

Wave player 요구조건

- 가정 : SD card내의 root directory에 wave file 만을 위치함. root directory 아래에는 file 만 있고 다른 directory는 없도록 할 것
 - Rotary encoder를 회전시켜 play 할 곡을 변경할 수 있도록 INT3를 활용
 - Rotary encoder의 push button을 눌러서 play/stop을 선택할 수 있도록 INT1 활용
 - 음악이 play 되는 동안 rotary encoder의 회전을 통한 곡 변경은 비활성화
 - 음악이 정지되어 있는 동안만 rotary encoder를 통한 선곡 가능
 - LCD 첫번째 줄 : 곡명 표시, LCD 두번째 줄 : STOPPED/PLAYING 표시
 - Stereo/mono, 16-bit, 8-bit, 44.1 KHz, 22.05Hz 등 다른 정보를 가진 곡의 연주 가능
[주] 최소한의 경우 16-bit stereo 44.1 KHz의 음악을 play 할 수 있어야 함.
 - 8-bit PWM은 Timer/Counter 2와 Timer/Counter 4를 이용하여 생성
 - 44.1 KHz 또는 22.05 KHz로 PWM을 갱신하는 것은 Timer/Counter 1의 timer interrupt를 활용
- [주] 음악 data가 소리가 큰 data를 가질 경우 USB로 speaker를 구동할 경우 충분한 전류를 제공못함. wave file 로 export 할 때 소리를 좀 줄여서 export 해볼 것.
- [주] Microcontroller에서 음악소리를 줄이려고 할 때는 16-bit data에 right shift 연산을 사용해 볼 수 있음. 하지만 시간이 많이 소모되 실시간 성질을 깨트릴 수 있으니 주의할 것

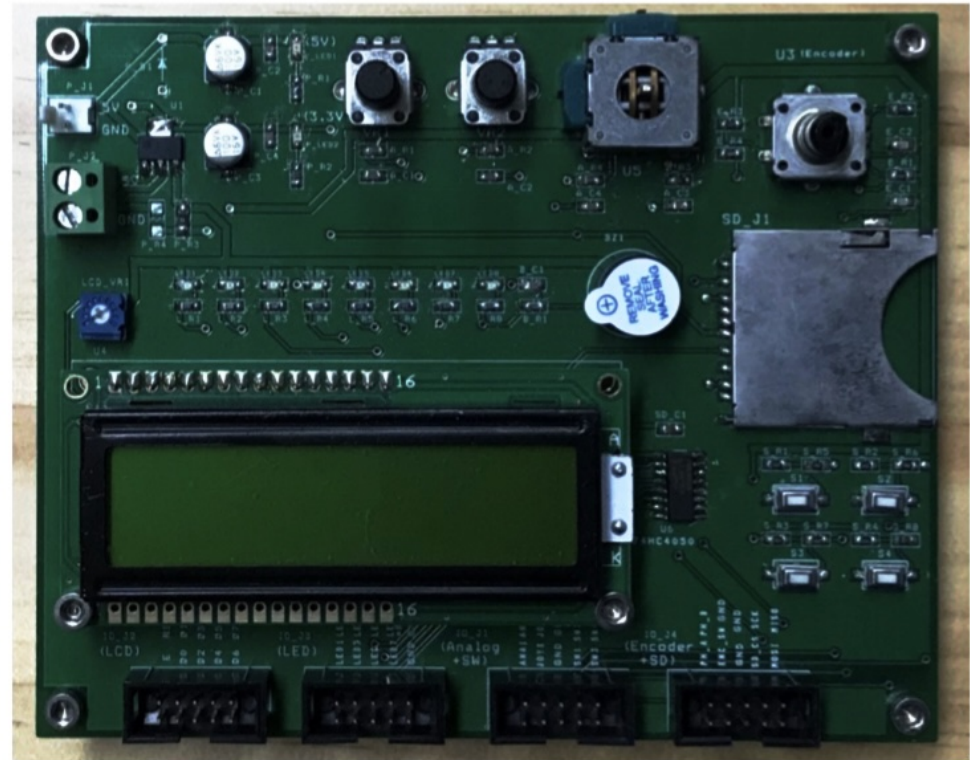
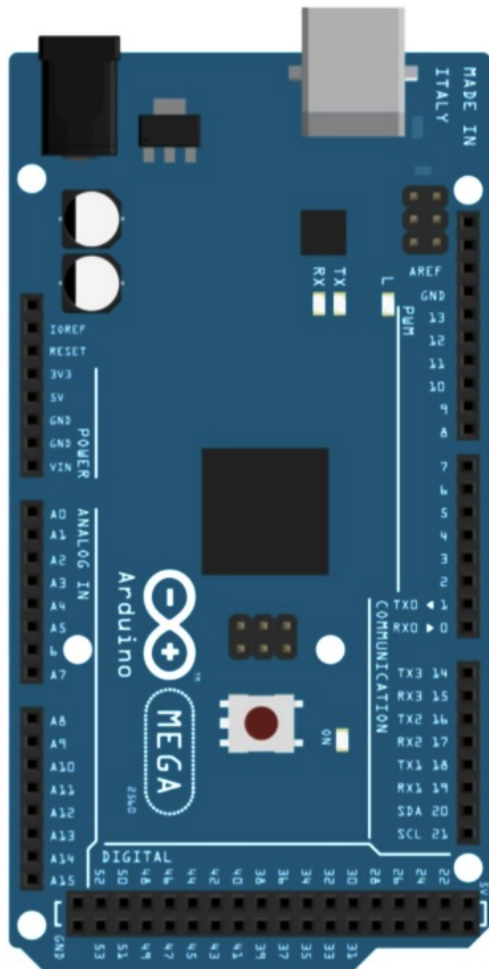
결선 정보 (1)

| Arduino Pin | Peripheral Board | |
|----------------|------------------|----------------|
| 12 | RS | LCD |
| 11 | E | |
| 5 | D4 | |
| 4 | D5 | |
| 3 | D6 | |
| 2 | D7 | |
| 18(PD3, INT3) | PH_A | Rotary Encoder |
| 19 (PD2) | PH_B | |
| 20 (PD1, INT1) | ENC_SW | |
| 50 (PB3, MISO) | MISO | SPI |
| 51(PB2, MOSI) | MOSI | |
| 52 (PB1, SCK) | SCK | |
| 53 (PB0, SS) | SD_CS | |

결선 정보 (2)

| Arduino Pin | Amp Board | |
|----------------|---------------|--------|
| 5V | 5V | Amp 연결 |
| GND | GND | |
| 10 (PB4, OC2A) | PWM_H (LEFT) | |
| 9 (PH6, OC2B) | PWM_L (LEFT) | |
| 6 (PH3, OC4A) | PWM_H (RIGHT) | |
| 7 (PH4, OC4B) | PWM_L (RIGHT) | |

| Amp Board | Speaker | |
|---------------|------------|------------|
| Left speaker | Speaker #1 | Speaker 연결 |
| Right speaker | Speaker #2 | |



개발 방법 및 Tip

- Divide and conquer

- 기능들을 분할 해서 각각의 기능들을 먼저 구현해 볼 것.
(Ex) Rotary encoder를 이용한 interface (음악 연주와 별개로 구현)
(Ex) SD card에서 file 이름 알아내고, wave file일 경우 configuration 정보 알아내기.
(Ex) 이름을 알고 있는 특정 wave file (stereo, 16-bit, 44.1 KHz 기준)을 play

- 합치기

- 각각의 기능들을 합쳐서 전체 application이 동작되도록 세부 작업 수행

- SD card로부터 data를 대량으로 받아와 buffer에 쌓아둘 것.

- buffer는 1개가 아니라 2개 사용.
- buffer 하나의 size : 3000 개.
- buffer의 data type은 unsigned char.
- 즉, `unsinged char buf[2][3000];` 으로 선언