

MakeUp
아두이노

#MakeUp아두이노 #04

아두이노 기초

아두이노 기초

delay() 없이 시간 측정

millis() 함수

아두이노 보드가 현재 프로그램을 돌리기 시작한 후 지난
밀리 초 숫자를 반환해준다.

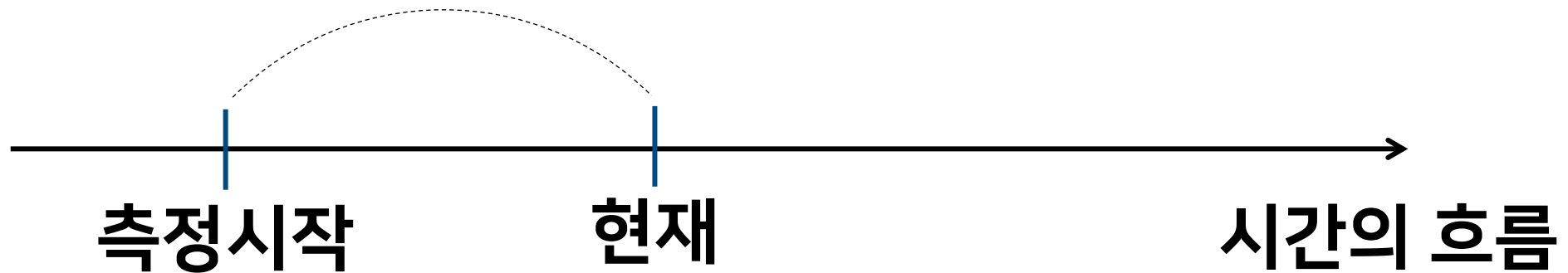
약 50일이 지나면 오버플로우 되어 다시 0부터 시작.

아두이노 기초

delay() 없이 시간 측정

시간의 측정

측정된 시간 = (현재시간) - (측정 시작 시간)



아두이노 기초

delay() 없이 시간 측정

```
1 unsigned long prevMillis = 0; // 측정 시작시간을 저장할 변수
2
3 void setup() {
4   // pinMode() 등 설정
5 }
6
7 void loop() {
8   unsigned long currentMillis = millis();
9
10  if (currentMillis - prevMillis >= 1000 /* 측정한 시간이 1000ms이상 경과했는가 */ ) {
11    prevMillis = currentMillis; // 기준점(측정 시작시간)을 현재로 리셋
12
13    /* 특정 시간마다 실행되기 원하는 코드를 작성 */
14  }
15 }
```

아두이노 기초

Led On/Off 동작 시 시간을 다르게 하려면?

측정 시작 시간에서 경과한 시간과 LED의 상태를 동시에 검사

변수 선언 및 초기화

```
1 const int ledPin = 13; // LED 핀 번호 상수 정의
2 unsigned long prevLedMillis = 0; // 측정 기준시각 저장할 변수
3 int ledState = LOW; // LED의 현재 상태 저장할 변수
4
5 void setup() {
6     pinMode(ledPin, OUTPUT);
7 }
```

아두이노 기초

Led On/Off 동작 시 시간을 다르게 하려면?

경과시각 과 LED상태에 따른 조건문 (loop 내)

```
9 void loop() {
10     unsigned long currentMillis = millis(); // 현재 시간을 구한다.
11
12     if (currentMillis - prevLedMillis >= 500 && ledState == LOW) {
13         /* 경과 시간이 500ms이고 (그리고 = AND = &&) LED가 꺼져있는 상태라면 (= LED꺼짐 500ms)*/
14         prevLedMillis = currentMillis;
15
16         ledState = HIGH;
17         digitalWrite(ledPin, ledState);
18     } else if (currentMillis - prevLedMillis >= 2000 && ledState == HIGH) {
19         /* 경과 시간이 2000ms이고 LED가 켜져있는 상태라면 (= LED켜짐 2000ms)*/
20         prevLedMillis = currentMillis;
21
22         ledState = LOW;
23         digitalWrite(ledPin, ledState);
24     }
25 }
```

아두이노 기초

Led On/Off 동작 시 시간을 다르게 하려면?

Buzzer를 추가해보자 - 필요한 변수들을 추가해준다.

```
1 const int ledPin = 13; // LED 핀 번호 상수 정의
2 const int bzPin = 12;  // Buzzer 핀 번호 상수 정의
3
4 unsigned long prevLedMillis = 0; // LED 측정 기준시각 저장할 변수
5 unsigned long prevBzMillis = 0;  // Buzzer 측정 기준시각 저장할 변수
6 int ledState = LOW; // LED 상태
7 int bzState = LOW;  // Buzzer 상태
8
9 void setup() {
10     pinMode(ledPin, OUTPUT); // Led핀 출력으로 설정
11     pinMode(bzPin, OUTPUT);  // Buzzer핀 출력으로 설정
12 }
```

아두이노 기초

Led On/Off 동작 시 시간을 다르게 하려면?

기존 LED의 On/Off 를 제어하는 로직

```
14 void loop() {  
15     unsigned long currentMillis = millis();  
16  
17     if (currentMillis - prevLedMillis >= 500 && ledState == LOW) {  
18         prevLedMillis = currentMillis;  
19  
20         ledState = HIGH;  
21         digitalWrite(ledPin, ledState);  
22     } else if (currentMillis - prevLedMillis >= 2000 && ledState == HIGH) {  
23         prevLedMillis = currentMillis;  
24  
25         ledState = LOW;  
26         digitalWrite(ledPin, ledState);  
27     }  
28 }
```


아두이노 기초

Led On/Off 동작 시 시간을 다르게 하려면?

Buzzer 제어 로직을 별도로 추가해준다.

```
28
29  if (currentMillis - prevBzMillis >= 3000 && bzState == LOW) {
30      prevBzMillis = currentMillis;
31
32      bzState = HIGH;
33      digitalWrite(bzPin, bzState);
34  } else if (currentMillis - prevBzMillis >= 1000 && bzState == HIGH) {
35      prevBzMillis = currentMillis;
36
37      bzState = LOW;
38      digitalWrite(bzPin, bzState);
39  }
40 }
```

아두이노 기초

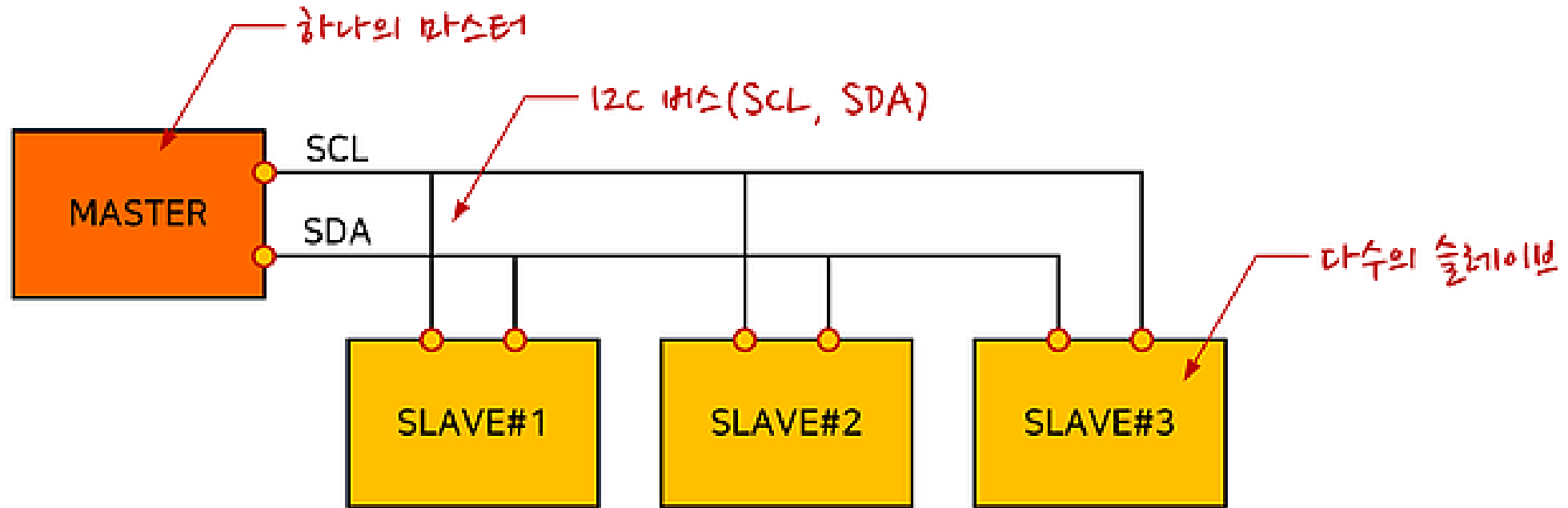
응용해보기1

- 버튼을 3초 이상 길게 눌렀을 때 부저가 짧게 3번 울리도록 하시오.
- 부저 On:200ms Off:600ms
- 1. delay()와 for문 사용
- 2. delay() 사용 없이

I2C(Inter-Integrated Circuit) 통신

- 두 개의 전선으로 여러 디바이스들을 연결할 수 있는 통신 인터페이스
- 다른 통신 인터페이스에 비해 간단하며 한 개의 마스터(master)와 여러 개의 슬레이브(slave)들을 연결하여 SDA(Serial Data)와 SCL(Serial Clock) 두 개의 신호를 통해 데이터를 주고받는다.

I2C(Inter-Integrated Circuit) 통신



아두이노 기초

I2C(Inter-Integrated Circuit) 통신

SCL(Serial Clock) :

시리얼 클럭으로 마스터와 슬레이브 간의 클럭을 맞춰 데이터를 송수신 할 수 있게 해준다.

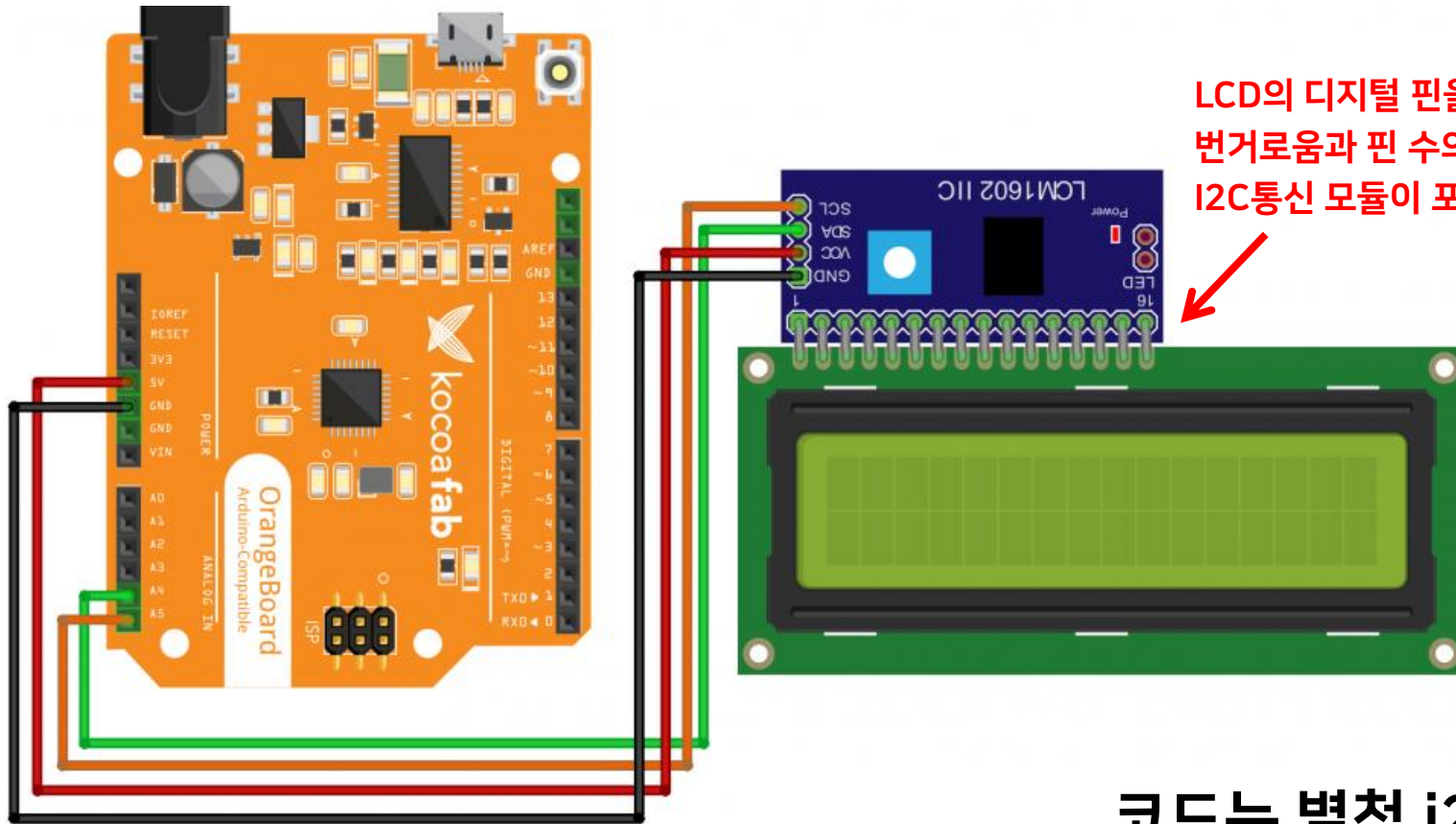
SDA(Serial Data) :

시리얼 데이터로 통신하고자 하는 데이터를 의미

아두이노 우노(Atmega328P)는 I2C 통신 기능을 지원하고
SDA 핀은 A4(아날로그 4번핀), SCL 핀은 A5(아날로그 5번핀)

아두이노 기초

I2C통신 LCD 출력



코드는 별첨 i2c_lcd.ino 참조

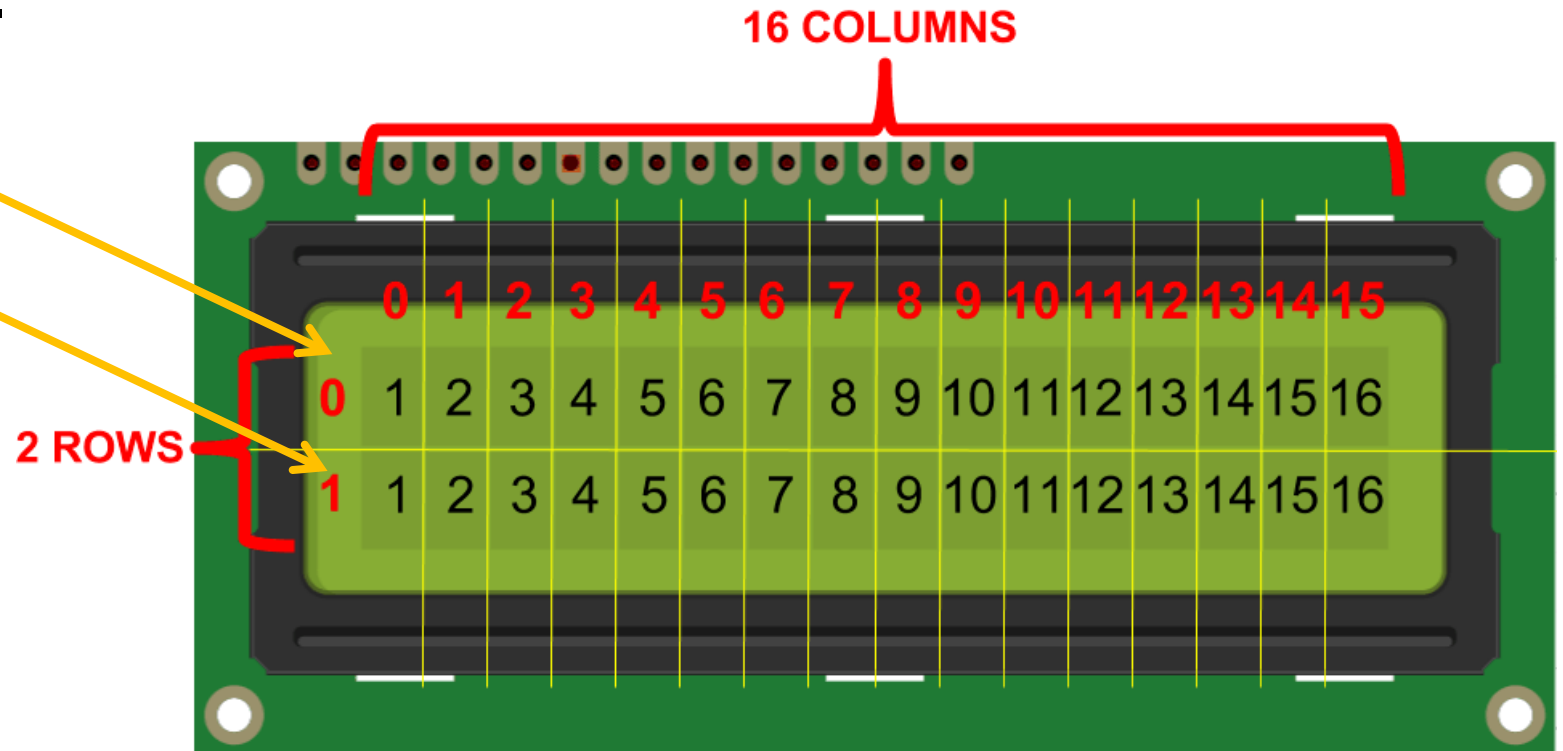
아두이노 기초

I2C통신 LCD 출력

lcd의 커서는 그림과 같다.

```
lcd.setCursor(0, 0);
```

```
lcd.setCursor(0, 1);
```



아두이노 기초

응용해보기1

- 가변저항 또는 조이스틱을 활용하여 `analogRead()`값을 LCD에 출력해보시오.

아두이노 기초

과제1

- 밝기 센서 또는 소리 센서(analog값)를 활용하여 다음 동작을 구현하라
- 밝기가 특정 임계값 기준으로 밝음→어두움으로 바뀌는 순간 감지
- 밝음→어두움으로 바뀔 때만 부저가 1초간 울린다.
- 부저가 1회 울린 후 계속 어두워도 부저는 계속 울리지 않는다.
- 다시 밝아진 후 밝음→어두움으로 바뀔 때를 감지하여 위의 동작을 수행