

C h a p t e r

07



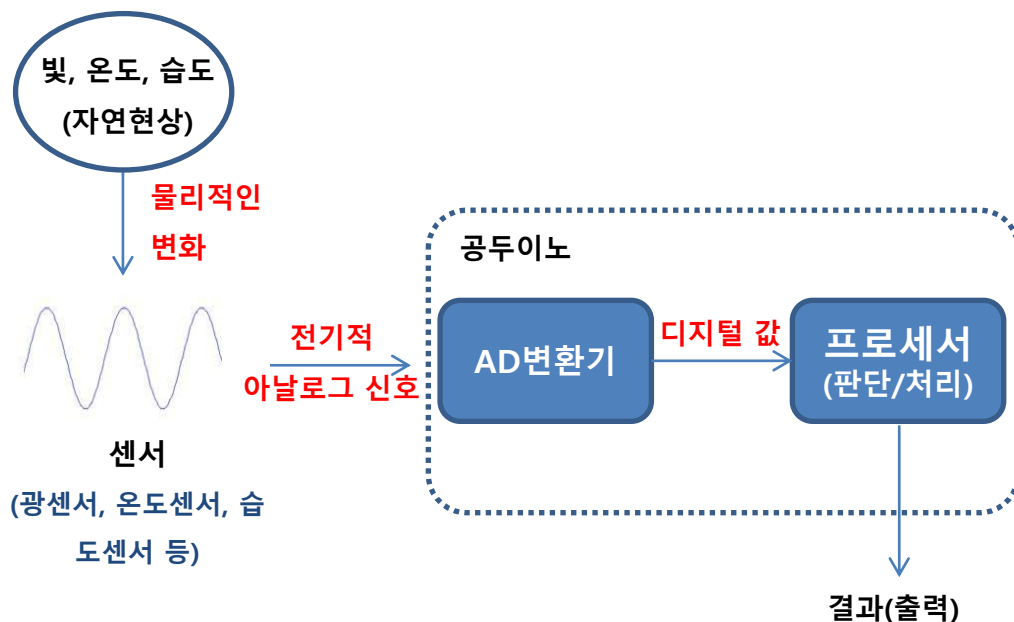
아날로그 신호(analog)

각종 아날로그 소자(가변 저항, 광센서 등)의 값을 읽어 프로그램으로 처리하는 방법을 학습합니다.

■ 아날로그 신호

자연환경 즉, 우리 주변에서 일어나는 모든 현상은 아날로그 특성을 가지고 있습니다. 이러한 아날로그적인 물리적인 현상을 처리하기 위해서는 전기적인 신호로 바꾸어 주어야 하고, 이러한 변환장치를 흔히 센서(Sensor)라고 부릅니다.

센서로 처리되어 발생하는 전기적인 신호는 연속적인 전압의 변화를 갖는 아날로그 전기 신호이므로, 이를 판단하고, 처리하기 위한 디지털 값으로 변환하여야 하며, 이를 AD 변환기(Analog-Digital Converter) 라고 합니다. 이 때 몇 비트의 데이터로 변환하느냐가 분해능(resolution)에 해당하고, 변환을 시작 시켜 변환 종료까지 소요되는 시간이 변환 시간이고, 어느 범위의 전압을 입력 하느냐 (입력 전압 영역) 등이 중요한 요소가 됩니다.



■ AD 변환기

일반적으로 많이 사용하는 분해능은 8/10/12/16/24 비트 등이 사용되고, 변환 시간은 가급적 빠를수록 좋지만 고가가 되므로 적절한 변환 속도를 가진 A/D 변환기를 사용합니다.

우리가 사용하는 공두이노는 10비트 분해능 (오차범위 2LSB), 최대 10KHz 속도, 8채널(아날로그 0번 핀-7번 핀)의 아날로그 입력 사양을 가지고 있습니다.



공두이노는 8개의 아날로그 입력 채널 중 한번에 한 개의 채널만 사용할 수 있으므로, 여러 개의 센서를 연결하여도, 한 개의 센서를 순차적으로 신호 입력 받아야 합니다.

실습 09

가변저항의 전압 입력 처리

베이스 보드의 가변 저항에 의한 전압 값을 아날로그 신호로 입력받아 이를 처리하는 프로그램을 작성합니다.

하드웨어 연결

1. 공두이노 보드의 아날로그 0번 핀과 베이스 보드의 가변저항(VR) 핀을 연결 선으로 연결합니다.

프로그램 작성1

exam031

가변저항의 전압 값을 AD 변환하여 시리얼 통신으로 PC로 전송하는 프로그램을 작성합니다.

```
int val = 0;           // AD 변환값을 저장할 변수

void setup() {
  Serial.begin(9600);
}

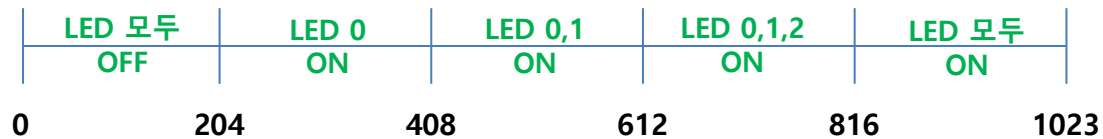
void loop() {
  val = analogRead(0); // 아날로그 0번 핀 (VR) 값을 입력
  Serial.print("VR = ");
  Serial.println(val);
  delay(200);
}
```

- ✓ 업로딩 후, 시리얼 모니터 창을 띄워 출력 화면을 보면서 가변저항을 손으로 돌려봅니다. 출력 화면의 값을 변화를 살펴봅니다.

가변저항의 전압 값을 AD 변환하여 LED 0부터 3까지 4개의 LED를 값에 따라 켜는 프로그램을 작성합니다.

공두이노 보드	연결방향	베이스 보드
2번 PIN	→	LED 0
3번 PIN	→	LED 1
4번 PIN	→	LED 2
5번 PIN	→	LED 3
아날로그 0번 PIN	←	VR

LED 동작



```

int val = 0;

void setup() { Serial.begin(9600); for (int i=2; i<=5; i++) pinMode(i, OUTPUT); }

void loop() {
    val = analogRead(0);
    for (int i=2; i<=5; i++) digitalWrite(i, LOW); //모두 LED OFF
    if (val > 816) digitalWrite(5, HIGH);
    if (val > 612) digitalWrite(4, HIGH);
    if (val > 408) digitalWrite(3, HIGH);
    if (val > 204) digitalWrite(2, HIGH);
    delay(200);
}

```

val 입력 값에 따라 LED 제어



exam033

- ✓ 공두이노의 아날로그 0번 핀에 베이스 보드의 가변저항(VR)을 연결하고, 공두이노의 6번 핀부터 13번 핀까지 베이스 보드의 FND a 부터 DP 까지 연결합니다.
- ✓ 가변저항을 손으로 돌리면 들어오는 전압 값에 따라 FND에 0부터 9까지 숫자로 나타나게 합니다.

베이스 보드의 광센서를 이용하여 여러가지 다양한 동작을 하는 프로그램을 작성합니다.

하드웨어 연결

1. 공두이노 보드의 아날로그 0번 핀과 베이스 보드의 광센서(CDS) 핀을 연결 선으로 연결합니다.

프로그램 작성1



광센서의 값을 AD 변환하여 시리얼 통신으로 PC로 전송하는 프로그램을 작성합니다.

```
int val = 0;           // AD 변환값을 저장할 변수

void setup() {
  Serial.begin(9600);
}

void loop() {
  val = analogRead(0); // 아날로그 0번 핀 (CDS) 값을 입력
  Serial.print("CDS = ");
  Serial.println(val);
  delay(200);
}
```

✓ 업로딩 후, 시리얼 모니터 창을 띄워 출력 화면을 보면서 CDS를 손으로 가릴 때와 그렇지 않을 때의 값의 변화를 살펴봅니다.

광센서의 값을 AD 변환하여 어두우면 LED 0을 켜고, 그렇지 않으면 LED 0을 끄는 가로등 프로그램을 작성합니다.

공두이노 보드	연결방향	베이스 보드
2번 PIN	→	LED 0
아날로그 0번 PIN	←	CDS

```
int val = 0;           // AD 변환값을 저장할 변수

void setup() {
  Serial.begin(9600);
  pinMode(2, OUTPUT); // LED 0을 출력
}

void loop() {
  val = analogRead(0); // 아날로그 0번 핀 (CDS) 값을 입력

  // 빛이 많으면 LED 0을 OFF
  if (val > 200) digitalWrite(2, LOW);
  else digitalWrite(2, HIGH); // 그렇지 않으면 LED 0을 ON
  delay(100);
}
```

- ✓ 업로딩 후, CDS 센서를 손으로 가려보면서 LED 0의 변화를 살펴봅니다.
- ✓ if 문 안의 `val > 200` 의 값 “200” 은 앞 “프로그램 작성1” 을 통해서 적절히 조절하여야 합니다. 일반 적인 경우 CDS 의 센서 값은 0 (어두울 때)부터 500 (밝을 때)사이의 값을 갖습니다.

광센서에 의해 LED 0을 켜고 끄는 프로그램을 수정하여, 판단하는 if 문의 기준 값을 베이스 보드의 가변저항으로 조절할 수 있도록 합니다.

공두이노 보드	연결방향	베이스 보드
2번 PIN	→	LED 0
아날로그 0번 PIN	←	CDS
아날로그 1번 PIN	←	VR

```
int cds = 0;      // 광센서 값을 입력받기 위한 변수
int val = 0;      // 가변저항 값을 입력받기 위한 변수

void setup() {
  Serial.begin(9600);
  pinMode(2, OUTPUT);
}

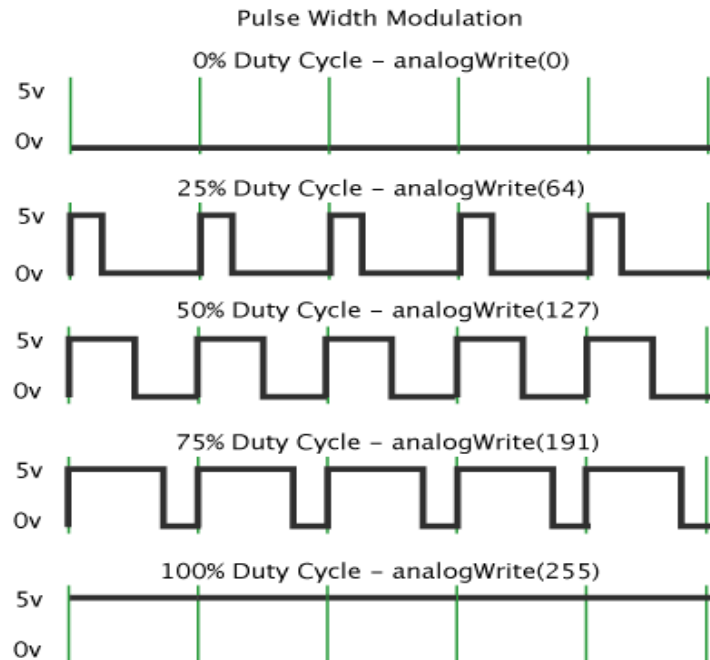
void loop() {
  cds = analogRead(0);  // CDS
  val = analogRead(1);  // VR
  if (cds > val) digitalWrite(2, LOW);
  else digitalWrite(2, HIGH);
  delay(100);
}
```

아날로그 신호를 출력하여 LED의 밝기를 조절하는 프로그램을 작성하고 동작하여 봅니다.

■ 아날로그 신호 출력 (PWM)

아날로그 신호 출력이란, 앞에서 설명한 AD와는 반대의 개념입니다. 즉, 디지털 값을 아날로그 신호로 변환하는 것을 말하며, DA (Digital to Analog Converter)라고 부릅니다. DA 회로는 전기적인 전압의 세기를 조절해야 하는 복잡한 회로로 구성되어, 일반적인 아날로그 신호 출력을 디지털 회로에서는 펄스의 폭을 가변으로 제어하는 PWM (Pulse Width Modulation)으로 표현합니다.

공두이노에서는 0부터 255까지의 8비트 디지털 값을 출력하는 `analogWrite()` 함수를 지원하며, 이를 이용하여 가변된 폭을 갖는 PWM 파형을 생성하여, 아날로그 신호 출력과 같은 효과를 냅니다. 생성되는 펄스의 주기(주파수)는 약 500Hz입니다.



■ 공두이노의 PWM 핀

공두이노에서 아날로그 출력(PWM)을 사용할 수 있는 핀은 정해져 있습니다.

보드의 3, 5, 6, 9, 10, 11 의 6개 핀입니다. 이 6개의 핀은 디지털 신호 입출력에도 사용되며, PWM 신호 출력 그리고 서보 모터의 제어 파형 신호 출력에도 사용됩니다.

하드웨어 연결

1. 공두이노 보드의 3번 핀과 베이스 보드의 LED 0 핀을 연결 선으로 연결합니다.

프로그램 작성1

exam037

공두이노 3번 핀의 아날로그 출력을 이용해 LED 0의 밝기를 점차 밝게 또는 점차 어둡게 합니다.

```
void setup() {  
  pinMode(3, OUTPUT);    // 3번 핀(LED 0)은 출력  
}  
  
void loop() {  
  for (int i=0; i<=255; i++) {  
    analogWrite(3, i);  
    delay(10);  
  }  
  
  for (int i=255; i>=0; i--) {  
    analogWrite(3, i);  
    delay(10);  
  }  
}
```

} LED 0을 점점 밝게

} LED 0을 점점 어둡게

가변 저항을 이용하여 LED 0의 밝기를 조절하는 프로그램을 작성합니다.

공두이노 보드	연결방향	베이스 보드
3번 PIN	→	LED 0
아날로그 0번 PIN	←	VR

```
int val = 0;

void setup() {
  pinMode(3, OUTPUT);
}

void loop() {
  val = analogRead(0);    // 가변저항의 값을 입력
  val = val / 4;          // 가변저항의 값 (10비트) -> PWM (8비트)
  analogWrite(3, val);    // LED0 밝기 출력
}
```



- ✓ 앞 예제의 가로등 프로그램을 수정하여 공두이노의 광센서의 값을 입력 받아 어두워질수록 LED 0이 더 밝게 빛나게 하는 프로그램을 작성합니다.

실습 12

톤(TONE) 출력을 이용한 사운드

아날로그 신호의 일종인 펄스 파형의 주파수를 제어하는 TONE()함수를 사용하여 사운드를 연주하는 프로그램 작성을 실습합니다.

■ 사운드 주파수

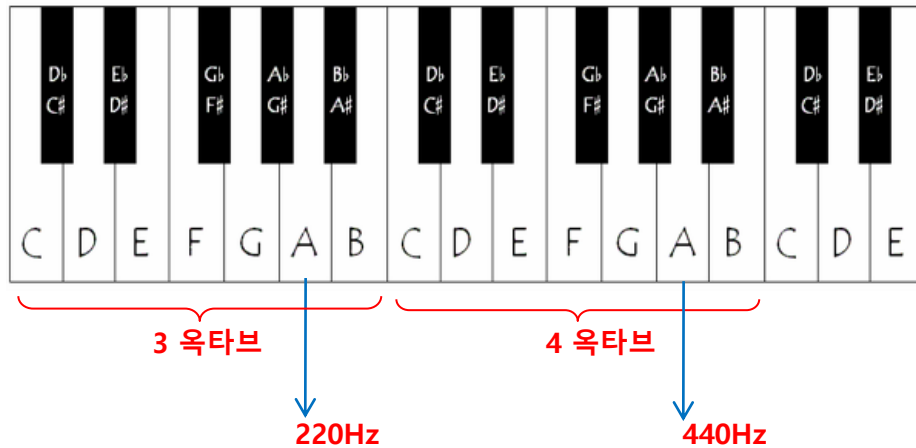
사람의 귀에 들리는 소리는 모두 주파수를 가지고 있습니다. 특히, 사람이 들을 수 있는 주파수를 가청 주파수라고 하며, 이는 20Hz 부터 20,000Hz 의 주파수를 갖습니다. 여기에서는 펄스의 주파수를 조절하여 다양한 사운드를 내는 방법에 대해서 알아봅니다. 악보에 의해 연주되는 노래는 각 음계마다 아래 표와 같은 주파수를 가지고 있습니다.

옥타브 및 음계별 표준 주파수

(단위 : Hz)

옥타브 음계	1	2	3	4	5	6	7	8
C(도)	32.7032	65.4064	130.8128	261.6256	523.2511	1046.502	2093.005	4186.009
C#	34.6478	69.2957	138.5913	277.1826	554.3653	1108.731	2217.461	4434.922
D(레)	36.7081	73.4162	146.8324	293.6648	587.3295	1174.659	2349.318	4698.636
D#	38.8909	77.7817	155.5635	311.1270	622.2540	1244.508	2489.016	4978.032
E(미)	41.2034	82.4069	164.8138	329.6276	659.2551	1318.510	2637.020	5274.041
F(파)	43.6535	87.3071	174.6141	349.2282	698.4565	1396.913	2793.826	5587.652
F#	46.2493	92.4986	184.9972	369.9944	739.9888	1479.978	2959.955	5919.911
G(솔)	48.9994	97.9989	195.9977	391.9954	783.9909	1567.982	3135.963	6271.927
G#	51.9130	103.8262	207.6523	415.3047	830.6094	1661.219	3322.438	6644.875
A(라)	55.0000	110.0000	220.0000	440.0000	880.0000	1760.000	3520.000	7040.000
A#	58.2705	116.5409	233.0819	466.1638	932.3275	1864.655	3729.310	7458.620
B(시)	61.7354	123.4708	246.9417	493.8833	987.7666	1975.533	3951.066	7902.133

다음페이지 건반 참조



하드웨어 연결

1. 공두이노 보드의 3번 핀과 베이스 보드의 PIEZO 핀을 연결 선으로 연결합니다.

프로그램 작성1

exam040

공두이노 3번 핀의 톤을 조절하여 피에조 소자에서 싸이렌 소리가 나도록 합니다.

```
void setup() {
  pinMode(3, OUTPUT); // 톤 주파수 출력 (피에조)
}

void loop() {
  tone(3, 500);        // 높은 주파수 (500Hz)
  delay(500);          // 0.5 초 유지
  tone(3, 250);        // 낮은 주파수 (250)Hz
  delay(500);          // 0.5 초 유지
}
```

피에조 소자에서 4옥타브 도에서 5옥타브 도까지 연주해 봅니다.

// 음계에 따른 주파수 정의

```
#define c_4 261
#define d_4 293
#define e_4 329
#define f_4 349
#define g_4 392
#define a_4 440
#define b_4 493
#define c_5 523
```

⋮

```
void setup() {
    pinMode(3, OUTPUT);
}

void loop() {
    tone(3, c_4); delay(800);
    tone(3, d_4); delay(800);
    tone(3, e_4); delay(800);
    tone(3, f_4); delay(800);
    tone(3, g_4); delay(800);
    tone(3, a_4); delay(800);
    tone(3, b_4); delay(800);
    tone(3, c_5); delay(800);
}
```

공두이노 3번 핀의 톤을 조절하여 피에조 소자에서 다음 동요를 연주합니다. (첫 줄만)

```
#define f_4 349
#define g_4 392
#define a_4 440

#define l_4 800
#define l_8 400
#define l_8p 600
#define l_16 200
```

사용되는 계명 주파수

음 길이 (4분 음표 800ms 기준)



```
void setup() {
  int melody[] = {a_4,g_4,f_4,g_4,a_4,a_4,a_4,g_4,g_4,g_4,a_4,a_4,a_4};
  int length[] = {l_8p,l_16,l_8,l_8,l_8,l_8,l_4,l_8,l_8,l_4,l_8,l_8,l_4};

  pinMode(3, OUTPUT);

  for (int i=0; i<13; i++) {
    tone(3, melody[i]); delay(length[i]); // 각 계명 연주
    noTone(3); delay(100); // 음 중간에 잠깐 소리 중단
  }
}

void loop() {
}
```

loop () 함수는 비움

실습 13

전자 피아노 만들기

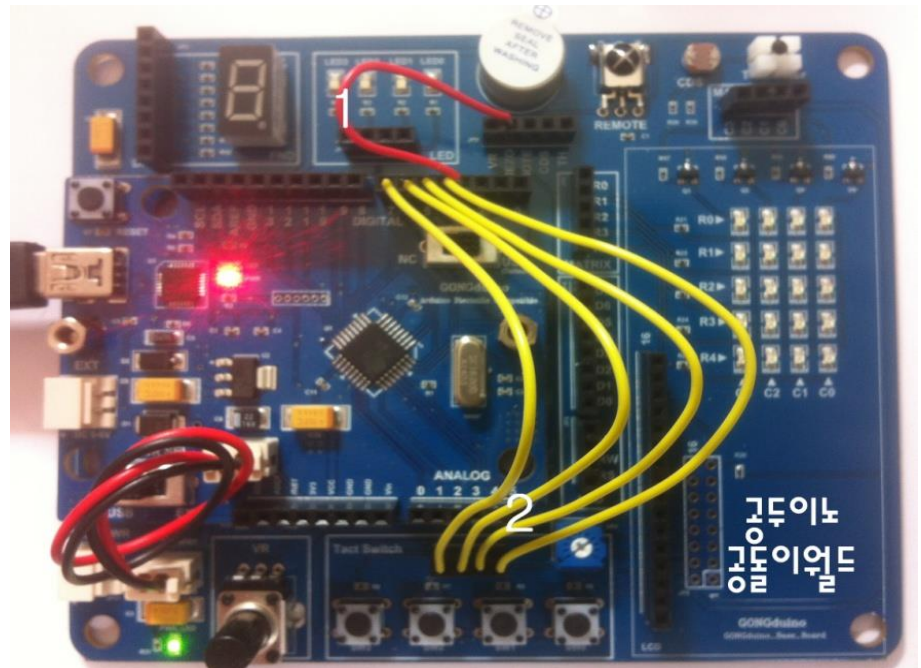
택트 스위치와 피에조를 사용하여 전자 피아노 프로그램을 작성합니다.

전자 피아노는 SW0에서 SW3에 “도” - “파” 까지 피에조를 통해 소리를 냅니다.

하드웨어 연결

1. 공두이노 보드와 베이스 보드(Gongduino-base-board)의 전원 커넥터를 케이블로 연결합니다.
2. 베이스 보드의 SW 연결 핀 4개(2)와 피에조 연결 핀(1)을 연결선으로 공두이노 보드와 연결합니다.

공두이노 보드	연결방향	베이스 보드	연주음
3번 PIN	→	PIEZO	
4번 PIN	←	SW0	파
5번 PIN	←	SW1	미
6번 PIN	←	SW2	레
7번 PIN	←	SW3	도



```
#define c_4  261          // 도
#define d_4  293          // 레
#define e_4  329          // 미
#define f_4  349          // 파

int inkey() {
    int sw = 0;
    for (int i=0; i<=3; i++){
        bitWrite(sw, i, digitalRead(i+4));
    }
    return sw;
}
```

⋮

```
void setup() {
    pinMode(3, OUTPUT);          // 피에조
    for (int i=4; i<=7; i++) pinMode(i, INPUT);    // 스위치
}

void loop() {
    int sw=0, new_sw=0;

    switch(sw = inkey()){        // 스위치 입력 받아서
        case 8: tone(3, c_4); break; // SW3 - 도
        case 4: tone(3, d_4); break; // SW2 - 레
        case 2: tone(3, e_4); break; // SW1 - 미
        case 1: tone(3, f_4); break; // SW0 - 파
    }

    delay(10);                    // 스위치 디바운스
    while(1) {                    // 스위치가 놓일 때까지 대기
        new_sw = inkey();
        if (sw != new_sw) break;
    }

    noTone(3);                    // 스위치 놓이면 소리를 끄
}
```

광센서 CdS 소자와 피에조 소자를 연결하여 빛의 양에 따라 소리를 다르게 내는 프로그램을 작성합니다.

하드웨어 연결

1. 공두이노 보드와 베이스 보드(Gongduino-base-board)의 전원 커넥터를 케이블로 연결합니다.
2. 베이스 보드의 CdS 연결 핀과 피에조 연결 핀을 연결선으로 공두이노 보드와 연결합니다.

공두이노 보드	연결방향	베이스 보드
3번 PIN	→	PIEZO
ANALOG 0	←	CDS

exam044

```
int value = 0;
int old_value = 0;

void setup() {
    pinMode(3, OUTPUT);           // 피에조
}

void loop() {
    value = analogRead(0);        // 광센서 (CDS) 값을 읽어
    if ((value < old_value-10) || (value > old_value+10)) { // 현재 센서 값이 이전 센서 값에서 ±10 범위를 벗어나면
        tone(3, value*2);         // 광센서 값 * 2배로 소리 냄
        old_value = value;        // 현재 값으로 이전 값 새로 고침
    }
}
```