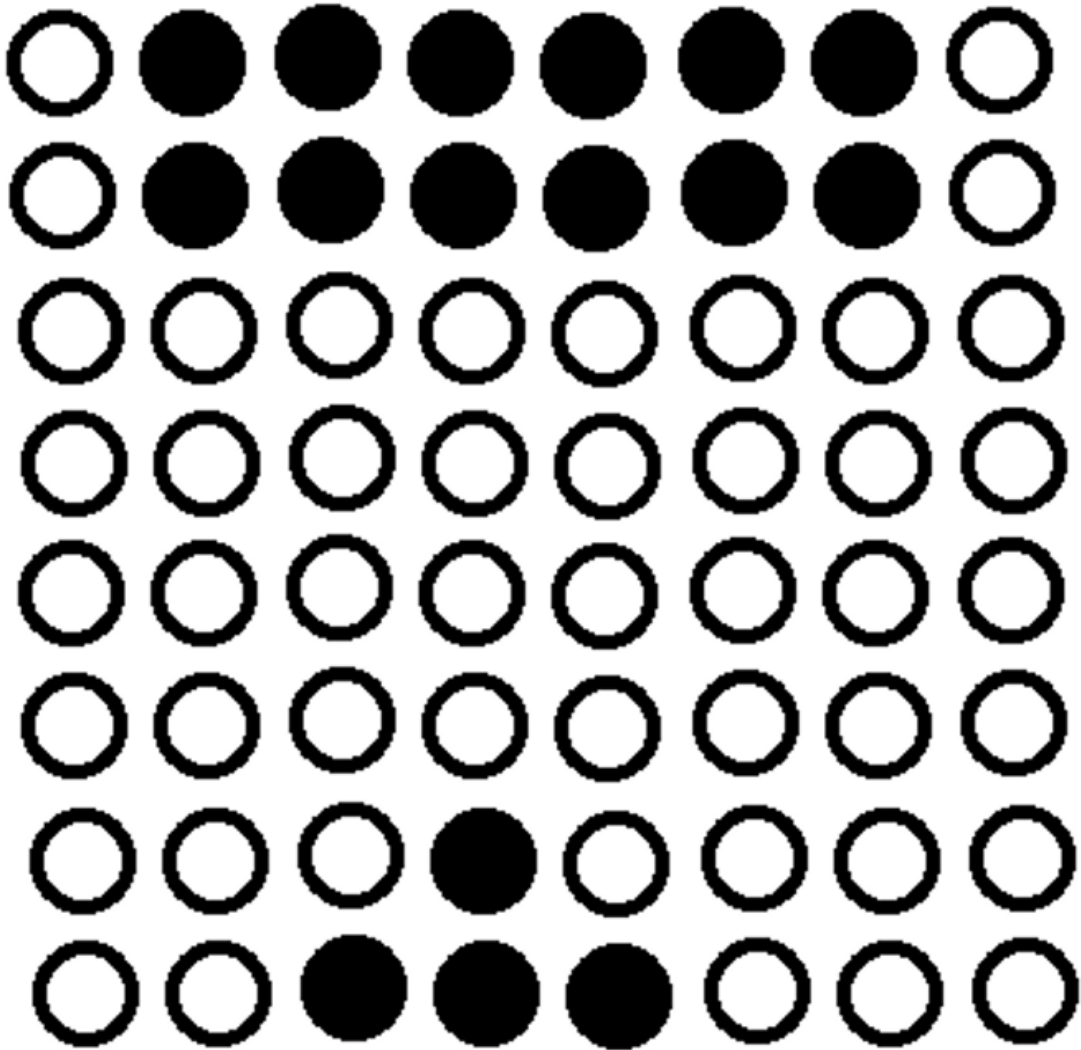




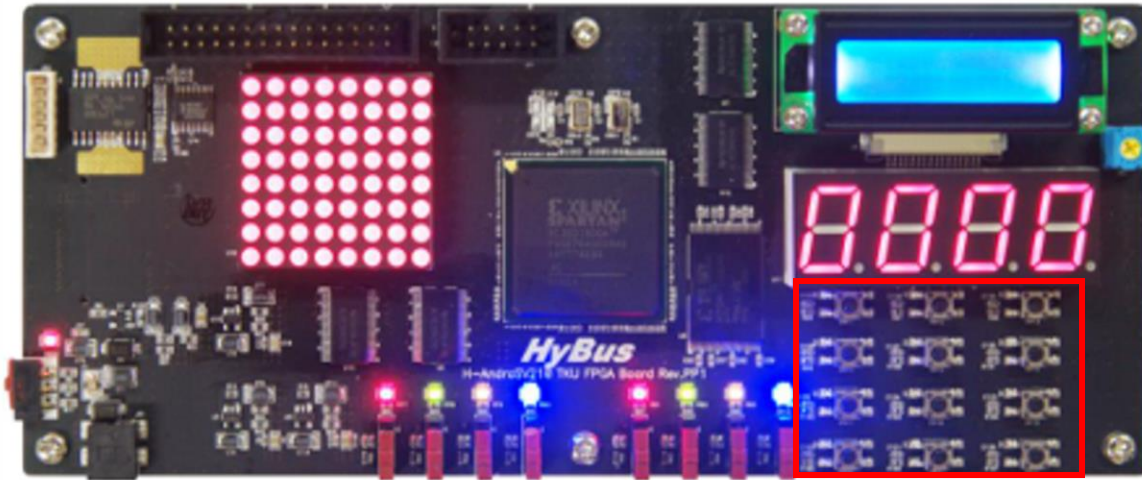
하드웨어
활용

8x8 Dot Matrix

- 적과 플레이어
플레이어의 총알을 출력

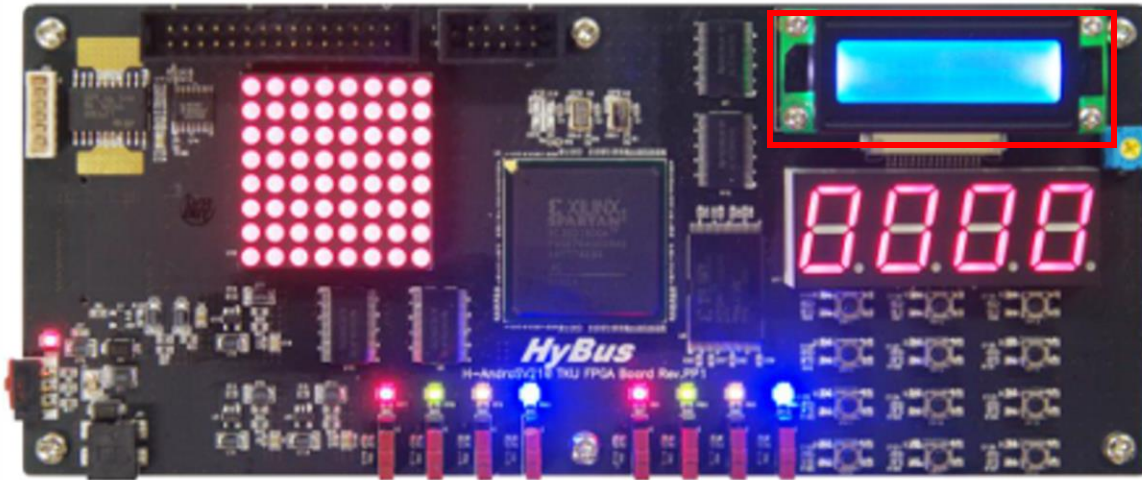


Tact Switch



- 플레이어의 상하좌우 이동에 네 개의 버튼 사용
- 플레이어의 총알 발사에 하나의 버튼 사용

CLCD



- 게임 시작 시
"Game Start" 문구 출력
- 게임 종료 시
"Game Over" 문구 출력
- 게임 중과 종료 시
점수 출력

```
// Writes The Matrix Data To Dot Matrix
void drawToMatrix()
{
    openDot();
    unsigned char rows[8];
    for(int i = 0; i < 8; i++)
    {
        rows[i] = 0;
        for(int j = 0; j < 8; j++)
        {
            if(matrix[i][j])
            {
                rows[i] |= 1 << j;
            }
        }
    }
    write(dot_fd, &rows, sizeof(rows));
    closeDot();
}

int main(int argc, const char *argv[])
{
    int dot;
    unsigned char c[26][8] =
    {{0x18, 0x24, 0x42, 0x42, 0x7E, 0x42, 0x42, 0x42}, //A
    {0x3E, 0x22, 0x22, 0x3C, 0x22, 0x22, 0x3C}, //B
    {0x1C, 0x22, 0x20, 0x20, 0x20, 0x22, 0x1C}, //C
    {0x38, 0x44, 0x42, 0x42, 0x42, 0x44, 0x38}, //D
    {0x3E, 0x20, 0x20, 0x3E, 0x20, 0x20, 0x3E}, //E
    {0x3E, 0x20, 0x20, 0x3E, 0x20, 0x20, 0x20}, //F
    {0x1C, 0x22, 0x42, 0x40, 0x40, 0x47, 0x42, 0x3C}, //G
    {0x42, 0x42, 0x42, 0x7E, 0x42, 0x42, 0x42}, //H
    {0x1C, 0x08, 0x08, 0x08, 0x08, 0x08, 0x1C}, //I
    {0x1C, 0x08, 0x08, 0x08, 0x08, 0x48, 0x30}, //J
    {0x44, 0x48, 0x50, 0x60, 0x50, 0x48, 0x44, 0x44}, //K
    {0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x3E}, //L
    {0x81, 0xC3, 0xA5, 0x99, 0x81, 0x81, 0x81, 0x81}, //M
    {0x42, 0x62, 0x52, 0x4A, 0x46, 0x42, 0x42, 0x42}, //N
    {0x3C, 0x42, 0x81, 0x81, 0x81, 0x42, 0x3C}, //O
    {0x7C, 0x42, 0x42, 0x42, 0x7C, 0x40, 0x40, 0x40}, //P
    {0x38, 0x44, 0x82, 0x82, 0x82, 0x8A, 0x44, 0x3A}, //Q
    {0x7C, 0x42, 0x42, 0x42, 0x7C, 0x48, 0x44, 0x42}, //R
    {0x3C, 0x42, 0x40, 0x3C, 0x82, 0x02, 0x42, 0x3C}, //S
    {0x3E, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08}, //T
    {0x42, 0x42, 0x42, 0x42, 0x42, 0x42, 0x3C}, //U
    {0x81, 0x42, 0x42, 0x42, 0x24, 0x24, 0x24, 0x18}, //V
    {0x81, 0x09, 0x09, 0x09, 0x09, 0x09, 0x5A, 0x24}, //W
    {0x81, 0x42, 0x24, 0x18, 0x18, 0x24, 0x42, 0x81}, //X
    {0x81, 0x81, 0x42, 0x42, 0x3C, 0x18, 0x18, 0x18}, //Y
    {0xFF, 0x03, 0x02, 0x04, 0x08, 0x10, 0x60, 0xFF}} //Z;
    printf("Test Code");
    if((dot_d = open(dot, O_RDWR)) < 0)
    {
        printf("Can't Open\n");
        exit(0);
    }
    for(i=0;i<26;i++)
    {
        write(dot_d,&c[i],sizeof(c[i]));
        sleep(1);
    }
    close(dot_d);
    return 0;
}
```

```
// Header File
#include<stdio.h> // 입출력 관련
#include<stdlib.h> // 문자열 변환, 메모리 관련
#include<unistd.h> // POSIX 운영체제 API에 대한 액세스 제공
#include<fcntl.h> // 타겟시스템 입출력 장치 관련
#include<sys/types.h> // 시스템에서 사용하는 자료형 정보
#include<sys/ioctl.h> // 하드웨어의 제어와 상태 정보
#include<sys/stat.h> // 파일의 상태에 대한 정보
#include <string.h> // 문자열 처리
#include <time.h> // 시간 관련

// Target System
#define dot "/dev/dot" // Dot Matrix
#define tact "/dev/tactsw" // Tact Switch
#define led "/dev/led" // LED
#define clcd "/dev/clcd" // Character LCD
#define COORD_SET
typedef struct coord // Coordinates Within 8X8 Dot Matrix
{
    int y;
    int x;
} coord;

class DotMatrix
{
private:
    int dot_fd; // File Descriptor To Represent Dot Matrix
    bool matrix[8][8]; // 8X8 Dot Matrix Status Boolean Array

    void openDot(){dot_fd = open(dot, O_WRONLY);} // Open Target Dot Matrix
    void closeDot(){close(dot_fd);} // Close Target Dot Matrix

public:
    DotMatrix() // Constructor Init
    {
        dot_fd = -1;
        memset(matrix, 0, sizeof(bool) * 8 * 8); // Reset Memory
    }
    ~DotMatrix(){close(dot_fd);} // Destructor

    // Reset All Dot Matrix
    void clear(){memset(matrix, 0, sizeof(bool) * 8 * 8);
    // Set Specific Pixel At Coordinates (C.y, C.x) To True
    void set(coord C){matrix[C.y][C.x] = true;} // Sets Specific Pixel At Coordinates

    // Serial Debug Print
    void printToSerial()
    {
        for(int i = 0; i < 8; i++)
        {
            for(int j = 0; j < 8; j++)
            {
                printf(matrix[i][j] ? "x " : "o ");
            }
        }
        printf("\n");
    }
}
```

작성한 임시 코드

대략적으로 DOT MATRIX에
대해서 간단하게 켜보는
예제

앞으로의 계획

- 임시 코드 실제 실행
 - 버그가 있으면 수정
 - 버그가 없다면 이어서 코드 작성
- 새로 추가할만한 기능 기획 및 작성
 - 예시
 - 사운드 출력 기능
 - LED로 적의 공격 위치 표시