

# REPORT

---



과목명		임베디드컴퓨팅
담당교수		김영필 교수님
학과		정보통신공학과
학년		4
학번		201901723
이름		최가은
제출일		2022.11.01.(화)

# 1. Introduction

## - Devel. Environment

Development	Status
OS	MAC OS
CPU	2.3 GHz 8코어 Intel Core i9, AMD Radeon Pro 5500M 4GB & Intel UHD Graphics 630 1536MB
RAM	16GB 2667MHz DDR4
Arduino Board Type	Uno R3
Arduino IDE Version	1.8.19

## - Devel.Plans

이전 과제 중 PlayMelody를 이용하여 개발을 진행하려고 했습니다.  
간단하게 Piezo Speaker를 사용하여 음계를 출력해보는 과제였는데 여기에  
수업시간에 배운 LED와 C언어 기초를 응용하여 해결하는 과제가 될 것이라  
고 생각했습니다.

Midterm Assignment 내용 중 Display Information 부분에서 current Sound Name  
출력을 어떻게 할 수 있을지 고민하던 중에 String 배열을 이용하여 간단하  
게 출력하는 방법을 생각해냈습니다.

또한 이러한 변수들을 이용하여 조건문을 통해 LED를 켜고 끄고 하면 되  
겠다고 생각했고 그대로 진행했습니다.

## - Status

구현 완료

## 2. Prototype Explanation

### 1. String NoteOfRabbits[]

// Current Sound Name 출력을 위해 새로 선언한 문자열  
`String NoteOfRabbits[] = {"SOL", "MI", "MI", "SOL", "MI", "DO", "RE", "MI", "RE", "DO", "MI", "SOL"};`

Current Sound Name을 출력하기 위해 String형으로 배열을 선언했습니다.

제가 출력할 음의 이름을 문자열로 저장한 후 for문을 사용하여 이를 하나씩 출력합니다.

### 2. void turnOnOffLed(int b, int fn){...}

```
// LED ON/OFF를 위한 함수 작성 -> 중복을 줄이기 위해 선언
void turnOnOffLed(int b, int fn){
    if(fn >= 392 && fn % 2 == 0){
        if(b >= 1000){
            Serial.println("RG");
            digitalWrite(dPin1, HIGH);
            digitalWrite(dPin2, HIGH);
            delay(b);
            digitalWrite(dPin1, LOW);
            digitalWrite(dPin2, LOW);
            delay(b);
        } else {
            Serial.println("G");
            digitalWrite(dPin1, HIGH);
            delay(b);
            digitalWrite(dPin1, LOW);
            delay(b);
        }
    } else {
        Serial.println("R");
        digitalWrite(dPin2, HIGH);
        delay(b);
        digitalWrite(dPin2, LOW);
        delay(b);
    }
}
```

// b = beats, fn = FrequencyNumber  
// Green & Red LED 조건 -> Green 조건을 만족하면서 1000ms 이상이면  
// 켜지는 LED 이름 출력  
// Green LED 조건 -> 제일 큰 if문만 만족하고 1000ms가 넘지 않으면  
// Red LED 조건  
// 켜지는 LED 이름 출력

Midterm Assignment에서는 first 12 notes, second 12 notes, After second 12 notes로 총 3가지의 경우가 있는데 각 경우마다 LED를 켜고 끄는 조건문이 있어야 합니다. 이에 코드의 중복을 줄이기 위해 함수를 따로 작성했습니다. 이를 사용함으로써 코드의 간결함을 확인해 볼 수 있습니다.

매개변수로 b(beats[], 음 지속 시간)과 fn(frequency number, 음계의 숫자)을 받아온 후 조건문을 작성했습니다.

fn이 even이면서 'FA'보다 높은 경우를 먼저 조건문으로 판별한 후 그 안에서 두 개의 LED가 동시에 켜지는 조건인 sound length가 1000보다 높은 경우를 조건문으로 작성했습니다. Sound length가 1000보다 높지 않으면 초록색 LED만 켜지도록 작성했고, 가장 바깥 조건문인 even과 FA 이상의 조건을 만족하지 않으면 무조건 빨간색 LED만 켜지도록 코드를 작성했습니다.

### 3. void displayInfo(int iter){...}

```
void displayInfo(int iter){
    Serial.print("[Iteration Number : ");
    Serial.print(12*cnt + iter);                // Iteration Number 출력
    Serial.print("]   Current Sound Name : ");
    Serial.print(NoteOfRabbits[iter]);           // Current Sound Name 출력
    Serial.print(",   Current Sound Frequency : ");
    Serial.print(melody[Rabbits[iter]]);         // Sound Frequency 출력
    Serial.print(",   Turned-on LED Name : ");
}
```

Iteration Number, Current Sound Name, Sound Frequency, Turned-on LED name을 경우에 맞게 출력해야 하므로 이 또한 중복을 제거하기 위해 함수로 따로 작성하였습니다.

Iteration Number는 global 변수 count에 Note의 개수인 12를 곱한 후 for문에서 나온 i를 매개변수로 받아와 더함으로써 출력합니다.

### 4. void loop(){...}

```
void loop() {
    while(true){
        if(cnt % 3 == 0){                // first 12 notes
            for(int i = 0 ; i < 12 ; i++){
                displayInfo(i);
                tone(dPin3, melody[Rabbits[i]], beats[i]); // Piezo Speaker 출력

                turnOnOffLed(beats[i], melody[Rabbits[i]]);
            }
        } else if(cnt % 3 == 1){        // second 12 notes
            for(int i = 0 ; i < 12 ; i++){
                displayInfo(i);
                tone(dPin3, melody[Rabbits[i]], (int)beats[i]/2);

                turnOnOffLed((int)beats[i]/2, melody[Rabbits[i]]);
            }
        } else {                        // play the melody silently
            for(int i = 0 ; i < 12 ; i++){
                displayInfo(i);

                turnOnOffLed(beats[i], melody[Rabbits[i]]);
            }
        }
        cnt++;
    }
}
```

무한반복을 위해서 while(true)를 사용하여 반복문을 작성합니다.

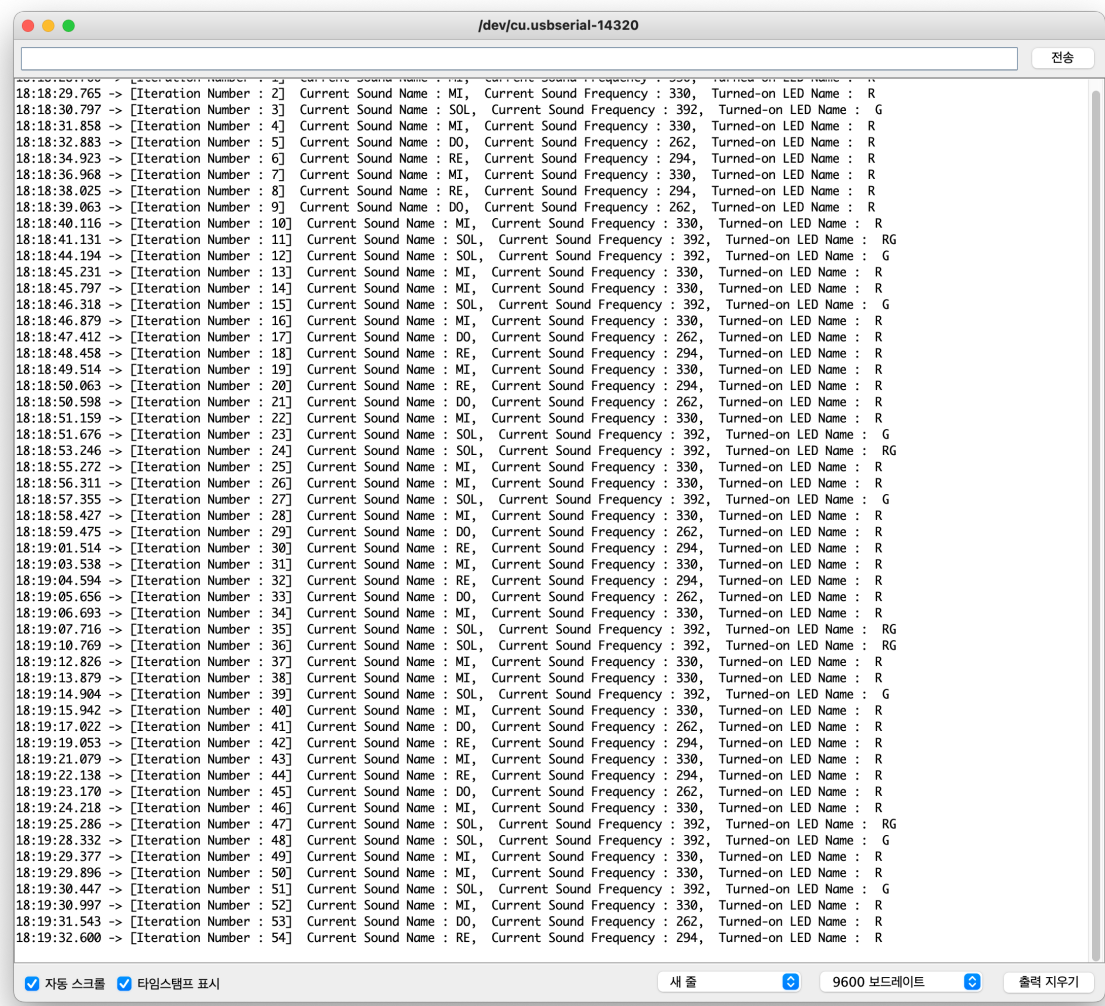
Global variable으로 cnt를 선언하였고 이를 사용하여 first 12 notes, second 12 notes, After second 12 notes를 구분합니다. 전역 변수인 cnt를 3으로 나누었을때 나머지 값을 사용하여 나머지가 0, 1, 2일때에 맞게 조건문을 작성했습니다.

First 12 notes의 경우 displayInfo()를 사용하여 시리얼 모니터에 Information을 출력을 하고, tone()을 사용하여 Piezo Speaker에 음을 출력합니다. 후에 만들어 둔 turnOnOffLed()를 사용하여 경우에 맞는 LED를 출력했습니다.

Second 12 notes의 경우에도 위의 경우와 같이 displayInfo()를 사용한 후 tone()을 사용합니다. 하지만 이 때는 sound length를 절반으로 줄여야하기 때문에 (int)beats[i] / 2를 사용하여 절반으로 줄여주었고, turnOnOffLed에 인자값으로 넘겨줄때도 절반의 소리 길이를 전달해 줍니다.

마지막 경우에는 tone을 사용하지 않고 displayInfo()와 turnOnOffLed()만을 사용합니다.

### 3. Results



제가 작성한 코드를 실행했을 때의 시리얼 모니터입니다.

Iteration Number를 출력하고 Note, Sound Frequency, Turned-On LED를 차례로 출력합니다.

Frequency Number	392	330	330	392	330	262	294	330	294	262	330	392
Sound Name	SOL	MI	MI	SOL	MI	DO	RE	MI	RE	DO	MI	SOL
First Sound Length	1000	500	500	500	500	1000	1000	500	500	500	500	1500
Second Sound Length	500	250	250	250	250	500	500	250	250	250	250	750
First 12 notes	RG	R	R	G	R	R	R	R	R	R	R	RG
Second 12 notes	G	R	R	G	R	R	R	R	R	R	R	G

Green LED가 켜지는 조건을 초록색으로 표시했고, 여기에 추가로 두 개의 LED가 켜져야 하는 추가 조건을 빨간색으로 표시했습니다.

## 4. Lessons

### 1. What was good or difficult

이전에 진행한 과제물에 추가적인 부분을 덧붙여 중간과제를 진행하는 것이 재미있었습니다. 또한, 코드 중복을 확인하고 코드를 최대한 깔끔하게 정리하기 위해 노력하였는데, 제가 직접 확인하기에도 코드의 간결함을 확인할 수 있어 굉장히 뿌듯했습니다.

기본적인 c 문법은 잘 알고 있어서 과제를 진행함에 있어 어려움은 없었으나, 어떻게 하면 더 쉽고 깔끔하게 작성할 수 있을지 고민하는 시간이 제게는 어려움이었다고 생각됩니다.

또한, pdf 자료를 보고 하는데 해석의 어려움이 있어 TA분께 조건이 맞는지 확인을 하는 과정에서 착오가 많아 코드 수정을 하루 사이에 여섯번을 했습니다. 다행히 함수로 따로 제작해 두었기 때문에 수정에 어려움이 많지는 않았지만 조금 귀찮은 과정을 겪었다고 생각합니다.

추가로, Red, Green LED가 동시에 점등될 때 약간의 딜레이가 나오는 것을 확인했는데 이 부분에 대해서는 없애는 방법을 찾지 못해 해결하지 못했습니다. 해결해보기 위해서 구글링을 진행해보았지만, 결국 해결하지 못한 점이 너무 아쉽습니다.

### 2. What you learned

코드의 간결함의 중요성을 다시한번 배웠습니다.

중복 제거를 통해서 코드의 가독성을 높여 깔끔하게 정리하는 방법을 확인했습니다.

추가로, 코드를 작성하는 부분에 대해선 c 언어 문법의 전체적인 부분과 Digital Output인 LED, Piezo Speaker를 다루는 방법을 확인할 수 있었습니다.