

LED를 이용한 메모리 게임

20151542 이승현

20171251 최진솔

목 차

1. 개발 동기
2. 구상
3. 소스 코드
4. FRIZING 회로도 구상
5. 시연 영상

개발 동기

1. 친구들끼리 간단한 밥 내기를 할 때 사다리 타기 같은 것도 좋지만 우리들이 만든 게임으로 하면 더 재밌지 않을까 하는 취지에서 시작.
2. 이 메모리 게임은 **LED**, 버튼, 피에조 부저로 만들 수 있는 간단한 게임이지만 단계가 올라갈수록 어려워지며 여러가지 부품을 쓰는것보단 몇 개의 부품으로 응용을 잘하는게 더 중요하다고 생각하여 이 게임을 만들게 됨.

구상

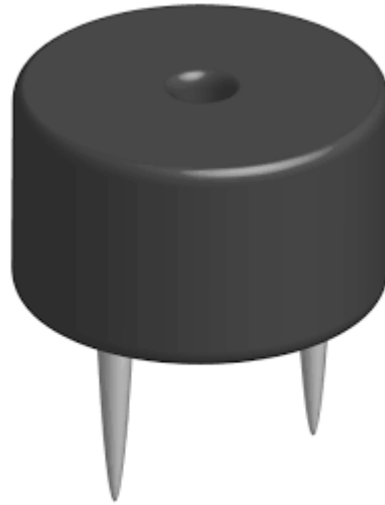
2주차 LED



6주차 스위치



10주차 피에조 부저



+ C 배열, 난수생성

구 상

1. 각각의 스위치에 **LED**와 부저를 연동하여 스위치를 누르면 **LED**가 점등하고 부저에서 그 스위치에 맞는 톤이 출력되게 설정함.
2. 난수를 하나 불러와 **4**개의 **LED**중 하나를 점등시킴.
그 **LED**에 대응되는 스위치를 누르면 다음단계로 넘어가고 동시에 난수를 하나 더 불러와 차례대로 스위치를 눌러주면 되는 메모리 게임
3. 실패 하였을 경우 **LED**가 모두 **1**번 점등된 후 다시 처음 **1**단계부터 시작하게 됨.

코드

```
const int button1 = 2;
const int button2 = 3;
const int button3 = 4;
const int button4 = 5;
const int led1 = 7;
const int led2 = 8;
const int led3 = 9;
const int led4 = 10;
const int buzzer = 12;
const int tones[] = {1915, 1700, 1519, 1432, 2700};

int buttonState[] = {0, 0, 0, 0};
int lastButtonState[] = {0, 0, 0, 0};
int buttonPushCounter[] = {0, 0, 0, 0};

void playTone(int tone, int duration) {
  for (long i = 0; i < duration * 1000L; i += tone * 2) {
    digitalWrite(buzzer, HIGH);
    delayMicroseconds(tone);
    digitalWrite(buzzer, LOW);
    delayMicroseconds(tone);
  }
}

void setup() {

  randomSeed(analogRead(0));

  pinMode(button1, INPUT);
  pinMode(button2, INPUT);
  pinMode(button3, INPUT);
  pinMode(button4, INPUT);
```

```
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
  pinMode(led4, OUTPUT);
  pinMode(buzzer, OUTPUT);
}
```

```
int game_on = 0;
int wait = 0;
int currentlevel = 1;
long rand_num = 0;
int rando = 0;
int butwait = 500;
int ledtime = 500;
int n_levels = 30;
int pinandtone = 0;
int right = 0; //다음
int speedfactor = 5;
int leddelay = 200;
```

```
void loop() {
```

```
  int n_array[n_levels];
  int u_array[n_levels];
```

```
  int i;
```

```
  if (game_on == 0)
    for (i = 0; i < n_levels; i = i + 1)
      n_array[i] = 0;
      u_array[i] = 0;
      rand_num = random(1, 200);
      if (rand_num <= 50)
        rando = 0;
      else if (rand_num > 50 && rand_num <= 100)
        rando = 1;
      else if (rand_num > 100 && rand_num <= 150)
        rando = 2;
      else if (rand_num <= 200)
        rando = 3;
```

코드

```
n_array[i] = rand();
}

game_on = 1;
}

// 게임 진행
if (wait == 0) {
    delay (200);
    i = 0;
    for (i = 0; i < currentlevel; i = i + 1) {

        leddelay = ledtime / (1 + (speedfactor / n_levels) * (currentlevel - 1));
        pinandtone = n_array[i];
        digitalWrite(pinandtone + 7, HIGH);
        playTone(tones[pinandtone], leddelay);
        digitalWrite(pinandtone + 7, LOW);
        delay(100 / speedfactor);
    }
    wait = 1;
}
i = 0;
int buttonchange = 0;
int j = 0;
while (j < currentlevel) {
    while (buttonchange == 0) {
        for (i = 0; i < 4; i = i + 1) {

            buttonState[i] = digitalRead(i + 2);
            buttonchange = buttonchange + buttonState[i];
        }
    }
    for (i = 0; i < 4; i = i + 1) {
        if (buttonState[i] == HIGH) {
            digitalWrite(i + 7, HIGH);
            playTone(tones[i], ledtime);
            digitalWrite(i + 7, LOW);
            wait = 0;
            u_array[j] = i;
            buttonState[i] = LOW;
            buttonchange = 0;
        }
    }
}
```

```
if (u_array[i] == n_array[i]) {
    j++;
    right = 1;
}

else {
    right = 0;
    i = 4;
    j = currentlevel;
    wait = 0;
}

if (right == 0) {
    delay(300);
    i = 0;
    game_on = 0;
    currentlevel = 1;

    for (i = 0; i < 4; i = i + 1) {
        digitalWrite(i + 7, HIGH);
    }
    playTone(tones[4], ledtime);
    for (i = 0; i < 4; i = i + 1) {
        digitalWrite(i + 7, LOW);
    }
    delay (200);
    for (i = 0; i < 4; i = i + 1) {
        digitalWrite(i + 7, HIGH);
    }
    playTone(tones[4], ledtime);
    for (i = 0; i < 4; i = i + 1) {
        digitalWrite(i + 7, LOW);
    }

    delay(500);
    game_on = 0;
}
```

코드

```
if (right == 1) {
    currentlevel++;
    wait = 0;
}

if (currentlevel == n_levels) {
    delay(500);

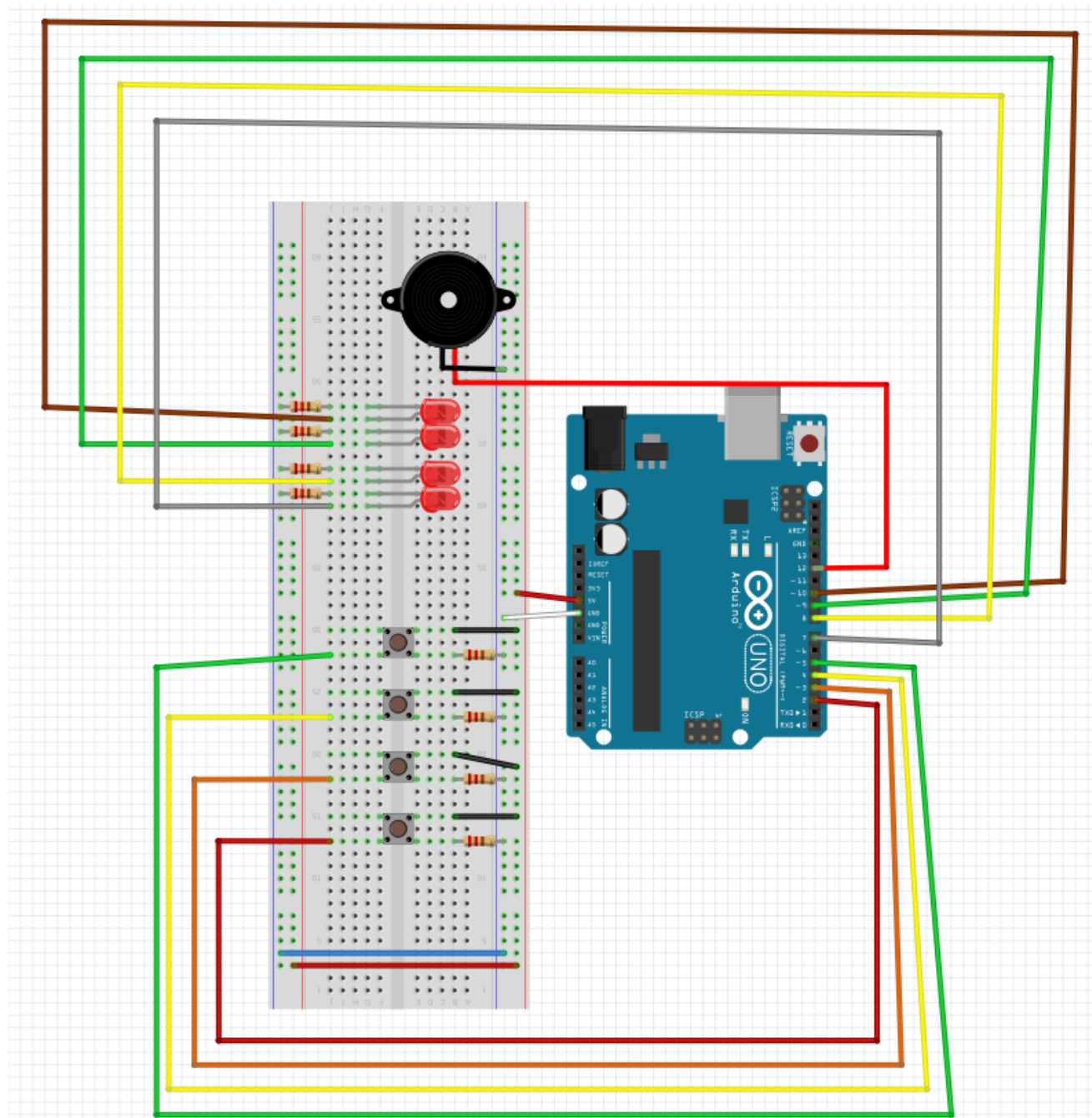
    int notes[] = {2, 2, 2, 2, 0, 1, 2, 1, 2};
    int note = 0;
    int tempo[] = {200, 200, 200, 400, 400, 400, 200, 200, 600};
    int breaks[] = {100, 100, 100, 200, 200, 200, 300, 100, 200};

    for (i = 0; i < 9; i = i + 1) {
        note = notes[i];
        digitalWrite(note + 7, HIGH);
        playTone(tones[note], tempo[i]);
        digitalWrite(note + 7, LOW);
        delay(breaks[i]);
    }

    game_on = 0;
    currentlevel = 1;
    n_levels = n_levels + 2;
    speedfactor = speedfactor + 1;
}
}
```


FRIZING

회로도



시연 영상

