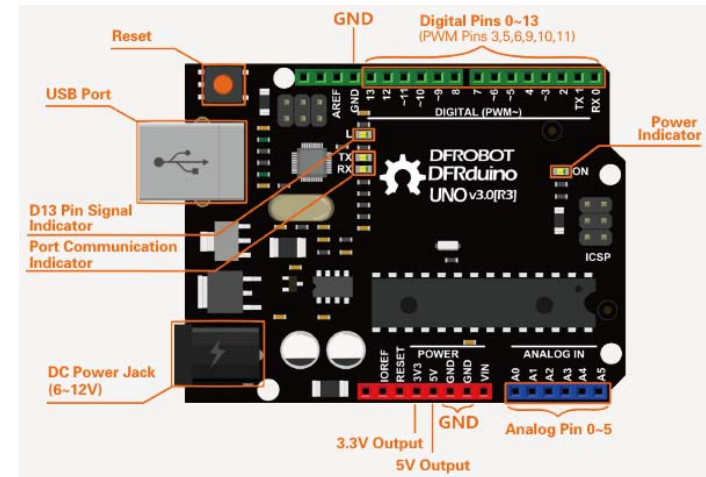


### 3. Arduino 사용하기

### Arduino UNO (DFRduino UNO R3)



2

### Arduino Uno R3 스펙

#### SPECIFICATION

- Microcontroller: ATmega328 (DIP Package)
- Operating Voltage: 5V
- Input Voltage (recommended): 7 ~ 12V
- Input Voltage (limits): 6 ~ 20V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6
- DC Current per I/O Pin: 40 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB of which 2KB used by bootloader
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Clock Speed: 16 MHz
- Size: 75 x54 x15 mm (2.95 x2.13 x0.59")
- Environment Friendly: Rohs Compliance

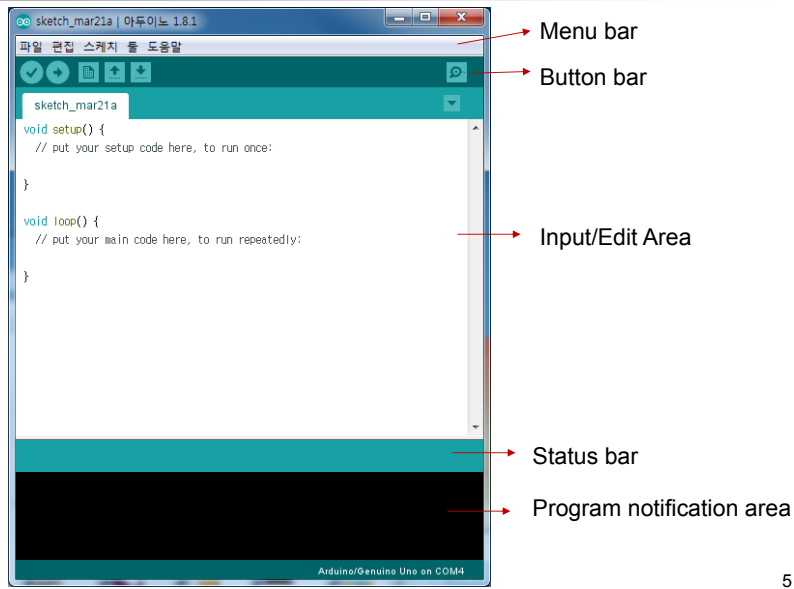
3

### 시작하기

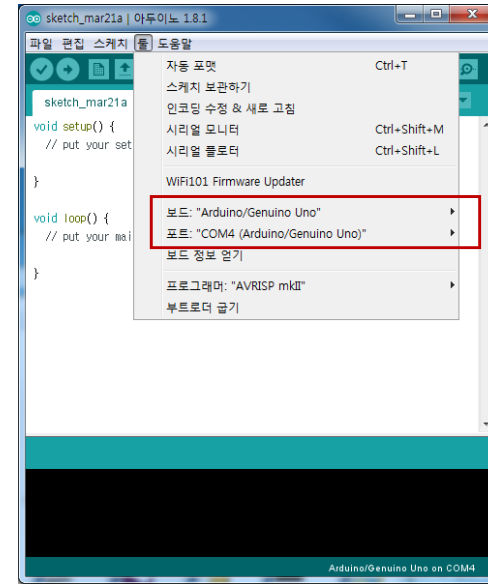
- Arduino 웹사이트
  - <http://arduino.cc>
  - Lean Arduino 링크
  - Install Arduino Desktop IDE : Windows 버전 다운로드 후 설치
- 실습 키트 웹사이트
  - <http://dfrobot.com>
  - Arduino → Arduino Kits → Beginner Kit for Arduino
  - Documents → WIKI → 실습 tutorial이 있음  
→ Beginner Kit Tutorial (pdf 파일)

4

## Arduino IDE



## Serial Port와 Board 선택

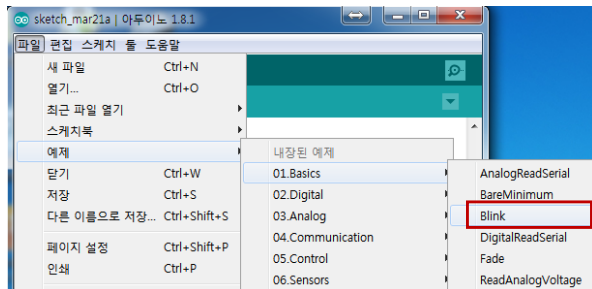


Board : Arduino Uno  
Port : COM4(Arduino Uno)

## Arduino 사용하기

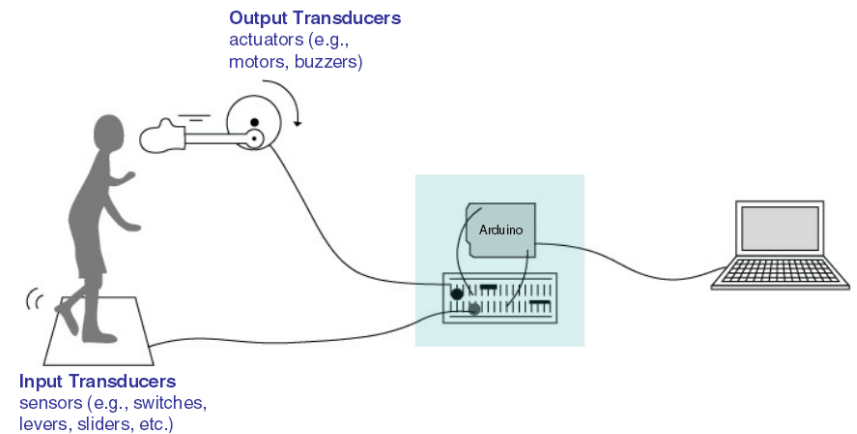
- Sketch 프로그램 작성
- Compile : 스케치 → 확인/컴파일(Ctrl-R)
- Upload : 스케치 → 업로드 또는

### ■ 예제 프로그램 실행 - Blink



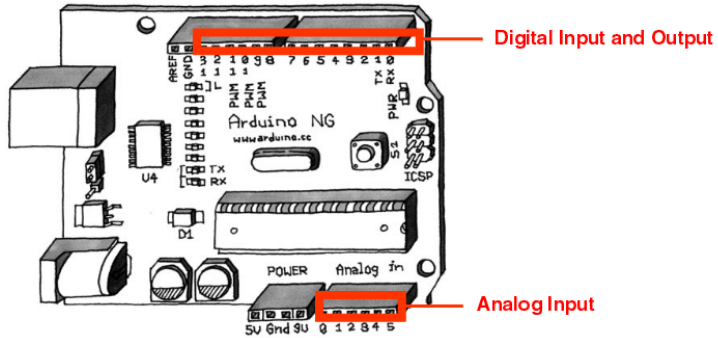
LED "L" 깜박임  
- D13 핀과 연결됨

## 입력/출력



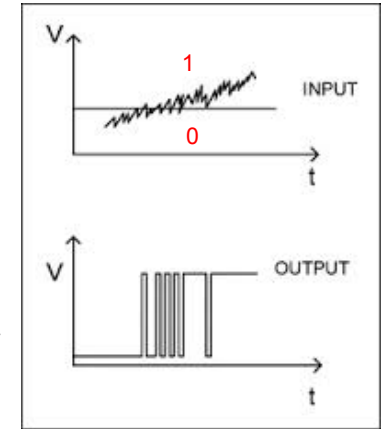
## I/O 핀

Two states (binary signal) vs. multiple states (continuous signal)



## Digital I/O

- `pinMode(pin, mode)`
  - pin을 INPUT, OUTPUT, 또는 INPUT\_PULLUP으로 설정
- `digitalRead(pin)`
  - pin 값을 입력
  - return값: HIGH(1) 또는 LOW(0)
- `digitalWrite(pin, value)`
  - pin에 value를 출력
  - value : HIGH 또는 LOW
- 전기적 특성
  - output pin은 40mA 전류 제공가능
  - input pin에 HIGH를 제공하려면 20kΩ pullup 저항을 설치함
- language reference: <http://www.arduino.cc/en/Reference>



## 프로그램 구조

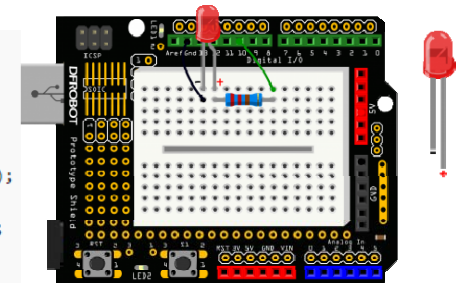
- 프로그램 기본 구조
 

```
void setup() {
    // put your setup code here, to run once:
}

void loop() {
    // put your main code here, to run repeatedly:
}
```

## Lab 1. Blinking a LED

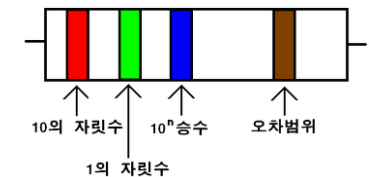
```
int ledPin = 10;
void setup() {
    pinMode(ledPin, OUTPUT);
}
void loop() {
    digitalWrite(ledPin, HIGH);
    delay(1000);
    digitalWrite(ledPin, LOW);
    delay(1000);
}
```



$22 \times 10^1 = 220 \Omega$

- `delay(ms)`
- `delayMicroseconds(us)`
- 저항 값 읽기

0	1	2	3	4	5	6	7	8	9	5%	10%
흑	갈	빨	주	노	녹	파	보	회	흰	금	은



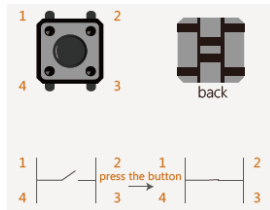
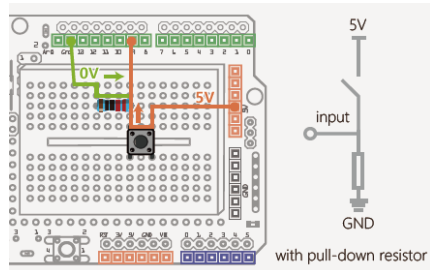
## Lab 2. Use a push-button to turn ON/OFF LED

```
int ledPin = 10;
int button = 9;

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

void loop() {
  int state = digitalRead(button);

  if (state==HIGH)
    digitalWrite(ledPin,HIGH);
  else
    digitalWrite(ledPin,LOW);
}
```



- button 입력
  - 누르면 : HIGH
  - 떴면 : LOW (pull-down 저항)

13

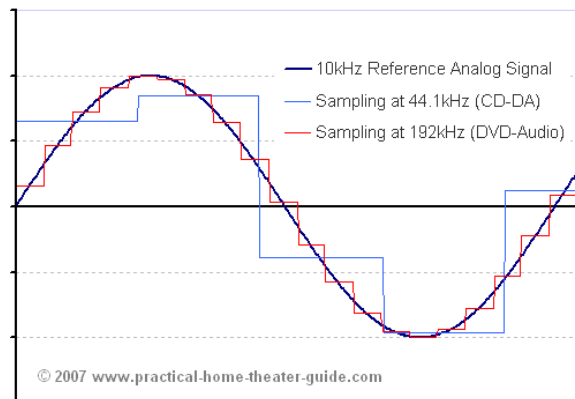
## Exercise

- LED의 초기 상태는 off
- Button을 5초 이상 누를 때마다(HIGH) LED가 on/off 상태를 변경함
- millis()
  - 현재 프로그램이 실행된 이후의 경과시간 (ms 단위)
- micros()
  - 현재 프로그램이 실행된 이후의 경과시간 (us 단위)
  - 16 MHz Arduino board에서 4us resolution을 가짐

14

## Analog Input

- Analog Input – continuous signal
- ADC(Analog to Digital converter) - digital 값으로 변환



15

## Analog Input (계속)

- resolution
  - 입력 신호를 양자화하는 데 사용되는 전압레벨의 개수
  - Arduino는 1024 states (10 비트)를 사용함
    - 최소 측정 전압 레벨 =  $5V / 1024 = 4.9 \text{ mV}$
- sample rate
  - 최대 sample rate : 1초당 10000번
- analogRead(pin)
  - 아날로그 pin으로부터 양자화된 값을 읽음 (10 bit) : **0 ~ 1023**
  - 아날로그 입력 소요시간 약 100us → 1초에 최대 10000번 입력 가능

16



## Lab 4. LED fade out

- Lab 1과 같은 회로 – pin 10을 PWM 출력으로 사용

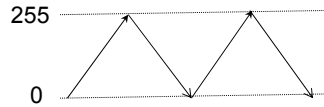
```

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

void loop(){
  fadeOn(1000);
  fadeOff(1000);
}

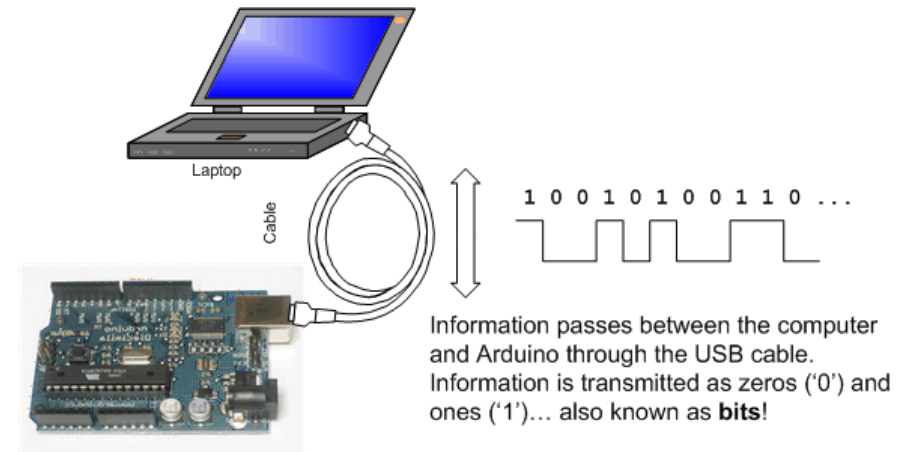
void fadeOn(unsigned int time){
  for (byte value = 0 ; value < 255; value+=5){
    analogWrite(ledPin, value);
    delay(time/(255/5));
  }
}

void fadeOff(unsigned int time){
  for (byte value = 255; value >0; value-=5){
    analogWrite(ledPin, value);
    delay(time/(255/5));
  }
}
    
```



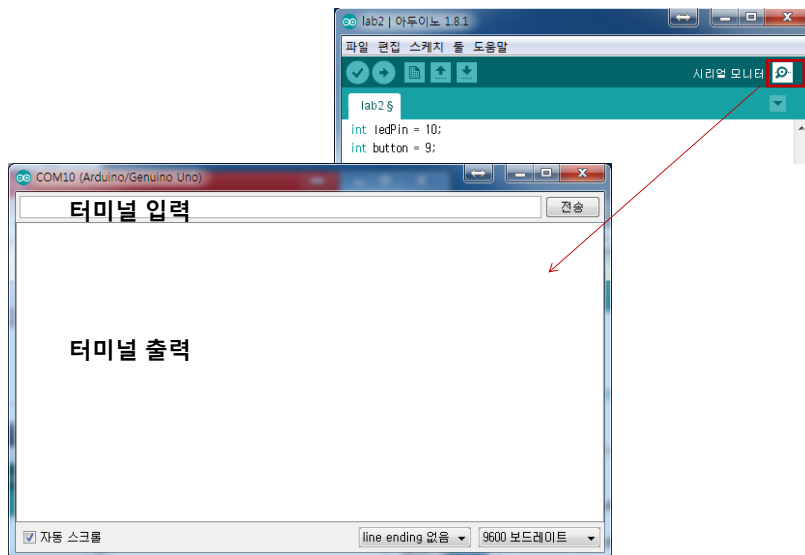
21

## Serial Communication (Terminal)



22

## Serial Monitor 사용



23

## Serial commands

- Serial.begin(speed) baud rate 설정
- Serial.print(value) Serial.print(value, base)  
Serial.println(value) Serial.println(value, base)
- Serial.read()
- Serial.available() available 입력 바이트 수
- Serial.write()
- Serial.parseInt() 정수입력, 정수가 아닌 입력은 skip

24

## Lab 5. 터미널 입출력 포함

### lab 4와 같은 회로

```

int ledPin = 10;
int interval = 1000;
void setup() {
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
  do {
    interval = Serial.parseInt();
    while (interval==0);
    Serial.println(interval);
  }
}

void loop(){
  fadeOn(interval);
  fadeOff(interval);
}

void fadeOn(unsigned int time){
  for (byte value = 0 ; value < 255; value+=5){
    Serial.println(value);
    analogWrite(ledPin, value);
    delay(time/(255/5));
  }
}

void fadeOff(unsigned int time){
  for (byte value = 255; value >0; value-=5){
    Serial.println(value);
    analogWrite(ledPin, value);
    delay(time/(255/5));
  }
}

```

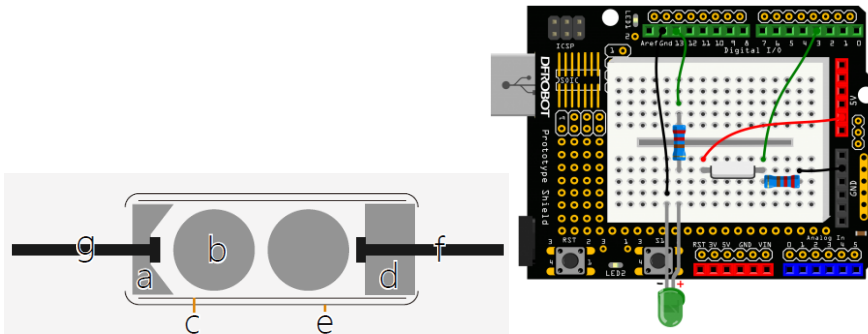
25

## Interrupt

- Interrupt를 지원하는 Digital pins – 보드마다 다름
  - Uno : pin 2, 3
- attachInterrupt(interrupt, ISR, mode)
- detachInterrupt(interrupt)
- interrupts()
- noInterrupts()

26

## Lab 6. 인터럽트 – Vibration 감지



- a. Bronze Cover
- b. Bronze Bead
- c. Bronze Pipe
- d. PC set
- e. Heat-shrinkable Pipe
- f. Bronze Conductive Pin
- g. Phosphor Copper Pinch Cock

Fig. 8-2 Vibration Sensor Diagram

27

```

int SensorLED = 13; //LED PIN
int SensorINPUT = 3; //Connect the sensor to digital Pin 3 which is Interrupts 1
unsigned char state = 0;

void setup() {
  pinMode(SensorLED, OUTPUT);
  pinMode(SensorINPUT, INPUT);

  // Trigger the blink function
  // when the falling edge is detected
  attachInterrupt(1, blink, RISING);
}

void loop(){
  if(state!=0){
    state = 0;
    digitalWrite(SensorLED,HIGH);
    delay(500);
  }
  else
    digitalWrite(SensorLED,LOW);
}

void blink(){ //Interrupts function
  state++;
}

```

28