

# 임베디드컴퓨팅

Embedded Computing  
(0009488)

# Digital Input

2022년 2학기

정보기술대학 정보통신공학과

김 영 필

[ypkim@inu.ac.kr](mailto:ypkim@inu.ac.kr)

# Input Devices

- Input devices include sensors, microphones, cameras, keyboards, mice, and many more.
- Depending on the electronics, you need to consider whether to connect to an analog input or a digital input.
- The functions used can be different.

입력 방법	아날로그 입력	디지털 입력
사용하는 전자 부품	<ul style="list-style-type: none"> <li>• 볼륨</li> <li>• 가변저항</li> <li>• 광센서</li> <li>• 온도 센서</li> <li>• 적외선 거리 센서</li> <li>• 가속도 센서</li> </ul>	<ul style="list-style-type: none"> <li>• 스위치</li> <li>• 진동/기울기 센서</li> <li>• 기울기 센서</li> <li>• 자기장 센서</li> <li>• 인체 감지 센서</li> <li>• 적외선 리모컨 수신 모듈</li> </ul>
사용하는 함수	analogRead	pinMode와 digitalRead

신호	아두이노 우노의 입력 포트(관련 함수)
아날로그 입력	A0~A5(analogRead 함수 사용)
디지털 입력	① D0~D13(pinMode와 digitalRead 함수 사용) (A0~A5는 각각 D14~D19로도 사용 가능) ② 시리얼 통신(UART, I2C, SPI)

\* 고급 센서 중에는 이 표와 다른 제품도 있다.

표 Source: 길벗, “모두의 아두이노”

# E.g. Input devices for Arduino

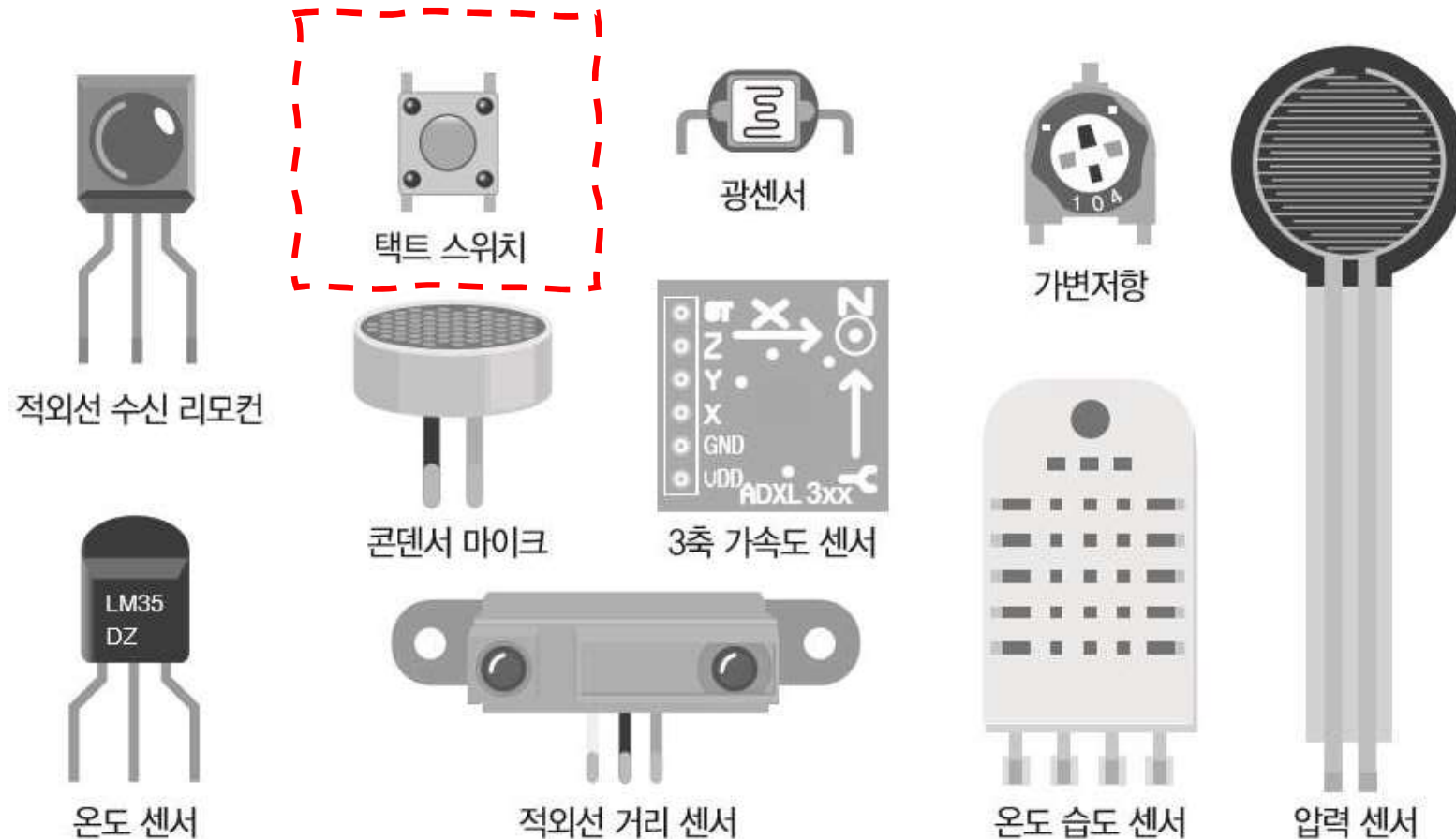


그림 Source: 길벗, "모두의 아두이노"

+ Touch sensor, fine dust sensor, passive infrared sensor (PIR), magnetic switch, etc. ..

# Analog Input

- To read data values by connecting analog electronic components to the Arduino, you must use **analogRead()**.
  - devices connected to analog pins
- The return value of **analogRead()** is an integer value 0 to 1023
  - the value read by the electronic component
  - **limited to the resolution of the analog to digital converter** (0-1023 for 10 bits or 0-4095 for 12 bits). Data type: **int**.

BOARD	OPERATING VOLTAGE	USABLE PINS	MAX RESOLUTION
Uno	5 Volts	A0 to A5	10 bits
Mini, Nano	5 Volts	A0 to A7	10 bits
Mega, Mega2560, MegaADK	5 Volts	A0 to A14	10 bits
Micro	5 Volts	A0 to A11*	10 bits
Leonardo	5 Volts	A0 to A11*	10 bits
Zero	3.3 Volts	A0 to A5	12 bits**
Due	3.3 Volts	A0 to A11	12 bits**
MKR Family boards	3.3 Volts	A0 to A6	12 bits**

Ref. -

<https://www.arduino.cc/reference/en/language/functions/analog-io/analogread/>

함수 이름	설명
analogRead(핀 번호);	아날로그 입력 값을 읽는다. 반환 값: 0~1023 0은 0V를 의미하고, 1023은 5V를 의미한다.

그림 Source: 길벗, “모두의 아두이노”

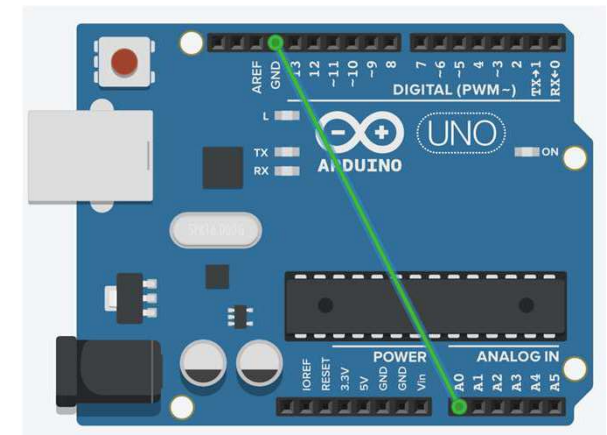
# Let's try Analog input

```
int val = 0;
void setup() {
  Serial.begin(9600);
}
void loop() {
  [REDACTED]
  Serial.print("Analog 3 input = ");
  Serial.println(val);
  [REDACTED]
  Serial.print("Analog 0 input = ");
  Serial.println(val);
  delay(1000);
}
```



What happens?

Connect A0 with GND, then read A0.

What happens?



# What we read?

- AnalogRead() map input voltages between 0 and the operating voltage(5V or 3.3V) into integer values   

- On an Arduino Uno, this yields a resolution between readings of: 5 volts / 1024 units
  - 0.0049 volts (4.9 mV) per unit.
- E.g. read value 290
  - 1.421 V
- E.g. read value 0
  - 0 V

# Analog input values need interpretation

- Returning an integer value of 0 to 1023 means that the analog value can be read by dividing it into 1024 divisions.
- To convert this value to a value with units (distance in centimeters, temperature in °C, humidity in %, etc.), you need to use the **conversion formula (calculation formula)** of each electronic component.
- For calculations, you need to **write a program** using graphs or equations in the electronic component manual.

# Let's make voltage monitor by analog input

```
int analogPin = A3;
int val = 0;
void setup() {
  Serial.begin(9600);
}
void loop() {
  val = analogRead(A3);
  Serial.print("Analog 3 input = ");
  Serial.println(val);
  
  Serial.println("V");
  
  Serial.print("Analog 0 input = ");
  Serial.println(val);
  
  Serial.println("V");
  delay(1000);
}
```

Connect POWER  
pins (5V or 3.3V) to  
A3 or A0.

What can you  
observe?

Calculation is  
required as sensing  
data.



# Digital Input

- For digital input, two functions: **pinMode()** and **digitalRead()** are used
- digitalRead produces 0 (LOW) or 1 (HIGH)

함수 이름	설명
pinMode(핀 번호, 모드);*	핀 동작을 입력이나 출력으로 설정한다. <ul style="list-style-type: none"><li>• 핀 번호: 설정할 핀 번호</li><li>• 모드: 입력(INPUT 또는 INPUT_PULLUP)</li></ul>
digitalRead(핀 번호);	디지털 입력 값을 읽는다. <ul style="list-style-type: none"><li>• 반환 값: On 상태(HIGH) 또는 Off 상태(LOW)</li><li>• 핀 번호: D0~D13 또는 A0(D14)~A5(D19) 사용 가능</li></ul>

\* 모드가 INPUT일 때는 생략할 수 있다.

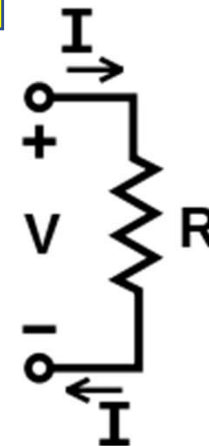
그림 Source: 길벗, “모두의 아두이노”

# pinMode(), again

- The **pinMode** function is used to set digital input/output.
- Syntax
  - pinMode(pin, mode)
  - **mode**
    - can be OUTPUT, INPUT, or INPUT\_PULLUP
- For INPUT mode, the pinMode function can be omitted.
- Then, INPUT\_PULLUP?

# Resistor

- Resistors are electronic components that control the flow of current
  - Carbon Film Resistors are commonly used.
- The unit of resistance is ohm ( $\Omega$ ).
  - When a current flows through both terminals of a resistor, **the larger the resistance, the less current flows,** and **the smaller the resistance, the more current flows.**
  - Also, unlike a diode, a resistor has



Ohm's law

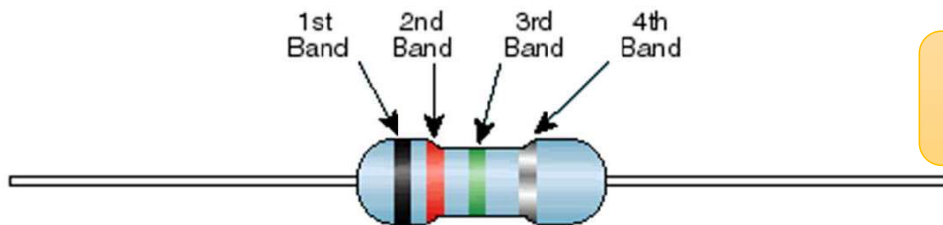
Fig. Ref -  
[https://en.wikipedia.org/wiki/Ohm%27s\\_law](https://en.wikipedia.org/wiki/Ohm%27s_law)

# How to distinguish resistor?

- Refer the Resistor color code
  - E.g. 4 Band: RED RED BROWN GOLD = ?
    - 220  $\Omega$  (-5%, 209 ~ +5%, 231)



**Standard EIA Color Code Table 4 Band:  $\pm 2\%$ ,  $\pm 5\%$ , and  $\pm 10\%$**

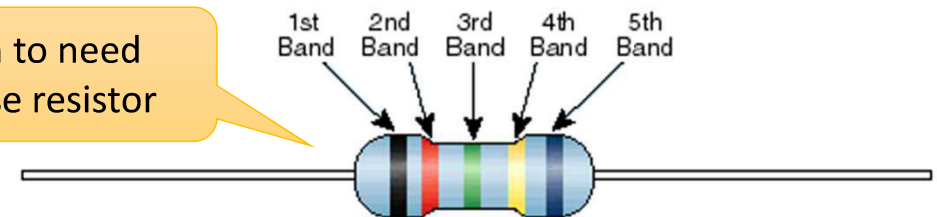


Color	1st Band (1st figure)	2nd Band (2nd figure)	3rd Band (multiplier)	4th Band (tolerance)
Black	0	0	$10^0$	
Brown	1	1	$10^1$	
Red	2	2	$10^2$	$\pm 2\%$
Orange	3	3	$10^3$	
Yellow	4	4	$10^4$	
Green	5	5	$10^5$	
Blue	6	6	$10^6$	
Violet	7	7	$10^7$	
Gray	8	8	$10^8$	
White	9	9	$10^9$	
Gold			$10^{-1}$	$\pm 5\%$
Silver			$10^{-2}$	$\pm 10\%$

Chart Provided By XICON

**Standard EIA Color Code Table 5 Band:  $\pm 1\%$ ,  $\pm 25\%$ ,  $\pm 5\%$ ,  $\pm 1\%$**

When to need  
precise resistor



Color	1st Band (1st figure)	2nd Band (2nd figure)	3rd Band (3rd figure)	4th Band (multiplier)	5th Band (tolerance)
Black	0	0	0	$10^0$	
Brown	1	1	1	$10^1$	$\pm 1\%$
Red	2	2	2	$10^2$	
Orange	3	3	3	$10^3$	
Yellow	4	4	4	$10^4$	
Green	5	5	5	$10^5$	$\pm 5\%$
Blue	6	6	6	$10^6$	$\pm 25\%$
Violet	7	7	7	$10^7$	$\pm 1\%$
Gray	8	8	8	$10^8$	
White	9	9	9	$10^9$	
Gold				$10^{-1}$	

Chart Provided By XICON

Figure Ref. - <https://www.parts-express.com/resistor-color-code-diagram>

# Switch with digital circuit

- A switch is a representative digital input device controlling electric current flow.
- Closed switch → ON
- Open switch → OFF
- Whenever a closed switch gets open (off), indeterminate voltage (floating state) can be generated.
- The digital circuit may be overloaded or damaged by the indeterminate voltage
- So, we need to stabilize the floating state.

# Let's make floating state

```
int pinNo = 8;
void setup() {
  Serial.begin(9600);
  pinMode(pinNo, INPUT);
}
void loop() {
  val = digitalRead(pinNo);
  Serial.print("val = ");
  Serial.println(val);
}
```

## Wiring setup ( x3 )

Arduino --- BreadBrd

5V --- (+)

GND --- (-)

D8 --- g15

Run this sketch!

What happens?  
(val)

Need to stabilize?

# Let's make pulldown/pullup resistors

```
int pinNo = 8;

void setup() {
  Serial.begin(9600);
  pinMode(pinNo, INPUT);
}

void loop() {
  val = digitalRead(pinNo);
  Serial.print("val = ");
  Serial.println(val);
}
```

## Resistor setup 1

*BreadBrd --- BreadBrd*  
i15 --- (-)

Open:

**pulldown  
to 0**

## Resistor setup 2

*BreadBrd --- BreadBrd*  
i15 --- (+)

Open:

**pullup  
to 1**

Run this sketch!

What happens?  
(val)

Need to stabilize?

# Let's open/close a (virtual) switch

```
int pinNo = 8;

void setup() {
  Serial.begin(9600);
  pinMode(pinNo, INPUT);
}

void loop() {
  val = digitalRead(pinNo);
  Serial.print("val = ");
  Serial.println(val);
}
```

## Resistor setup 1

*BreadBrd --- BreadBrd*  
i15 --- (-)

Close:

0 -> 1

## Wiring setup 1

*BreadBrd --- BreadBrd*  
f15 --- (+)

## Resistor setup 2

*BreadBrd --- BreadBrd*  
i15 --- (+)

## Wiring setup 2

*BreadBrd --- BreadBrd*  
f15 --- (-)

Close:

1 -> 0

Run this sketch!

What happens?  
(val)



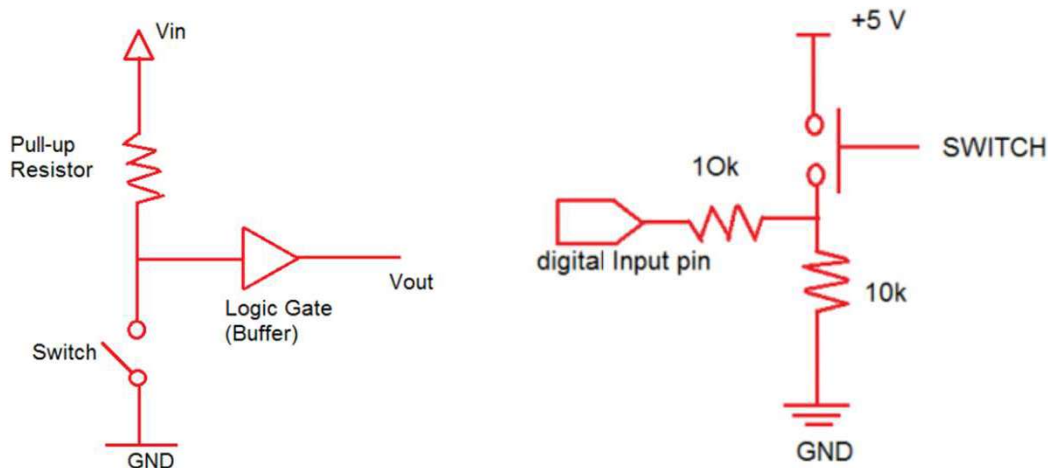
# Why Pullup/down resistor?

- **Stabilizing floating state to HIGH**

- **Pullup resistor** ensures a well-defined voltage (i.e. VCC, or logical high) across the remainder of the circuit when the switch is open.

- **Stabilizing floating state to LOW**

- Conversely, for a switch that connects to VCC, a **pull-down resistor** ensures a well-defined ground voltage (i.e. logical low) when the switch is open.



Pull-down Resistors	Pull-up Resistors
Less commonly used	More commonly used
Keeps the input "Low"	Keeps the input "High"
Connect between an I/O pin and ground, with an open switch connected between I/O and +Supply.	Connect between I/O pin and +supply voltage, with an open switch connected between I/O and ground.

Fig. ref -

<https://www.circuitbasics.com/pull-up-and-pull-down-resistors/>

# Pulldown vs. Pullup resistor

- Pulldown resistor
  - The resistor connected to ground (GND)
  - Resistor close to relatively low voltage (0V in Arduino)
- Pullup resistor
  - The resistor connected to the power supply (5V or 3.3V)
  - Resistor close to relatively high voltage (5V in Arduino)
- Pulldown and pullup resistor ensure a known state for a signal
  - HIGH (5V) or LOW (0V)
- The Arduino board has a **built-in** pull-up register
  - Use the built-in pull-up resistor by setting a digital input pin by pinMode() with INPUT\_PULLUP
  - Useful more resistance is required (or multiple buttons, etc).
  - Arduino Uno has 20K $\Omega$  pullup resistors.

# Let's try Digital input

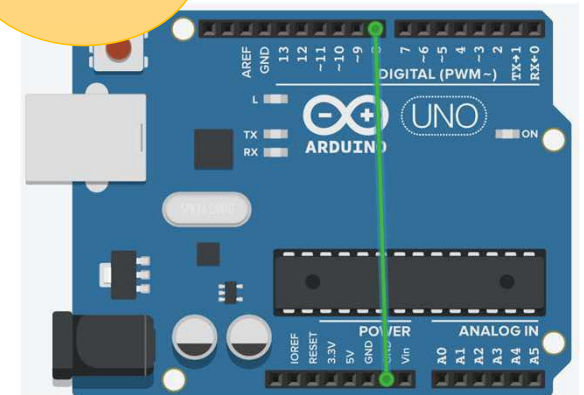
```
int dPin1 = 8;  
int val = 0;  
void setup() {  
    Serial.begin(9600);  
    pinMode(dPin1, INPUT);  
}  
void loop() {  
    val =   
    Serial.print("Digital read:");  
    Serial.println(val);  
}
```

Open Serial monitor.

What happens?

Connect D8 with  
GND, then read D8.

What happens?



# Let's try Pullup resistor

```
int dPin1 = 8, dPin2 = 7;
```

```
int val = 0;
```

```
void setup() {
```

```
  Serial.begin(9600);
```

```
  pinMode(dPin1, INPUT);
```

```
  pinMode(dPin2, [REDACTED])
```

```
}
```

```
void loop() {
```

```
  val = [REDACTED]
```

```
  Serial.print(dPin1);
```

```
  Serial.print(" pin Digital read:");
```

```
  Serial.println(val);
```

```
  val [REDACTED]
```

```
  Serial.print(dPin2);
```

```
  Serial.print(" pin Digital read:");
```

```
  Serial.println(val);
```

```
}
```

Disconnect D7, D8  
with GND, then read  
D7, D8.

What happens?

# Let's try Digital I/O with counter

```
int dPin1 = 7, dPin2 = 8;
int val = 0;
unsigned long cnt;
void setup() {
  Serial.begin(9600);
  pinMode(dPin1, OUTPUT);
  pinMode(dPin2, INPUT);
  cnt = 0;
}
void loop() {
  val = digitalRead(dPin2);
  Serial.print(cnt);
  Serial.print(")Digital read:");
  Serial.println(val);
  cnt++;
  if(cnt % 5 == 0) digitalWrite(dPin1, HIGH);
  else digitalWrite(dPin1, LOW);
  delay(1000);
}
```

Open Serial monitor.

What happens?

# Assignment: Button simulator

- Requirements

- Based on the example of digital I/O counter, write a sketch program as follows.
  - LED blinking every 5 seconds
  - Connect LED (+) with Arduino built-in pullup resistor
  - Connect LED (-) to GND
  - Connect LED (+) with D8
  - Read D8 every 1 second, and display "LED ON" if it is 1; or display "LED OFF" via serial monitor
- Write block-type comments in the top of your source code, which includes "your student no., your name, writing date, what you feel about this assignment, etc."

- Results

- (a source code file) sketch source code ( "***sketchfilename.ino***")
- (a Arduino board capture file) a photo capture showing how you setup your circuit (max. 1GB file).