

아두이노

임성국

Install Arduino IDE



Search the Arduino Website



Home

Buy

Download

Products ▾

Learning ▾

Forum

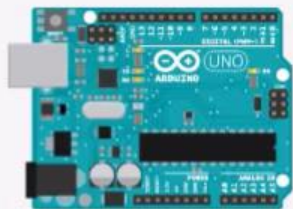
Support ▾

Blog

LOG IN

SIGN UP

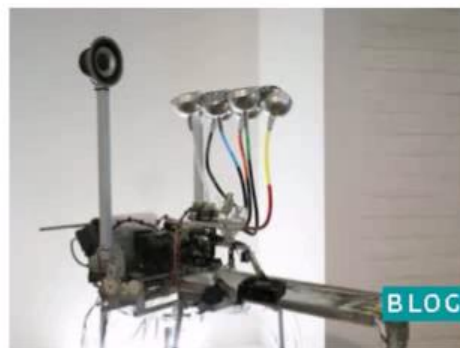
WHAT IS ARDUINO?



BUY AN ARDUINO



LEARN ARDUINO



BLOG

AN ELECTRO-MECHANICAL
DRAWING MACHINE DRIVEN
BY MUSIC

+ ◁ ▷ ▢
ESLOV
IoT KIT

BACK US ON
KICKSTARTER



winduino.co.kr



[Home](#) [Buy](#) [Download](#) [Products](#) [Learning](#) [Forum](#) [Support](#) [Blog](#)

[LOG IN](#) [SIGN UP](#)

DOWNLOAD

ENGLISH

Download the Arduino Software



ARDUINO 1.6.12

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 Installer

Windows ZIP file for non admin install

Mac OS X 10.7 Lion or newer

Linux 32 bits

Linux 64 bits

Linux ARM (experimental)

[Release Notes](#)

[Source Code](#)

[Checksums \(sha512\)](#)

winduino.co.kr



Buy

Download

Products ▾

Learning ▾

Forum

Support ▾

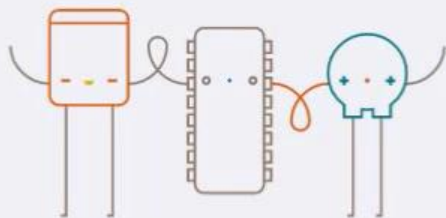
Blog

LOG IN

SIGN UP

Support the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)



SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED **10,424,020** TIMES. (IMPRESSIVE!) NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS, HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING THE IDE TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEITS. HELP ACCELERATE ITS DEVELOPMENT WITH A SMALL CONTRIBUTION! REMEMBER: OPEN SOURCE IS LOVE!



\$5

\$10

\$25

\$50

OTHER

JUST DOWNLOAD

CONTRIBUTE & DOWNLOAD

https://www.arduino.cc/en/Make

응용 프로그램 도구 temp

파일 홈 공유 보기 관리

temp 검색

내 PC > 새 볼륨 (D:) > temp

이름	수정된 날짜	유형
S1_일학습병행제		
10		
oCam		
Photos		
출판사 프리렉		
OneDrive		
내 PC		
GDrive		
다운로드		
동영상		
바탕 화면		
사진		
음악		
로컬 디스크 (C:)		
새 볼륨 (D:)		
Elements (F:)		
새 볼륨 (H:)		
새 볼륨 (I:)		
HDD1 (\\wipt2		
Elements (F:)		

13개 항목 1개 항목 선택함 84.2MB

as_accept
DSLR_Dat
gw100
gx2
IMAGE
MAGNITUDE
Photos
CTCU

2015-10-02 오후... 파일 폴더
2016-09-07 오전... 파일 폴더
2015-10-28 오후... 파일 폴더
2015-10-28 오후... 파일 폴더
2016-10-04 오후... 파일 폴더
2016-09-07 오후... 파일 폴더
2016-10-05 오후... 파일 폴더
2015-11-11 오전... 파일 폴더
2015-11-11 오전... 파일 폴더
5 오후... 파일 폴더
8 오후... 파일 폴더
5 오후... 응용 프로그램
7 오후... IMG 파일

Arduino Setup: Installation Options

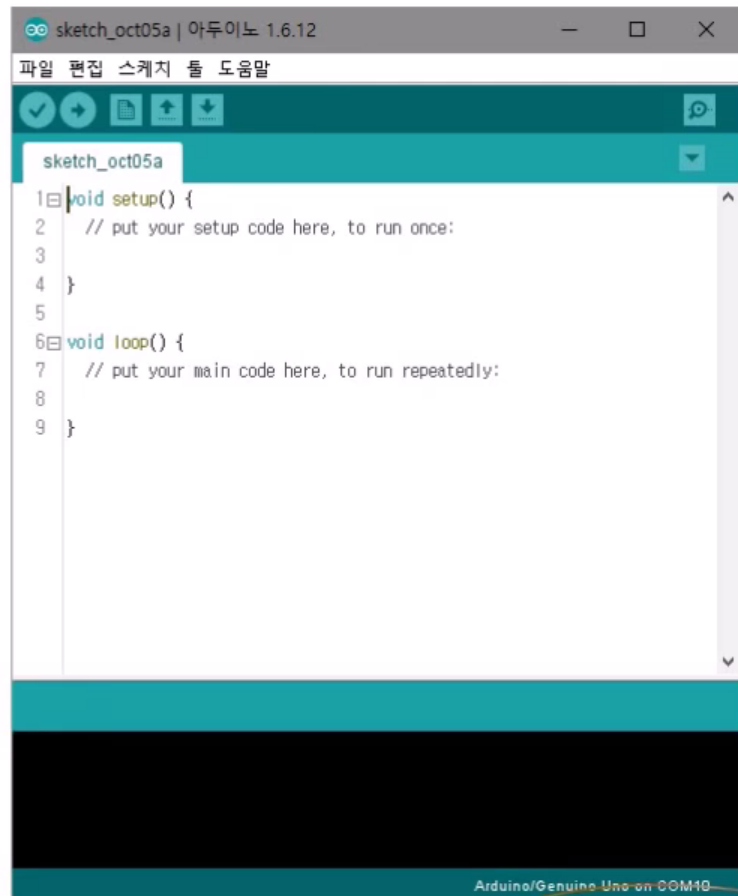
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

- ☒ Install Arduino software
- ☒ Install USB driver
- ☒ Create Start Menu shortcut
- ☒ Create Desktop shortcut
- ☒ Associate .ino files

Space required: 402.3MB

Cancel Nullsoft Install System v3.0 < Back Next >

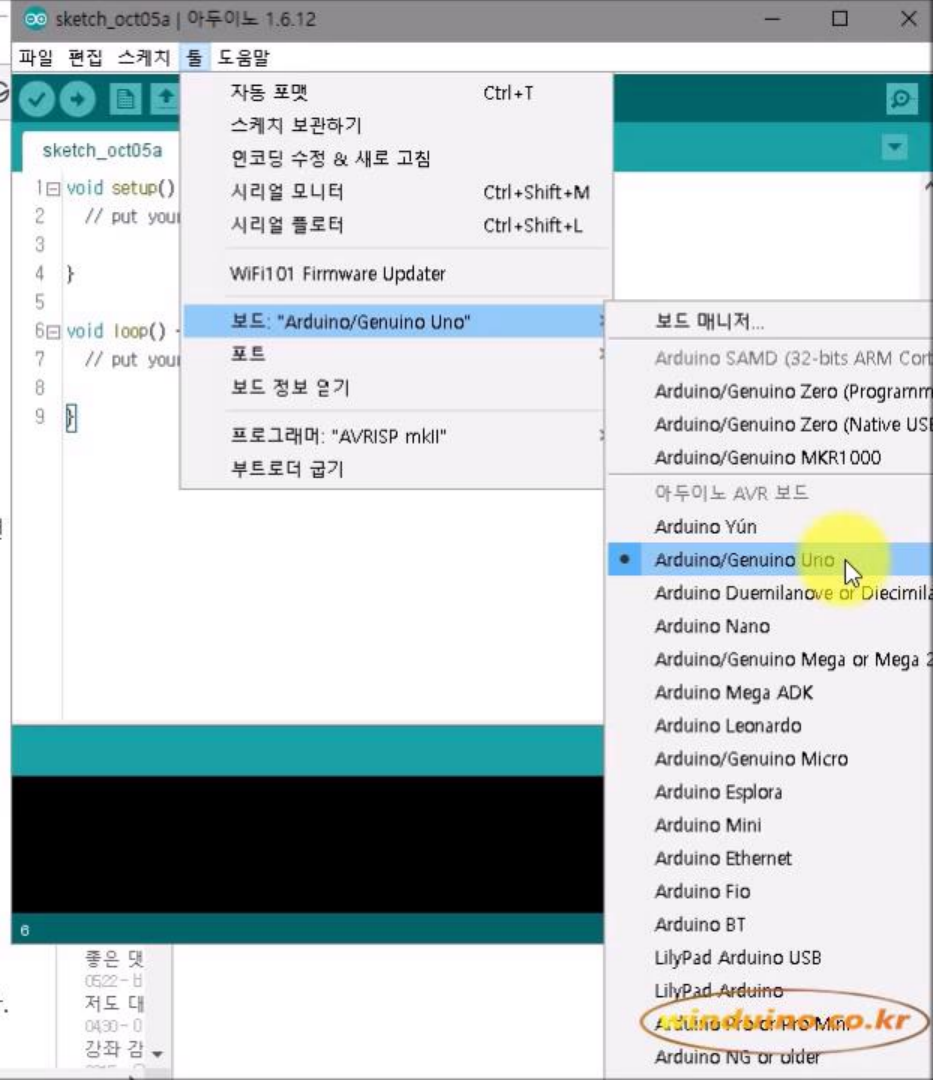




저 주소로 접속하면 다운받을 프로그램의 링크가 다시 나옵니다. 영어를 읽을 수 있으면
읽어서 그대로 따라하시면 됩니다. 아니면 아래에 있는 링크를 따라갑시다.



선택해서 클릭합니다.

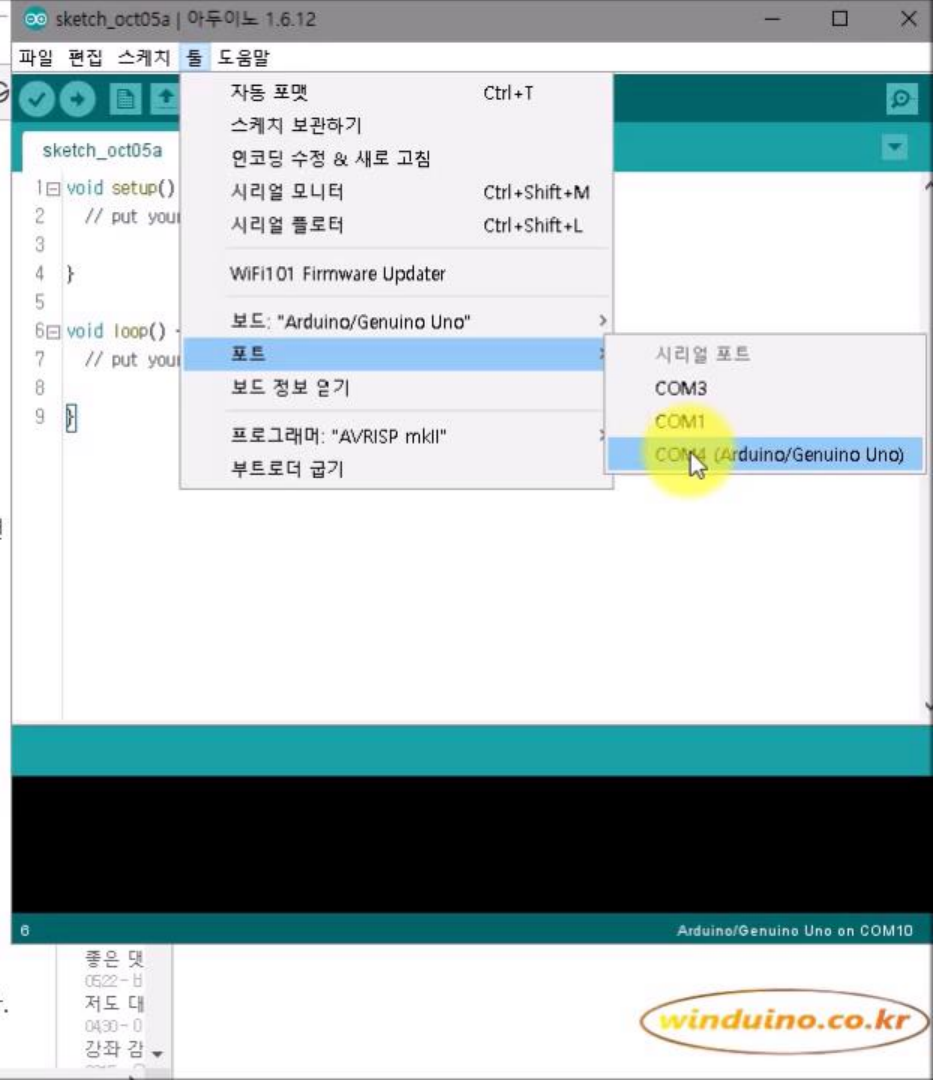




저 주소로 접속하면 다운받을 프로그램의 링크가 다시 나옵니다. 영어를 읽을 수 있으면 읽어서 그대로 따라하시면 됩니다. 아니면 아래에 있는 링크를 따라갑시다.



선택해서 클릭합니다.



Arduino IDE Update

아두이노 IDE 업데이트

Blink | 아두이노 1.6.12

파일 편집 스케치 툴 도움말

Blink \$

10 This example code is in the public domain.
11
12 modified 8 May 2014
13 by Scott Fitzgerald
14
15 modified 2 Sep 2016
16 by Arturo Guadalupi
17 */
18
19
20
21 // the setup function runs once when you press reset
22 void setup() {
23 // initialize digital pin LED_BUILTIN as an output:
24 pinMode(LED_BUILTIN, OUTPUT);
25 }
26
27 // the loop function runs over and over again forever
28
29

내보내기 취소, 변경은 먼저 저장해야 합니다.

20 Arduino/Genuino Uno on COM10

라이브러리 매니저

타입 All 토픽 All 검색에 필터하기...

Adafruit MCP23008 library by Adafruit
Arduino Library for the MCP23008 (and '9) I2C I/O expander
[More info](#)

Adafruit MCP23017 Arduino Library by Adafruit
Library for the MCP23017 I2C Port Expander Library for the MCP23017 I2C Port Expander
[More info](#)

Adafruit MCP4725 by Adafruit
MCP4725 12-bit I2C DAC MCP4725 12-bit I2C DAC
[More info](#)

Adafruit MCP9808 Library by Adafruit
Arduino library for the MCP9808 sensors in the Adafruit shop
[More info](#)

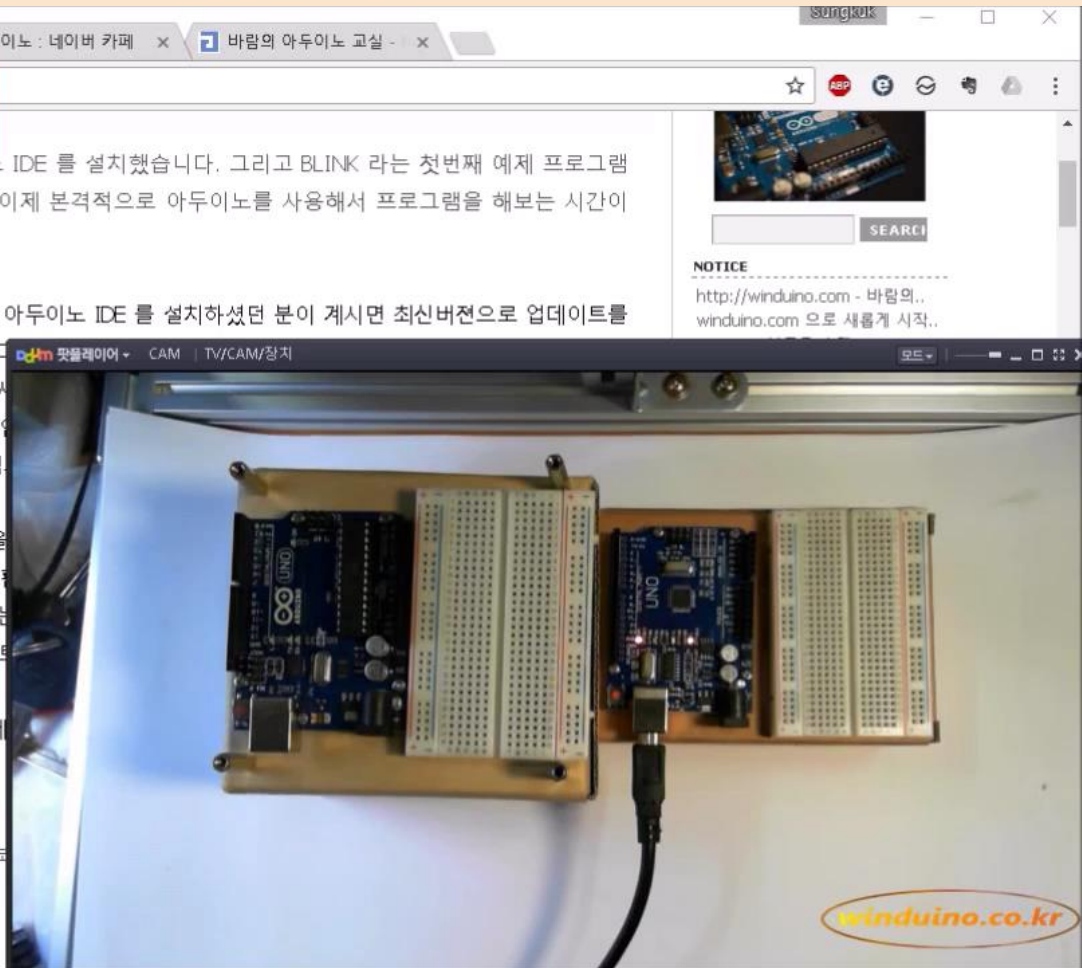
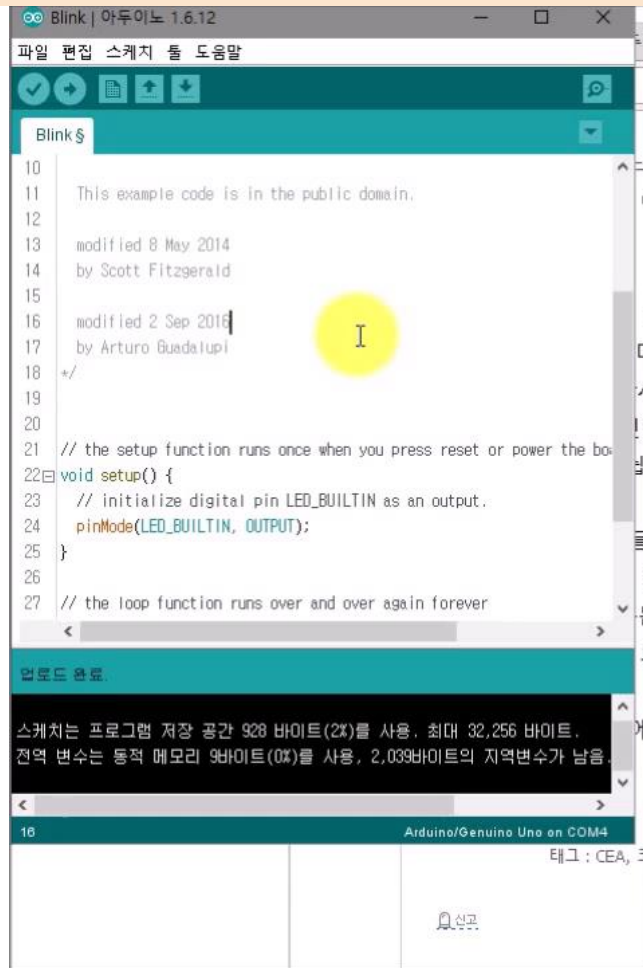
설치

닫기

태그 : CEA, 코딩교육, 아두이노, arduino, 아두이노 강의, 아두이노 IDE, 업데이트, 라이브러리, 보드

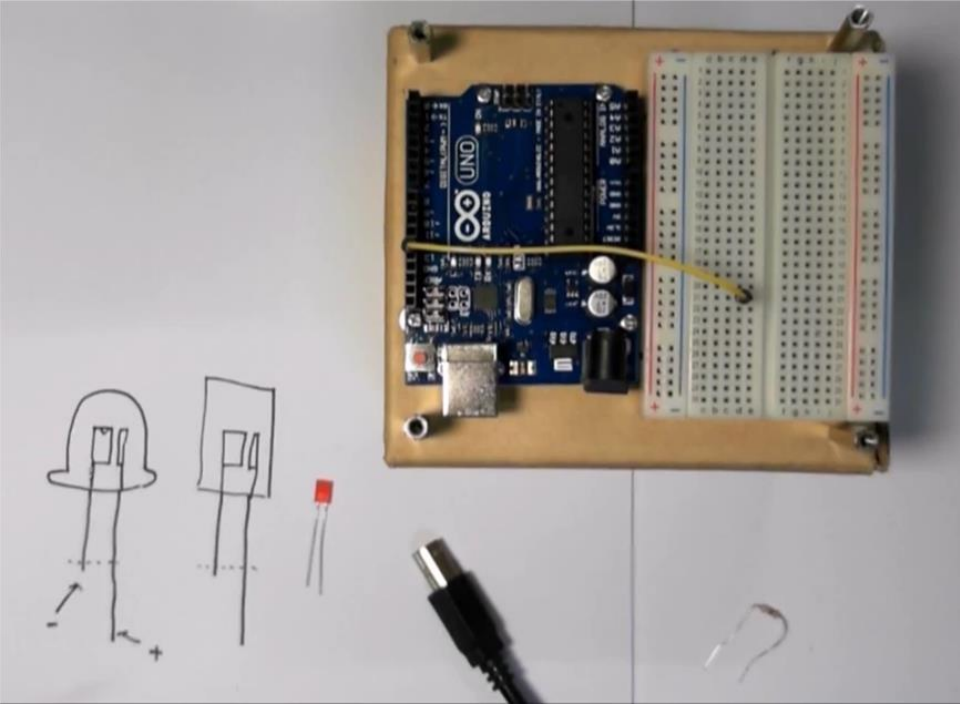
winduino.co.kr

아두이노 IDE 업데이트

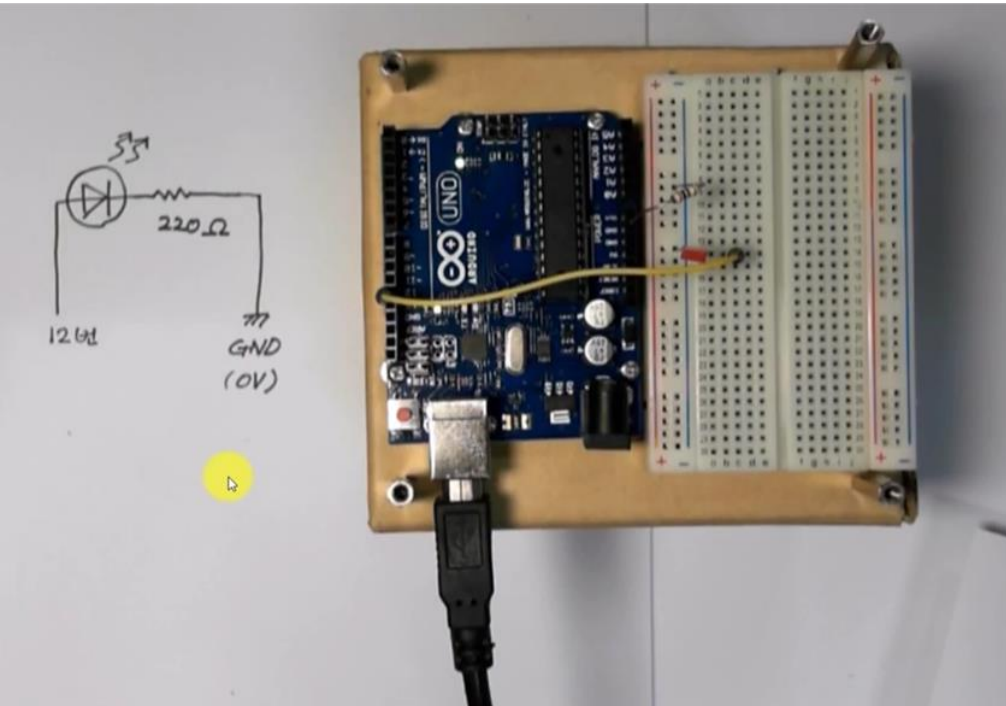


Blink LED -1

LED



LED 와 저항



12번 핀 LED 켜고 끄기

C:\Program Files (x86)\Arduino\hardware\arduino\avr\variants\standard\pins_arduino.h - Notepad++

파일(F) 편집(E) 찾기(S) 보기(V) 인코딩(N) 언어(L) 설정(T) 매크로 실행 플러그인 장 관리 ?

Folder as Workspace

hdocs
img
503.html
applications.html
bitnami.css
favicon.ico
helloworld.html
index.html
index.php
page_vc.html

43 static const uint8_t SS = PIN_SPI_SS;
44 static const uint8_t MOSI = PIN_SPI_MOSI;
45 static const uint8_t MISO = PIN_SPI_MISO;
46 static const uint8_t SCK = PIN_SPI_SCK;
47
48 #define PIN_WIRE_SDA (18)
49 #define PIN_WIRE_SCL (19)
50
51 static const uint8_t SDA = PIN_WIRE_SDA;
52 static const uint8_t SCL = PIN_WIRE_SCL;

pins_arduino.h

CEA_007_1.ino | 아두이노 1.6.12

파일 편집 스케치 도움말

CEA_007_1.ino

1 void setup() {
2 pinMode(12, OUTPUT);
3 }
4
5 void loop() {
6 digitalWrite(12, HIGH);
7 digitalWrite(12, LOW);
8 }

스케치는 프로그램 저장 공간 734 바이트(2K)를 사용, 최대 32,256 바이트.
변수 선언은 동적 메모리 96바이트(0K)를 사용, 2,039바이트의 지역변수가 남음. 최대는 2,048 바이트.

Arduino/Genuino Uno on COM4

winduino.co.kr

12V

3V

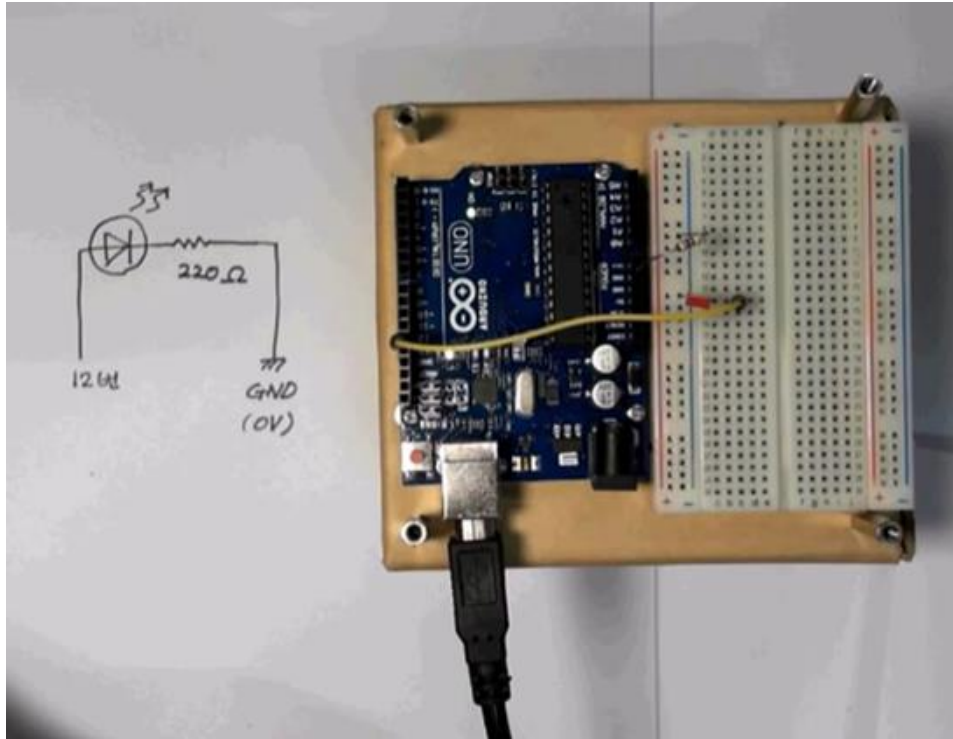
220Ω

GND (0V)

Arduino Uno

Blink LED -2

LED



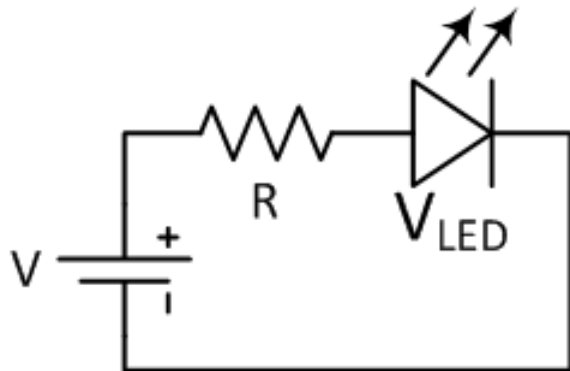
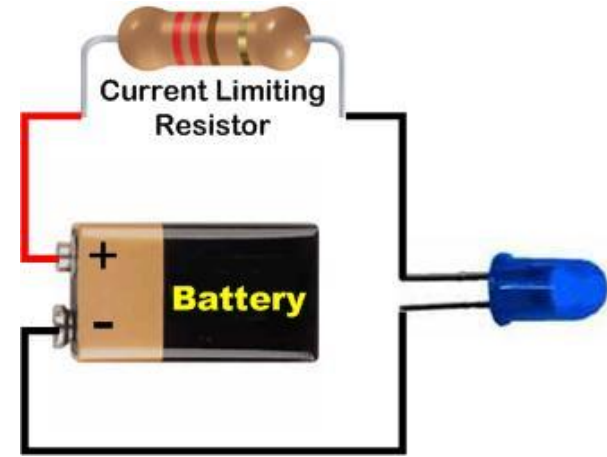
```
Blink | 아두이노 1.8.3
파일 편집 스케치 툴 도움말

Blink $
1 void setup() {
2   pinMode(LED_BUILTIN, OUTPUT);
3 }
4
5 void loop() {
6   digitalWrite(LED_BUILTIN, HIGH);
7   delay(1000);
8   digitalWrite(LED_BUILTIN, LOW);
9   delay(1000);
10 }
```

4 Arduino/Genuino Uno on COM13

$$V = I \times R$$

$$I = V / R$$



LED 의 전압강하 = 2V

$$1.5V - 2V = 3V$$

$$2. I = (5-2)/R(300)$$

$$3. I = 10mA$$

$$(I < 30mA)$$

12번 핀 LED 켜고 끄기

```
pinMode(12, OUTPUT);
```

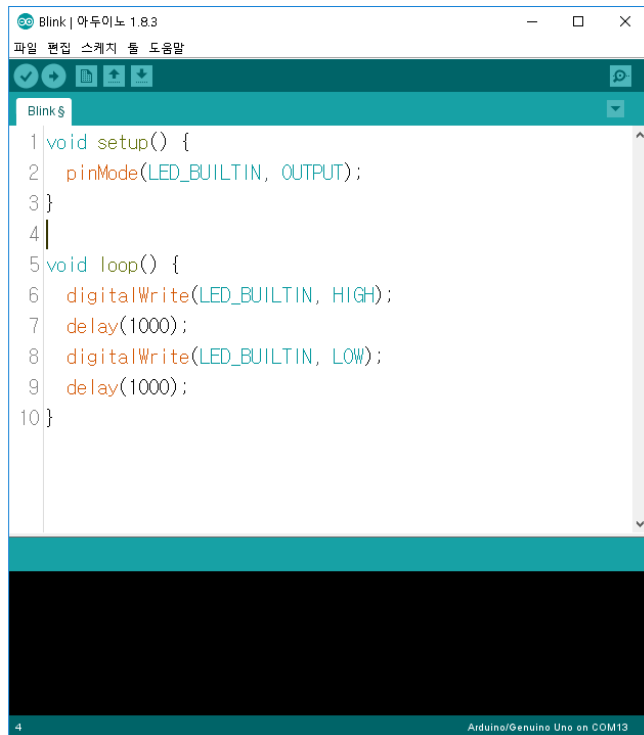
. . .

```
digitalWrite(12, 1);
```

```
delay(1000);
```

```
digitalWrite(12, 0);
```

```
delay(1000);
```





아두이노용 C 언어

– `setup()` / `loop()`

순차적 프로그램

시작 →

주소	내용	int main(void) {
0	jc8eJ7df7832#sajhfas~18hjkdsfa87j-dfal312-JKc-[int i = 1;
1	~!gbvdfjkjC0jkyJKhI76832J76%^%H2J13dfS90	int j = 2;
2	JHYUKGKGYUKhBJ%(*^5643%\$^ytfU&^6fSal3	int k, l;
3	jhhNBJe%\$DS%\$DS54fJHJHHDD%^RFFGYU&^6	k = i+j;
4	JJhjHF*&^\$%#@GCHHJ(&NBM DSHFG215466TFY	l = i*j;
5	vyu87*-0iJ_l80OIhbgV6%4GBVYy*Y58DCXRvhyuYT	printf("i+j = %d, i*j = %d", k, l);
...	...	return 0;
99	4KJKbhYUT%^65#@254#@.....	}

→ 끝

<http://codepad.org/>

<http://ideone.com/>


```
main()  
{
```

setup();

한번만 실행

```
while(1) {
```

loop();

무한 반복 실행

```
}
```

sketch_oct15a | 아두이노 1.6.7

파일 편집 스케치 툴 도움말

sketch_oct15a

```
1 void setup() {  
2   // put your setup code here, to run once:  
3  
4 }  
5  
6 void loop() {  
7   // put your main code here, to run repeatedly:  
8  
9 }
```

Arduino/Genuino Uno on COM13

END

아두이노 부트로더 이해하기

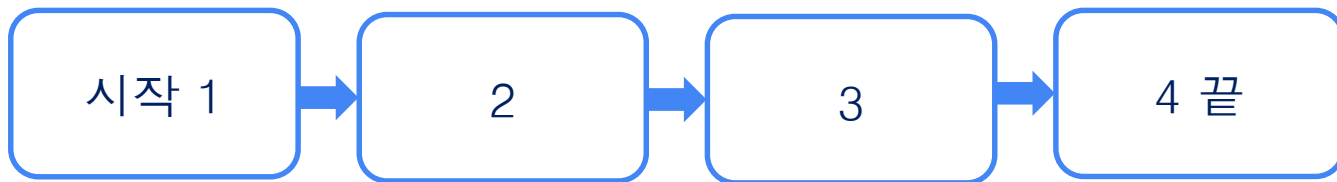
질문 ??

- 질문 : 순차적으로(순서대로) 명령을 실행한다면 어떻게 프로그램을 실행하는 중간에 새로운 프로그램을 써 넣을 수 있을까?

부트로더 (Bootloader)의 이해

- 순차적 프로그램 (Sequence Program)
- 레지스터
- 퓨즈비트
- 부트로더

1. 프로그래밍언어의 기본 특징 : 순차, 반복, 조건



순차 - 처음부터 끝까지, 순서대로, 한번에 하나씩

순차 + 반복

LED 를 1초 간격으로 깜박이는 프로그램

주소	명령(할일)
4	주소 0번 으로 간다.
3	1초 기다린다.
2	LED 를 끈다
1	1초 기다린다.
0	LED 를 켜다.

순차적으로 명령을 실행한다면 어떻게 프로그램을 실행하는 중간에 새로운 프로그램을 써 넣을 수 있을까?

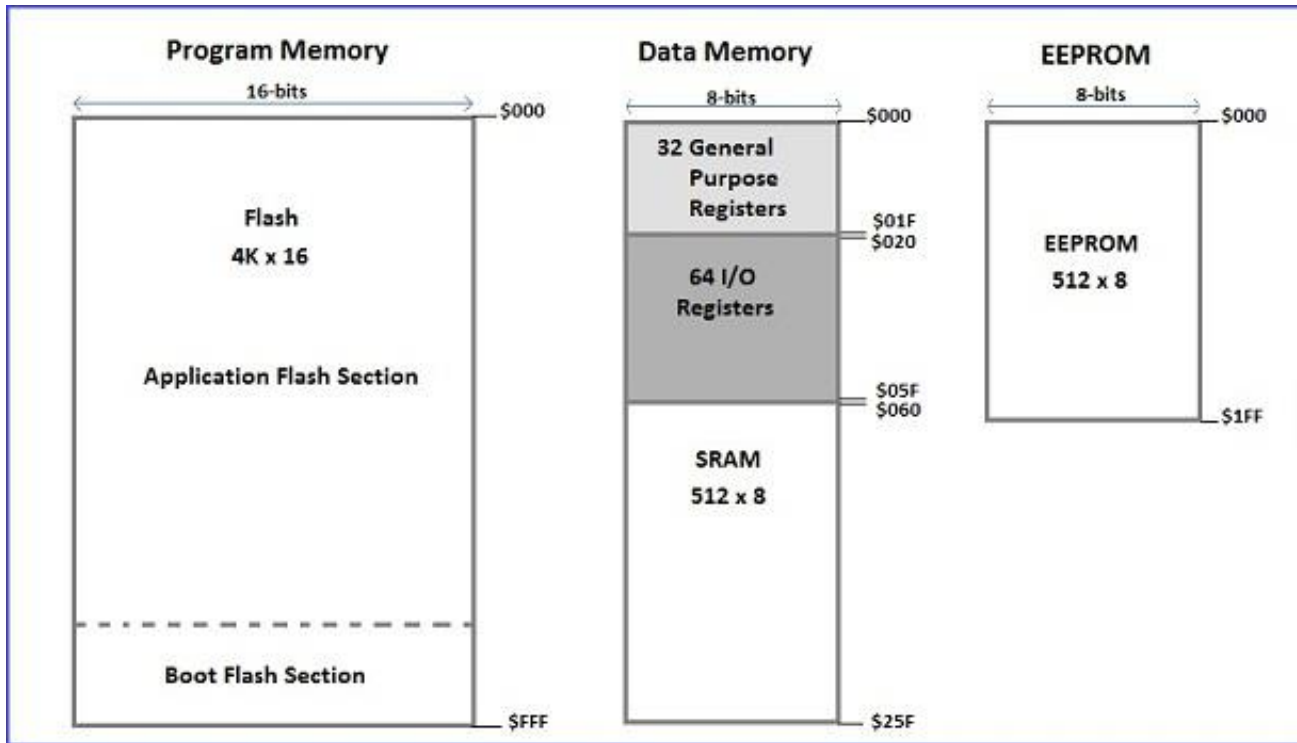
LED 를 1초 간격으로 깜박이는 프로그램

주소	명령(할일)
4	주소 0번 으로 간다.
3	1초 기다린다.
2	LED 를 끈다
1	1초 기다린다.
0	LED 를 켜다.

2. 부트로더 (Bootloader) 를 이해하기 위한 레지스터 (기초)

- 부트로더 : 마이크로프로세서 시작과 동시에 실행되는 기본 프로그램
- 레지스터 : 프로세서 내부에 있는 자료 보관 혹은 처리용 기억장치 [스위치]

아두이노 우노 Atmega328p Memory Map



(0x9C)	Reserved	–	–	–	–	–	–	–	–	
(0x9B)	Reserved	–	–	–	–	–	–	–	–	
(0x9A)	Reserved	–	–	–	–	–	–	–	–	
(0x99)	Reserved	–	–	–	–	–	–	–	–	
(0x98)	Reserved	–	–	–	–	–	–	–	–	
(0x97)	Reserved	–	–	–	–	–	–	–	–	
(0x96)	Reserved	–	–	–	–	–	–	–	–	
(0x95)	Reserved	–	–	–	–	–	–	–	–	
(0x94)	Reserved	–	–	–	–	–	–	–	–	
(0x93)	Reserved	–	–	–	–	–	–	–	–	
(0x92)	Reserved	–	–	–	–	–	–	–	–	
(0x91)	Reserved	–	–	–	–	–	–	–	–	
(0x90)	Reserved	–	–	–	–	–	–	–	–	
(0x8F)	Reserved	–	–	–	–	–	–	–	–	
(0x8E)	Reserved	–	–	–	–	–	–	–	–	
(0x8D)	Reserved	–	–	–	–	–	–	–	–	
(0x8C)	Reserved	–	–	–	–	–	–	–	–	
(0x8B)	OCR1BH	Timer/Counter1 - Output Compare Register B High Byte								135
(0x8A)	OCR1BL	Timer/Counter1 - Output Compare Register B Low Byte								135
(0x89)	OCR1AH	Timer/Counter1 - Output Compare Register A High Byte								135
(0x88)	OCR1AL	Timer/Counter1 - Output Compare Register A Low Byte								135
(0x87)	ICR1H	Timer/Counter1 - Input Capture Register High Byte								135
(0x86)	ICR1L	Timer/Counter1 - Input Capture Register Low Byte								135
(0x85)	TCNT1H	Timer/Counter1 - Counter Register High Byte								134
(0x84)	TCNT1L	Timer/Counter1 - Counter Register Low Byte								134
(0x83)	Reserved	–	–	–	–	–	–	–	–	
(0x82)	TCCR1C	FOC1A	FOC1B	–	–	–	–	–	–	134
(0x81)	TCCR1B	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	133
(0x80)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	–	–	WGM11	WGM10	131
(0x7F)	DIDR1	–	–	–	–	–	–	AIN1D	AIN0D	236
(0x7E)	DIDR0	–	–	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D	251
(0x7D)	Reserved	–	–	–	–	–	–	–	–	
(0x7C)	ADMUX	REFS1	REFS0	ADLAR	–	MUX3	MUX2	MUX1	MUX0	248
(0x7B)	ADCSRB	–	ACME	–	–	–	ADTS2	ADTS1	ADTS0	251
(0x7A)	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	249

0x38 (0x58)	Reserved	–	–	–	–	–	–	–	–	
0x37 (0x57)	SPMCSR	SPMIE	(RWWSB) ⁵	SIGRD	(RWWSRE) ⁵	BLBSET	PGWRT	PGERS	SPMEN	278
0x36 (0x56)	Reserved	–	–	–	–	–	–	–	–	
0x35 (0x55)	MCUCR	–	BODS ⁽⁶⁾	BODSE ⁽⁶⁾	PUD	–	–	IVSEL	IVCE	45/68/91
0x34 (0x54)	MCUSR	–	–	–	–	WDRF	BORF	EXTRF	PORF	54
0x33 (0x53)	SMCR	–	–	–	–	SM2	SM1	SM0	SE	40
0x32 (0x52)	Reserved	–	–	–	–	–	–	–	–	
0x31 (0x51)	Reserved	–	–	–	–	–	–	–	–	
0x30 (0x50)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	235
0x2F (0x4F)	Reserved	–	–	–	–	–	–	–	–	
0x2E (0x4E)	SPDR	SPI Data Register								169
0x2D (0x4D)	SPSR	SPIF	WCOL	–	–	–	–	–	SPI2X	168
0x2C (0x4C)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	167
0x2B (0x4B)	GPOR2	General Purpose I/O Register 2								26
0x2A (0x4A)	GPOR1	General Purpose I/O Register 1								26
0x29 (0x49)	Reserved	–	–	–	–	–	–	–	–	
0x28 (0x48)	OCR0B	Timer/Counter0 Output Compare Register B								
0x27 (0x47)	OCR0A	Timer/Counter0 Output Compare Register A								
0x26 (0x46)	TCNT0	Timer/Counter0 (8-bit)								
0x25 (0x45)	TCCR0B	FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00	
0x24 (0x44)	TCCR0A	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	
0x23 (0x43)	GTCCR	TSM	–	–	–	–	–	PSRASY	PSRSYNC	140/159
0x22 (0x42)	EEARH	(EEPROM Address Register High Byte) ⁵								22
0x21 (0x41)	EEARL	EEPROM Address Register Low Byte								22
0x20 (0x40)	EEDR	EEPROM Data Register								22
0x1F (0x3F)	EECR	–	–	EEP1	EEP0	EERIE	EEMPE	EEPE	EERE	22
0x1E (0x3E)	GPOR0	General Purpose I/O Register 0								26
0x1D (0x3D)	EIMSK	–	–	–	–	–	–	INT1	INT0	72
0x1C (0x3C)	EIFR	–	–	–	–	–	–	INTF1	INTF0	72
0x1B (0x3B)	PCIFR	–	–	–	–	–	PCIF2	PCIF1	PCIF0	
0x1A (0x3A)	Reserved	–	–	–	–	–	–	–	–	
0x19 (0x39)	Reserved	–	–	–	–	–	–	–	–	
0x18 (0x38)	Reserved	–	–	–	–	–	–	–	–	
0x17 (0x37)	TIFR2	–	–	–	–	–	OCF2B	OCF2A	TOV2	158
0x16 (0x36)	TIFR1	–	–	ICF1	–	–	OCF1B	OCF1A	TOV1	136

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x15 (0x35)	TIFR0	–	–	–	–	–	OCF0B	OCF0A	TOV0	
0x14 (0x34)	Reserved	–	–	–	–	–	–	–	–	
0x13 (0x33)	Reserved	–	–	–	–	–	–	–	–	
0x12 (0x32)	Reserved	–	–	–	–	–	–	–	–	
0x11 (0x31)	Reserved	–	–	–	–	–	–	–	–	
0x10 (0x30)	Reserved	–	–	–	–	–	–	–	–	
0x0F (0x2F)	Reserved	–	–	–	–	–	–	–	–	
0x0E (0x2E)	Reserved	–	–	–	–	–	–	–	–	
0x0D (0x2D)	Reserved	–	–	–	–	–	–	–	–	
0x0C (0x2C)	Reserved	–	–	–	–	–	–	–	–	
0x0B (0x2B)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	92
0x0A (0x2A)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	92
0x09 (0x29)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	92
0x08 (0x28)	PORTC	–	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	91
0x07 (0x27)	DDRC	–	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	91
0x06 (0x26)	PINC	–	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	92
0x05 (0x25)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	91
0x04 (0x24)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	91
0x03 (0x23)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	91
0x02 (0x22)	Reserved	–	–	–	–	–	–	–	–	
0x01 (0x21)	Reserved	–	–	–	–	–	–	–	–	
0x0 (0x20)	Reserved	–	–	–	–	–	–	–	–	

- Note:**
1. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
 2. I/O Registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions.
 3. Some of the Status Flags are cleared by writing a logical one to them. Note that, unlike most other AVRs, the CBI and SBI instructions will only operate on the specified bit, and can therefore be used on registers containing such Status Flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.
 4. When using the I/O specific commands IN and OUT, the I/O addresses 0x00 - 0x3F must be used. When addressing I/O Registers as data space using LD and ST instructions, 0x20 must be added to these addresses. The ATmega48A/PA/88A/PA/168A/PA/328/P is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.
 5. Only valid for ATmega88A/88PA/168A/168PA/328/328P.
 6. BODS and BODSE only available for picoPower devices ATmega48PA/88PA/168PA/328P

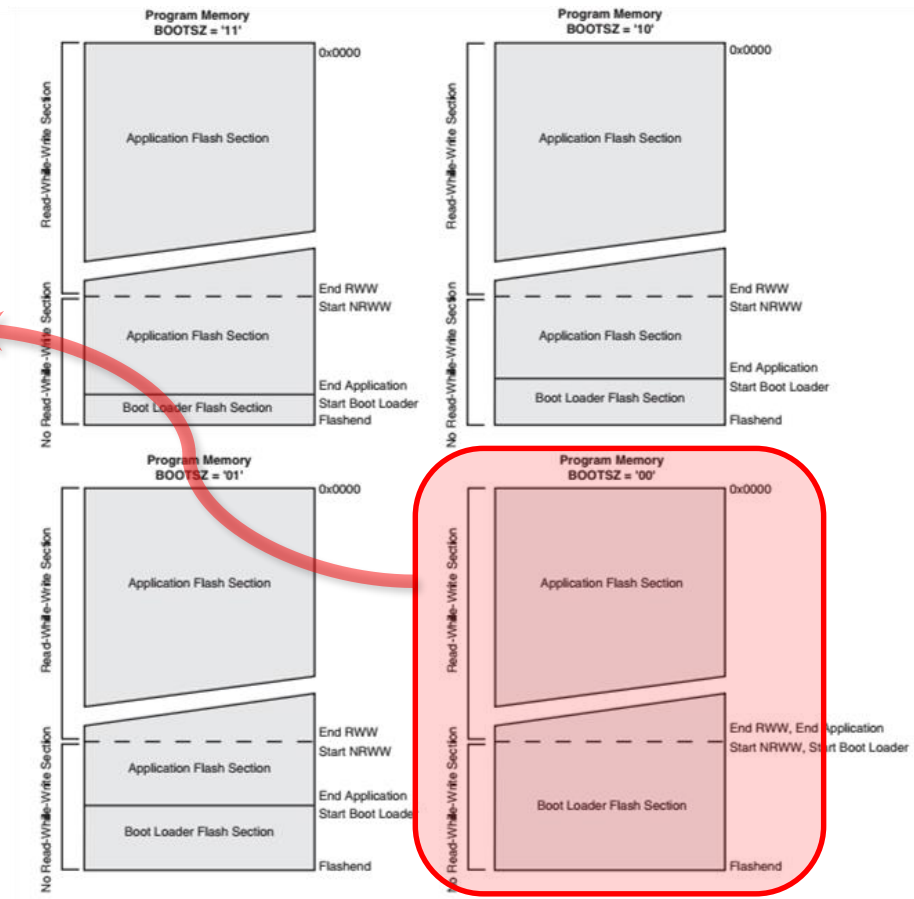
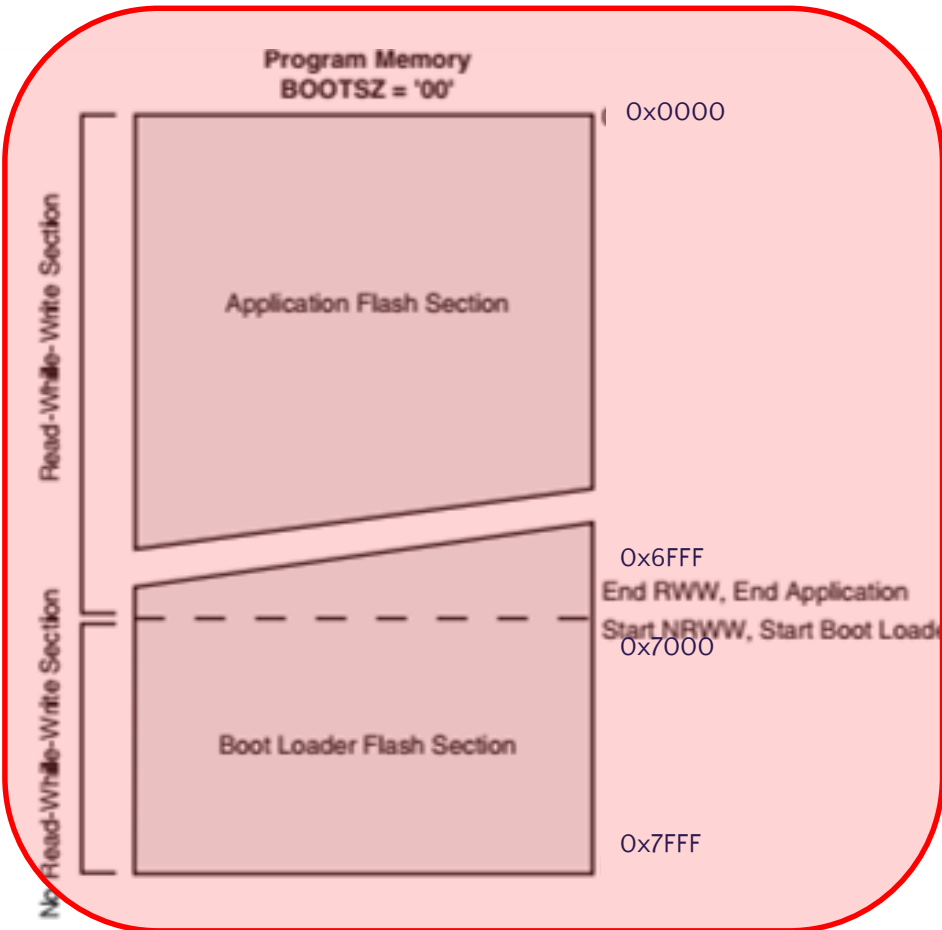
3. 퓨즈비트

- 레지스터가 제어장치에 달린 수십개의 스위치 뭉치라면
퓨즈비트는 그 전원장치에 붙은 “관계자외 금지” 조작스위치

Table 27-13. Boot Size Configuration, ATmega328/328P

BOOTSZ1	BOOTSZ0	Boot Size	Pages	Application Flash Section	Boot Loader Flash Section	End Application Section	Boot Reset Address (Start Boot Loader Section)
1	1	256 words	4	0x0000 - 0x3EFF	0x3F00 - 0x3FFF	0x3EFF	0x3F00
1	0	512 words	8	0x0000 - 0x3DFF	0x3E00 - 0x3FFF	0x3DFF	0x3E00
0	1	1024 words	16	0x0000 - 0x3BFF	0x3C00 - 0x3FFF	0x3BFF	0x3C00
0	0	2048 words	32	0x0000 - 0x37FF	0x3800 - 0x3FFF	0x37FF	0x3800

Note: The different BOOTSZ Fuse configurations are shown in [Figure 27-2 on page 266](#).



4. 부트로더

- 처음 시작 : 롬을 램으로 복사
- 지금 : 다양한 시작 프로그램 기능

- A. 아두이노에 전원 공급
- B. 레지스터 살핌 (IVSEL , IVCE 설정 확인)
- C. 정해진 부트로더 주소로 이동
(우노 0x3800(words) 0x7000(bytes))
- D. 프로그램 실행
- E. 부트로더 실행 후 정해진 주소로 이동
(우노 0x0000)
- F. 순차적 실행



아두이노용 C 언어

- 하드웨어 구성

신호등 하드웨어 사용

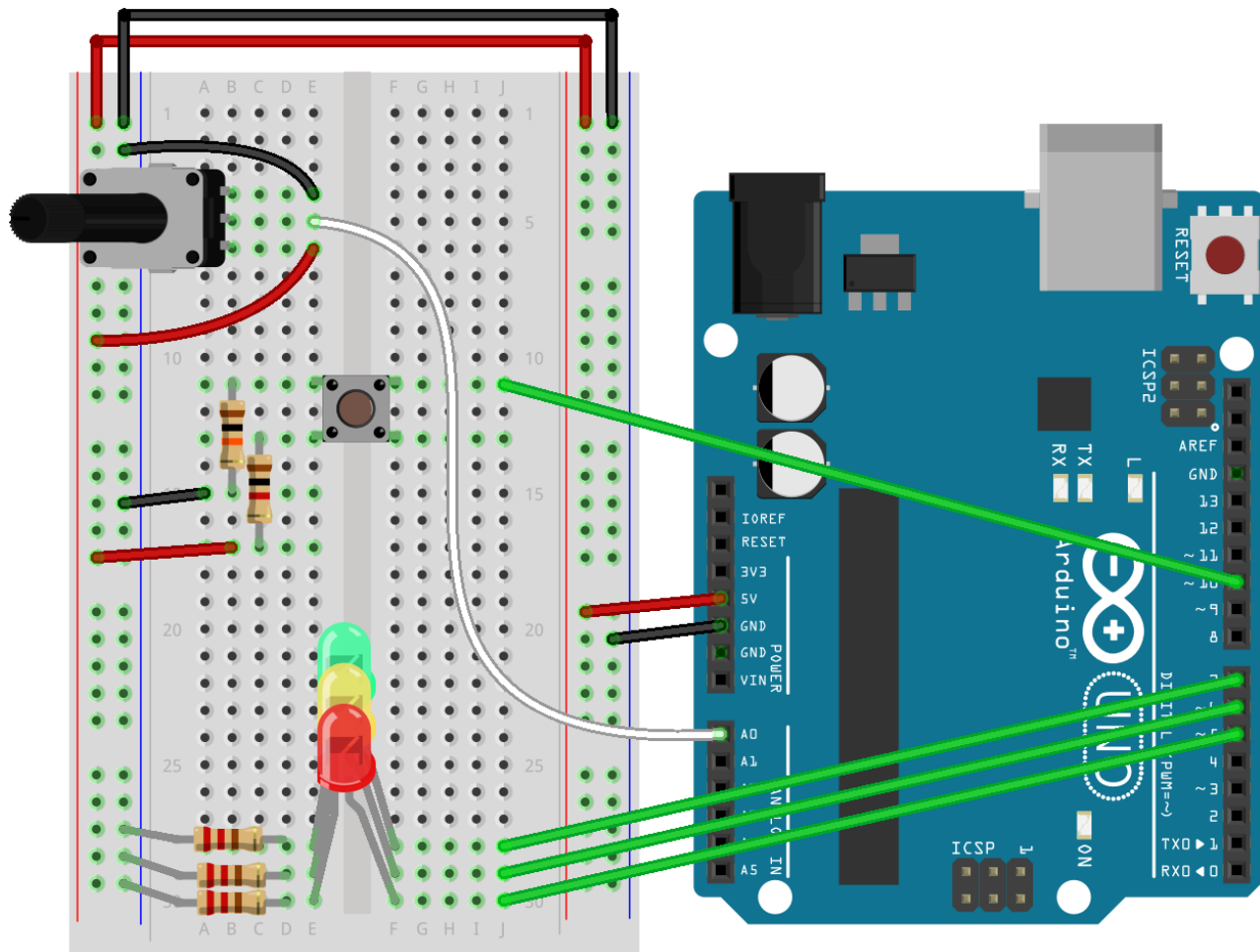
디지털출력 4,5,6

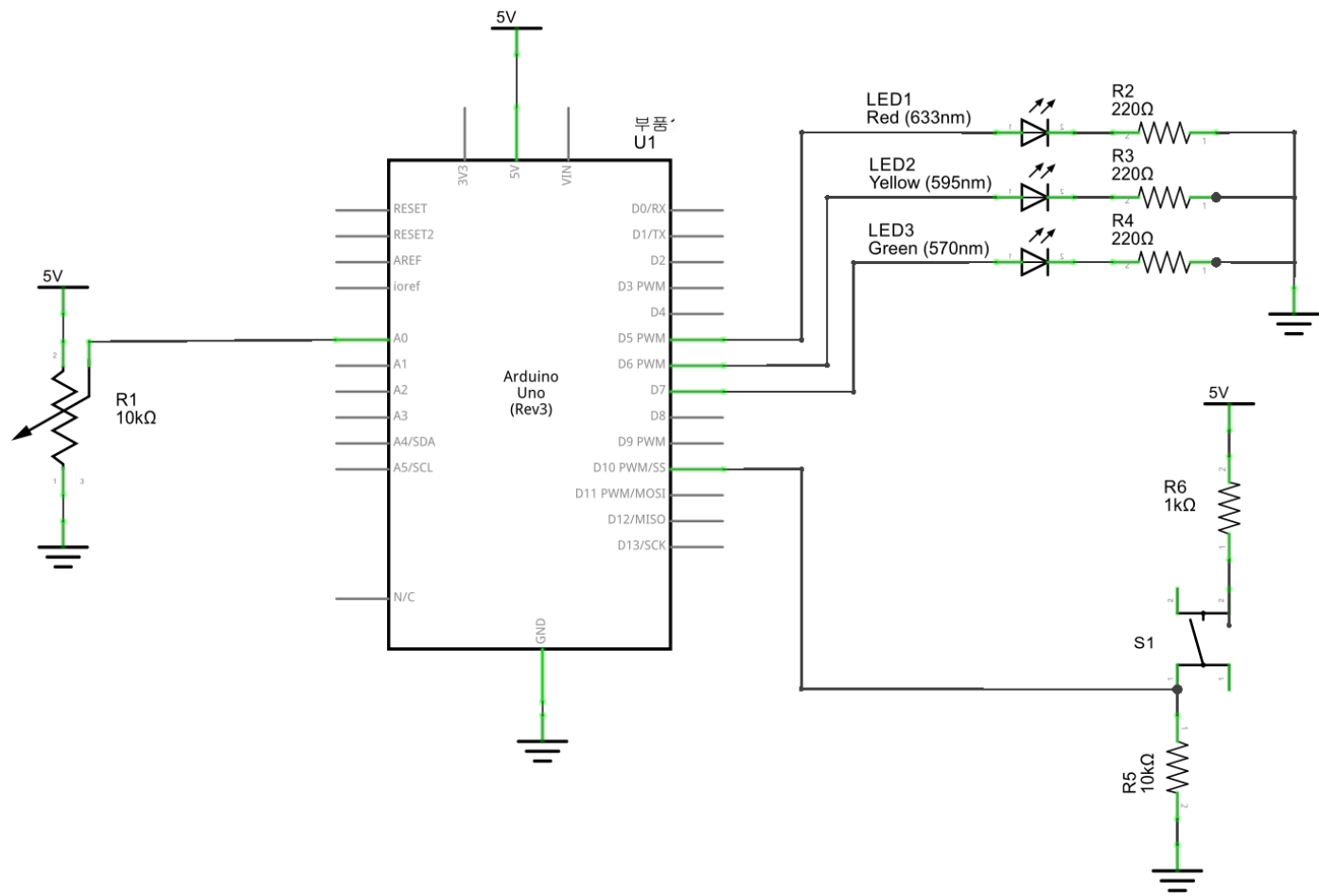
디지털입력 10

아날로그입력 A0

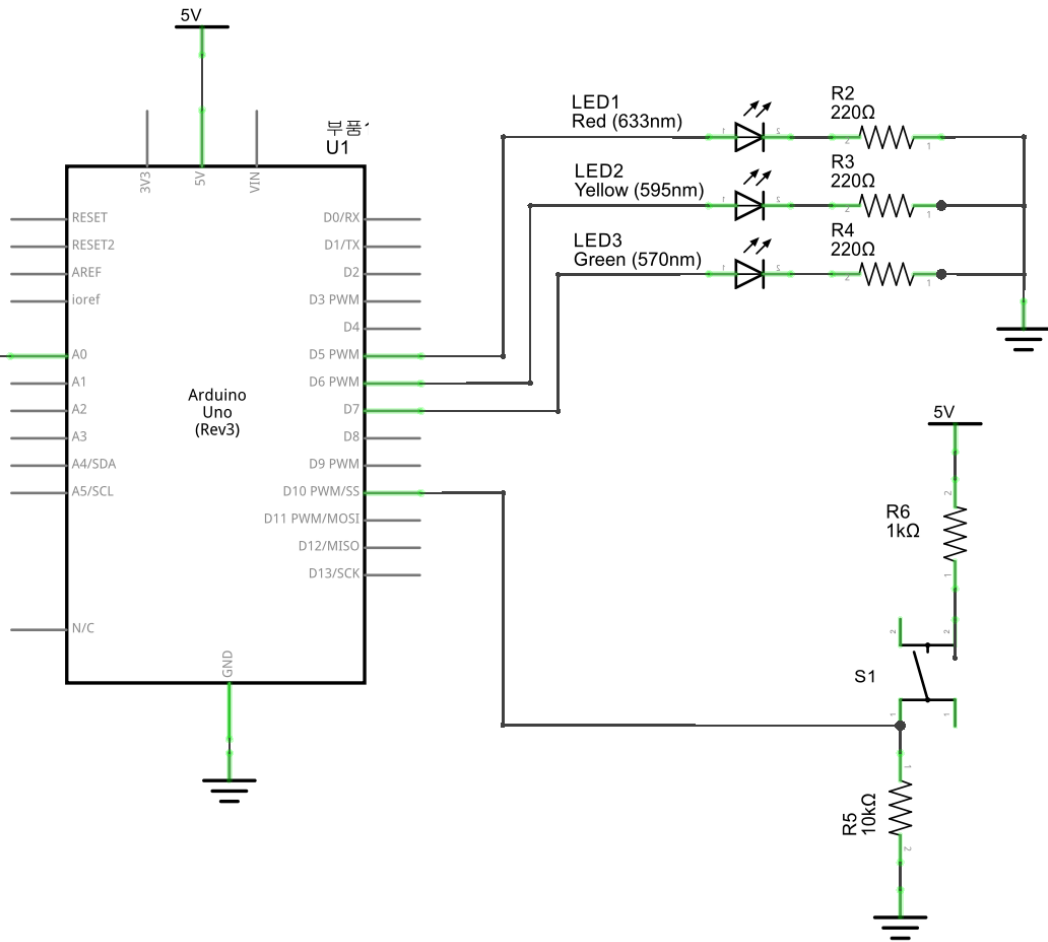
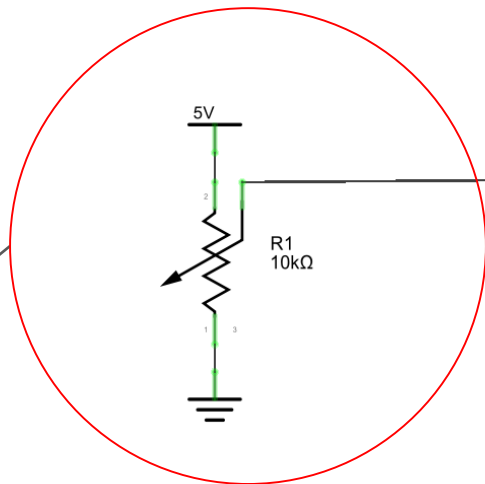
아날로그출력 5,6

시리얼통신

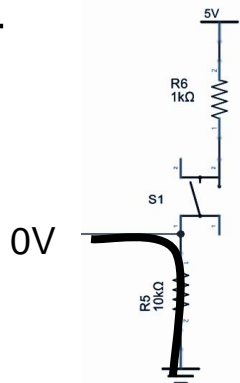




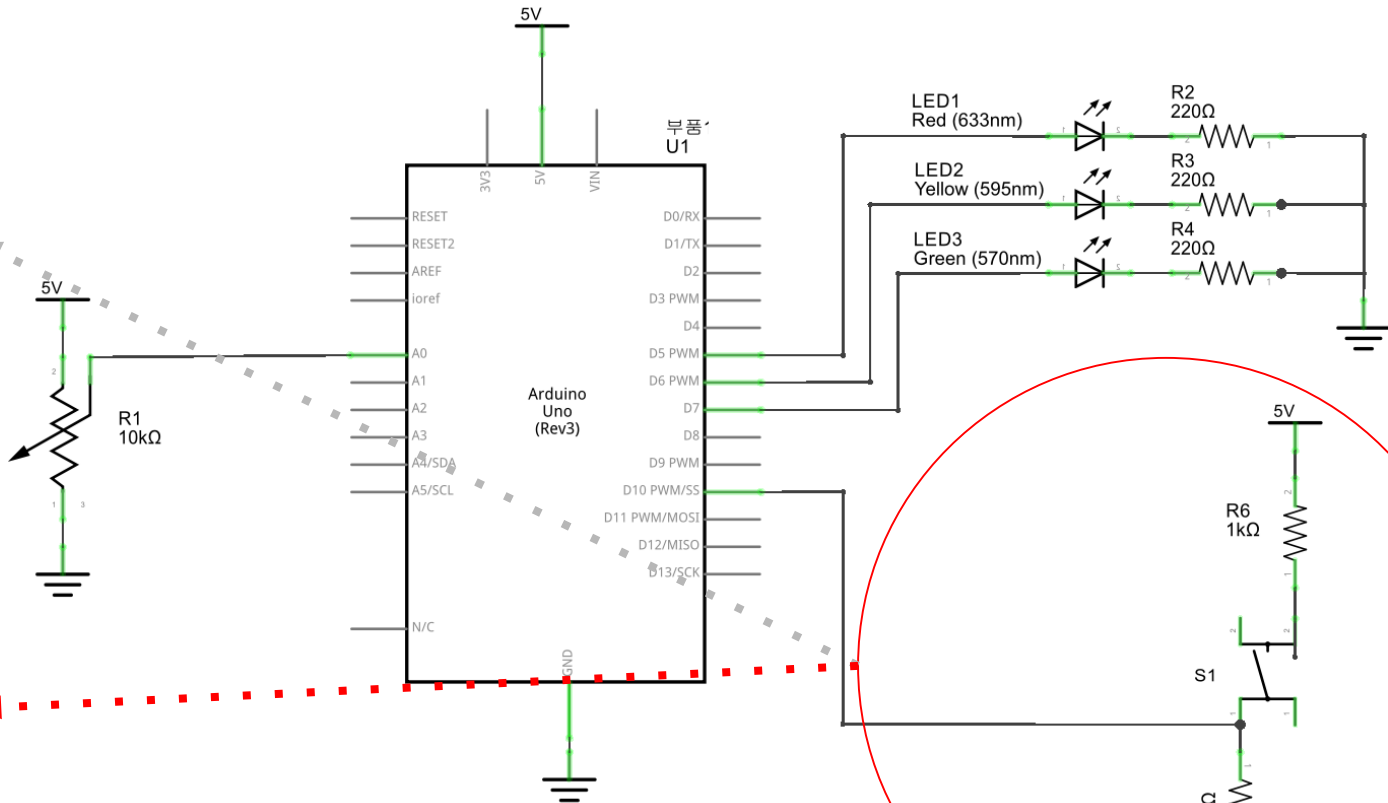
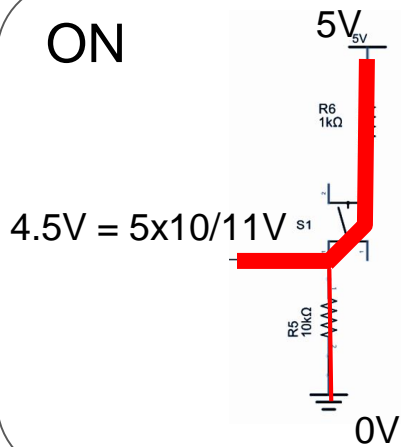
아날로그
입력



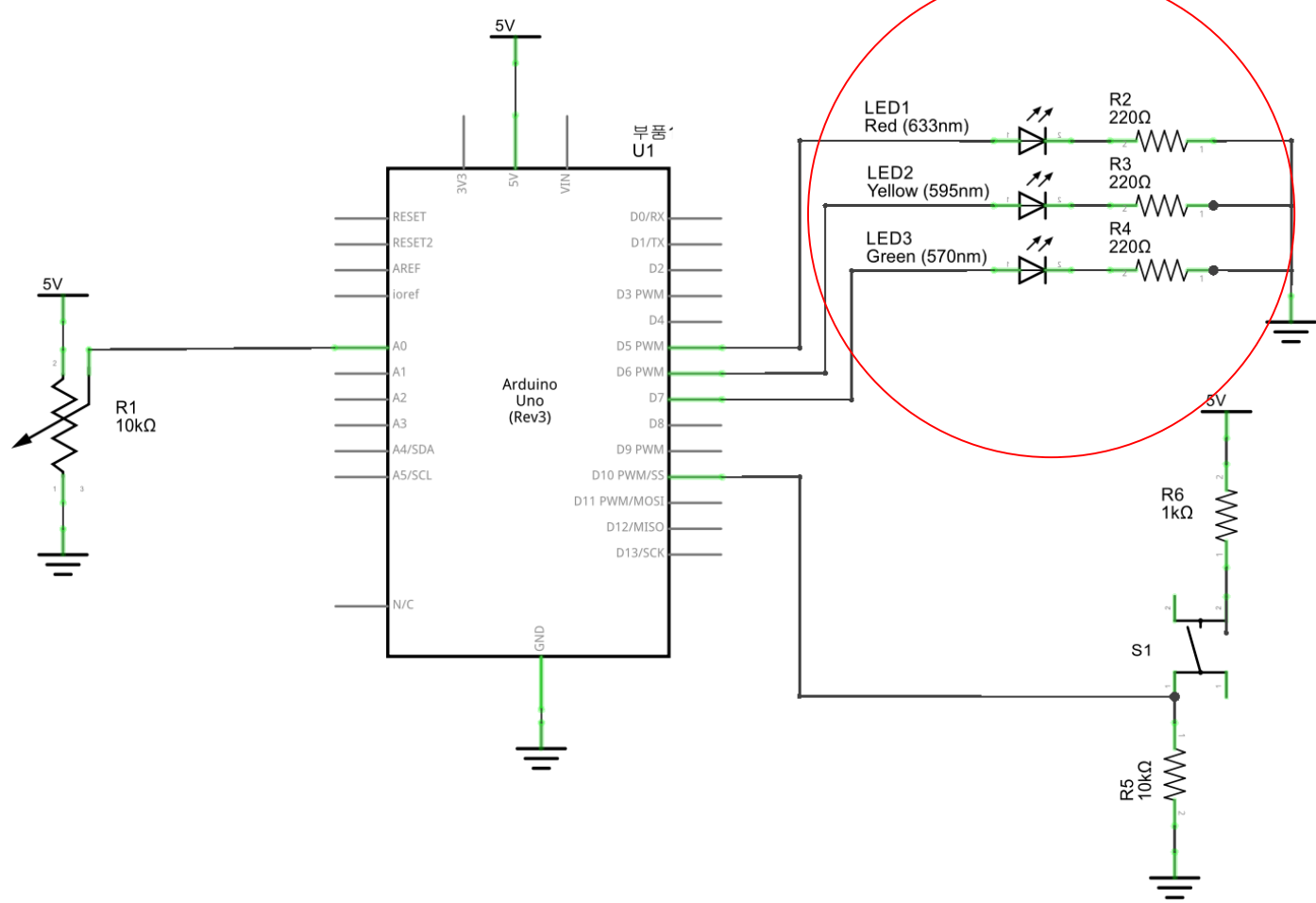
OFF

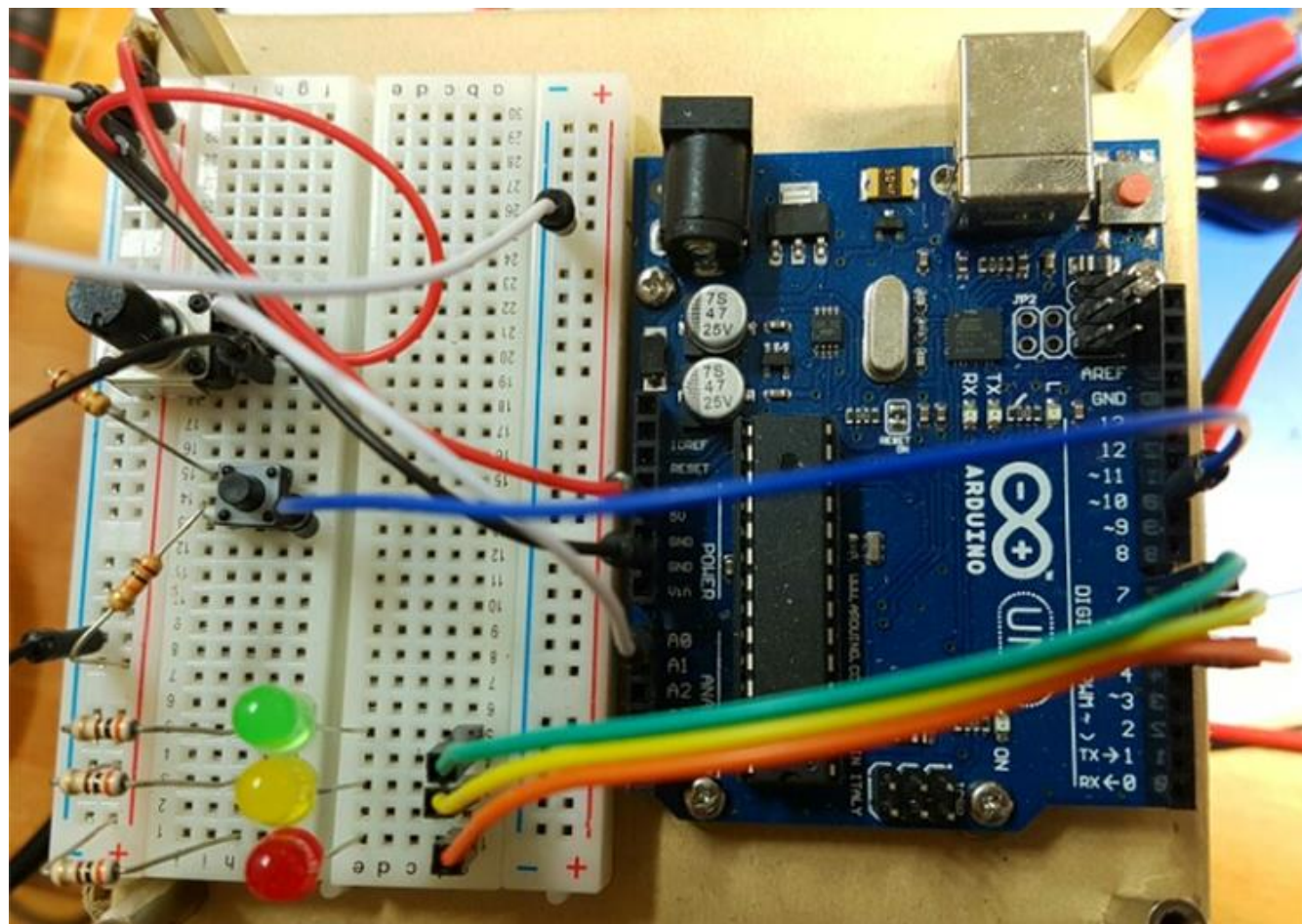


ON



디지털입력





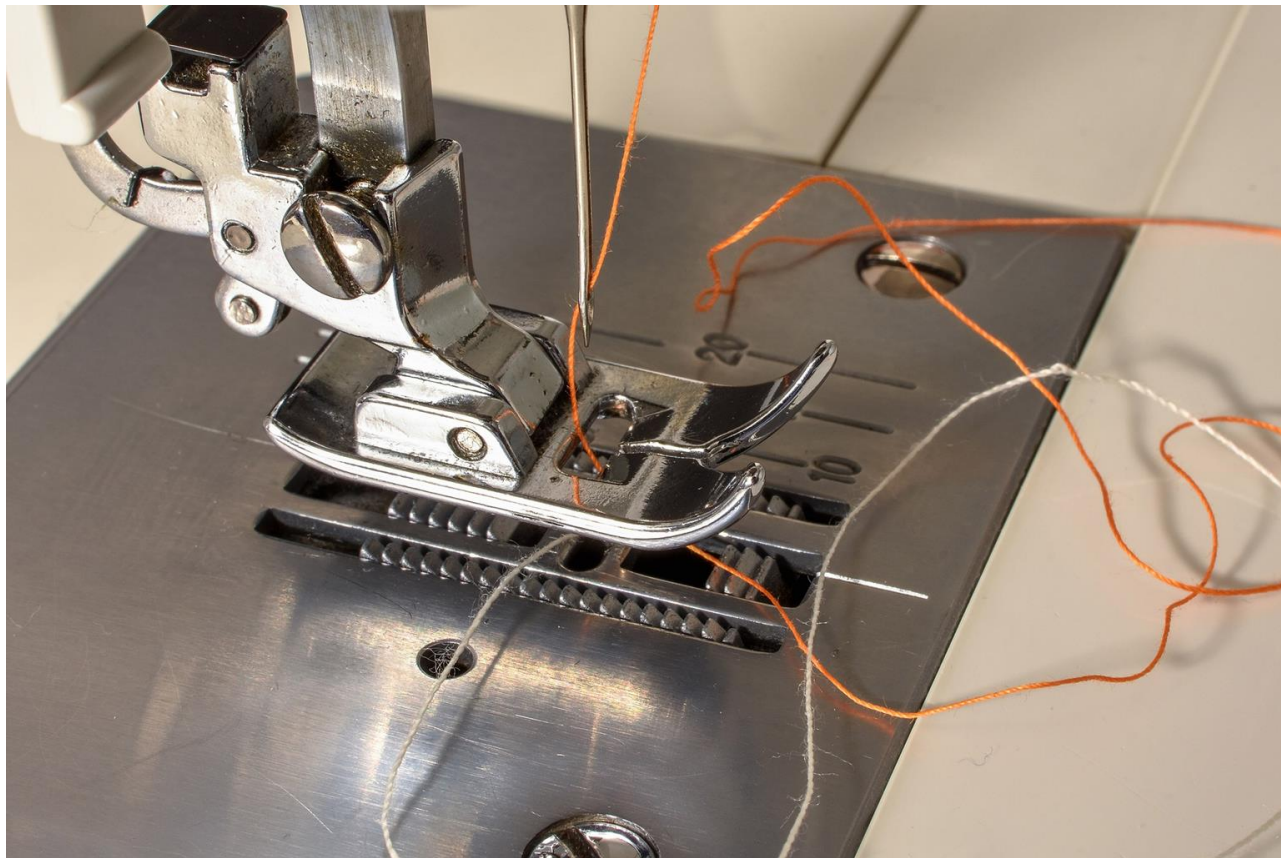


아두이노용 C 언어

– setup()

setup()

1회 실행



```
void setup()  
{  
    pinMode(5, OUTPUT);  
}
```

```
void setup()  
{  
    pinMode(5, OUTPUT);  
    pinMode(6, OUTPUT);  
    pinMode(7, OUTPUT);  
    pinMode(10, INPUT);  
}
```

pinMode(pin, mode)

mode :

INPUT

OUTPUT

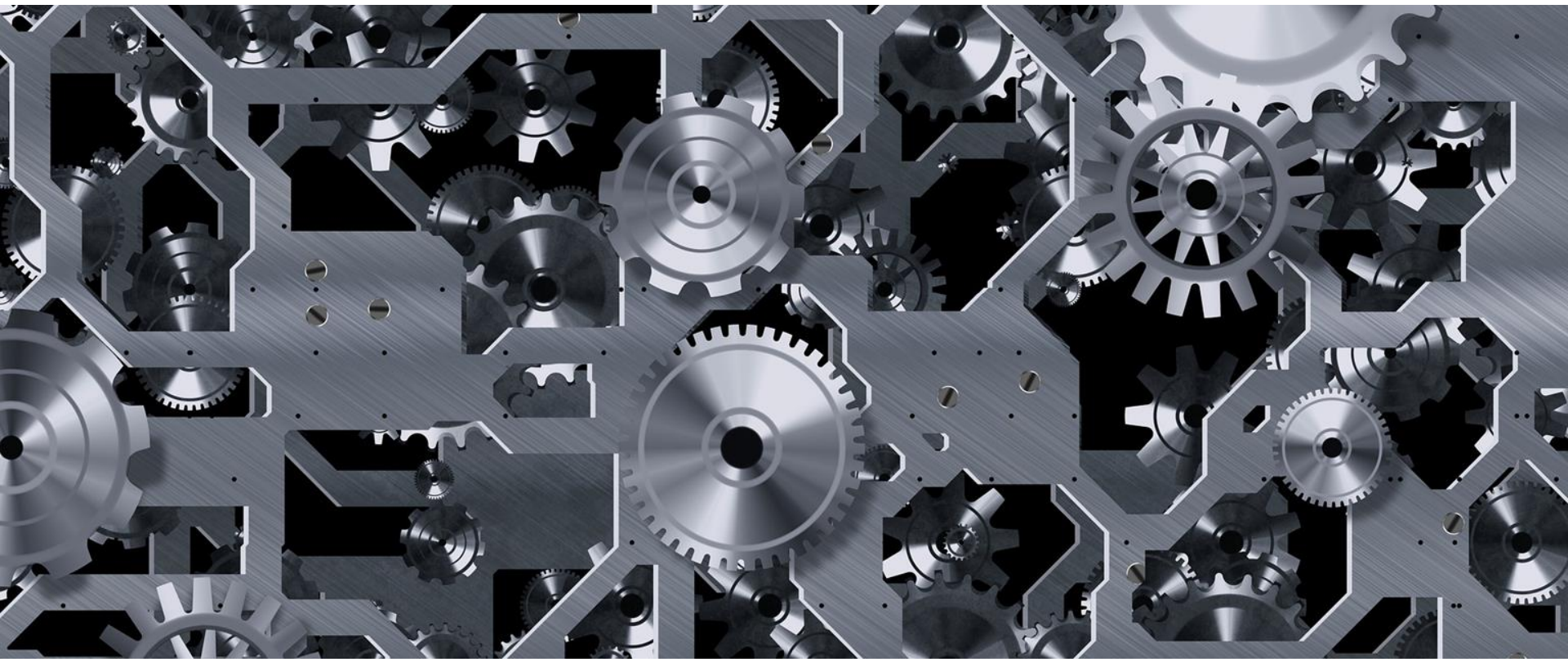
INPUT_PULLUP



아두이노용 C 언어

– loop()

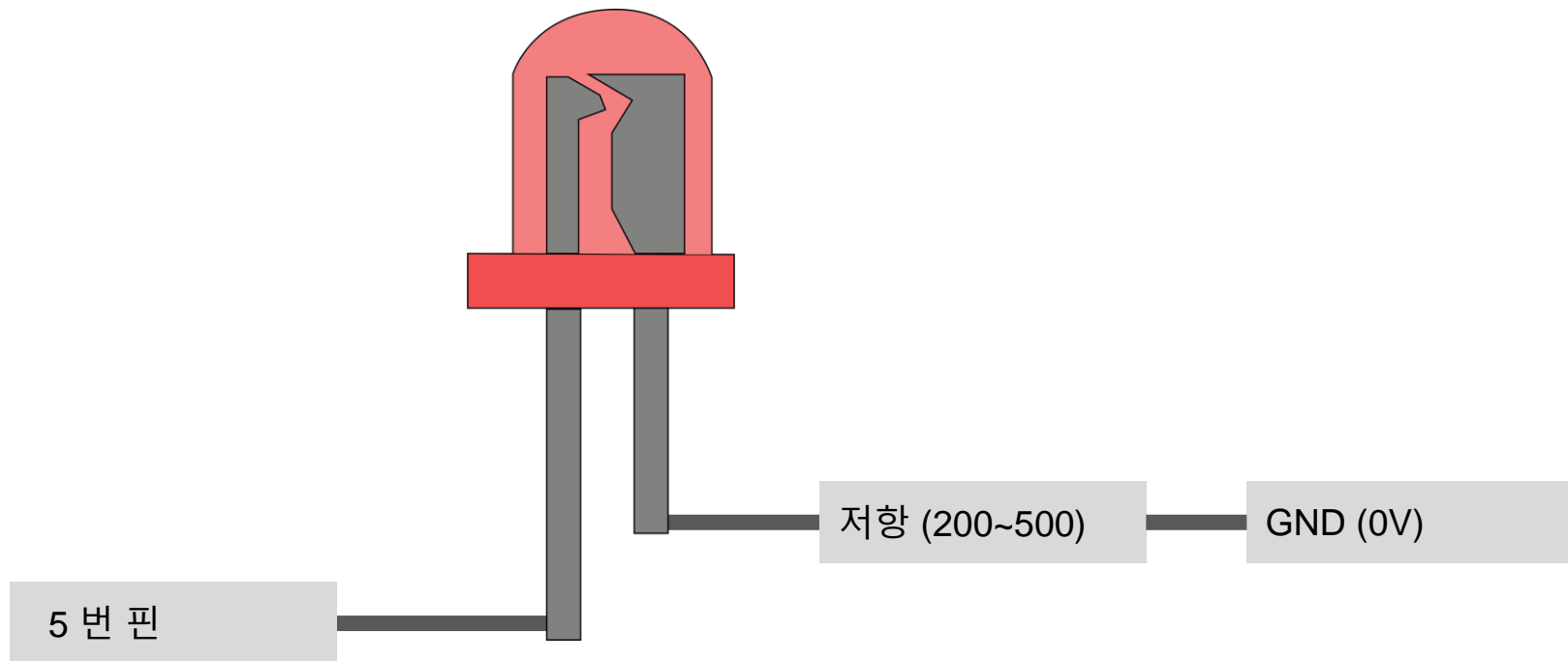
loop()




```
void loop()  
{  
    digitalWrite(5, HIGH);  
}
```

```
void setup()  
{  
    digitalWrite(5, HIGH);  
}
```

```
void loop()  
{  
    digitalWrite(5, HIGH);  
}
```



```
void setup() {  
    pinMode(5, OUTPUT);  
}
```

```
void loop() {  
    digitalWrite(5, 1);  
    delay(1000);  
    digitalWrite(5, 0);  
    delay(1000);  
}
```

1 == HIGH

0 == LOW

```
void setup() {  
    pinMode(5, OUTPUT);  
}
```

```
void loop() {  
    digitalWrite(5, HIGH);  
    delay(1000);  
    digitalWrite(5, LOW);  
    delay(1000);  
}
```




아두이노용 C 언어

- 함수

$$X_{1/2} = \frac{\dots}{2a}$$



$$X^2 + px + q = 0$$



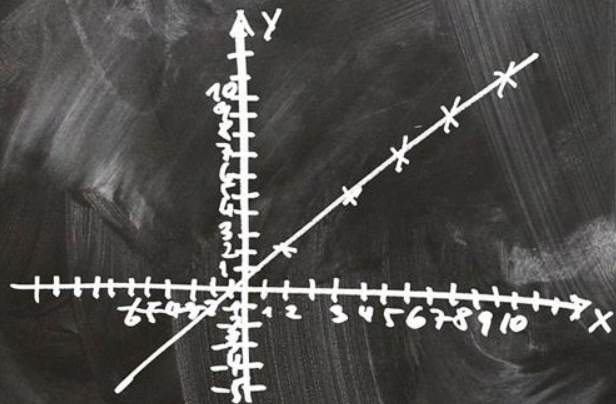
$$X_{1/2} = -\frac{p}{2} \pm \sqrt{\left(\frac{p}{2}\right)^2 - q}$$

$$X = 6 - 2y$$

$$X + a = b$$

$$f(x) = \tan x$$

$$f(x) = \sin x$$



함수 만드는 법

[(1)나가는값] [함수이름] (들어오는 값들)

{

처리를 위한 명령문들...;

}

(1)

나가는 값이 없을땐 void

나가는 값이 int 타입이면 int

함수 만들기 예제

```
void laught()  
{  
    digitalWrite(5, HIGH);  
}
```

```
int add2(int a, int b)  
{  
    int c = a + b;  
    return c;  
}
```

함수 사용하기

함수이름(함수로 전달하는 값들);

단, 사용하기 전에 만들어져 있어야 한다.

함수 사용 예제

```
void laught()  
{  
    digitalWrite(5, HIGH);  
}
```

만들고

```
int add2(int a, int b)  
{  
    int c = a + b;  
    return c;  
}
```

```
void loop(){  
    laught();  
}
```

사용한다

```
void loop(){  
    int sum = add2(5, 3);  
    Serial.print(sum);  
}
```

```

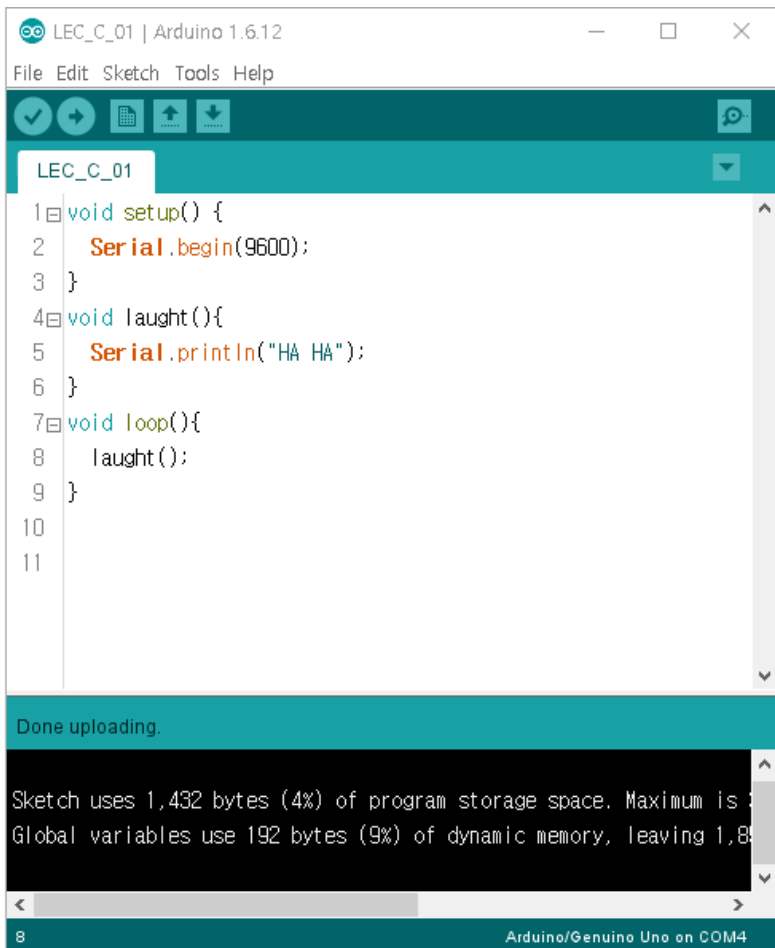
void setup() {
    Serial.begin(9600);
}

void laught(){
    Serial.println("HA HA");
}

void loop() {
    laught();
}

```

	프로그램의 실행 순서는 다음과 같습니다.
1	setup()을 실행합니다. 시리얼통신을 하도록 초기화합니다.
2	loop()을 실행시킵니다.
3	loop() 안에 있는 laught()를 실행시킵니다. # laught() 안에 있는 Serial.println("HA HA")을 실행합니다. # laught() 함수 끝까지 실행 한 다음 종료합니다. # laught() 가 종료되면 loop() 함수 6번 줄의 끝으로 갑니다. # loop() 가 종료됩니다.
4	다시 loop() 가 시작됩니다. : 3번으로 갑니다. (# 표시된 줄을 계속 반복합니다.)





아두이노용 C 언어

- { } 괄호의 의미

{

}

{

의미있는 그룹

}

함수에서의 { }

```
int add2(int a, int b)
```

```
{
```

```
    int c = a + b ;
```

```
    return c;
```

```
}
```

{ } 안의 변수

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

시작하는 {

{ } 블록 안 변수
그 안에서만 사용

끝나는 }

{ } 안의 { } 의 변수

```
void loop() {
```

```
    int a = 1;
```

```
    int b = 2;
```

```
    {
```

```
        int c = 3;
```

```
        int d = a+b+c;
```

```
    }
```

```
}
```

시작하는 {

끝나는 }

{ } & 변수의 범위

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

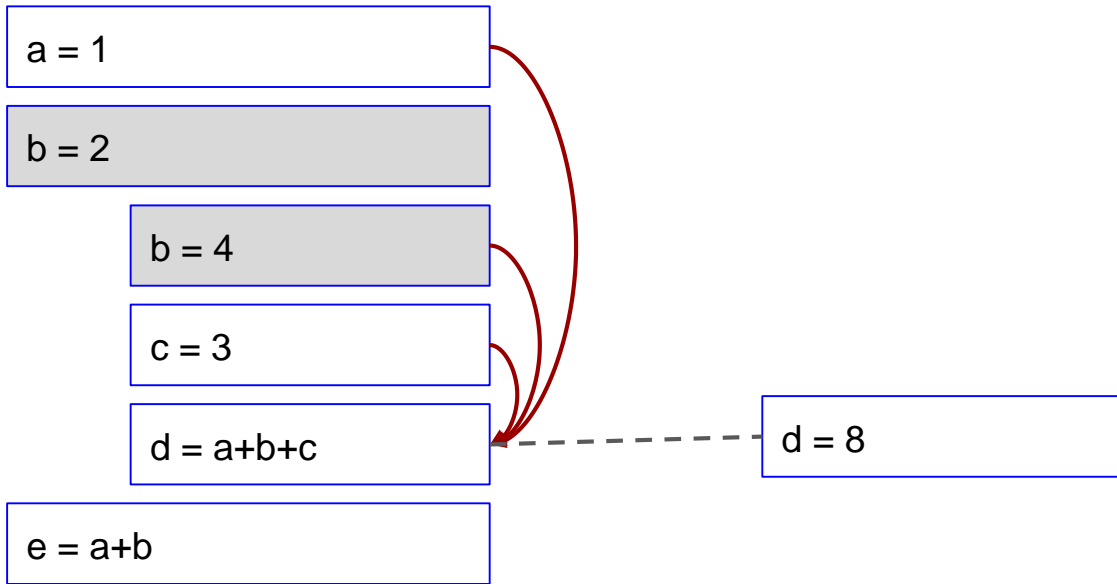
에러가 나는 곳

[Question] d 와 e 의 값을 얼마?

```
void loop() {  
  int a = 1;  
  int b = 2;  
  {  
    int b = 4;  
    int c = 3;  
    int d = a+b+c;  
    Serial.println(d);  
  }  
  int e = a+b;  
  Serial.println(e);  
}
```

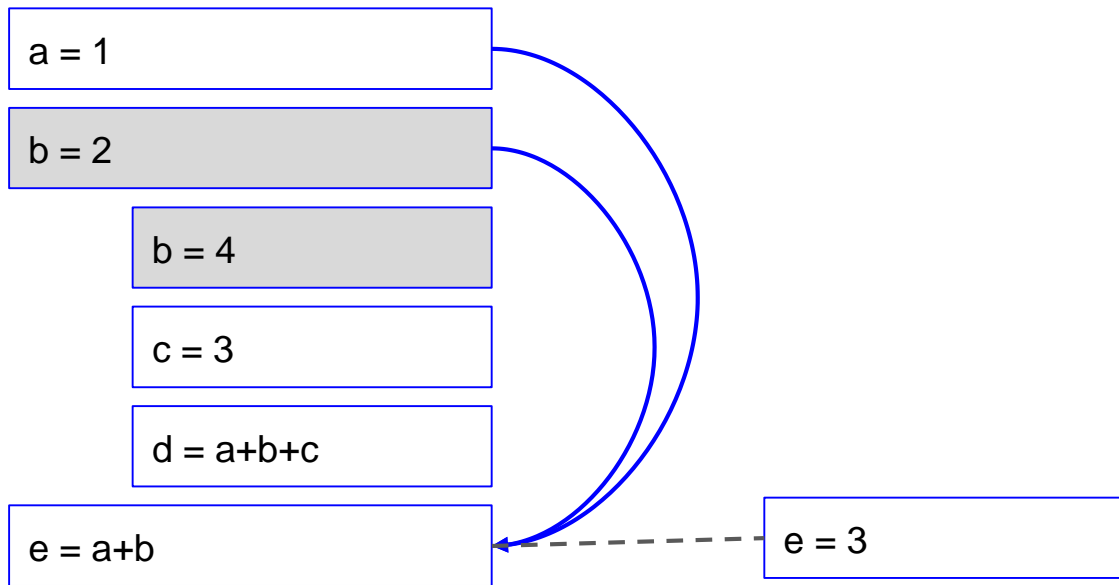

[Question] d 와 e 의 값을 얼마?

```
void loop() {  
  int a = 1;  
  int b = 2;  
  {  
    int b = 4;  
    int c = 3;  
    int d = a+b+c;  
    Serial.println(d);  
  }  
  int e = a+b;  
  Serial.println(e);  
}
```



[Question] d 와 e 의 값을 얼마?

```
void loop() {  
  int a = 1;  
  int b = 2;  
  {  
    int b = 4;  
    int c = 3;  
    int d = a+b+c;  
    Serial.println(d);  
  }  
  int e = a+b;  
  Serial.println(e);  
}
```



[Question] d 와 e 의 값을 얼마?

```
void loop() {  
  int a = 1;  
  int b = 2;  
  {  
    int b = 4;  
    int c = 3;  
    int d = a+b+c;  
    Serial.println(d);  
  }  
  int e = a+b;  
  Serial.println(e);  
}
```

