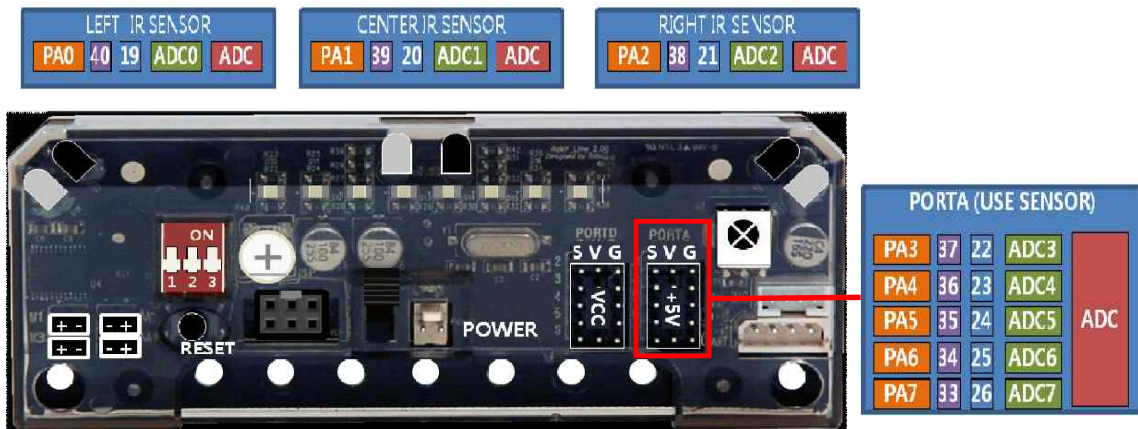


2. ADC(Analog – Digital Converter)

1) 개요

ADC(Analog to Digital Converter)는 전기적인 아날로그 량을 디지털 량으로 변환시키는 장치이다. 자연계의 현상들을 전자적인 방법으로 측정하면 대체로 전류나 전압의 연속적인 값으로 나타내어지고 이것을 컴퓨터가 처리하기 위해서는 디지털 신호로 변환 되어져야 한다.

스마트 인벤터 보드의 ADC 장치는 아래 그림처럼 배치되어 있다.

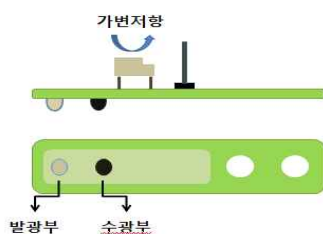


ADC의 기능을 갖고 있는 핀은 19번부터 26번까지이며 19번은 A0, 20번은 A1, 21번은 A2로 정의되어 있다. 이런식으로 핀 번호 대신 A0, A1,... A7을 사용해도 된다. 내장된 전방 적외선 센서 3개는 19번, 20번, 21번이며 나머지 22번부터 26번은 외부 ADC 장치와 연결할 수 있도록 구성하였다.

2) 연결방법

내장된 전방 적외선 센서는 배선이 필요 없이 제어할 수 있으나 외부 적외선 센서 보드를 사용하는 경우엔 PORTA에 배선을 해야 한다.

아래 그림처럼 적외선 센서보드는 LED모듈과 같이 전원 핀(V와G) 그리고 신호(S)핀으로 연결부가 구성되어 있으며 역시 **3핀 점퍼 와이어**를 사용하여 배선한다. 직사광선이 비치지 않는 실내에서 약 7cm 이내의 감지 거리를 가질 수 있다. 가변저항을 좌우로 돌려서 감도를 조절할 수 있는 기능도 가지고 있다.



포텐서미터 보드(위 그림 오른쪽)는 가변 저항 모듈이라고도 한다. 다이얼을 돌려 회전하는 각도의 변화량에 비례하여 출력 전압의 변화가 생긴다. 이 보드 역시 ADC기능이 있는 핀에 연결하여 사용하며 적외선 센서 보드와 핀 구성이 동일하고 3핀 점퍼 와이어를 사용하여 배선한다.

3) 아두이노 함수 설명

`analogRead(pin);`

pin: 아날로그 핀 번호

반환: 0 ~ 1023 정수값

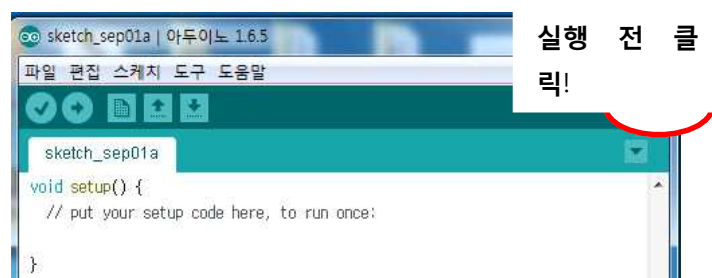
위의 함수는 0 ~ 5V 이내의 아날로그 입력 값을 0 ~ 1023사이의 디지털 값으로 반환하여 준다. 핀 번호는 핀 맵의 번호를 사용해도 되지만 앞에서도 언급한 것처럼 A0, A1, A2와 같은 기호를 사용할 수 있다.

4) 프로그래밍

예제1 : 내장된 전방 적외선 센서 값 출력하기

아래 예제는 전방에 있는 3개의 ADC값을 출력한다. 실행시킨 후 시리얼 모니터를 열어서 값을 출력하는 도중 전방에 있는 세 개의 센서에 물체나 손을 갖다 대면 값의 변화가 있는 것을 확인할 수 있다.

```
void setup() {  
    Serial.begin(9600);    //시리얼 통신 포트를 연다.  
}  
  
void loop() {  
    int frontSensor0 = analogRead(A0);           //전방의 센서 3개의 값을 읽어온다.  
    int frontSensor1 = analogRead(A1);  
    int frontSensor2 = analogRead(A2);  
  
    Serial.print(frontSensor0);                   //값을 출력한다.  
    Serial.print(" ");  
    Serial.print(frontSensor1);  
    Serial.print(" ");  
    Serial.println(frontSensor2);  
}
```



예제2: 전방 적외선 센서로 내부 LED 제어하기

전방에 있는 센서를 감지시킬 때 내부 LED가 켜지는 예제이다. 왼쪽 전방 센서를 감지하면 가장 왼쪽의 LED가 켜지고, 중앙의 전방 센서를 감지하면 가운데 LED 두개가 켜지고, 오른쪽 전방 센서를 감지하면 가장 오른쪽의 LED가 켜지게 된다.

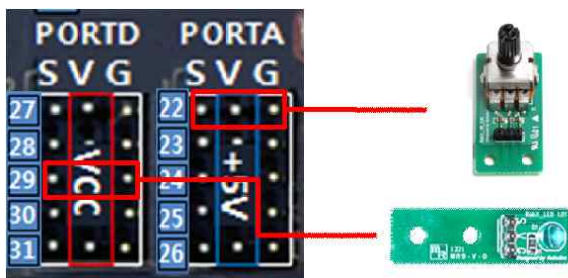
```
void setup() {  
    pinMode(11, OUTPUT);  
    pinMode(14, OUTPUT);  
    pinMode(15, OUTPUT);  
    pinMode(18, OUTPUT);  
}  
  
void loop() {  
    int frontSensor0 = analogRead(A0);  
    int frontSensor1 = analogRead(A1);  
    int frontSensor2 = analogRead(A2);  
  
    if(frontSensor0 < 100) digitalWrite(11, HIGH);           //각각의 센서가 감지되었는 지 판단  
    else digitalWrite(11, LOW);  
    if(frontSensor1 < 100) {digitalWrite(14, HIGH); digitalWrite(15, HIGH);}  
    else {digitalWrite(14, LOW); digitalWrite(15, LOW);}  
    if(frontSensor2 < 100) digitalWrite(18, HIGH);  
    else digitalWrite(18, LOW);  
}
```

***예제 검토:** 센서가 감지되었는 지의 여부를 100의 값을 기준으로 하여 판단하였으나 이 값은 주위의 환경이나 감지되는 물체의 종류에 따라서 상당히 달라질 수 있다. 따라서 예제2를 실행하기 전에 예제1을 실행하여 센서 값의 상태를 확인해 보는 것이 좋다.

예제3: 외부 포텐서미터 보드를 사용하여 LED 밝기 조절하기

필요부품: LED 보드 1, 포텐서미터 보드 1, 3핀 와이어 2

연결방법: LED 보드 -> 29번, 포텐서미터 보드 -> 22번



본 예제는 포텐셔미터 모듈로 LED의 밝기를 조절하는 예제이다. 포텐셔미터 모듈을 통해 변화하는 ADC의 값이 LED의 밝기를 조절하는 전압값으로 변환되어 출력된다.

```
void setup() {  
    pinMode(29, OUTPUT);  
}  
  
void loop() {  
    int potentiometer = analogRead(A3);  
    int ledLight = map(potentiometer, 0, 1023, 0, 255);  
    // 값의 범위를 아날로그 출력에 맞추어 변환  
    analogWrite(29, ledLight);  
}
```

예제 검토: 아날로그 값의 범위는 0 ~ 1023이므로 이 값을 그대로 전압 값에 대입하지 않고 map 함수를 이용하여 범위를 0 ~ 255로 조정하여 사용하였다.