

아두이노

임성국



아두이노용 C 언어

– 세미콜론 ;



문장... 끝 ;

{ } 안에서 ; 사용

```
void loop()  
{  
    digitalWrite(13, HIGH);  
    delay(1000);  
    digitalWrite(13, LOW);  
    delay(1000);  
}
```

{ } 안에서 ; 사용

```
void loop()
```

```
{
```

```
    digitalWrite(13, HIGH);    delay(1000);
```

```
    digitalWrite(13, LOW);    delay(1000);
```

```
}
```




아두이노용 C 언어

- 주석 /* */, //

/*

*** /**

/ *

이곳은 설명을 위한 주석입니다.
이곳에 쓰는 모든 것은 볼 수는 있지만
프로그램과는 무관합니다.

* /

{ } 안에서 ; 사용

```
void loop()  
{  
    /*  
        5번 LED 를 1초 동안 켜고  
        다시 5번 LED 를 1초 동안 끄는 동작을  
        반복하는 프로그램  
    */  
    digitalWrite(5, HIGH);  
    delay(1000);  
    digitalWrite(5, LOW);  
    delay(1000);  
}
```

//

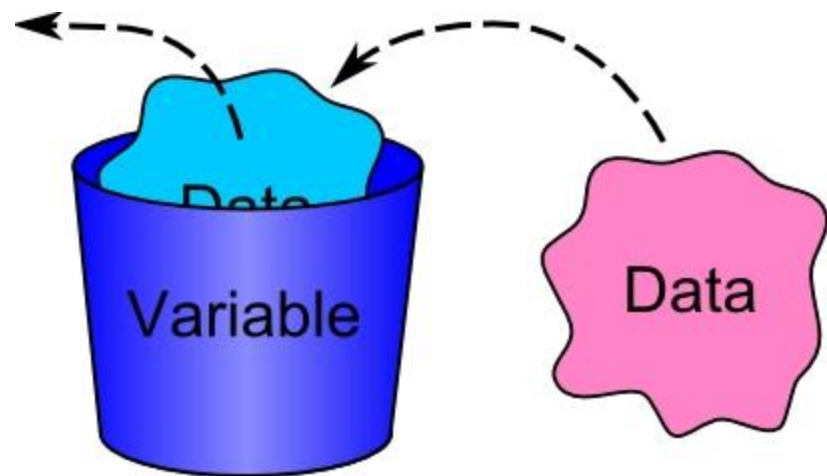
{ } 안에서 ; 사용

```
void loop()  
{  
    // loop 함수 시작  
    digitalWrite(5, HIGH); // 5번 LED 를 켜다  
    delay(1000);           // 1초 기다린다  
    digitalWrite(5, LOW);  // 5번 LED 를 끈다  
    delay(1000);           // 1초 기다린다  
}
```




아두이노용 C 언어

- 변수



```
int value;
```

```
int value;  
value = 5;
```

```
int value = 5;
```

전역변수 VS 지역변수

전역변수

- 외부에서 만들어진 변수

```
int a;  
void setup(){  
  a = 0;  
}  
void loop(){  
  a = 1;  
}
```

지역변수

- 블록 내부에서 만들어진 변수

```
void setup(){  
  int b;  
}  
void loop(){  
  b = 2;  
}
```

```
void loop(){  
    int a = 1;  
    {  
        int a = 2;  
        int b = 3;  
        int c = a + b;  
    }  
    int b = a;  
}
```

시작하는 {

시작하는 {

끝나는 }

끝나는 }



아두이노용 C 언어

- 변수의 종류

부호	이름	크기	범위
	bool	1	false, true
	byte	1	0 ~ 255
(signed)	char	1	-128 ~ 127
unsigned			0 ~ 255
(signed)	int	2	-32768 ~ 32767
unsigned			0 ~ 65535
(signed)	long	4	-2,147,483,648 ~ 2,147,483,647
unsigned			0 ~ 4,294,967,295
	float	4	$-3.4028235 \times 10^{38} \sim -3.4028235 \times 10^{38}$

BYTE

BYTE

unsigned char

1 byte = 8 bits

이진수	십진수	계산
0000 0000	0	$= 0 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$
0000 0001	1	$= 0 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
0000 0010	2	$= 0 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$
0000 0011	3	$= 0 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$
0000 0100	4	$= 0 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$
...
1111 1101	253	$= 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
1111 1110	254	$= 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$
1111 1111	255	$= 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$

byte sample = 125;

byte sample = 255;

sample = sample + 1;

byte sample = 255;

sample = sample + 1;

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

+1

1	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

CHAR

CHAR

1 byte = 8bits

```
void loop(){  
    char sample = 'A';  
    Serial.println(sample);  
}
```

이진수		십진수	
XXXX XXXX		-128 ~ 127	
$X \times 2^7 + X \times 2^6 + X \times 2^5 + X \times 2^4 + X \times 2^3 + X \times 2^2 + X \times 2^1 + X \times 2^0$			
첫번째 비트가 0		첫번째 비트가 1	
이진수	십진수	이진수	십진수
0000 0000	0	1000 0000	-128
0000 0001	1	1000 0001	-127
0000 0010	2	1000 0010	-126
0000 0011	3	1000 0011	-125
0111 1111	127	1111 1111	-1

2의 보수

ASCII TABLE

INT

INT

2byte = 16bits

LONG

LONG

4byte = 32bits

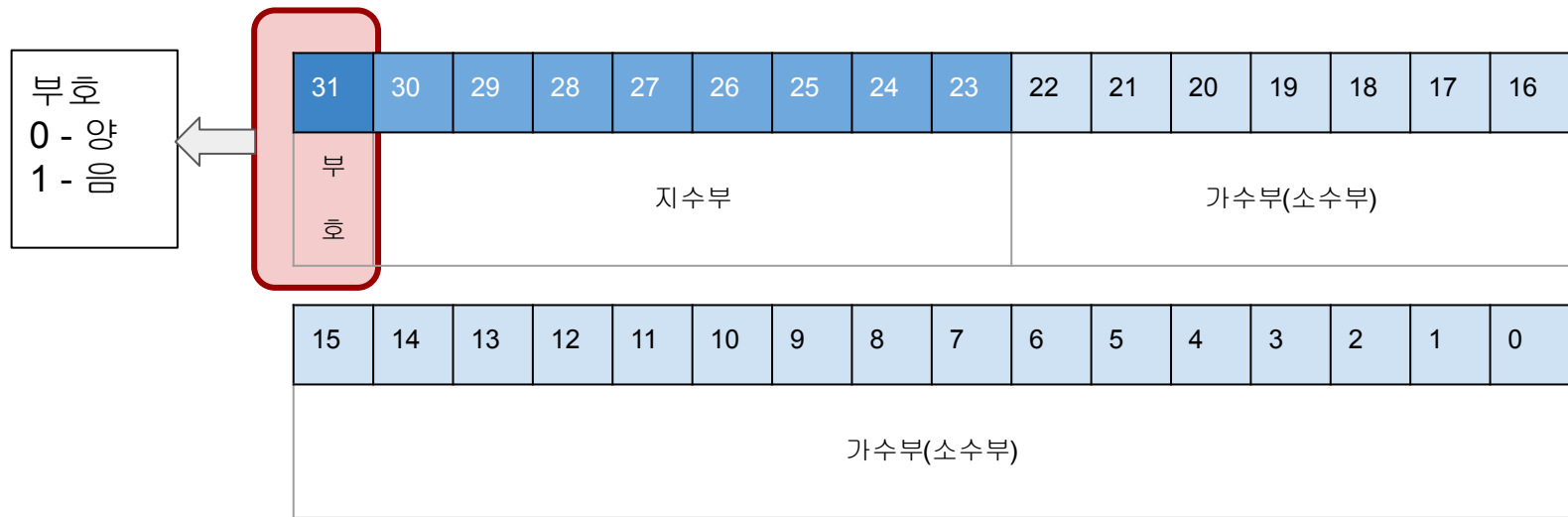
FLOAT

FLOAT

4byte = 32bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
부 하	지수부								가수부(소수부)						

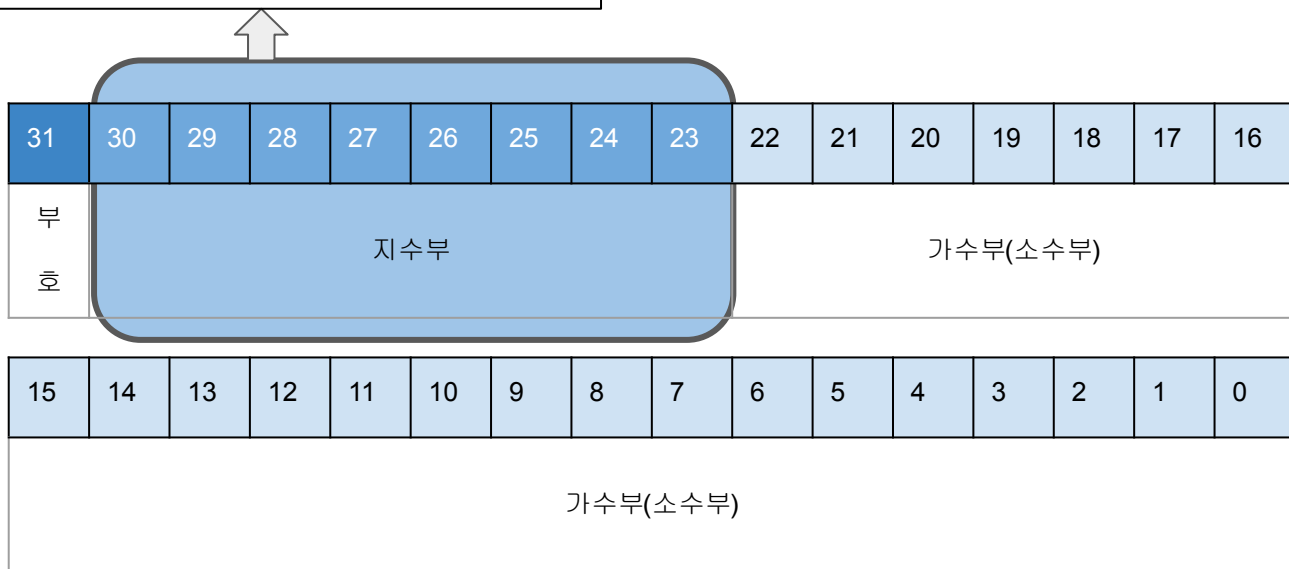
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
가수부(소수부)															



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
부 호	지수부								가수부(소수부)						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
가수부(소수부)															

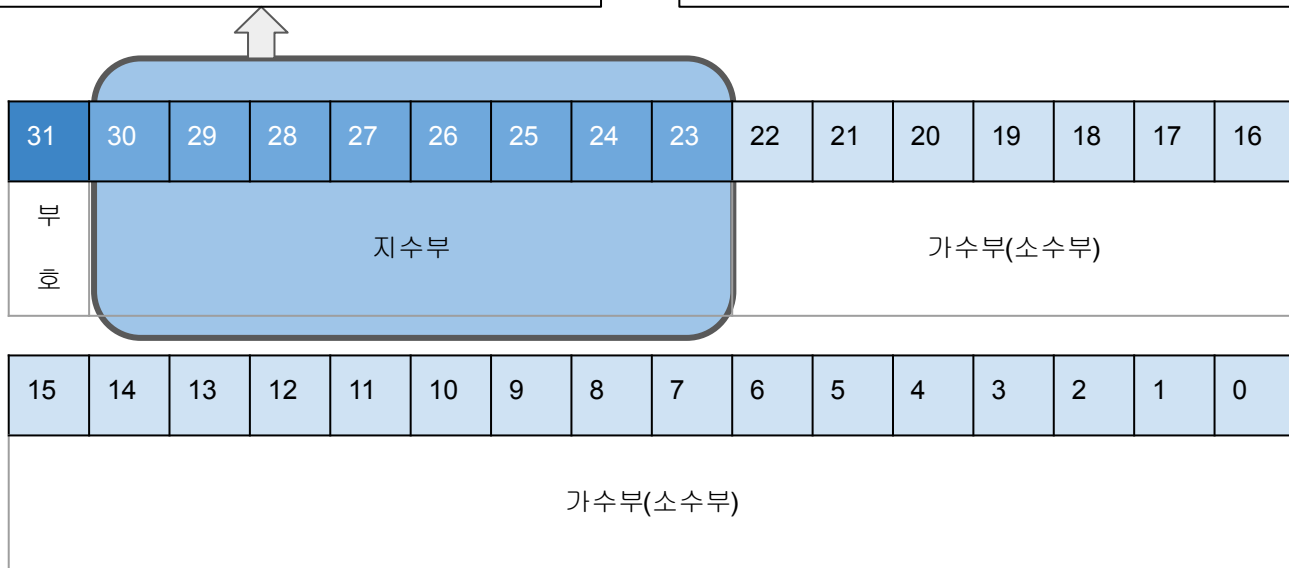
지수부 : 8 비트 0 ... 127 ... 256
표현범위 -126 ~ 128 (Bias 127)

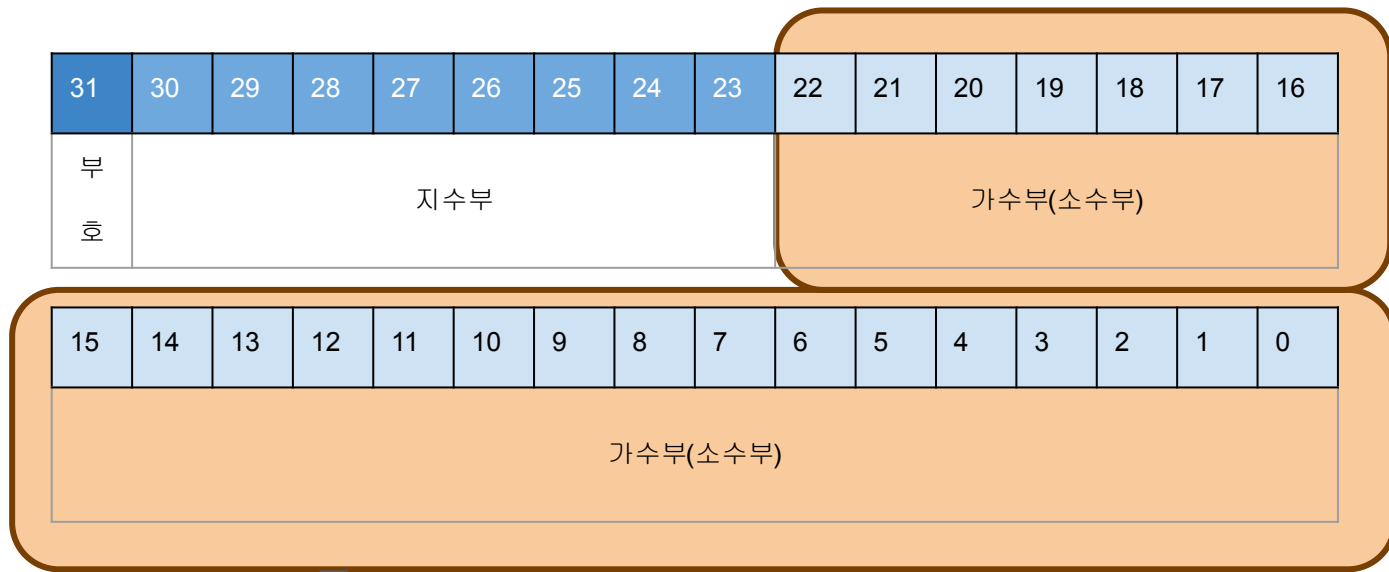


지수부 : 8 비트 0 ... 127 ... 256
표현범위 -126 ~ 128 (Bias 127)

$$2^{128} = 3.4028235 \times 10^{38}$$

$$2^{-127} = 1.1754944 \times 10^{-38}$$

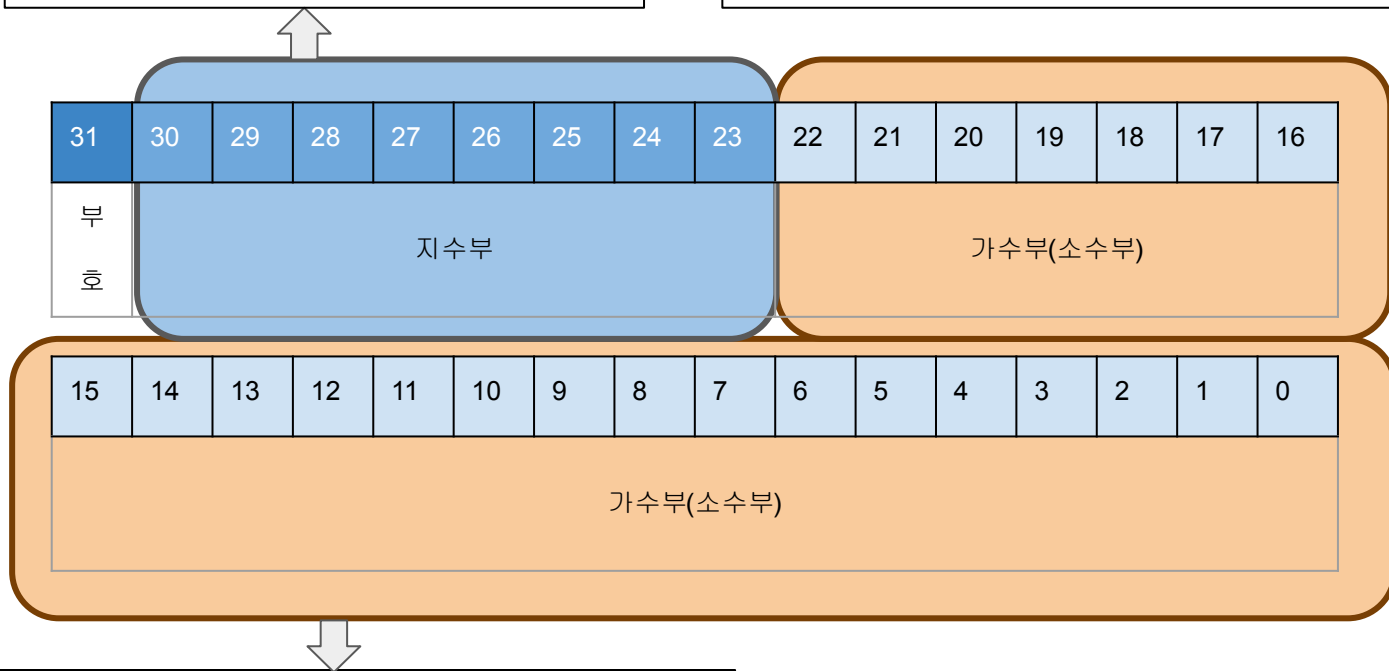




소수부 : 23비트 0 ... 8,388,608
표현범위 소수점이하 6번째 자리까지

지수부 : 8 비트 0 ... 127 ... 256
표현범위 -126 ~ 128 (Bias 127)

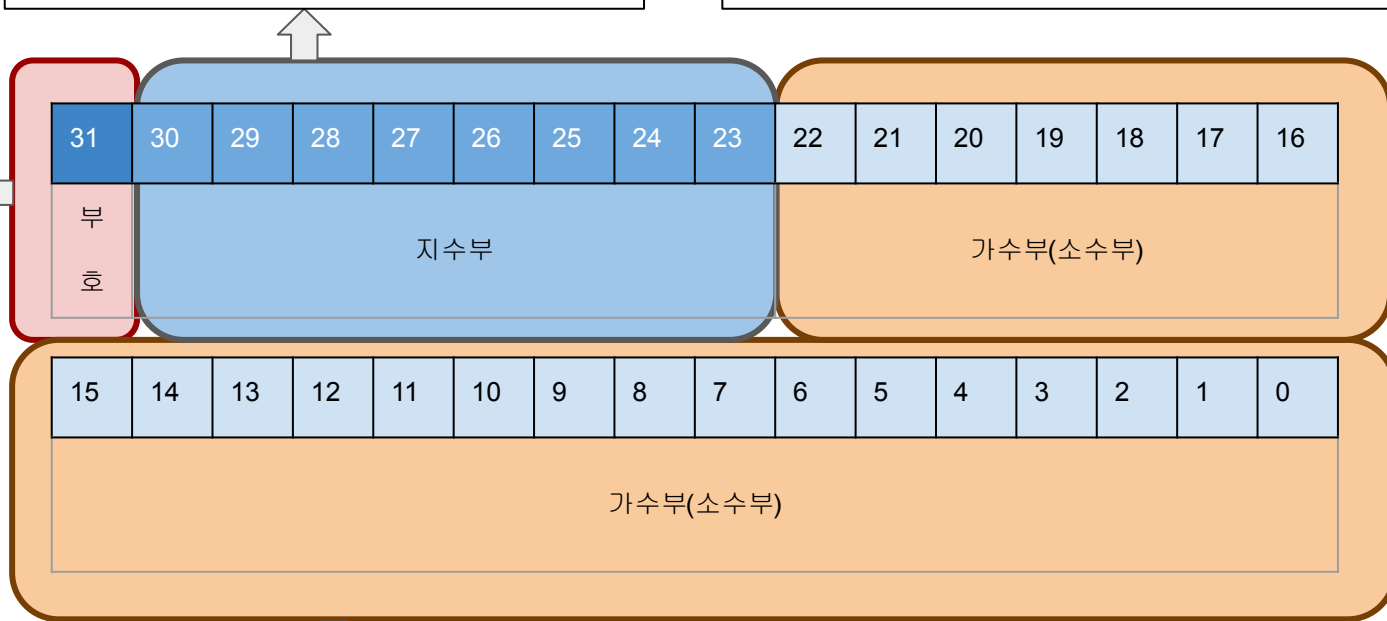
$$2^{128} = 3.4028235 \times 10^{38}$$
$$2^{-127} = 1.1754944 \times 10^{-38}$$



소수부 : 23비트 0 ... 8,388,608
표현범위 소수점이하 6번째 자리까지

지수부 : 8 비트 0 ... 127 ... 256
표현범위 -126 ~ 128 (Bias 127)

$$2^{128} = 3.4028235 \times 10^{38}$$
$$2^{-127} = 1.1754944 \times 10^{-38}$$



소수부 : 23비트 0 ... 8,388,608
표현범위 소수점이하 6번째 자리까지

String

```
void loop() {  
    String sample = "Remember 0416 ";  
    Serial.println(sample);  
    delay(1000);  
}
```




아두이노용 C 언어

- 2의 보수

2의 보수

10 에 -10 을 더하면 0

10 에 X 를 더했더니 결과가 0

X 는 10을 0 으로 만드는 수

십진수 10은 이진수로 1010

십진수 10은 8비트 이진수로

0000 1010

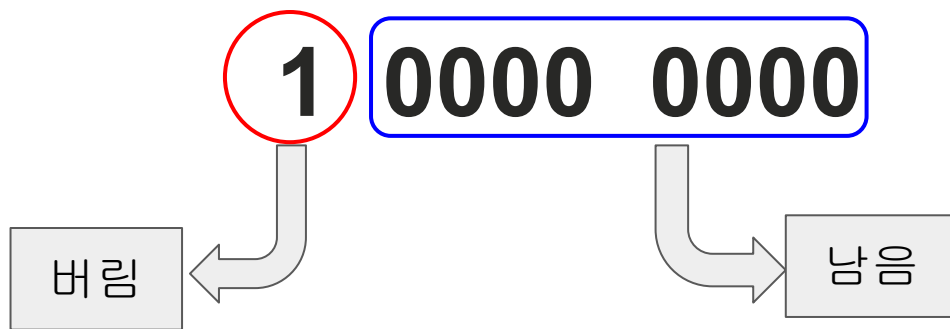
0000 1010

0000 1010 + X 가 0 이되게 하는 X 는?

TIP : 1111 1111 + 1

TIP : 1111 1111 + 1
1 0000 0000

TIP : 1111 1111 + 1



0000 1010 + X 가 0 이되게 하는 X 는?

0000 1010

+ ???? ????
=====

1111 1111

0000 1010

+ 1111 0101

=====

1111 1111

$$\begin{array}{r}
 0000 \ 1010 \\
 + \ 1111 \ 0101 \\
 \hline
 \end{array}$$

1을 0으로
0을 1로
바꿈

=====

1111 1111

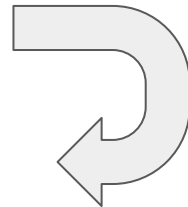
$$\begin{array}{r} 0000 \ 1010 \\ + \ 1111 \ 0101 \\ \hline \end{array}$$

1을 0으로
0을 1로
바꿈

=====

1111 1111

0000 1010
+ 1111 0101



1을 0으로
0을 1로
바꿈

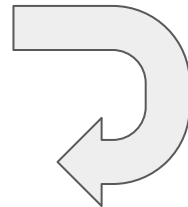
=====

1111 1111
+ 1

=====

1 0000 0000

0000 1010
+ 1111 0101



1을 0으로
0을 1로
바꿈

=====

1111 1111
+ 1

1



1을 더함

=====

1 0000 0000

$$\begin{array}{r}
 0000\ 1010 \\
 +\ 1111\ 0101 \\
 \hline\hline
 1111\ 1111 \\
 +\ 1 \\
 \hline\hline
 1\ 0000\ 0000
 \end{array}$$

1을 0으로
0을 1로
바꿈

1을 더함

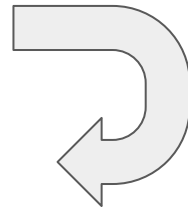
2의 보수 만들기

1. 10진수를 2진수로 변환

2. 2진수의 1과 0을 모두 바꿈

3. 결과에 1을 더함

0000 1010
+ 1111 0101



1을 0으로
0을 1로
바꿈

1111 1111
+ 1



1을 더함

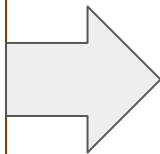
1 0000 0000

2의 보수 만들기

1. 10진수를 2진수로 변환
2. 2진수의 1과 0을 모두 바꿈
3. 결과에 1을 더함

2의 보수 만들기

1. 10진수를 2진수로 변환
2. 2진수의 1과 0을 모두 바꿈
3. 결과에 1을 더함



10 의 2의 보수

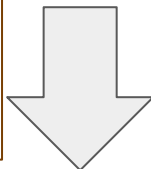
1. (십진수)10
= (이진수)0000 1010
2. 1111 0101
3. 1111 0110

2의 보수 만들기

1. 10진수를 2진수로 변환
2. 2진수의 1과 0을 모두 바꿈
3. 결과에 1을 더함

10 의 2의 보수

1. (십진수)10
= (이진수)0000 1010
2. 1111 0101
3. 1111 0110



```
0000 1010
+ 1111 0110
=====
1 0000 0000
```




아두이노용 C 언어

- 아스키코드
ASCII TABLE

ASCII TABLE

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

0x30 - '0'

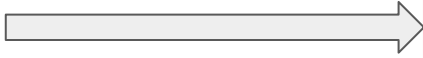
0x41 - 'A'

0x61 - 'a'

0x30 - '0'

0x41 - 'A'

0x61 - 'a'



```
void loop() {  
    char a = 0x30;  
    for (int i =0; i<10; i++)  
    {  
        Serial.println(a);  
        a++;  
    }  
    delay(1000);  
}
```

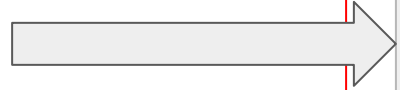
0x30 - '0'

0x41 - 'A'

0x61 - 'a'



```
void loop() {  
    char a = 0x30;  
    for (int i =0; i<10; i++)  
    {  
        Serial.println(a);  
        a++;  
    }  
    delay(1000);  
}
```



0
1
2
3
4
5
6
7
8
9

0x30 - '0'

0x41 - 'A'

0x61 - 'a'

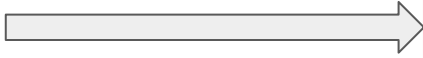


```
void loop() {  
    char a = '0';  
    for (int i =0; i<10; i++)  
    {  
        Serial.println(a);  
        a++;  
    }  
    delay(1000);  
}
```

0x30 - '0'

0x41 - 'A'

0x61 - 'a'



```
void loop() {
```

```
    char a = '0';
```

```
    for (int i =0; i<10; i++)
```

```
    {
```

```
        Serial.println(a);
```

```
        a++;
```

```
    }
```

```
    delay(1000);
```

```
}
```



0
1
2
3
4
5
6
7
8
9

0x30 - '0'

0x41 - 'A'

0x61 - 'a'

0x30 - '0'

0x41 - 'A'

0x61 - 'a'



```
void loop() {  
    char a = 'A'; // char a = 0x41  
    for (int i =0; i<10; i++)  
    {  
        Serial.println(a);  
        a++;  
    }  
    delay(1000);  
}
```

0x30 - '0'

0x41 - 'A'

0x61 - 'a'



```
void loop() {  
    char a = 'A'; // char a = 0x41  
    for (int i =0; i<10; i++)  
    {  
        Serial.println(a);  
        a++;  
    }  
    delay(1000);  
}
```



A
B
C
D
E
F
G
H
I
J

```
char a = '0';
```

```
char a = 0;
```


ASCII TABLE

```
char a = 0;
```

```
char a = '0';
```

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

ASCII TABLE

`char a = 0;`

`char a = '0';`

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

0x30 - '0'

0x41 - 'A'

0x61 - 'a'



```
void loop() {
```

```
  char a = 0;
```

```
  for (int i =0; i<10; i++)
```

```
  {
```

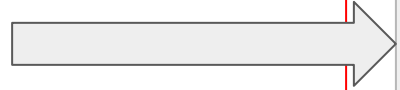
```
    Serial.println(a);
```

```
    a++;
```

```
  }
```

```
  delay(1000);
```

```
}
```



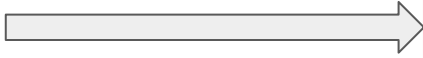
??

?
?
?
?
?
?
?
?
?
?
?

0x30 - '0'

0x41 - 'A'

0x61 - 'a'



```
void loop() {  
    char a = '0';  
    for (int i =0; i<10; i++)  
    {  
        Serial.println(a);  
        a++;  
    }  
    delay(1000);  
}
```



0
1
2
3
4
5
6
7
8
9



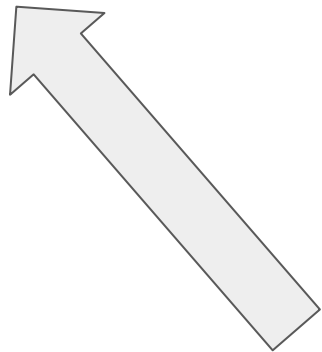
아두이노용 C 언어

– 연산/비교/논리



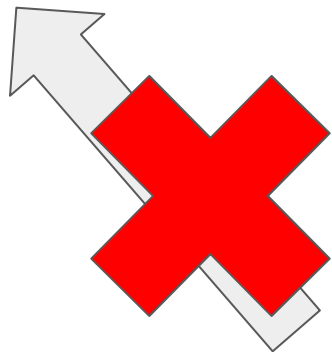
$$a = b$$

a는 b와 같다.



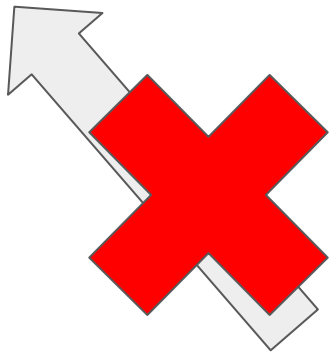
$$a = b$$

a는 b와 같다.

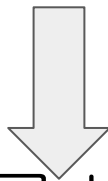


$$a = b$$

a는 b와 같다.

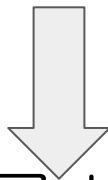


$$a = b$$



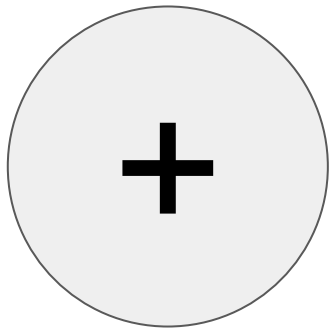
a에 b를 넣어라

$$a = b$$



a에 b를 넣어라

=

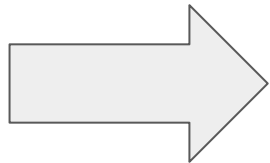


+

-

/

%



+ **=**

- **=**

***** **=**

/ **=**

% **=**

`x = x + 10;`

`x += 10;`

X++

$x = x + 1$

$x += 1$

$x++$

기본 수학연산 (+, -, *, /, %)

```
result = result + a;  
result = result - a;  
result = result * a;  
result = result / a;  
result = result % a;
```

```
result += a;  
result -= a;  
result *= a;  
result /= a;  
result %= a;
```

변수 **result** 에 **a** 를 더해서 저장합니다
변수 **result** 에 **a** 를 빼서 저장합니다
변수 **result** 에 **a** 를 곱해서 저장합니다
변수 **result** 를 **a** 로 나누고 저장합니다
변수 **result** 를 **a** 로 나눈 나머지를 저장합니다

田 邑

x == y

$x == y$

$x > y$

$x >= y$

$x < y$

$x <= y$

$x != y$

논리

& & : 논리곱

| | : 논리합

! : 부정

&& 논리곱

A	B	A && B
T	T	T
T	F	F
F	T	F
F	F	F

|| 논리곱

A	B	A B
T	T	T
T	F	T
F	T	T
F	F	F

! 부정

A	!A
T	F
F	T

&& 논리곱

A	B	A & B
T	T	T
T	F	F
F	T	F
F	F	F

|| 논리합

A	B	A B
T	T	T
T	F	T
F	T	T
F	F	F

! 부정

A	!A
T	F
F	T

(x > 0 && x < 5)

(x > 10 || x < 5)

(! x)

if ((x > 0) && (x < 10))

비트

~ NOT

& AND

| OR

^ XOR

<< Shift Left

>> Shift Right

~ **NOT**

& AND

| OR

^ XOR

<< Shift Left

>> Shift Right

~

NOT

~ NOT

```
byte A = 10;
```

```
// A = 0000 1010
```

```
// ~A => 1111 0101
```

~ NOT

& AND

| OR

^ XOR

<< Shift Left

>> Shift Right

~ NOT

& AND

| OR

^ XOR

<< Shift Left

>> Shift Right

&

AND

& AND

A = 10, B = 8 이라면

A & B =>

& AND

A = 10, B = 8 이라면

A & B => 0000 1010

& 0000 1000

=====

0000 1000

& AND

A = 10, B = 8 이라면

A & B => 0000 1010

& **0000 1000**

=====

0000 1000

~ NOT

& AND

| OR

^ XOR

<< Shift Left

>> Shift Right

~ NOT

& AND

| **OR**

^ XOR

<< Shift Left

>> Shift Right

|

OR

| OR

A = 10, B = 7 이라면

A | B =>

| OR

A = 10, B = 7 이라면

A | B => 0000 1010

& 0000 0111

=====

0000 1111

| OR

A = 10, B = 1 이라면

A | B => 0000 1010

 | 0000 0001

=====

0000 1011

| OR

A = 10, B = 1 이라면

A | B => 0000 101**0**

 | 0000 000**1**

=====

0000 101**1**

~ NOT

& AND

| OR

^ XOR

<< Shift Left

>> Shift Right

~ NOT

& AND

| OR

^ XOR

<< Shift Left

>> Shift Right

^

XOR

^

XOR

A = 10, B = 3 이라면

A ^ B =>

^

XOR

A = 10, B = 3 이라면

A ^ B => 0000 1010

^ 0000 0011

=====

0000 1001

^

XOR

A = 10, B = 3 이라면

A	^	B	=>	0	0	0	0	1	0	1	0
		^		0	0	0	0	0	0	1	1
								=====			
				0	0	0	0	1	0	0	1

~ NOT

& AND

| OR

^ XOR

<< Shift Left

>> Shift Right

~ NOT

& AND

| OR

^ XOR

<< **Shift Left**

>> **Shift Right**

<< Shift Left

>> Shift Right

<< Shift Left

>> Shift Right

A = 10, A<<2 라면?

A : 0000 1010

<< Shift Left

>> Shift Right

$A = 10$, $A \ll 2$ 라면?

A : 0000 1010 (왼쪽 2칸 이동)

$A \ll 2$: 0010 1000



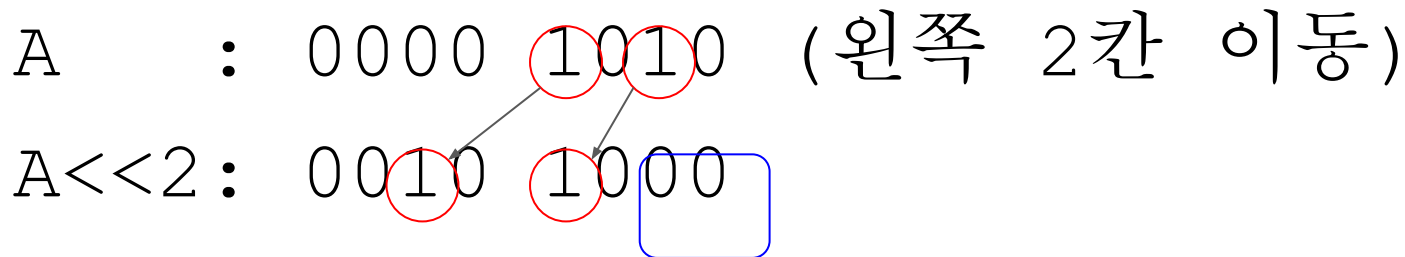
<< Shift Left

>> Shift Right

$A = 10$, $A \ll 2$ 라면?

A : 0000 1010 (왼쪽 2칸 이동)

$A \ll 2$: 0010 1000





아두이노용 C 언어

- 상수

#define

```
#define SIZE 1024
```

```
const int SIZE 1024;
```

상수 VS 변수

<pre>int LED = 13; ... digitalWrite(LED, HIGH); ...</pre>	<pre>#define LED 13 ... digitalWrite(LED, HIGH); ...</pre>	<pre>const int LED = 13; ... digitalWrite(LED, HIGH); ...</pre>
변수	상수 - 추천	
	변수 메모리할당 없음	

<pre>int LED = 13; ... digitalWrite(LED, HIGH); ...</pre>	<pre>#define LED 13 ... digitalWrite(LED, HIGH); ...</pre>	<pre>const int LED = 13; ... digitalWrite(LED, HIGH); ...</pre>
변수 - 비추천	상수	
변수 메모리할당		

<pre>int LED = 13; ... digitalWrite(LED, HIGH); ...</pre>	<pre>#define LED 13 ... digitalWrite(LED, HIGH); ...</pre>	<pre>const int LED = 13; ... digitalWrite(LED, HIGH); ...</pre>
변수 - 비추천	상수	
변수 메모리할당		

<pre>int LED = 13; ... digitalWrite(LED, HIGH); ...</pre>	<pre>#define LED 13 ... digitalWrite(LED, HIGH); ...</pre>	<pre>const int LED = 13; ... digitalWrite(LED, HIGH); ...</pre>
변수 - 비추천	상수 - 추천	
변수 메모리할당	변수 메모리할당 없음	



아두이노용 C 언어

– 조건문



참과 거짓

1은 참

0은 거짓

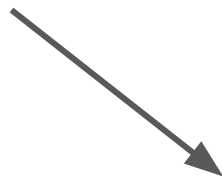
```
if ( a == 12 )  
{  
    할 일 ;  
}
```

```
if ( a == 12 )  
{  
    할 일 ;  
}
```



참은 1

```
if ( a == 12 )  
{  
    할 일 ;  
}
```



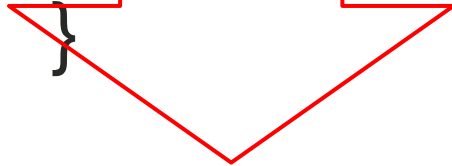
거짓은 0

```
if ( 1 )
```


```
{
```

```
    할 일 ;
```

```
}
```



```
if ( 0 )  
{  
    할 일 ;  
}
```



건너뛴

HIGH ≡ 1

LOW ≡ 0

```
digitalWrite(13, 1);
```

```
digitalWrite(13, 0);
```



```
digitalWrite(13, HIGH);
```

```
digitalWrite(13, LOW);
```

if

```
if ( 조건 ) // 조건부분
```

```
{
```

```
    // 조건이 참이면(1이면) 이 부분을 실행합니다.
```

```
    // 조건이 거짓이면 이 부분을 건너뜁니다.
```

```
}
```

```
int condition =0;      // 전역변수를 만들고 값을 0으로 만든다.
void setup(){
    Serial.begin(9600);
}

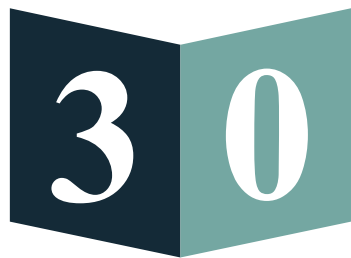
void loop(){
    condition ++;  // 전역변수의 값을 1 증가
    if ((condition % 5)==0) {
        Serial.print("condition = ");
        Serial.println(condition);    // 값을 화면에 프린트
    }
}
```

if , else

```
if ( 조건 )  
{  
    // 조건이 참이면 이 부분을 실행합니다.  
}  
else  
{  
    // 조건이 거짓이면 이 부분을 실행합니다.  
}
```

```
int condition =0;      // 전역변수를 만들고 값을 0으로 만든다.
void setup(){
    pinMode(13, OUTPUT);
}

void loop(){
    condition ++;  // 전역변수의 값을 1 증가
    if ((condition % 5)==0) {
        digitalWrite(13,HIGH); delay(100);
    }
    else {
        digitalWrite(13,LOW);  delay(100);
    }
}
```

아두이노용 C 언어

– for 반복문

for

```
for ( 초기화 ; 조건 ; 증감표현)
```

```
{
```

```
    반복할 일들 ;
```

```
}
```

```
for ( int i = 0 ; i<20 ; i++ )  
{  
    digitalWrite(13, HIGH);  
    delay(500);  
    digitalWrite(13, LOW);  
    delay(500);  
}
```

20 회 반복

```
for ( int i = 0 ; i<20 ; i++ )  
{  
    digitalWrite(13, HIGH);  
    delay(500);  
    digitalWrite(13, LOW);  
    delay(500);  
}
```

```
void setup() {  
    pinMode(13, OUTPUT);  
    for (int i =0; i<20 ; i++)  
    {  
        digitalWrite(13, HIGH);  
        delay(500);  
        digitalWrite(13, LOW);  
        delay(500);  
    }  
}  
  
void loop() {  
}
```




아두이노용 C 언어

– while 반복문

while

while (조건)

{

반복할 일들 ;

}

```
while (i<20) {  
    digitalWrite(13, HIGH); delay(500);  
    digitalWrite(13, LOW); delay(500);  
    i++;  
}
```

```
int i =0;
void setup() {
    pinMode(13, OUTPUT);
    while (i<20){
        digitalWrite(13, HIGH); delay(500);
        digitalWrite(13, LOW); delay(500);
        i++;
    }
}
void loop() {
}
```




아두이노용 C 언어

- 아두이노 함수

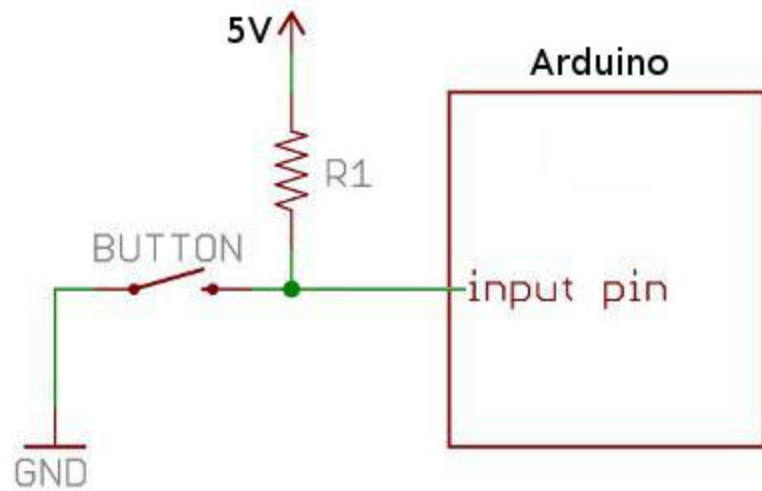
pinMode()

```
pinMode (13, OUTPUT) ;
```

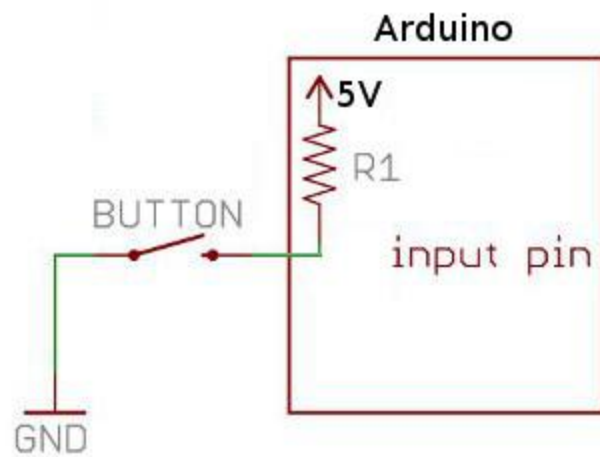
```
pinMode (12, INPUT) ; .
```

```
pinMode (11, INPUT_PULLUP) ;
```


External pull-up

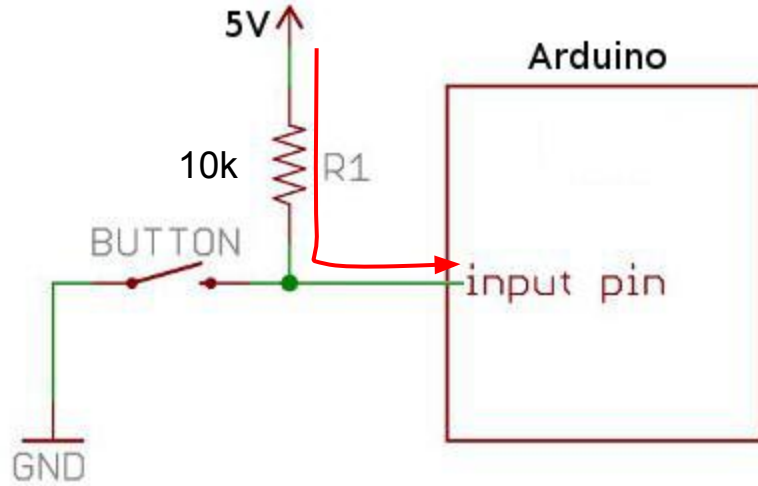


Internal pull-up

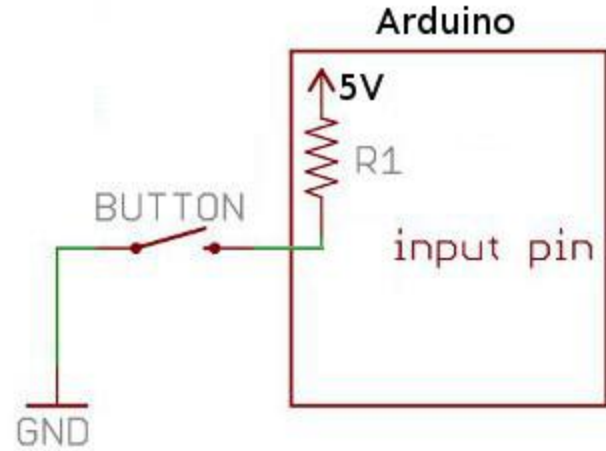


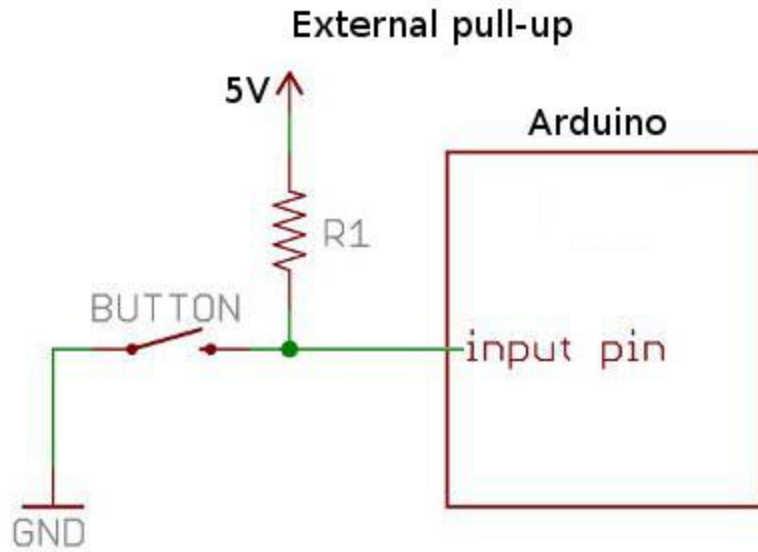
$$I = V / R = 5 / 10000 = 0.5\text{mA}$$

External pull-up

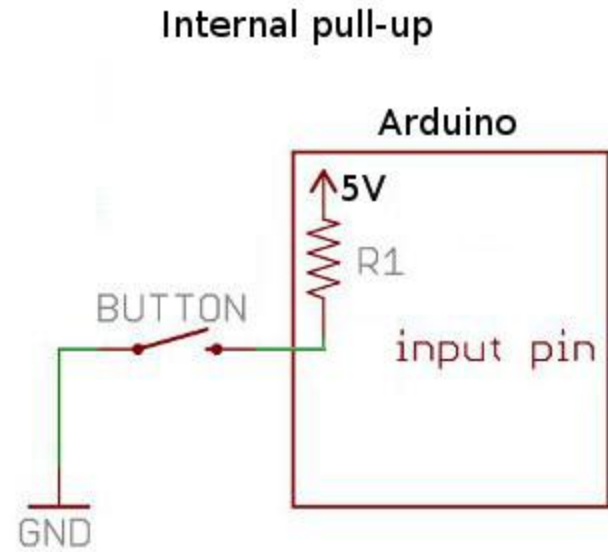


Internal pull-up

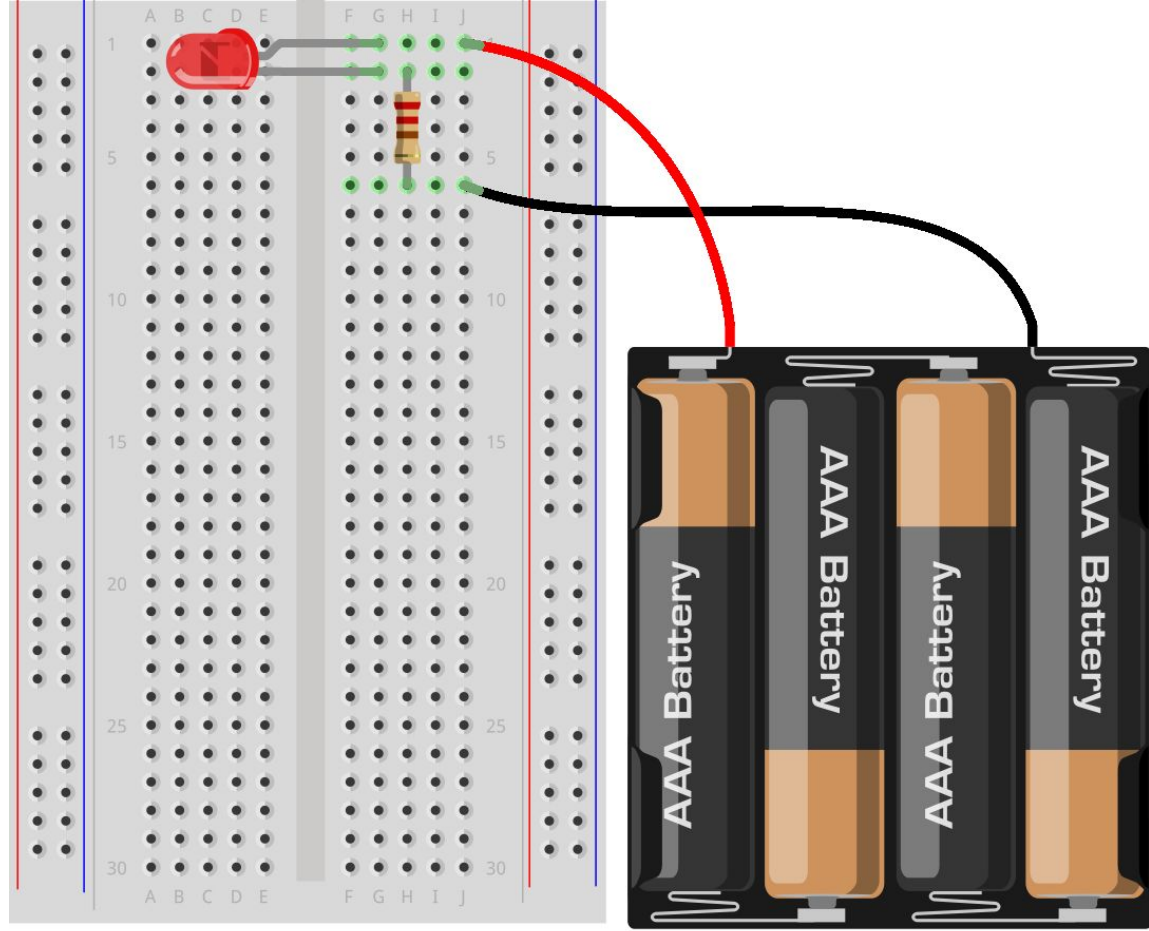


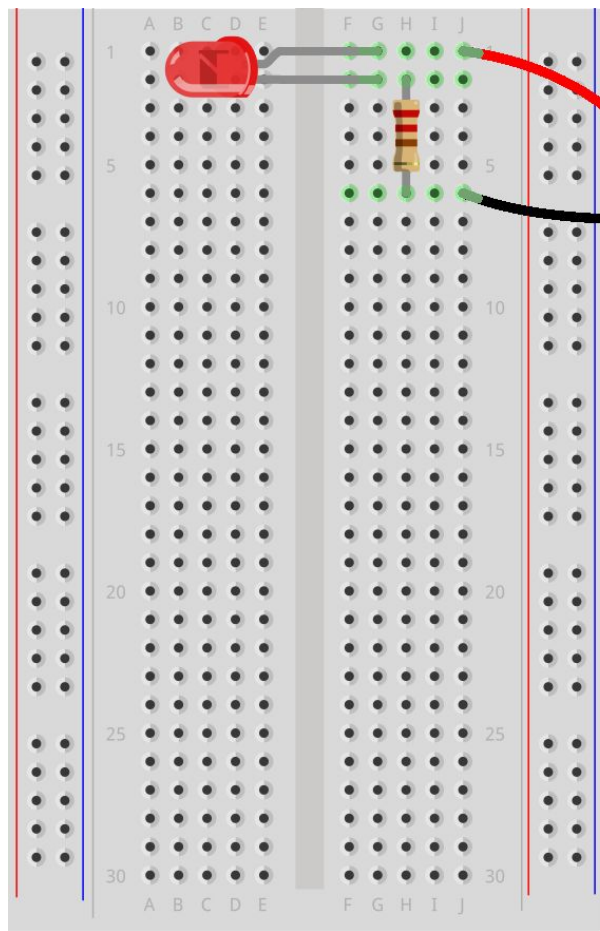
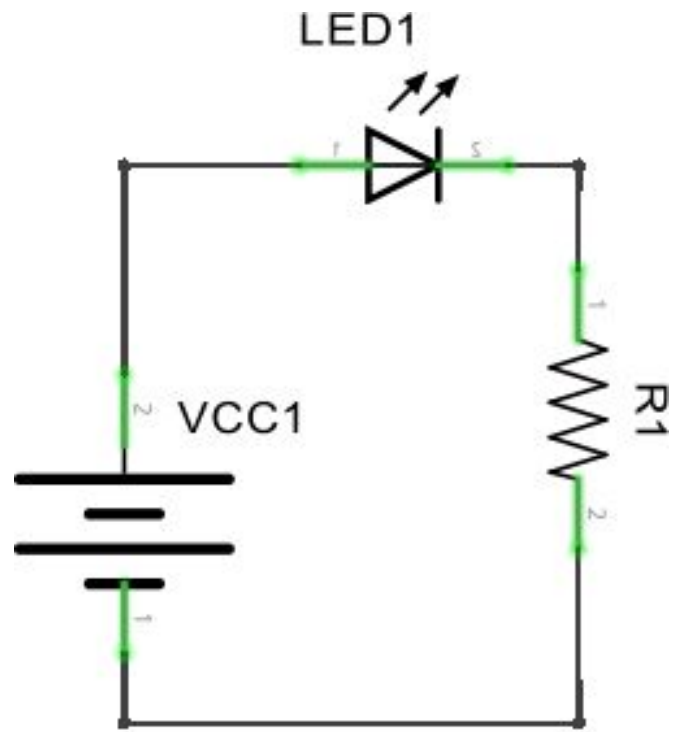


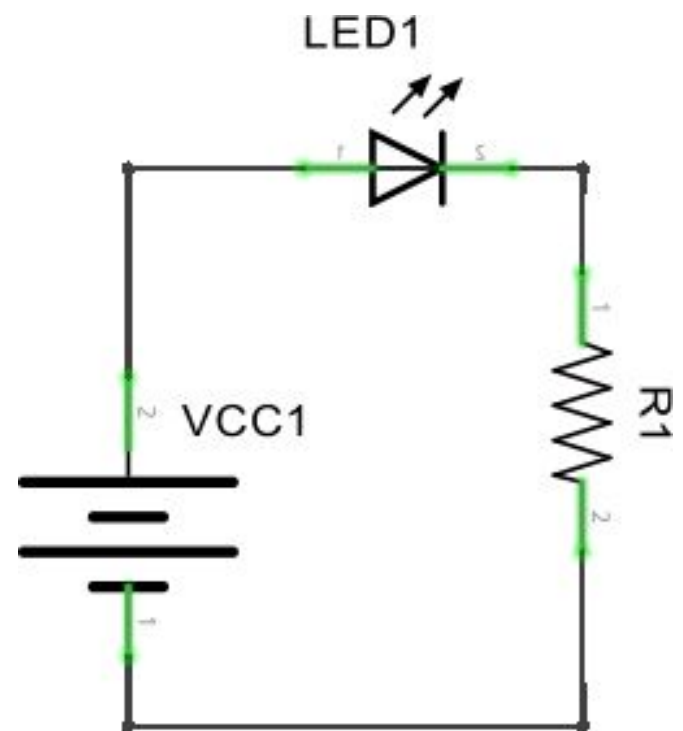
```
pinMode(pin, INPUT_PULLUP)
```

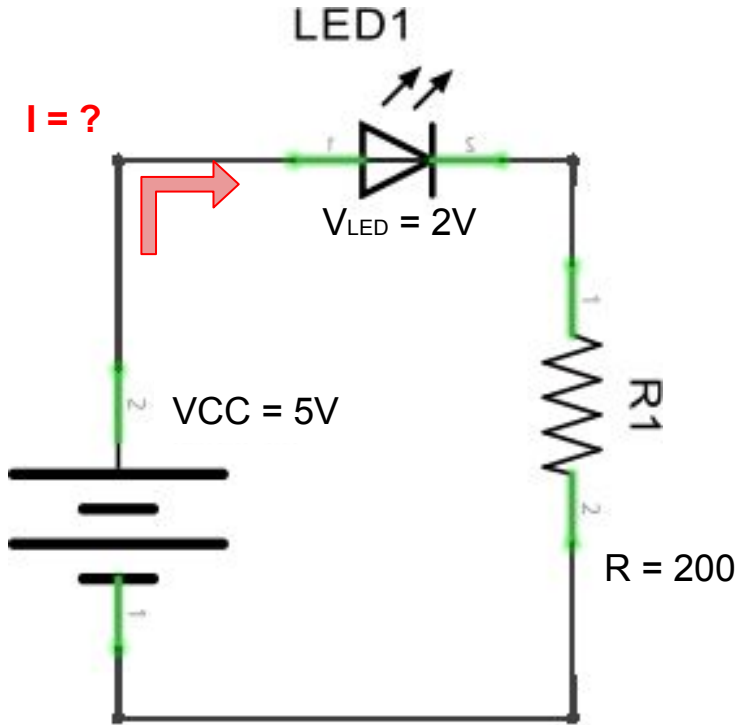


LED 전류



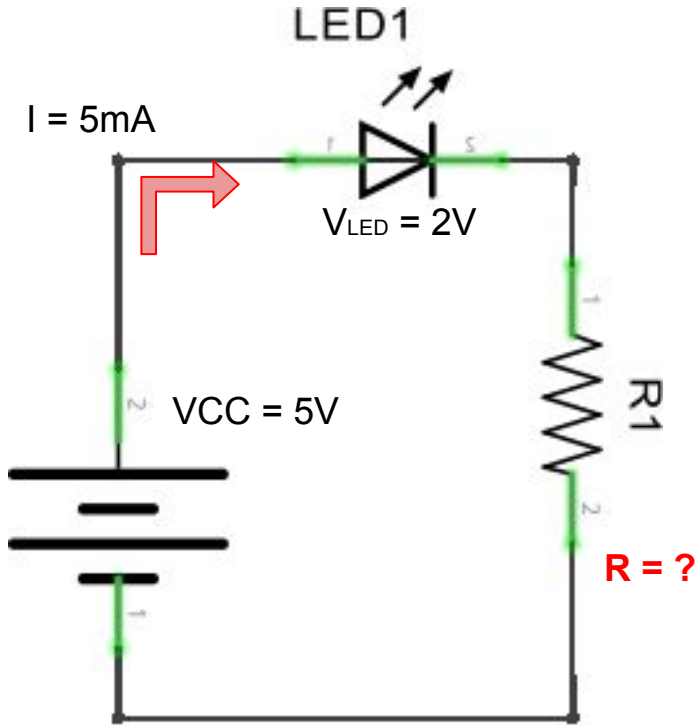






$$V = I \cdot R$$

$$\begin{aligned} I &= V / R \\ &= (5 - 2) / 200 \\ &= 0.015 \text{ A} \\ &= 15 \text{ mA} \end{aligned}$$



$$V = I \cdot R$$

$$\begin{aligned} R &= V / I \\ &= (5 - 2) / 0.005 \\ &= 600 \text{ ohm} \end{aligned}$$

digitalRead()

```
pinMode (13, INPUT) ;
```

```
digitalRead (13) ;
```

```
pinMode (13, INPUT) ;
```

```
digitalRead (13) ;
```



0 또는 1 을 반환

```
pinMode (13, INPUT);
```

```
value = digitalRead(13);
```



0 또는 1 을 반환

```
pinMode(13, INPUT);  
if(digitalRead(13))  
{  
    //  
}
```



0 또는 1 을 반환

digitalWrite()

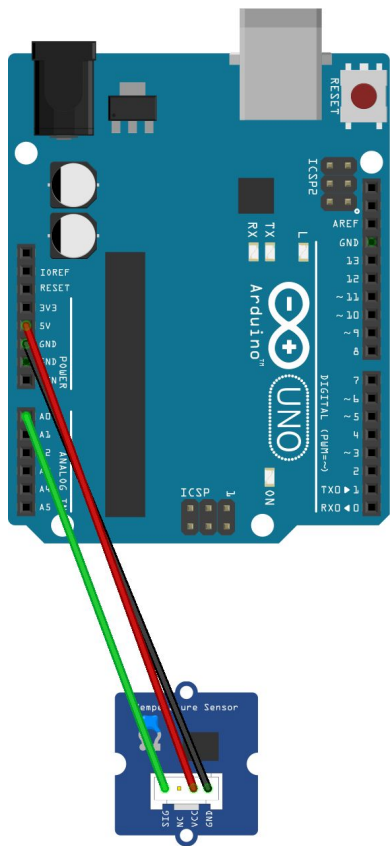
```
pinMode (13, OUTPUT) ;
```

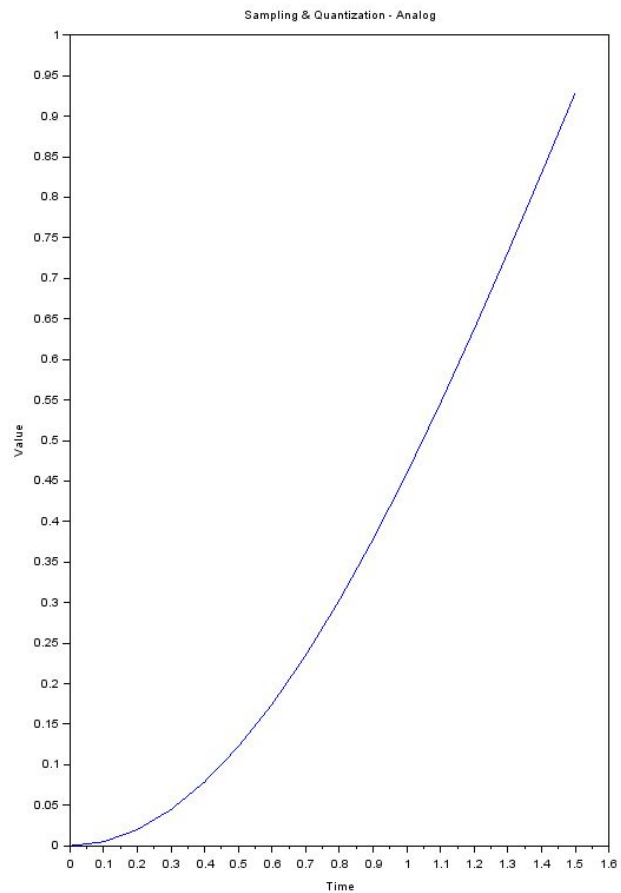
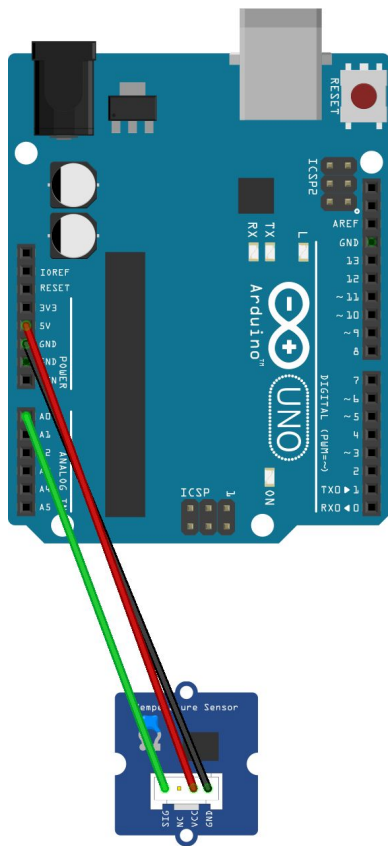
```
digitalWrite (13, HIGH) ;
```

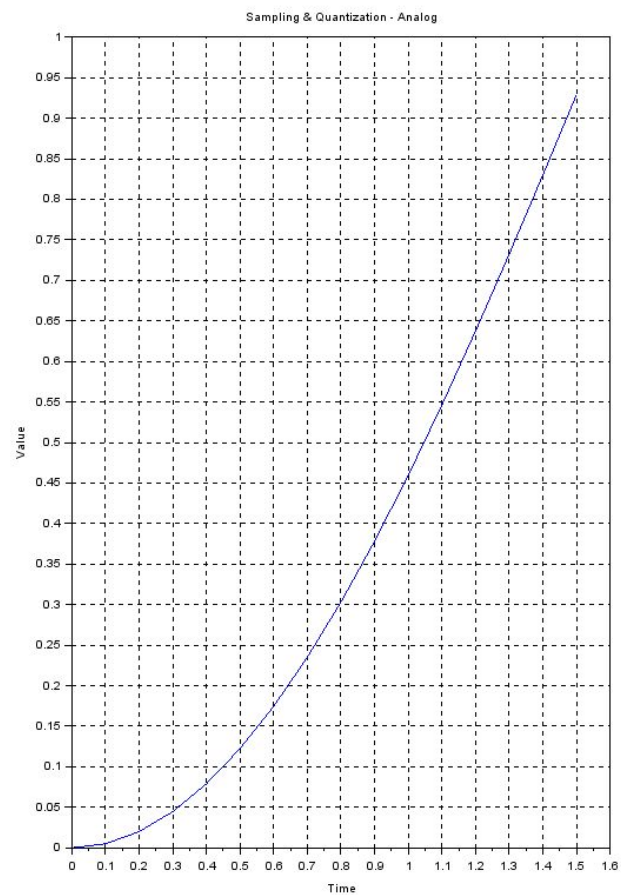
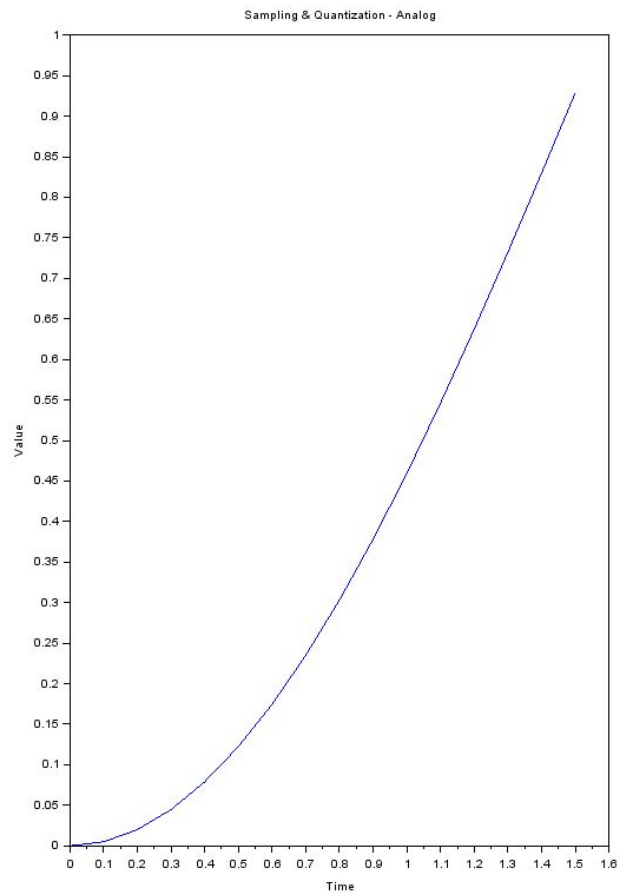
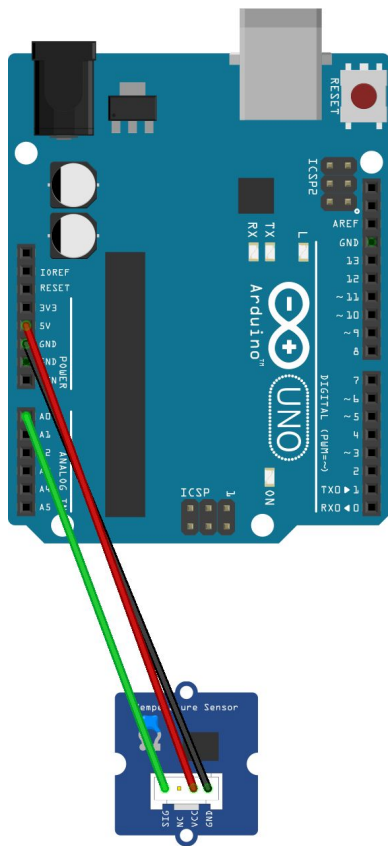
```
digitalWrite (13, LOW) ;
```


analogRead()

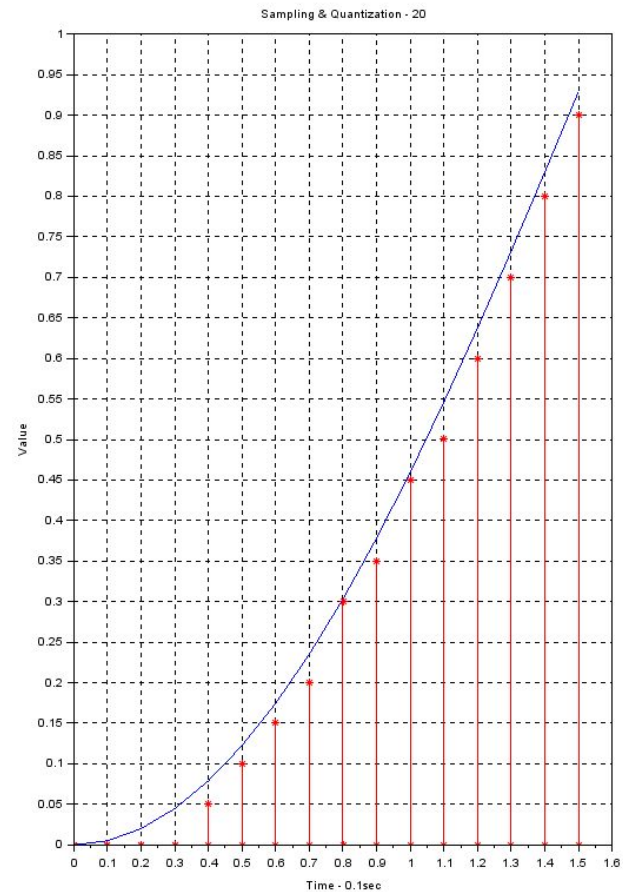
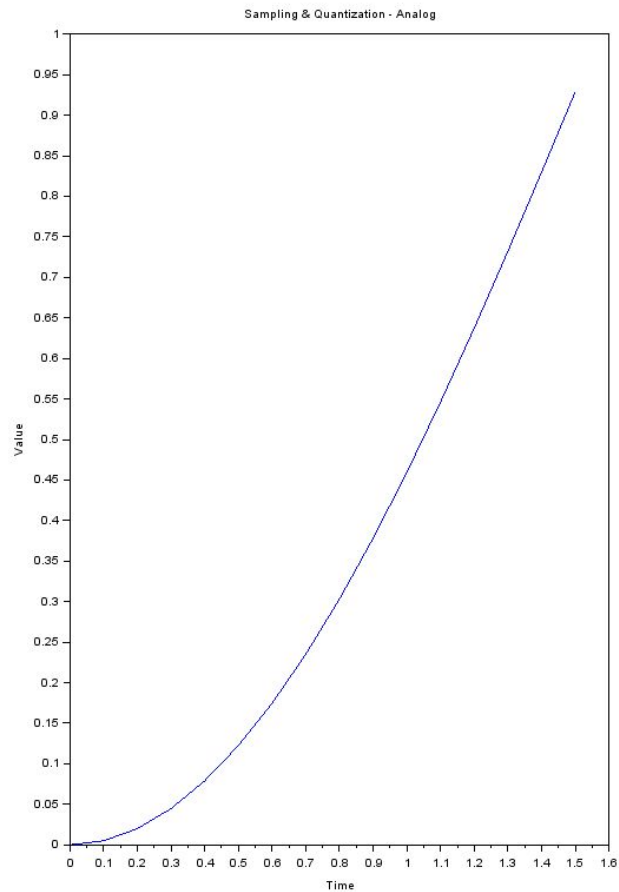
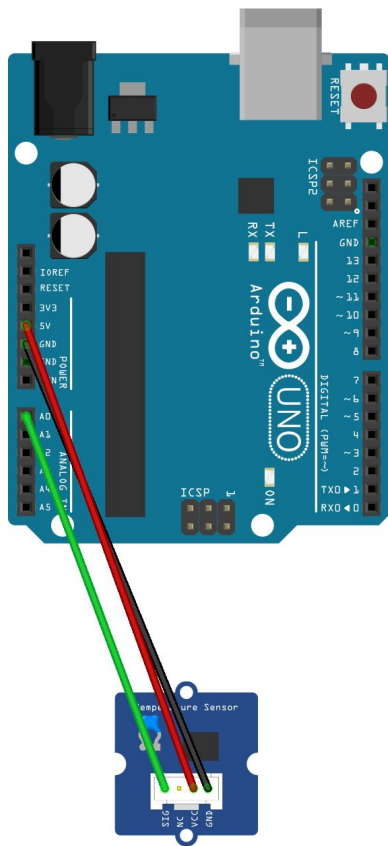
```
value = analogRead(A0);
```







샘플링 & 양자화



analogWrite()


```
analogWrite(3,127);
```

PWM 출력 가능한 핀
UNO (3,5,6,9,10)

`analogWrite(3,127);`

3번핀 출력

`analogWrite(3,127);`

3번핀 출력

`analogWrite(3,127);`

0 ~ 255 사이

delay()

시간 (mili-second)

`delay (1000) ;`

milis()

```
unsigned long value;  
value = millis();
```



```
unsigned long previousMillis = 0;
unsigned long currentMillis = millis();
if (currentMillis - previousMillis >= 1000) {
    previousMillis = currentMillis;
    if (ledState == LOW) { ledState = HIGH; }
    else { ledState = LOW; }
}
```

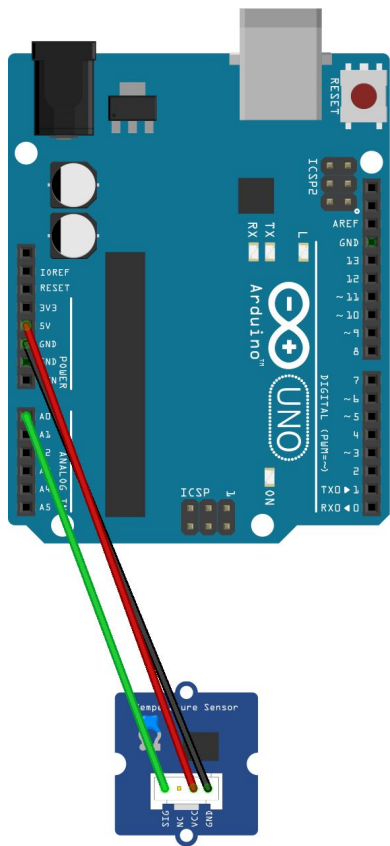


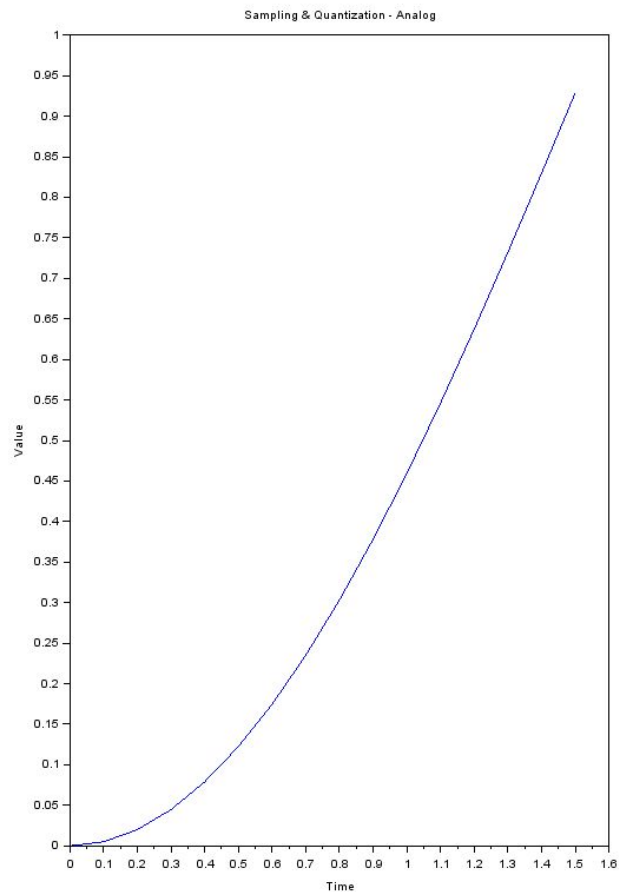
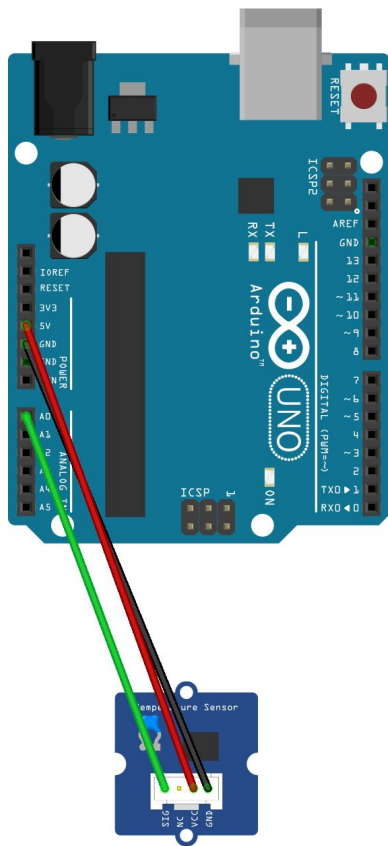

아두이노용 C 언어

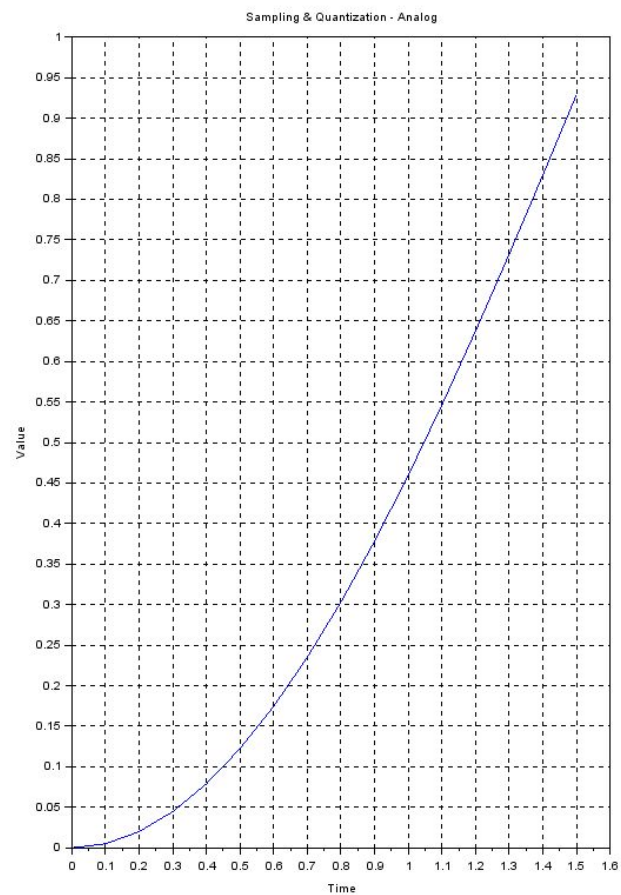
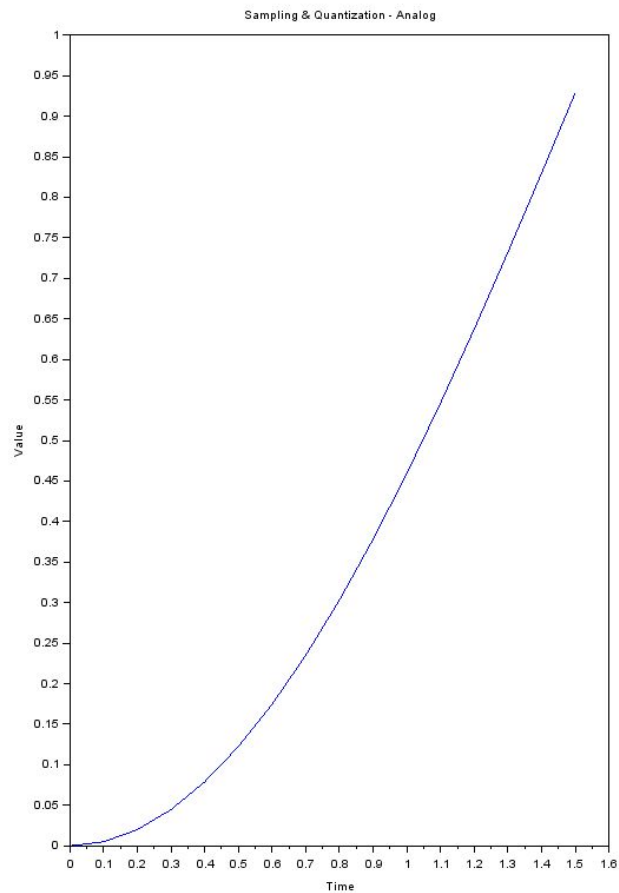
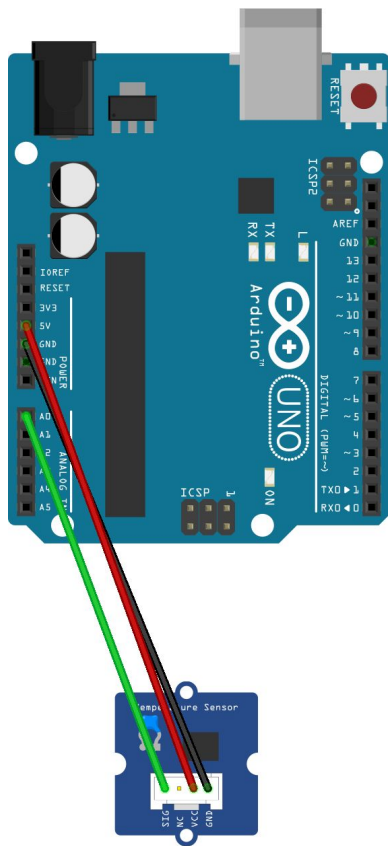
– Sampling, Quantization

analogRead()

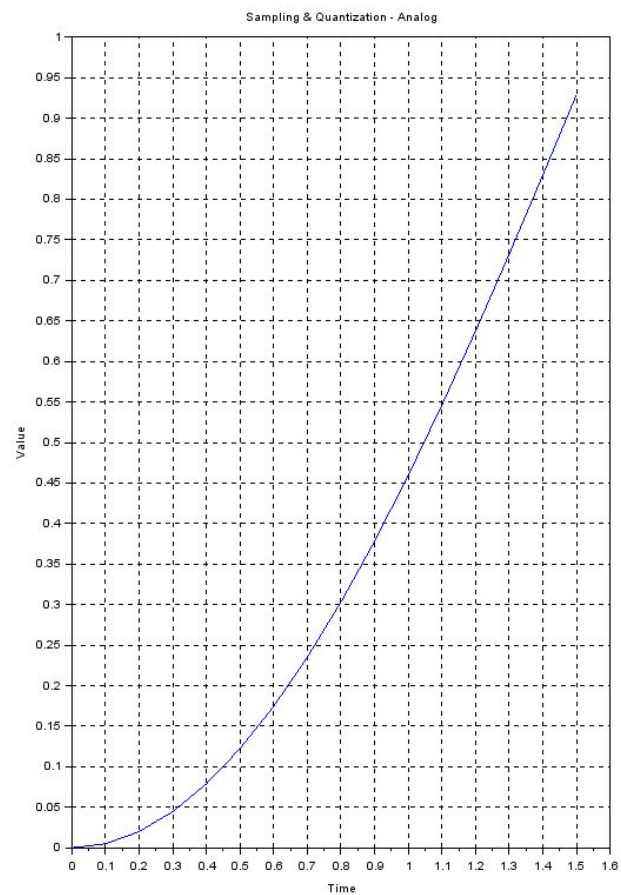
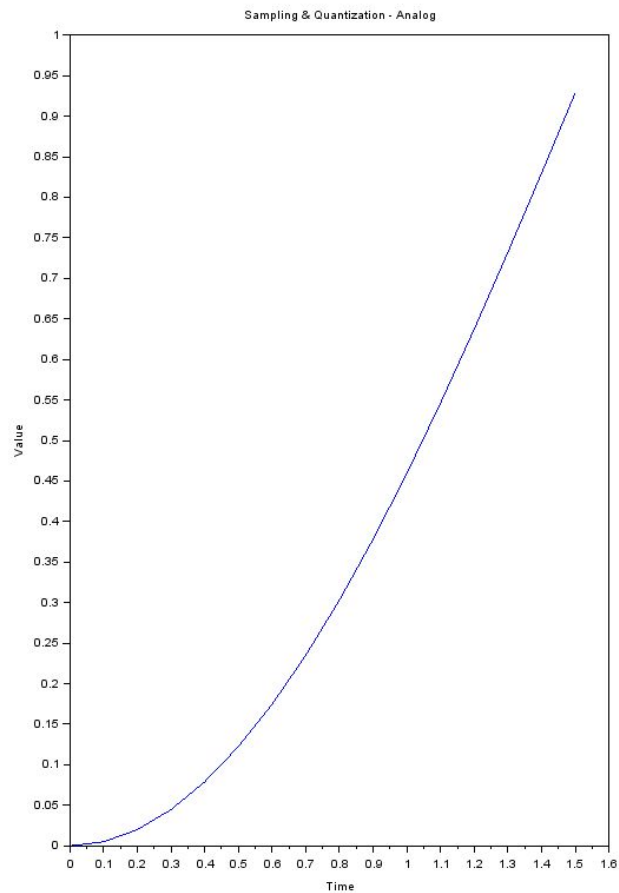
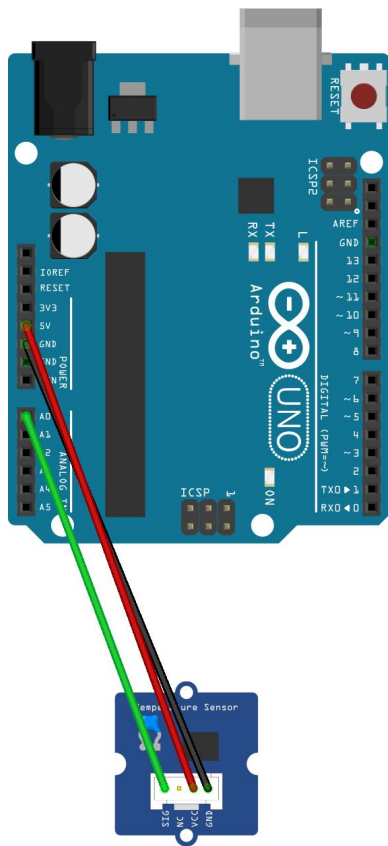
```
value = analogRead(A0);
```

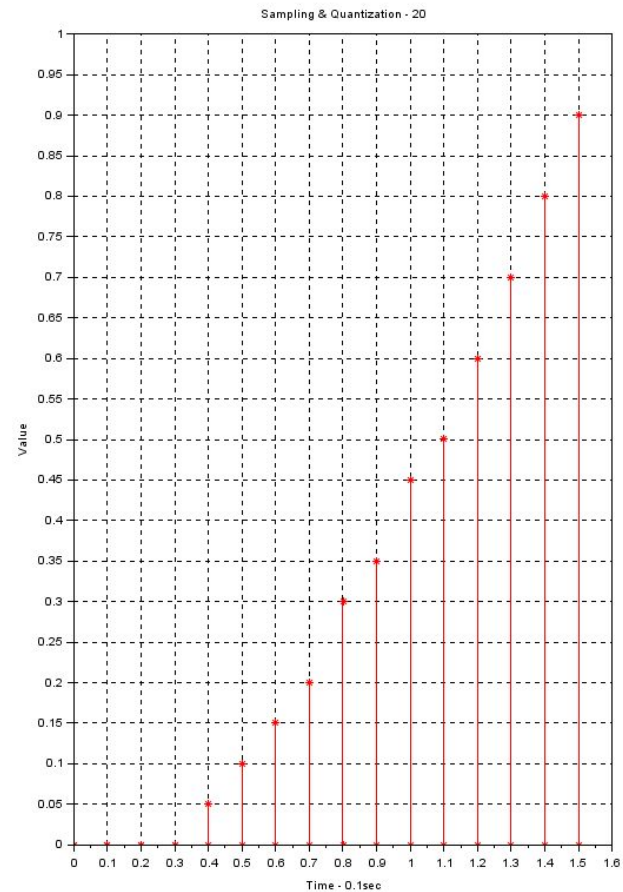
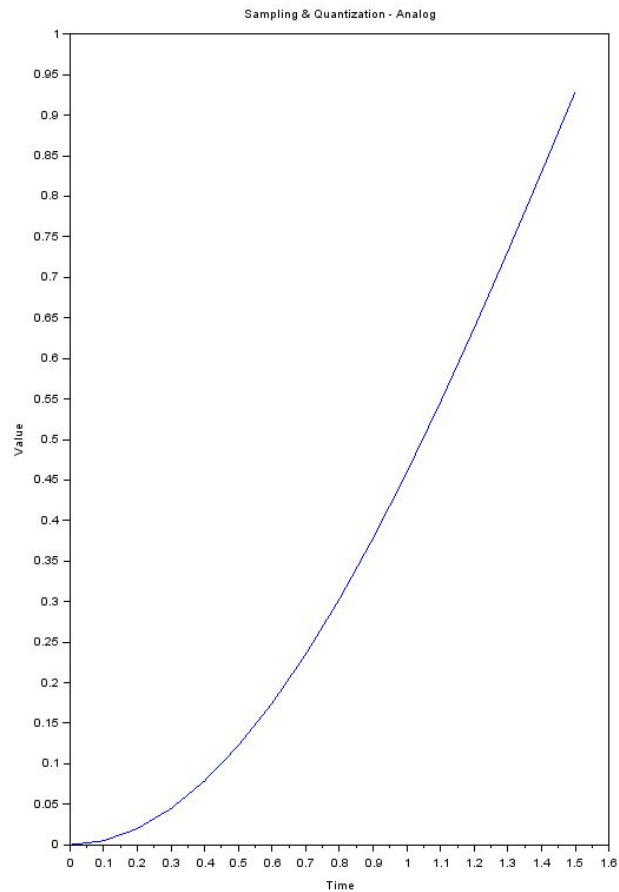
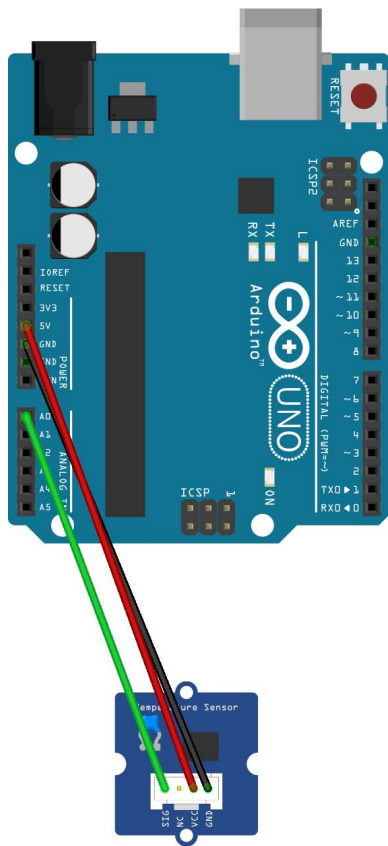


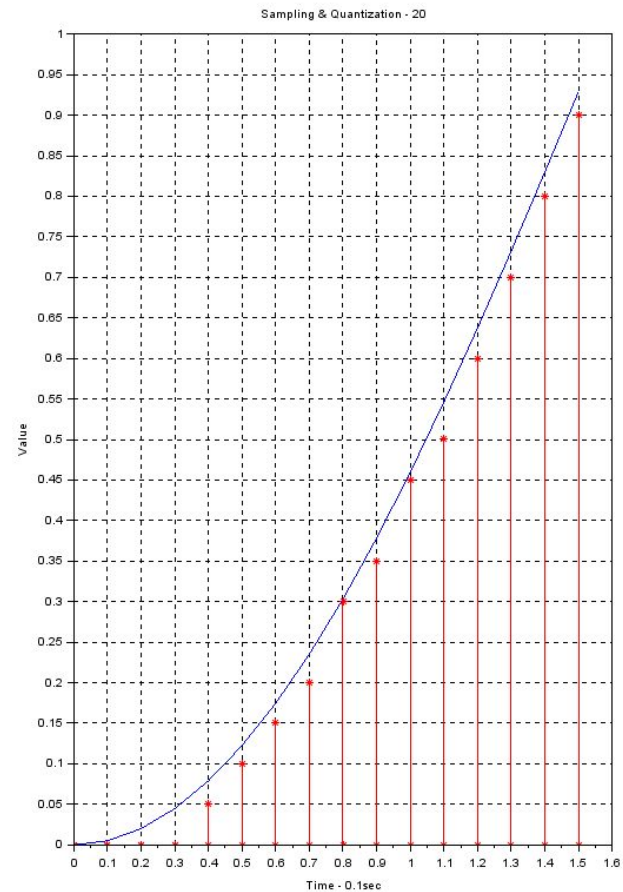
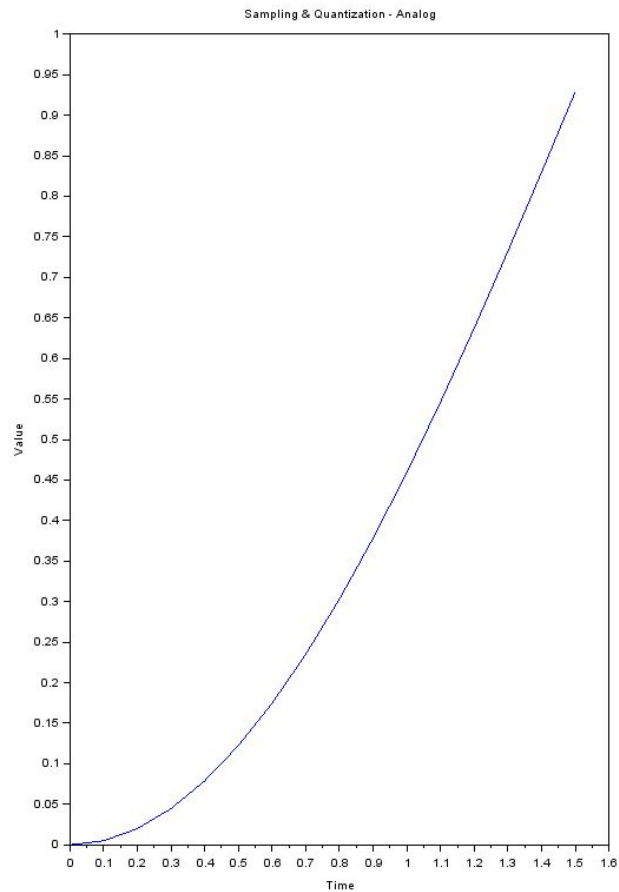
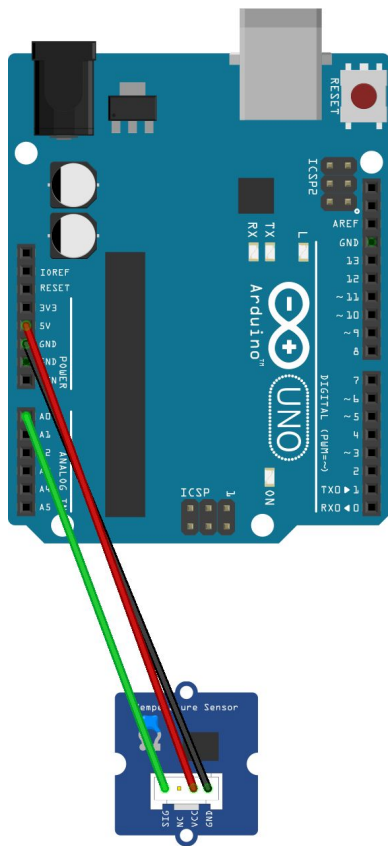


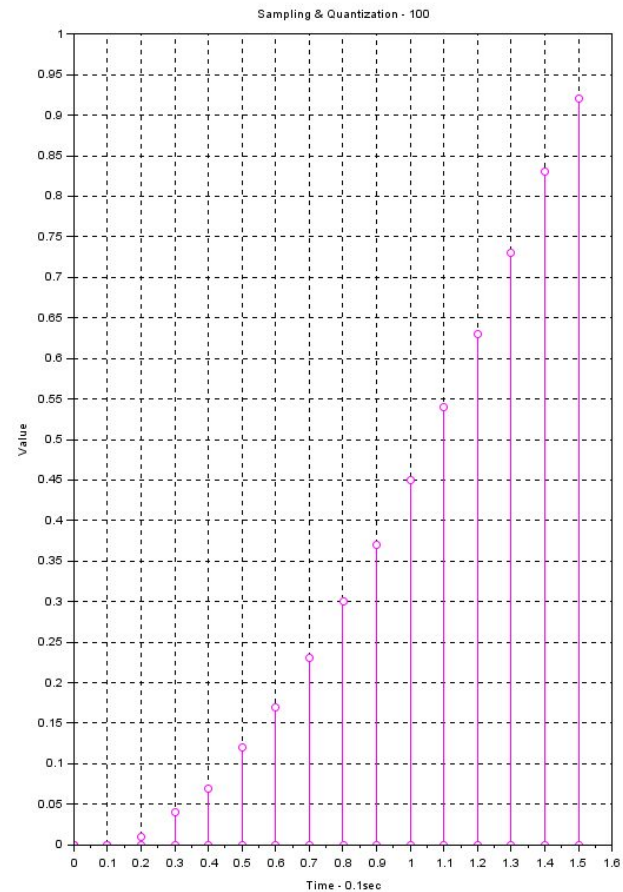
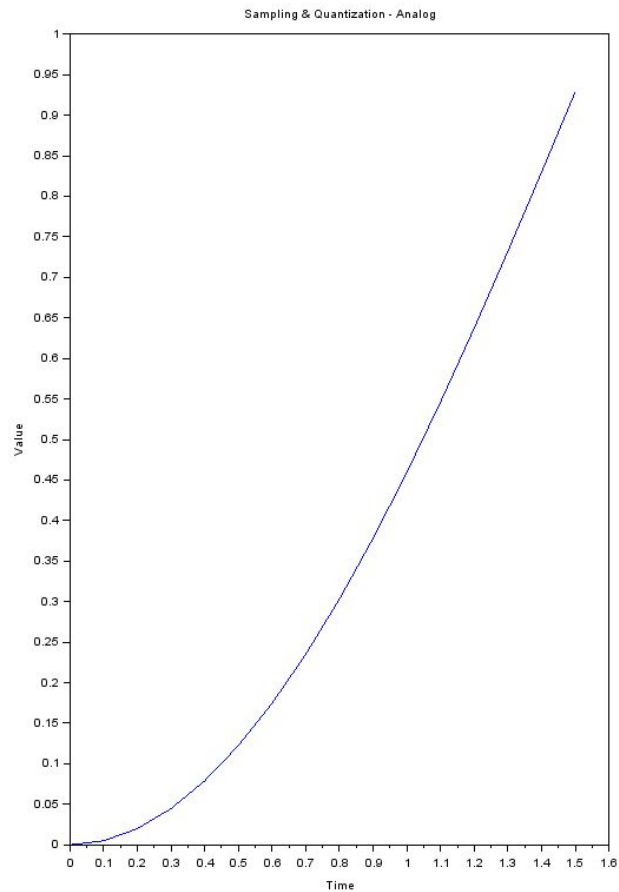
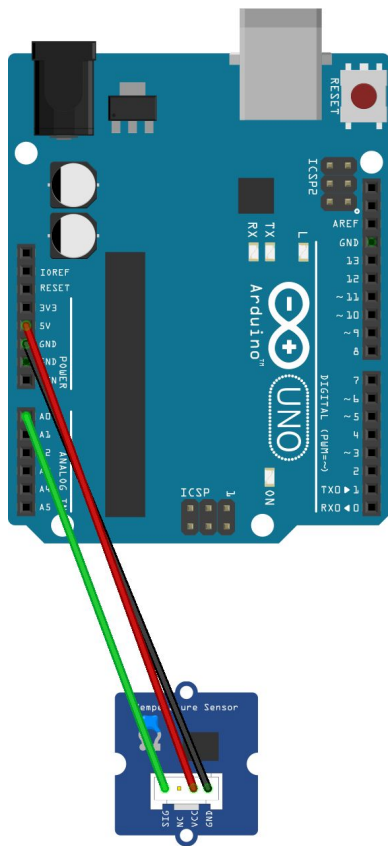


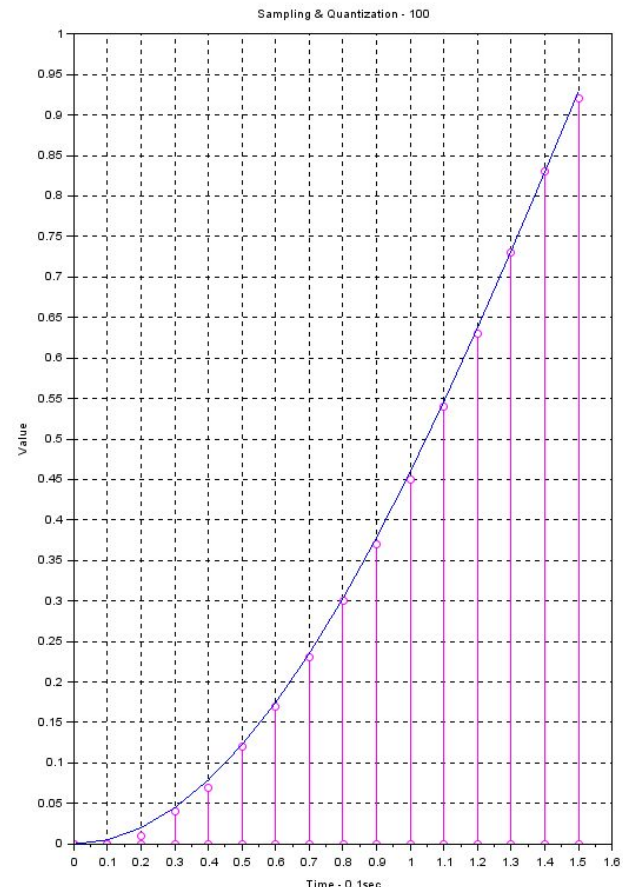
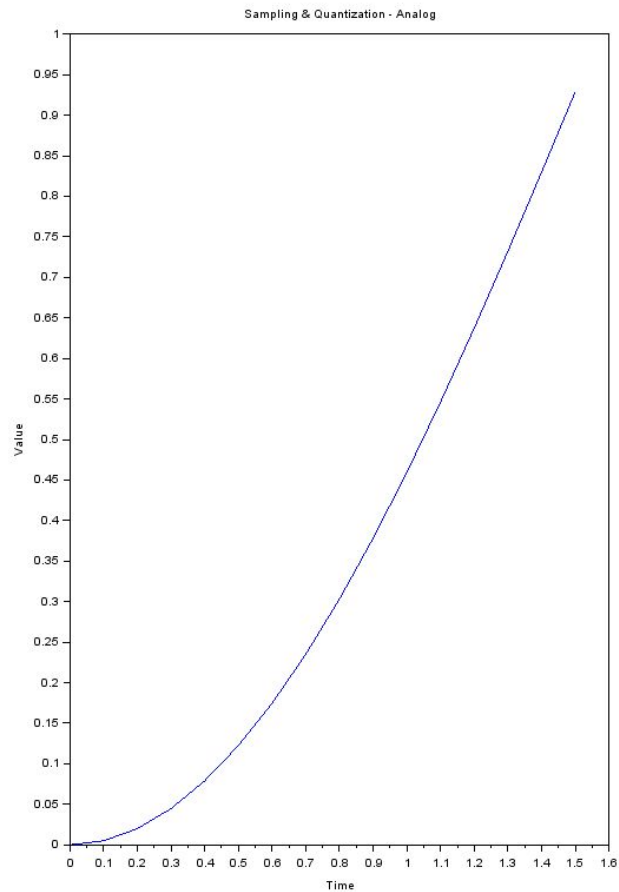
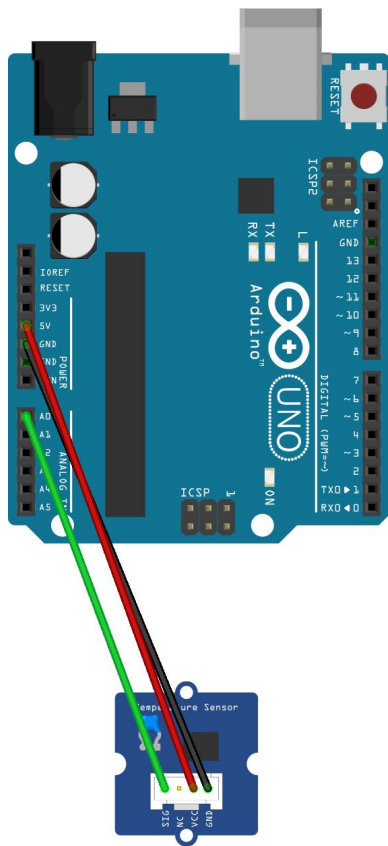
샘플링 & 양자화











샘플링 간격 : 시간 (T) or 주파수($f = 1/T$)

양자화(Quantization) 정도 : 비트

8 비트 => $0 \sim 2^8$, 256 단계

9 비트 => $0 \sim 2^9$, 512 단계

10 비트 => $0 \sim 2^{10}$, 1024 단계 (아두이노 UNO)

12 비트 => $0 \sim 2^{12}$, 4096 단계 (아두이노 DUE)

16 비트 => $0 \sim 2^{16}$, 65536 단계

Question :

온도에 민감한 고감도 필름을 보관하고 있으며 이 장소에 대한 온도를 측정하려고 한다. 사용하는 센서는 **-20** 도에서 **+130** 도까지 측정가능하고, 이때 **0V** 에서 **5V** 사이의 아날로그 전압을 출력한다. 아두이노 우노를 사용할 때 **0.1** 도의 온도차이를 측정할 수 있을까? 측정가능한 온도차이가 얼마인지를 계산하시오.

Solution :

1. 양자화

온도 범위 :

-20 ~ 130,

$$130 - (-20) = 150$$

2. 아두이노 ADC 10Bits , 1024 단계

3. 범위 : 0.146 도 = $150/1024$

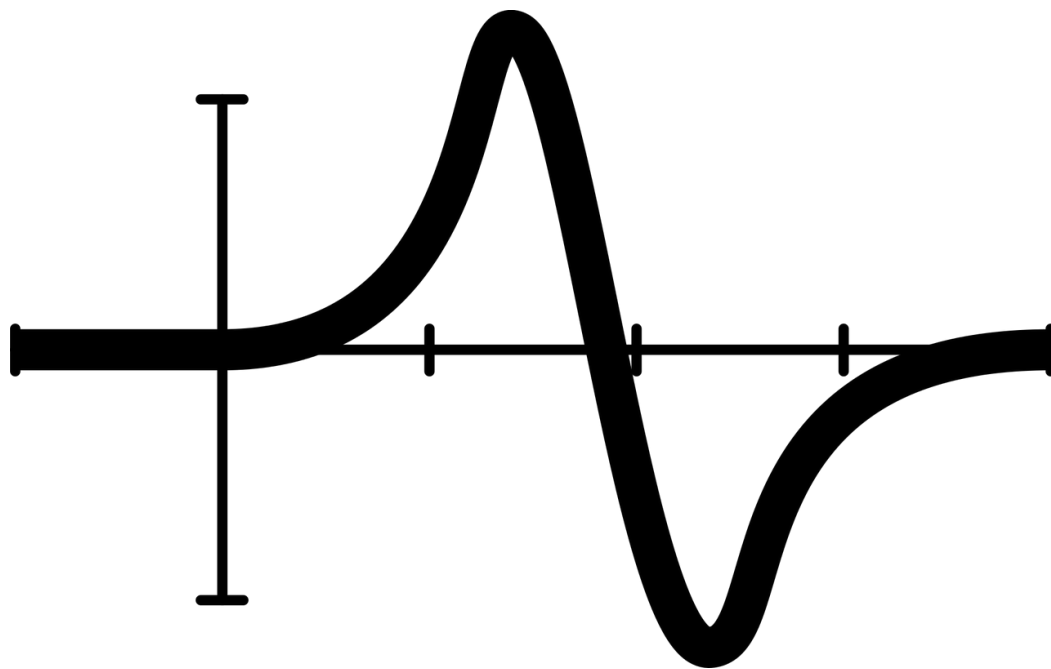
** 측정가능한 최소차이는 0.146도로 0.1 도의 온도차이는 측정할 수 없다.

*** -20 도에서 130도 까지 0.146 도의 차이로 판별할 수 있다.



아두이노용 C 언어

- 수학함수



`min (a, b) ;`

`max (a, b) ;`

`randomSeed (value) ;`

`random (max) ;`

`random (min, max) ;`

```
int number;  
randomSeed(analogRead(A0));  
number = random(1000);  
number = random(50, 200);
```

```
#undef max
```

```
#define max(a,b)  ( (a) > (b) ? (a) : (b) )
```

```
#undef min
```

```
#define min(a,b)  ( (a) > (b) ? (b) : (a) )
```



```
#undef min
```

```
#define min(a,b)  ( (a) > (b) ? (b) : (a) )
```

```
#undef max
```

```
#define max(a,b)  ( (a) > (b) ? (a) : (b) )
```

기타 수학함수들

abs (x)

sin (x)

asin ()

ceil (x)

cos (x)

acos ()

floor (x)

tan (x)

atan ()

exp (x)

atan2 (x, y)

log (x)

sqrt (x)

pow (x)



아두이노용 C 언어

- 디버깅

DEBUG

```
void setup()  
{  
    Serial.begin(9600);  
}
```

```
Serial.print(data);
```

```
Serial.println(data);
```



```
loop () {
```

```
...
```

```
// 의심되는 부분
```

```
Serial.println(의심되는 데이터);
```

```
...
```

```
}
```

```
int pushButton =2;

void setup() {
    Serial.begin(9600);
    pinMode(pushButton, INPUT_PULLUP);
    pinMode(13, OUTPUT);
}

void loop() {
    int buttonState = digitalRead(pushButton);
    Serial.print("buttonState = ");
    Serial.println(buttonState);
    if (buttonState) {
        digitalWrite(13, HIGH);
        Serial.println("Push SW ON");
    }
    else {
        digitalWrite(13, LOW);
        Serial.println("Push SW OFF");
    }
    delay(1);
}
```

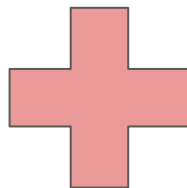



아두이노용 C 언어

– millis() 함수

millis ()

```
01: unsigned long previousMillis = 0;
02: unsigned long currentMillis = millis();
03:
04:     if (currentMillis - previousMillis >= 1000) {
05:         previousMillis = currentMillis;
06:         if (ledState == LOW) { ledState = HIGH; }
07:         else { ledState = LOW; }
08:     }
```



버튼을 누르는 동안
LED2 를 켜고 싶다.

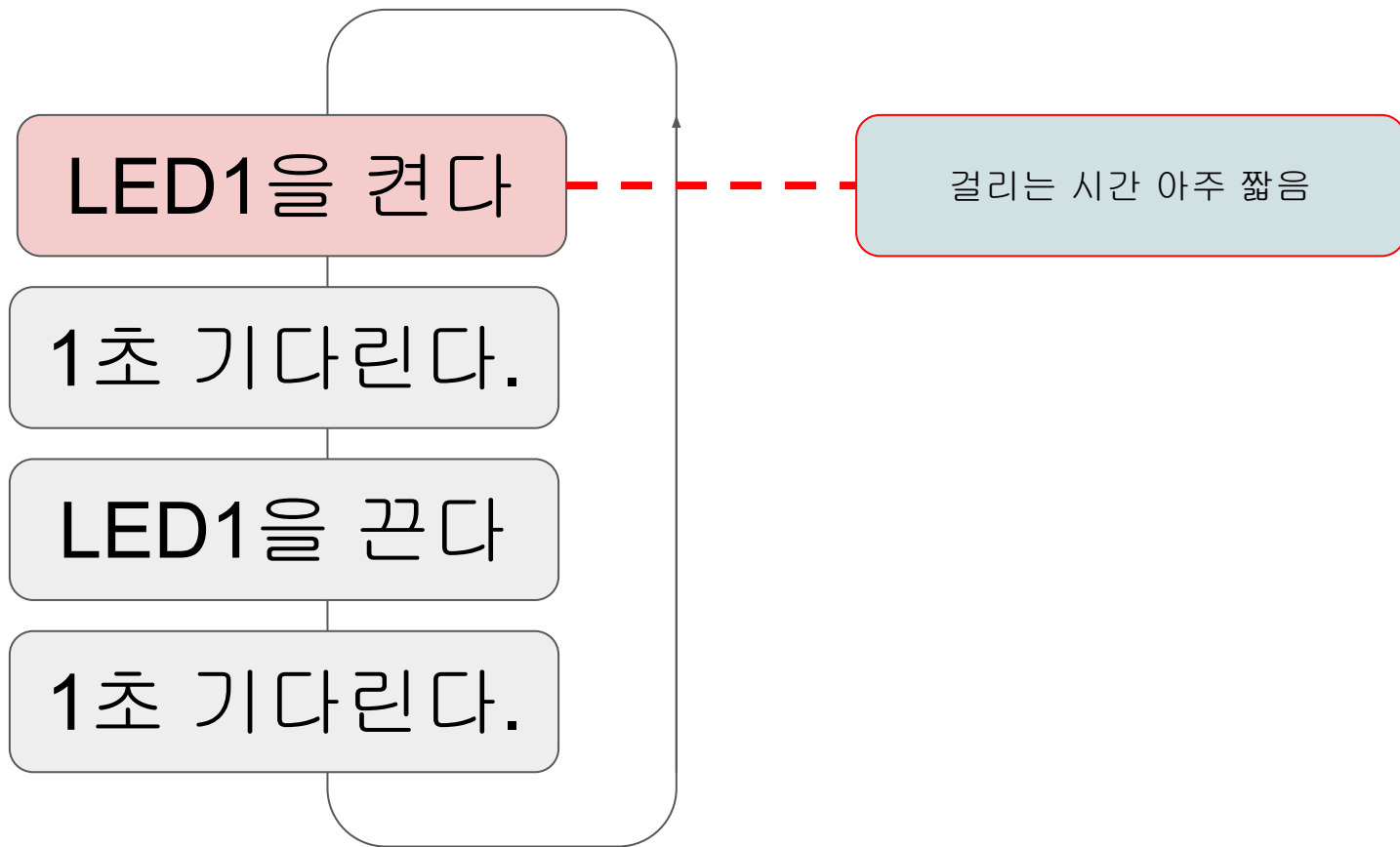
LED1을 켜다

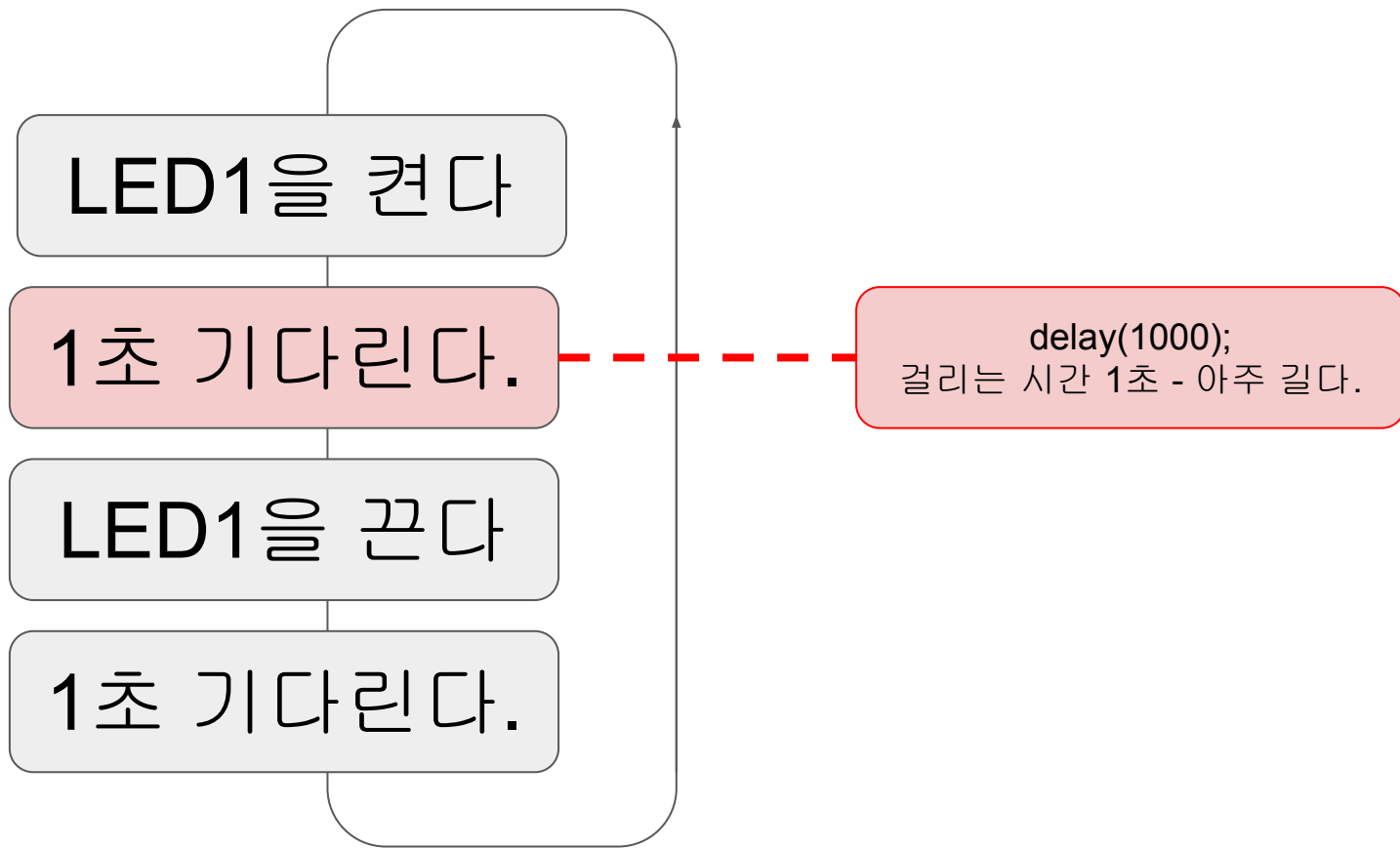
1초 기다린다.

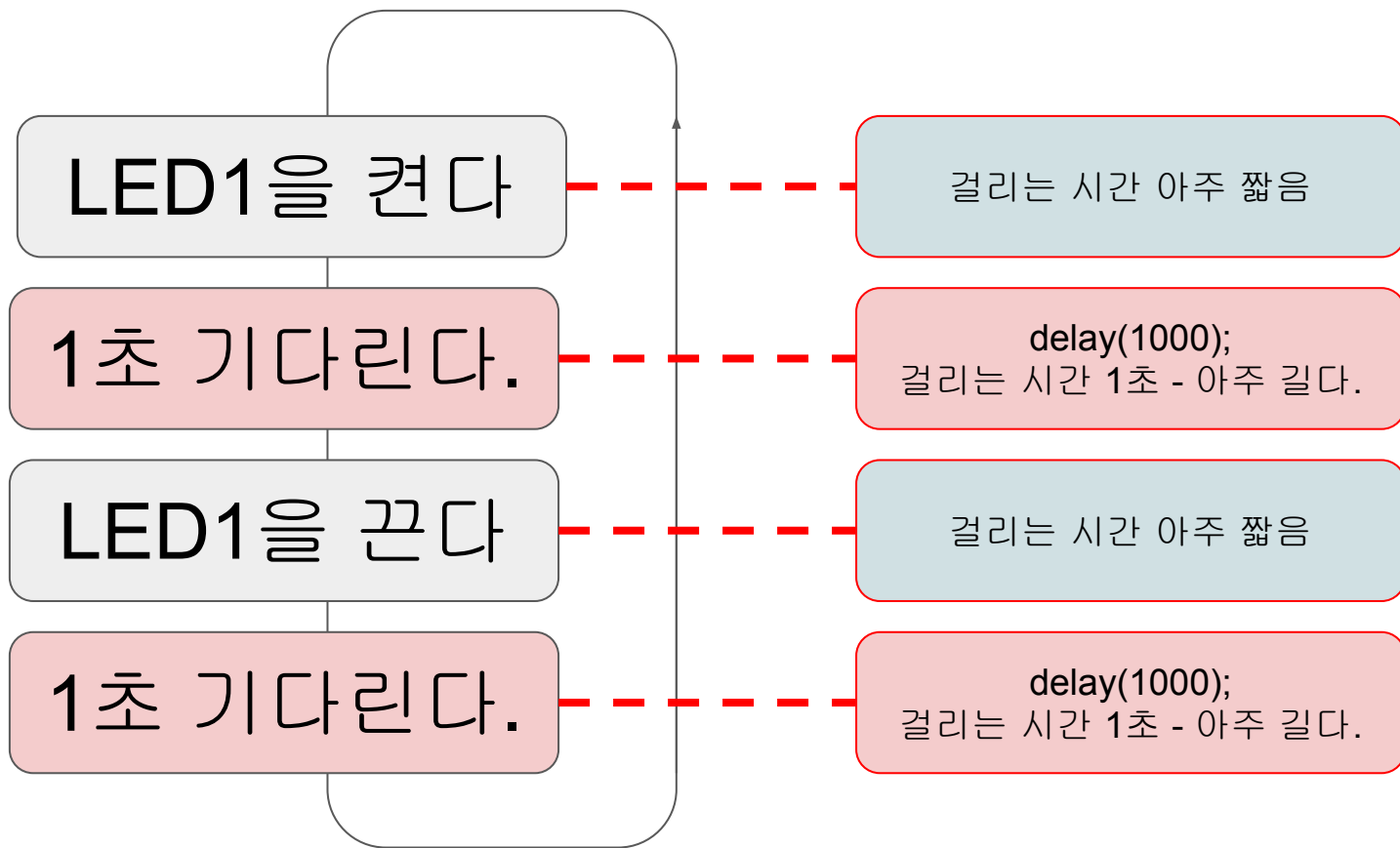
LED1을 끈다

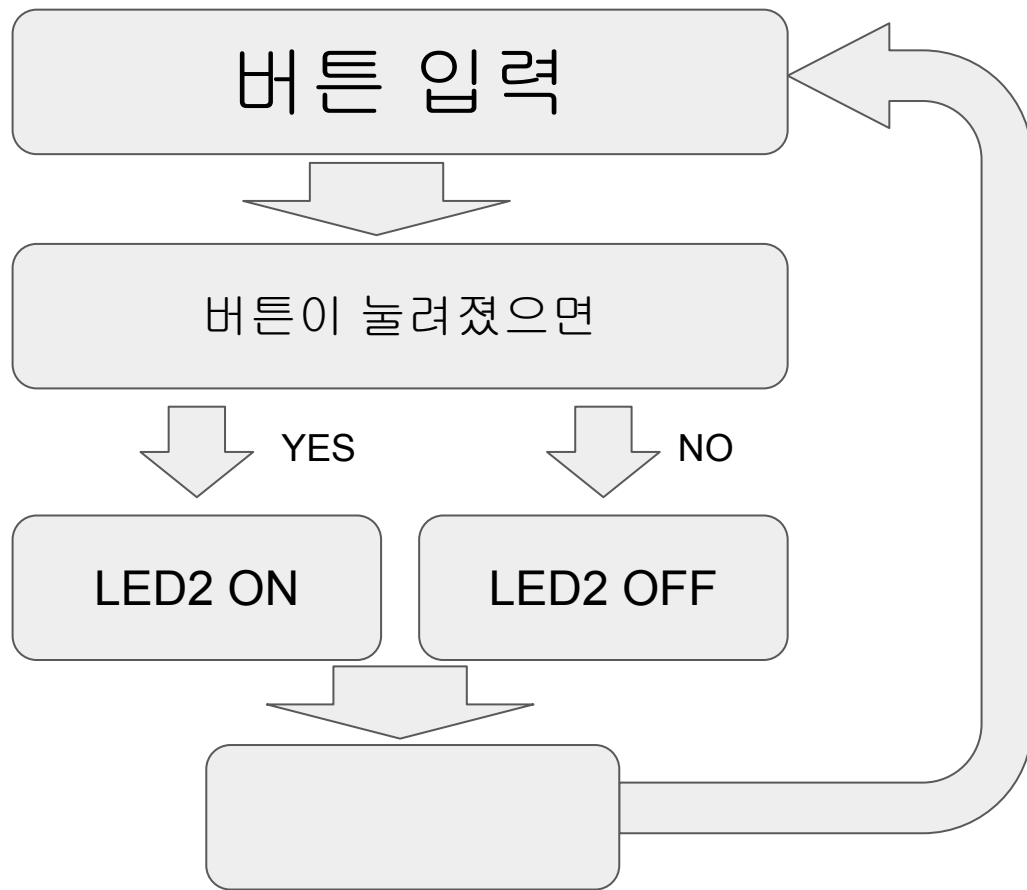
1초 기다린다.

걸리는 시간 아주 짧음









LED1을 켜다

1초 기다린다.

LED1을 끈다

1초 기다린다.

버튼 입력

버튼이 눌러졌으면

YES

NO

LED2 ON

LED2 OFF





1 단계

1 초 간격으로 LED 1 을 끄고 켜다

2 단계

버튼이 눌러졌을때만 LED2 를 켜다



```
const int LED1 = 6;
void setup() {
  pinMode(LED1, OUTPUT);
}
void loop() {
  digitalWrite(LED1, HIGH);
  delay(1000);
  digitalWrite(LED1, LOW);
  delay(1000);
}
```



```
const int LED2 = 7;
const int SW = 10;
void setup() {
  pinMode(LED2, OUTPUT);
  pinMode(SW, INPUT);
}
void loop() {
  if(digitalRead(SW)) digitalWrite(LED2, HIGH);
  else digitalWrite(LED2, LOW);
}
```



```
const int LED1 = 6;
const int LED2 = 7;
const int SW = 10;

void setup() {
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(SW, INPUT);
}

void loop() {
  digitalWrite(LED1, HIGH);
  delay(1000);
  digitalWrite(LED1, LOW);
  delay(1000);
  if(digitalRead(SW)) digitalWrite(LED2, HIGH);
  else digitalWrite(LED2, LOW);
}
```

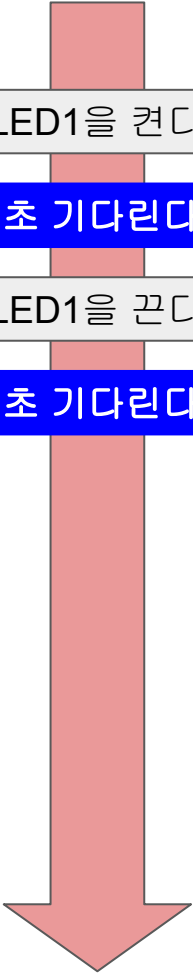



1 단계

1 초 간격으로 LED 1 을 끄고 켜다

2 단계

버튼이 눌러졌을때만 LED2 를 켜다



LED1을 켜다

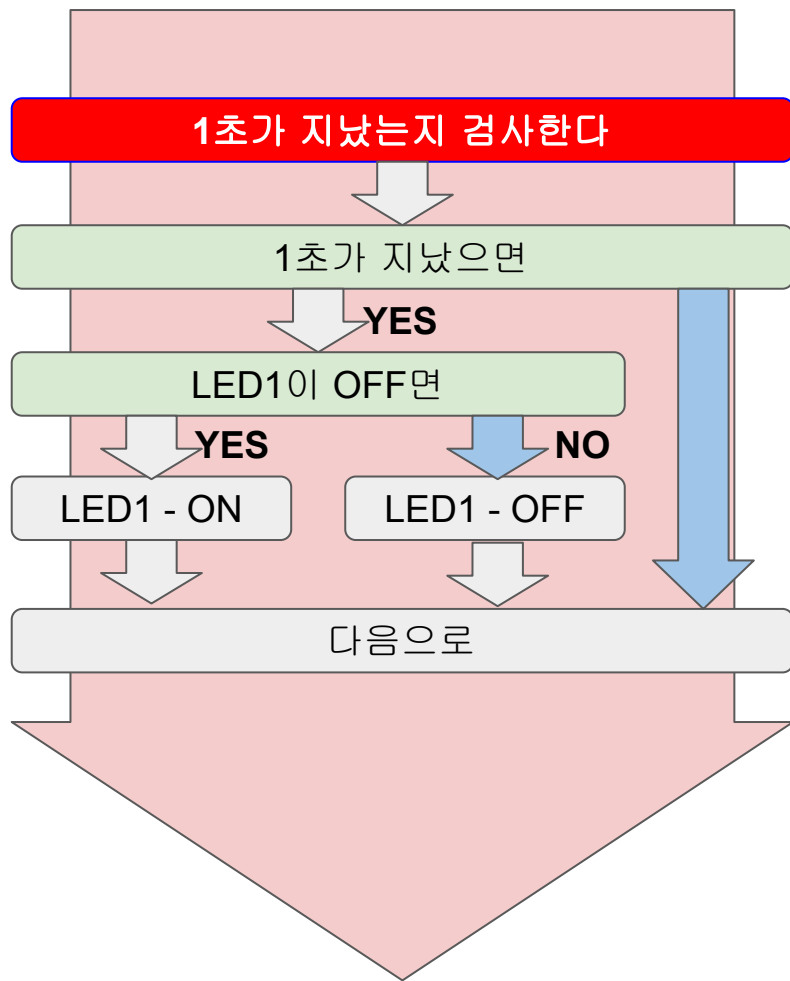
1초 기다린다.

LED1을 끈다

1초 기다린다.

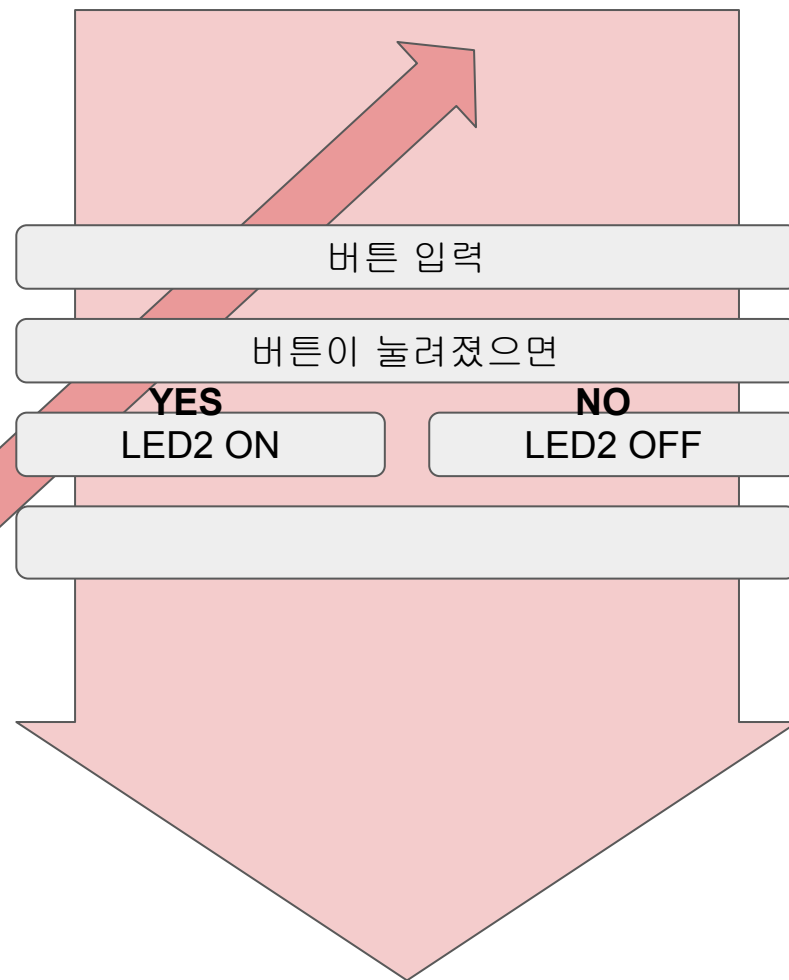
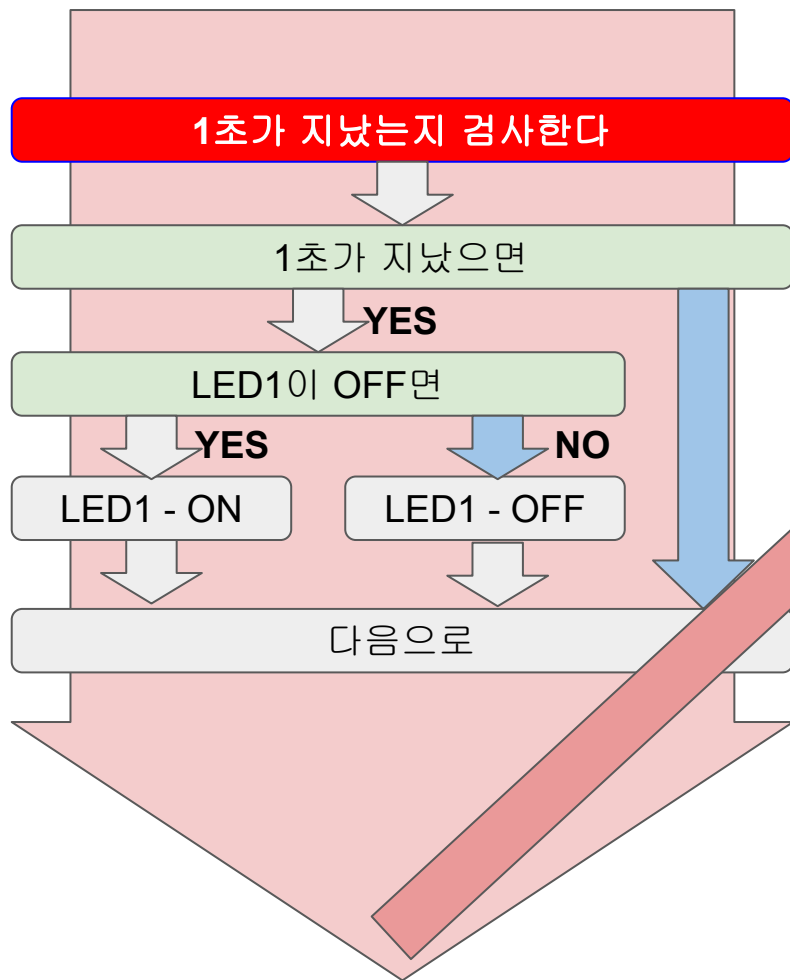
1 단계

1 초 간격으로 LED 1 을 끄고 켜다

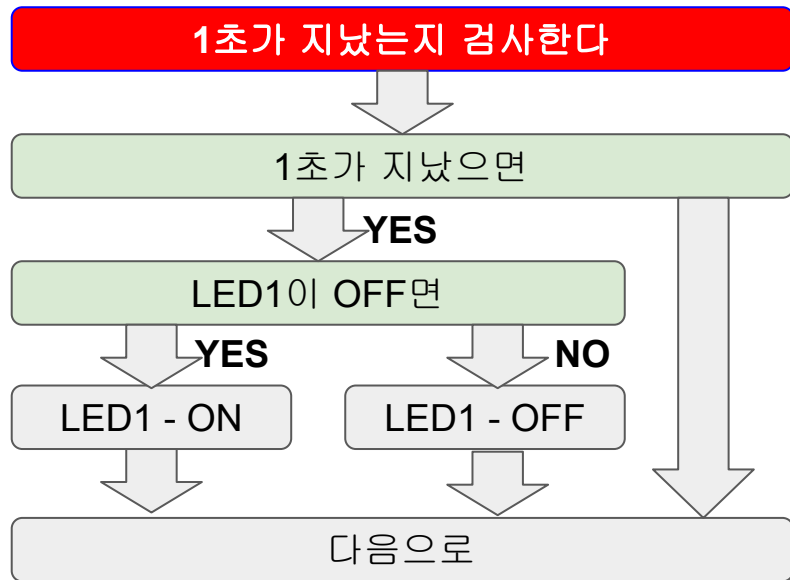


1 단계

1 초 간격으로 LED 1 을 끄고 켜다



```
01: unsigned long previousMillis = 0;
02: unsigned long currentMillis = millis();
03:
04:     if (currentMillis - previousMillis >= 1000) {
05:         previousMillis = currentMillis;
06:         if (ledState == LOW) { ledState = HIGH; }
07:         else { ledState = LOW; }
08:     }
```



```
unsigned long previousMillis = 0;
```

```
unsigned long currentMillis = millis();
```

```
if (currentMillis - previousMillis >= 1000)
```

```
{
```

```
    previousMillis = currentMillis;
```

```
    if (ledState == LOW) {
```

```
        ledState = HIGH;
```

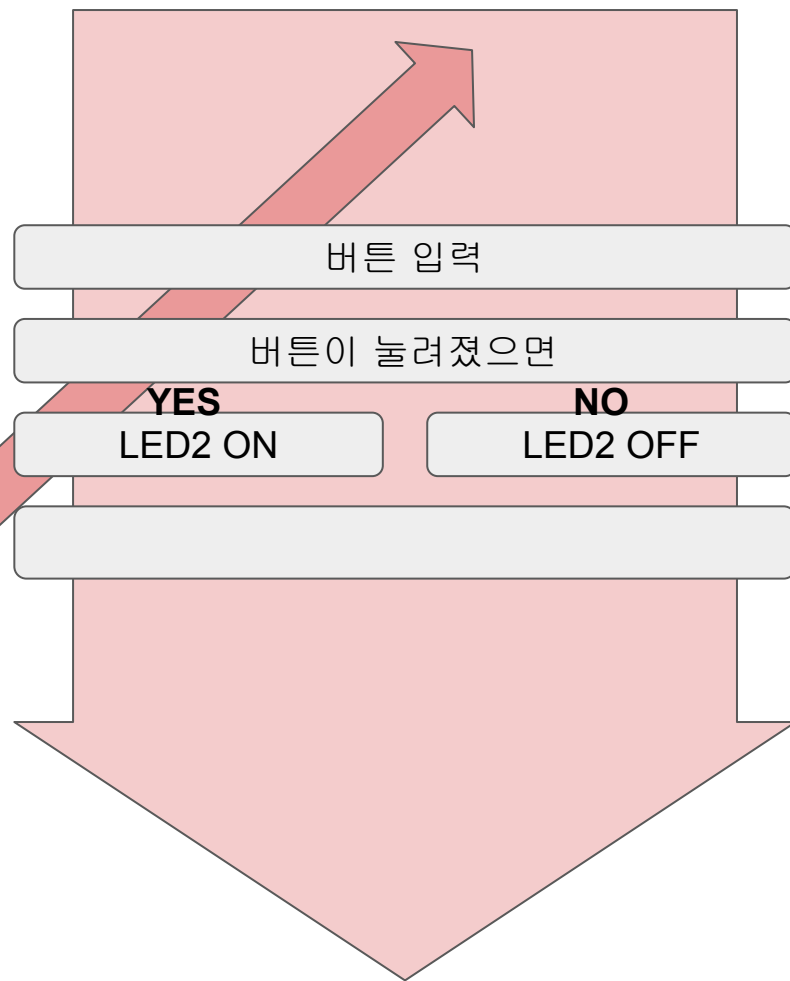
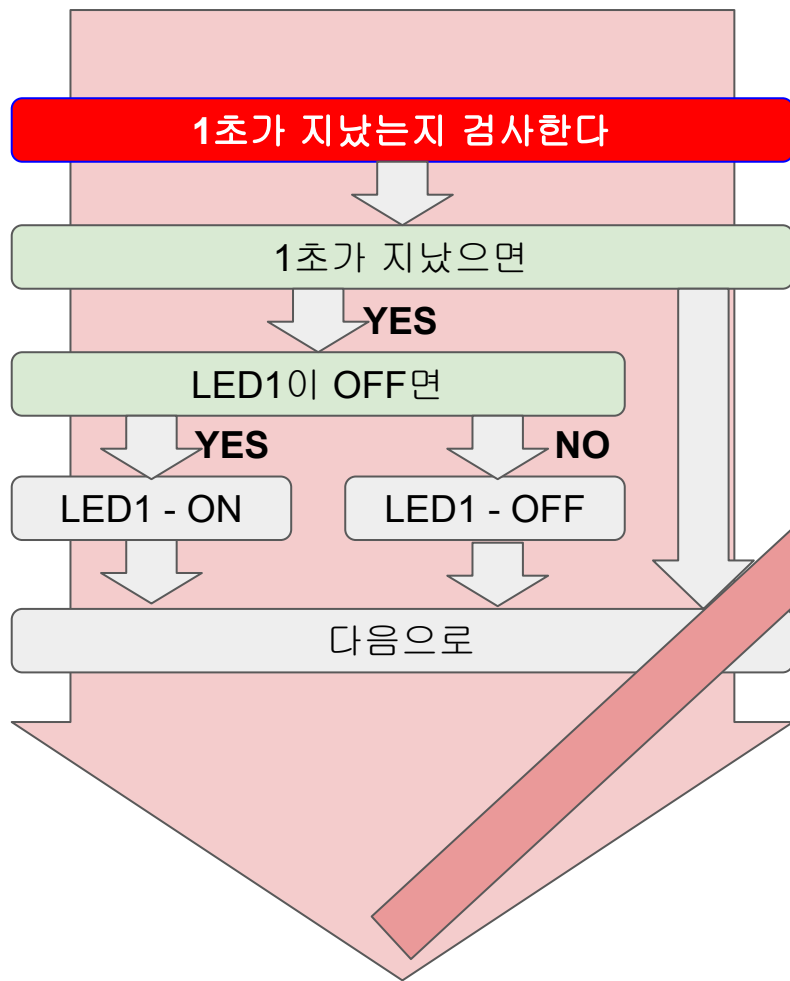
```
    }
```

```
    else {
```

```
        ledState = LOW;
```

```
    }
```

```
}
```



1초가 지났는지 검사한다

1초가 지났으면

YES

LED1이 OFF면

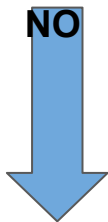
YES

LED1 - ON

NO

LED1 - OFF

NO



다음으로

버튼 입력

버튼이 눌러졌으면

YES

LED2 ON

NO

LED2 OFF

1 단계

1 초 간격으로 LED 1 을 끄고 켜다

2 단계

버튼이 눌러졌을때만 LED2 를 켜다

1초가 지났는지 검사한다

1초가 지났으면

YES

LED1이 OFF면

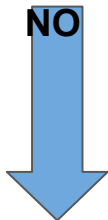
YES

LED1 - ON

NO

LED1 - OFF

NO



다음으로

버튼 입력

버튼이 눌러졌으면

YES

LED2 ON

NO

LED2 OFF

```
const int LED1 = 6;  
const int LED2 = 7;  
const int SW = 10;
```

```
int ledState = LOW;  
unsigned long previousMillis = 0;
```

```
void setup() {  
  pinMode(LED1, OUTPUT);  
  pinMode(LED2, OUTPUT);  
  pinMode(SW, INPUT);  
}
```

```
void loop() {  
  unsigned long currentMillis = millis();  
  
  if (currentMillis - previousMillis >= 1000) {  
    previousMillis = currentMillis;  
    if (ledState == LOW) ledState = HIGH;  
    else ledState = LOW;  
    digitalWrite(LED1, ledState);  
  }  
  
  if(digitalRead(SW)) digitalWrite(LED2, HIGH);  
  else digitalWrite(LED2, LOW);  
}
```

END