

* 본 자료는 서경대학교 아두이노 프로그래밍 수업 수강자를 위해 작성된 강의교재입니다.

강의교재- 아두이노 프로그래밍

2장 LED와 GPIO

서경대학교 김진현

2

LED와 GPIO

차례

- 2.1 LED 점등실험이 왜 중요한가?
- 2.2 LED 점등 실험: 구현 방안
- 2.3 LED 점등 실험: 결선방법
- 2.4 아두이노의 GPIO 데이터 출력 표준 함수
- 2.5 LED 점등 실험-프로그래밍
- 2.6 GPIO에 대한 이론적 고찰
- 2.7 고찰

2.1 LED 점등 실험이 왜 중요한가?1)

임베디드 시스템을 비롯한 마이크로프로세서를 사용한 보드 개발은 다음 그림의 5단계의 절차로 구성되며 이 때문에 보통 여러 기관이 협업이 필요하다.

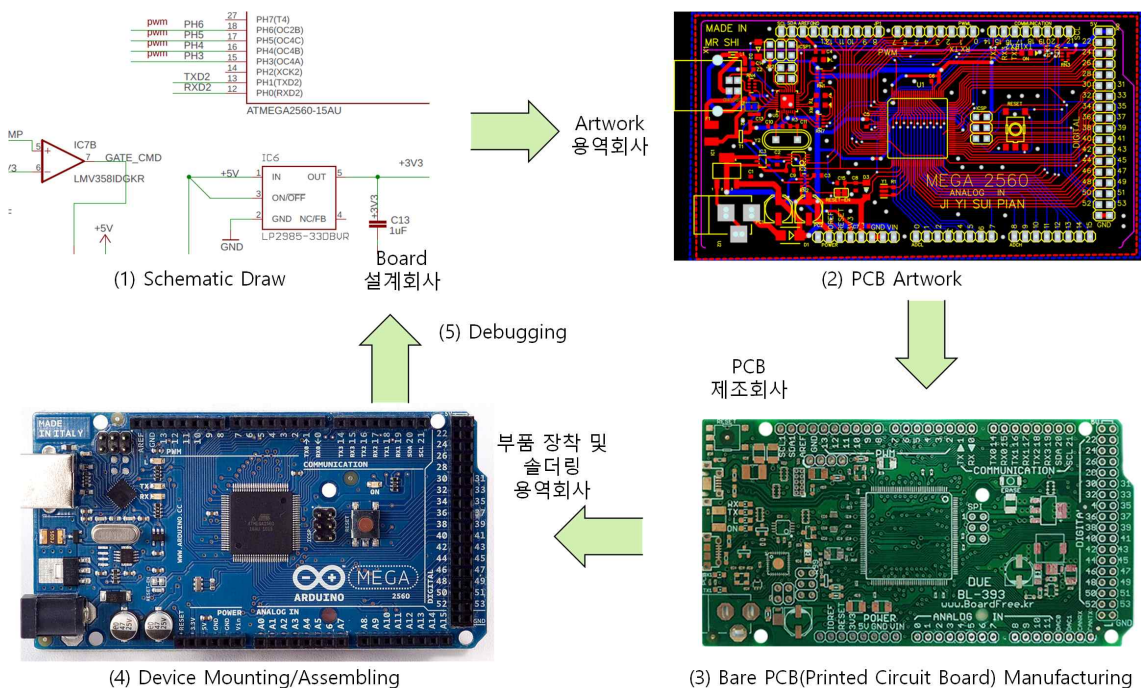


그림 2.1.1 보드 개발의 절차

1) 본 절의 설명은 사실 아두이노와 관련이 없습니다. 따라서 아두이노만 알고 싶은 학우는 무시하고 건너뛰어도 나무라지 않습니다. 그러나 이런 실무적인 관점의 전문용어(terminology)들을 이번 기회에 미약하게나마 살펴보면 어떨까 합니다. 아는 만큼 세상이 보이기 때문입니다. 이 글을 읽는 컴공 학우가 혹시 보드 개발자로 진출한다면 (1)과 (5)의 업무에 종사할 가능성이 높으니 관심 갖고 보아주기 바랍니다.

□ 보드 개발의 절차

(1) 설계, 회로도 그리기, 부품 목록 작성

보드 개발의 주체가 되는 회사가 진행한다. 회로도를 그리고 부품 목록(BOM, Bill Of Material)을 작성하여 PCB artwork를 전달하는 회사에 넘긴다. 보통 OR-CAD와 같은 회로 그리기 전문 CAD(Computer Aided Design) 도구를 사용하여 회로도를 그린다. CPU가 들어가는 디지털 회로는 컴공 출신이, 아날로그 회로는 전자과 출신이 담당한다고 보면 될 것이다. hopefully ~.

(2) artwork 작업: 부품 배치 및 패턴 그리기

회로도를 보고 가상 PCB 평면위에 부품들을 배치하고 각 부품사이의 선로를 연결하는 작업을 artwork라고 한다. 보통 Artwork 설계 전문 CAD 도구를 사용해서 상당 부분 자동화에 의지하여 작업이 이루어진다. 이 자동화 프로그램은 실제 PCB를 제조하기 위한 도면 파일을 만들어 낸다. 이 파일은 산업용의 사실상 표준인 거버(Gerber) 형식으로 배포된다.

(3) bare PCB(Print Circuit Board) 제조

최근의 PCB들은 보통 8~10 정도의 내부 층들이 모여 구성되어 있다. 도면 파일을 바탕으로 각 층(layer)에 따라 얇은 동판을 사진 식각 기술로 에칭(etching)하여 여러 장을 만들어 낸 후 적층압착(laminating) 방식으로 쌓아서 한 개의 PCB로 만들어 낸다. PCB 제조회사는 부품이 장착되지 않은 bare PCB를 출고한다.

(4) 부품 장착 및 soldering => SMD machine에서 자동화 작업으로 진행

최근의 부품들은 거의 초소형의 SMD(Surface Mounting Device)을 사용하기 때문에 납땜(soldering)을 수작업으로는 할 수 없다. 보통 부품장착 회사에는 SMD machine이라 부르는 장비로 컴퓨터 프로그램의 도움을 받아 PCB에 부품을 자동으로 장착한다.

(5) Board H/W debugging

보드 설계회사로 입고된 보드에 대해서 보통 다음 2가지(①, ②)의 테스트가 진행된다. 이렇게 보드가 정상 작동하는지를 알아보는 과정을 debugging이라 표현한다. 대개 이것만 통과해도 이후 보드의 H/W 상의 동작은 거의 문제가 발생하지 않을 것으로 낙관하는 것이 보편적이다.

① LED ON/OFF 점등실험

LED를 프로그램에서 지정하는 대로 제대로 On/Off 되는지 확인하는 실험이다. 이번 챕터가 이래서 중요하다.

이 실험이 반드시 수반되기 때문에 보드를 설계할 당시 LED 점등 회로는 기본적으로 내장하는 것이 불문율이다. 프로그램으로 LED를 점등 혹은 소등시키는 간단한 실험이 중요한 이유는 이 동작이 성공하면 CPU는 메모리의 프로그램을 수행할 수 있다고 것이고, 이후 프로그래머가 LED를 통해 프로그램의 진행을 모니터링할 수 있는 수단이 완성되기 때문이다.

② RS-232 직렬 데이터 통신

PC와의 통신을 구현해야 조금 더 심도 깊은 실험을 진행 할 수 있을 것이다. 개발 초기 단계에서는 가장 단순한 통신 수단인 RS-232²⁾로 구현한다. 근래의 임베디드 프로세서들은 Ethernet 통신장치/ USB 제어기 등을 내장하고 있지만 초기 H/W 보드 디버깅 과정에서는 단순한 구조의 RS-232 직렬통신 동작을 제일 먼저 구현한다.

직렬통신은 신호 선로가 2개 밖에 없고 이를 구현하기 위한 S/W 설계가 손쉬운 반면 이를 이용해 임베디드 보드와 개발 PC간의 데이터 통신으로 다양한 정보를 주고 받을 수 있어 매우 편리하다. 한 마디로 개발 초기 단계에서는 필수 불가결한 작업으로 이해해도 무방할 것이다. 자세한 설명은 해당 챕터(6장)에서 다루기로 한다.

2) 표준화된 직렬 통신 규약의 이름이다. 접지선(GND)와 수신 데이터 선로(RxD)와 발신 데이터 선로(TxD), 총 3개의 신호선으로 데이터를 직렬로 만들어 주고 받는 통신 규약이다. 본 교재의 6장에서 다룬다.

2.2 LED 점등 실험: 구현 방안

* 본 챕터의 LED 실험을 위해서는 부록 B1(저항), B3(LED)와 부록 C의 내용을 사전에 학습하는 것이 좋습니다.

개발된 보드가 정상적으로 이루어지고 있는가를 확인하는 방법으로 가장 많이 활용되는 방법은 LED(Light Emitting Diode)를 연결하여 LED가 프로그램에 원하는 대로 점등 혹은 소등되는지를 확인하는 것이다.

프로그램으로 LED의 점등/소등을 제어하는 방안은 논리 1과 논리 0의 출력을 제어하는 것이다. 논리 1은 물리적으로는 5V를 의미하고 논리 0은 물리적으로는 0V를 의미한다. 따라서 프로그램으로 5V와 0V를 제어하는 것은 LED에 공급할 전압을 제어하는 것이므로 점등/소등을 제어하는 것이다.

임베디드 프로세서를 비롯한 거의 모든 마이크로컨트롤러들은 바로 이렇게 **디지털 데이터**³⁾를 출력할 수 있는 **GPIO**(General Purpose *digital data* Input Output)라 불리는 **범용 디지털 데이터 입출력 장치**를 내장하고 있다. GPIO는 좀 더 범용적인 사용을 위해 디지털 데이터의 입력과 출력을 겸용할 수 있게 만든 장치로 이해할 수 있다. GPIO는 입력용으로 쓰이지 출력용으로 사용할지는 사전에 프로그램으로 설정할 수 있다.

1) S/W 관점의 구현 방안

LED 점등 테스트를 위해서는 S/W 관점에서는 다음과 같은 동작을 수행하는 프로그래밍이 필요하다.

3) **디지털 데이터**라 함은 **논리 1과 0**을 의미한다. **기호(symbol)**로는 **H와 L**을 사용한다. **물리적으로는 5V 혹은 0V**를 의미한다. 휴대용 전자 장치들은 3.3V 전원으로 많이 동작하는데 이때는 논리 1과 0가 물리적으로 3.3V와 0V로 동작한다. 더 구체적으로 보면 전력소모를 줄이기 위해 CPU core 내부는 1.0V 이하의 전압으로 동작하는 경우도 많다. 이렇듯 디지털 회로는 논리 1과 0으로 동작하는 회로이지만 실제 전압은 때에 따라 다를 수 있다.

- 1) GPIO 장치의 특정 단자의 모드를 출력 모드로 설정한다.
 - 2) 그 단자(출력용)에 디지털 데이터를 내보내어 LED가 점등되게 한다.
- 이 동작을 아두이노에서는 1) 모드 설정을 위한 함수, pinMode(단자번호, 모드), 2) 데이터 출력 함수로는 digitalWrite(단자번호, 값)가 제공된다.

2) H/W 관점의 구현 방안

H/W 관점에서는 그림 2.2.1에서 제시된 2가지 결선 방법 중 한 가지로 결선할 수 있다. 그림 (a)와 (b)는 완전히 같은 동작을 수행한다. 이 둘은 LED에 흐르는 전류량이 같으므로 동일한 동작을 수행한다.

현재의 연결에서는 GPIO가 H를 출력하면 점등되고 L을 출력하면 소등 동작이 일어난다⁴⁾. 저항소자가 필요한 이유는 LED에 흐르는 전류의 양을 적절히 차감하기 위해서이다. **저항이 없으면 과대 전류가 흘러 LED 뿐만 아니라 주변 소자(AVR 포함)에 손상이 일어날 수 있으니 주의해야 한다.**

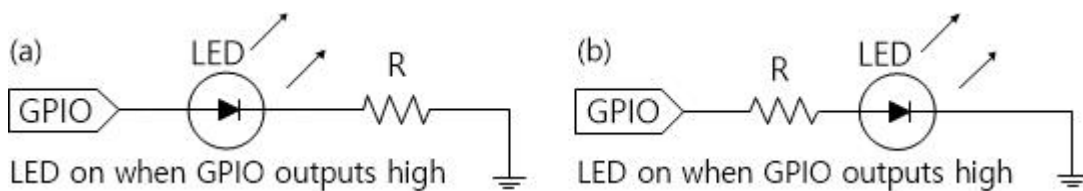


그림 2.2.1 H를 출력하면 점등되는 LED 결선도

4) H는 High, L은 Low를 의미한다. 논리기호(Logic Symbol)로 표기한 것이다. 이들은 각각 논리값(Logic Value) 1, 0을 갖는다. 실제로는 각 논리값은 5V, 0V의 물리적 값(Physical Value)을 대표한다.

2.3 LED 점등 실험: 결선 방법

그림 2.3.1은 arduino mega 2560 보드의 단자 배열을 보인 것이다. 그림의 좌측은 전체적인 단자 배열을 보인 것이고, 우측은 일부 단자의 확대된 모습을 보인 것이다. 우측 하단의 표기는 색상에 따라 해당 단자는 어떤 기능을 하는지 보인 것이다. GPIO 단자는 푸른 색 박스로 표기하였다. 우측 그림에서 GPIO 단자는 위에서부터 D54, D55, ... D69의 순서로 나열되어 있다.

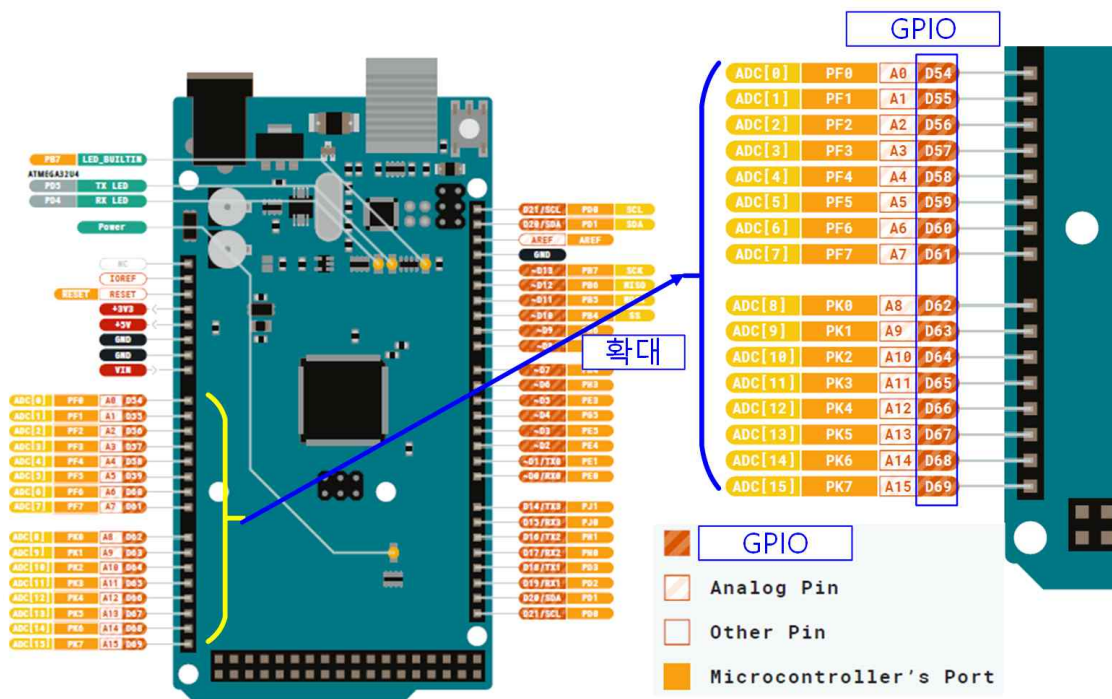


그림 2.3.1 Arduino mega 2560의 GPIO 단자

그림 2.3.2는 본 실험 키트의 만능기판(bread board)를 이용해서 LED를 결선하는 개념을 그림으로 보인 것이다.

아두이노는 GPIO의 수십 여개의 GPIO 단자를 갖고 있으며 각 단자에는 논리적 단자번호(logical pin number)를 부여하여 프로그래밍할 때는 바로 이 논리 단자번호를 사용한다. 본 사례에서는 Pin 13(논리단자)번을 사용하여

설명한다⁵⁾.

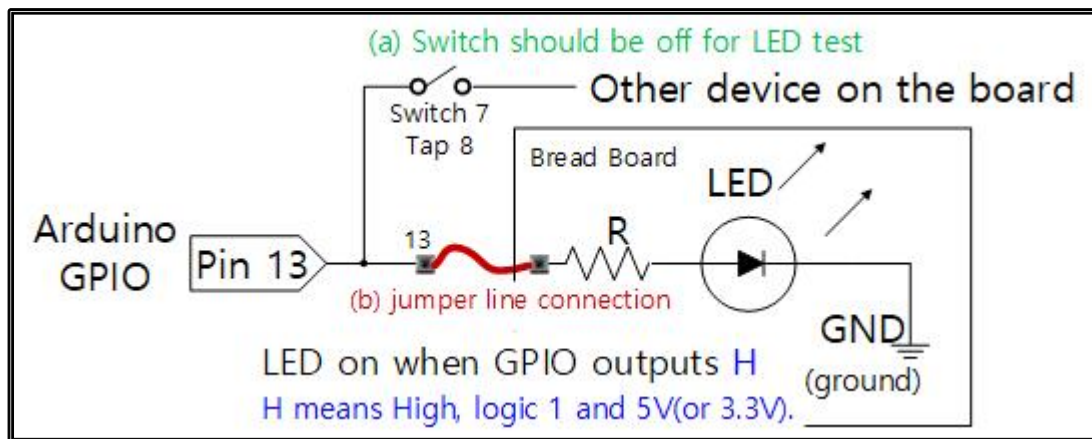


그림 2.3.2 실험 키트에서의 LED 실험을 위한 결선 개념도

GPIO 13번 단자는 위의 그림 2.3.2(a)와 (b)에 보인 바와 같이 Switch 7번 tap 8과 13번 점퍼 연결 소켓에 연결되어 있다. 스위치가 하는 역할은 GPIO 13번 단자가 키트 내의 다른 장치로 신호가 공급되는 것을 차단하는 역할을 한다. 다른 장치와의 충돌을 방지하기 위해 이 스위치는 off 상태로 설정해야 한다.

5) Arduino Mega2560 AVR 프로세서에는 총 100개의 핀이 존재하며 이 중의 54개 단자가 GPIO 용도로 활용되고 있다. 아두이노의 논리 Pin 번호와 AVR 프로세서의 실제(물리) Pin(단자) 번호와는 다르다는 것을 유의하기 바란다. AVR 프로세서의 실제 핀 번호와 아두이노 각 핀 번호 배정은 다음 URL에서 확인할 수 있다.
<http://arduino.cc/en/Hacking/PinMapping2560>

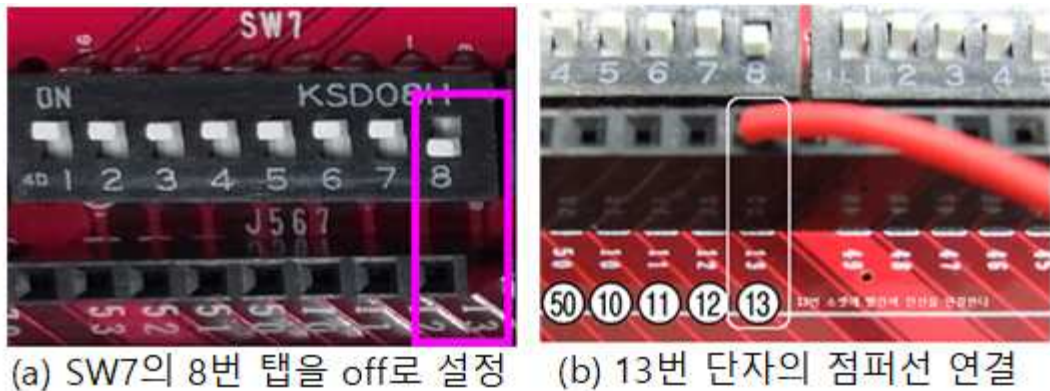


그림 2.3.3 스위치 조치사항(a)과 13번 단자의 점퍼선 연결(b)

그림 2.3.3에는 실제 사진을 이용하여 단자 13번에 관계된 키트 내의 스위치와 점퍼 소켓을 보였다.

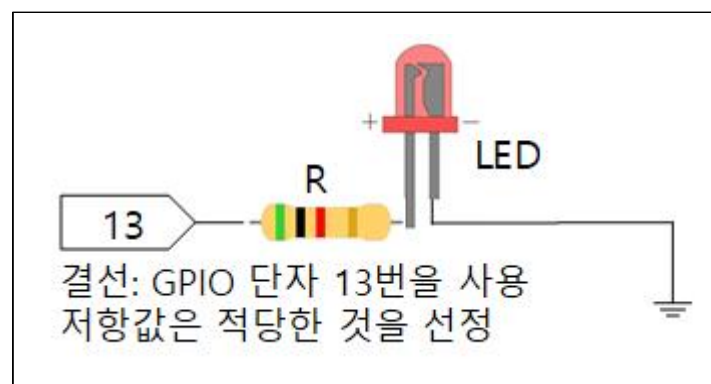


그림 2.3.4 실물을 이용한 실제 결선 개념도

GPIO 단자 번호를 13번을 사용하여 실제의 부품과의 결선을 보이는 개념도를 그림 2.3.4에 보였다. 그림에서 LED 단자의 짧은 쪽(-)은 GND에 연결하고 긴 쪽(+)은 GPIO의 출력단자 방향으로 연결해야 한다.

Ardu-Ez 상의 만능기판에서 결선한 사진을 그림 2.3.5에 보였다. 실제 작업에서는 아두이노 핀 번호와 스위치 번호를 주의 깊게 찾아서 활용해야 한다.

만능기판의 배선 상태의 확대 그림을 그림 2.3.6에 보였다. 그림에서 옅은 녹색은 실제로는 보이지 않는 선이다.

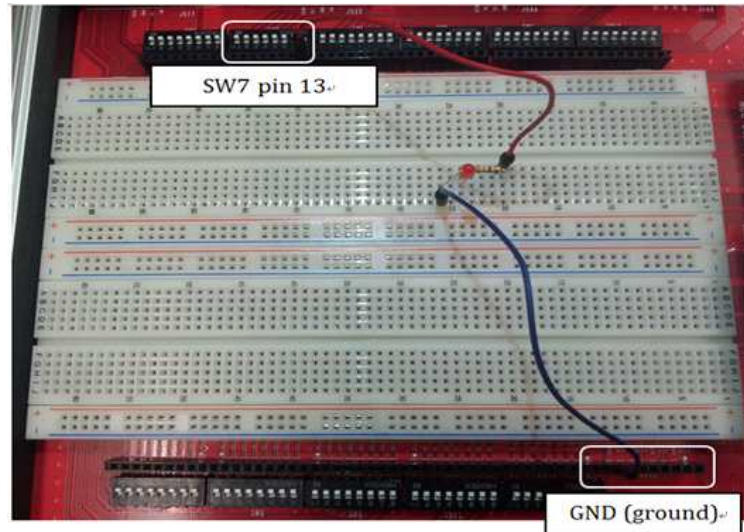


그림 2.3.5 LED 실험을 위한 실제 결선 사진(전체)

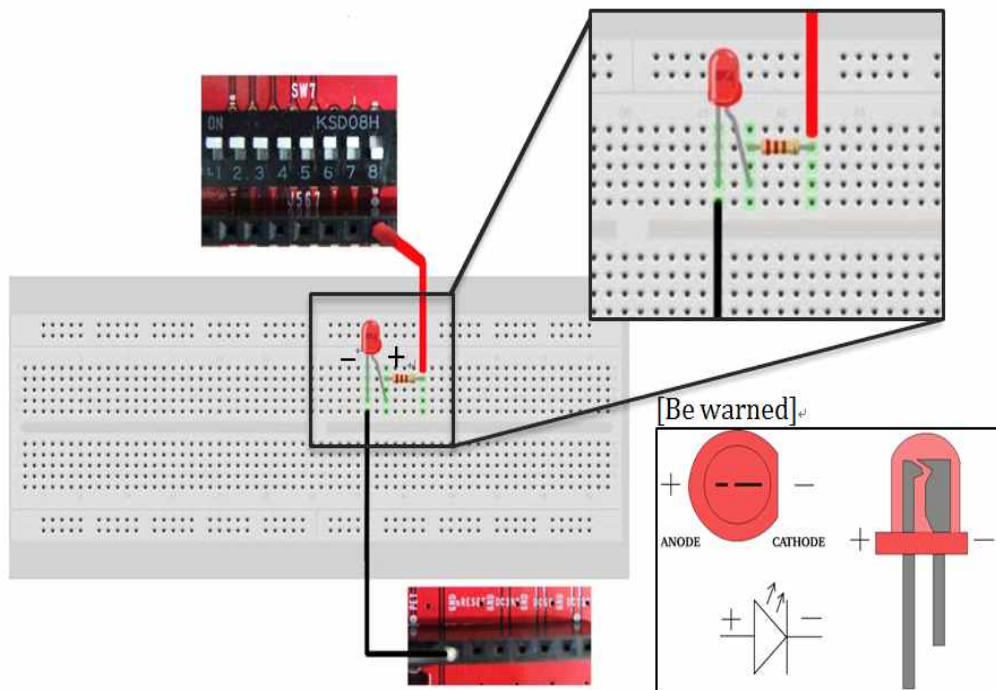


그림 2.3.6 LED 실험을 위한 실제 결선 사진(중요 사항 강조)

2.4 아두이노의 GPIO 데이터 출력 표준 함수

아두이노에서는 다음과 같이 GPIO의 특정 단자의 모드를 결정하는 표준 함수 pinMode()와 지정한 단자에 대해 논리값을 출력하는 digitalWrite() 함수를 제공한다.

void pinMode(pinNum, mode)		
기능	해당 단자를 입력으로 사용할지 출력으로 사용할지 설정	
매개변수	uint8_t pinNum	설정하고 싶은 아두이노 단자(pin) 번호 사용 가능한 GPIO가 단자 번호는 부록 A.1을 참조 바란다.
	uint8_t mode	mode=INPUT : 입력 모드 설정 mode=OUTPUT : 출력 모드 설정
리턴 값	void	없음

한다.

void digitalWrite(pinNum, value)		
기능	지정된 핀에 디지털 값 출력	
매개변수	uint8_t pinNum	출력할 아두이노 단자 번호 GPIO가 가능한 단자 번호를 선택해야 한다.
	uint8_t value	value=0 : 논리 0, LOW(0V) value=1 : 논리 1, HIGH(5V)
리턴 값	void	없음

2.5 LED 점등 실험-프로그래밍

예제(1) LED의 ON/OFF 동작 구현

pinMode() 함수는 GPIO 단자를 입력모드 혹은 출력 모드로 설정한다.
digitalWrite() 함수를 통해 지정한 출력 단자에 지정한 데이터를 출력한다.
delay() 함수는 지정한 시간(ms, 1/1000 sec.)만큼 시간 지연을 행한다.

□ 예제 1 : 1초 간격으로 LED가 ON, OFF를 무한 반복한다.

Blink.ino : 아두이노 표준 함수 - pinMode(), digitalWrite()

```
01 // Blink
02 /* Turns on an LED on for one second, then off for one second, repeatedly. This
03 example code is in the public domain. */
04 // Pin 13 has an LED connected on most Arduino boards.
05 int led = 13;
06 // the setup routine runs once when you press reset:
07 void setup() {
08     // initialize the digital pin as an output.
09     pinMode(led, OUTPUT);
10 }
11
12 // the loop routine runs over and over again forever:
13 void loop() {
14     digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
15     delay(1000); // wait for a second
16     digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
17     delay(1000); // wait for a second
18 }
19 // 실험이 끝나면 14번 단자에 Low를 출력하면 점등하고 High를 출력하면 소등하는 회로
20 를 결선하시오. 그리고 2초가 점등하고 1초간 소등하는 프로그램을 작성하시오.
```

2.6 GPIO에 대한 이론적 고찰

GPIO(General Purpose Input Output)는 범용 입출력장치, 즉, 디지털 데이터를 받아들이거나 출력하는 장치를 말한다. 여기서 디지털 데이터란 단어를 이해하는 것이 중요하다.

1) 디지털 데이터란?

디지털 데이터는 논리값 1 혹은 논리값 0으로 이루어진 데이터를 말한다. 디지털 데이터의 기본 단위는 비트(bit)이며 8개를 묶어 바이트(byte)를 구성한다⁶⁾. GPIO 장치란 1개 혹은 여러 개의 비트(혹은 여러 바이트) 단위의 디지털 데이터를 보내거나 읽어들이는 장치이다.

디지털 데이터를 내보낸다는 것(출력)은 무슨 의미일까. 이는 전기적인 신호 0V(논리 0) 혹은 5V(논리 1)를 출력한다는 것을 의미한다. 즉, 논리값이 0인 비트 데이터를 출력할 때는 0V를 출력하고, 논리값이 1인 비트 데이터를 출력할 때는 5V를 출력한다는 것을 의미한다.

최근에는 전력 소비를 줄이기 위해 물리적 전압값을 5V를 사용하지 않고 3.3V 혹은 1.8V 등을 사용하기도 한다. 한 보드 상에서 이들은 혼성 전압을 사용하여 Chip 내부에서는 낮은 전압(1.8V, 1V 등)으로 이루어진 디지털 회로를 사용하다가 외부로 신호가 출력될 때는 3.3V 혹은 5V를 출력하기도 한다. 어떤 전압이 되었든 이들 1V, 1.8V, 3.3V, 5V는 모두 논리 1을 대표하며, 0V는 논리 0을 의미한다⁷⁾. 논리 0과 1에는 때에 따라 심벌 L과 H를 사용하

6) 잘 알다시피 short 자료형(type)은 16비트, long형은 32비트로 구성된다.

7) 장치 간의 거리가 먼 디지털 데이터 통신에서는 전압이 낮으면 신호가 제대로 전달될 수 없고 외부의 잡음에 쉽게 노출되기 때문에 전압을 높혀 교신하지 않으면

기도 한다. 이들은 Low와 High의 약자로 사용된 것이다. 표 2.6.1에 디지털 데이터의 표현 방법과 실제 물리적 신호와의 관계를 보였다.

표 2.6.1 디지털 데이터의 표현 및 실제 물리적 구현

구분	상태 값		비고	
논리기호	1	0	bit. 8개가 모이면 byte라 칭함	
심벌	H	L	High, Low 문자로 디지털 신호선의 상태를 표기	
물리적 전기 신호	5V	0V	정통 디지털 회로의 전압	실제로 회로 상에서 흐르는 전압. 전압 값이 중요한 것이 아니라 두 상태의 전압 값이 인정되는 것이 디지털 회로의 특징이다. 이밖에 다른 사례의 전압이 활용되기도 한다. 동일 보드에서 여러 전압의 조합이 활용될 수도 있다.
	3.3V	0V	최근의 저전력 추구형의 디지털 회로 전압	
	1.8V	0V	CPU core 내부의 동작 전압	
	-12V	+12V	원거리 통신을 위한 RS-232의 사례	

2) GPIO 장치란?

GPIO 장치는 한 마디로 디지털 데이터의 입력 혹은 출력하는 장치이다. 입력용으로 쓸 것인가 출력용으로 쓸 것인가를 프로그램 설정으로 쉽게 설정할 수 있게 해서 GPIO의 앞글자들은 범용(general purpose)이라는 수식어로 만들어진 것이다.

GPIO 장치는 최근의 임베디드 프로세서에는 기본적으로 내장되어 있다. 초

정보가 제대로 전달되지 않는다. RS-232 직렬통신의 경우 논리 0의 전압으로 송신단 측에서는 +5V~+15V를 출력하고, 논리 1을 전송하기 위해서는 -5V~-15V를 사용한다. 수신단 측에서는 +3V~+25V, -3V~-25V를 논리 0와 논리 1로 인지한다. 이렇듯 디지털 회로는 실제의 물리적인 전압값은 절대적인 것이 아니며 그보다는 회로 내에서는 논리 0과 논리 1의 2개의 상태(binary state)가 존재한다는 것이 중요한 특징이다.

기화 과정에서 출력 혹은 입력 장치로 설정하여 사용하거나 프로그래밍을 통해 동작 중에도 수시로 입출력을 바꾸어 사용하기도 한다.

출력장치의 단순한 응용 사례로서 본 챕터에서는 LED의 점등/소등 제어에 활용하였다. 이 같은 장치의 CPU가 정상 동작하는지 육안으로 확인하는 손쉬운 방법을 제공하기 때문에 교육용 목적으로 뿐만 아니라 실제 전문적인 장치들에서도 많이 활용되고 있음을 서두에서 강조하였다.

입력장치의 단순한 사례로는 스위치(switch) 장치를 들 수 있다. GPIO를 입력으로 설정하여 스위치의 H, L 상태를 GPIO를 통해 읽어 들여 ON, OFF 상태 여부를 프로그램으로 점검할 수 있다. 이는 곧 3장에서 다르게 될 것이다.

3) 아두이노와는 어떤 관계?

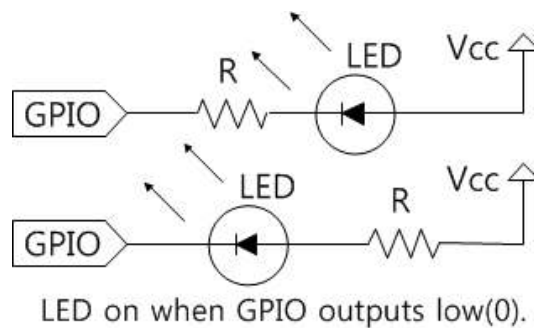
아두이노의 경우 `digitalWrite()` 함수로 GPIO 장치를 통해 디지털 데이터를 출력할 수 있으며, `digitalRead()` 함수를 통해 외부 장치의 디지털 상태 값을 입력 받을 수 있다. 또 `pinMode()` 함수는 이들 GPIO 단자에 대해 입력 장치로 설정할 것인가 혹은 출력 장치로 활용할 것인가를 결정한다.

Ardu-Ez에서 사용하는 ATmega 프로세서 단자의 상당수를 GPIO 단자로 배정하여 활용할 수도 있고 또 다른 용도로 배정하여 사용할 수도 있다. **현대의 임베디드 프로세서나 마이크로컨트롤러들은 단자를 이와 같이 GPIO 기능 외에도 그 외 다른 용도로 선택하여 사용할 수 있다. 이러한 용도 선택은 프로그램으로 설정할 수 있는 것이 일반적이다. 아두이노를 통해서 GPIO를 알게 되기를 바라며 이를 통해서 일반 프로세서들이 구사하는 기술에 대해 이해력을 가졌으면 좋겠습니다. 이러한 전략으로 본 교과의 목표를 달성하고자 합니다.**

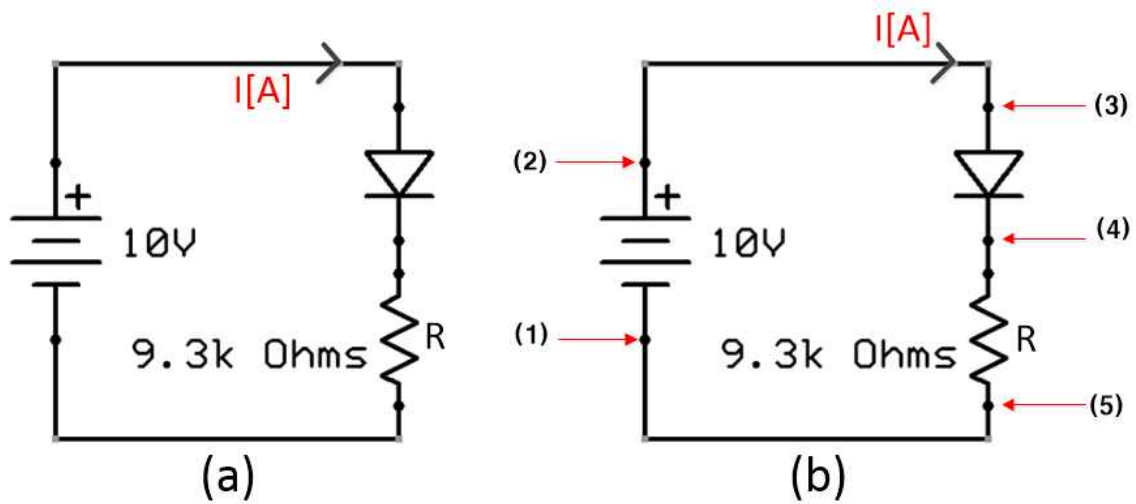
2.7 고찰

다음 주제에 대하여 답할 수 있는지 검토해 보면서 그동안 습득한 지식을 정리해 보자.

1. LED 점등 회로에 저항을 직렬 연결하는 이유는 무엇인가?
2. LED에 역방향의 전압을 인가하면 어떻게 되는가?
3. GPIO의 출력단자에 아래 그림과 같이 역방향으로 LED를 연결해도 LED를 점등할 수 있다. LED를 점등하기 위해서는 해당 단자를 어떤 값을 출력해야 하는가?



4. 디지털 데이터의 특징을 한 마디로 요약한다면?
5. 디지털 회로가 아날로그 회로에 비해 유리한 장점은 무엇인가?
6. 아래 회로(a) 에서 다이오드의 전압강하를 0.7V로 가정할 경우 이 다이오드를 통과하는 전류, $I[A]$ 를 계산하시오. 저항 R에 흐르는 전류와 R에 인가되는 전압은 얼마인지 밝히시오,



7. 위 회로 (b)에서 각 단자 (1)-(2), (2)-(3), (3)-(4), (4)-(5), (5)-(1) 간의 전압을 밝히시오. 또한 (3)-(5)의 전압도 밝히시오.

참고: 전압 측정은 다음과 같은 디지털 멀티미터로 측정 가능하다.

