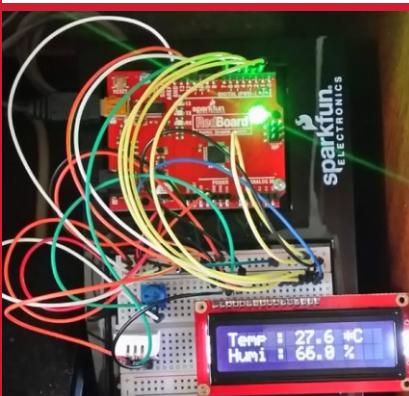


HW Programming

wk09 :

Arduino II.

Coding using IDE



Basic HW coding using Arduino and EV3 (RP3)

HIT, INJE University

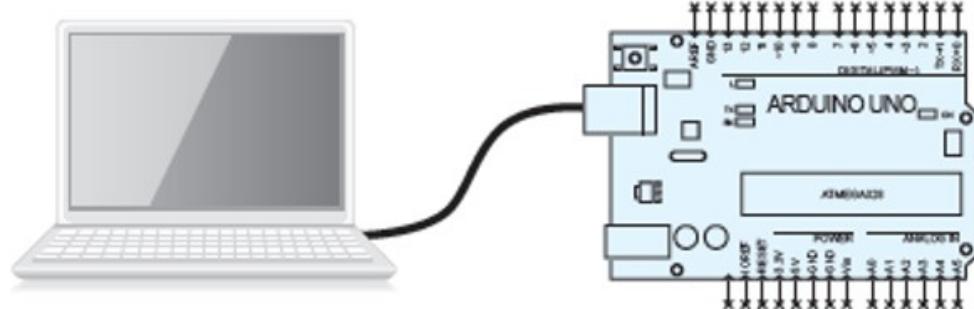
1st semester, 2017

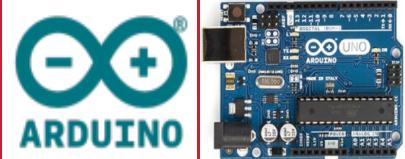
Email : yish@inje.ac.kr

Weekly schedule of HP– 1st semester, 2017

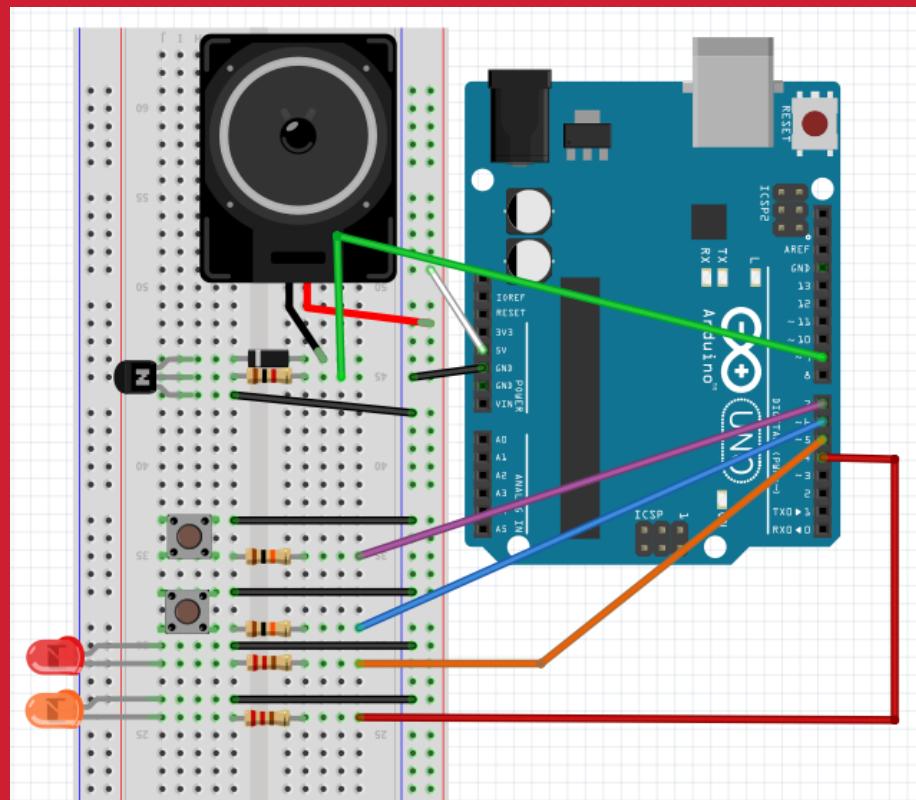
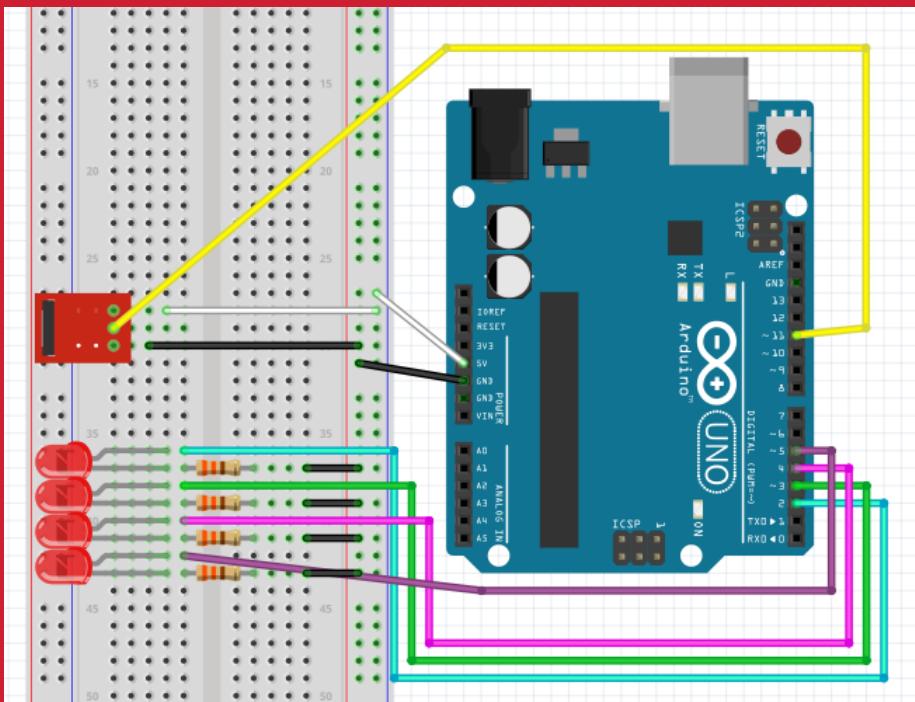


- **wk01 :**
- **wk02 :**
- **wk03 :**
- **wk04 :**
- **wk05 :**
- **wk06 : Introduction to class and enrollment in cyber class, and installing SW**
- **wk07 : Basic HW : Arduino I. – circuit**
- **wk08 : Mid-term exam. → Practice of various circuits**
- **wk09 : Basic HW : Arduino II. – coding by Arduino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





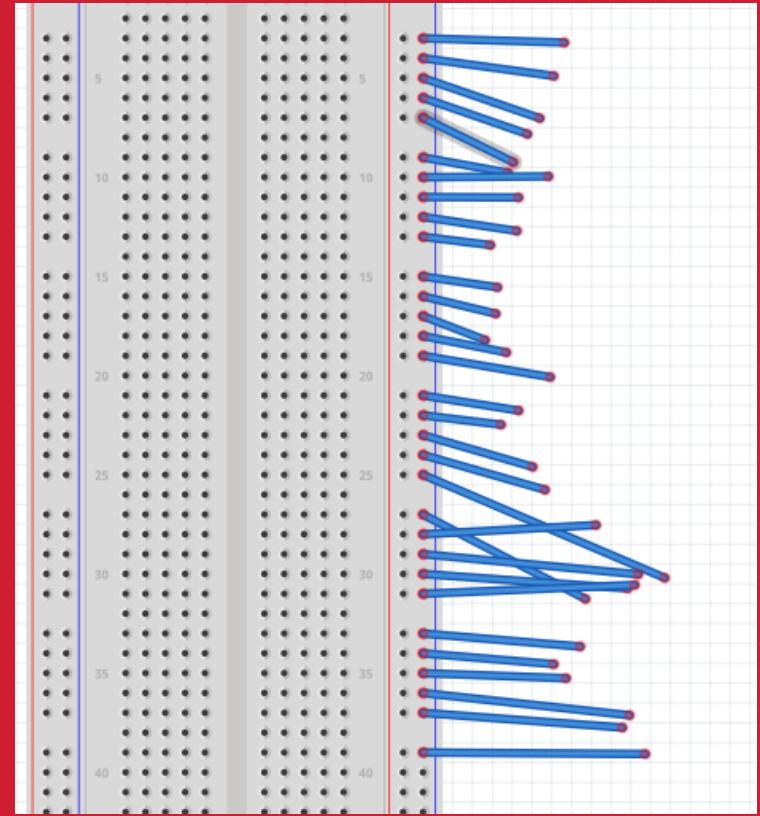
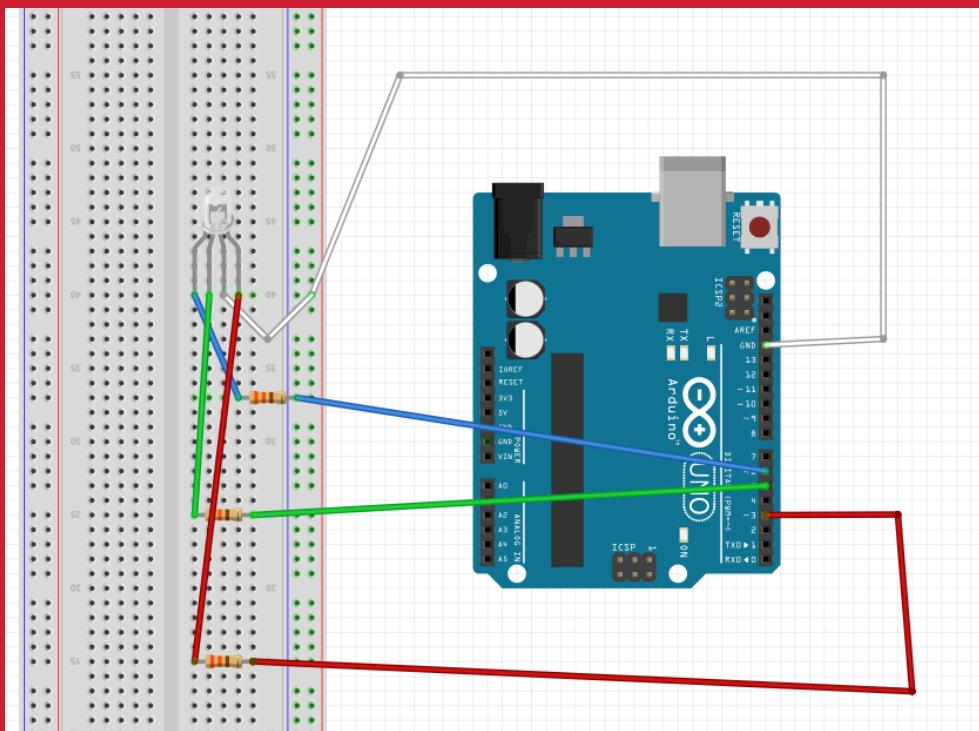
Mid exam(!!)



**최고: 13
최저: 1**



Mid 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

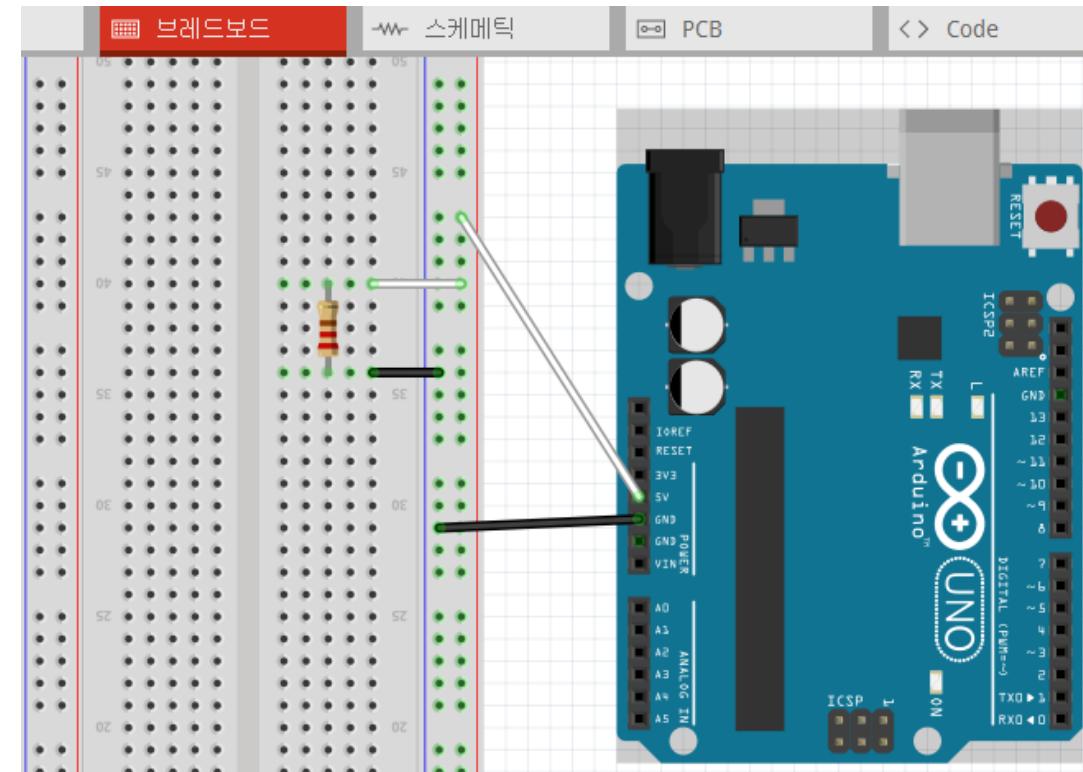
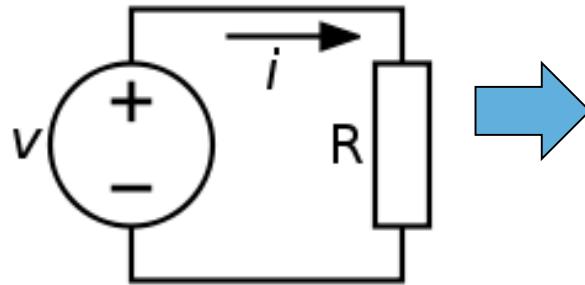
Fritzing can only act as a creative platform if many

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



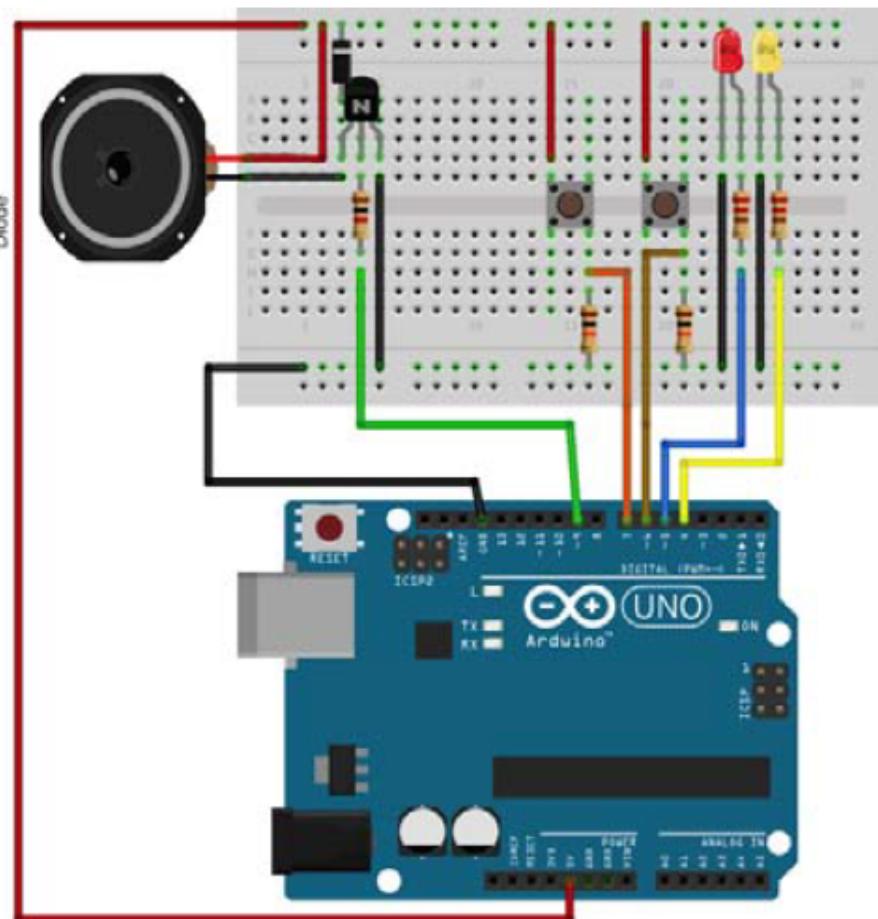
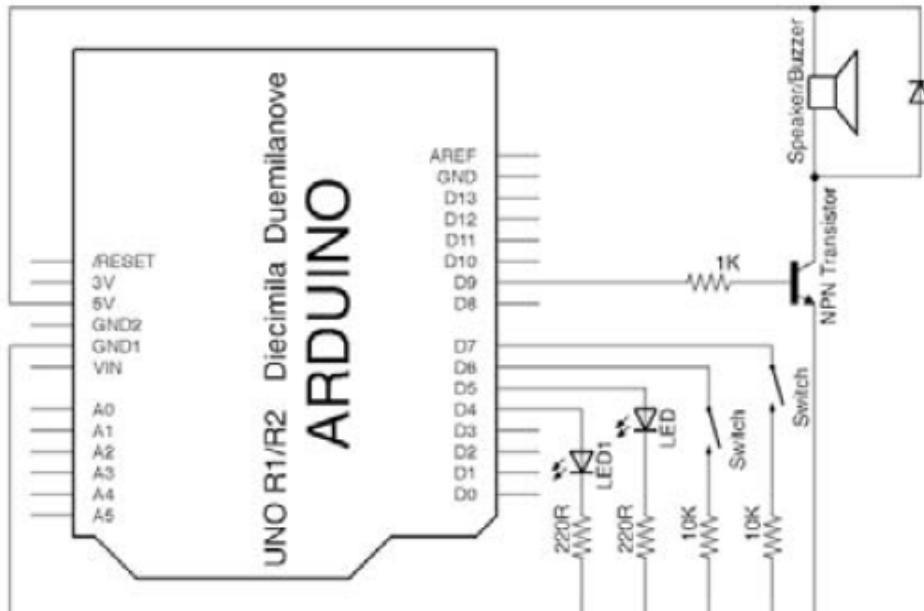
DIY : my 1st circuit

[Fritzing] Simple R circuit





Arduino circuits : challenge – (Metronome)



Circuit schematic and breadboard connections diagram for a digital metronome



Arduino

Starter Kit



Arduino Starter Kit





Arduino Starter Kit : contents

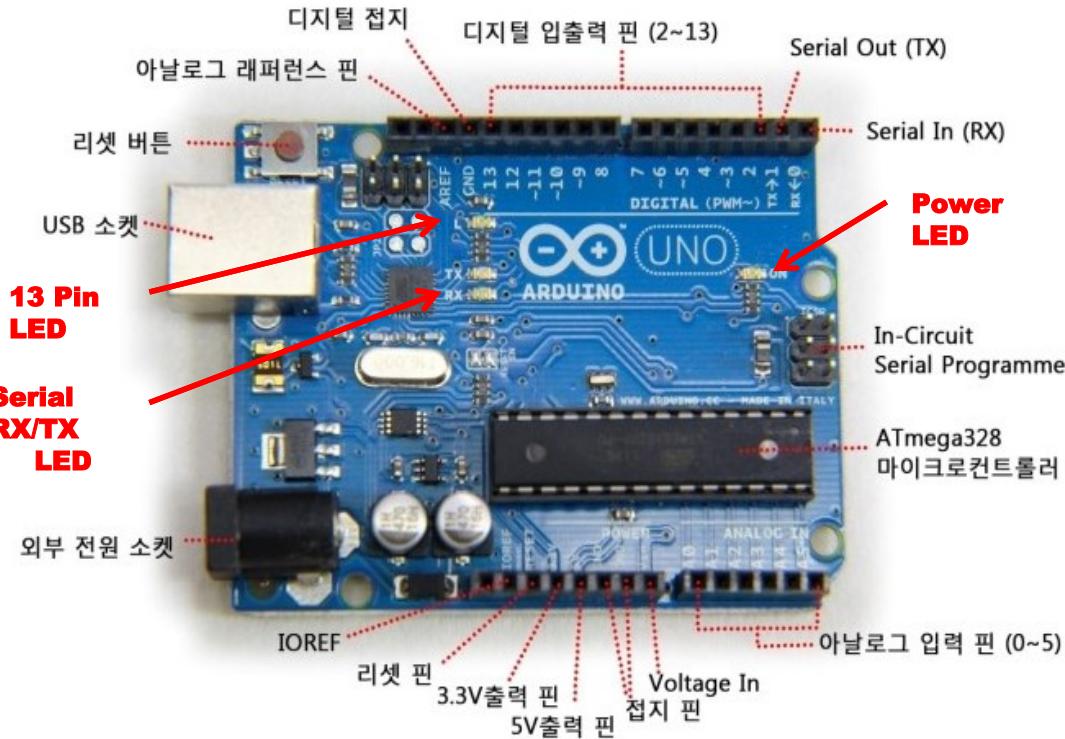
Arduino set

- **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 hardware : Uno R3

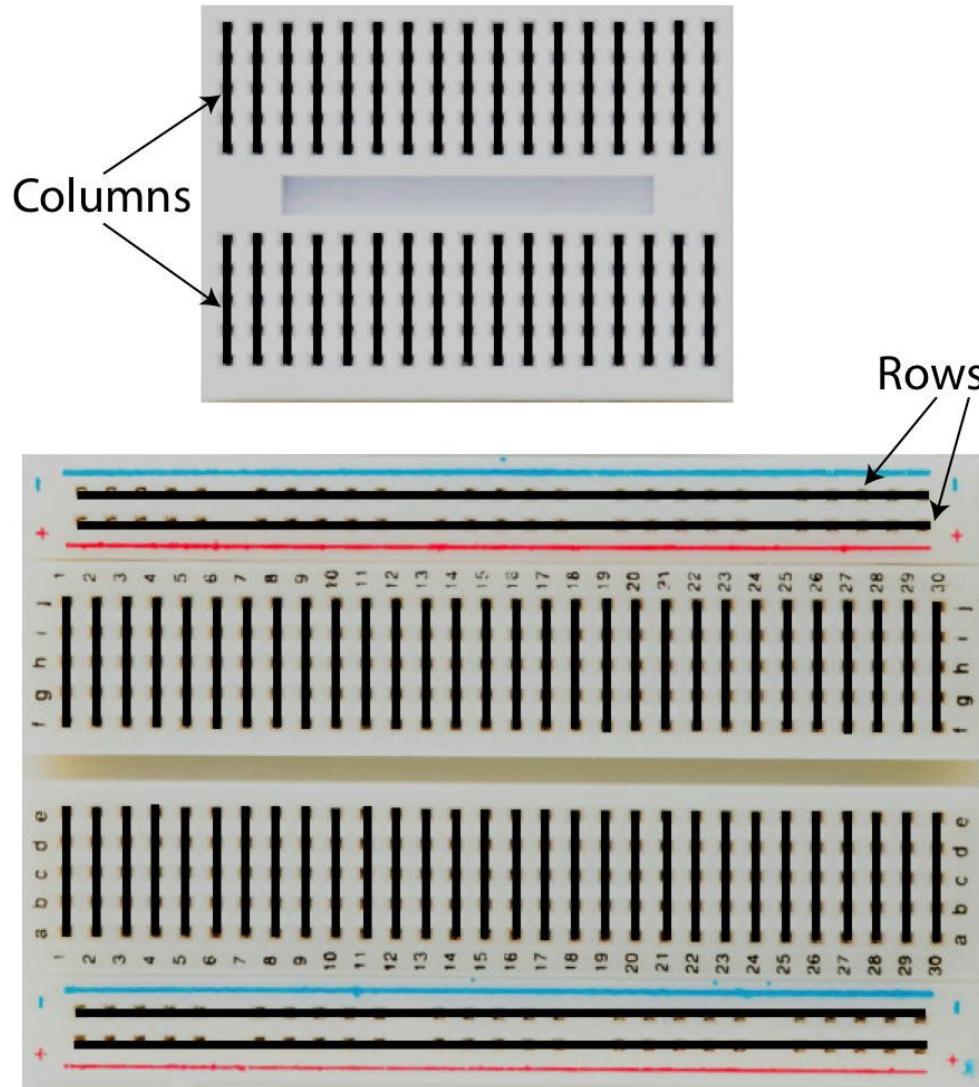


✓ Arduino UNO R3

- ATmega328 microcontroller
- Input voltage: 7~12V
- 14 Digital I/O Pins (6 PWM outputs)
- 6 Analog Inputs
- 32KB Flash Memory
- 16Mhz Clock Speed

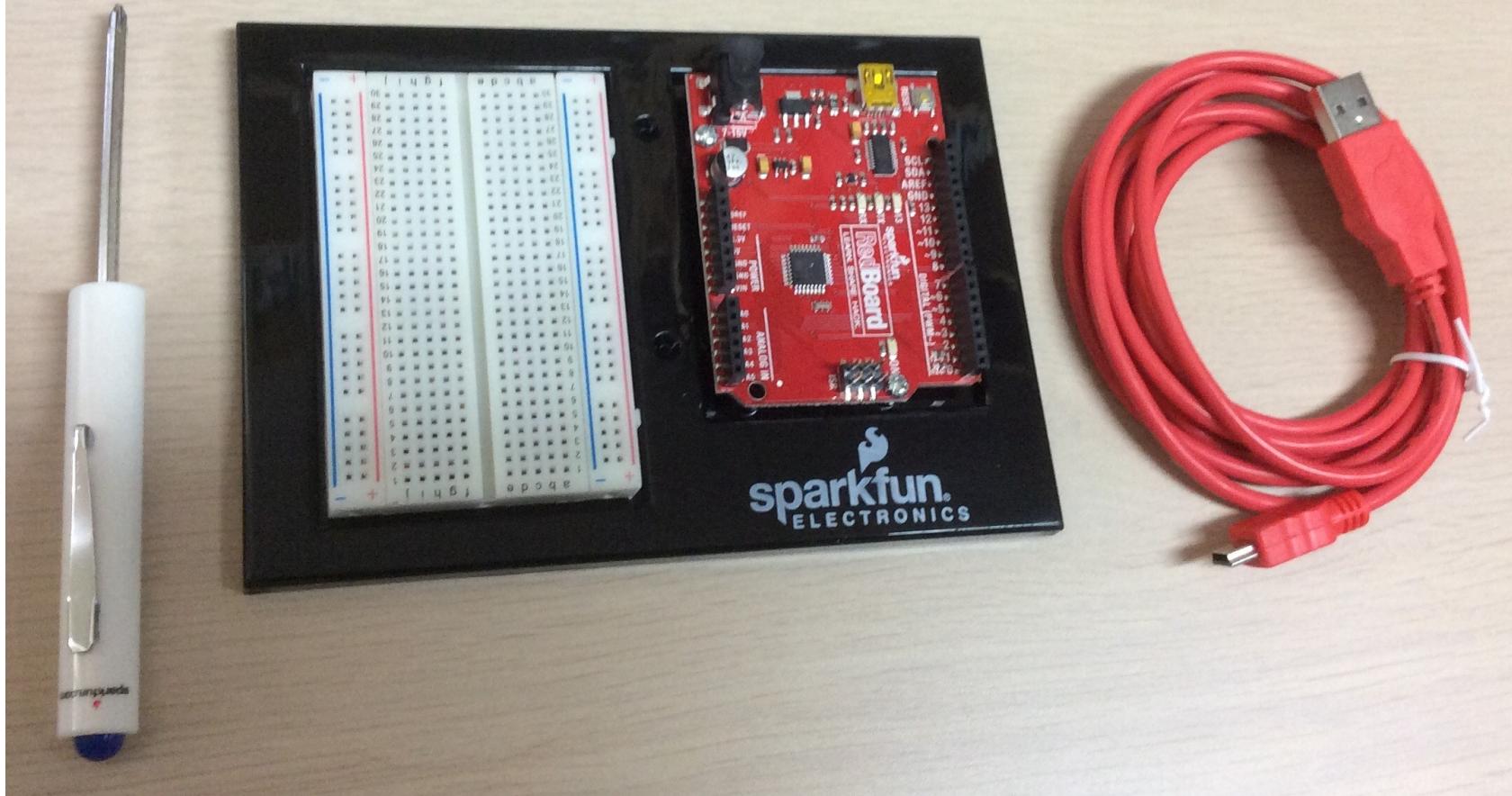


브레드 보드 (Bread board)





Arduino Starter Kit (Board setting)



Arduino coding I.



- 1. Arduino SW : IDE**
- 2. Blink a LED**
- 3. Serial monitor**



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 shade), PRODUCTS, LEARNING, FORUM, SUPPORT, and BLOG. The background of the header is a teal color.

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



1.1 Arduino IDE

Download the Arduino IDE



ARDUINO 1.8.2

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions.

Windows ZIP file for non admin install

Windows app [Get](#)

Mac OS X 10.7 Lion or newer

Linux 32 bits

Linux 64 bits

Linux ARM

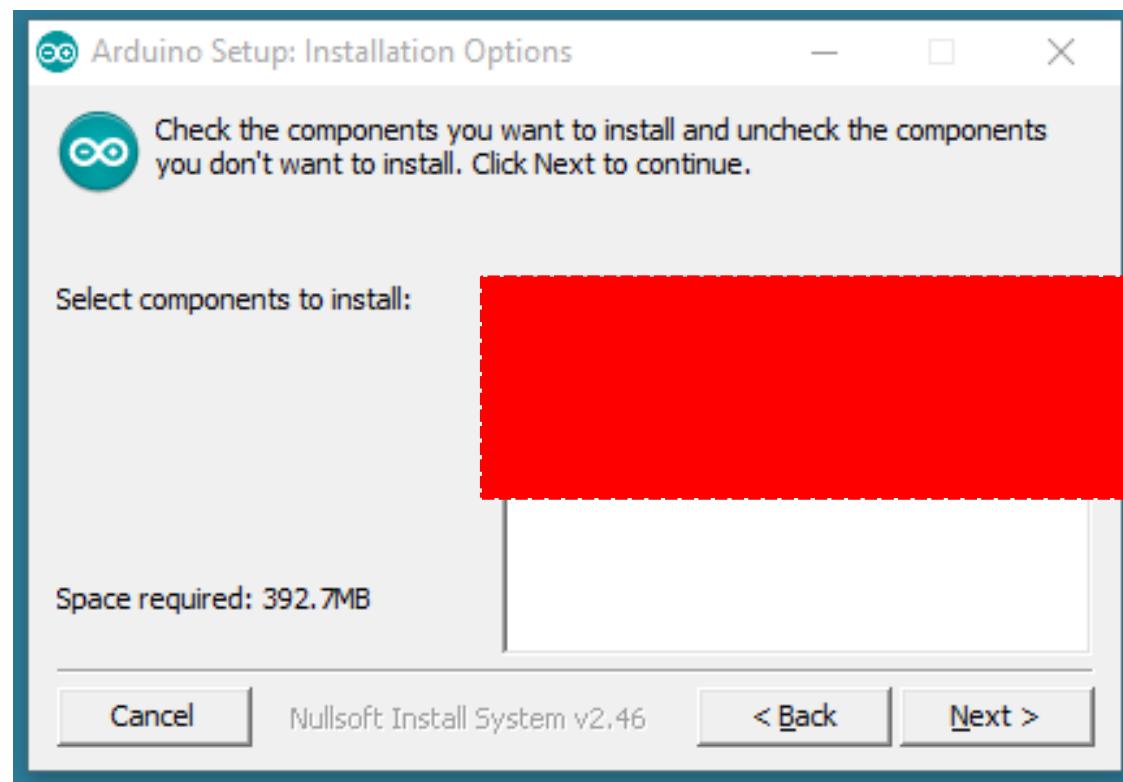
[Release Notes](#)

[Source Code](#)

[Checksums \(sha512\)](#)

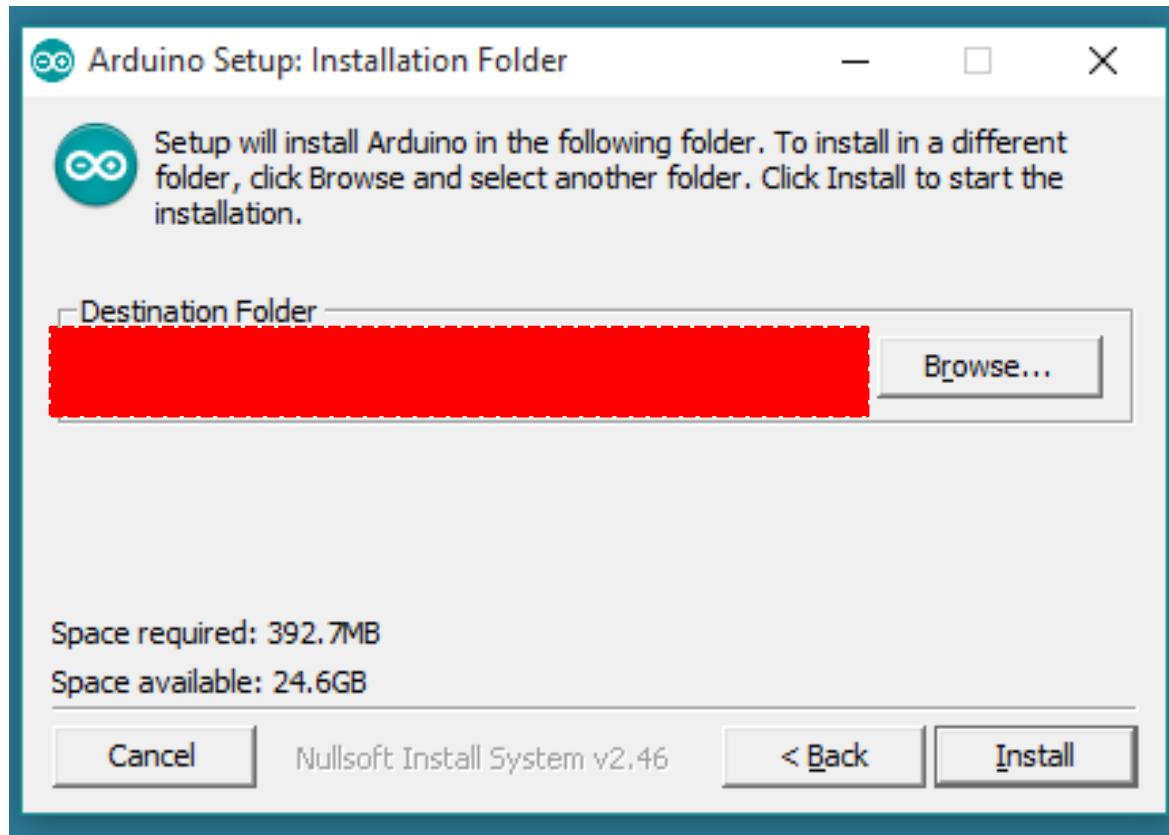


1.2 Arduino IDE



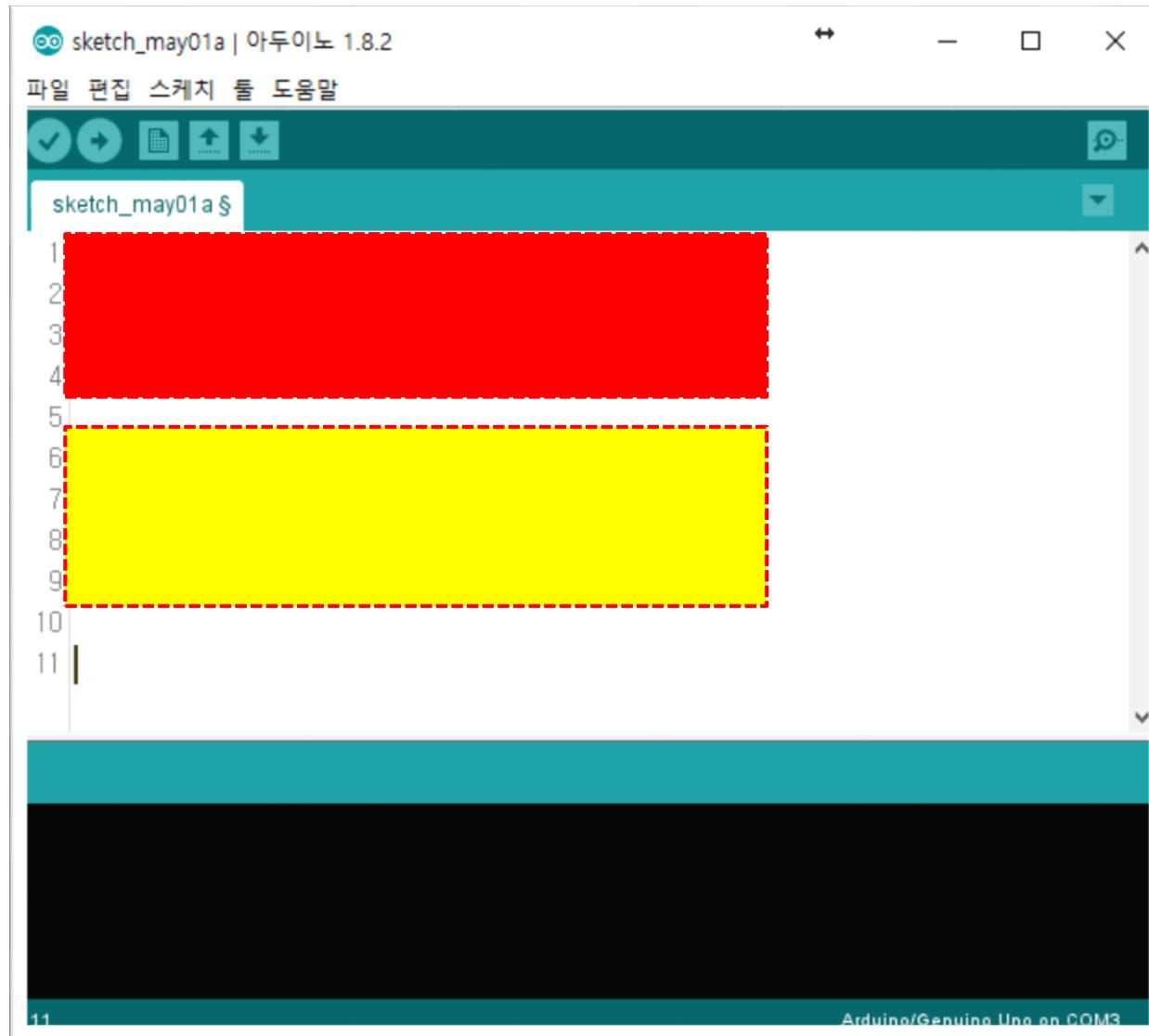


1.3 Arduino IDE





1.4 Arduino IDE





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

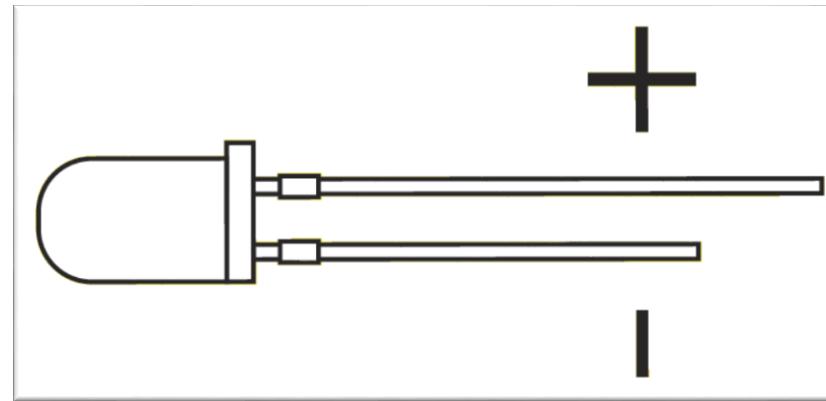




2. Blink

a LED

Polarity of LED



Find the longer leg, which should indicate the positive, anode pin.

<https://learn.sparkfun.com/tutorials/polarity/diode-and-led-polarity>



2.1 Blink a LED!

The screenshot shows the Arduino IDE interface with the following details:

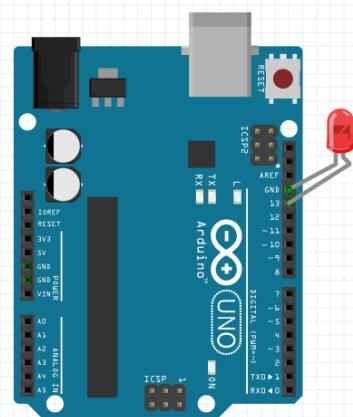
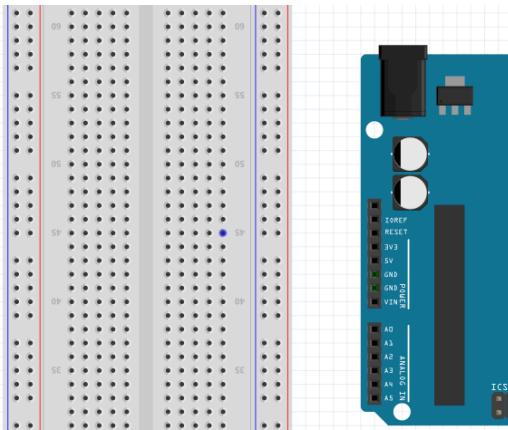
- Title Bar:** hp00_blink | 아두이노 1.8.2
- Menu Bar:** 파일 편집 스케치 둘 도움말
- Toolbar:** Checkmark, Refresh, Save, Open, New, Print, Help
- Code Editor:** The code for the 'Blink' sketch is displayed. Lines 10 and 11 are highlighted with a red rectangle.

```
1 /*  
2  * Blink  
3  * Turns on an LED on for one second, then off for one second, repeatedly.  
4 */  
5  
6  
7 // the setup function runs once when you press reset or power the board  
8 void setup() {  
9     // initialize digital pin LED_BUILTIN as an output.  
10    // [REDACTED LINE]  
11 }  
12  
13 // the loop function runs over and over again forever  
14 void loop() {  
15     digitalWrite(LED_BUILTIN, HIGH);      // turn the LED on (HIGH is the voltage level)  
16     delay(1000);                      // wait for a second  
17     digitalWrite(LED_BUILTIN, LOW);       // turn the LED off by making the voltage LOW  
18     delay(1000);                      // wait for a second  
19 }
```

- Upload Status:** 업로드 완료.
- Message Area:** 스케치는 프로그램 저장 공간 928 바이트(2%)를 사용, 최대 32256 바이트.
전역 변수는 동적 메모리 9바이트(0%)를 사용, 2039바이트의 지역변수가 남음, 최대는 2048 바이트.
- Bottom Bar:** 16, Arduino/Genuino Uno on COM3



2.2 Blink a LED!



```
Blink $  
1 /*  
2  * Blink  
3  * Turns on an LED on for one second, then off for one second, repeatedly.  
4 */  
5  
6  
7 // the setup function runs once when you press reset or power the board  
8 void setup() {  
9     // initialize digital pin 13 as an output.  
10    //  
11 }  
12  
13 // the loop function runs over and over again forever  
14 void loop() {  
15     digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)  
16     delay(1000); // wait for a second  
17     digitalWrite(13, LOW); // turn the LED off by making the voltage LOW  
18     delay(1000); // wait for a second  
19 }
```



DIY1. Blink a LED

Change output pin to 12

```
Blink $  
1 /*  
2  Blink  
3   Turns on an LED on for one second, then off for one second, repeatedly.  
4 */  
5  
6  
7 // the setup function runs once when you press reset or power the board  
8 void setup() {  
9   // initialize digital pin 13 as an output.  
10 }  
11  
12  
13 // the loop function runs over and over again forever  
14 void loop() {  
15   // turn the LED on (HIGH is the voltage level)  
16   delay(1000);  
17   // wait for a second  
18   // turn the LED off by making the voltage LOW  
19   delay(1000);  
20   // wait for a second  
21 }
```

What do you do?

HPnn_blink.png



3. Serial monitor & plotter



3. Serial monitor & plotter

COM3 (Arduino/Genuino Uno)

```
14 sec
15 sec
16 sec
17 sec
18 sec
19 sec
20 sec
0 sec
1 sec
2 sec
3 sec
4 sec
5 sec
6 sec
```

자동 스크롤

line ending 없음

9600 보드레이트

Serial moniter





3. Serial monitor & plotter

- 3.1 Arduino에서 컴퓨터로 데이터 전송하기
- 3.2 변수 유형별로 컴퓨터에 전송하기
- 3.3 Arduino에서シリ얼 통신을 이용하여
데이터 수신하기 (next week)

3. 시리얼 통신 (Serial communication)

시리얼 통신

UART (Universal Asynchronous Receiver/Transmitter)

RS-232

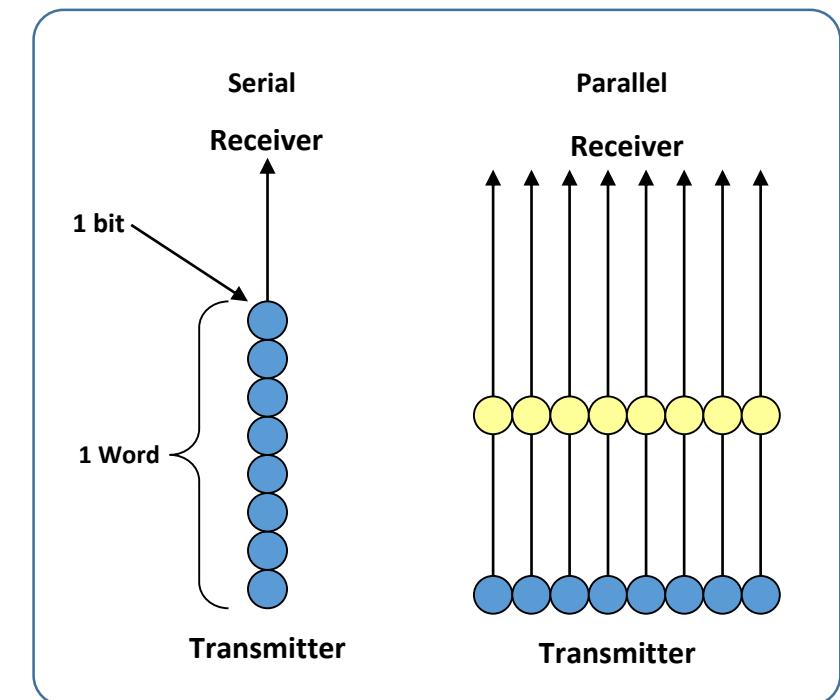
RS-422

RS-485

Arduino에서는 다음과 같은 목적으로 사용

Debugging : 프로그램의 오류를 수정하는 작업

데이터 통신 : Arduino와 컴퓨터 혹은 다른 장치와의
통신





3.1.1 Arduino에서 컴퓨터로 데이터 전송하기

EX 2.1

Arduino에서 컴퓨터로 변수와 문자열 전송하기 (1/3)

- 실습목표**
1. Arduino에서 문자열과 데이터를 시리얼 통신을 이용하여 컴퓨터로 전송한다.
 2. 전송할 데이터는 0부터 1초 간격으로 1씩 증가하는 숫자와 'sec'라는 문자열이다.
 3. Arduino IDE의 시리얼 모니터에서 이를 확인해 본다.

Hardware Arduino와 PC를 USB 케이블로 연결한다.

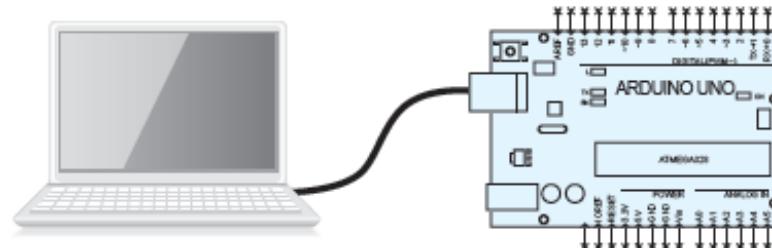


그림 2.1 Arduino와 PC와의 연결



3.1.2 Arduino에서 컴퓨터로 데이터 전송하기

EX 2.1

Arduino에서 컴퓨터로 변수와 문자열 전송하기 (2/3)

Commands

- **Serial.begin**(전송속도)

시리얼 통신 포트를 컴퓨터와 연결한다. 전송속도는 bps(bits per sec)로 일반적으로 9600으로 설정한다. 19200, 57600, 115200 등의 값을 설정할 수 있다.

- **Serial.print**(전송내용)

괄호 안의 내용을 시리얼 통신으로 전송한다. 따옴표로 구분된 부분은 텍스트를 직접 전송하고 따옴표 없이 변수를 써주면 변수의 값이 전송된다.

- **Serial.println**(전송내용)

'Serial.print'와 같으나 전송 뒤 줄 바꿈을 한다.

- **delay**(지연시간 in ms)

지연시간에는 잠시 동작을 지연시키기 위한 값을 넣는다. 1/1000초 단위로 넣는다.
즉 1초를 지연시키기 위해선 1000의 값을 입력시킨다.



3.1.3 Arduino에서 컴퓨터로 데이터 전송하기

ex_2_1 | 아두이노 1.8.2

파일 편집 스케치 둘 도움말



ex_2_1

```
1 /*  
2 예제 2.1  
3 Arduino에서 컴퓨터로 변수와 문자열 전송하기  
4 */  
5  
6 int number = 0; // -32768~32767 범위의 변수 number 설정, 초기값은 0  
7  
8 void setup() {  
9   Serial.begin(9600); // 9600bps로 시리얼 통신 설정  
10 }  
11  
12 void loop() {  
13   Serial.print(number); // number 변수값 출력  
14   Serial.println(" sec"); // " sec"를 출력 후 줄 바꿈  
15   delay(1000); // 1초동안 지연시킨다.  
16   number++; // number 변수값을 하나 증가시킨다.  
17 }
```

업로드 완료.

스케치는 프로그램 저장 공간 1862 바이트(5%)를 사용. 최대 32256 바이트.

전역 변수는 동적 메모리 192바이트(9%)를 사용, 1856바이트의 지역변수가 남음. 최대는 2048 바이트.



COM3 (Arduino/Genuino Uno)

36 sec
37 sec
38 sec
39 sec
40 sec
41 sec
42 sec
43 sec
44 sec
45 sec
46 sec
47 sec
48 sec
49 sec
50 sec
51 sec
52 sec
53 sec
54 sec
55 sec
56 sec
57 sec
58 sec
59 sec
60 sec

자동 스크롤



3.1.4 Arduino에서 컴퓨터로 데이터 전송하기

EX 2.1

Arduino에서 컴퓨터로 변수와 문자열 전송하기 (3/3)

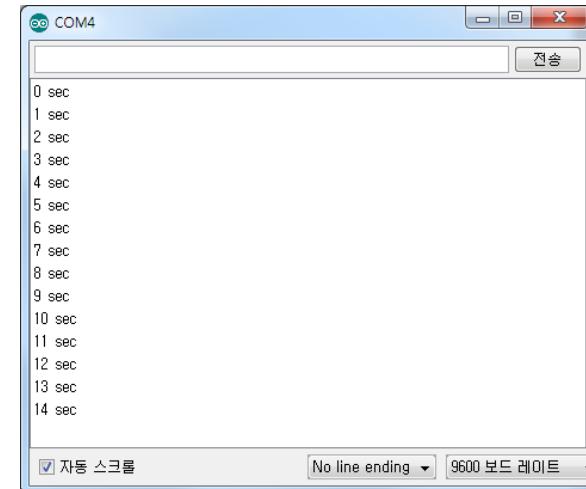
Sketch 구성 1. 시리얼통신을 시작한다. 'Serial.begin()' 명령어로 할 수 있다.

2. 변수를 전송하기 위해 'Serial.print()' 명령어를 사용하고,

문자열 전송 후 줄 바꿈을 하기 위해서 'Serial.println()' 명령어를 사용한다.

3. 루프를 1초마다 실행하기 위해서 'delay()'명령어로 시간지연을 시켜준다.

실습 결과 IDE의 시리얼 모니터를 실행시켜 Arduino에서 전송되는 메시지를 확인할 수 있다.



응용 문제 1. 2초, 5초 단위로 시간을 변경해 보자.

2. 자신만의 메시지를 출력해보자.

3. [DIY1]

delay를 0.2초로 설정후

5초마다 number를 초기화하여

시리얼플로터로 톱니파를 발생.

시간은 ms로 계산해서 출력

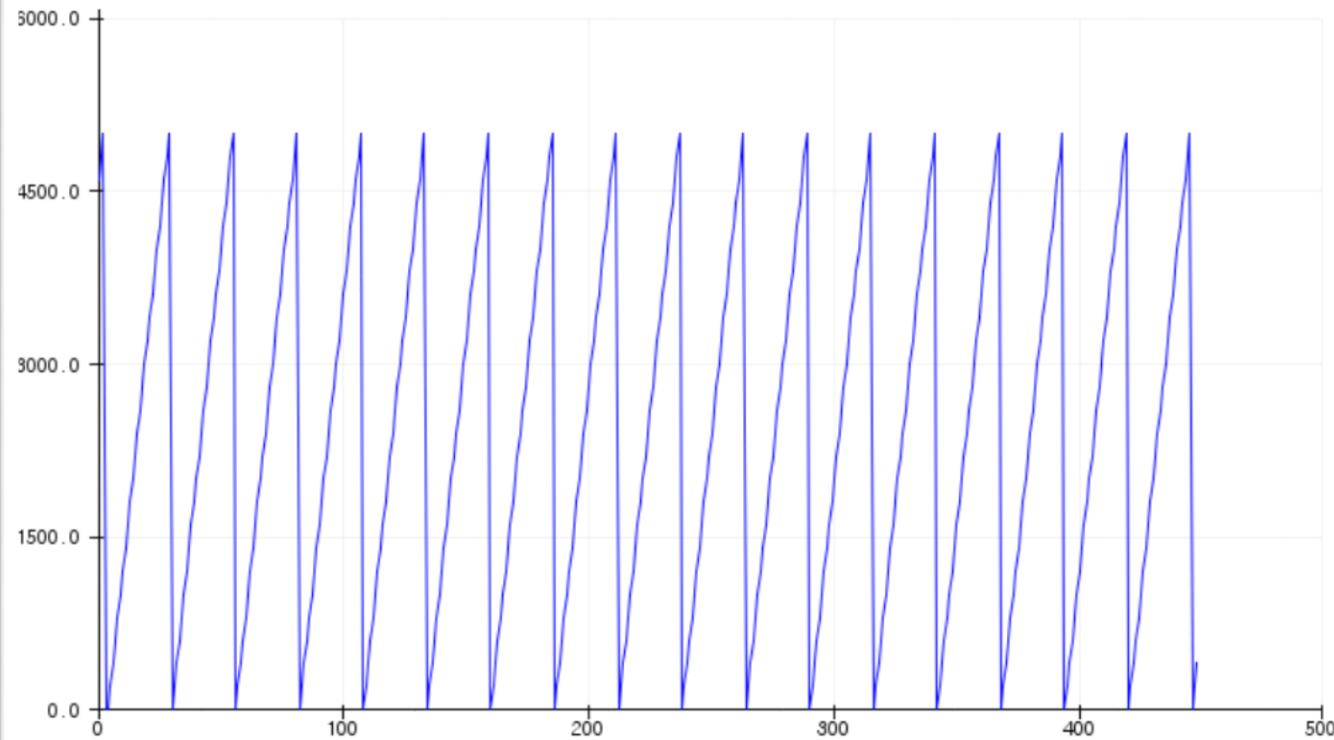


DIY2. sawtooth signal

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

COM3 (Arduino/Genuino Uno)



HPnn_sawtooth.png



3.2.1 변수 유형별로 컴퓨터에 전송하기

- ✓ 사용 목적에 따라 다양한 변수 유형 중 선택하여 사용
- ✓ C 언어와 유사함

표 2.1 Arduino 변수 유형

변수 유형	바이트	변수 범위	용도
boolean	1	true(1) 혹은 false(0)	참(1) 아니면 거짓(0) 값을 나타냄.
char	1	-128~127 혹은 아스키 코드 값과 매칭되는 문자	부호가 있는 1바이트 숫자를 나타냄
unsigned char	1	0~255	0~255의 숫자를 나타낼 때 사용함
byte	1	0~255	unsigned char과 동일
int	2	-32,768~32,767	부호가 있는 2바이트 숫자를 나타냄
short	2	-32,768~32,767	int와 동일
unsigned int	2	0~65,535	부호가 없는 2바이트 숫자를 나타냄
word	2	0~65,535	unsigned int와 동일
long	4	-2,147,483,648 ~2,147,483,647	부호가 있는 4바이트 숫자를 나타냄
unsigned long	4	0~4,294,967,295	부호가 없는 4바이트 숫자를 나타냄
float	4	$-3.4028235 \times 10^{38}$ $\sim 3.4028235 \times 10^{38}$	소수점 있는 숫자를 나타냄
double ¹⁾	4 or 8	$-3.4028235 \times 10^{38}$ $\sim 3.4028235 \times 10^{38}$ Or $-1.79 \times 10^{308} \sim 1.79 \times 10^{308}$	소수점 있는 숫자를 나타냄. 4바이트의 경우 float과 동일 8바이트의 경우 변수 범위가 커짐
String	-	문자열에 따라 다름	문자열을 나타냄.



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



3.2.3 변수 유형별로 컴퓨터에 전송하기

EX 2.2

변수 유형별 Arduino에서 컴퓨터로 전송하기 (1/4)

- 실습목표**
1. Arduino에서 컴퓨터로 데이터를 전송할 때 변수 유형별로 출력한다.
 2. char로 선언된 변수, int로 선언된 변수, float로 선언된 변수를 'Serial.print' 명령어를 이용하여 PC로 전송하자.
 3. 'Serial.print' 명령어의 출력 옵션을 변경하여 전송해 보자.
 4. 'Serial..write' 명령어로 문자열을 출력해 보자.
 5. 각 변수 유형별 출력되는 차이를 비교해 보자.

Hardware Arduino와 PC를 USB 케이블로 연결한다.

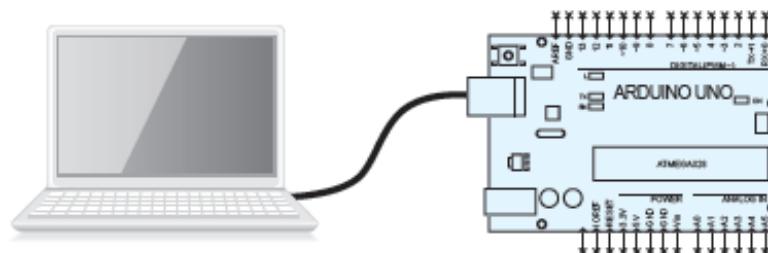


그림 2.1 Arduino와 PC와의 연결



3.2.4 변수 유형별로 컴퓨터에 전송하기

EX 2.2

변수 유형별 Arduino에서 컴퓨터로 전송하기 (2/4)

Commands

- `Serial.write(char 변수);`

char변수에 해당하는 ASCII 코드값의 문자를 출력한다.

- `Serial.print(변수,BIN);`

변수를 2진수(Binary)로 표시한다.

- `Serial.print(변수,DEC);`

변수를 10진수(Binary)로 표시한다.

- `Serial.print(변수,HEX);`

정해진 변수를 16진수(Hexadecimal)로 표시한다.



3.2.5 변수 유형별로 컴퓨터에 전송하기

EX 2.2

변수 유형별 Arduino에서 컴퓨터로 전송하기 (3/4)

- Sketch 구성
1. '65'란 숫자를 char형, int형, float형 변수에 각각 저장한다.
 2. 'Binary:', 'Decimal:', 'Hexadecimal:', 'ASCII:' 등 네 가지 문자열을 저장하여 호출하여 사용한다.
 3. 변수에 저장된 숫자를 2진수형, 10진수형, 16진수형, ASCII 코드형 등 `Serial.print` 명령어의 옵션을 변경하여 전송한다.
 4. 옵션이 변경될 때마다 문자열을 호출하여 함께 출력한다.
 5. `loop`가 반복될 때마다 숫자를 1씩 증가시킨다. float형은 0.1씩 증가시킨다.



3.2.6 변수 유형별로 컴퓨터에 전송하기

EX 2.2

변수 유형별 Arduino에서 컴퓨터로 전송하기 (3/4)

- Sketch 구성
1. '65'란 숫자를 char형, int형, float형 변수에 각각 저장한다.
 2. 'Binary:', 'Decimal:', 'Hexadecimal:', 'ASCII:' 등 네 가지 문자열을 저장하여 호출하여 사용한다.
 3. 변수에 저장된 숫자를 2진수형, 10진수형, 16진수형, ASCII 코드형 등 `Serial.print` 명령어의 옵션을 변경하여 전송한다.
 4. 옵션이 변경될 때마다 문자열을 호출하여 함께 출력한다.
 5. `loop`가 반복될 때마다 숫자를 1씩 증가시킨다. float형은 0.1씩 증가시킨다.

3.2.7 변수 유형별로 컴퓨터에 전송하기

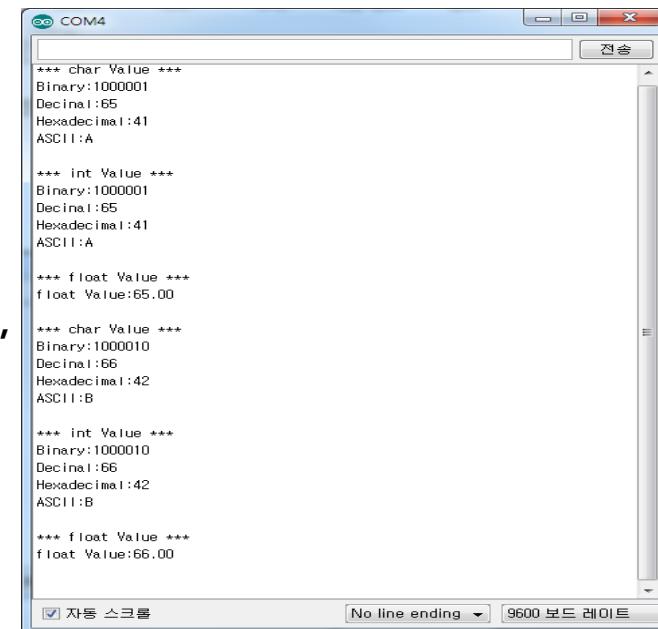
EX 2.2

변수 유형별 Arduino에서 컴퓨터로 전송하기 (4/4)

실습 결과 IDE의 시리얼 모니터를 실행시켜

Arduino에서 전송되는 메시지를
확인할 수 있다.

10초 간격으로 증가된 값에 대하여
출력하게 된다. 이때 2진수, 10 진수,
16진수로 표시되고, 증가된 숫자에
해당하는 아스키코드의 문자가 전송
된다. 'float'형 변수는 소수점이 함
께 표시된다.



The screenshot shows the Arduino Serial Monitor window titled 'COM4'. It displays five sets of data transmissions, each consisting of four lines: 'char Value', 'int Value', 'float Value', and another 'char Value'. The data is transmitted at a rate of 9600 baud. The first set shows 'A' as the character value. Subsequent sets show the character value increasing by one each time (B, C, D, E). The binary, decimal, and hexadecimal representations of the values also change accordingly.

Transmission Type	Value	Binary	Decimal	Hexadecimal	ASCII
char Value	A	1000001	65	41	A
int Value	65	1000001	65	41	A
float Value	65.00				
char Value	B	1000010	66	42	B
int Value	66	1000010	66	42	B
float Value	66.00				
char Value	C	1000011	67	43	C
int Value	67	1000011	67	43	C
float Value	67.00				
char Value	D	1000100	68	44	D
int Value	68	1000100	68	44	D
float Value	68.00				
char Value	E	1000101	69	45	E
int Value	69	1000101	69	45	E
float Value	69.00				

Quiz 1. `Serial.write(floatValue);`

위의 명령이 오류가 나는 이유는?



DIY3. 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 number를 1씩 증가**
3. **옆의 방식으로 결과 출력**
4. **number가 15를 초과하면 loop() 탈출**
exit(0); // loop 탈출 함수

HPnn_loop_escape.png



DIY4. 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 `number`를 1씩 증가시키면서 `sum`에 합한다.
4. 옆의 방식으로 결과 출력
5. `startNum` 가 100을 초과하면 `loop()` 탈출
`exit(0); // loop 탈출 함수`

HPnn_sum100.png



[Practice]

◆ [wk09]

- **Arduino coding I.**
- **Complete your codes**
- **Save results in png files.**
- **Upload file name : HPnn_Rpt04.zip**

◆ [Target of this week]

- Complete your 4th project
- Save your outcome and compress all png files.

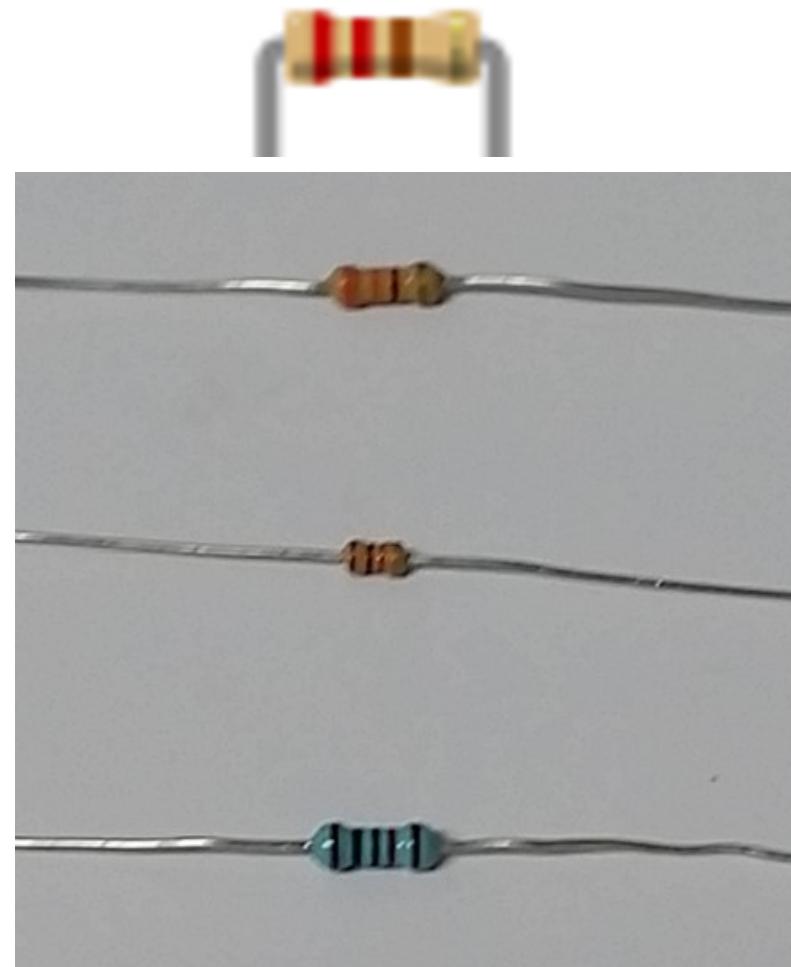
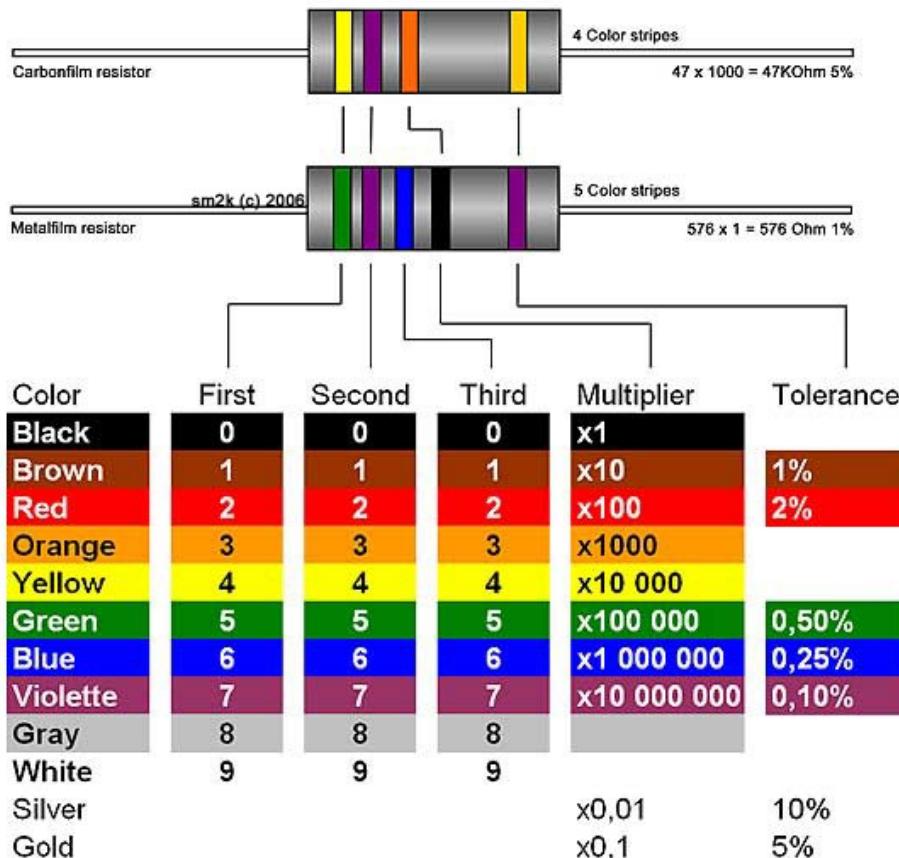
제출파일명 : **HPnn_Rpt04.zip**

- 압축할 파일들

- ① **HPnn_blink.png**
- ② **HPnn_sawtooth.png**
- ③ **HPnn_loop_escape.png**
- ④ **HPnn_sum100.png**

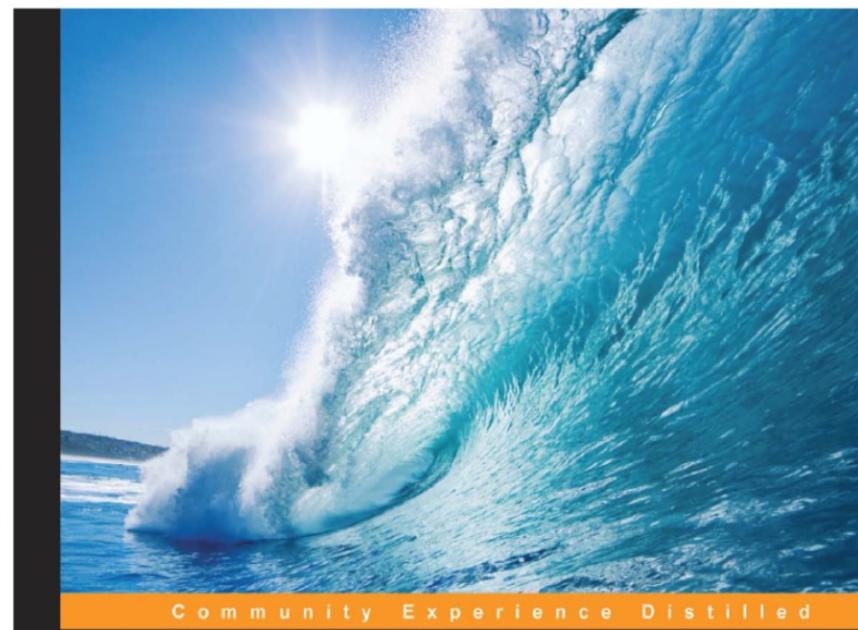


[참고 : 저항 값 읽기]





References



Arduino Essentials

Enter the world of Arduino and its peripherals and start creating interesting projects

Francis Perea

[PACKT]
PUBLISHING

