

# 최종 결과 보고서 (4팀)

주제 : 음표를 잡아라 - 미니 게임 만들기!

학번 : 20213009, 20213046

이름 : 강서현, 박하선

## 1. 시스템 개요 및 필요성

4x4 패드와 16x2 LCD를 사용해서 간단한 미니 게임을 제작한다. 짧은 동요를 선택해 특정 숫자를 미리 LCD 화면에 출력하여 보여 주면 게임 참여자는 해당하는 숫자를 4x4 패드로 입력하고, 실시간으로 입력된 음이 출력된다. 게임 시작 시 모드는 (1) 점수제 (2) 타임 어택 중 하나를 선택할 수 있도록 하고, 각 모드에 해당하는 룰은 다음과 같다. 게임이 끝나면 재시작할지 게임을 종료할지 정할 수 있다.

(1) 16x2의 LCD 패널 첫 행에 출력된 숫자를 4x4 패드에서 알맞게 눌렀으면 바로 다음 행에 'O'를 출력한다. 틀리면 점수가 깎이고 'X'를 출력한다. 점수제에서는 정답과 오답에 관계없이 숫자를 입력하면 바로 다음 열로 넘어가게 한다. 노래가 끝나면 점수를 합산해 최종 점수를 화면에 출력해 준다.

(2) 최대한 빨리 노래를 완성하는 것이 목표이다. 점수제와 마찬가지로 첫 행에 눌러야 하는 숫자를 출력한다. 출력된 숫자와 같은 숫자를 입력하면 숫자가 한 칸씩 왼쪽으로 스크롤된다. 맞게 입력한 숫자는 보이지 않게 되는 구조이다. 만약 출력된 숫자와 다른 숫자를 입력하면 경고음이 울리고 빨간색 LED가 점등한 다음 맞을 때까지 다음 열로 넘어가지 않도록 한다. 게임이 끝나면 끝날 때까지 걸린 시간을 화면에 출력해 준다.

16x2 LCD 화면이 콘솔 창과 비슷하다는 생각이 들었다. 마침 콘솔 게임을 하던 중 LCD 화면을 이용해서 비슷한 형태의 게임을 만들 수 있겠다는 생각에 보유 중인 다양한 요소들로 더 풍부한 기능을 가진 게임을 기획하게 되었다.

## 2. 계획 대비 변경 내용

[초기 화면]

- LCD 화면이 켜지면 사용자가 바로 모드 선택을 할 수 있도록 하려고 했으나, 게임 제목을 먼저 출력해 준 다음에 모드 선택하는 법을 출력해 주는 것으로 변경했다.

- 가변 저항으로 게임 모드가 정해지면 바로 게임을 실행할 수 있게 기획했지만 4x4 패드의 15번 버튼을 엔터 기능처럼 사용해 버튼을 눌러야 선택한 모드의 게임이 실행될 수 있도록 변경했다. (15번을 눌러야 한다는 안내문을 LCD에 출력해 준다.)

- 재시작 시 게임 제목, 게임 모드 선택 및 선택 방법, 15번 키 입력 메시지가 전부 출력되게 했으나 효율적이지 않다고 생각해서 처음 실행할 때만 출력될 수 있도록 `setup()`에 제목을 넣고 초기 화면 함수를 수정해 재시작이면 게임 모드 선택과 15번 키 입력 메시지만을 출력할 수 있도록 바꿨다.

#### [점수제 모드]

- 만점을 100점으로 생각했지만 점수제 노래에서 출력되는 숫자가 총 72개인 것을 고려해  $72 \times 5 = 360$ 으로 변경했다.
- 게임 결과가 만점이라면 "Full Combo!!!"를 출력해 준 다음에 점수를 출력하기로 했다. 만약 만점이 아니라면 "Clear!!!"를 출력한 후 점수를 출력해 준다.

#### [타임 어택 모드]

- 기존에는 점수제처럼 정답 여부를 'O', 'X'로 구분해서 보여 주는 것으로 했으나 타임 어택만의 차별을 두기 위해 맞은 키를 눌렀다면 첫 행에 출력된 문자열을 왼쪽으로 한 칸씩 스크롤되는 것으로 바꿨다. 만약 틀렸다면 LCD가 왼쪽으로 스크롤 되지 않고 경고음이 울리며 빨간 LED가 점등한 후 올바른 숫자를 입력할 때까지 기다린다. 첫 행에 출력된 숫자가 전부 스크롤되면 다음에 입력해야 할 숫자들을 첫 행에 출력하는 것을 반복한다.
- 게임 결과가 만점이라면 "Full Combo!!!"를 출력해 준 다음에 게임 실행 시간을 출력하기로 했다. 만약 만점이 아니라면 "Clear!!!"를 출력한 후 게임 실행 시간을 출력해 준다.

#### [공통적으로 바뀐 부분]

- 기존에는 LCD 화면에 입력해야 할 게이름을 영문으로 출력하기로 했다. (ex. GGCCAB) 그러나 4x4 키패드에는 각 버튼이 숫자로 표기되어 있고 사용자가 어떤 키를 눌러야 할지 이해하지 못하는 경우를 고려해서 키패드에 표기된 숫자를 그대로 활용하기로 했다. 그래서 현재 LCD 화면의 첫 행에는 키패드로 입력해야 하는 숫자가 출력된다.

### 3. 최종 시스템 구성 및 기능

입력 담당 아두이노(스위치 2개, 가변 저항, 4x4 패드)와 출력 담당 아두이노(16x2 LCD, LED 2개, 소리 부저)로 나뉘 SoftwareSerial을 통해 아두이노 간 유선 통신을 하도록 했다.

#### INPUT.ino

4x4 키패드에서 누른 값을 받아 온다. 15번 키가 눌리면 엔터로 인식하고 가변 저항 값을 읽어 와서

1023/2보다 작은 값이면 점수제 모드로 게임이 실행되도록 OUTPUT.ino에 해당하는 아두이노에게 'S'를 보낸다. 1023/2보다 큰 값이면 타임 어택 모드로 게임이 실행되도록 OUTPUT.ino에 해당하는 아두이노에게 'T'를 보낸다.

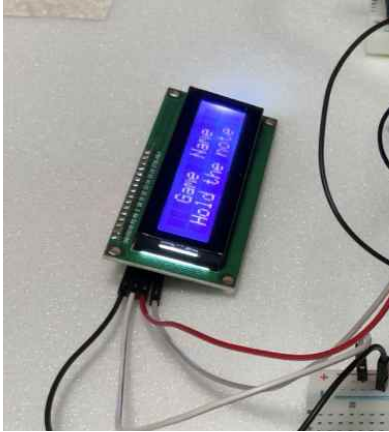
재시작 yes 버튼이 눌렸다면 OUTPUT.ino에 해당하는 아두이노에게 'Y'를 보낸다. 재시작 no 버튼(종료 버튼)이 눌렸다면 OUTPUT.ino에 해당하는 아두이노에게 'N'을 보낸다.

## OUTPUT.ino

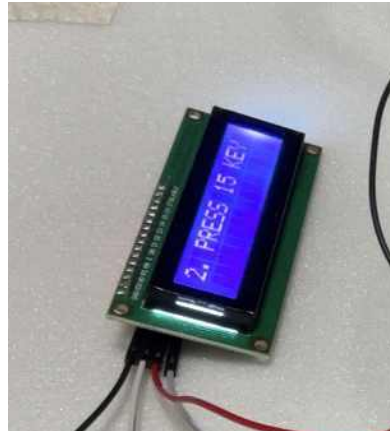
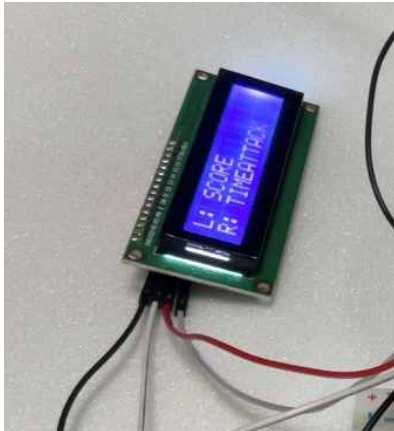
- `firstCommment()`: 게임 실행할 때 초기 화면을 출력해 주기 위한 함수이다.  
LCD 화면 출력 시 부팅음처럼 게임 시작 음이 출력된다.  
게임 제목 출력, 모드 선택 방법, 15번(엔터 기능) 키 입력 순으로 진행된다.  
재시작했을 때는 게임 제목과 모드 선택 방법을 제외하고 모드 선택 및 15번 키 입력만을 출력한다.  
사용자가 가변 저항을 왼쪽으로 돌렸는지 오른쪽으로 돌렸는지에 따라 이 함수에서 각 모드에 해당하는 함수를 실행한다.
- `restartOrShutDown()`: 게임 재시작 여부를 결정하기 위한 함수  
yes 버튼을 누른 경우 초록색 LED를 3초간 켜 다음 게임을 재시작한다.  
no 버튼을 누른 경우 게임이 끝났다는 메시지와 종료음을 출력하고 빨간색 LED를 3초간 켜 다음 모두 끄고 게임을 종료한다.
- `toneOutputByPressingButton(char tmp)`: 버튼별로 음을 다르게 출력하기 위한 함수  
switch문으로 눌러진 버튼을 구별해서 해당하는 음을 출력한다.
- `scoreMode()`: 점수제 모드로 게임 실행을 위한 함수  
만점을 360점으로 설정하고 틀릴 때마다 5점씩 깎는다.  
사용자가 누른 키 값에 따라 소리가 출력되도록 `toneOutputByPressingButton` 함수를 호출한다.  
LCD 첫 행에 눌러야 하는 번호들을 출력한다.  
맞았다면 'O'를 출력하고 틀렸다면 빨간색 LED를 0.1초 동안 켜 후 'X'를 출력한다.  
주어진 행이 다 끝나면 LCD를 지우고 다음에 입력할 숫자들을 첫 행에 출력하는 것을 반복한다.  
게임 결과가 만점이면 "Full Combo!!!!"를 출력해 주고 그렇지 않으면 "Clear!!!"를 출력한다.  
최종 점수를 출력해 주고 재시작 여부를 묻는 `restartOrShutDown` 함수를 호출한다.
- `BeepBeep()`: 타임 어택 모드에서 틀렸을 때 부저음 출력을 위한 함수  
틀리면 부저음을 출력하고 빨간색 LED를 0.1초 동안 켜는 과정을 2번 반복하고 종료한다.
- `timeAttackMode()`: 타임 어택 모드로 게임 실행을 위한 함수  
사용자가 누른 키 값에 따라 소리가 출력되도록 `toneOutputByPressingButton` 함수를 호출한다.  
LCD 첫 행에 눌러야 하는 번호들을 출력한다.  
맞았다면 왼쪽으로 한 칸 스크롤되게 하고, 틀렸다면 `BeepBeep` 함수를 호출한 후 맞을 때까지 키 값을 입력받는다. (틀리면 다시 `BeepBeep()` 호출)  
주어진 행이 다 끝나면 LCD를 지우고 다음에 입력할 숫자들을 첫 행에 출력하는 것을 반복한다.  
게임 결과가 만점이면 "Full Combo!!!!"를 출력해 주고 그렇지 않으면 "Clear!!!"를 출력한다.  
게임 실행 시간을 출력해 주고 재시작 여부를 묻는 `restartOrShutDown` 함수를 호출한다.

## 4. 기능별 동작 화면

### - 초기 화면



재시작 시 출력

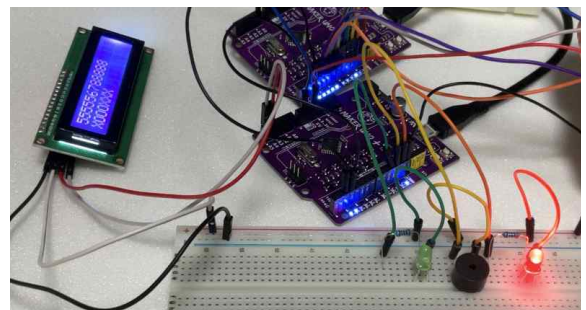


재시작 시 출력

### - 점수제 모드



맞으면 O 출력

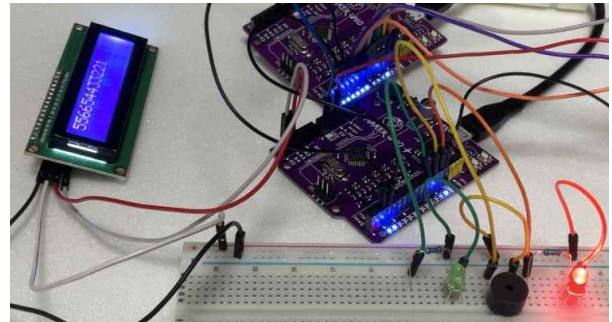


틀리면 X 출력 및 빨간색 LED

- 타임 어택 모드



맞으면 왼쪽으로 스크롤



틀리면 경고음과 빨간색 LED, 안 넘어감

- 결과 화면



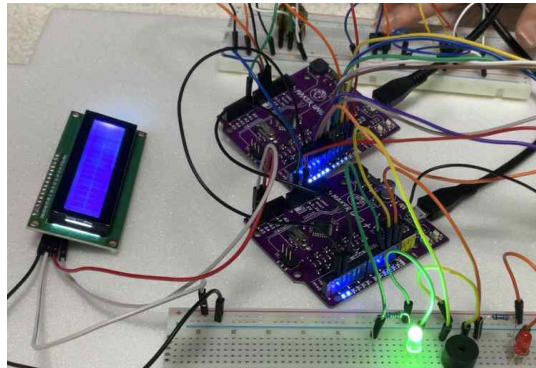
점수제



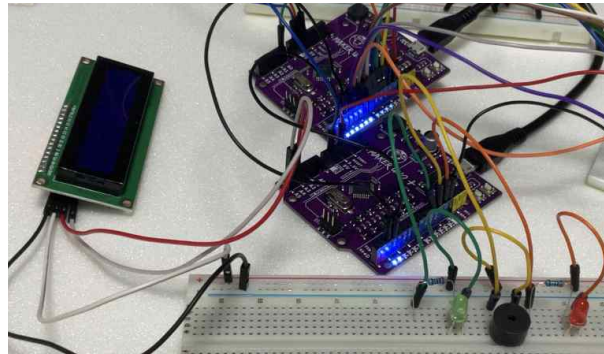
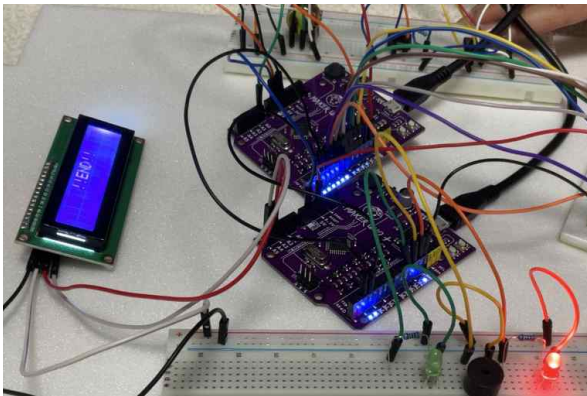
타임 어택



## - 재시작



yes에 해당하는 버튼 누르면 초록색 점등



no에 해당하는 버튼 누르면 종료음과 빨간색 led, 종료 메시지 출력 후 모두 끄고 종료

## 5. 문제점 및 해결 방법

- C언어의 scanf()처럼 입력 대기 기능이 존재하지 않아서 키 값을 읽어오는 데 문제가 있을 거라고 생각했다. 다행히 아두이노 간 통신을 한 덕분에 해결책을 찾게 되었다. `while(Serial.available()==0) {}` // 입력 대기 이 함수를 이용해 입력 대기의 기능을 쓸 수 있어 사용자가 키 값을 입력하기도 전에 문자열이 바로 넘어가는 현상을 해결할 수 있었다.

- 초기 화면 출력 시 15번 키(엔터 기능)를 눌렀음에도 게임 실행 화면으로 넘어가지 않고 초기 화면 출력이 반복되는 문제가 있었다. 그래서 15번 키를 누르라는 메시지 출력 후 LCD를 초기화시키지 않고 키 값이 들어오면 다음 화면으로 넘어갈 수 있게 했다.

- 키패드를 빠르게 눌렀을 때 각 버튼에 해당하는 음이 늦게 출력되는 문제가 있었지만 여러 번의 시도를 통해 적당한 delay 값을 찾아서 곧바로 출력될 수 있게 바꿨다.

- 타임 어택 모드에서 틀렸을 때 빨간색 led가 켜지고 경고음이 출력되는 부분이 사용자가 키 값을 입력했음에도 계속 반복되는 문제가 있었다. 그래서 아예 경고음 출력을 위한 함수를 생성하고 while문과 if문을 사용해서 적절한 조건일 때 함수가 호출될 수 있게 바꿔 문제를 해결했다.

## 6. 최종 구성 요소

No.	Items	Purpose of use
1	빨간색 LED, 초록색 LED	빨간색 LED ON: 틀렸을 때, 종료할 때 초록색 LED ON: 맞았을 때, 재시작할 때
2	소리 부저	키패드 누를 때와 틀렸을 때 음 출력 게임을 처음 시작할 때 시작음 출력 게임 종료할 때 종료음 출력
3	4x4 키패드	숫자 입력 및 엔터 기능을 위한 도구
4	16x2 LCD	첫 행에 눌러야 할 숫자 출력 정답 여부 출력
5	가변 저항	모드 선택
6	버튼 스위치 2개	재시작 여부 선택 (YES, NO)
7		
8		

## 7. 개인별 역할

이름	담당 업무
강서현	4x4 키패드와 16x2 LCD로 리듬 게임 아이디어 제공, INPUT.ino 구현, OUTPUT.ino의 restartOrShutDown(), toneOutputByPressingButton(), scoreMode() 구현, 입력 담당 아두이노 HW 연결, 제안서 및 보고서 초안 작성
박하선	점수제와 타임 어택 모드 아이디어 제공, 타임 어택 모드에서 맞은 숫자를 왼쪽으로 스크롤하는 아이디어 제공, OUTPUT.ino의 firstComment(), timeAttackMode(), BeepBeep() 구현, 출력 담당 아두이노 HW 연결, 제안서 및 보고서 수정 및 마무리