2023년 IoT기반 스마트 솔루션 개발자 양성과정



### **Embedded Application**

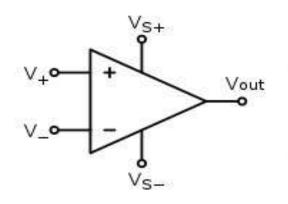
### **10-Circular Queue**

담당 교수 : 윤 종 이
010-9577-1696
ojo1696@naver.com
https://cafe.naver.com/yoons2023



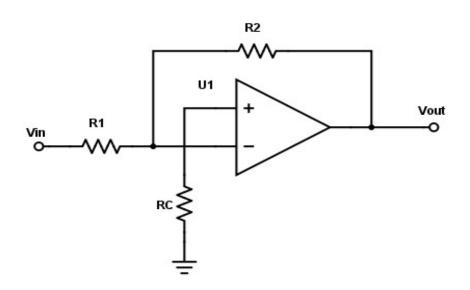
### **OP-AMP**

- 연산 증폭기(op-amp, Operational amplifier)
- 차동 입력과, 한 개의 단일 출력을 가지는 직류 연결형(DC-coupled) 고이득 전압 증폭기.
- 각종 센서의 신호를 증폭하여 사용함

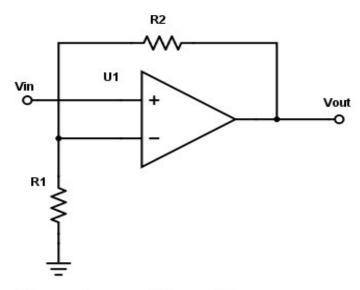


- ullet  $V_+$ : 비반전 신호 입력
- ullet  $V_-$  반전 신호 입력
- ullet  $V_{
  m out}$ : 출력
- ullet  $V_{
  m S+}$ : 양의 전원 공급 전압
- ullet  $V_{
  m S-}$  음의 전원 공급 전압

## **Amplifier**



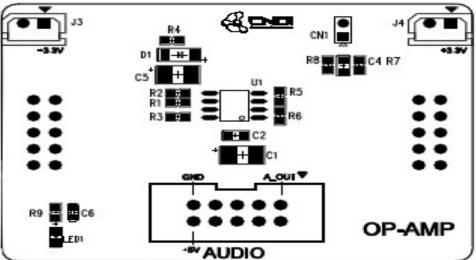
**Inverting Op-amp** 



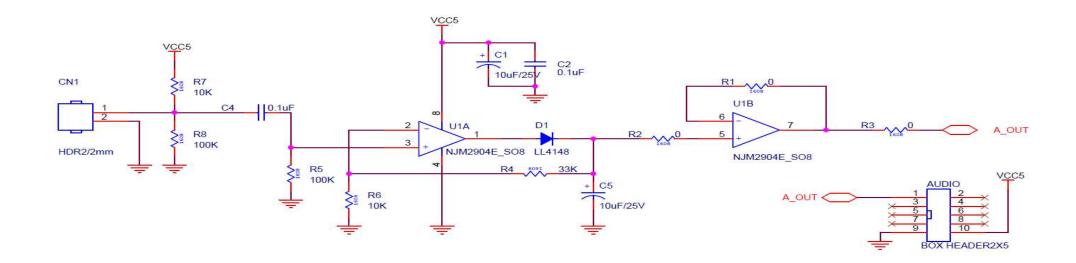
Non-inverting Op-amp

## OP-AMP 모듈





### **OP-AMP Circuit**

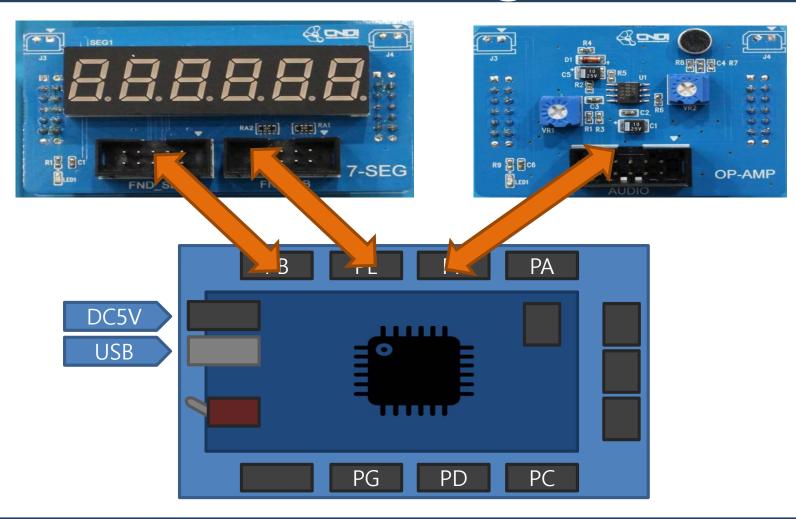


# Ex-1: 소음 변환기

• 주변의 소음을 측정하여 FND에 표시하여 보자



# Ex-1: Wiring





충북대학교 공동훈련센터

### Ex-1: Define

```
#define F CPU 14745600UL
#define FND SEL PORTB
#define FND DB PORTE
#define dTime 3
#include <avr/io.h>
#include <util/delay.h>
unsigned char FND[17]=\{0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x27, 0x7f, 0x6f, 0x77, 0x7c, 0x58, 0x5e, 0x40, 0x49, 0x49,
0x40}:
unsigned char DGT[6]={0xfe, 0xfd, 0xfb, 0xf7, 0xef, 0xdf};
unsigned char NUM[6]=\{0x00, 0x00, 0x00, 0x00, 0x00, 0x00\};
unsigned int Sound=0;
unsigned int ADConv( unsigned char Channel ) {
              ADMUX = (Channel & 0x03):
               ADCSRA = (1<<ADEN) | (1<<ADSC) | (1<<ADPS1) | (1<<ADPS0);
               _delay_us(100);
              while( ~ADCSRA & 0x10 );
               return ADC:
```

### Ex-1 : sub

```
void Hex2Dec(unsigned int No){
   unsigned int tmp=No;
   NUM[5]=8;
                        //'8'
                       //'-'
   NUM[4]=16;
   NUM[3]=tmp/1000;
   tmp=tmp%1000;
   NUM[2]=tmp/100;
   tmp=tmp%100;
   NUM[1]=tmp/10;
   NUM[0]=tmp%10;
void FND Display(){
   for (unsigned char k=0; k<6; k++) {
      FND_SEL=DGT[k];
      FND_DB=FND[ NUM[k] ];
      _delay_ms(dTime);
```

### Ex-1: main

```
void CPU_Setup( ) {
   DDRB=0xff;
   DDRE=0xff;
   DDRF=0xF0;
int main(void) {
  CPU_Setup();
  while (1) {
     Sound=ADConv(0);
     Hex2Dec(Sound);
     FND_Display();
```

### Data processing

#### Data Processing

- 획득 되어진 자료를 유용한 형태로 변환하는 과정
- 알고리즘의 도출, 논리적 추론, 통계적 계산으로 한정되어 실 생활의 모든 정보의 변환 과정
- 자료는 올바르게 표현되면 편리하고 실용적인 정보가 되기 때문에, 데이터 처리 시스템은 실용성을 강조하기 위해 정보 시스템

#### Data Mining

- 대규모로 저장된 데이터 안에서 체계적이고 자동적으로 통계적 규칙이나 패턴을 찾아 내는 것
- 분류, 군집화, 연관성, 연속성, 예측 등

Information intelligence Data Processing Data mining Useable Data

### Circular Queue

- Linear Queue
  - 먼저 집어 넣은 데이터가 먼저 나오는 FIFO (First In First Out)구조
  - Insert(Put)
    - Data[F] = Data
    - F=F+1
    - OverFlow when F>Maxindex
  - Delete(Get)
    - Loop { Data[R]=Data[R+1] }
    - Data[MaxIndex)] = Null
- Circula Queue
  - 선형 큐의 문제점을 보완한 것
  - Insert(Put)
    - Data[F] = Data
    - F=F+1
    - If (R>MaxIndex), F=0
  - Delete(Get)
    - R=R+1
    - If (R>MaxIndex), R=0





### Ex-2: Circular Queue

- 주변의 소음을 측정하여 FND에 표시하여 보자
- 환형 큐를 이용하여 Smooth하게 표현해 보자



### Ex-2: CircularQueue

```
#define MaxQue 5
unsigned int QSound[MaxQue];
unsigned char QPoint=0;
void CircularQueue( int Value ) {
  QSound[ QPoint ] = Value;
  if ( ++QPoint > MaxQue - 1 ) QPoint = 0;
unsigned int mean() {
  unsigned int Temp = 0;
  for ( int K = 0 ; K < MaxQue ; K++ ) {
      Temp += QSound[K];
  return Temp / MaxQue;
```

## Ex-2: main

```
int main(void) {
  CPU_Setup();
  while (1) {
     Sound=ADConv(0);
     CircularQueue(Sound);
     Hex2Dec( mean() );
     FND_Display();
```

## 통계적 처리

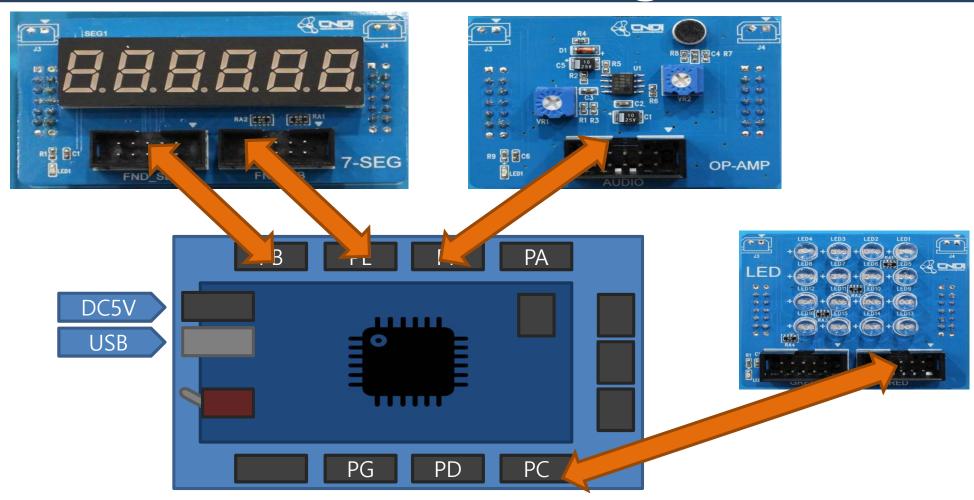
- 현재 값이 실제 값일까? (노이즈)
- 변화량이 많다면 실제값은 어떤값일까?
- 통계적 처리방법론
  - 산술평균값 (mean value)
  - 중앙값 (median value)
  - 최빈값 (Mode value)
- 선택조건
  - 현실값을 반영하여야 한다
  - 변화량을 반영하여야 한다
  - 계산량이 적어야 한다
  - 알고리즘이 간단하여야 한다

### Ex-3: 비교

- 주변의 소음을 측정하여 FND에 표시하여 보자
- 환형 큐를 이용하여 Smooth하게 표현해 보자
- 설정된 값 보다 크면 LED Module을 켜자



# Ex-3: Wiring





충북대학교 공동훈련센터

## Ex-3: Program

```
void LED OnOff(unsigned int RealValue){
  if (SetValue < RealValue){</pre>
    LED MODULE=0xFF;
  } else {
    LED MODULE=0x00;
void CPU_Setup( ) {
  DDRB=0xff:
  DDRE=0xff;
  DDRF=0xF0;
  DDRC=0xff;
```

```
int main(void) {
  CPU_Setup();
  while (1) {
    Sound=ADConv(0);
    CircularQueue(Sound);
    SoundMean=mean();
    Hex2Dec( SoundMean );
    FND_Display();
    LED_OnOff( SoundMean );
```