Embedded Systems Designs

Prof. 전석희

기말 과제

Avoid Balls!!!

Atmega128과 HC-06 Bluetooth module을

사용한, 안드로이드 스마트폰 게임

2012104030

정 준 영

1. 설계 목표 및 제작 동기

라즈베리파이와 같이 OS가 존재하는 개발 보드를 사용한 경험은 많았으나, Atmega128과 같은 AVR 개발 경험이 부족하여 많은 고민을 하였다. 따라서 기존에 라즈베리파이를 사용하여 안드로이드 스마트폰과 블루통신을 통한 게임 개발을 한 것을 이번 기말 과제로 정하여, 어셈블리와 C 언어를 통한 AVR 개발의 경험을 쌓아보고 싶었다.

처음 목표는 아이패드를 통한 ios 개발도 같이 하고 싶었으나, 어려움을 느끼고 안드로이드 개발에 집중하는 것으로 방향을 바꿨으며, AVR 시스템은 OS가 존재하는 시스템과 달리 단일 프로세스로 코딩을 해야 한다는 점에서 힘들었다. 하지만 그 만큼 많은 고민을 하게 되었으며, 앞으로도 더 공부가 필요함을 느끼게 되었다.

2. 설계 내용

A. 기능 설명

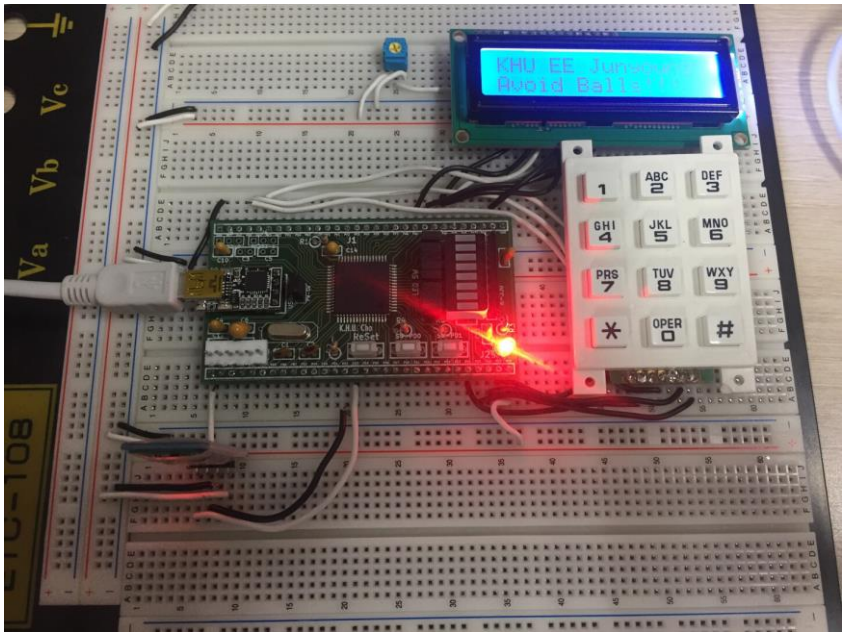
Atmega128에 UART1에 해당하는 PD2(RXD1), PD3(TXD1)포트에 HC-06 Bluetooth module을 연결한다. 또한, 키보드와 LCD도 연결한다. 이렇게 구성된 임베디드 보드는 안드로이드 스마트폰 어플리케이션과 블루투스 통신을 한다. 임베디드 보드의 키패드 "2, 8, 4, 6번" 키는 각각 "위, 아래, 왼쪽, 오른쪽"에 맵핑되어, 안드로이드 어플리케이션에 전달되고 안드로이드에서는 이를 이용하여 공 피하기 게임이 진행된다.

B. 임베디드 보드

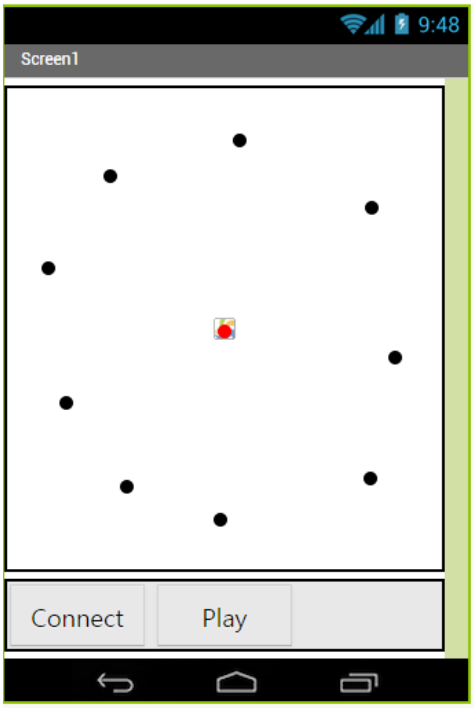
① LCD : 기본 정보(개발자 이름, 어플리케이션 명) 출력

② 키패드 : "Avoid Balls!!!" 에서 유저를 움직일 수 있는 방향키. "2, 8, 4, 6번" 키는 각각 "위, 아래, 왼쪽, 오른쪽"과 매핑된다.

③ HC-06 : 블루투스 모듈. UART1을 통해 Atmega128이 안드로이드 스마트폰과 블루투스 통신을 할 수 있게 한다.



< 전체 하드웨어 구성 >



< 안드로이드 View 화면 >

C. 안드로이드 어플리케이션

Connect 버튼을 통해 임베디드 보드와 블루투스 연결을 마친 후, Play 버튼을 통해 게임을 시작

한다. 게임이 시작하면 가운데의 빨간 공을 임베디드 보드의 키패드를 통해 움직일 수 있으며, 나머지 10개의 공들은 랜덤하게 움직인다. 랜덤하게 움직이는 공들은 빨간 공과 속도가 느린 공 3개, 같은 공 4개, 빠른 공 3개가 존재한다. 이 공들은 벽에 부딪히면 튕기면서 움직인다. 유저는 빨간 공을 움직여서 다른 공들에 부딪히지 않는 것을 목표로 한다.

D. 코드

임베디드 보드의 main 함수는 다음과 같다.

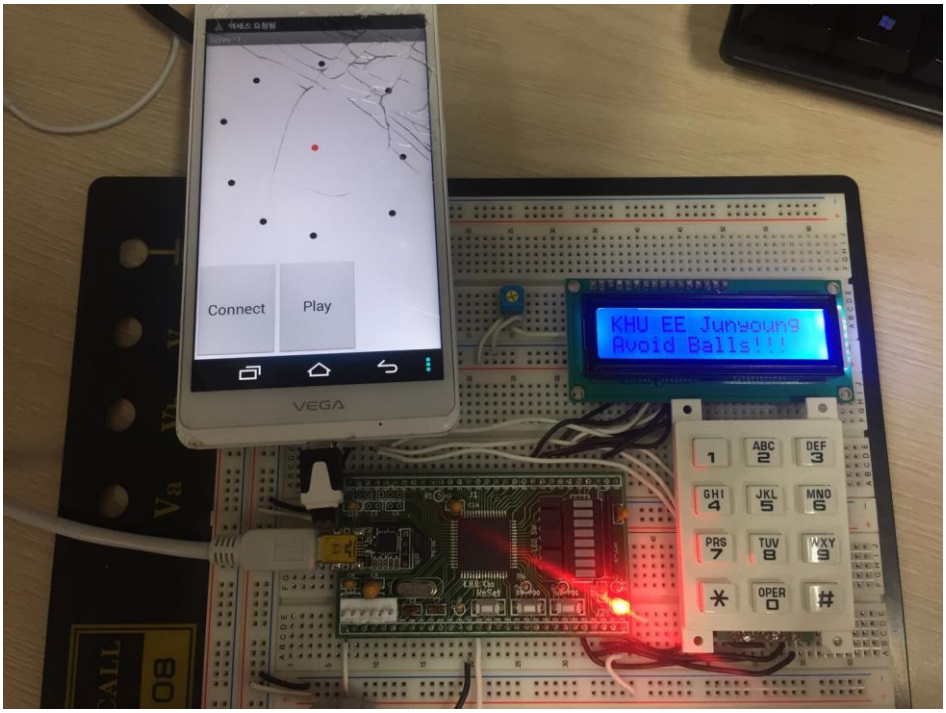
```
#  ifndef atmega128
    #define atmega128
#  endif
#define FOSC 16000000// Clock Speed#define BAUD 19200
#define BAUD 19200
#define MYUBRR FOSC/16/BAUD-1

#include <avr/io.h>           // Definition file for ATmega128
#include <avr/interrupt.h>
#include <stdio.h>

#include "cho_asm_io_def.h"   // I/O Port Definition for ATmega128
#include "GCC_c_asm_soft_delay_u_mSec.h"
#include "GCC_c_asm_uart.h"
#include "GCC_c_asm_lcd_hd44_4bit.h"
#include "GCC_get_keypad_multi.h"
#include "gnu_sciutil.h"

// SRAM Definitions
#define _SFR_OFFSET 0          // Needed to subtract 0x20 from I/O addresses

#define INPUT_FIRST_OPERAND 1
#define INPUT_SECOND_OPERAND 2
#define INIT_OPERATION 0

#define F_CPU 16000000                  // Clock Speed
#define BAUD0 19200
#define BAUD1 115200

#define MYUBRR0 (F_CPU/16/BAUD0-1)
#define MYUBRR1 (F_CPU/8/BAUD1-1)       // Double the USART Transmission Speed

void uart1_init(void);
unsigned char rx1_char(void);
void tx1_char(unsigned char data);

unsigned char operation_mode;   // Operation Mode

void Init_devices(void);
static int put_char(char c, FILE *stream);

volatile unsigned int cnt = 0;


static int put_char(char c, FILE *stream)
{
    UART_putchar(c);
    tx1_char(c);
    return 0;
}
```

```
//call this routine to initialize all peripherals
void Init_devices(void)
{
    unsigned int ubrr = MYUBRR;     // baud rate
    asm volatile(" cli ");          //disable all interrupts
    UART_Init(ubrr);                // UART 0 초기화, Set baud rate(19200)
    keypad_init();          // I/O Port init
    asm volatile(" sei ");  //  Global Interrupt Enable
    LCD_init();
    fdevopen(put_char,0);
    //all peripherals are now initialized
    operation_mode = INPUT_FIRST_OPERAND;
    keyCodeMode = KEY_CODE_TABLE_1;
}

void SCI_OutChar(char letter){
    UART_putchar(letter);
    tx1_char(letter);
}

unsigned char rx1_char(void)
{
    while ((UCSR1A&0x80) == 0);

    return UDR1;
}

void tx1_char(unsigned char data)
{

    while ((UCSR1A&0x20) == 0);

    UDR1 = data;
}

unsigned short SCI_InChar(){
  return (get_key());
}


unsigned short ASM_getchar(void){
  return (get_key());
}

int main (void)
{
    unsigned int num = 0;
    char input;
```

```
    Init_devices();

    UCSR1A = 0x00;
    UCSR1B = (1<<RXEN0)|(1<<TXEN0);
    UCSR1C = (1<<UCSZ01)|(1<<UCSZ00);
    UBRR1H = 0x00;
    UBRR1L = 51;

    LCD_SetCsr(1, 1);
    LCD_OutString("KHU EE Junyoung");
    LCD_SetCsr(2, 1);
    LCD_OutString("Avoid Balls!!!");


    // Infinite loop
    while(1){
        num = SCI_InUDec_OpCode(&input);

        input = get_key();

        SCI_OutChar(input);


    }

    return 0;
}
```

3. 결과

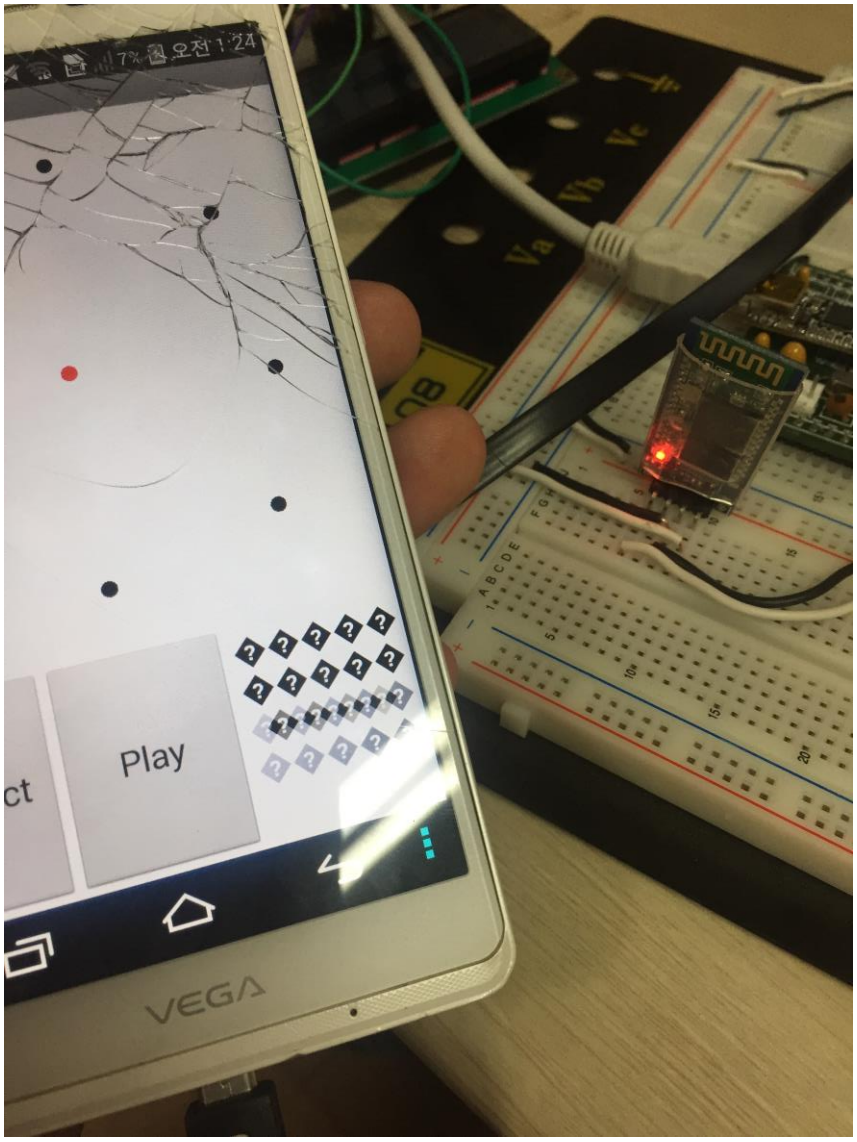
A. 실행 결과



< Avoid Balls!!! 실행 화면 >

B. 문제점

블루투스를 통해 전달 받은 키패드 입력 값이, 인코딩이 깨지면서 나오는 문제가 있었다. UART0에 연결되어 있는 USB Serial 통신과 같은 형식으로 코딩을 하였으나, ASCII 코드 형태가 아닌 Hex 코드 형태로 블루투스 값이 전달되었기 때문이다. 따라서 이를 인코딩하는 코드를 따로 코딩하여 집어 넣어야 했으며, 이 문제점을 찾고 해결하는데 많은 시간이 걸렸다.



< 어플리케이션이이 받은 블루투스 값이 깨져서 나오는 모습 >

4. 결론

처음 의도와는 다르게 결과물이 나와서 많이 아쉽다. 처음엔 React-native라는 페이스북에서 개발 중인 프로그램 언어를 사용하여, ios 코딩을 해보고 싶었으나, 프로젝트 제출 기간 안에 완성을 못하여서 급하게 안드로이드 코딩으로 변경을 하였다. 또한 그 코딩에 너무 집중하여, 주가 되어야 할 Atmega128 코딩을 깔끔하게 정리하지 못한 것이 아쉬웠다. 향후, 방학 동안 리팩토링에 힘써 완성을 시키는 것을 목표로 공부를 할 예정이다.