

## 팀 프로젝트 결과 보고서

6조

정보컴퓨터공학부  
김무섭(201224419)  
정수희(201324518)  
쉬상(201424477)

## 목 차

1. 설계 목표	p.3
2. 설계 시스템의 흐름도	p.6
3. 설계 시스템의 구성도에 대한 블록 다이어그램	p.8
4. 설계 시스템의 부품 관련 설명	p.11
5. 구현 결과 (기능 설명 등)	p.12

## 1. 설계 목표 및 배경이론

- 1) 설계 목표 : ARM 기반 Cortex-M3 보드를 활용하여 ‘압력을 이용한 디지털 악기’ 시스템을 설계 및 개발한다. 컵을 두드리면 컵에 들어있는 음료의 양에 따라 서로 다른 음이 보드에서 흘러나온다.
- 2) 애플리케이션 설명: 본 애플리케이션은 일종의 전자악기로써 컵에 들어 있는 물의 양에 따라 각 컵에 다른 음이 할당되어 연주할 수 있다. 부가적인 기능으로는 녹음기능이 있어, 자신이 연주한 곡을 녹음하고 재생시켜 들을 수 있다.
- 3) 배경이론
  - 타이머
    - ① 타이머(timer)는 특별한 종류의 시계로 일련의 사건이나 프로세스를 제어하는 데 사용할 수 있다. 스톱워치가 0부터 숫자를 올려 세는 것과 달리 타이머는 모래시계처럼 특정한 시간 간격으로부터 숫자를 내려 센다. 타이머는 기계적, 전자기계적, 전기적, 소프트웨어적인 방식을 취하기도 하며 현대의 모든 컴퓨터들은 하나 이상의 디지털 타이머를 포함하기도 한다.
    - ② 컴퓨터에서 가장 필수적인 모듈로 사용한다. 소프트웨어로 정확한 시간을 측정하는 것은 프로그램이 복잡해지면 거의 불가능해진다. 보통의 컴퓨터 시스템에서는 여러 가지 일이 복합적으로 이뤄지기 때문에 하드웨어에 의한 시간 회로가 필요하다. 따라서 거의 모든 컴퓨터에서 사용되며, 임베디드 시스템에서도 필수적이다.
    - ③ 타이머는 over 혹은 down flow를 trigger로 하여 인터럽트 혹은 이벤트가 발생한다. 이를 이해하기 위해서는 타이머의 블록 다이어그램(그림1)을 살펴보아야한다. 초록색으로 표시한 Internal Clock는 Clock Tree에서 TIMxCLK에 해당한다. 이 클럭은 별다른 중간 작업이 없다면, CK\_PSC로 전달된다. 이후 PSC를 거쳐 CK\_INT로 가공되고 이는 Counter의 CLK가 된다. Counter의 값은 Capture/Compare register로 전달되고, 레지스터에서 over 혹은 under flow가 발생하여 이벤트 혹은 인터럽트를 발생시킨다.

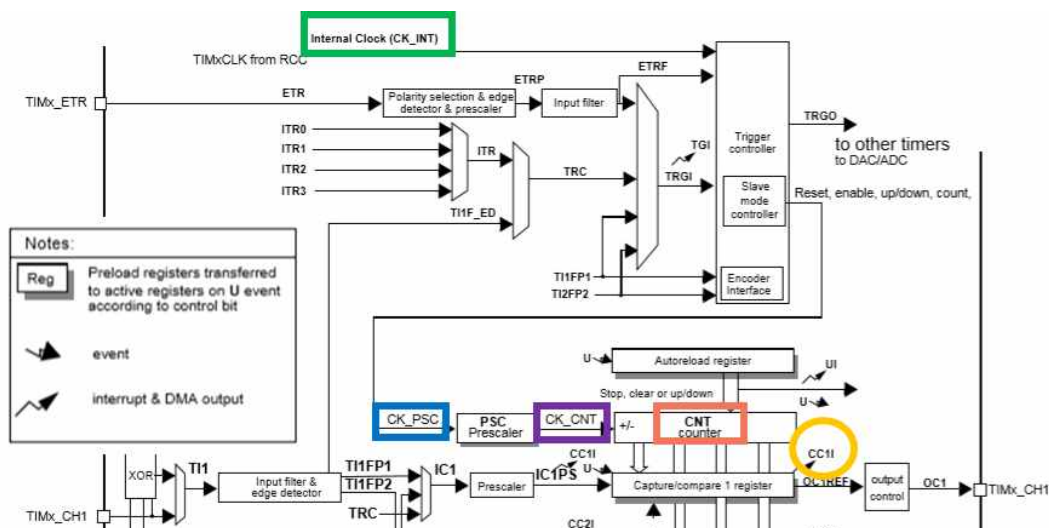


그림 1 General-purpose timer block diagram

- 인터럽트

- ① 폴링(Polling) 방식은 장치(또는 프로그램)의 상태를 주기적으로 검사하여 일정한 조건을 만족할 때 해당 처리 루틴을 실행해 처리하는 방식을 말한다. 이와는 대비되는 개념으로 인터럽트(Interrupt) 방식은 CPU가 프로그램을 실행하고 있을 때, 입출력 하드웨어 등의 외부 장치에서 들어오는 이벤트 또는 예외상황이 발생하여 처리가 필요한 경우에 CPU에게 알려 처리할 수 있도록 하는 방식이다.
- ② CPU 하드웨어는 인터럽트 요청 라인(Interrupt request line)이라고 불리는 선을 하나 갖는데 CPU는 매 명령어의 실행(Execute)을 끝내고 다음 명령어를 수행하기 전에 인터럽트 요청 라인을 검사한 후 해당 작업을 수행한다.(그림2)

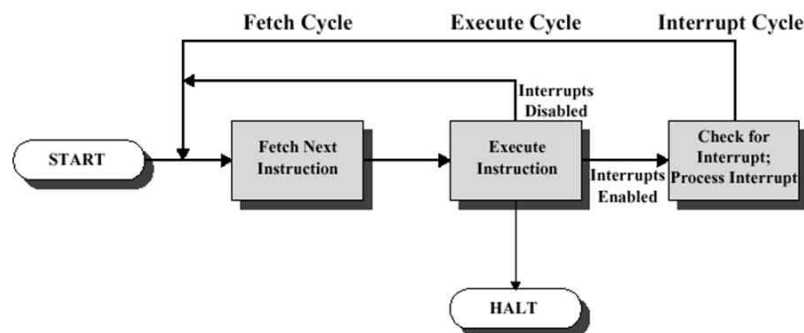


그림 2. Instruction Cycle with Interrupts

- 입/출력 하드웨어 제어가 요청 라인에 신호를 보낸다.
- CPU는 각종 레지스터 값과 상태 정보를 저장한 다음 메모리상의 인터럽트 핸들러 루틴으로 이동 한다.
- 인터럽트 핸들러는 인터럽트의 발생 원인을 조사하고 필요한 작업을 수행한 후 CPU를 인터럽트 이전 상태로 복구하여 인터럽트 당한 프로그램 중단된 곳에서부터 다시 수행한다.

- ADC(Analog to Digital Converter)

- ① ADC는 아날로그 전기 신호를 디지털 전기 신호로 변화하는 전자 회로이다.
- ② 아날로그 신호를 디지털 신호로 변화하는 ADC의 동작을 표본화(Sampling)이라고 한다.

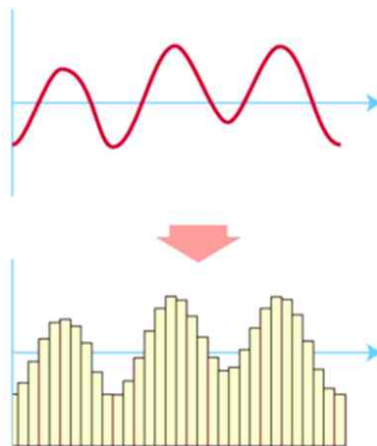


그림 3 ADC의 개념

- ③ 아날로그 신호는 변질, 손실되기 쉽고 압축과정이 없어 전파에 담을 수 있는 정보의 양이 한정

적이다. 그에 반해 디지털 신호는 아날로그 신호보다 외부의 노이즈에 강하고 온도나 기타 외부 요소에 의한 영향을 덜 받으며, 저장, 편집, 가공, 전송 등에서 아날로그 신호에 비해 편리하다.

- DMA

- ① DMA(Direct Memory Access)는 기존의 PIO(Programmed Input/Output) 방식(장치들 사이에 전송되는 모든 데이터가 CPU를 거쳐가는 방식)과는 다르게 주변장치들(하드디스크, 그래픽 카드, 네트워크 카드, 사운드 카드 등)이 메모리에 직접 접근하여 읽거나 쓸 수 있도록 하는 기능이다. DMA가 지원되면 CPU가 데이터 전송에 관여하지 않아 다른 연산을 위해 사용될 수 있고, 고속으로 데이터를 전송하여 높은 성능을 얻을 수 있다.
- ② 주변장치의 데이터의 양이 많은 경우, 데이터의 이동으로 인한 부담이 커지는데 이러한 문제를 해결하기 위해 DMA를 이용한다.
- ③ STM32에서의 DMA
  - 두 개의 DMA controller는 DMA1은 7개, DMA2는 5개로 총 12개의 채널을 갖는다. (DMA2 controller는 오직 high-density, XL-density 디바이스에서만 이용 가능하다)
  - 12개의 채널 각각은 해당하는 DMA 요청에 연결된다.
  - 각각의 채널에 해당하는 요청 사이에는 우선순위를 둘 수 있다. (very high, high, medium, low)
  - Circular mode를 지원하여 지속적인 정보를 받을 수 있다.
  - 주변장치에서 메모리로, 메모리에서 주변장치로, 또는 주변장치에서 주변장치로 데이터 전송이 가능하다. (Flash, SRAM, APB1, APB2, 그리고 AHB 주변장치에서 데이터를 전송하거나 수신할 수 있다)
- ④ ADC-DMA
  - STM32 보드에는 ADC1, ADC2 두 개의 ADC(analog to digital convert)가 있는데, ADC의 채널을 사용하면 ADC1 하나로 두 개 이상의 아날로그 데이터를 받을 수 있다.
  - 보통 ADC1에서 아날로그 데이터를 디지털 데이터로 변환한 후, ADC1->DR 레지스터에 값을 저장하는데, 두 개 이상의 아날로그 입력을 동시에 포트로부터 받게 될 경우, ADC1->DR 정보가 덮어쓰워지기 때문에 문제가 될 수 있는데, 이때 DMA를 이용하여 문제를 해결할 수 있다.
  - 포트로 아날로그 정보를 받은 후, ADC1을 거쳐 변환된 데이터를 DMA를 이용하여 지정된 메모리 주소에 저장하게 할 수 있다. 이러한 동작은 해당 메모리를 ADC 채널에 연결시킴으로써 할 수 있다.

## 2. 설계 시스템의 흐름도(Flow chart)

### 1) 시스템 흐름도

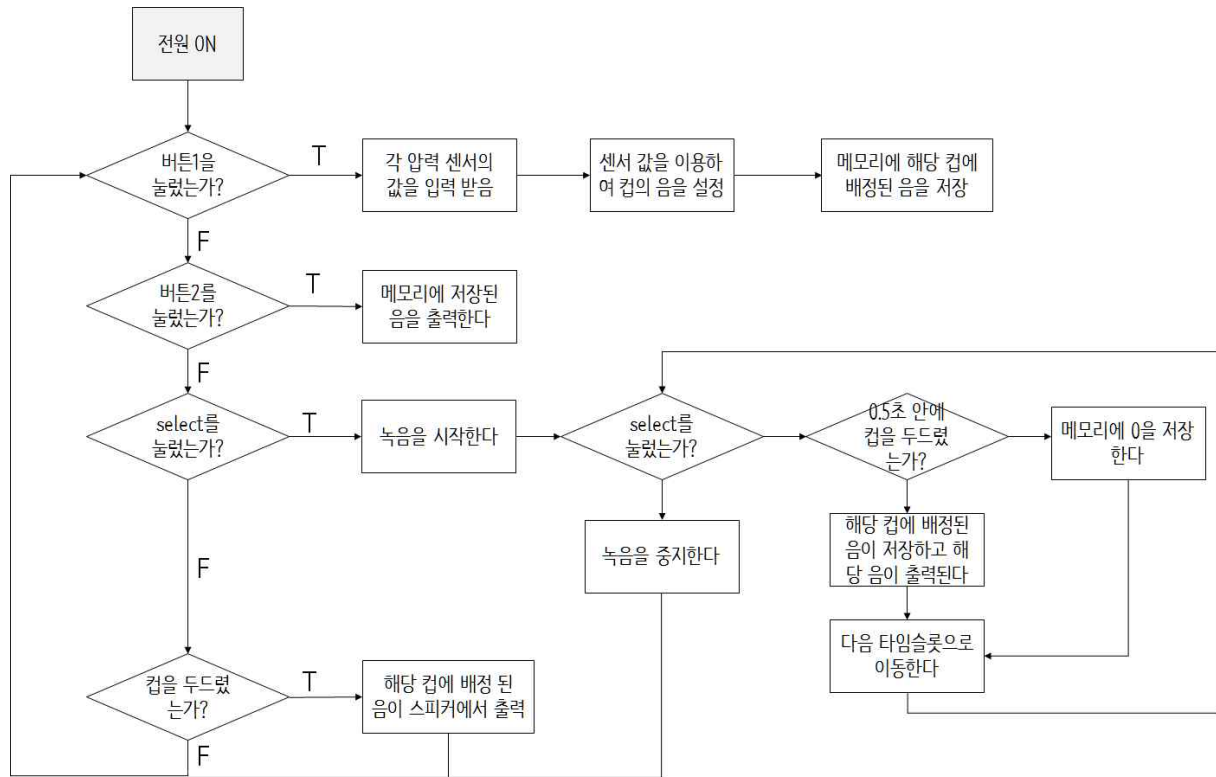


그림 4 전체 흐름도

### 2) 시나리오

- 기능1 : 물의 양에 따라 각 컵에 음을 할당한다.
  - ① 버튼1을 누른다.
  - ② 각 압력 센서로 부터 값을 입력 받는다.
  - ③ 메모리에 저장되어 있는 압력 정보와 음 배정 정보를 기준으로 하여 각 컵에 음을 배정하고 해당 맵핑 정보를 메모리에 저장한다.
- 기능2 : 컵을 두드리면 음을 출력한다.
  - ① 컵을 두드리면 컵에 부탁 된 진동센서에서 두드림을 감지한다.
  - ② 두드려진 컵에 배정 된 음을 출력한다.
- 기능3 : 녹음
  - ① joystick의 select를 누른다.
  - ② 0.5초 간격으로 메모리의 타임 슬롯에 값을 저장한다. 만약 0.5초 안에 컵을 두드리면 해당 컵에 배정된 음이 저장되고 만약 두드려지지 않았다면 0이 저장된다.
  - ③ 0.5초가 지나면 타임 슬롯을 한 칸 이동시킨다.
  - ④ joystick의 select를 누르면 녹음이 중지 된다.

- 기능4 : 재생
  - ① 버튼2를 누른다.
  - ② 저장 된 음들을 출력한다.
  - ③ 출력이 모두 완료 되거나 버튼2를 한 번 더 누르면 재생이 종료된다.

### 3. 설계 시스템의 구성도에 대한 블록 다이어그램과 상세 기능

#### 1) 블록 다이어그램1 - 전체 시스템

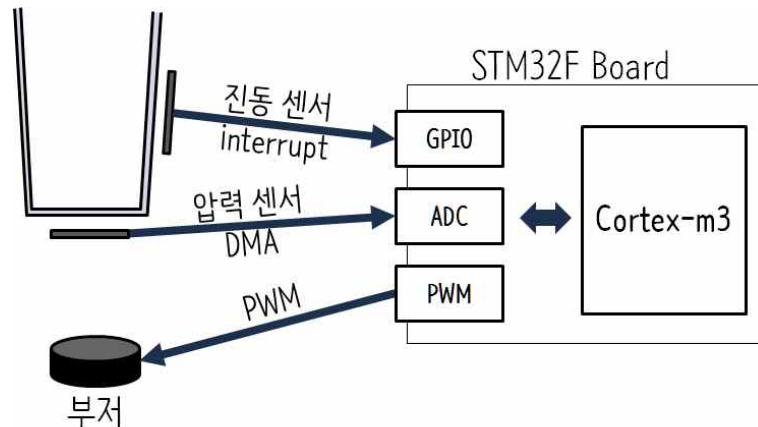


그림 5 전체 시스템에 대한 블록 다이어그램

#### 2) 블록 다이어그램2 - 기능1 (물의 양에 따라 각 컵에 음을 할당한다)

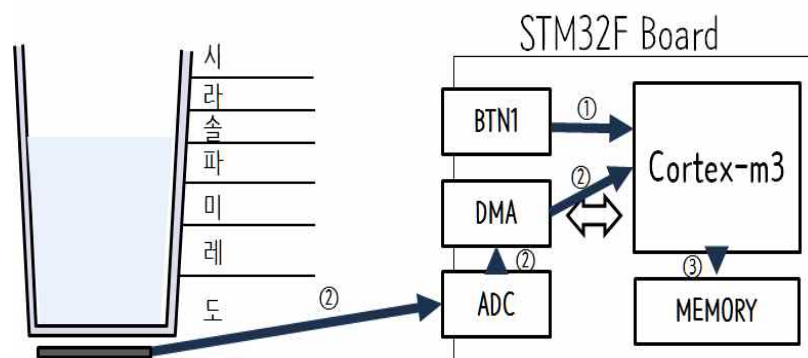


그림 6 기능1에 대한 블록 다이어그램

- 압력센서의 값은 ADC 포트를 이용하여 읽어 들인다.
- DMA를 이용하여 ADC 포트에 읽어 들인 값을 메모리에 쓴다.
- 버튼1을 누르면 인터럽트가 발생한다. 버튼1의 인터럽트 핸들러는 메모리에 저장된 압력센서 값을 이용하여 각 컵에 음을 맵핑한다.
- 맵핑 기준을 만들기 위해서 컵에 적당한 양의 물을 넣어 그 높이를 컵에 표시한다. 이를 '도'로 설정하고 동일한 양의 물을 컵에 넣어 그 높이를 표시한다. 이를 '레'로 설정한다. 이와 같은 것을 반복하여 '시'까지의 높이를 설정한다. 이 후 압력센서마다 해당 높이에서의 압력을 측정한다. 각 센서마다 같은 물체이더라도 그 압력 값을 다르게 출력함을 실험적으로 확인하여 각 센서마다 다른 맵핑 기준을 만들었다.
- 각 음을 맵핑하는 기준은 아래의 표와 같다. 표의 가장 왼편의 PCx(x=0,1,3)은 압력센서의 값을 읽어드리는 ADC 포트이다. MAX는 해당 음계가 가지는 가장 큰 압력센서 수치이고, MIN은 해당 음계가 가지는 가장 작은 압력센서 수치이다.



PC0		도	레	미	파	솔	라	시
	MAX	4090	4021	3985	3880	3818	3724	3664
	MIN	4021	3985	3880	3818	3724	3664	-
PC1		도	레	미	파	솔	라	시
	MAX	4090	4045	3872	3877	3842	3780	3703
	MIN	4045	3872	3877	3842	3780	3703	-
PC3		도	레	미	파	솔	라	시
	MAX	4090	4010	3938	3848	3797	3745	3682
	MIN	4010	3938	3848	3797	3745	3682	-

### 3) 블록 다이어그램3 - 기능2 (컵을 두드리면 음을 출력한다)

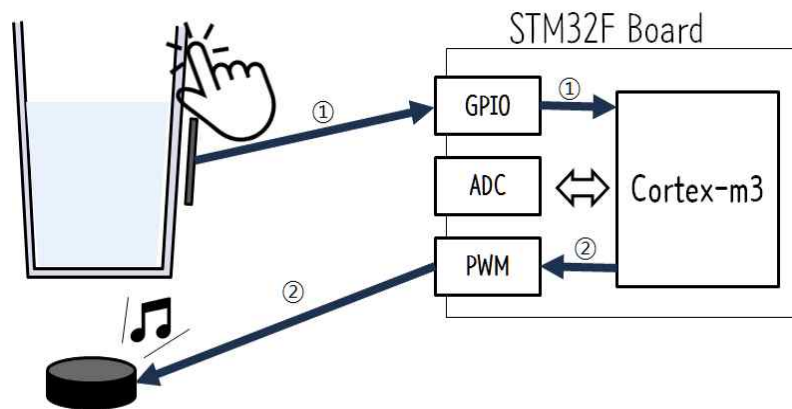


그림 7 기능2에 대한 블록 다이어그램

- 컵을 두드리게 되면 컵에 부착된 진동센서가 진동을 감지한다. 진동센서의 인터럽트는 falling edge에서 발생한다.
- 인터럽트 핸들러에서는 해당 진동센서가 부착된 컵의 음 정보를 찾아 부저에서 음이 출력 되도록 PWM에 사용되는 타이머의 설정을 변경해 준다. 타이머의 설정을 변경하여 부저로 내보내는 주파수를 조정하면 부저에서 출력하는 음의 높이가 달라진다. 각 음에 대한 주파수와 그 값을 찾기 위한 타이머 설정 값을 아래의 표와 같다.

	도	레	미	파	솔	라	시
주파수	1047	1175	1319	1397	1568	1760	1976

	도	레	미	파	솔	라	시
목표 주파수	1047	1175	1319	1397	1568	1760	1976
목표 주기 (1/주파수)	0.00095 5	0.00085 1	0.00075 8	0.00071 6	0.00063 8	0.00056 8	0.00050 6
$f_{CLK}$	$36 \times 10^6$	$36 \times 10^6$	$36 \times 10^6$	$36 \times 10^6$	$36 \times 10^6$	$36 \times 10^6$	$36 \times 10^6$
$TIM\_Prescaler$	36	36	36	36	36	36	36
$TIM\_Period$	955	850	758	716	638	568	506

$$(\ast \text{ 목표 주기} = \frac{1}{f_{CLK}} \times TIM\_Prescaler \times TIM\_Period)$$

#### 4) 블록 다이어그램4 - 기능3 (녹음)

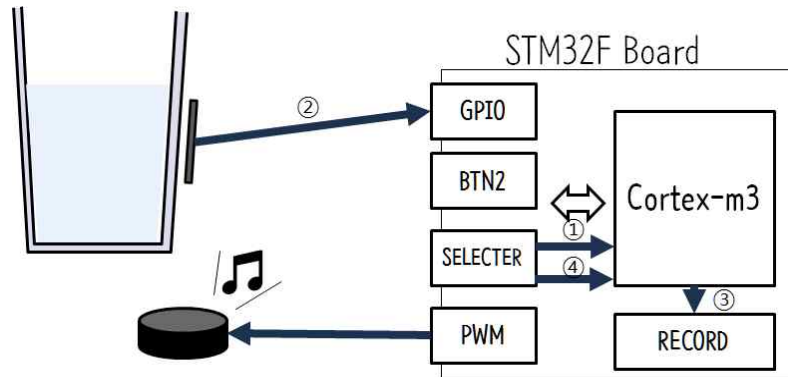


그림 8 기능3에 대한 블록 다이어그램

- joystick의 select를 누르게 되면 인터럽트가 발생한다. select의 인터럽트는 녹음 중이면 녹음을 종료하고 만약 녹음 중이지 않으면 녹음을 시작하는 두 가지 기능을 수행한다.
- 녹음은 0.5초 간격으로 진행되며, 실제 녹음은 0.5초 주기로 발생하는 타이머 인터럽트와 각 컵에 부착된 진동센서의 인터럽트에서 이루어진다.
- 위의 블록 다이어그램(그림8)에서 RECORD에 저장된다.

#### 5) 블록 다이어그램5 - 기능4 (재생)

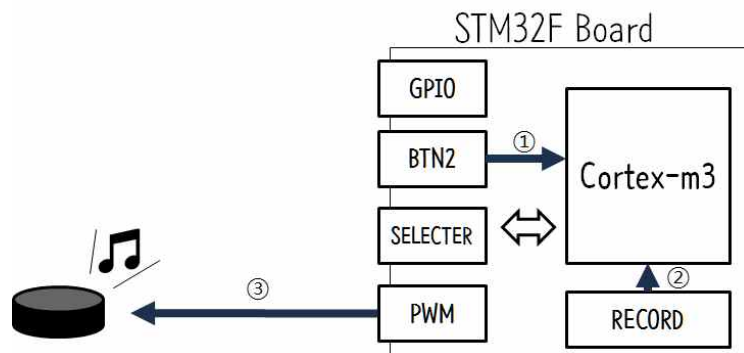


그림 9 기능4에 대한 블록 다이어그램

- 버튼2를 누르면 인터럽트가 발생하여, RECORD에 저장된 음들이 출력된다.
- 시스템 전원을 켜면 초기 RECORD에 샘플 음이 저장되어 있기 때문에 녹음을 하지 않고, 바로 버튼2를 누르면 샘플 곡의 재생을 들을 수 있다.

## 4. 설계 시스템의 부품 관련 설명

### 1) 진동 센서

- 사용목적 : 컵을 두드리는 것을 인식하기 위해 사용된다.
- 링크 : [https://www.dfrobot.com/wiki/index.php/DFRobot\\_Digital\\_Vibration\\_Sensor\\_V2\\_SKU:DFR0027](https://www.dfrobot.com/wiki/index.php/DFRobot_Digital_Vibration_Sensor_V2_SKU:DFR0027)
- 입출력 : 디지털 입력을 통하여 진동여부를 확인할 수 있다.
- 연결방법 : 센서의 빨간색 선에는 전원을 검은색 선에는 GND를 초록색선에는 입력을 받아들이는 포트를 연결시킨다.

### 2) 압력 센서

- 사용목적 : 컵에 들어있는 액체의 양을 파악하여 음을 배정하기 위해 사용한다.
- 링크 : <http://www.interlinkelectronics.com/FSR402.php>
- 입출력 : 아날로그 입력을 통하여 센서의 값을 읽어 들일 수 있다.
- 연결방법 : 그림10에서 확인 할 수 있듯이 해당 센서는 압력에 따라 내부 저항이 증가하는 방식으로 작동한다. 즉 일정한 전압이 주어질 때 압력을 세게 가할수록 센서에 걸리는 전압이 줄어들게 된다. 프로젝트에서는 전압분배의 원리를 이용하여 압력센서에 인가되는 전압을 ADC를 통하여 측정한다.

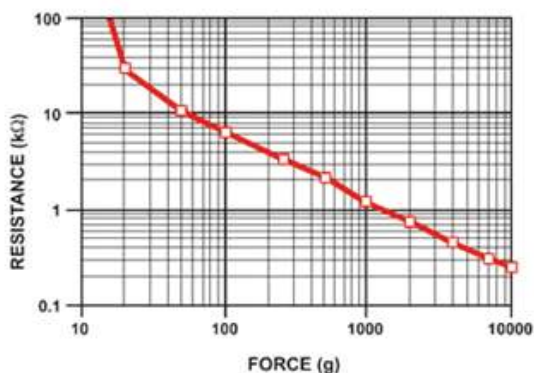


그림 10 힘에 따른 저항의 변화

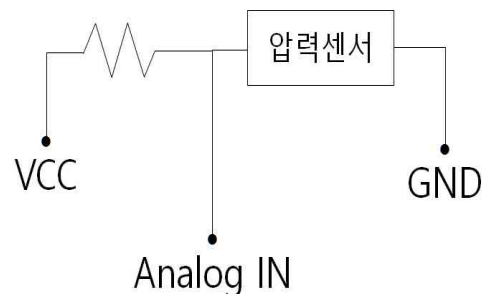


그림 11 압력센서의 연결

### 3) 부저

- 사용목적 : 음을 출력하기 위해 사용한다.
- 입출력 : 일정한 패턴을 가지는 디지털 신호를 입력받는다.
- 연결방법 : 부저의 한쪽에는 디지털 입력을 받고 나머지 한쪽에는 GND를 연결한다.

## 5. 구현 결과

### 1) 전체 시스템 구성 모습

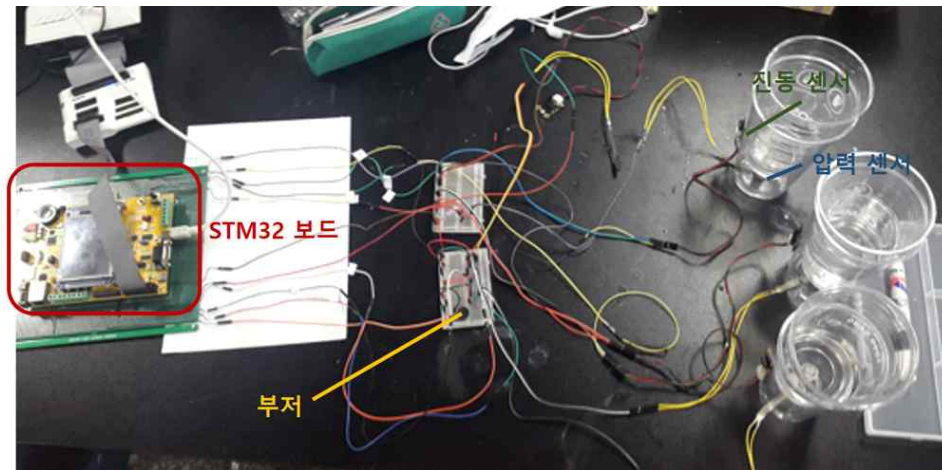


그림 12 전체 시스템 구성

### 2) 음 설정하기 (기능 1)

- 버튼1을 누르면 컵에 들어 있는 물의 양에 따라 (압력에 따라) 각 컵에 음이 설정된다.

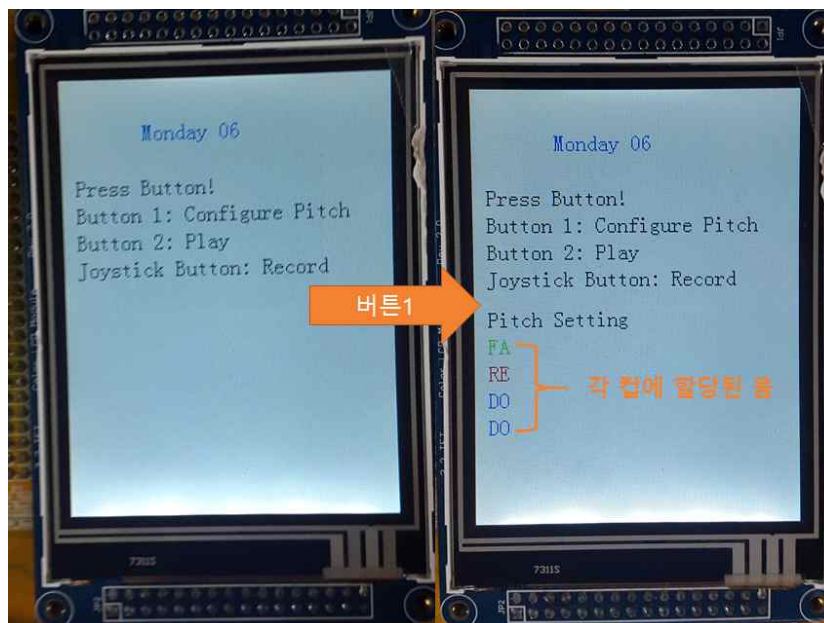


그림 13 기능1 (음 설정하기)

### 3) 소리내기 (기능 2)

- 각 컵에 음이 설정된 상태에서 컵을 두드리면 각 컵에 할당된 음이 부저를 통해 출력된다.



그림 14 기능2 (소리)

#### 4) 녹음하기 (기능 3)

- joystick select 버튼을 누르면 녹음이 시작된다. 녹음되는 중에 컵을 두드리면 해당 음이 그 시간에 저장된다.
- 녹음 시간은 최대 5분으로 설정되어 있다.
- 녹음 중 다시 한 번 joystick select 버튼을 누르면 녹음이 종료된다.

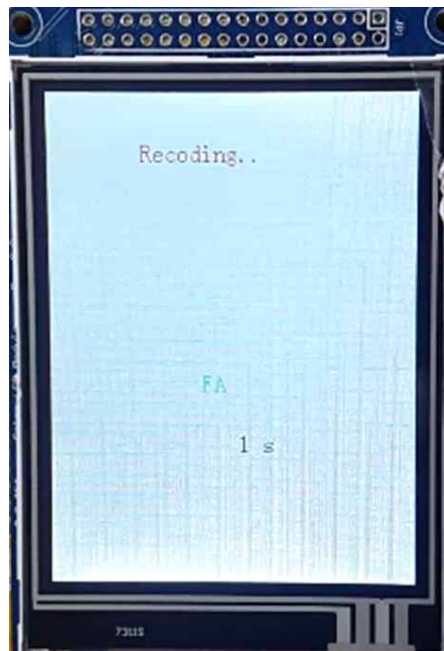


그림 15 기능3 (녹음)

#### 5) 재생하기 (기능 4)

- 버튼2를 누르면 녹음된 소리가 재생된다.
- 시스템 전원을 키면 샘플 곡이 저장되어 있는데, 녹음을 하지 않고, 재생을 하면 이 곡을 들을 수 있다.

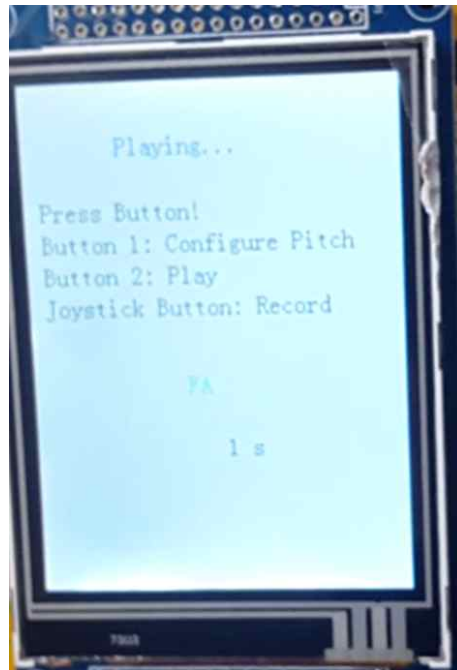


그림 16 기능4 (재생)