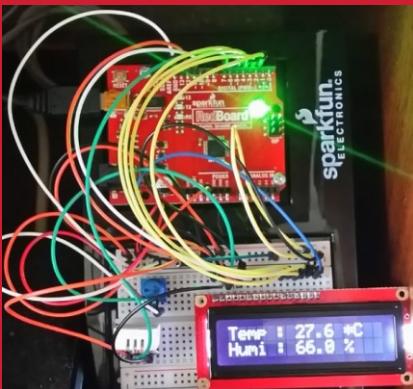


HW Programming

wk11 :

Arduino Coding III.

LED



Basic HW coding using Arduino and EV3 (RP3)

HIT, INJE University

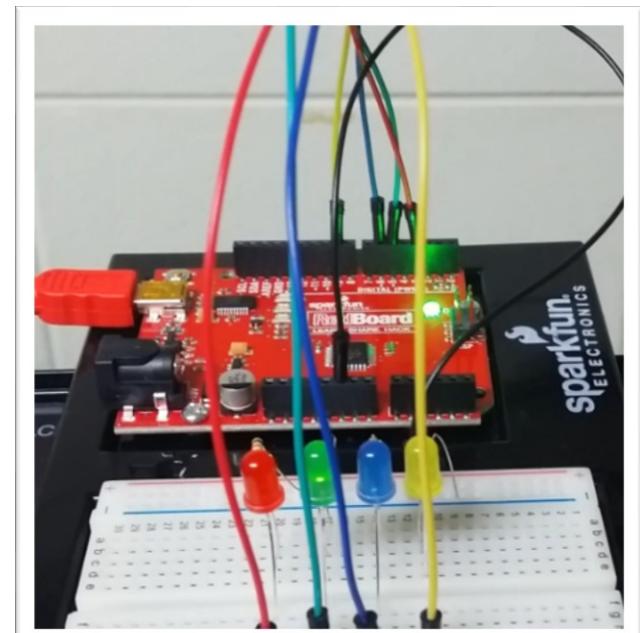
1st semester, 2017

Email : yish@inje.ac.kr

Weekly schedule of HP– 1st semester, 2017



- **wk01 :**
- **wk02 :**
- **wk03 :**
- **wk04 :**
- **wk05 :**
- **wk06 : Introduction to class and enrollment in cyber class, and installing SW**
- **wk07 : Basic HW : Arduino I. – circuit**
- **wk08 : Mid-term exam. → Practice of various circuits**
- **wk09 : Basic HW : Arduino II. – coding by Arduino IDE**
- **wk10 : Arduino Coding II. – Serial monitor**
- **wk11 : Arduino Coding III. – LED**
- **wk12 :**
- **wk13 :**
- **wk14 :**
- **wk15 : Final exam.**





Arduino SW

fritzing.org Fritzing Fritzing

fritzing

electronics
made easy

Projects Parts Download Learning Services Contribute FORUM FAB

fritzing APP

Download the free Fritzing App and start building immediately!

Fritzing is an open-source hardware initiative that makes electronics accessible as a creative material for anyone. We offer a software tool, a community website and services in the spirit of Processing and Arduino, fostering a creative ecosystem that allows users to document their prototypes, share them with others, teach electronics in a classroom, and layout and manufacture professional pcbs.

Download and Start

Download our latest version 0.9.3b released on June 2, 2016 and start right away.

Produce your own board

With Fritzing Fab you can easily and inexpensively turn your circuit into a real, custom-made PCB. Try it out now!

Participate

Fritzing can only act as a creative platform if many

<http://fritzing.org/home/>



Arduino

Starter Kit



Arduino Starter Kit





Arduino Starter Kit : contents

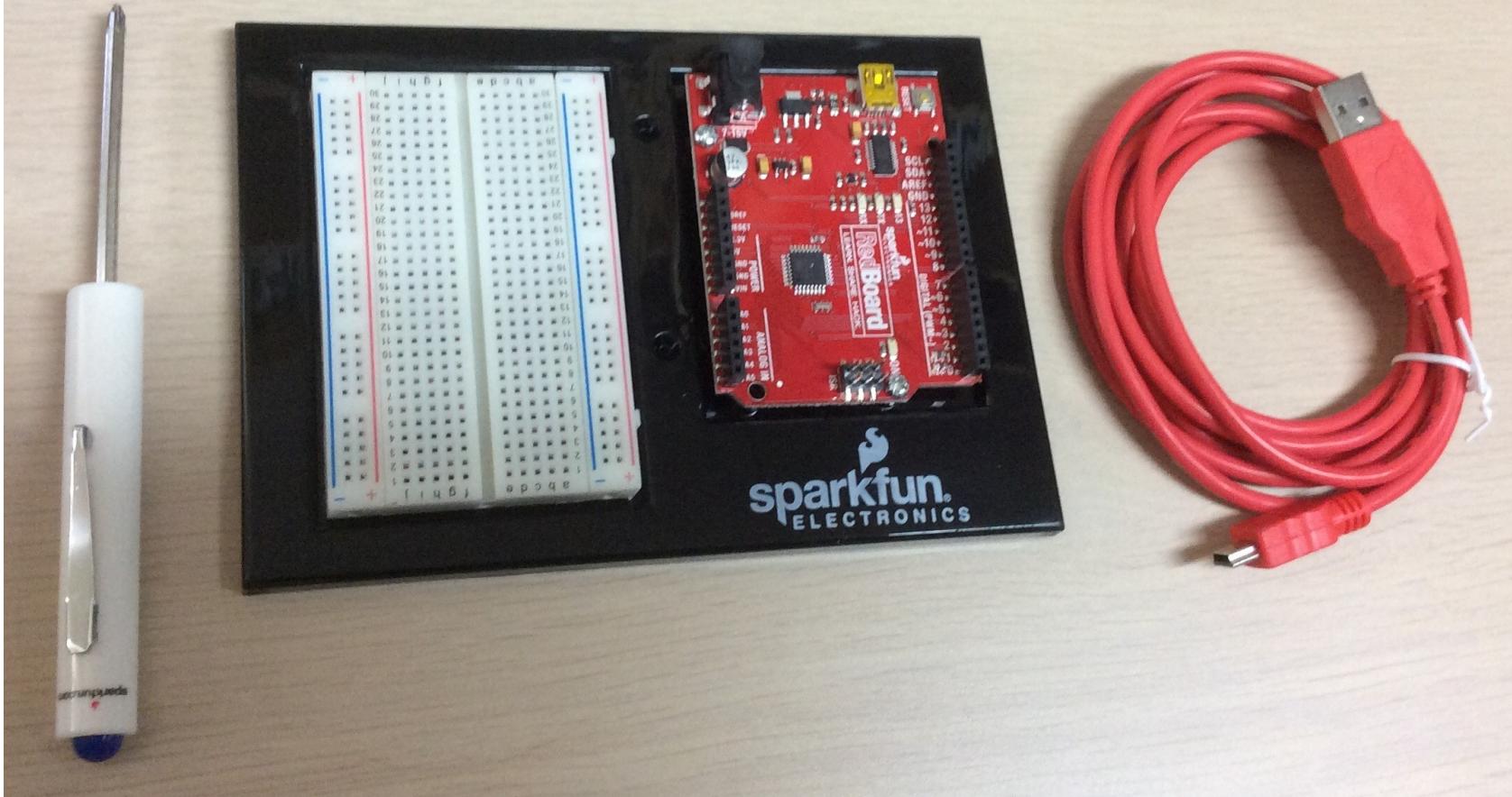
Arduino set

- **SparkFun 레드보드(아두이노 uno 호환)**
- 아두이노, 레드보드 고정판
- 한글판 가이드 북
- **400옴 브레드보드**
- 부품케이스
- **16x2 White on Black LCD**
- **74HC595** 쉬프트레지스터
- **2N2222** 트랜지스터
- **1N4148** 다이오드 (2)
- **기어달린 DC 모터**
- 서보모터
- **SPDT 5V 릴레이**
- **TMP36 온도센서**

- **플렉스 센서(Flex sensor)**
- **소프트 포텐시오미터(Softpot)**
- **SparkFun USB Cable**
- 점퍼케이블 (65)
- **Photocell**
- **RGB LED**
- **적색, 황색, 청색, 초록색 LED (5)**
- **10K 포텐시오미터 (가변저항, Trimpot)?**
- 피에조 부저
- **풀사이클 푸시버튼 (4)**
- **330, 10K 저항 (20)**

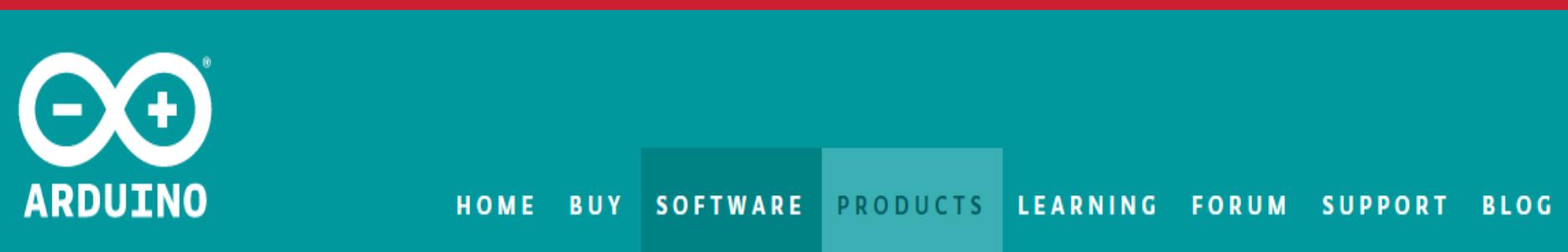


Arduino Starter Kit (Board setting)





1. Arduino SW: IDE



The screenshot shows the top navigation bar of the Arduino website. On the left is the Arduino logo. To its right are several menu items: HOME, BUY, SOFTWARE (which is highlighted in a darker teal color), PRODUCTS, LEARNING, FORUM, SUPPORT, and BLOG. The background of the header is a teal gradient.

<https://www.arduino.cc/>



1.5 Arduino IDE

sketch_may01a | 아두이노 1.8.2

파일 편집 스케치 툴 도움말

- 새 파일 Ctrl+N
- 열기... Ctrl+O
- 최근 파일 열기 >
- 스케치북 >
- 예제 >
- 닫기 Ctrl+W
- 저장 Ctrl+S
- 다른 이름으로 저장... Ctrl+Shift+S
- 페이지 설정 Ctrl+Shift+P
- 인쇄 Ctrl+P
- 환경설정 Ctrl+Comma
- 종료 Ctrl+Q

환경설정

설정 네트워크

스케치북 위치: C:\Users\yish-HC\Documents\Arduino

에디터 언어: 시스템 기본설정 (마두이노를 재시작해야 함)

에디터 글꼴 크기: 14

Interface scale: 자동 100 % (마두이노를 재시작해야 함)

다음 동작중 자세한 출력 보기: 컴파일 업로드

컴파일러 경고: None

줄 번호 표시

코드 풀딩 사용하기

업로드 후 코드 확인하기

외부 에디터 사용

Aggressively cache compiled core

시작시 업데이트 확인

스케치 파일을 저장할 때 새로운 확장자(.pde -> .ino)로 업데이트

검증 또는 업로드 할 때 저장하기

추가적인 보드 매니저 URLs

추가적인 환경 설정은 파일에서 직접 편집할 수 있습니다
C:\Users\yish-HC\AppData\Local\Arduino15\preferences.txt
(마두이노가 실행되지 않는 경우에만 수정 가능)

확인 취소

Arduino coding II.



1. Arduino SW : IDE

2. Blink a LED

3. Serial monitor

4. LED Control



3. Serial monitor & plotter



3. Serial monitor & plotter

- 3.1 Arduino에서 컴퓨터로 데이터 전송하기
- 3.2 변수 유형별로 컴퓨터에 전송하기
- 3.3 Arduino에서 시리얼 통신을 이용하여
데이터 수신하기



3. 시리얼 통신 (Serial communication)

시리얼 통신

UART (Universal Asynchronous Receiver/Transmitter)

RS-232

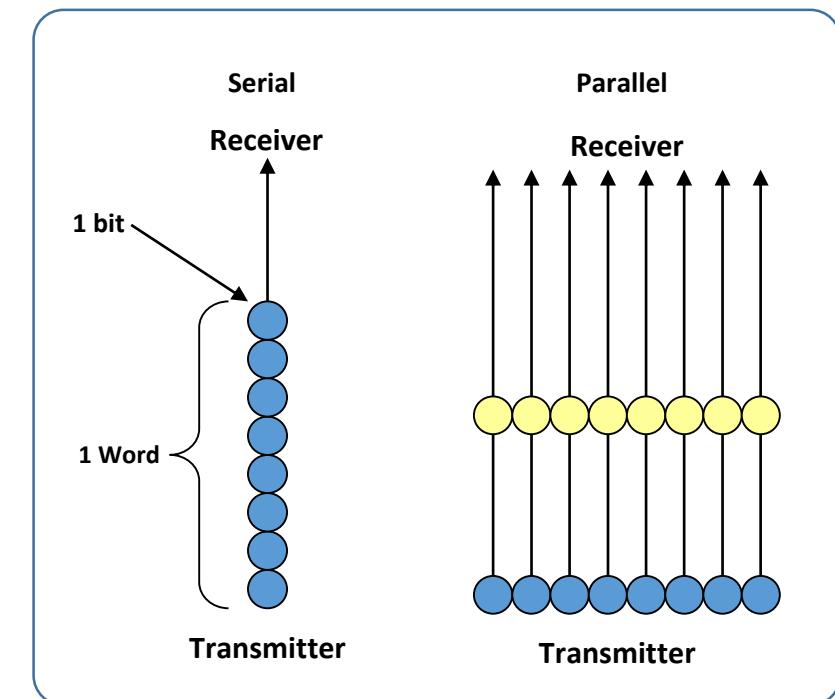
RS-422

RS-485

Arduino에서는 다음과 같은 목적으로 사용

Debugging : 프로그램의 오류를 수정하는 작업

데이터 통신 : Arduino와 컴퓨터 혹은 다른 장치와의
통신

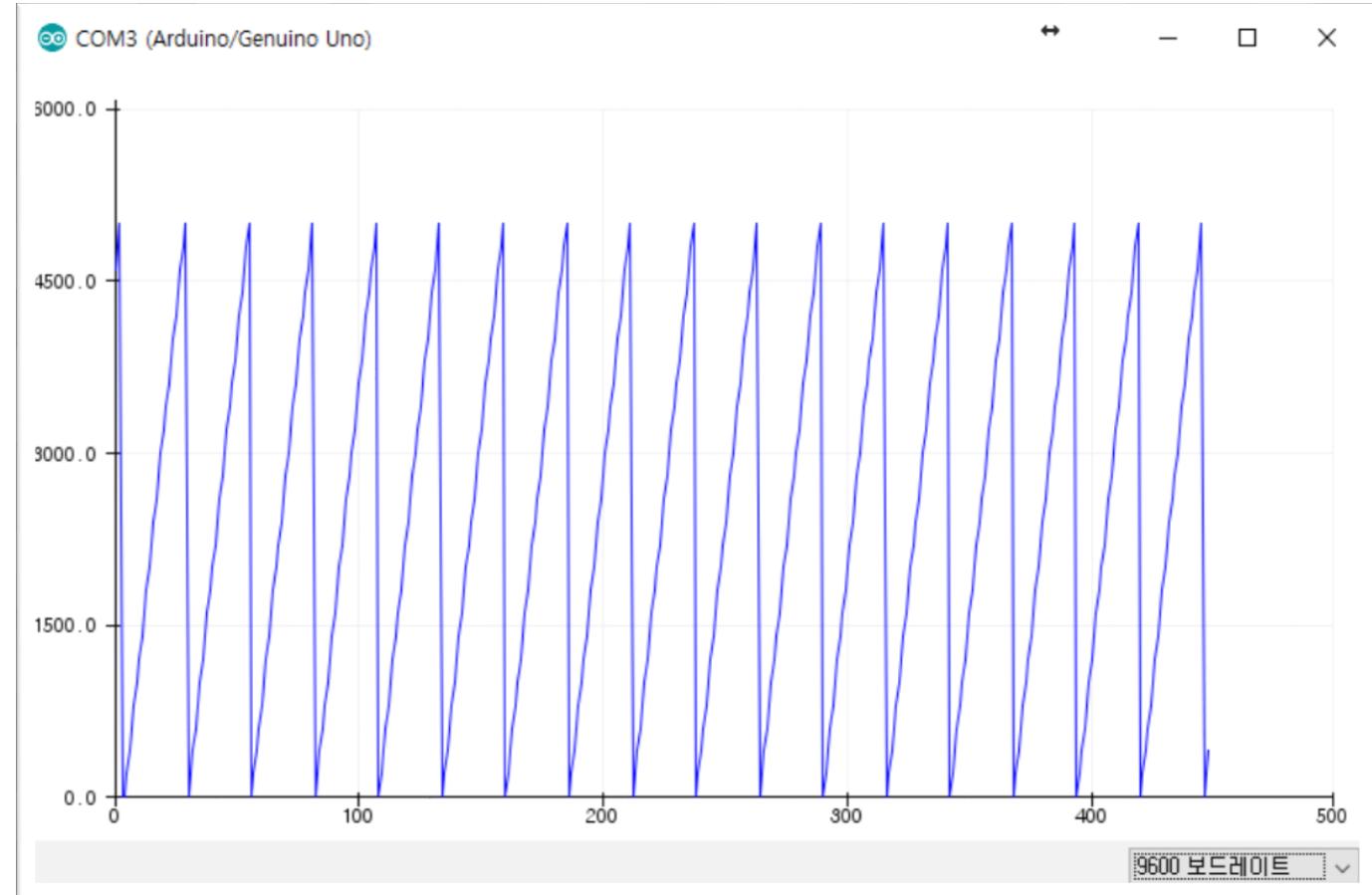




DIY2. sawtooth signal

COM3 (Arduino/Genuino Uno)

4600 msec
4800 msec
5000 msec
0 msec
200 msec
400 msec
600 msec
800 msec
1000 msec
1200 msec
1400 msec
1600 msec
1800 msec
2000 msec
2200 msec
2400 msec
2600 msec
2800 msec
3000 msec
3200 msec



HPnn_sawtooth.png



3.2.2 변수 유형별로 컴퓨터에 전송하기

✓ 실제 전송은 아스키코드 (ASCII Code)를 전송함

ASCII CODE TABLE

10	HEX	문자	10	HEX	문자	10	HEX	문자	10	HEX	문자	10	HEX	문자	10	HEX	문자
0	0x00	NULL	22	0x16	STN	44	0x2C	.	66	0x42	B	88	0x58	X	110	0x6E	n
1	0x01	SOH	23	0x17	ETB	45	0x2D	-	67	0x43	C	89	0x59	Y	111	0x6F	o
2	0x02	STX	24	0x18	CAN	46	0x2E	.	68	0x44	D	90	0x5A	Z	112	0x70	p
3	0x03	ETX	25	0x19	EM	47	0x2F	/	69	0x45	E	91	0x5B	[113	0x71	q
4	0x04	EOT	26	0x1A	SUB	48	0x30	0	70	0x46	F	92	0x5C	₩	114	0x72	r
5	0x05	ENQ	27	0x1B	ESC	49	0x31	1	71	0x47	G	93	0x5D]	115	0x73	s
6	0x06	ACK	28	0x1C	FS	50	0x32	2	72	0x48	H	94	0x5E	^	116	0x74	t
7	0x07	BEL	29	0x1D	GS	51	0x33	3	73	0x49	I	95	0x5F	_	117	0x75	u
8	0x08	BS	30	0x1E	RS	52	0x34	4	74	0x4A	J	96	0x60	.	118	0x76	v
9	0x09	HT	31	0x1F	US	53	0x35	5	75	0x4B	K	97	0x61	a	119	0x77	w
10	0x0A	LF	32	0x20	SP	54	0x36	6	76	0x4C	L	98	0x62	b	120	0x78	x
11	0x0B	VT	33	0x21	!	55	0x37	7	77	0x4D	M	99	0x63	c	121	0x79	y
12	0x0C	FF	34	0x22	"	56	0x38	8	78	0x4E	N	100	0x64	d	1222	0x7A	z
13	0x0D	CR	35	0x23	#	57	0x39	9	79	0x4F	O	101	0x65	e	123	0x7B	{
14	0x0E	SO	36	0x24	\$	58	0x3A	:	80	0x50	P	102	0x66	f	124	0x7C	
15	0x0F	SI	37	0x25	%	59	0x3B	;	81	0x51	Q	103	0x67	g	125	0x7D	}
16	0x10	DEL	38	0x26	&	60	0x3C	<	82	0x52	R	104	0x68	h	126	0x7E	~
17	0x11	DC1	39	0x27	'	61	0x3D	=	83	0x53	S	105	0x69	i	127	0x7F	DEL
18	0x12	DC2	40	0x28	(62	0x3E)	84	0x54	T	106	0x6A	j			
19	0x13	DC3	41	0x29)	63	0x3F	?	85	0x55	U	107	0x6B	k			
20	0x14	DC4	42	0x2A	*	64	0x40	@	86	0x56	V	108	0x6C	l			
21	0x15	NAK	43	0x2B	+	65	0x41	A	87	0x57	W	109	0x6D	m			

그림 2.2 ASCII 코드표



3.2.8 DIY-1

```
*** char Value ***
```

```
Binary:1000001
```

```
Decimal:65
```

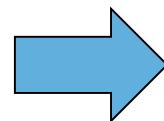
```
Hexadecimal:41
```

```
ASCII:A
```

```
*** int Value ***
```

```
*** float Value ***
```

```
float Value:65.00
```



```
*** Hello Arduino by HPnn***
```

```
*** char Value ***
```

```
Binary:1000001
```

```
Decimal:65
```

```
Hexadecimal:41
```

```
ASCII:A
```

```
*** int Value ***
```

```
int Value:65
```

```
char(intValue):A
```

```
*** float Value ***
```

```
float Value:65.00
```

HPnn_hello.png



DIY2. Escape from loop()

응용 문제 [DIY2] 0~15까지 10진수를 2진수와 16진수로 출력하는 스케치를 작성해보자

```
COM3 (Arduino/Genuino Uno)

Number = 0, Binary:0, Hexadecimal:0
Number = 1, Binary:1, Hexadecimal:1
Number = 2, Binary:10, Hexadecimal:2
Number = 3, Binary:11, Hexadecimal:3
Number = 4, Binary:100, Hexadecimal:4
Number = 5, Binary:101, Hexadecimal:5
Number = 6, Binary:110, Hexadecimal:6
Number = 7, Binary:111, Hexadecimal:7
Number = 8, Binary:1000, Hexadecimal:8
Number = 9, Binary:1001, Hexadecimal:9
Number = 10, Binary:1010, Hexadecimal:A
Number = 11, Binary:1011, Hexadecimal:B
Number = 12, Binary:1100, Hexadecimal:C
Number = 13, Binary:1101, Hexadecimal:D
Number = 14, Binary:1110, Hexadecimal:E
Number = 15, Binary:1111, Hexadecimal:F
Mission completed!
```

[Hint]

1. **int number = 0; // starting number**
2. **loop()**에서 1초 간격으로 **number**를 1씩 증가
3. 옆의 방식으로 결과 출력
4. **number**가 15를 초과하면 **loop()** 탈출
exit(0); // loop 탈출 함수

HPnn_loop_escape.png



DIY2. Escape from loop()

응용 문제 [DIY2 - hint] 0~15까지 10진수를 2진수와 16진수로 출력하는 스케치를 작성해보자

hp00_diy2

```
1 /*  
2 DIY-2  
3 */  
4  
5 // start number  
6 int number = 0;  
7  
8 // 문자열 세가지를 설정한다.  
9 String stringValue[]={"Binary:", "Hexadecimal:"}; // array  
10  
11 void setup() {  
12 // 9600bps로 시리얼 통신 설정  
13 Serial.begin(9600);  
14 }
```

```
16 void loop() {  
17  
18 // 'char Value'를 출력하고 문자열과 숫자를 변수 유형별로 출력한다.  
19 Serial.print("Number = ");  
20 Serial.print(number);  
21 Serial.print(", ");  
22 Serial.print(stringValue[0]); // stringValue 중 첫 번째 문자열 출력  
23 Serial.print(number,BIN); // 2진수 형태로 출력  
24 Serial.print(", ");  
25 Serial.print(stringValue[1]); // stringValue 중 첫 번째 문자열 출력  
26 Serial.print(number,HEX); // 16진수 형태로 출력  
27 // 줄바꿈  
28 Serial.println();  
29  
30 number++; // number 1 증가  
31  
32 if(number > 15) {  
33 Serial.print("Mission completed!");  
34 delay(1000);  
35 exit(0);  
36 }  
37  
38 delay(1000); // 1초동안 지연시킨다.  
39 }
```



DIY3. sum from 1 to 100

응용 문제 [DIY3] 1에서 100까지 정수의 합을 계산하는 스케치를 작성해보자

COM3 (Arduino/Genuino Uno)

```
Number = 85, Sum = 3655  
Number = 86, Sum = 3741  
Number = 87, Sum = 3828  
Number = 88, Sum = 3916  
Number = 89, Sum = 4005  
Number = 90, Sum = 4095  
Number = 91, Sum = 4186  
Number = 92, Sum = 4278  
Number = 93, Sum = 4371  
Number = 94, Sum = 4465  
Number = 95, Sum = 4560  
Number = 96, Sum = 4656  
Number = 97, Sum = 4753  
Number = 98, Sum = 4851  
Number = 99, Sum = 4950  
Number = 100, Sum = 5050
```

HPnn: 1 + 2 + ... + 100 =5050

[Hint]

1. **int number = 0; // starting number**
 2. **int sum = 0;**
 3. **loop()**에서 **0.1초 간격으로 number를 1씩 증가시키면서 sum에 합한다.**
-
- ✓ 옆의 방식으로 결과 출력
 - ✓ startNum 가 100을 초과하면 loop() 탈출
exit(0); // loop 탈출 함수

HPnn_sum100.png



DIY3. sum from 1 to 100 (code)

응용 문제 [DIY3] 1에서 100까지 정수의 합을 계산하는 스케치를 작성해보자

```
14 void loop() {  
15  
16     number++;  
17     sum += number;  
18     Serial.print("Number = ");  
19     Serial.print(number);  
20     Serial.print(", Sum = ");  
21     Serial.println(sum);  
22  
23     if(number == 100) {  
24         Serial.println();  
25         Serial.print("HPnn: 1 + 2 + ... + 100 =");  
26         Serial.println(sum);  
27         delay(1000);  
28         exit(0);  
29     }  
30  
31     delay(100); // 0.1초동안 지연시킨다.  
32 }
```



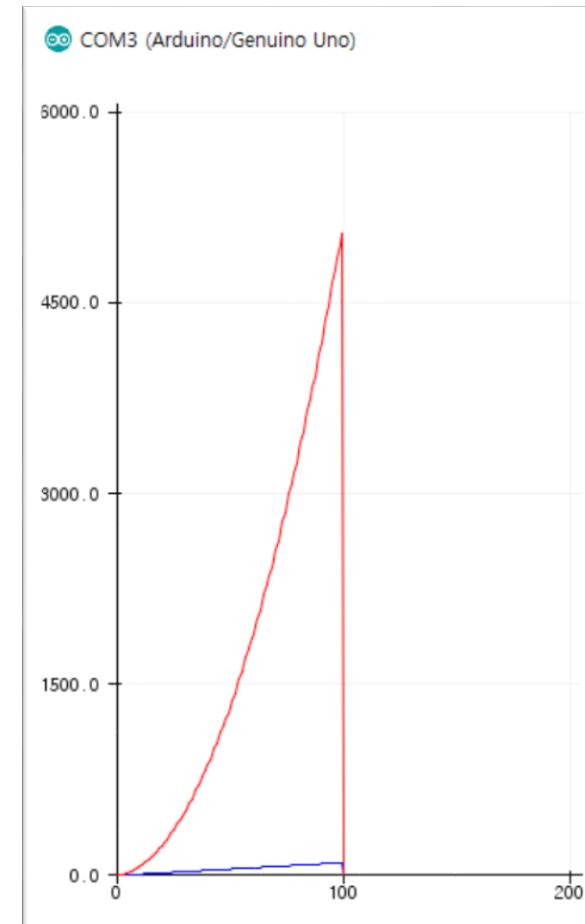
DIY3. sum from 1 to 100 (result)

응용 문제 [DIY3] Results on serial monitor and plotter

COM3 (Arduino/Genuino Uno)

```
Number = 82, Sum = 3403
Number = 83, Sum = 3486
Number = 84, Sum = 3570
Number = 85, Sum = 3655
Number = 86, Sum = 3741
Number = 87, Sum = 3828
Number = 88, Sum = 3916
Number = 89, Sum = 4005
Number = 90, Sum = 4095
Number = 91, Sum = 4186
Number = 92, Sum = 4278
Number = 93, Sum = 4371
Number = 94, Sum = 4465
Number = 95, Sum = 4560
Number = 96, Sum = 4656
Number = 97, Sum = 4753
Number = 98, Sum = 4851
Number = 99, Sum = 4950
Number = 100, Sum = 5050

HPnn: 1 + 2 + ... + 100 =5050
```





3.3 Serial monitor & plotter

3.3

시리얼 통신을 이용하여 데이터 수신하기

A screenshot of the Arduino Serial Monitor window titled "COM3 (Arduino/Genuino Uno)". The window shows the number "7" in the input field and five lines of received data:

```
7
입력한 수:2, Binary:10, Octal:2
입력한 수:5, Binary:101, Octal:5
입력한 수:9, Binary:1001, Octal:11
입력한 수:1, Binary:1, Octal:1
입력한 수:3, Binary:11, Octal:3
```



3.3.1 시리얼 통신을 이용하여 데이터 수신하기

EX 2.3

변수 유형별 Arduino에서 컴퓨터로 전송하기 (1/3)

- 실습목표
1. 컴퓨터에서 Arduino로 0~9의 숫자를 전송한다.
 2. Arduino에서는 전송 받은 숫자만큼 Arduino 보드의 LED를 점멸시킨다.

Hardware Arduino와 PC를 USB 케이블로 연결한다.

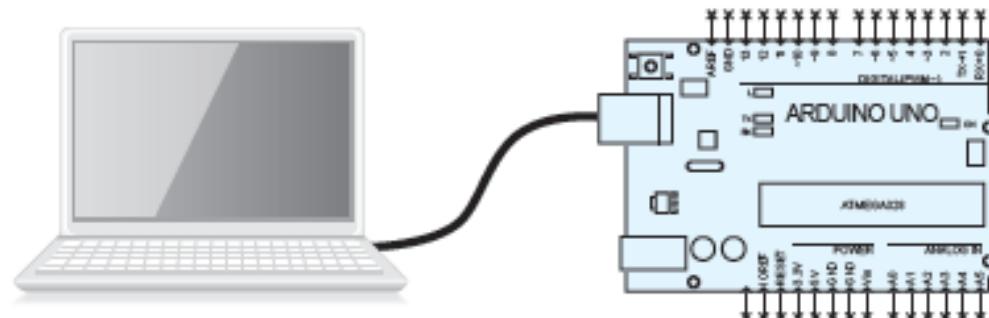


그림 2.1 Arduino와 PC와의 연결



3.3.2 시리얼 통신을 이용하여 데이터 수신하기

EX 2.3

변수 유형별 Arduino에서 컴퓨터로 전송하기 (2/3)

Commands • `Serial.available()`

시리얼 통신에 수신된 데이터가 있는지 확인한다. 있을 경우 참(true)의 값을 갖는다.

• `Serial.read()`

시리얼 통신을 통하여 수신된 값을 읽는다.

• `isDigit(변수)`

변수의 값이 ASCII 코드의 0~9의 숫자 범위에 있는지 여부를 판단. 범위에 있을 경우 참(true)의 값을 갖는다.

• `pinMode(핀번호, 설정)`

핀의 입출력 모드를 설정한다. '핀번호'에는 설정하고자 하는 핀의 번호와 '설정'에는 입력으로 사용하기 위해선 'INPUT', 출력으로 사용하기 위해선 'OUTPUT', 입력이며 풀업 사용시 'INPUT_PULLUP'을 적는다.

• `digitalWrite(핀번호, 값)`

핀에 디지털 출력(High or Low)을 한다. '핀번호'에는 출력하고자 하는 핀의 번호를, '값'에는 'HIGH' 혹은 'LOW'를 설정하여 High 혹은 Low 출력을 한다.



3.3.3 시리얼 통신을 이용하여 데이터 수신하기

EX 2.3

변수 유형별 Arduino에서 컴퓨터로 전송하기 (3/3)

- Sketch 구성
1. 13번 핀에 연결된 내장 LED를 이용한다.
 2. 시리얼 통신 상태를 감시한 후 시리얼 통신으로 입력되는 데이터가 있을 때 이를 저장한다.
 3. 전송된 값은 ASCII 코드값이므로 이를 숫자로 변경한다.
 4. 숫자만큼 LED를 0.2초 간격으로 점멸시킨다.

실습 결과 IDE의 시리얼 모니터를 실행시켜 전송란에 0~9의 값을 입력한 후 Arduino의 LED가 입력한 값 만큼 점멸하는지를 확인해 본다.
11 이상의 수를 입력했을 때의 결과는?

응용 문제

1. 0~9의 입력 값에 따라 점멸 주기가 변화하는 스케치를 작성해 보자.
2. 0~9의 숫자를 전송하면 전송된 수의 2진수와 8진수를 컴퓨터로 전송하는 스케치를 만들어보자. (hint: 예제 2.2를 참고하자)

DIY4. 점멸 주기가 변화

응용 문제 [DIY4] 0~9의 입력 값에 따라 점멸 주기가 변화하는 스케치를 작성해 보자.

- 시리얼모니터에 입력한 수를 표시
- 입력한 수에 비례해서 LED 켰 상태를 길게 유지.

```
COM3 (Arduino/Genuino Uno)

입력한 수:5
입력한 수:9
입력한 수:1
입력한 수:3
```

완성된 스케치 code를
[**HPnn_period_change.ino**](#)
로 저장해서 제출.



DIY4. 점멸 주기가 변화 (code)

[Hint: DIY4] 0~9의 입력 값에 따라

점멸 주기가 변화하는 스케치를 작성해 보자.

- 시리얼모니터에 입력한 수를 표시
- 입력한 수에 비례해서 **LED** 캠 상태를 길게 유지.
- 2자리 수를 입력하면 어떻게 되는가?

```
void loop() {
    // 시리얼 통신으로 입력 받은 데이터가 있는지를 검사하여
    // 데이터가 있을 경우에 if문 안의 명령어를 실행
    if (Serial.available()) {
        // val 변수에 시리얼 통신값 읽어오기
        char val = Serial.read();
        // 입력된 값이 0~9의 숫자인지를 판단
        if (isdigit(val)) {
            // val은 ASCII 코드값이므로 숫자로 바꿔주기 위하여
            // '0'의 아스키 코드값을 빼줌
            // blinkNumber에는 실제 숫자가 저장된다.
            blinkNumber = (val - '0');

        }
        Serial.print("입력한 수:");
        Serial.println(blinkNumber);
        //   Serial.println();
        delay(2000);

        // blinkNumber 만큼 LED의 캠상태를 길게 유지.
        for (char i = 0; i < blinkNumber; i++) {
            digitalWrite(ledPin, HIGH);
            delay(100 * blinkNumber);
            digitalWrite(ledPin, LOW);
            delay(100);
        }
    }
    // 점멸 횟수를 리셋함
    blinkNumber = 0;
}
```

DIY5. 입력된 수를 변환하여 출력 (1/2)

응용 문제 [DIY5] 0~9의 숫자를 전송하면 전송된 수의 2진수와 8진수를 컴퓨터로 전송하는 스케치를 만들어보자. (hint: 예제 2.2를 참고하자).

- 아래 출력 참조.

```
COM3 (Arduino/Genuino Uno)

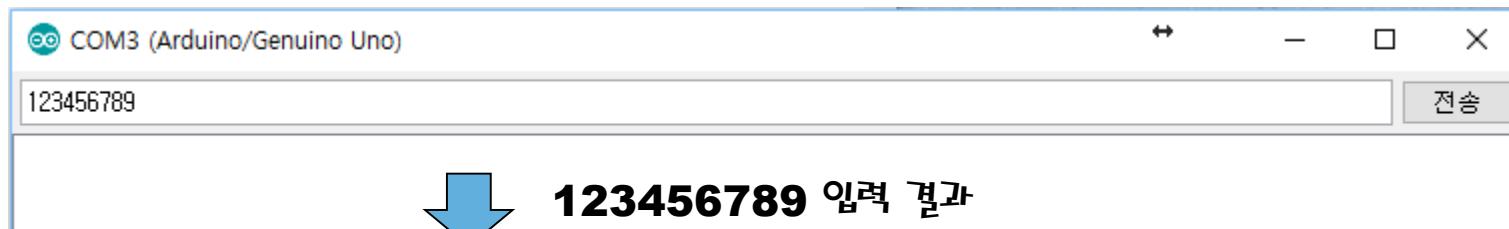
7
입력한 수:2, Binary:10, Octal:2
입력한 수:5, Binary:101, Octal:5
입력한 수:9, Binary:1001, Octal:11
입력한 수:1, Binary:1, Octal:1
입력한 수:3, Binary:11, Octal:3
```

완성된 스케치 **code**를
HPnn_number_output.ino
로 저장해서 제출.

DIY5. 입력된 수를 변환하여 출력 (2/2)

응용 문제 [DIY5] 0~9의 숫자를 전송하면 전송된 수의 2진수와 8진수를 컴퓨터로 전송하는 스케치를 만들어보자. (hint: 예제 2.2를 참고하자).

- 아래 출력 참조.



A screenshot of a terminal window titled "COM3 (Arduino/Genuino Uno)". The output displays the conversion of numbers 1 through 9 into their binary and octal representations:

```
입력한 수:1, Binary:1, Octal:1
입력한 수:2, Binary:10, Octal:2
입력한 수:3, Binary:11, Octal:3
입력한 수:4, Binary:100, Octal:4
입력한 수:5, Binary:101, Octal:5
입력한 수:6, Binary:110, Octal:6
입력한 수:7, Binary:111, Octal:7
입력한 수:8, Binary:1000, Octal:10
입력한 수:9, Binary:1001, Octal:11
```



DIY5. 입력된 수를 변환하여 출력 (code)

[Hint: DIY5] 0~9의 숫자를 전송하면 전송된 수의 2진수와 8진수를 컴퓨터로 전송하는 스케치를 만들어보자. (hint: 예제 2.2를 참고하자).

```
// LED 출력을 할 핀 번호 설정  
const int ledPin = 13;
```

```
// 점멸횟수 변수 설정  
int blinkNumber = 0;
```

```
// 문자열 배열을 설정한다.  
String stringValue[] = {"Binary:", "Octal:"};
```

```
// 시리얼 통신으로 입력 받은 데이터가 있는지를 검사하여  
// 데이터가 있을 경우에 if문 안의 명령어를 실행  
if (Serial.available()) {  
    // val 변수에 시리얼 통신값 읽어오기  
    char val = Serial.read();  
    // 입력된 값이 0~9의 숫자인지를 판단  
    if (isDigit(val)) {  
        // val은 ASCII 코드값이므로 숫자로 바꿔주기 위하여  
        // '0'의 아스키 코드값을 빼줌  
        // blinkNumber에는 실제 숫자가 저장된다.  
        blinkNumber = (val - '0');  
        if (blinkNumber < 10) {  
            Serial.print("입력한 수:");  
            Serial.print(blinkNumber);  
            delay(2000);  
        }  
    }  
}
```

Fill in your code!



4. LED control

LED (Light Emitting Diode)

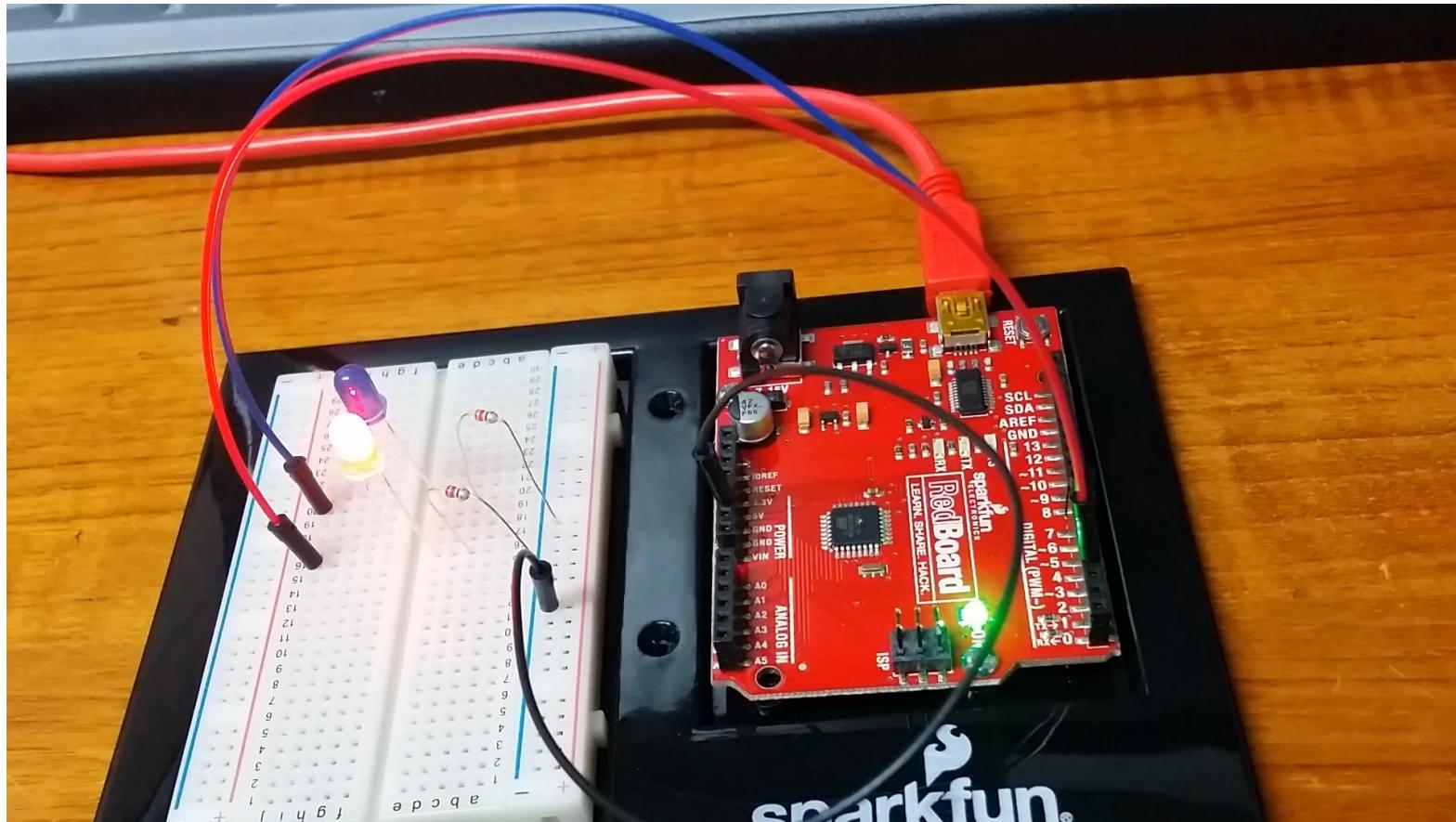
- ✓ 전기 신호를 빛으로 출력하는 반도체 소자
- ✓ 고효율, 반영구적 수명
- ✓ 가정용 실내등, 산업용 특수등, 자동차용 전조등 및 실내등에 사용





4.1 LED control

4.1 LED 교차 점멸





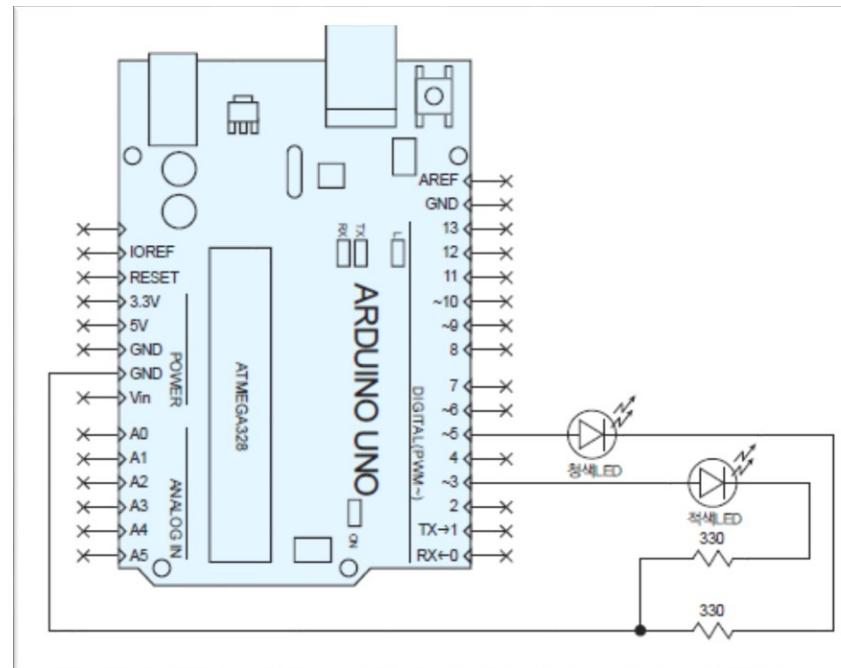
4.1 LED control – 교차 점멸

EX 4.1

LED 교차 점멸 (1/3)

실습목표 두 개의 LED를 0.1초 간격으로 교차하여 점멸시키자.

Hardware





4.1 LED control – 교차 점멸

EX 4.1

LED 교차 점멸 (2/3)

Commands

- **pinMode**(핀번호, 설정)

핀의 입출력 모드를 설정한다. ‘핀번호’에는 설정하고자 하는 핀의 번호와 ‘설정’에는 입력으로 사용하기 위해선 ‘INPUT’, 출력으로 사용하기 위해선 ‘OUTPUT’, 입력이며 풀업 사용시 ‘INPUT_PULLUP’을 설정한다.

- **digitalWrite**(핀번호, 값)

핀에 디지털 출력 (High or Low) 을 한다. ‘핀번호’에는 출력하고자 하는 핀의 번호를, ‘값’에는 ‘HIGH’ 혹은 ‘LOW’ 를 설정하여 High 혹은 Low 출력을 한다.

Sketch 구성

1. LED의 핀 번호를 설정한다.
2. `setup()`에서는 LED 출력으로 사용할 핀을 출력핀으로 설정한다.
3. `loop()`에서는 하나의 LED를 켜 후 일정시간이 지난 후에 소등하고, 다른 LED를 켠다.



4.1 LED control – 교차 점멸

EX 4.1

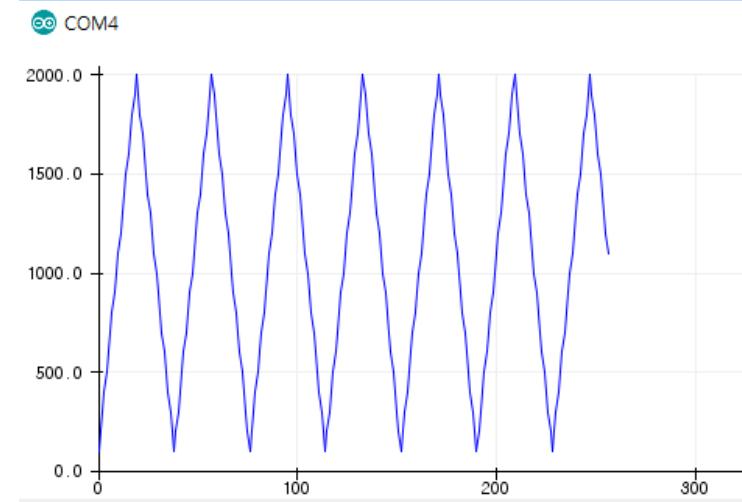
LED 교차 점멸 (3/3)

실습 결과 LED A와 B가 0.1초 단위로 교차하며 점멸한다.

응용 문제 점멸 주기가 0.1초부터 2초로 0.1초 단위로 증가하였다가 다시 반대로 2초부터 0.1초까지 감소하는 동작을 반복하는 스케치를 작성해 보자.
(hint: delay 명령어의 괄호 안의 숫자를 증감시킨다.)

```
delay = 1600 msec  
delay = 1700 msec  
delay = 1800 msec  
delay = 1900 msec  
delay = 2000 msec  
delay = 1900 msec  
delay = 1800 msec  
delay = 1700 msec  
delay = 1600 msec
```

```
delay = 500 msec  
delay = 400 msec  
delay = 300 msec  
delay = 200 msec  
delay = 100 msec  
delay = 200 msec  
delay = 300 msec  
delay = 400 msec  
delay = 500 msec
```



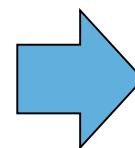


4.1 LED control – 교차 점멸 (code)

ex_4_1

```

1 /*
2 예제 4.1
3 LED 점멸
4 */
5
6 const int ledA = 3;
7 const int ledB = 5;
8
9 void setup()
10 {
11   pinMode(ledA, OUTPUT);
12   pinMode(ledB, OUTPUT);
13 }
14
15 void loop()
16 {
17   digitalWrite(ledA,HIGH);
18   digitalWrite(ledB,LOW);
19   delay(100);
20   digitalWrite(ledA,LOW);
21   digitalWrite(ledB,HIGH);
22   delay(100);
23 }
```



완성된 스케치 code를
HPnn_2led.ino
 로 저장해서 제출.

```

6 const int ledA = 3;
7 const int ledB = 5;
8
9 int number = 1;
10 boolean flag = true;

12 void setup()
13 {
14   Serial.begin(9600);
15   pinMode(ledA, OUTPUT);
16   pinMode(ledB, OUTPUT);
17 }
```

```

19 void loop()
20 {
21   digitalWrite(ledA, HIGH);
22   digitalWrite(ledB, LOW);
23   delay(100 * number);
24   digitalWrite(ledA, LOW);
25   digitalWrite(ledB, HIGH);
26   Serial.print("delay = ");
27   Serial.print(100 * number);
28   Serial.println(" msec");
29   delay(100 * number);

30
31   if (flag) {
32     number++;
33   } else {
34     number--;
35   }

36
37
38
39
40
41
42
43 }
```

Fill in your code!

4.2 LED control – 밝기 조절

밝기 조절 : 디밍 (Dimming)

- ✓ LED에 입력되는 전력은 **PWM (Pulse Width Modulation)**을 이용하여 조절.
- ✓ PWM : 고속의 스위칭으로 High와 Low 신호의 비율을 조절하여
LED의 밝기, 모터의 회전 등을 조절하는 방법
- ✓ Arduino에서는 **analogWrite()** 명령어로 구현
- ✓ Arduino UNO의 경우 **3, 5, 6, 9, 10, 11** 번 핀이 PWM을 지원한다.



4.2 LED control – 밝기 조절

PWM (Pulse Width Modulation)

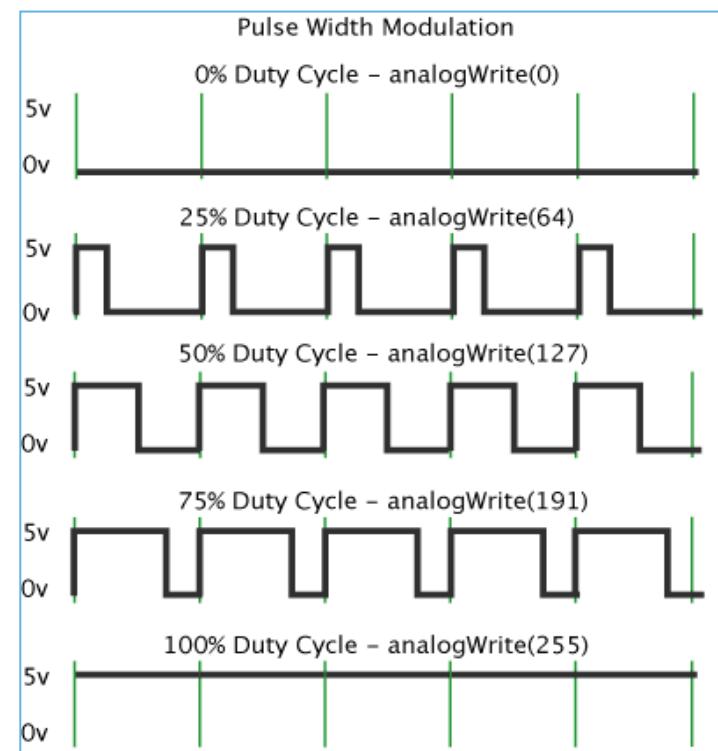
Using [analogWrite\(pin, pwm_value\)](#) function in fading an LED off and on.

AnalogWrite uses [pulse width modulation \(PWM\)](#), turning a digital pin on and off very quickly with different ratio between on and off, to create a fading effect.

A call to [analogWrite\(\)](#) is on a scale of **0 - 255**, such that analogWrite(255) requests a 100% duty cycle (always on), and analogWrite(127) is a 50% duty cycle (on half the time)

PWM frequency = 500 Hz

<https://www.arduino.cc/en/Tutorial/PWM>





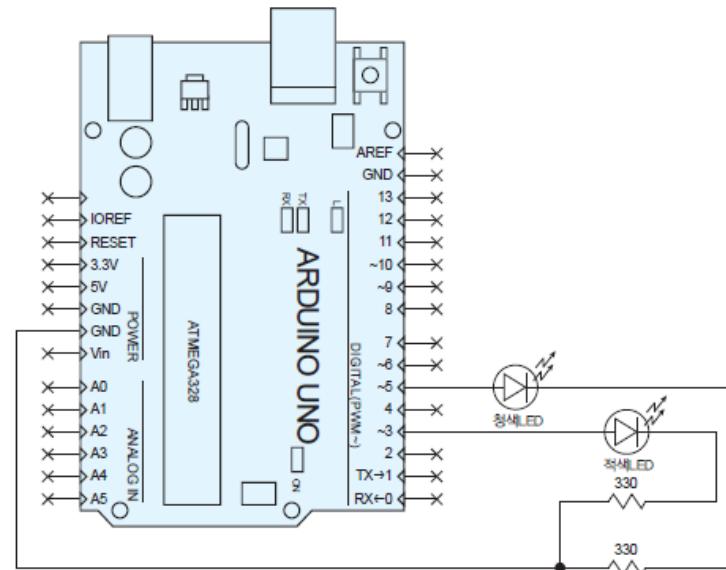
4.2 LED control – 밝기 조절

EX 4.2

LED 밝기 조절 (1/2)

- 실습목표
1. 두 개의 LED의 밝기를 조절하자.
 2. 각각의 LED가 교차하며 밝아졌다 어두워 졌다를 반복하도록 하자.

- Hardware
1. 청색과 적색 LED의 Anode핀을 Arduino의 3번 5번 핀에 연결한다.
 2. Cathode핀에 330Ω 저항을 연결하여 저항의 반대쪽은 Arduino의 GND에 연결한다.
 3. LED가 연결된 핀에 HIGH신호가 출력될 때 LED가 점등된다.





4.2 LED control – 밝기 조절

EX 4.2

LED 밝기 조절 (2/2)

Commands • `analogWrite(핀번호, 값)`

정해진 핀에 아날로그 출력을 한다. ‘값’에는 0~255의 값을 넣는다.

Sketch 구성 1. LED의 핀 번호를 설정한다.

2. `setup()`에서는 LED 출력으로 사용할 핀을 출력핀으로 설정한다.

3. 밝기를 저장할 변수를 설정한다.

4. 하나의 LED가 밝아질 때 다른 LED는 어두워져야 하므로 이를 조절할 변수를 설정한다.

5. `loop()`에서는 밝기와 밝기 변수 증감을 위한 변수를 조절하여 두 개의 LED를 교차 점멸시키는 동작을 반복한다.

실습 결과 LED A와 B가 밝기가 변화하며 점멸한다.

응용 문제 1. 네개의 다른 색깔의 LED를 Arduino에 연결한다.

2. 네개의 LED가 순서대로 디밍하는 스케치를 작성해보자.



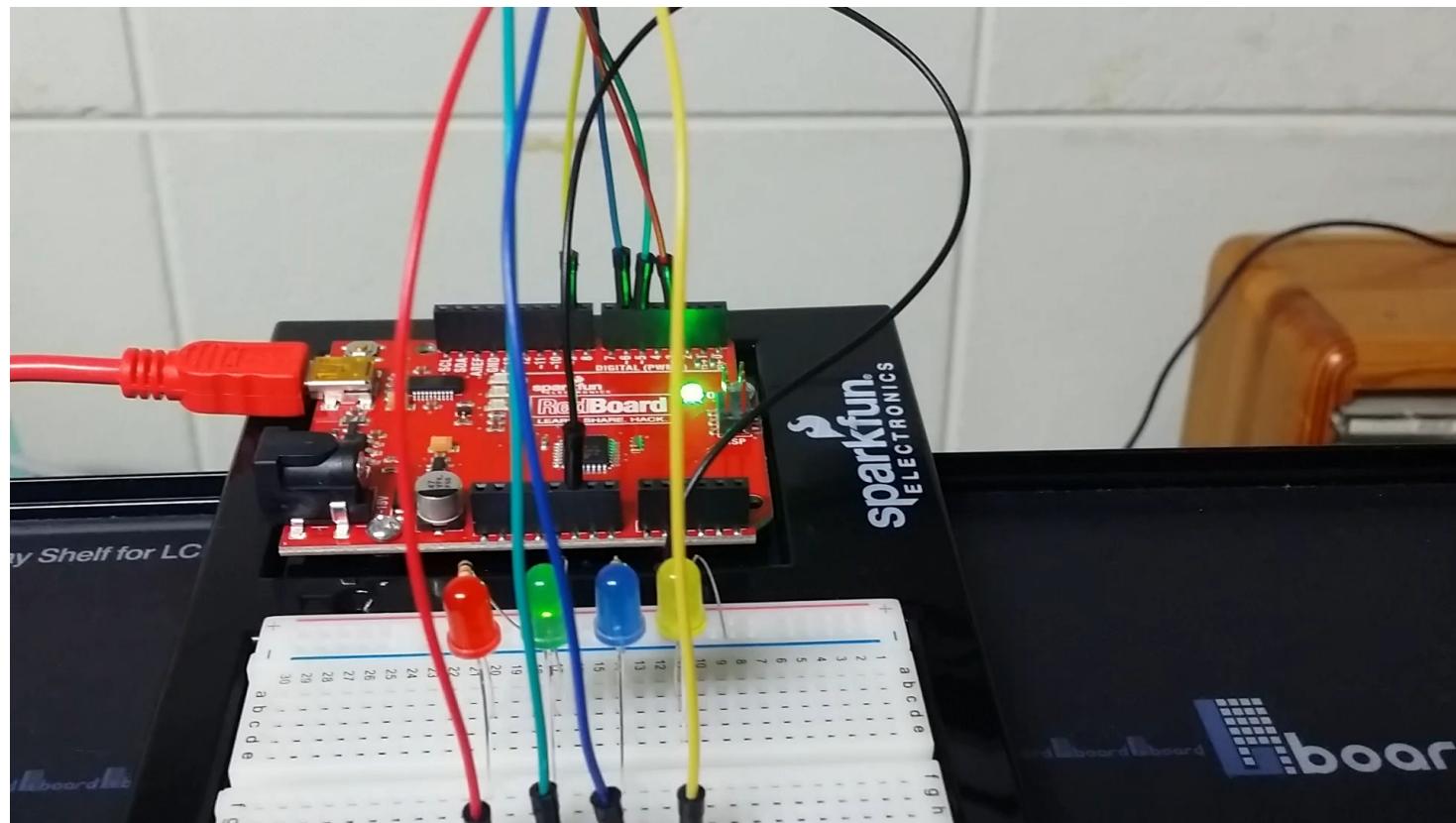
4.2 LED control – 밝기 조절 (code)

```
ex_4_2_start
1 /*
2 예제 4.2
3 LED 밝기 조절
4 */
5
6 const int ledA = 3; //LED A를 3번핀에 연결
7 const int ledB = 5; //LED B를 5번핀에 연결
8 int brightness = 0; //밝기를 조절하기 위한 변수
9 int increment = 1; //밝기 변수 증감을 위한 변수
10
11 void setup()
12 {
13 // analogWrite 핀에는 별도의 설정이 불필요하다.
14 }
15
16 void loop()
17 {
18 analogWrite(ledA,brightness); // LED A 밝기 조절
19 analogWrite(ledB,255-brightness); // LED B 밝기 조절
20
21 brightness = brightness + increment; // 밝기 조절
22 if((brightness >= 255) || (brightness <= 0)) increment = -increment; // 밝기 변수 증감 방향 변경
23 delay(10); // 0.01 초간 지연
24 }
```



4.2 4 LED control – 밝기 조절

- DIY 6. 1. 네개의 다른 색깔의 LED를 Arduino에 연결한다.
2. 네개의 LED가 순서대로 디밍하는 스케치를 작성해보자.





4.2 4 LED control – 밝기 조절 (code-1)

```

1 /*
2 Dimming 4 leds
3 */
4
5 int ledR = 3; // LED connected to digital pin 3
6 int ledG = 5;
7 int ledB = 6;
8 int ledY = 9;
9
10 int dimTime = 20;
11
12 void setup() {
13 // nothing happens in setup
14 }

```

완성된 스케치 code를
[HPnn_4led.ino](#)
 로 저장해서 제출.

```

16 void loop() {
17 // fade in from min to max in increments of 5 points:
18 for(int fadeValue = 0 ; fadeValue <= 255; fadeValue +=5) {
19 // sets the value (range from 0 to 255):
20 analogWrite(ledR, fadeValue);
21 // wait for 30 milliseconds to see the dimming effect
22 delay(dimTime);
23 }
24
25 // fade out from max to min in increments of 5 points:
26 for(int fadeValue = 255 ; fadeValue >= 0; fadeValue -=5) {
27 // sets the value (range from 0 to 255):
28 analogWrite(ledR, fadeValue);
29 // wait for 30 milliseconds to see the dimming effect
30 delay(dimTime);
31 }
32
33 for(int fadeValue = 0 ; fadeValue <= 255; fadeValue +=5) {
34 // sets the value (range from 0 to 255):
35 analogWrite(ledG, fadeValue);
36 // wait for 30 milliseconds to see the dimming effect
37 delay(dimTime);
38 }

```

각 led에 동일한 dimming code 적용



4.2 4 LED control – 밝기 조절 (code-2)

```
1 /*  
2 Dimming 4 leds  
3 */  
4  
5 int ledR = 3; // LED connected to digital pin 3  
6 int ledG = 5;  
7 int ledB = 6;  
8 int ledY = 9;  
9  
10 int dimTime = 20;  
11  
12 void setup() {  
13 // nothing happens in setup  
14 }
```

```
16 void loop() {  
17 // fade ledR  
18 dimLed(ledR);  
19 // fade ledG  
20 dimLed(ledG);  
21 // fade ledB  
22 dimLed(ledB);  
23 // fade ledY  
24 dimLed(ledY);  
25 }  
26 void dimLed(int led) {  
27 // fade in from min to max in increments of 5 points:  
28 for(int fadeValue = 0 ; fadeValue <= 255; fadeValue +=5) {  
29 // sets the value (range from 0 to 255):  
30 analogWrite(led, fadeValue);  
31 // wait for 20 milliseconds to see the dimming effect  
32 delay(dimTime);  
33 }  
34 // fade out from max to min in increments of 5 points:  
35 for(int fadeValue = 255 ; fadeValue >= 0; fadeValue -=5) {  
36 // sets the value (range from 0 to 255):  
37 analogWrite(led, fadeValue);  
38 // wait for 20 milliseconds to see the dimming effect  
39 delay(dimTime);  
40 }  
41 }
```

각 led에 동일한 dimming code 적용

dimLed(int led) 반복 사용



[Practice]

◆ [wk11]

- **Arduino coding III**
- **Complete your codes**
- **Upload file name : HPnn_Rpt06.zip**

wk11 : Practice-06 : HPnn_Rpt06.zip

◆ [Target of this week]

- Complete your 6th project
- Save your outcome and compress all ino files.

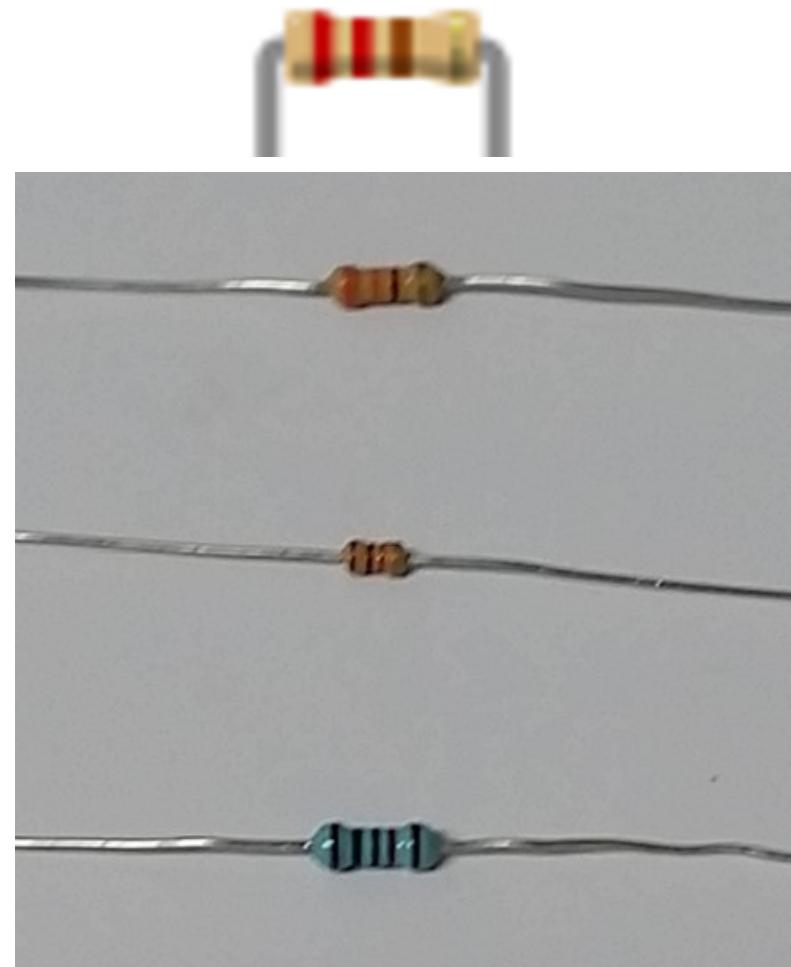
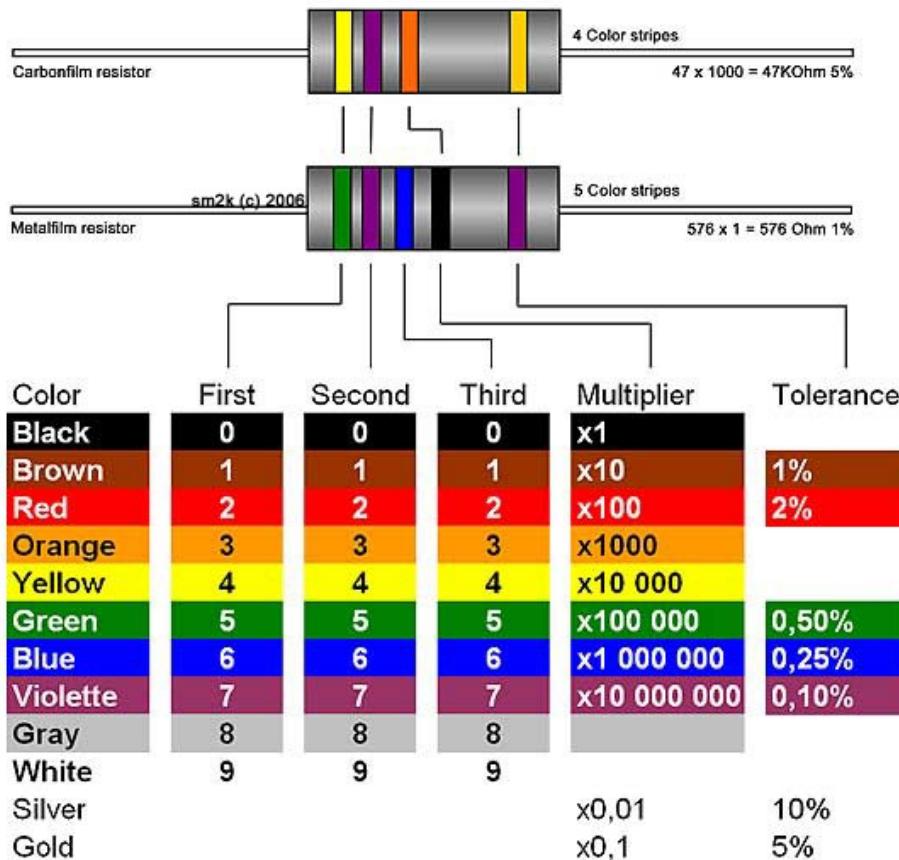
제출파일명 : **HPnn_Rpt06.zip**

- 압축할 파일들

- ① **HPnn_period_change.ino**
- ② **HPnn_number_output.ino**
- ③ **HPnn_2led.ino**
- ④ **HPnn_4led.ino**



[참고 : 저항 값 읽기]





교재



1장 Arduino 소개 및 사용법

2장 시리얼 통신

3장 LCD 출력

4장 LED 출력 1

4장 LED 출력 2

4장 LED 출력 3

5장 디지털 신호 입력

중간평가

6장 아날로그 신호 입력 1

| 6장 아날로그 신호 입력 2

| 7장 모터 구동

| 8장 적외선 리모컨

| 9장 여러 가지 부품들

| 10장 프로젝트 1

| 10장 프로젝트 2

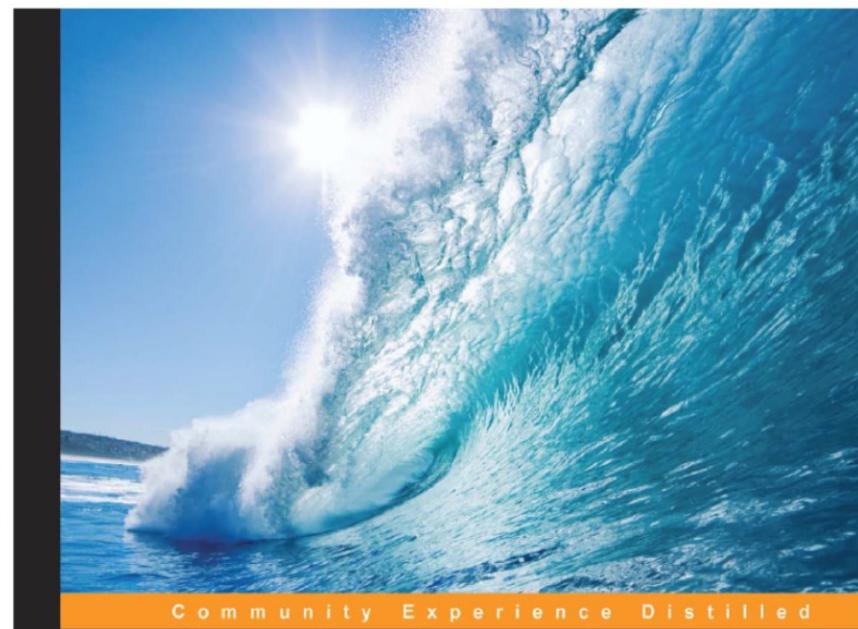
| 기말평가

생능출판사

2016

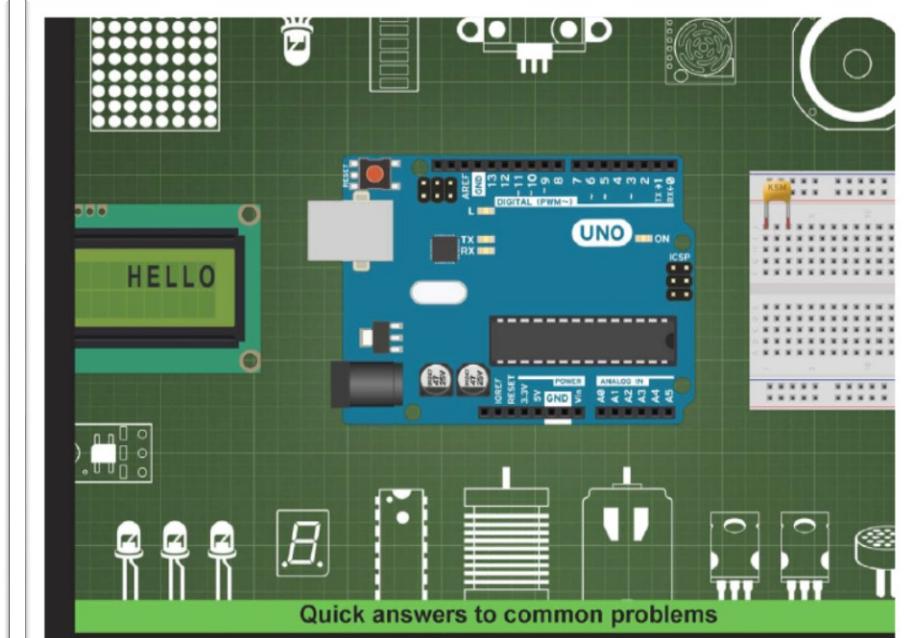


References



Francis Perea

[PACKT]
PUBLISHING



Cornel Amariei

[PACKT] open source*
PUBLISHING