임베디드컴퓨팅
Embedded Computing
(0009488)
Digital Output

2022년 2학기

정보기술대학 정보통신공학과
김 영 필
ypkim@inu.ac.kr

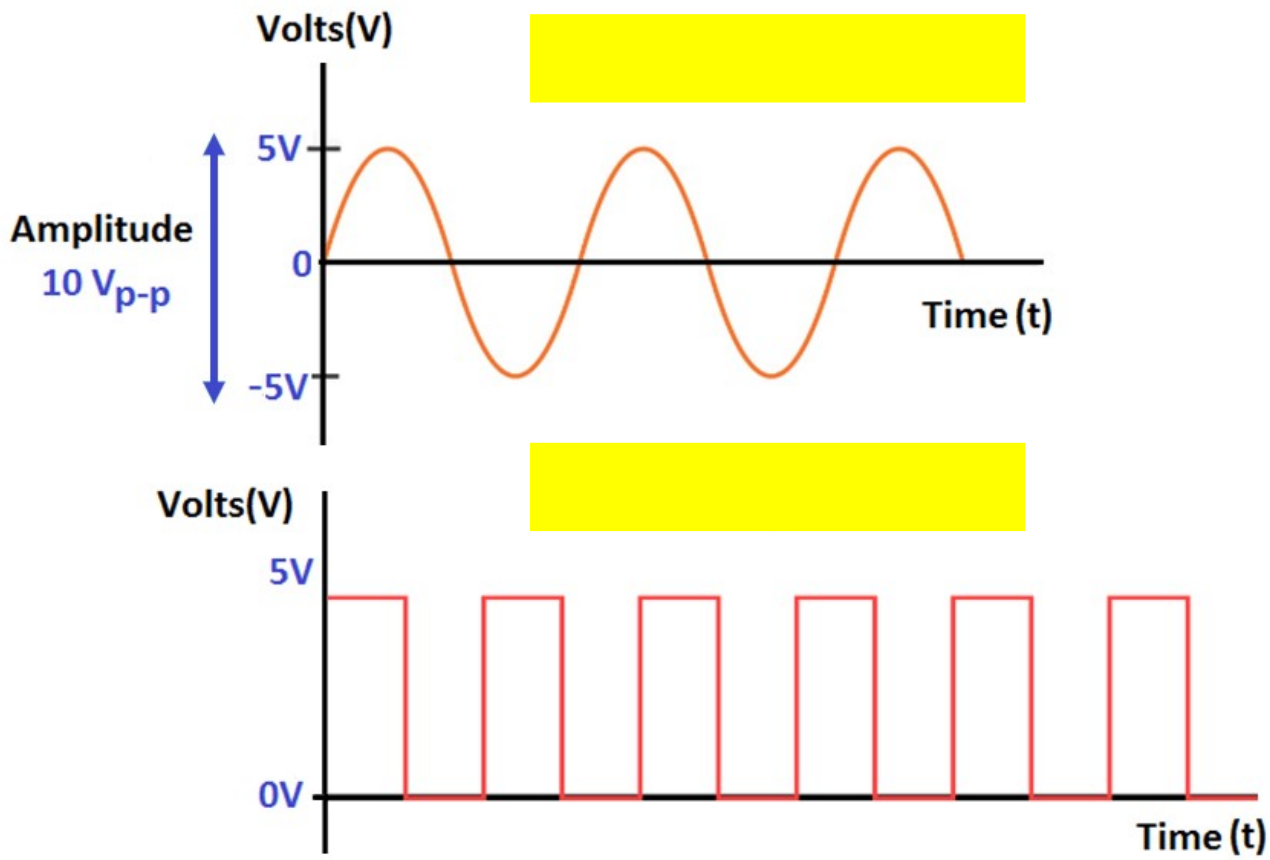
인천대학교
INCHEON NATIONAL UNIVERSITY

Analog vs. Digital

- Analog and digital signals are used to
- Common
  - The information (audio or video) is transformed into
- The difference in terms of translation of information
  - translated into electric pulses of varying amplitude.
    - Continuous as time varying
  - translated into into binary format (zero or one) where each bit is representative of two distinct amplitudes.
    - Discrete as time varying

Ref - https://www.diffen.com/difference/Analog_vs_Digital

E.g. Volts signal



Ref - https://instrumentationtools.com/what-are-analog-and-digital-signals-differences-examples/

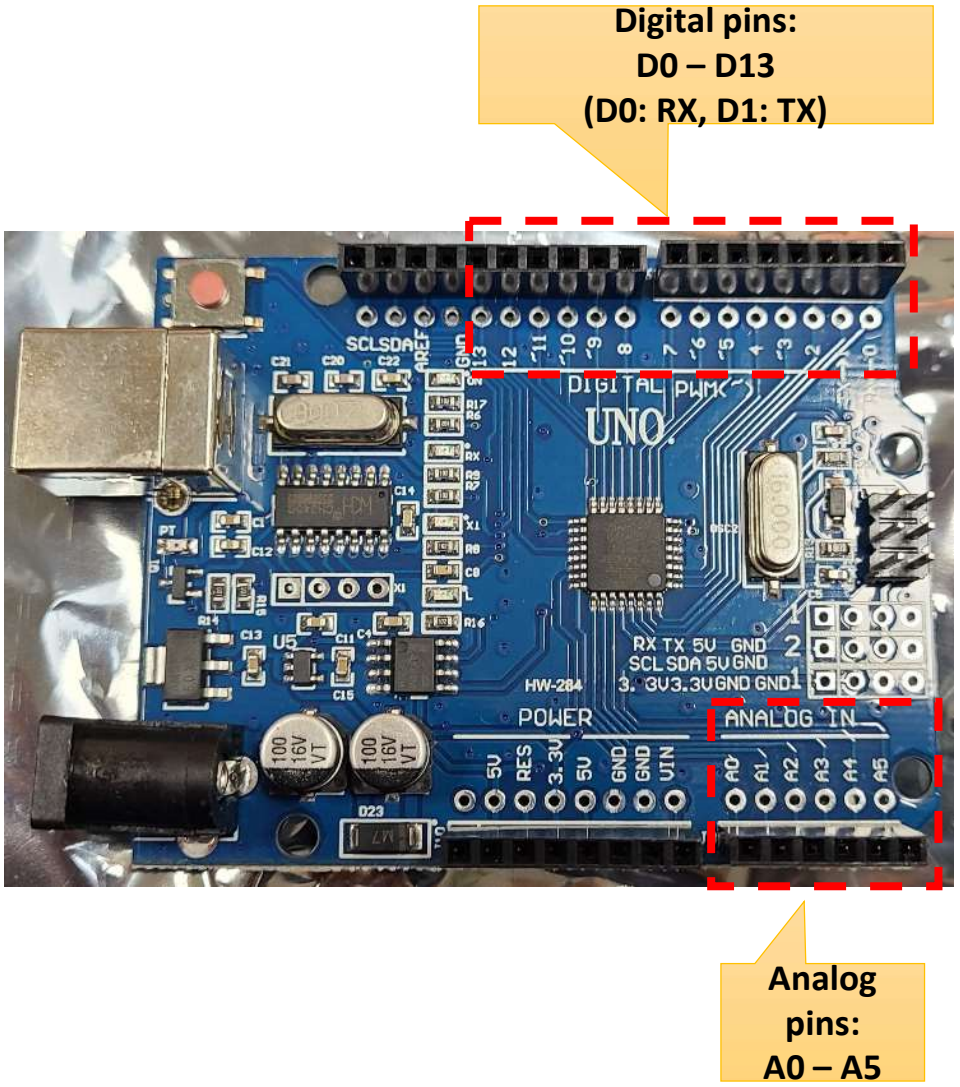
Comparison of Analog vs. Digital

Ref - https://www.diffen.com/difference/Analog_vs_Digital

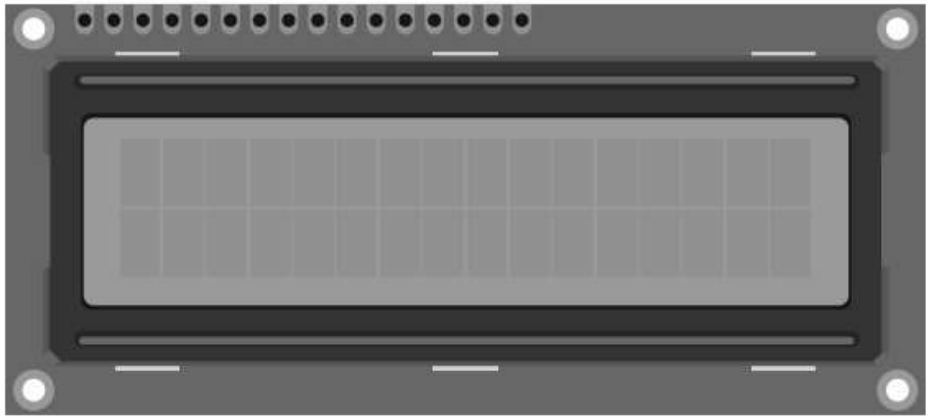
	Analog	Digital
Signal	which represents physical measurements.	s generated by digital modulation.
Waves	Denoted by	Denoted by
Information Representation	Uses continuous range of values	Uses discrete or discontinuous values
Example	Human voice in air, analog electronic devices.	Computers, CDs, DVDs, and other digital electronic devices.
Recording Technology	Records	and records them.
Data transmissions		
Response to Noise	More likely to get affected reducing accuracy	Less affected since noise response are analog in nature
Uses	Can be used in analog devices only. Best suited for audio and video transmission.	Best suited for Computing and digital electronics.
Applications	Thermometer	PCs, PDAs
Signal processing / Bandwidth		
Memory	Stored in the form of wave signal	Stored in the form of binary bit
Cost	Low cost and portable	Cost is high and not easily portable

How to measure signal related value in Arduino?

- Need to control, I/O pins (analog/digital) in Arduino
- How to control?
  - Use built-in functions
    - analogWrite(), analogRead()
    - pinMode(), digitalWrite(), digitalRead()
- What to control?
  - LED
- Need to determine control method
  - Analog or Digital



E.g. Output devices for Arduino



LCD(액정 디스플레이)

출력 방법	아날로그 출력	디지털 출력
사용하는 함수	analogWrite	pinMode와 digitalWrite
사용하는 전자 부품	• LED, 스피커 • 팬 • 일부 모터* 등	• LED, 스피커 • 적외선 리모컨용 LED 등

* 일반 모터는 아날로그로 제어할 수도 있고 디지털로 제어할 수도 있다.

그림 Source: 길벗, "모두의 아두이노"

LED

LED

-
  - Two-terminal electronic component that conducts current primarily in one direction (+ → -)

- **(발광 다이오드)**
  - Glow in
  - Different colors of light depending on the chemical
  - Infrared LED for remote control and UV LED for sterilization are also available.
  - Connect to the I/O pin to see data bits

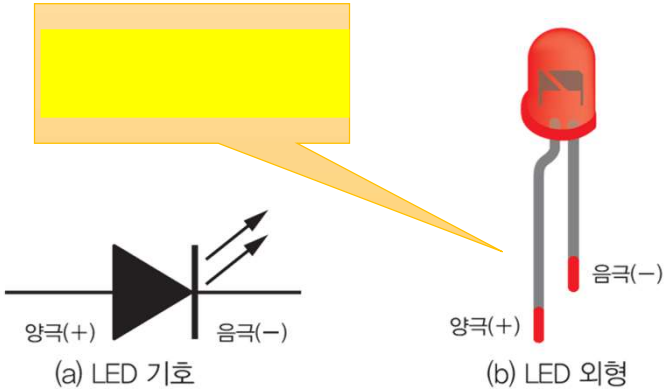


그림 Source: 한빛미디어,
"따라하면서 배우는 아두이노"

LED on Arduino

- For digital output, connect two pins of LED into D13 and GND
  - Long pin (Anode) ➔ D13
  - Short pin (Cathode) ➔ GND
- Let's re-run blink.ino!

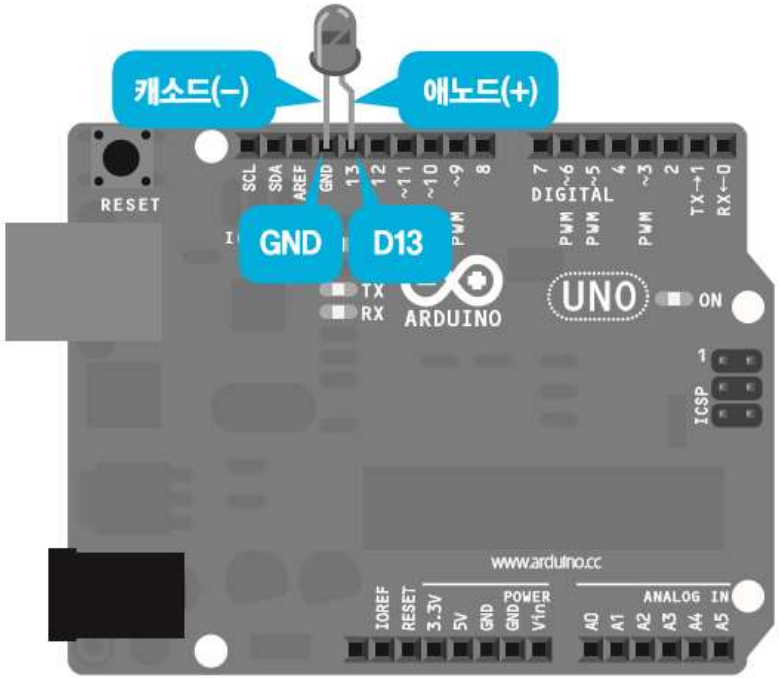


그림 Source: 길벗, "모두의 아두이노"

Blink, again

```
int led = 13;

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

void loop() {
      [redacted]
}
```

Let's change delay time (ms) :

[redacted]

Brightness

```
int led = 13;

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

void loop() {
    digitalWrite(led, HIGH);

    digitalWrite(led, LOW);
}
```

Let's declare two delay variables:

Let's change on_delay, off_delay:

Check the brightness. Can you catch the difference?

Brightness with loop

```
int led = 13;
void setup() {
      pinMode(led, OUTPUT);
}

void loop() {
                                          ;
}
```

Let's declare two new local variable:

Then, remove old delay variables.

Add two **for-loops**:

What can you observe?

Mechanism of LED brightness

- We use two variables - high_delay, i - to control LED brightness
  - high_delay =
  - i =

- Turn on LED during high_delay ms, and turn off during 9-high_delay ms.
  -
  -

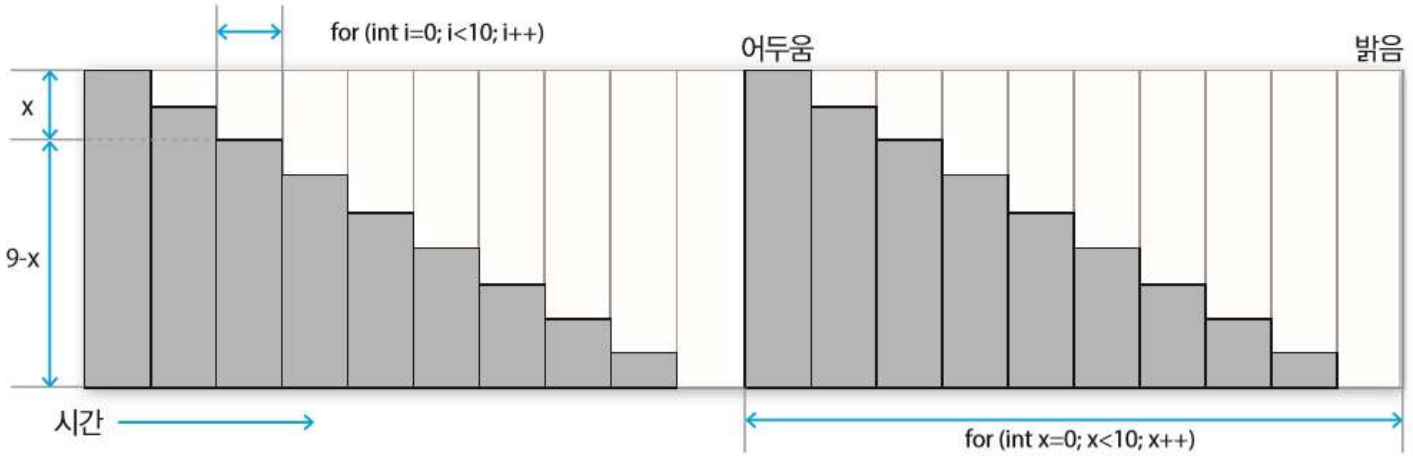
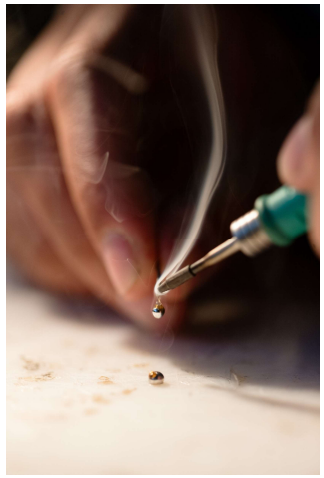


그림 Source: 길벗, "모두의 아두이노"

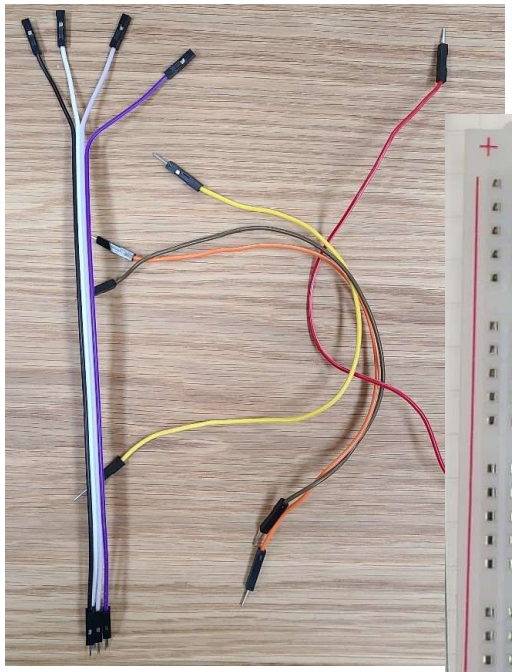
Assignment : bright and dark

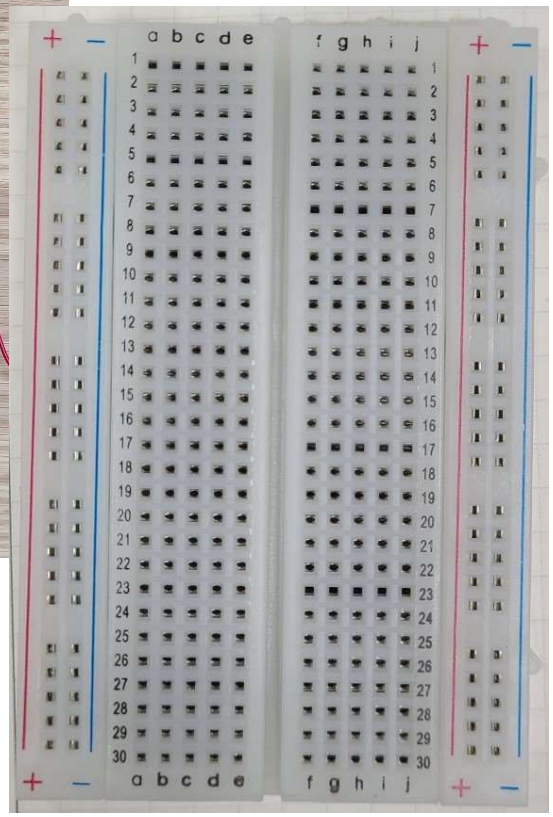
- Write sketch program according to the requirements
- Requirements
  - Change 10 times the brightness of LED
    - Make LED bright gradually (10 steps)
    - Make LED dark gradually (10 steps)
  - Use iteration statements (for-loop or do-while)
  - Implement the above in “void loop()”

Let’s use breadboard



- Electronic work requires assembling electronic components and **connecting or soldering** between them with
- For testing, we can simply connect electronic components with just a **breadboard and jumper wires.**
- Easy to disassemble or assemble components to make electronic circuits





Standard breadboard

- Standard breadboards have the **same pitch (2.54 mm spacing, 0.1 inches)** with pins that can be plugged in.
- Connected vertically and horizontally
- The **two rows on the top and bottom** are **connected**
  - Mainly used to connect power and GND.
- The **middle part** is divided into **upper and lower parts**.
  - The upper and lower parts are
  - Useful for IC chip
    - Electronic components with a structure that both sides are not connected

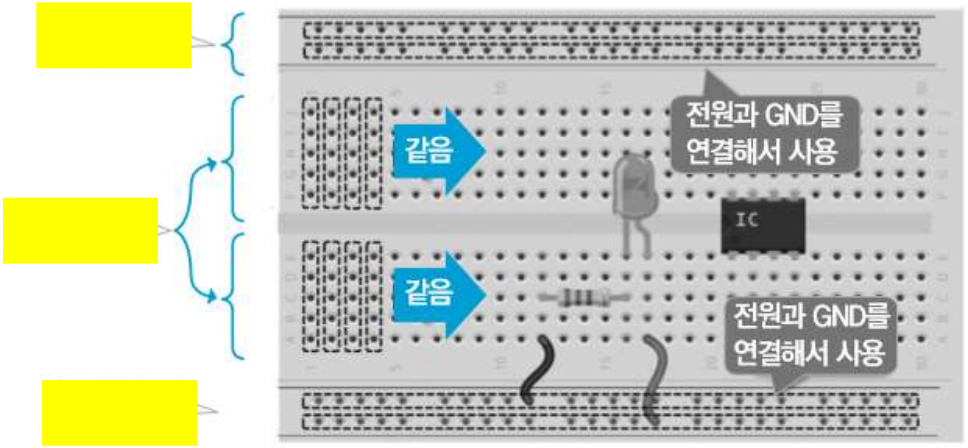


그림 Source: 길벗, "모두의 아두이노"

Basic use of breadboard

- **Electricity flow** has
  - When **a voltage** is applied, the **(electric) current** flows to the **ground (GND)**
  - When the wire is disconnected, no current flows.
- If there is **an electronic component in the middle of the flow**, current flows into the of the electronic component and flows out toward the
- Some electronic components have polarity and some do not.
  - **Polarity:** the positive and negative state are predetermined.
  - If you connect the power side and the GND side in reverse, some electronic parts can be broken.

Lab: LED on breadboard

- Preparation
  - Four jumper cables
  - A breadboard
  - LED,
  - Resistance (220 ohm)
- Installation steps
  - Set jumper wires
    - Arduino (GND) ➔
    - Arduino (D13) ➔
    - Breadboard (-) ➔
    - Breadboard (+) ➔

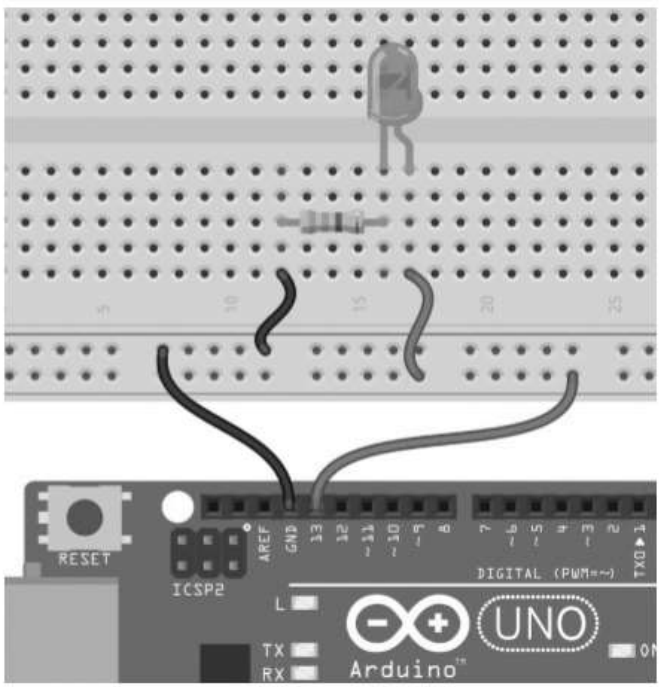


그림 Source: 길벗, "모두의 아두이노"

Upload blink.ino

Is the LED on?