

# 소프트웨어 프로젝트 2

## 아두이노 기초 (1)

2024년 2학기

국민대학교  
소프트웨어학부/인공지능학부  
주용수, 최진우, 한재섭, 허대영  
{ysjoo, jaeseob, jnwochoi, dyheo}@kookmin.ac.kr

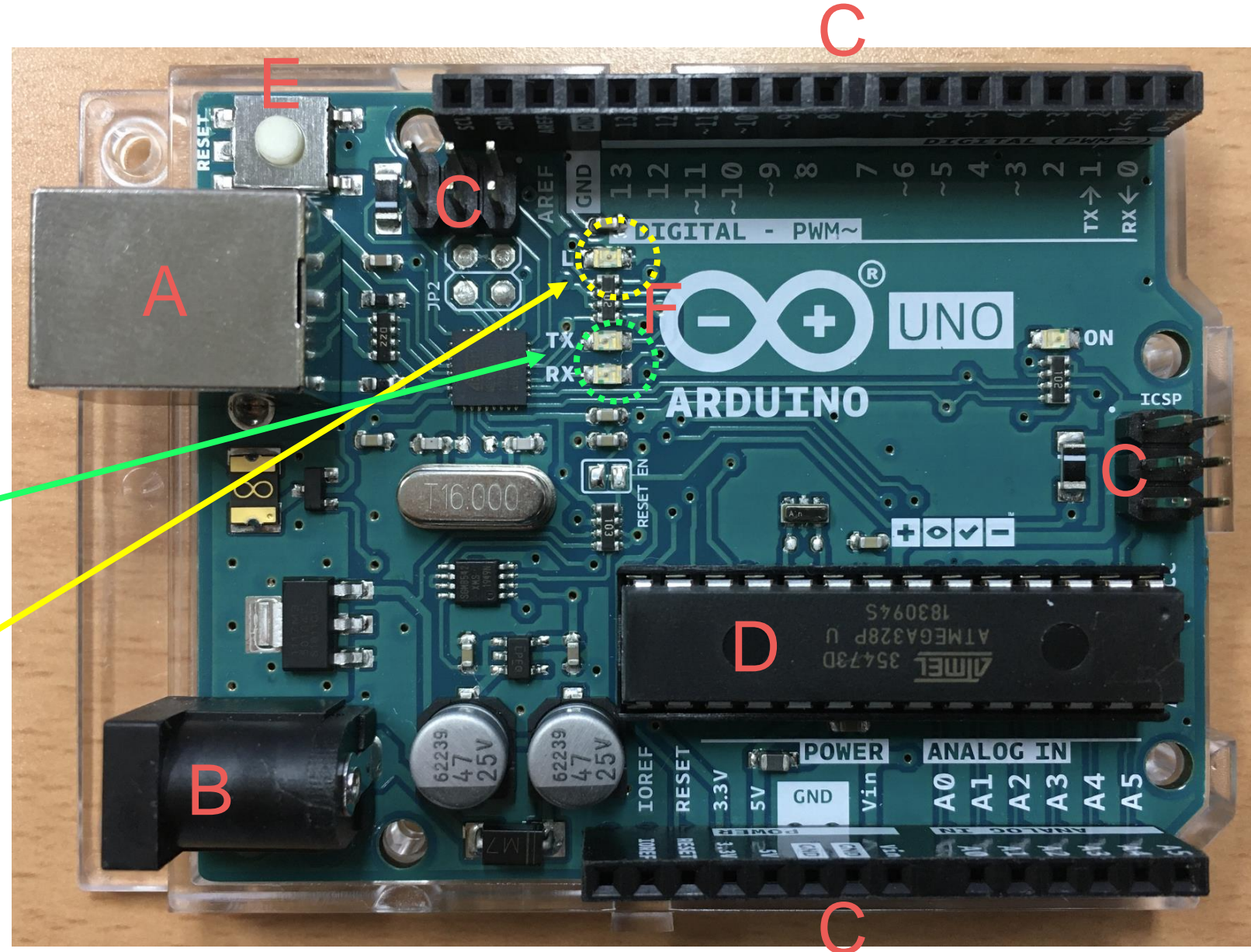
# 아두이노 (Arduino)

- IDI(Interaction Design Institute Ivera, Italy), 2005년 개발
  - 인터랙티브 디자인 교육 목적
- 하드웨어
  - 16MHz 8bit 마이크로컨트롤러 (Micro-Controller, MCU)
  - 32KB 플래시(10K Program/Erase cycle) + 2KB SRAM
  - USB 인터페이스: 호스트 컴퓨터와의 통신
  - 범용 입출력 포트 및 ADC(아날로그-디지털 변환기) 제공
  - 다양한 직렬통신 인터페이스 지원: SPI, I<sup>2</sup>C 등
- 소프트웨어
  - 통합 개발환경(IDE) 제공
  - Sketch: 아두이노에서 실행되는 프로그램 (C++ 문법을 따름)
  - 다양한 주변장치 라이브러리 제공

# 아두이노 회로기판 구조

- 주요 구성요소

- A: 호스트 컴퓨터 연결용 USB
- B: 전원공급용 DC 어댑터 단자(7~12V)
- C: 각종 입출력 포트
- D: ATmega328P 마이크로컨트롤러
- E: 초기화(reset) 버튼
- F: On-board LEDs (L, TX, RX)
  - TX, RX: Serial 통신 송수신시 발광.
  - L: digital I/O 13번에 연결되어 간단한 LED 제어 테스트 가능.



# 아두이노 통합개발환경

- 통합개발환경 (IDE, integrated development environment)
  - 아두이노 소프트웨어 개발을 위해  
호스트 컴퓨터(PC)에서 실행하는 프로그램
  - 코딩, 컴파일, 실행파일 전송, 디버깅 등의 작업 수행 가능
  - 아두이노 보드에 내장된 serial 터미널을 통해
    - 아두이노 보드에서 실행하는 프로그램의 Text 출력을 모니터링
      - 정해진 규칙을 따르면, 그래프로도 그려줌
    - PC에서의 키보드 입력을 아두이노 보드에 전달 가능

# 아두이노 통합개발환경 (Sketch)

main()

setup()  
loop()

```
Arduino  파일 편집 스케치 툴 도움말
hello_world | 아두이노 1.8.11

hello_world
void setup() {
  //Initialize serial and wait for port to open:
  Serial.begin(115200);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB
  }
}

void loop() {
  Serial.println("Hello World!");
  delay(1000);
}

컴파일 완료.

스케치는 프로그램 저장 공간 1634 바이트(5%)를 사용. 최대 32256 바이트.
전역 변수는 동적 메모리 200바이트(9%)를 사용, 1848바이트의 지역변수가 남음. 최대는 2048 바이트.

7 Arduino Uno on /dev/cu.usbmodem146201
```

다음 시간에 실습 예정



# C/C++ 문법/자료형

- 각 문장의 끝에 세미콜론(;) 필요
- 저장하고 싶은 데이터에 따라 자료형 명시
  - 1 byte: **bool**, (**unsigned**) **char**
  - 2 bytes: (**unsigned**) **int**
    - **int a;** // 저장 가능한 숫자: -32768 ~ 32767 ( $-2^{15} \sim 2^{15} - 1$ )
    - **unsigned int a;** // 저장 가능한 숫자: 0 ~ 65535 ( $0 \sim 2^{16} - 1$ )
  - 4 bytes: (**unsigned**) **long**
  - 참고: <https://www.arduino.cc/reference/en/>
- 주석 (comment)
  - **int a; // this is a comment.**
  - **/\*  
Configurable parameters  
\*/**

# Sketch의 소스코드 기본 구조

(아래 코드에는 에러가 포함되어 있음)

Line

```
01  #define PIN_LED 13
02  unsigned int count, toggle;
03
04  void setup() {
05      pinMode(PIN_LED, OUTPUT)
06      Serial.begin(115200); // Initialize serial port
07      while (!Serial) {
08          ; // wait for serial port to connect.
09      }
10      Serial.println("Hello World!");
11      count = toggle = 0;
12      digitalWrite(PIN_LED, toggle); // turn off LED.
13  }
14
15  void loop() {
16      Serial.println(++count);
17      toggle = toggle; // toggle LED value.
18      digitalWrite(PIN_LED, toggle); update LED status.
19      delay(1000); // wait for 1,000 milliseconds
20  }
```

A

B

C

# Sketch 기본 구조 (A 영역)

- A: 전처리 및 전역변수 선언부
  - 각종 주변장치나 확장기능에 대한 헤더파일 추가
  - **#define, typedef** 등의 각종 전처리 선언
  - 각종 상태값, 제어값들에 대한 전역변수 선언
- 코드 설명
  - Line 01: 디지털 입출력 핀 중 13번을 PIN\_LED로 사용.
  - Line 02: **count, toggle**를 전역변수로 선언.  
자료형은 부호없는 정수형(unsigned integer).

```
01 | #define PIN_LED 13  
02 | unsigned int count, toggle;
```

- |  |   |
|--|---|
| <ul style="list-style-type: none"><li>▪ 전역변수<ul style="list-style-type: none"><li>▪ 함수 외부에서 선언</li><li>▪ 함수 내부에서 접근 가능</li><li>▪ 함수 종료시에도 값이 유지됨</li></ul></li></ul> | <ul style="list-style-type: none"><li>▪ 지역변수<ul style="list-style-type: none"><li>▪ 함수 내부에서 선언</li><li>▪ 선언된 함수 내부에서만 접근 가능</li><li>▪ 함수 종료시 값이 유지되지 않음</li></ul></li></ul> |
|--|---|



# Sketch 기본 구조 (B 영역)

- B: **setup()** 함수
  - 아두이노 전원을 켜 직후 또는 리셋 버튼을 누른 직후 자동으로 실행
  - 최초 1회만 실행
  - 각종 변수 및 장치 초기화 용도
- 코드 설명 (여러 포함됨)
  - Line 05: PIN\_LED(=13번 핀)을 출력 모드로 설정
  - Line 06: 시리얼 포트를 115,200 bps(bits per second) 속도로 초기화
  - Line 07-09: 장치 초기화 완료까지 대기
  - Line 10: 메시지 출력
  - Line 11: 변수 초기화
  - Line 12: LED 상태 초기화

```
04 void setup() {
05     pinMode(PIN_LED, OUTPUT)
06     Serial.begin(115200); // Initialize serial port
07     while (!Serial) {
08         ; // wait for serial port to connect.
09     }
10     Serial.println("Hello World!");
11     count = toggle = 0;
12     digitalWrite(PIN_LED, toggle); // turn off LED.
13 }
```

# Sketch 기본 구조 (C 영역)

- C: `loop()` 함수
  - `setup()` 함수가 실행된 후 계속 반복하여 실행됨
  - 프로그램의 핵심 구현 내용이 위치함
  - 계속 변화하는 외부 입력에 반응하여 출력을 갱신하는 인지-판단-제어 동작 구현에 적합

```
15 void loop() {  
16     Serial.println(++count);  
17     toggle = toggle; // toggle LED value.  
18     digitalWrite(PIN_LED, toggle); update LED status.  
19     delay(1000); // wait for 1,000 milliseconds  
20 }
```

- 코드 설명 (어려 포함됨)
  - Line 16: **count** 변수를 1만큼 먼저 증가시키고 나서 해당 값을 출력
  - Line 17: LED 상태 변수 **toggle**의 값을 새로운 값으로 갱신
  - Line 18: **toggle** 값을 PIN\_LED(13번) 핀으로 출력(1: 켜짐, 0: 꺼짐)
  - Line 19: 1,000 밀리초(= 1초)만큼 대기

# 사용자 정의 함수 추가도 가능

- 코드가 매우 긴 경우
  - `loop()` 내부에 모든 코드를 기입할 경우 가독성 저하 및 디버깅 어려움
  - 코드를 기능별로 분할하여 여러 개의 함수로 구성
  - `setup()` 또는 `loop()` 함수 내부에서 필요한 함수를 호출
- 함수 추가 예 (에러 포함됨)

```
...
void loop() {
    Serial.println(++count);
    toggle = toggle_state(toggle); // toggle LED value.
    digitalWrite(PIN_LED, toggle); // update LED status.
    delay(1000); // wait for 1,000 milliseconds
}

int toggle_state(int toggle) {
    return toggle;
}
```