



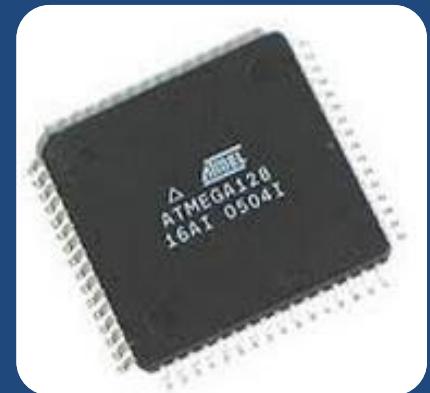
Hardware/Iot Hacking

AVR 프로그래밍

mongii@grayhash

마이크로 컨트롤러 소개

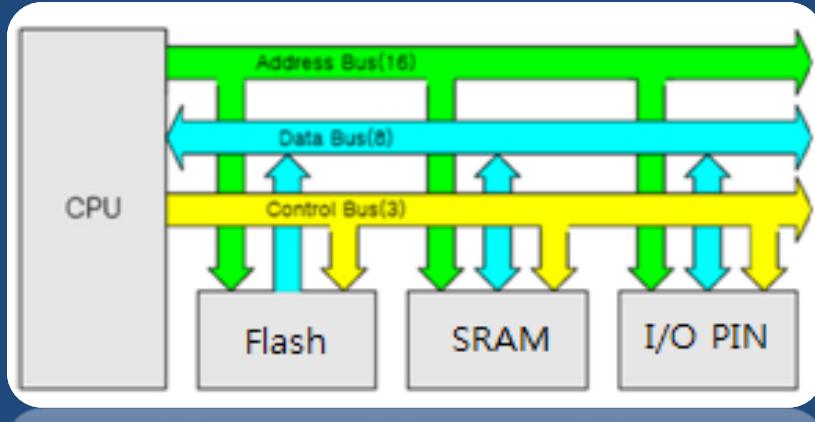
- MCU = Micro Controller Unit
- 한마디로 “작은 CPU 혹은 작은 컴퓨터”
- 특수목적을 수행하는 소형화 된 CPU
- 주변장치를 추가해나가며 기능확장(Control) 가능
- 주로 C언어를 이용하여 프로그래밍
- 칩 내부에 RAM과 ROM 등을 포함
 - System on a chip (SoC)



주요 MCU들

- AVR
 - 아트멜(Atmel)사에서 개발
 - 마이크로 컨트롤러 초급자들에게 인기
 - 개발환경 구축이 용이하며, 관련 자료가 많음
- 8051
 - 인텔에서 만든 MCU
 - 초창기 MCU 중 하나
 - 역시 초급자들에게 인기가 많음
- PIC
 - 마이크로칩사에서 만든 MCU
 - 산업용으로 특히 많이 사용됨

MCU의 구조



- 하나의 칩에 모든 기본기능이 내장됨
 - CPU, FLASH, SRAM, ROM, I/O
- MCU 하나만으로 동작 가능
- MCU 자체로서 하나의 작은 컴퓨터

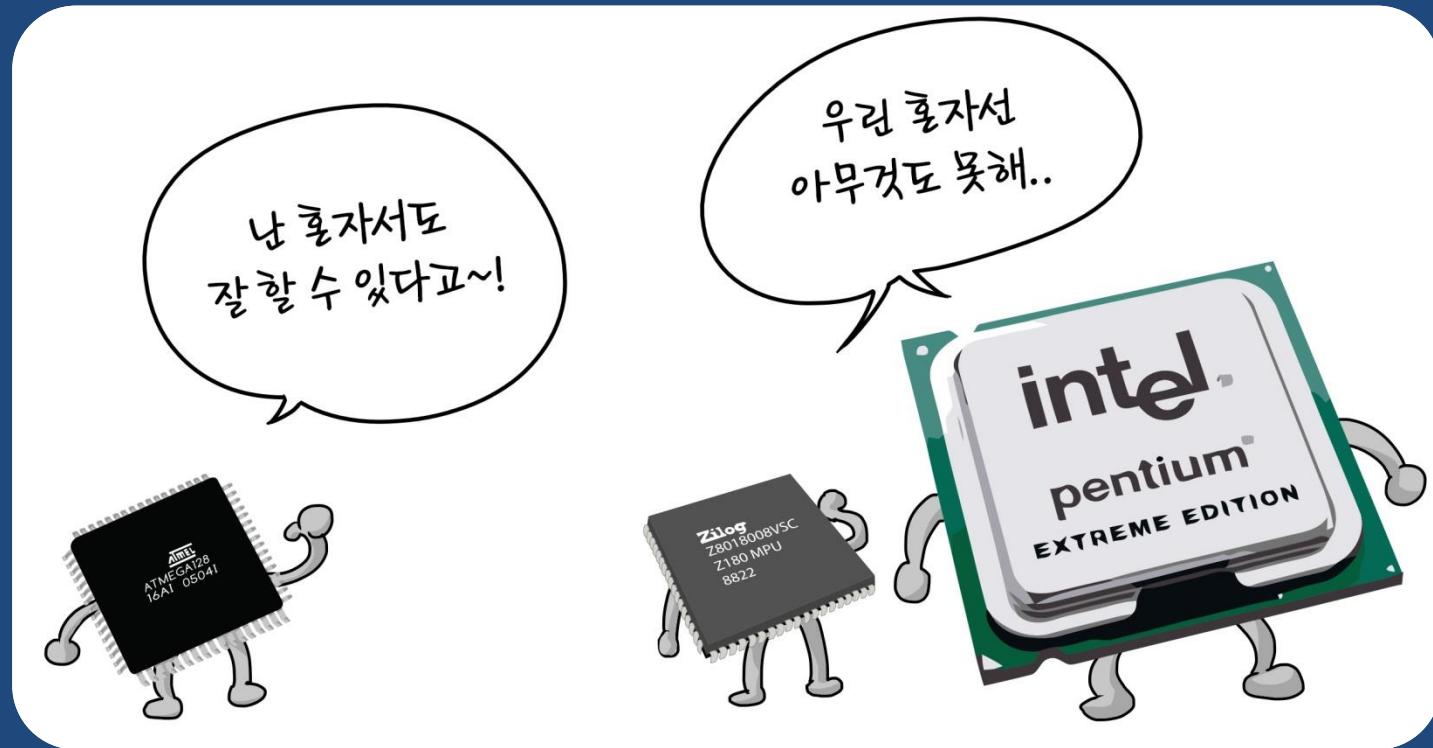
MCU와 CPU의 차이

MCU	CPU
통합형 칩셋	산술, 논리연산 특화
혼자 동작 가능	혼자 동작 불가능
임베디드 장비에 사용	일반 PC에 사용

- MCU : 통합형, 혼자 동작 가능
 - CPU : 산술/논리 특화, 혼자 동작 불가능
- * 결국 MCU 안에는 CPU가 포함되어 있음

MCU와 CPU 구별법

- 단일칩으로 작동할 수 있느냐 없느냐의 차이



AVR Programming의 세계로!

AVR의 장점

- AVR은 하드웨어 프로그래밍 분야의 기본이다!
 - 가장 먼저, 그리고 필수적으로 학습하기를 추천
 - Low Level의 지식들을 학습할 수 있음
- 풍부한 학습 자료
- 무료 개발 툴
- 하드웨어 구매 비용 저렴
- 가격대비 만족스러운 성능

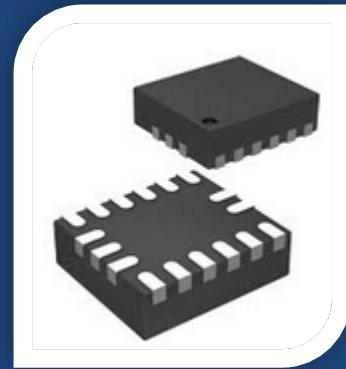
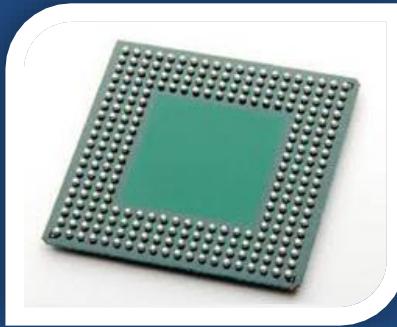
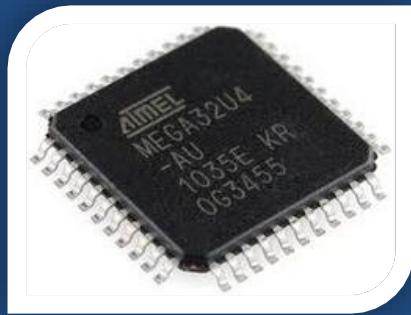
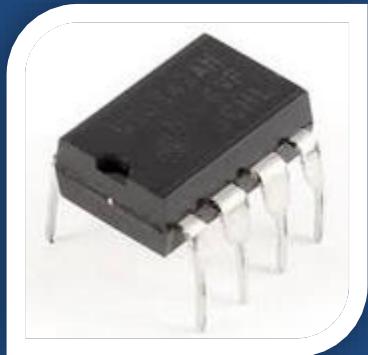
AVR의 종류들

- 다양한 구매자의 요구를 충족하기 위한
매우 다양한 종류의 제품들을 판매
 - 참고 : <http://www.devicemart.co.kr>

제품군	제품명	프로그램용량	EEPROM	SRAM	패키지	속도	작동전압	기타
AVR Tiny	ATtiny12-8SI	1KB	64B	-	SOP8	0~8MHz	4.0~5.5V	
	ATtiny12V-1SI	1KB	64B	-	SOP8	0~1MHz	1.8~5.5V	
	ATtiny13-20PI	1KB	64B	64B	PDIP8	0~20MHz	2.7~5.5V	
	ATtiny13V-10SI	1KB	64B	64B	SOP8	0~10MHz	1.8~5.5V	
	ATtiny2313-20MU	2KB	128B	128B	MLF20	0~20MHz	2.7~5.5V	
	ATtiny2313-20PU	2KB	128B	128B	PDIP20	0~20MHz	2.7~5.5V	
	ATtiny2313-20SU	2KB	128B	128B	SOP20	0~20MHz	2.7~5.5V	
	ATtiny2313V-10PU	2KB	128B	128B	PDIP20	0~10MHz	1.8~5.5V	
	ATtiny2313V-10SI/U	2KB	128B	128B	SOP20	0~10MHz	1.8~5.5V	
	ATtiny24-20PU	2KB	128B	128B	PDIP14	0~10MHz	2.7~5.5V	

패키지 타입에 따른 분류

- 칩이 PCB에 연결되는 방식을 의미
- 순서대로 : DIP, SOIC, TSOP, TQFP, BGA, MLF



Atmega128A 소개

- 8비트 마이크로 컨트롤러
- 최대 16Mhz의 속도
- 128Kbytes의 플래시 메모리 내장
- 4Kbytes의 SRAM 내장
- 4Kbytes의 EEPROM 내장
- 133개의 instruction 제공
- 2.7~5.5v 전압에서 동작



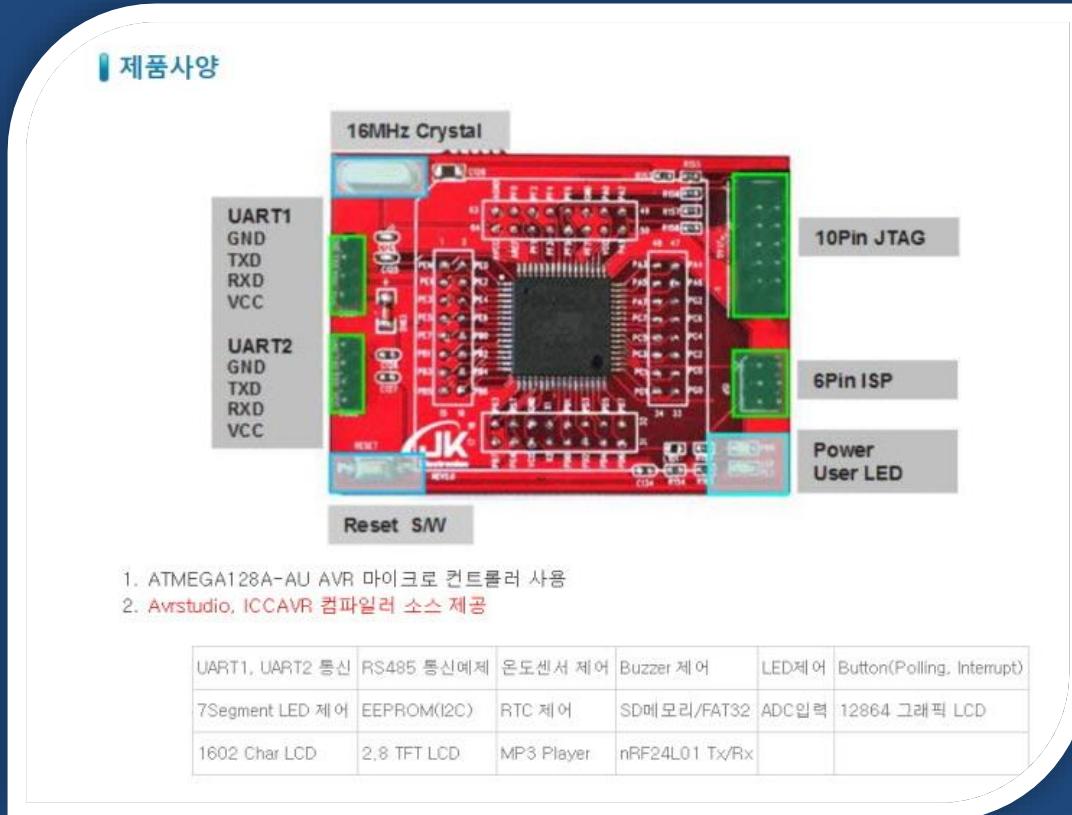
Atmega128 VS Atmega128A

- 동작 전압의 차이
 - Atmega128 : 4.2~5.5V
 - Atmega128A : 2.7~5.5V
- 일부 핀 배열의 차이
- 출력 전류 크기의 차이

실습용 개발보드 소개

- Atmega128 Rabbit Board

- Atmega128A MCU 기반
- <http://itempage3.auction.co.kr/DetailView.aspx?ItemNo=B420146899>



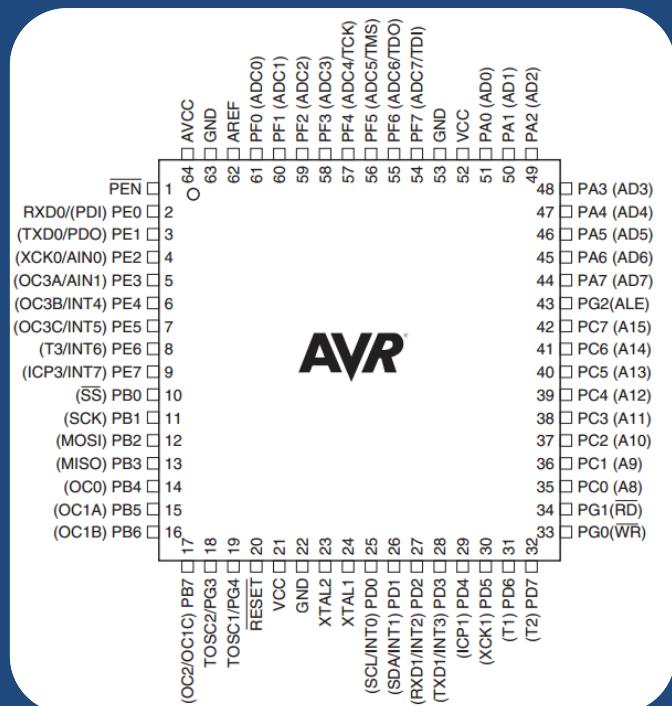
AVR 개발 도구 소개

- Atmel-STUDIO
 - Atmel 개발, 무료, 통합개발환경 제공
- CodeVision
 - 예제코드 자동 생성 가능
 - 유료, 약 30만원
- IAR EWAVR
 - 유료, 수백만원



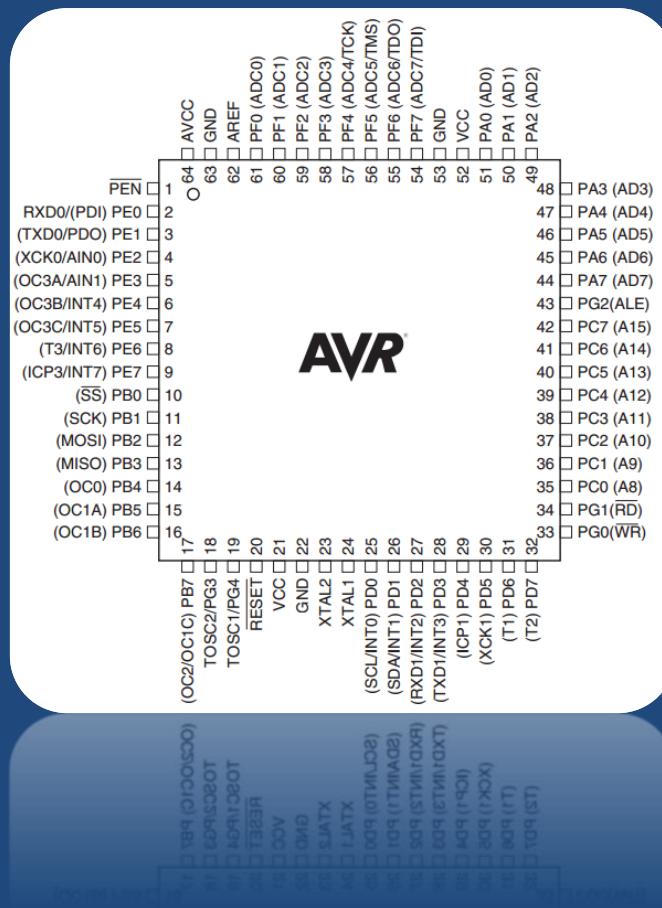
입출력 포트의 이해

- 하드웨어 분야에서 있어서 포트(Port)란, 전기적 신호 데이터가 오가는 통로를 의미
- Atmega128A
 - 총 64개의 핀 노출
 - 총 7개의 포트
 - Port A : PA0 ~ PA7
 - Port B : PB0 ~ PB7
 - Port C : PC0 ~ PC7
 - Port D : PD0 ~ PD7
 - Port E : PE0 ~ PE7
 - Port F : PF0 ~ PF7
 - Port G : PG0 ~ PG4



입출력 포트의 이해

- 범용포트와 특수포트
 - 범용포트 : 프로그래머 마음대로 사용 가능
 - 특수포트 : 정해진 기능 수행
 - RX/TX
 - JTAG
 - ISP
 - ADC



레지스터의 이해

- 레지스터란 MCU 내부에 존재하는 소형 메모리로서, MCU의 상태를 제어하거나 혹은 데이터 이동의 매개체 역할을 함
- 범용 입출력 관련 레지스터들
 - PORTA, DDRA, ...

```
1 : #include <avr/io.h>
2 :
3 : int main(void)
4 : {
5 :     DDRA = 0xff;
6 :     PORTA = 0xff;
7 :
8 :     return 0;
9 : }
```

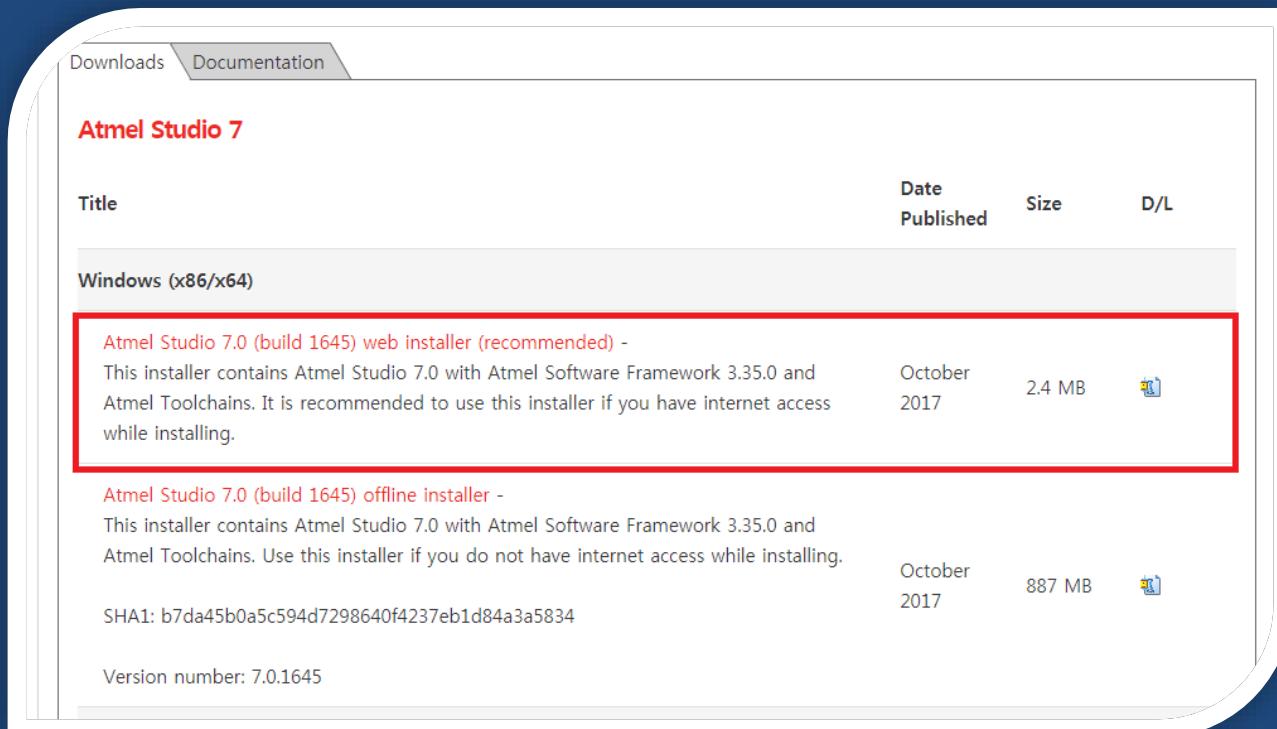
잠시 후에 해볼 것

- MCU에 프로그램 집어넣기 (downloading)
- LED 제어
- 모터 제어
- 부저 스피커 제어
- UART 통신
- 온도센서 제어
- 7 Segment 제어
- Switch 입력 처리
- 등등..

개발 도구 설치

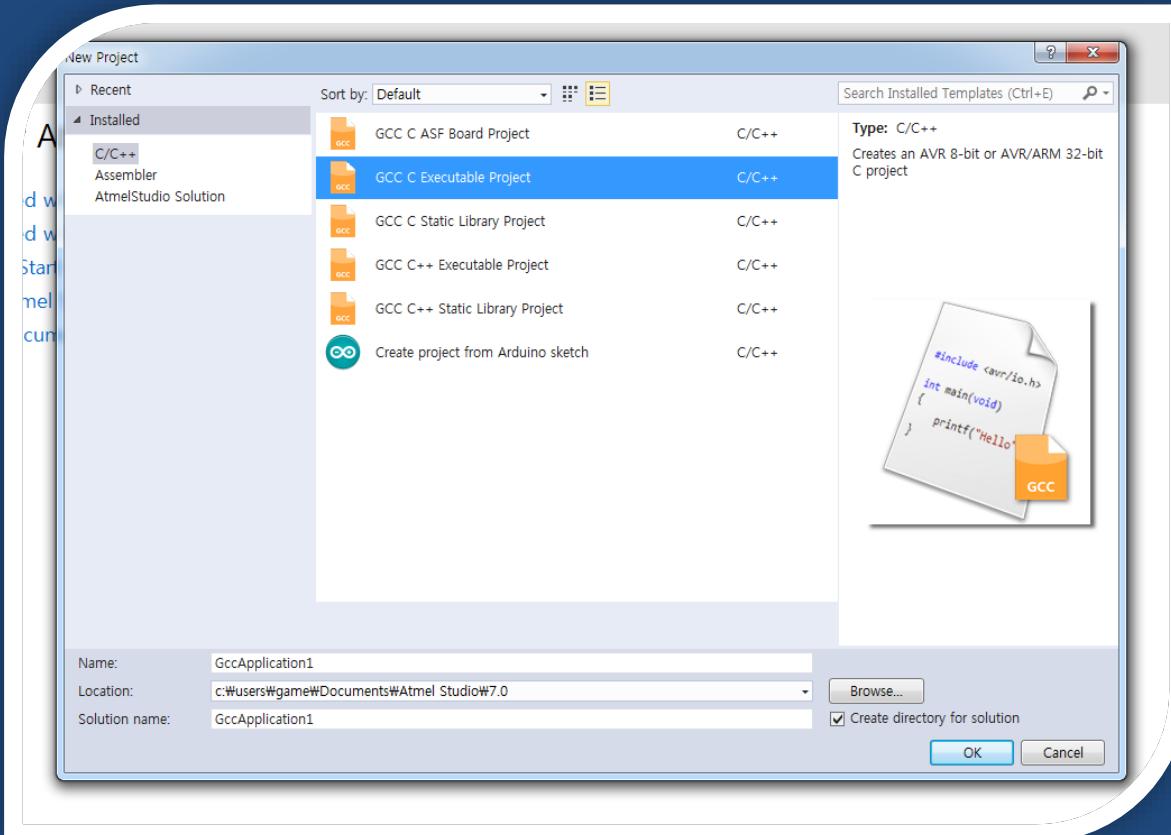
- Atmel Studio 7

- <https://www.microchip.com/avr-support/atmel-studio-7>
- <http://studio.download.atmel.com/7.0.1645/as-installer-7.0.1645-web.exe> EXE 바로 다운로드



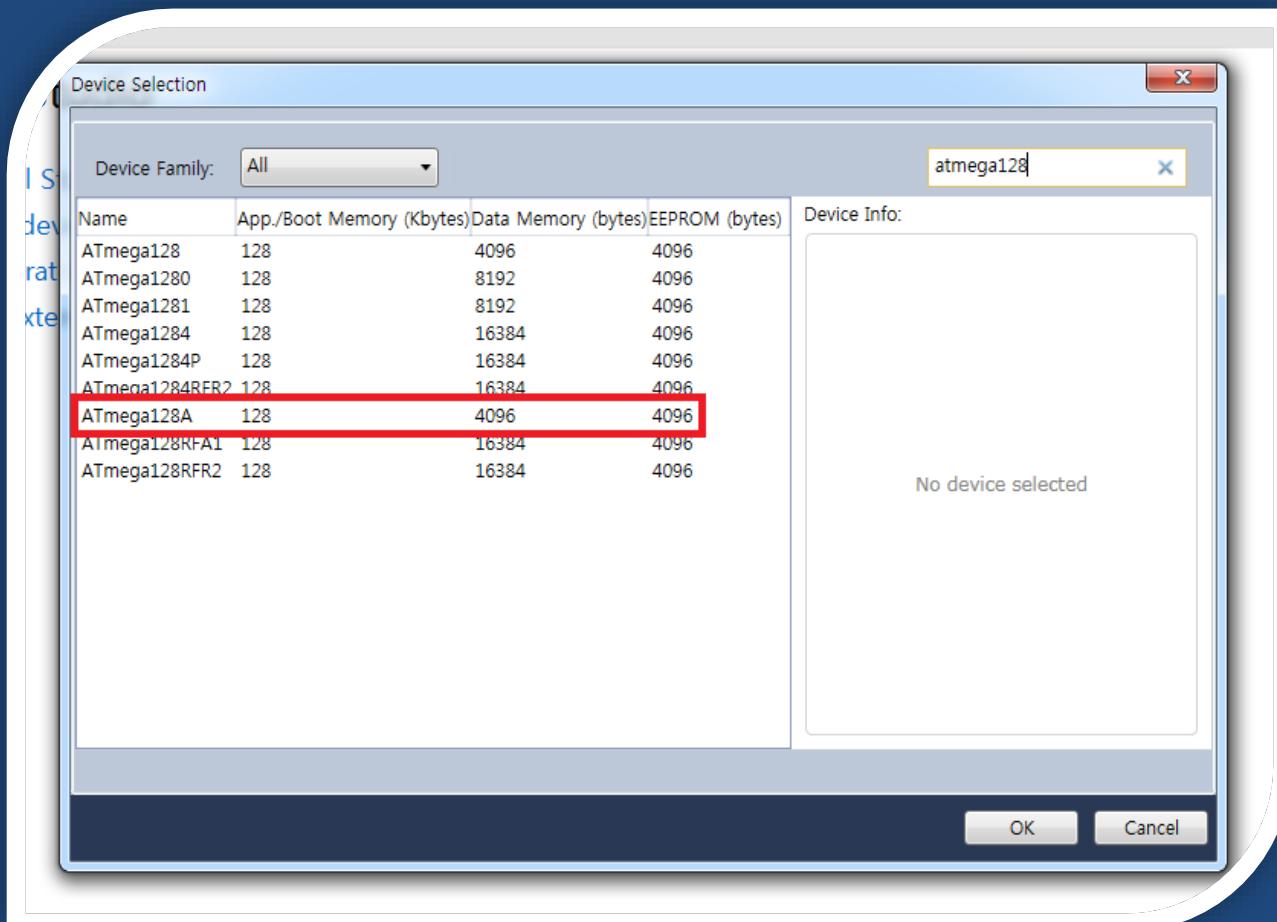
프로젝트 생성

- Atmel Studio 7 실행
- File -> New -> Project -> GCC C Executable Project



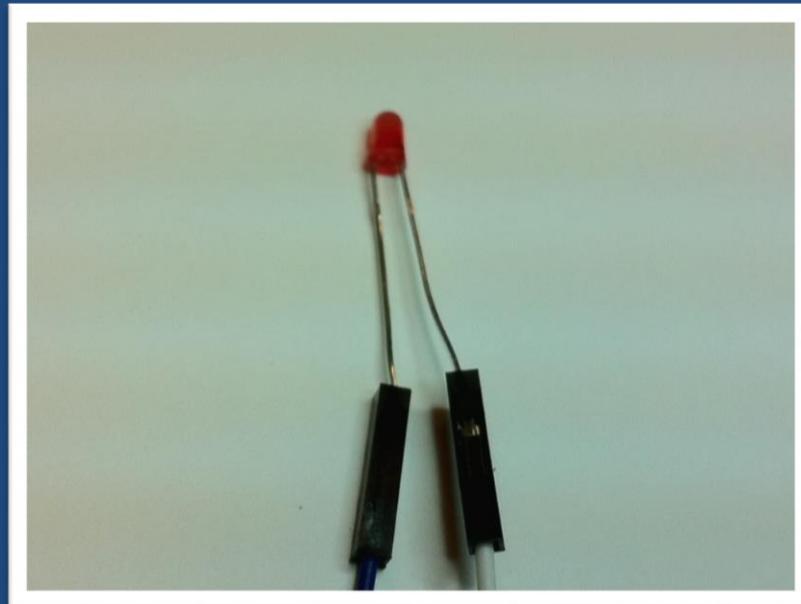
프로젝트 생성

- ATmega128A 선택



LED 연결

- LED의 긴다리 = 양극 => PORTA에 입력
- LED의 짧은다리 = 음극 => GND에 입력



Hello, AVR World!

- 소스코드 입력

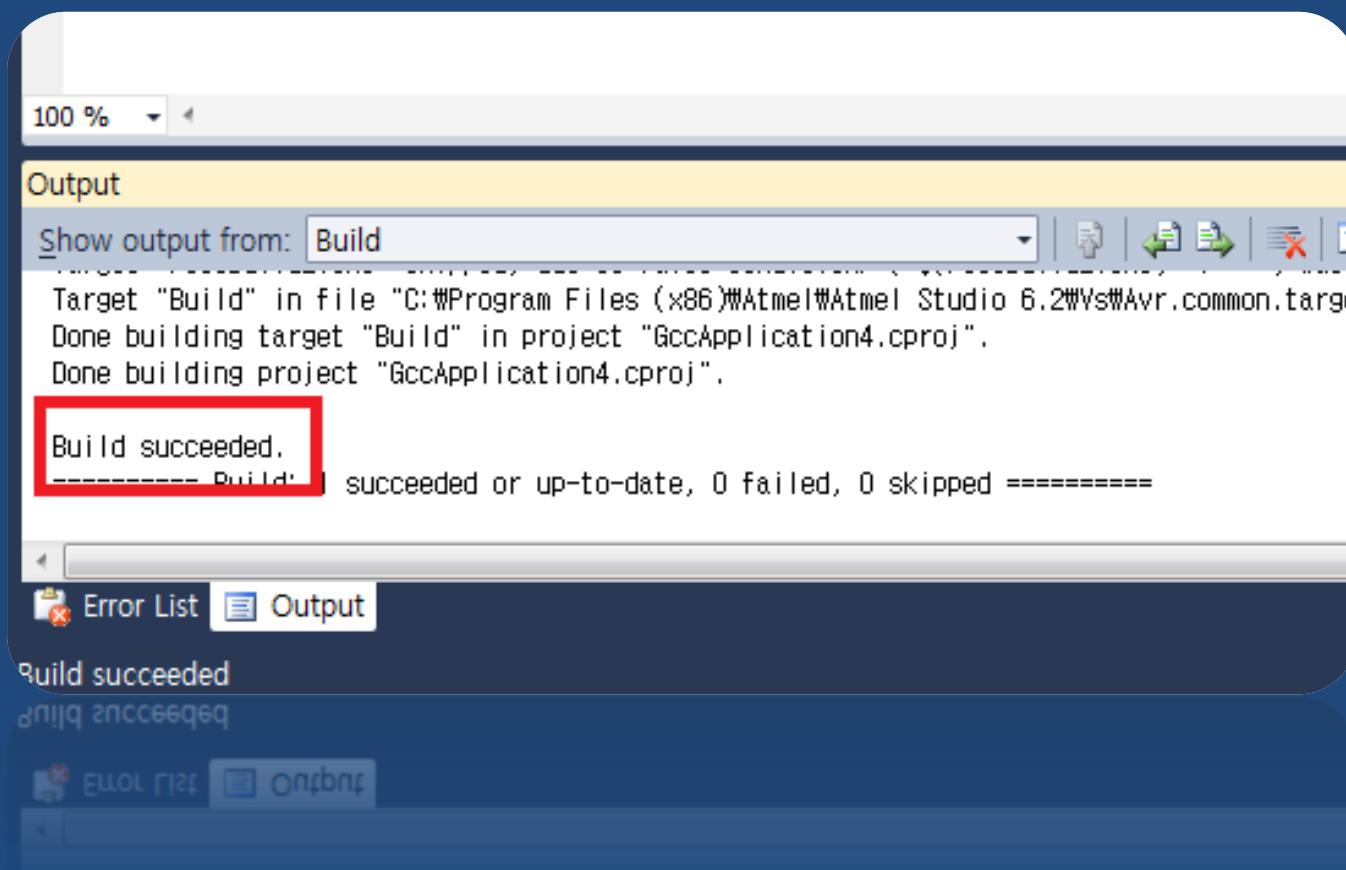
```
#include <avr/io.h>

int main(void)
{
    DDRA = 0xff;
    PORTA = 0xff;

    return 0;
}
```

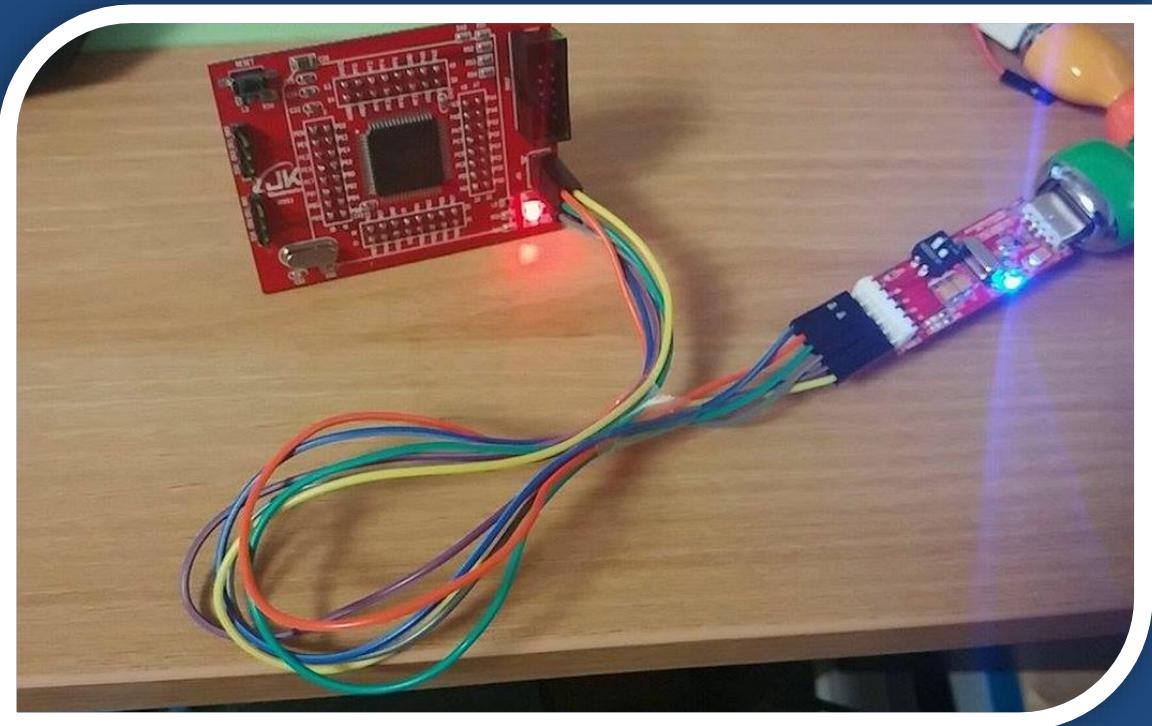
컴파일(Build, F7)

- Build succeeded 확인



USB-ISP 도구 연결

- VCC
- GND
- MOSI
- MISO
- SCK
- RST



- 서로 같은 이름에 연결

ISP란?

- In-System Programming
 - 플래шив롬을 탈착하지 않고 프로그램 기록 가능한 형태를 의미
- 컴파일된 프로그램을 AVR 칩에 기록하는 작업을 "다운로딩"이라고 부르며, 이를 퓨징(Fusing), 플래싱(Flashing), 혹은 프로그래밍(Programming)이라고 부르기도 함



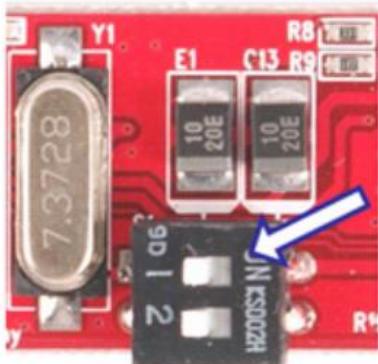
VCC와 GND

- VCC
 - Voltage collector
 - (+) 라인
 - 높은 전압 ex> 12v, 5v, 3.3v
 - Vdd라고도 함 (TTL/CMOS 차이)
- GND
 - Ground
 - (-) 라인
 - 낮은 전압 == 기준 전압 == 0v
 - Vss라고도 함

USB-ISP의 스위치

- 1번 스위치 ON : USB 전원 사용

- ◆ USB 버스 전원 사용 방법

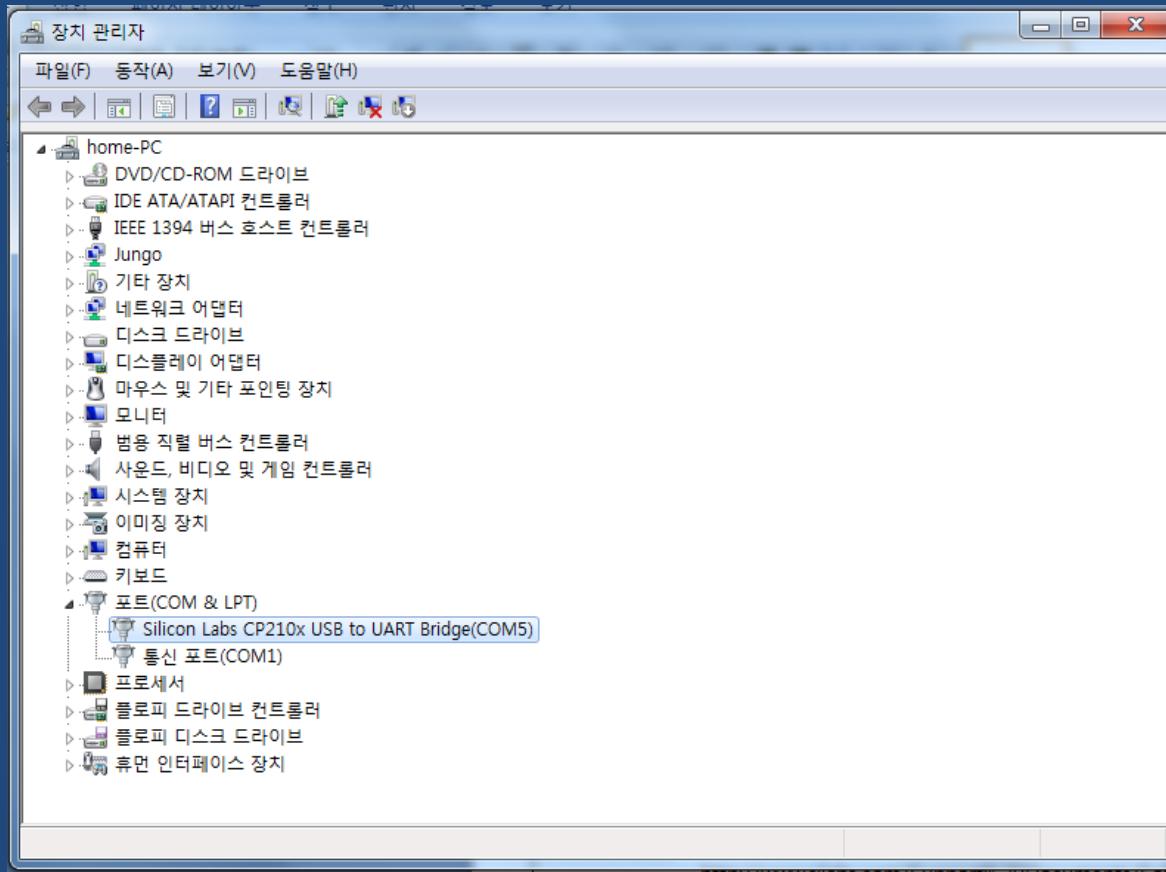


USB-ISP 를 사용하면서 USB 의 전원을 사용하고 싶다면 USB-ISP 의 DIP 스위치 중 안쪽 1번 스위치를 ON(오른쪽으로) 시키면, PC 에서 공급되는 USB 전원을 사용할 수 있습니다. 여기서 공급되는 전원으로 모터를 구동하거나 많은 전류가 흐르는 회로를 동작시킬 경우 USB-ISP 나 USB 포트에 무리를 줄 수 있습니다.

USB 스펙상으로는 500mA 까지 사용할 수 있지만 순간적으로 많은 전류가 흐를 경우 문제가 생길 수 있기 때문에 100mA까지만 사용하는 것을 권장하고 있습니다. 기본 셋팅은 USB 전원을 사용하지 않은 것 (스위치 왼쪽) 으로 되어 있습니다.

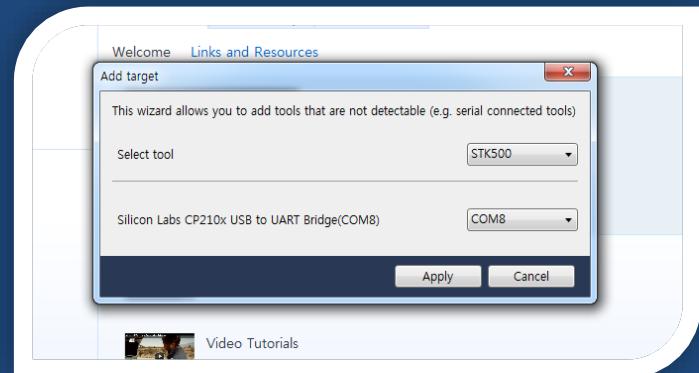
Driver 인식 실패 시

- USB to UART 드라이버 설치
- <http://www.silabs.com/products/mcu/pages/usbtouartbridgevcpdrivers.aspx>

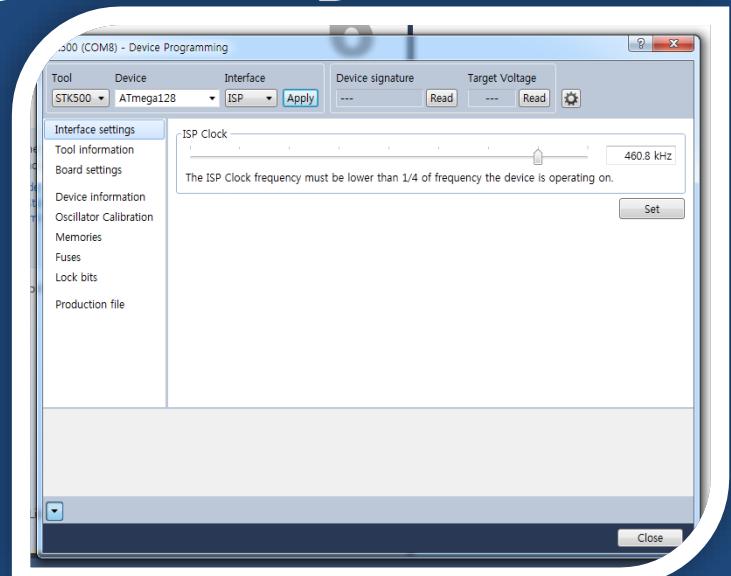


펌웨어 Programming

- 1. Tools -> Add Target
 - STK500
 - COM 포트 선택

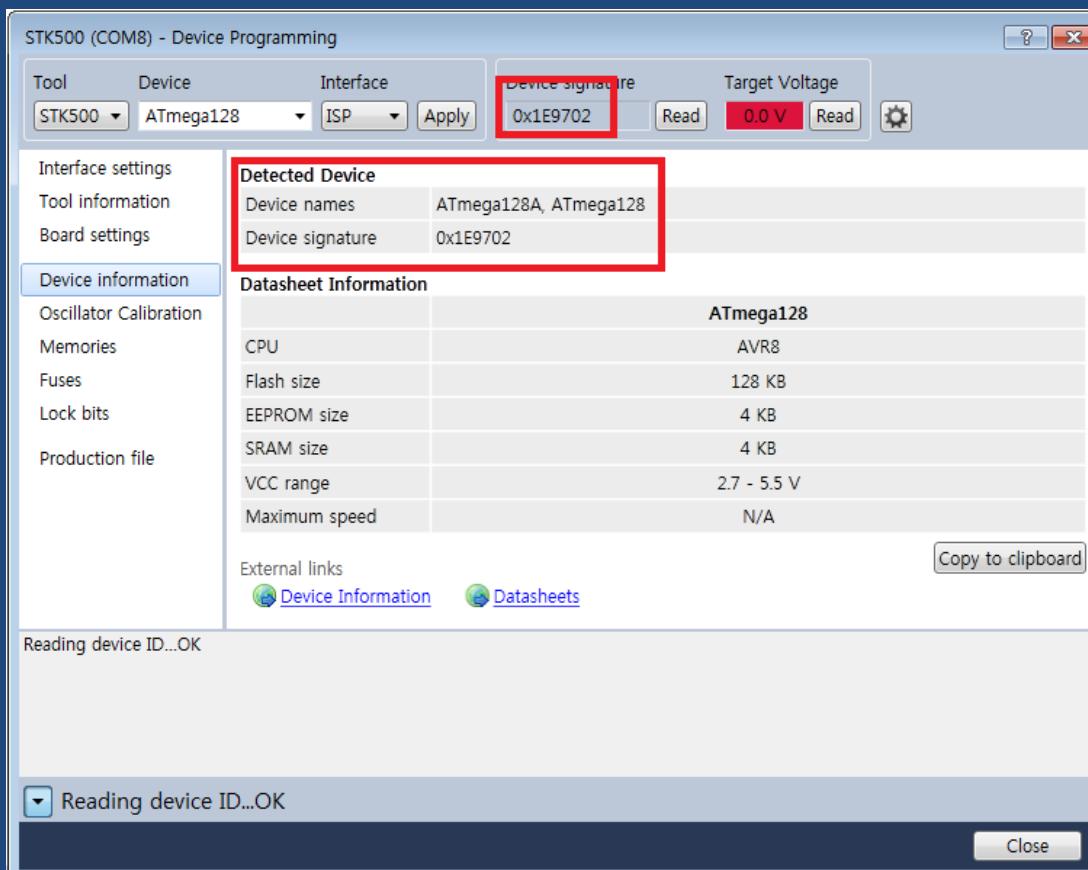


- 2. Tools -> Device Programming
 - Tool -> STK500 COM(n)
 - Device -> Atmega128
 - Interface -> ISP
 - Apply 클릭



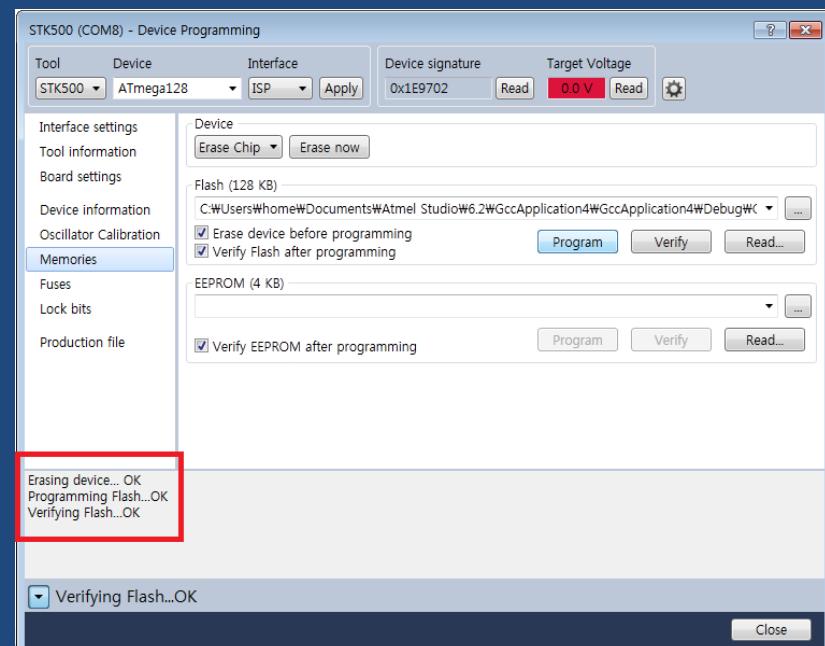
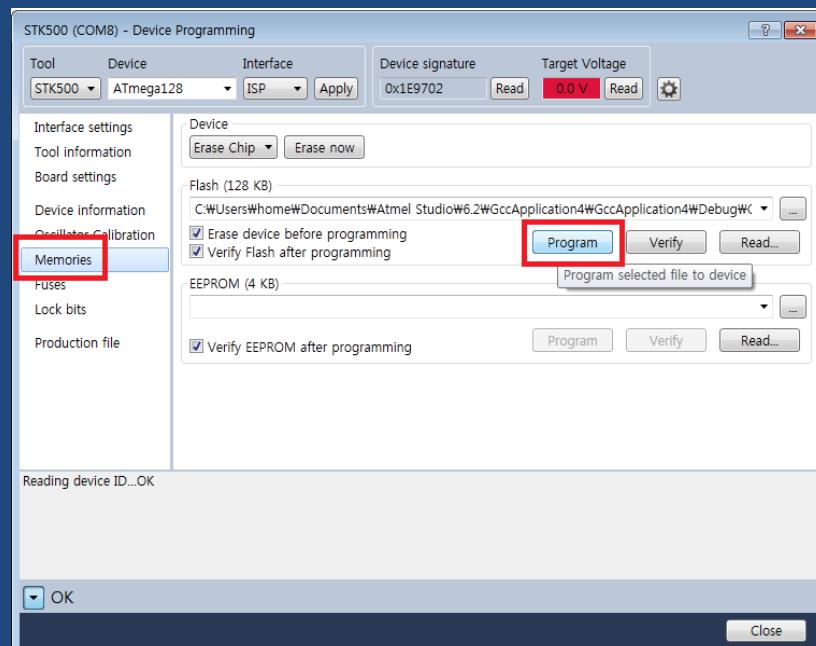
펌웨어 Programming

- Device signature 확인

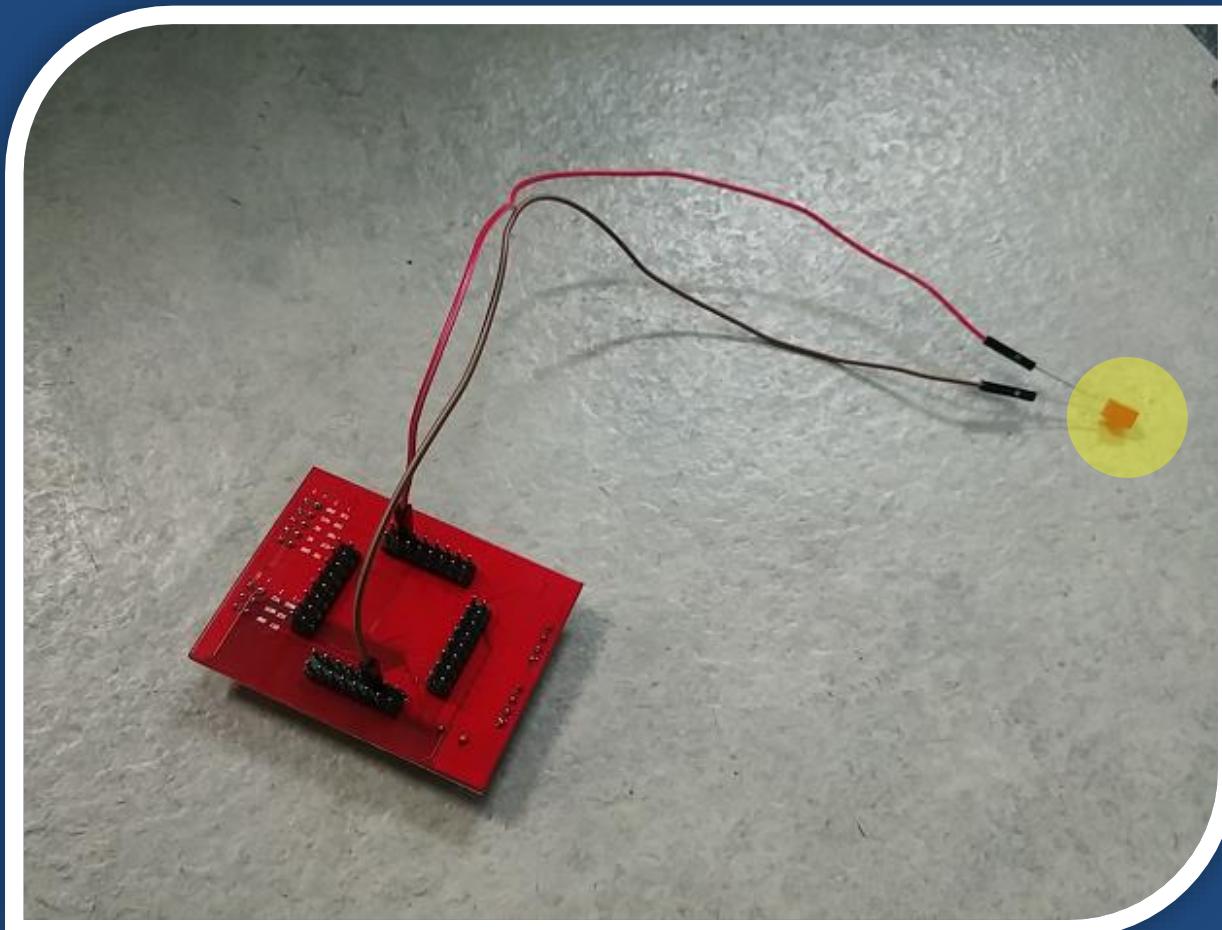


펌웨어 Programming

- Downloading == Programming
== Fusing == Writing
- 개발 PC → AVR MCU로 프로그램 입력



빛나라 LED야



소스코드 설명

- DDRA : Data Direction Register
 - 0 : 데이터 입력
 - 1 : 데이터 출력
- PORTA
 - 0 => 0볼트 출력
 - 1 => 5볼트 출력

```
#include <avr/io.h>

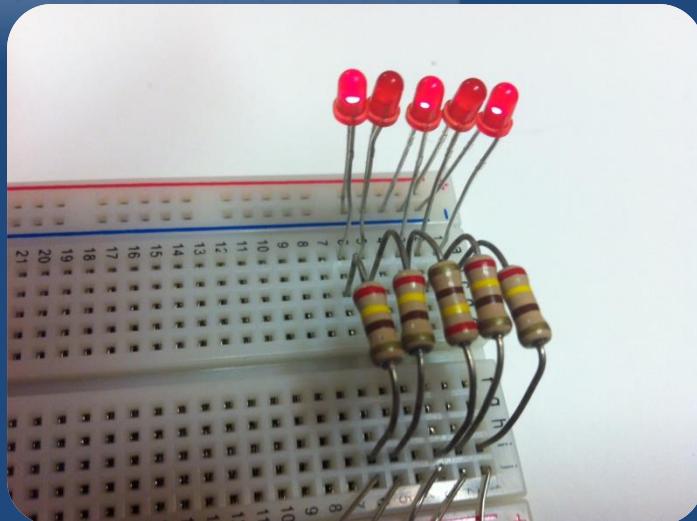
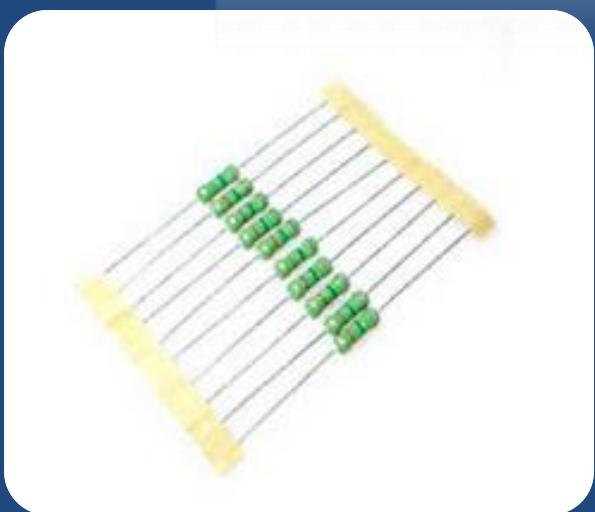
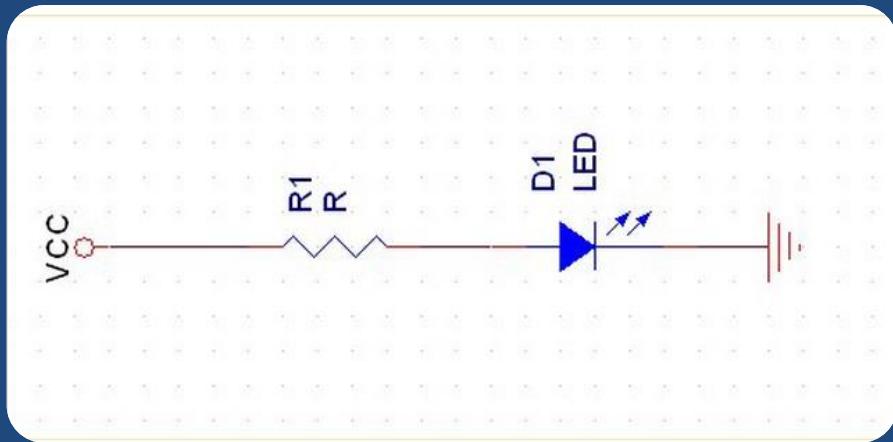
int main(void)
{
    DDRA = 0xff;    // 0b11111111;
    PORTA = 0xff;

    return 0;
}
```

저항 연결하기

- 일반적인 LED의 스펙
 - 20mA에서 이상적으로 작동
 - AVR 핀 하나에서 나오는 전류량 = 50mA
- 저항 없이 사용 시
 - 장시간 사용 시 LED나 MCU에 악영향
 - 저 품질 LED 사용 시 훼손
 - LED 발열로 인한 화상 우려

저항 연결

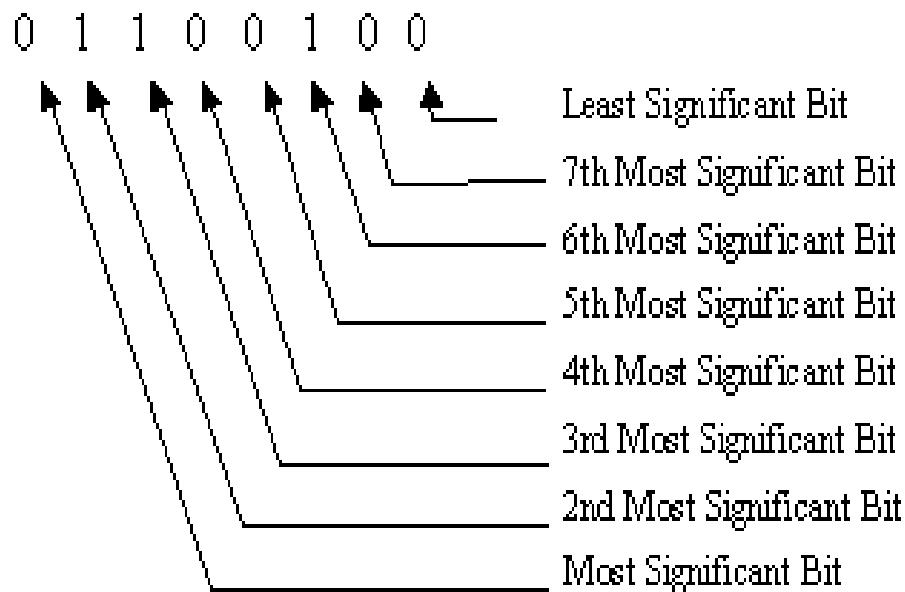


실습 문제

- 오직 PA3번에 연결했을 때만 LED에 불이 켜지게 해보세요.
- 즉, PA0,1,2나 PA4,5,6,7에선 불이 들어오면 안 됨

PA0은 MSB인가 LSB인가?

- PORTA = 0b01100100;
- PA0 == LSB 이다.



실습 문제

- PORTA와 PORTB를 이용하여 LED의 불을 켜보세요.
- 즉, GND 핀을 사용하지 않습니다.

실습 문제

- PORTA 하나만 이용하여 LED에 불을 켜보세요.
- 즉, GND나 다른 PORT를 사용하지 않습니다.

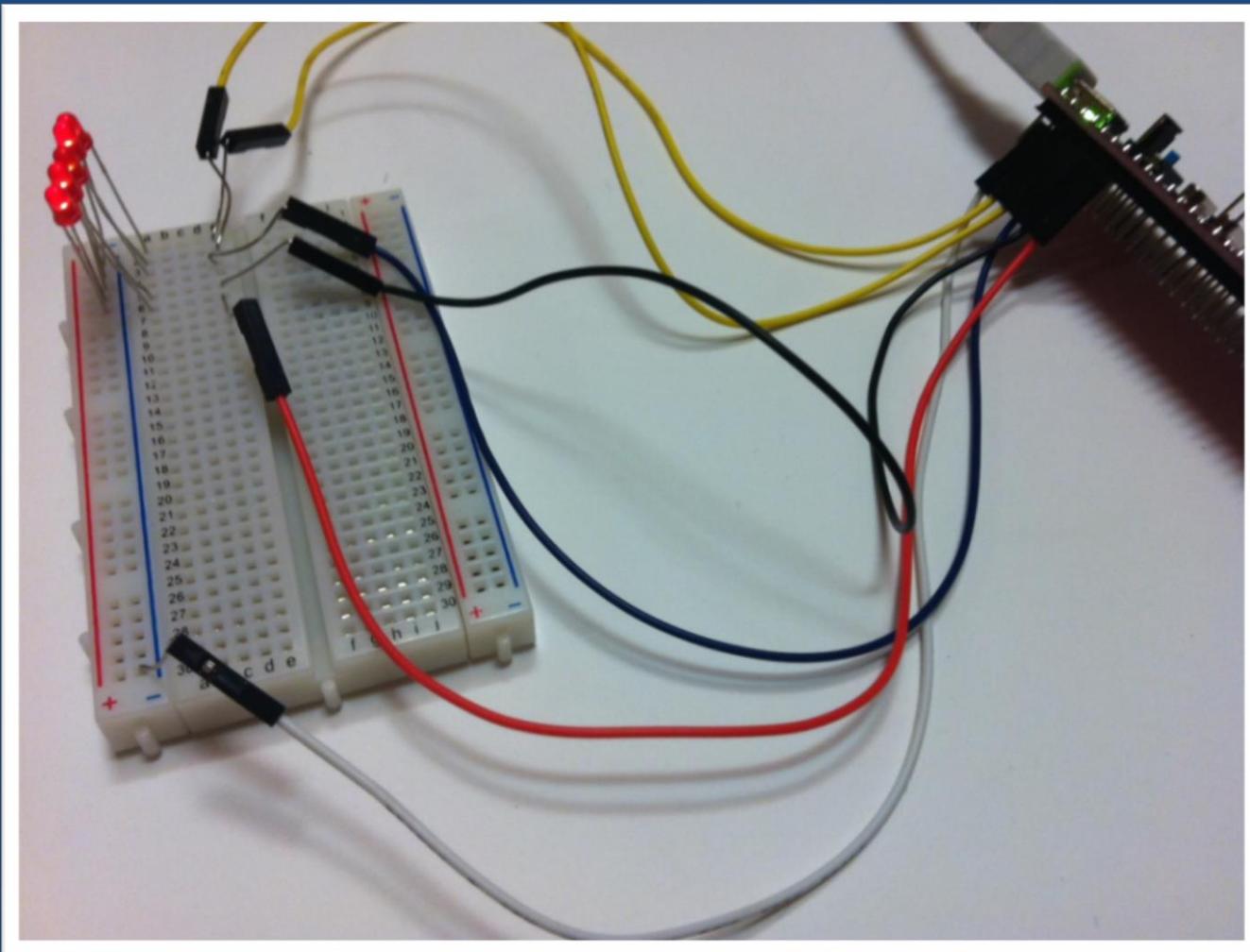
점멸하라 LED여!

```
#define F_CPU 8000000UL
#include <avr/io.h>
#include <util/delay.h>

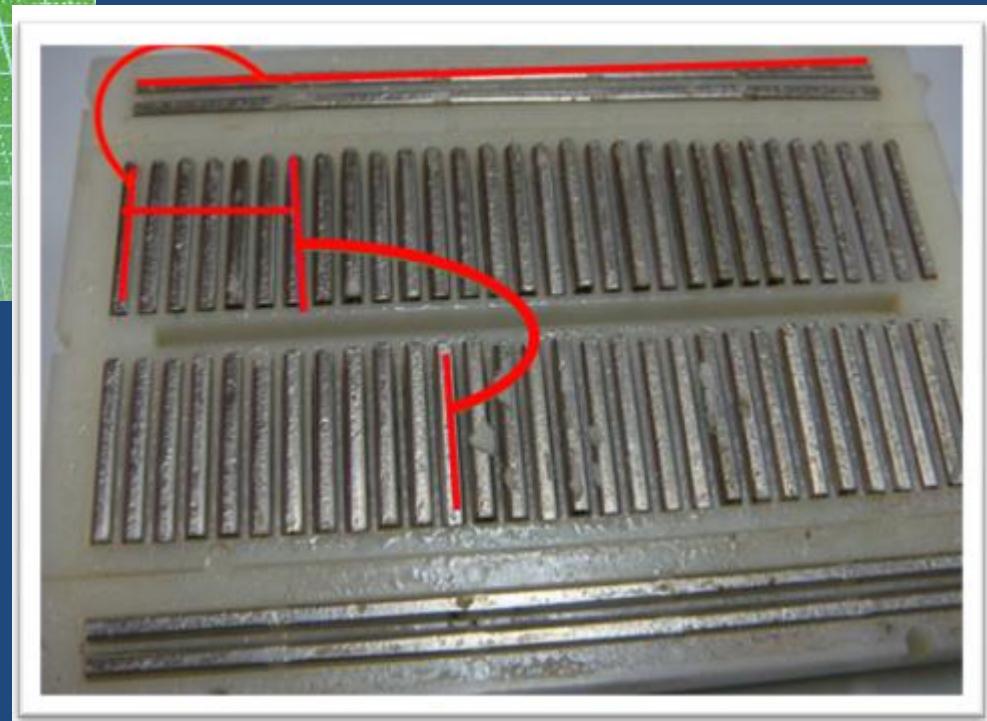
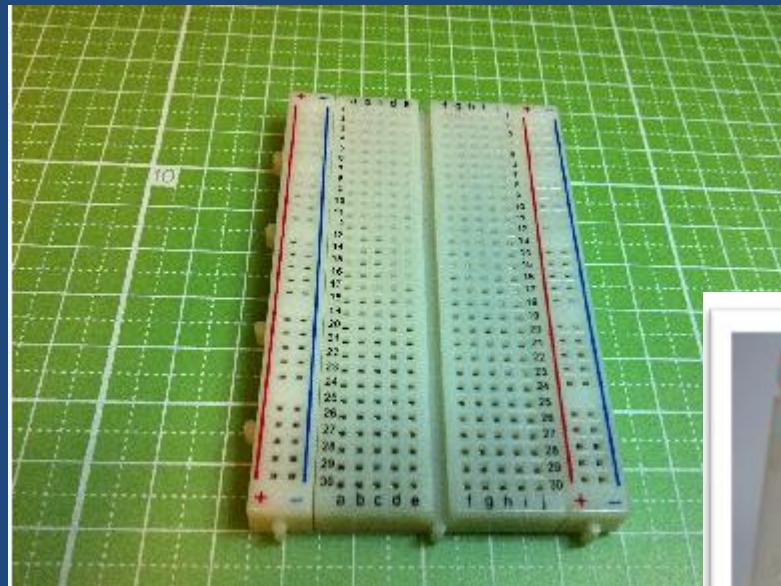
int main(void)
{
    DDRA = 0xff;

    while(1)
    {
        PORTA = 0xff;
        _delay_ms(1000);
        PORTA = 0x00;
        _delay_ms(1000);
    }
}
```

여러 개의 LED 제어



빵판 사용하기



여러 개의 LED Show

- 문제1
 - 홀수번째 LED만 켜지게 해보세요.
- 문제2
 - 짝수번째 LED만 켜지게 해보세요.
- 문제3
 - 움직이는 LED 쇼를 자유롭게 만들어 보세요.

LED의 밝기를 반으로

```
#define F_CPU 8000000UL
#include <avr/io.h>
#include <util/delay.h>

int main(void){
    DDRA = 0xFF;

    while(1){
        PORTA = 0xff;
        _delay_ms(10);

        PORTA = 0x00;
        _delay_ms(10);
    }
}
```

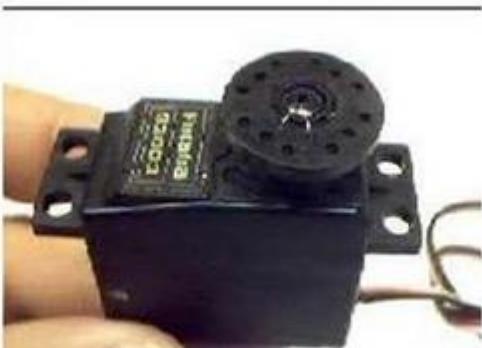
달려라 모터야



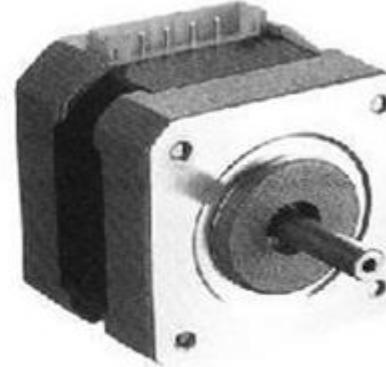
모터의 종류



DC모터



서보모터



스테핑모터

- DC 모터 : 일반 모터
 - RC카, 선풍기 등
- 서보 모터 : 원하는 각도 조정 가능 (EX> 90도, 45도)
 - 로봇 팔 등
- 스텝링 모터 : 원하는 각도만큼 미세하게 조정 가능 (EX> 1도)
 - 프린터, CNC 공작기, 디스크헤더 등

실습 문제

- 모터를 돌려보세요~

Any Vendor 소형 모터 3V (기어포함)

구동 3V DC모터입니다. 기어 포함(0.4모듈 잇수는 7) / 12파이 소형 모터, 벌크타입의 소형모터입니다. 아주 소형중에 하나라서 여러분의 아이디에 사용을 하실수 있습니다.



> 상품코드	21857
> 판매가격	790원 (부가세 미포함)
> 적립금	0원
> 평균준비기간	1~2일
> 브랜드	Any Vendor [브랜드몰바로가기]
> 최소주문수량	1 개
> 수량	<input type="text" value="1"/> ▲ ▼

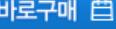
대량물량 구매상담
견적요청
대량구매/현키구매

대학, 공공기관 구매상담
후불구매
후불구매 및 구매대행

매장 방문 구매
매장방문
본사, 구로 방문 구매

위 상품 이미지는 참조용 대표 이미지이며, 정확한 사양은 데이터시트에서 확인하셔야 합니다.

 [큰이미지 보기](#)

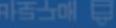
 바로구매

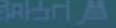
 장바구니

 관심상품

④ 드라이버 카드
드라이버 카드는 데이터시트에서 확인하세요.
하위 이미지는 예상 대표 이미지이며

 미출고

 예매

 당일유동

부저 스피커 사용하기

SPE-1404W-48

회로외장형 피에조 부저, 사이즈 : $\Phi 14.0 \times 4.0(h)$, Input : 10V, 크기 : 85db



위 상품 이미지는 참조용 대표 이미지이며,
정확한 사양은 데이터시트에서 확인하셔야 합니다.

큰이미지 보기

> 상품코드	32844
> 판매가격	360원 (부가세 미포함가)
> 제조사	KANGSEO EST
> 적립금	0원
> 평균준비기간	1~2일
> 최소주문수량	1 개
> 수량	<input type="text" value="1"/>

바로구매

장바구니

관심상품

이미지 확대
클릭한 이미지는 데이터시트에서 확인하세요.
이미지는 참조용 대표 이미지이며,

바로구매

장바구니

관심상품

부저 스피커 사용하기

- 반복되는 전기 신호를 전달하여 진동 생성

```
#define F_CPU 8000000UL

#include <avr/io.h>
#include <util/delay.h>

int main(void)
{
    DDRA = 0xff;

    while(1){
        PORTA = 0xff;
        _delay_ms(1);

        PORTA = 0x00;
        _delay_ms(1);
    }
}
```

실습 문제

- 주파수를 변경하여 다양한 음을 만들어 보세요.
 - 더 높은 음
 - 더 낮은 음

	C	C#	D	Eb	E	F	F#	G	G#	A	Bb	B
0	16.35	17.32	18.35	19.45	20.60	21.83	23.12	24.50	25.96	27.50	29.14	30.87
1	32.70	34.65	36.71	38.89	41.20	43.65	46.25	49.00	51.91	55.00	58.27	61.74
2	65.41	69.30	73.42	77.78	82.41	87.31	92.50	98.00	103.8	110.0	116.5	123.5
3	130.8	138.6	146.8	155.6	164.8	174.6	185.0	196.0	207.7	220.0	233.1	246.9
4	261.6	277.2	293.7	311.1	329.6	349.2	370.0	392.0	415.3	440.0	466.2	493.9
5	523.3	554.4	587.3	622.3	659.3	698.5	740.0	784.0	830.6	880.0	932.3	987.8
6	1047	1109	1175	1245	1319	1397	1480	1568	1661	1760	1865	1976
7	2093	2217	2349	2489	2637	2794	2960	3136	3322	3520	3729	3951
8	4186	4435	4699	4978	5274	5588	5920	6272	6645	7040	7459	7902

8 4186 4435 4699 4978 5274 5588 5920 6272 6645 7040 7459 7902

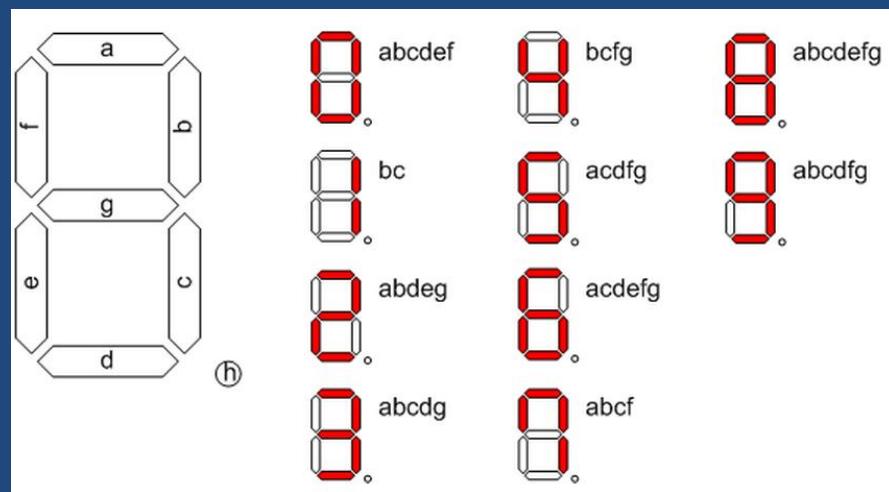
9 5083 5541 5346 5486 5631 5164 5288 5250 5215 5040 5426 5605

참고 자료

- 주파수에 따른 소리의 차이
 - <https://www.youtube.com/watch?v=qNf9nzvnd1k>
- 슈퍼마리오 연주하기
 - <https://www.youtube.com/watch?v=ADe89-w3JIA>

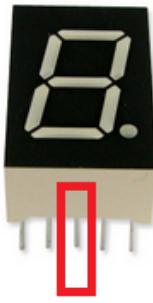
7 Segment 제어

- 7 Segment = FND(Flexible Numeric Display)
- 0에서 9까지의 숫자를 표현할 수 있는 장치
- Anode 타입과 Cathode 타입이 있음
 - Anode : 가운데 핀에 +를 입력하는 방식
 - Cathode : 가운데 핀에 -를 입력하는 방식



7 Segment 제어

Any Vendor 5101CSR
7세그먼트x1, 색상:빨강, Cathode타입, 12.7mm x 19mm



- 입력

위 상품 이미지는 참조용 대표 이미지이며,
정확한 사양은 데이터시트에서 확인하셔야 합니다.

④ 큰이미지 보기

> 상품코드	11552
> 판매가격	420원 (부가세 미 포함)
> 제조사	DAKWANG (중국)
> 적립금	0원
> 평균준비기간	2~3일
> 브랜드	Any Vendor [브랜드몰 바로가기]
> 최소주문수량	1 개
> 수량	<input type="text" value="1"/> ▲ ▼

바로구매 

장바구니 

관심상품 

실습 문제

- 숫자 3을 출력해 보세요.
- 숫자 9를 출력해 보세요.
- 숫자 1부터 9까지를 1초 단위로 출력해 보세요.

쉬어가기

AVR로 만들 수 있는 것들의 예제

LED DOT MATRIX



http://www.youtube.com/watch?v=cgh_SVRqch8

3D 큐브



<http://www.youtube.com/watch?v=6mXM-oGggrM>

MP3 플레이어



http://www.youtube.com/watch?v=A5Xi_2CtD7Y

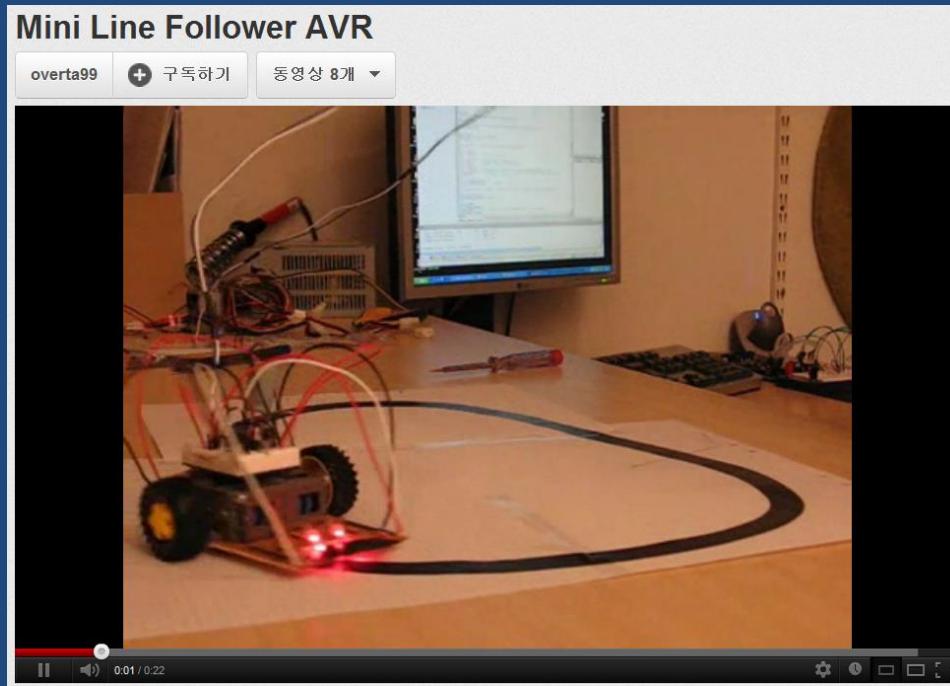
* MP3 Codec 칩 내장

RC카



<http://www.youtube.com/watch?v=pd99SZygJGs>

라인 트레이서



http://www.youtube.com/watch?v=F1vyGUk_QIA

* 적외선 발광부와 수광부 이용 (바닥이 검정색일 때 반사되지 않음)

2족 보행 로보트



<http://www.youtube.com/watch?v=fCcXYluxVIA>

6족 보행 로보트

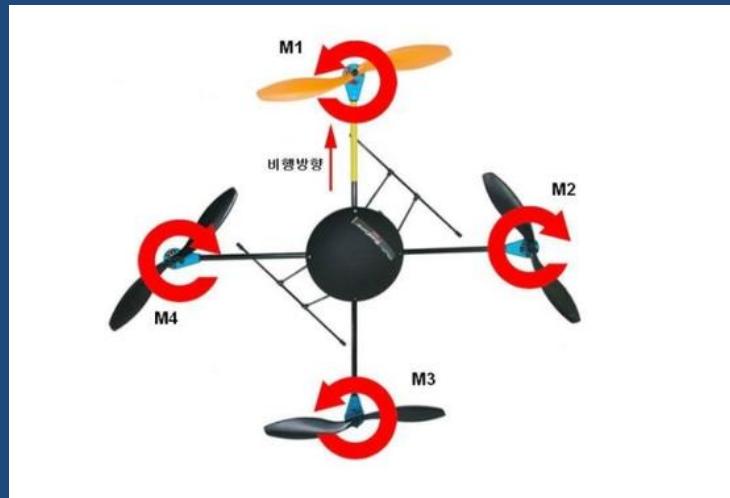


<http://www.youtube.com/watch?v=0CcFp-Rnq68>

쿼드콥터

Home build 30cm mini quad copter

white1203 구독하기 동영상 43개



<http://www.youtube.com/watch?v=FEXxMsqbbrU>

* 자이로센서(기울기), 가속도센서(기울기), 지자기센서(방향) 등 내장

게임기



<http://www.youtube.com/watch?v=RYyAIQygg-k>

웹 서버



<http://www.youtube.com/watch?v=5Ur5oy0S-ag>

* Ethernet or WIFI 칩 내장

GPS 장치



<http://www.youtube.com/watch?v=Q4IDyVV3HNc>

* GPS 센서 내장

3D 프린터



<http://www.youtube.com/watch?v=OzvCqc2jTX8>

UART 프로그래밍

- UART란?
 - Universal Asynchronous Receiver/Transmitter
 - 하드웨어 통신 프로토콜 중 하나
 - 특히 디버깅 메시지 출력 용도로 많이 쓰임

UART 프로그래밍

```
#define F_CPU 8000000UL
#include <avr/io.h>
#include <util/delay.h>

int main(void)
{
/* Status Register 0A */
UCSR0A = 0x00;

/* Status Register 0B */
/* RX/TX Enable = 00001000 */
UCSR0B = 0x08;

/* Status Register 0C */
/* No parity, 8bit = 0110 */
UCSR0C = 0x06;

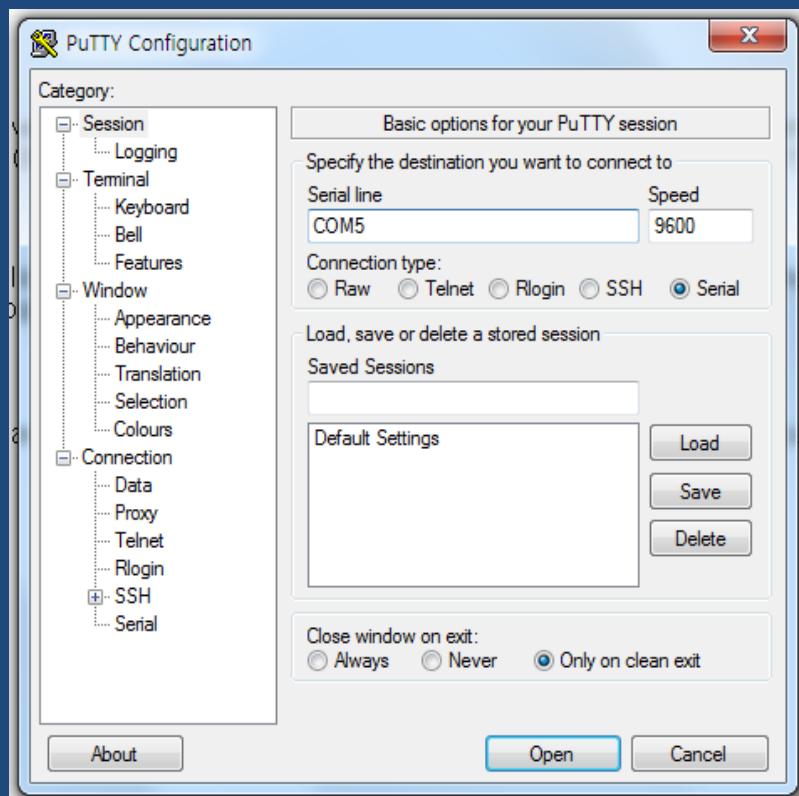
/* 중요 : Baud Rate 설정 */
/* BPS = 9600 */
UBRR0H = 0;
UBRR0L = 47;

while(1)
{
    UDR0 = 'A';
    _delay_ms(1000);
}
```

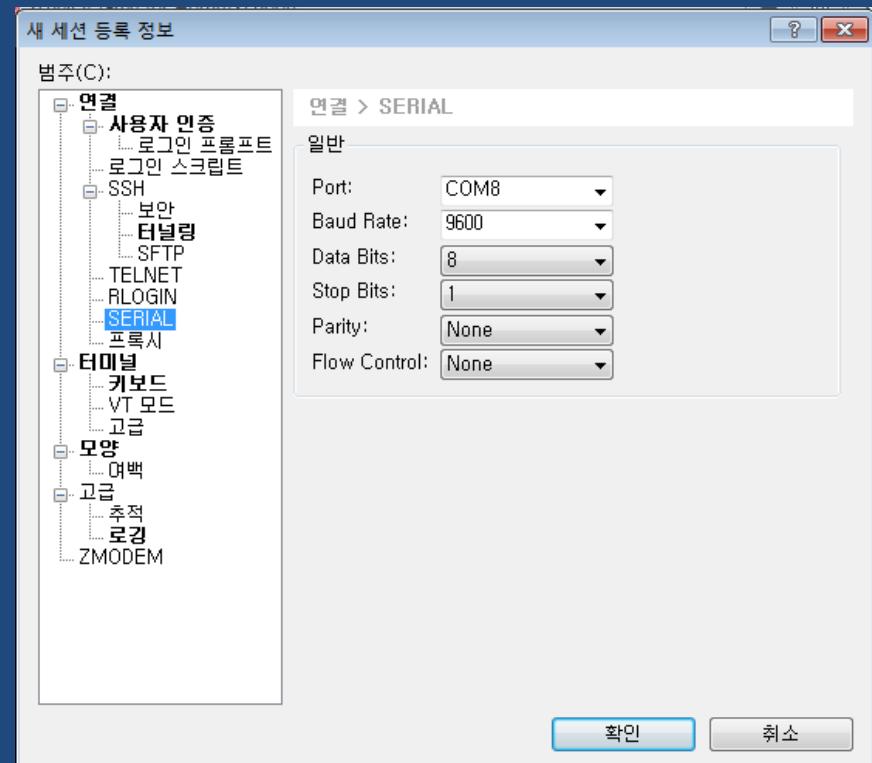
UART 핀 연결

- VCC <-> VCC
- GND <-> GND
- TX <-> RX
- RX <-> TX
- ISP 연결은 해제
- 주의 : 래빗보드와 UART 장비 연결 시에만 예외적으로 RX<->RX, TX<->TX로 연결해 주세요.
 - 래빗보드측의 RX/TX 인쇄가 문제

UART 포트 접속

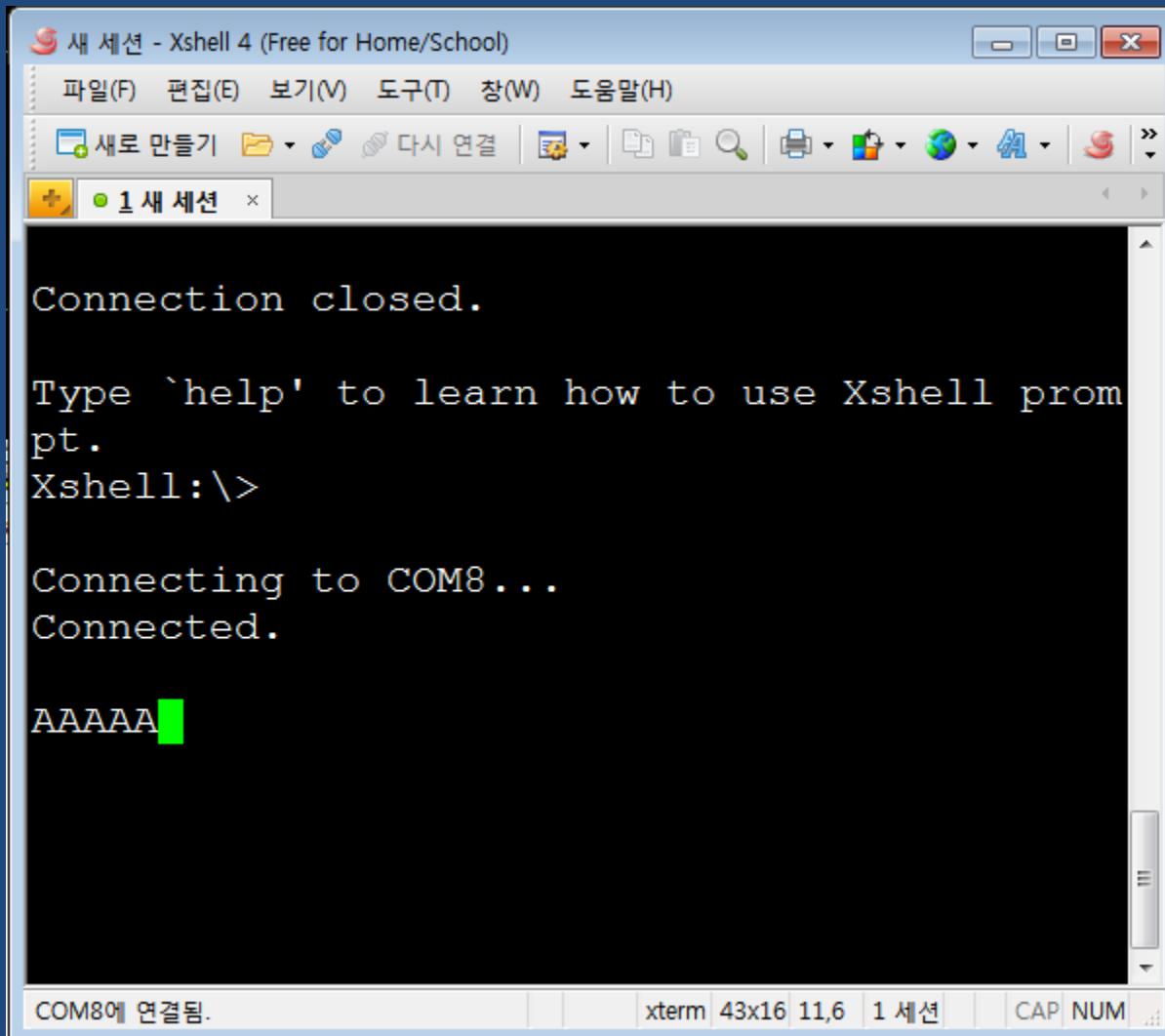


putty



xshell

UART 포트 접속

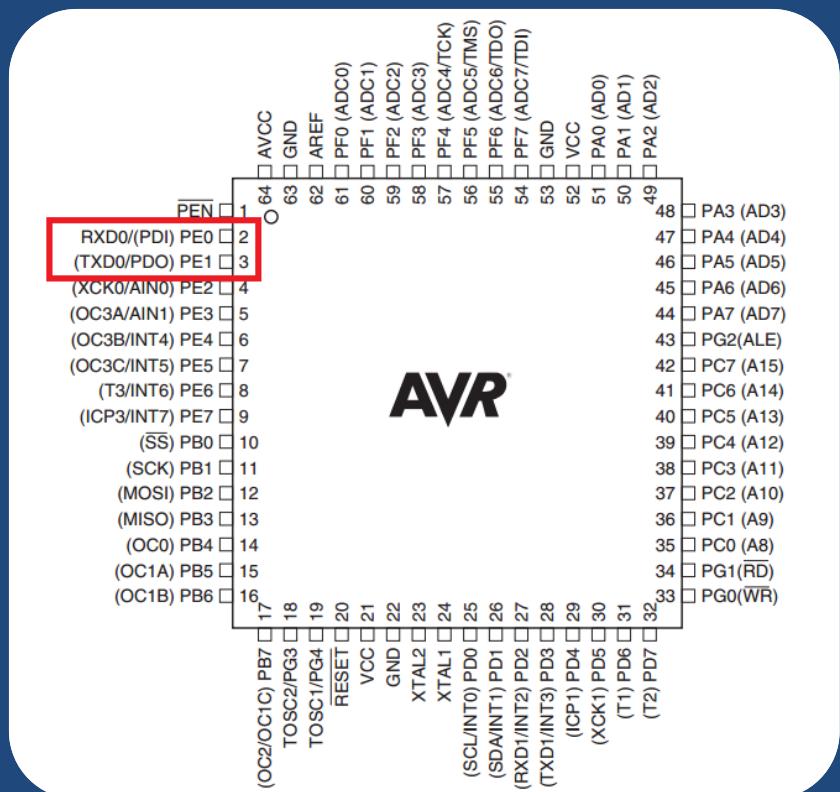


UART 프로그래밍

- UART에 사용되는 레지스터 목록

- UCSR0A
- UCSR0B
- UCSR0C
- UBRR0H
- UBRR0L
- UDR0

- UART에 사용되는 핀들
- PE0 (RXD0)
- PE1 (TXD0)



코드 설명

- UCSR 레지스터
 - Control and Status Register의 약자
 - 통신 환경 설정
- UBRR 레지스터
 - Baud Rate Register의 약자
 - 통신 속도 정의
 - UART CALC : <http://wormfood.net/avrbaudcalc.php>
- UDR 레지스터
 - UART 값 전달

코드 설명

- UCSR0A 레지스터
 - 송수신되는 데이터의 현황이 기록됨 (00000000)

7	6	5	4	3	2	1	0
데이터↓ 수신완료 시 1	데이터 ↓ 송신완료 시 1	송신 버퍼가↓ 비어있으면 1 0					

- UCSR0B 레지스터
 - 기능 설정 (00001000)

7	6	5	4	3	2	1	0
			수신(RX)부 활성화	송신(TX)부 활성화			

코드 설명

- UCSR0C 레지스터
 - 통신 설정 (00000110) => 8비트 데이터

7	6	5	4	3	2	1	0
X	동기or비동 기모드 선택	패리티 모드(1)	패리티 모드(2)	스톱↓ 비트	데이터↓ 비트(1)	데이터↓ 비트(2)	

- UBRR0H, UBRR0L 레지스터
 - 통신 속도 설정
 - 초당 전송되는 bit의 수
 - /* BPS = 9600 */
 - UBRR0H = 0;
 - UBRR0L = 47;

실습 문제

- UART 터미널에 “hello world”를 출력해 보세요.

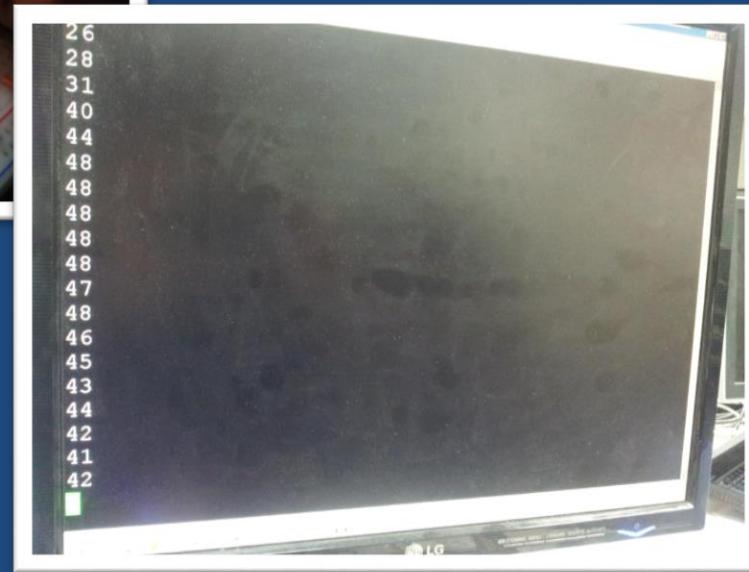
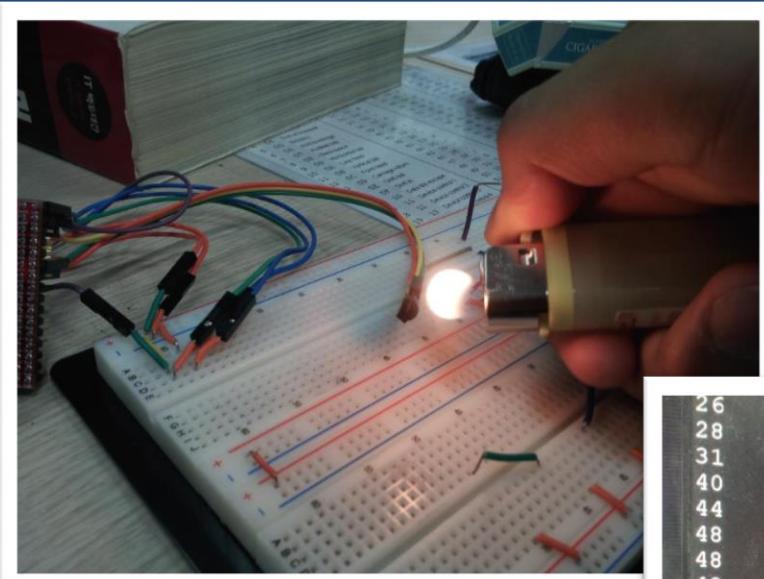
실습 문제

- 문자열 출력을 쉽게 해주는 함수를 만들어 보세요.
- Ex> Puts(“hello world\rf\fn”);

정답

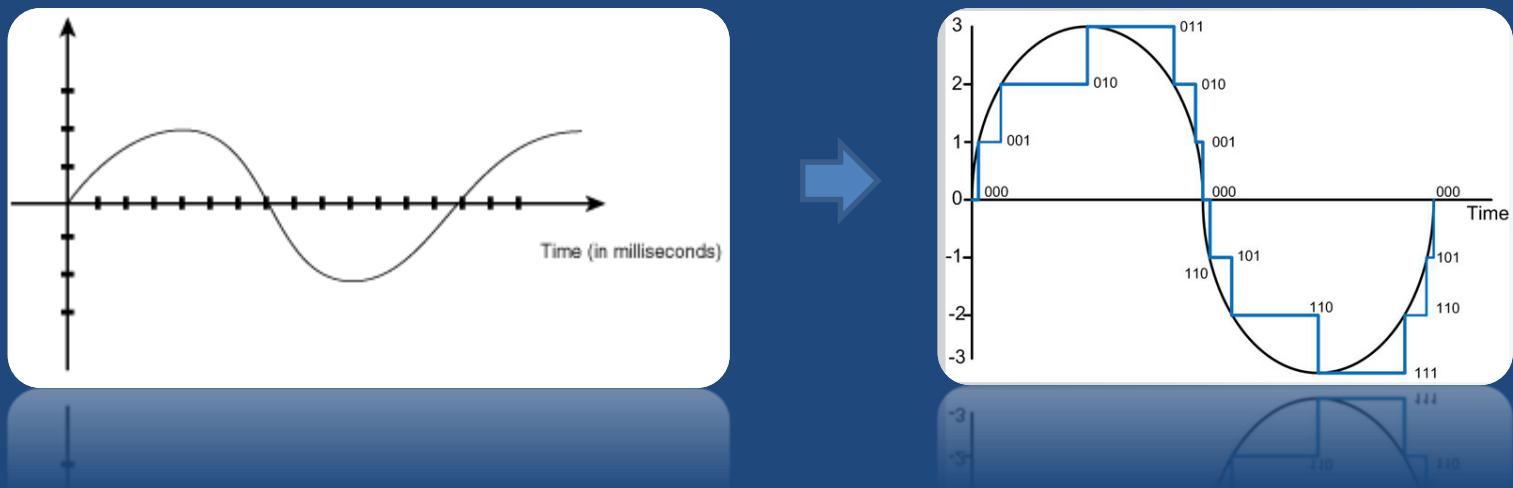
```
void Puts(char *data)
{
    int i;
    for(i=0; i<strlen(data); i++)
    {
        while((UCSR0A & 0x20) == 0);
        UDR0 = data[i];
    }
}
```

온도 센서 사용 실습



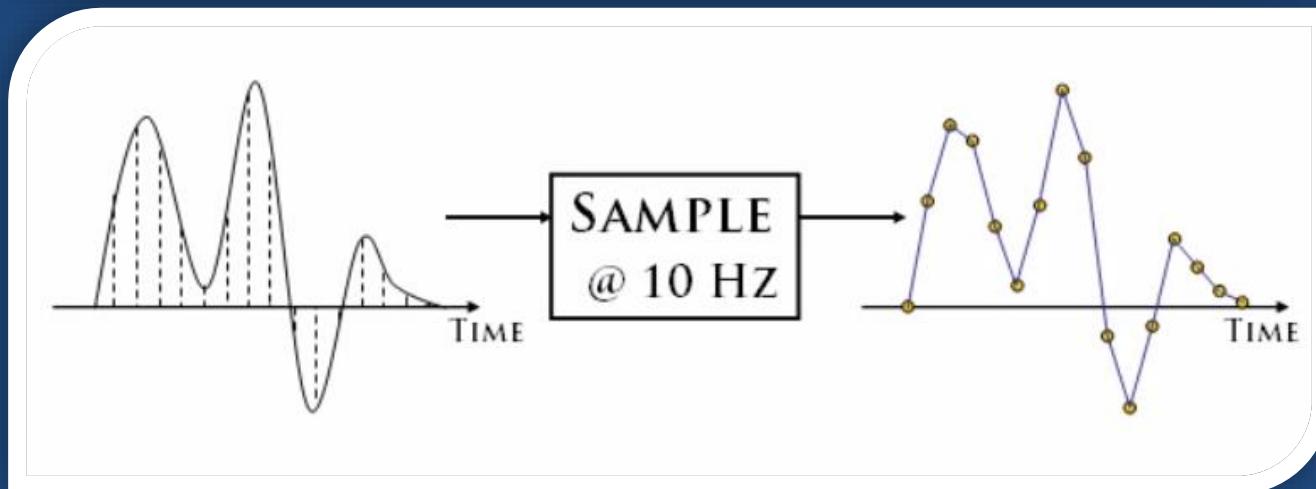
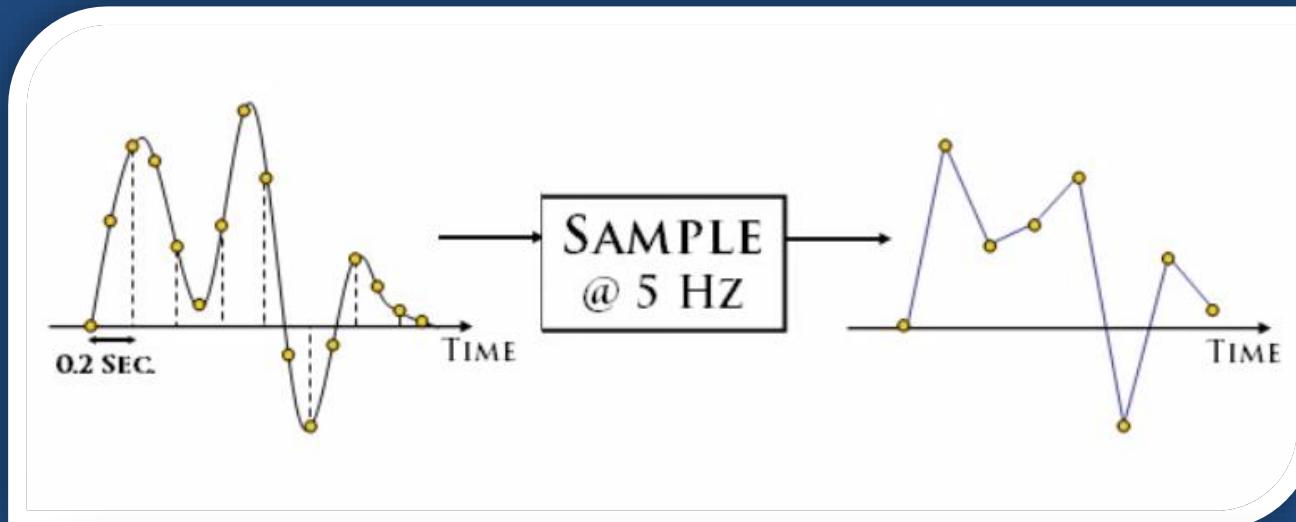
온도 센서 사용 실습

- ADC란?
 - Analog to Digital Converter의 약자
 - 아날로그 형태의 데이터를 디지털 형태의 데이터로 변환하는 장치를 의미



Sampling

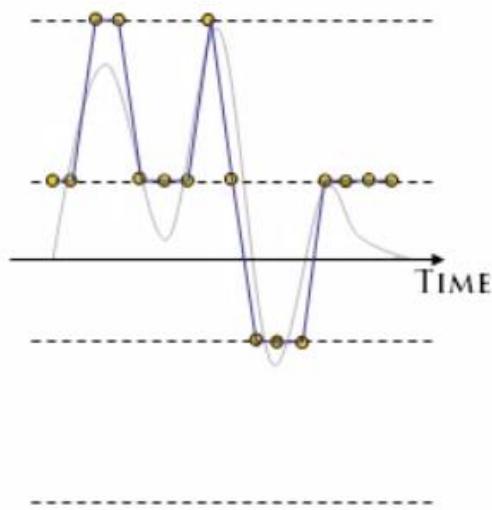
(<https://www.youtube.com/watch?v=zC5KFnSUPNo>)



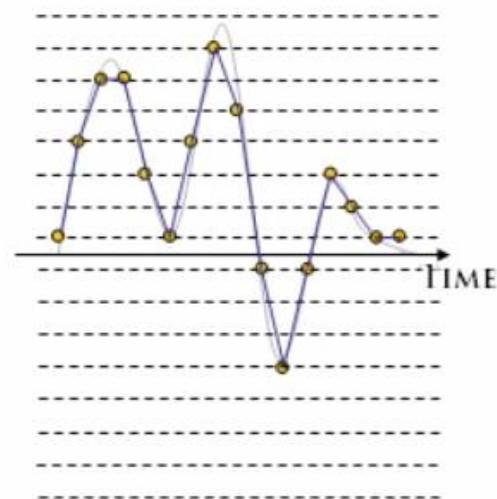
Quantizing(bit depth, 분해능)

- 디지털 값을 얼마나 세밀하게 표현하는지를 결정

2-BIT QUANTIZATION



4-BIT QUANTIZATION



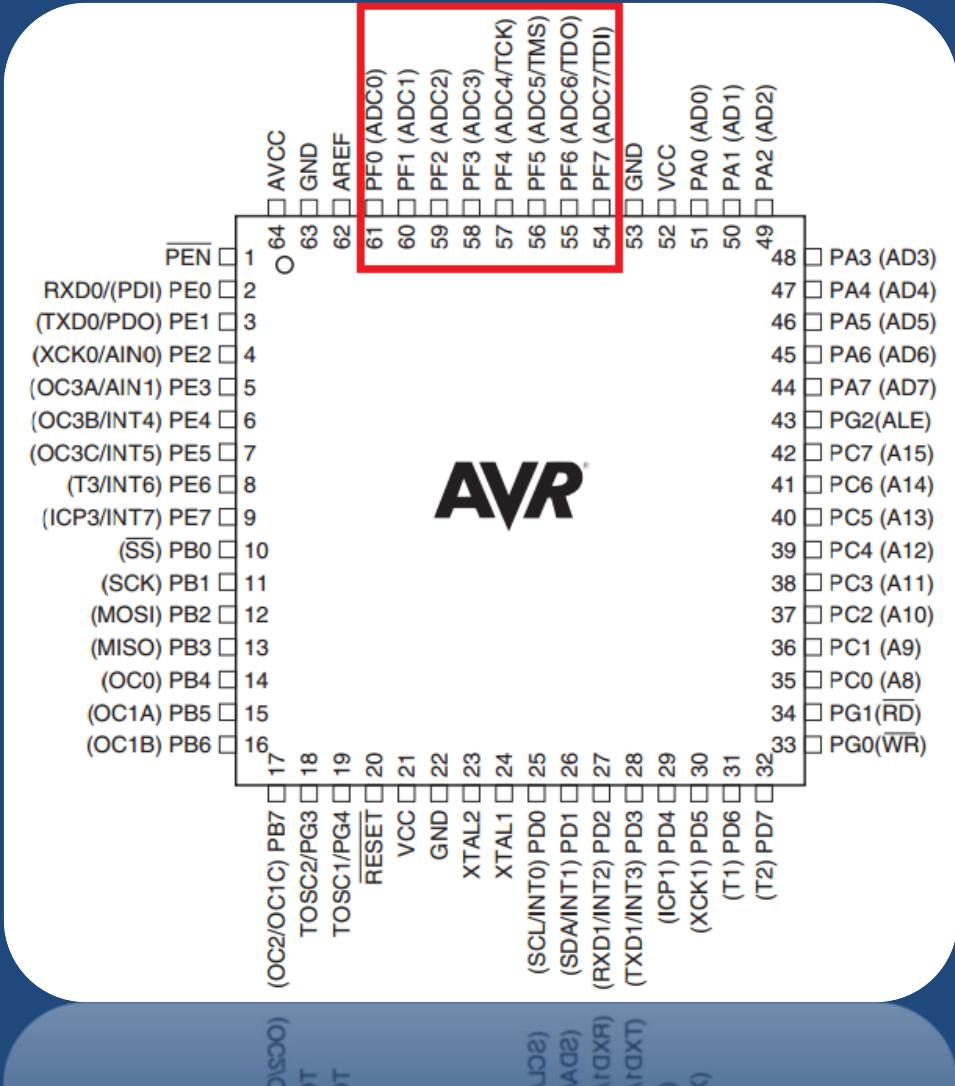
ADC 핀 목록

- ATmega128A의 ADC

- PF0~PF7

- 8채널

- 8개까지 연결 가능



온도 센서 사용 실습

```
#define F_CPU 8000000UL
#include <avr/io.h>
#include <util/delay.h>

void Puts(char *data)
{
    int i;
    for(i=0; i<strlen(data); i++)
    {
        while((UCSROA & 0x20) == 0);
        UDR0 = data[i];
    }
}

int main(void)
{
    char result[20];
    int value;

    DDRA = 0xff;
    PORTA = 0xff;

    UCSROA = 0x00;
    UCSROB = 0x88;
    UCSROC = 0x06;
    UBRROH = 0;
    UBRROL = 47;           // bps == 9600

    while(1)
    {
        ADMUX = 0; // ADC 설정 초기화 시작
        ADCH = 0;
        ADCL = 0;
        ADCSRA = 0xc7;

        // 설정이 완료될 때까지 기다림
        while((ADCSRA & 0x10) == 0);

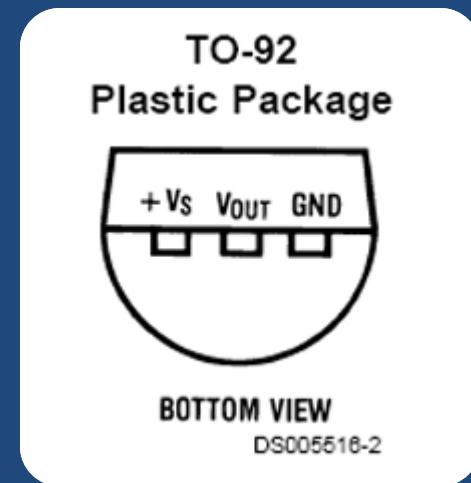
        // ADC0의 상하위 바이트 값을 이용하여 데이터 복원
        value = ADCL | (ADCH << 8);

        memset(result, 0, 20);
        sprintf(result, "Temperature : %dW\r\n", (value*100/205));
        Puts(result);
        _delay_ms(1000);
    }
}
```

온도 센서 사용 실습

- BOTTOM VIEW임에 주의
- 다리 아래로부터 봐야 한다.

- LM35의 GND → 보드의 GND
- LM35의 VCC → 보드의 PORTA0
- Vout은 PF0에 연결
- 보드의 AREF에 PORTA1 연결



소스 설명

- ADMUX (00000000)

- ADC 설정 초기화

ADMUX							
7	6	5	4	3	2	1	0
기준전압	데이터저장순서	ADC 입력 채널 선택					

- 모두 0으로 설정 (기준전압 = AREF)

- ADCSRA (11000111)

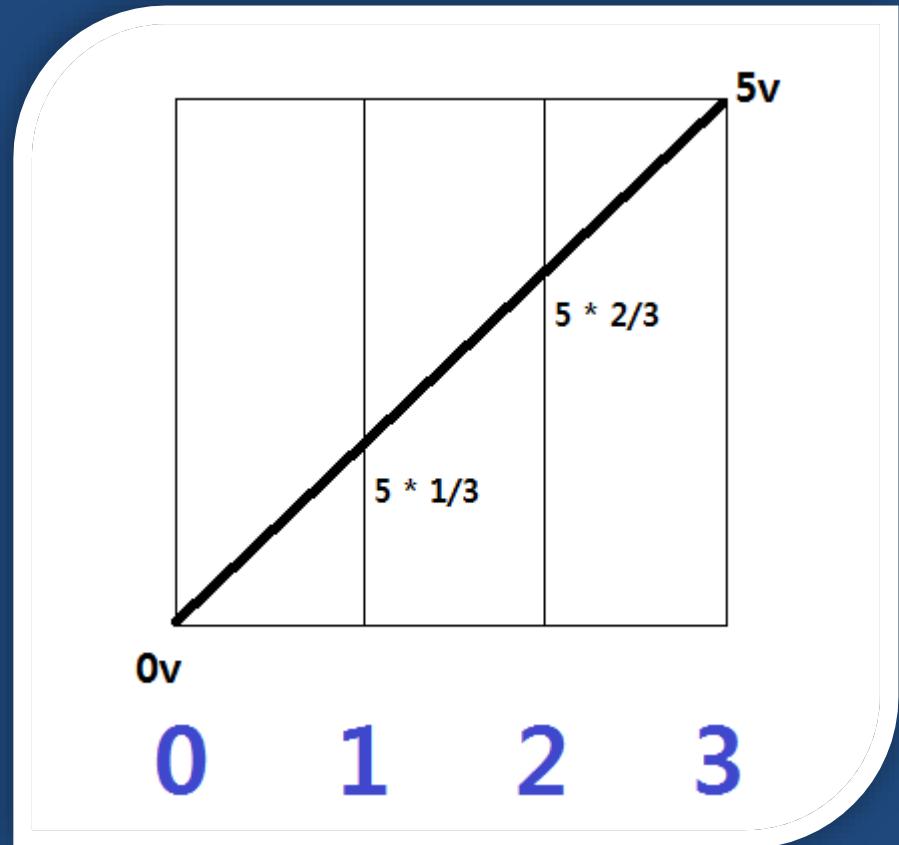
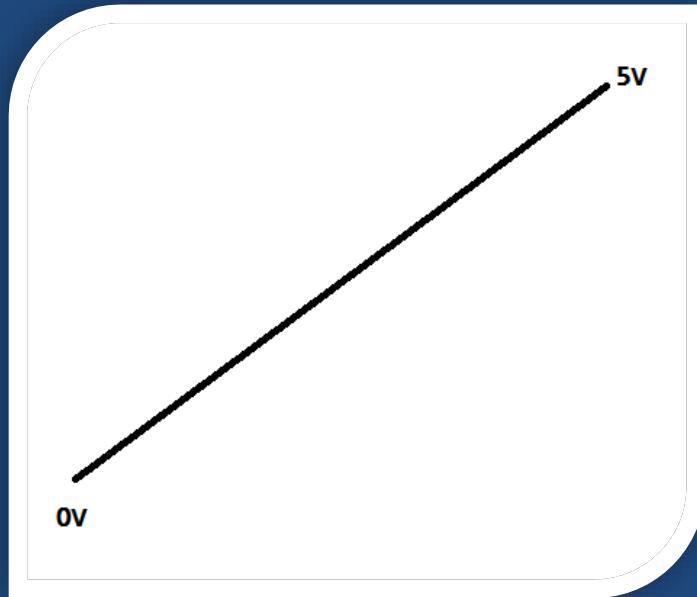
7	6	5	4	3	2	1	0
ADEN	ADSC	변환모드	인터럽트 설정	분주비 설정			

- ADEN : ADC 활성화
 - ADSC : ADC 변환 시작
 - 변환 모드 : 단일 변환
 - 인터럽트 : 사용 X
 - 분주비 : ADC로 전달되는 Clock 값 설정 (50~200khz)
- value*100/205 : 센서로부터 얻은 값을 섭씨 온도로 변환하는 공식

$$T(\text{temperature}) = \text{value} * 100 / 205$$

- 센서로부터 얻은 값을 섭씨 온도로 변환하는 공식
- $100\text{도} : 205 = T\text{도} : \text{value}$
 - $205 * T\text{도} = \text{value} * 100\text{도}$
 - $T\text{도} = (\text{value} * 100\text{도}) / 205$

기준 전압(5v)의 필요성



2비트 bit depth 예, 실제 Atmega128A의 bit depth는 10비트(1024)

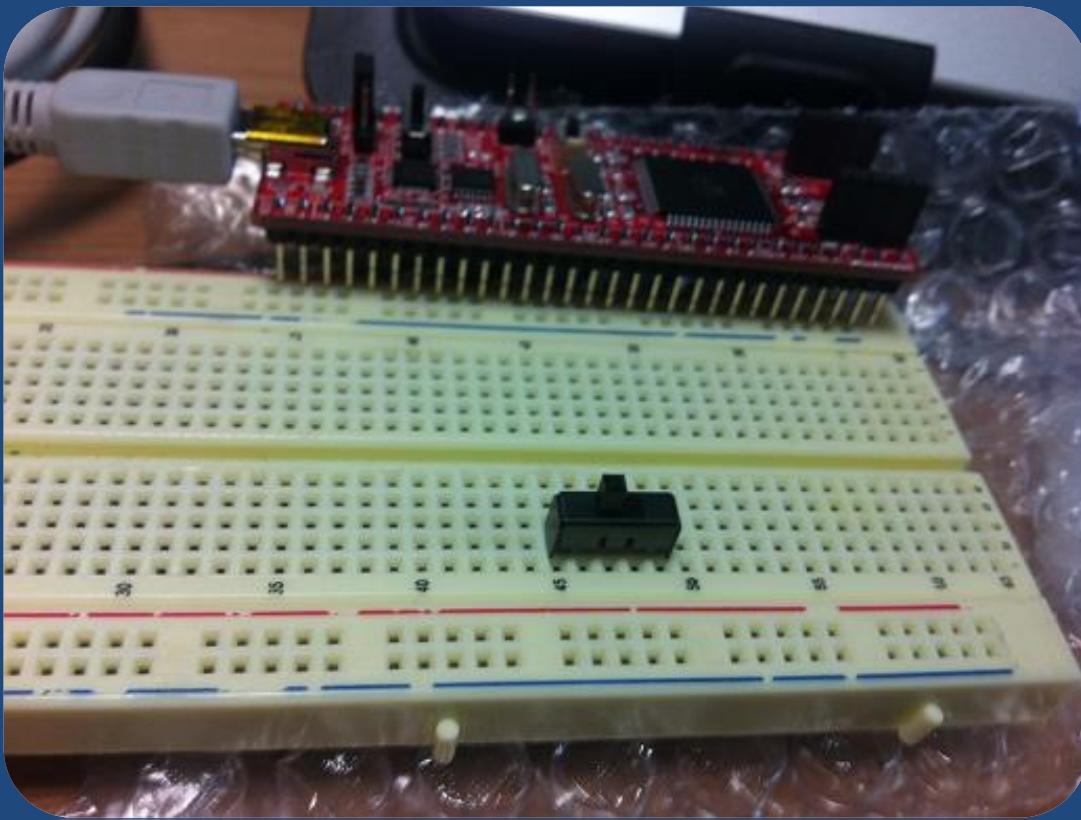
실습 문제

- 온도가 40도가 넘었을 때 부저가 경고를 울리게 해보세요.



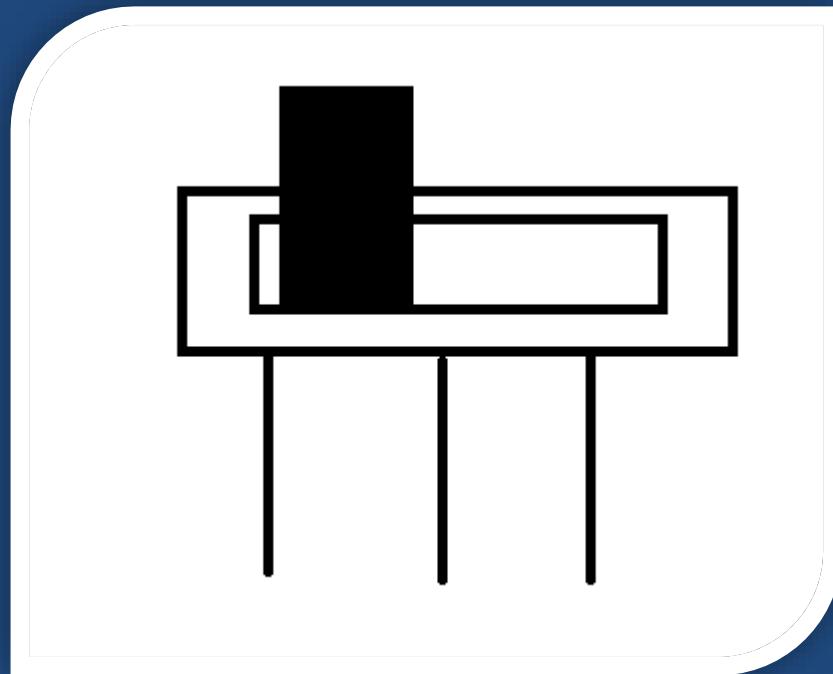
입력 핀 사용 실습

스위치 사용하기



스위치 사용 방법

- 스위치가 옮겨진 쪽의 두 핀이 서로 연결된다.



스위치 사용하기

- PB0에 전기신호 입력 시 PA0 켜기

```
#include <avr/io.h>

int main(void)
{
    int Input = 0;
    DDRA = 0xff;
    DDRB = 0x00;
    DDRC = 0xff;

    PORTC = 0xff;

    while(1)
    {
        Input = PINB;
        if(Input & 0x01)
            PORTA = 0xff;
        else
            PORTA = 0x00;
    }
}
```

실습 문제

- 스위치를 왼쪽으로 옮기면 LED의 불이 켜지고, 오른쪽으로 옮기면 LED의 불이 꺼지게 해보세요.

입력 핀 사용시의 주의사항

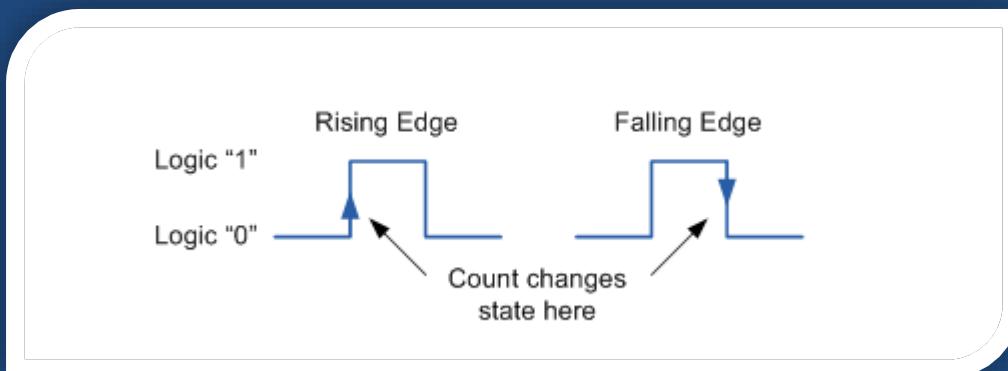
- 해당 핀은 항상 1(HIGH) 또는 0(LOW) 둘 중 하나의 상태가 되어야 함.
- 1또는 0이 아닐 경우엔 “무엇인지 알 수 없는 상태” 가 되어버림.
- Floating 상태
 - 0v도 5v도 아닌 아무런 입력이 가해지지 않은 상태.
 - 주변의 영향에 따라 어떤 값이 될지 모른다.

인터럽트

- 사전적으로는 방해하다, 중단시키다라는 의미를 가지며, MCU가 하고 있던 동작을 잠시 멈춘 후 다른 작업을 지시하는 기능
- 전기 신호가 변하는 시점을 탐지할 수 있음

Rising Edge와 Falling Edge

- 전기 신호의 Logic이 변하는 시점
- Rising Edge : LOW \rightarrow HIGH
- Falling Edge : HIGH \rightarrow LOW
- 데이터 동기화 작업에 사용 됨



좋지 못한 코드의 예

- PB0에 전기신호 입력 시 PA0 켜기

```
#include <avr/io.h>

int main(void)
{
    int Input = 0;
    DDRA = 0xff;
    DDRB = 0x00;
    DDRC = 0xff;

    PORTC = 0xff;

    while(1)          // 무한루프가 프로그램의 효율을 떨어뜨림
    {
        Input = PINB; // - CPU 클럭 소모
        if(Input & 0x01) // - while(1)문 안에 다른 시간이 오래 걸리는
            PORTA = 0xff; // 코드가 추가될 경우 Input 탐지 delay
    }
}
```

좋은 코드의 예

- INT0 입력 발생 시 PA0 켜기

```
#include <avr/io.h>
#include <avr/interrupt.h>

ISR(INT0_vect)
{
    PORTA = 0xff;
}

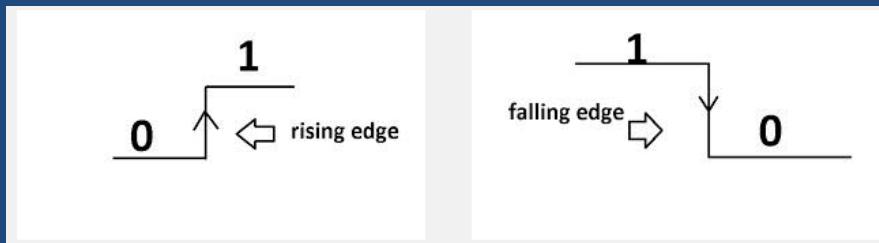
int main(void)
{
    DDRA = 0xff;
    DDRC = 0xff;
    PORTC = 0xff;

    EIMSK = 0b00000001; /* INTO ENABLE */
    EICRA = 0b00000011; /* INTO -> rising edge로 설정 */
    EIFR = 0b00000000; /* flag clear */
    sei(); // 인터럽트 기능 활성화

    while(1){ // 다른일 ... }
}
```

좋은 코드의 예

- EIMSK : MCU의 INT0 인터럽트 기능을 활성화시킴
 - EIMSK = 0b00000001 : 첫 번째 인터럽트 핀 활성화
- EICRA : 전기 신호가 들어오거나 나가는 시점 중 언제 감지할지를 결정
 - EICRA = 0b00000011 :



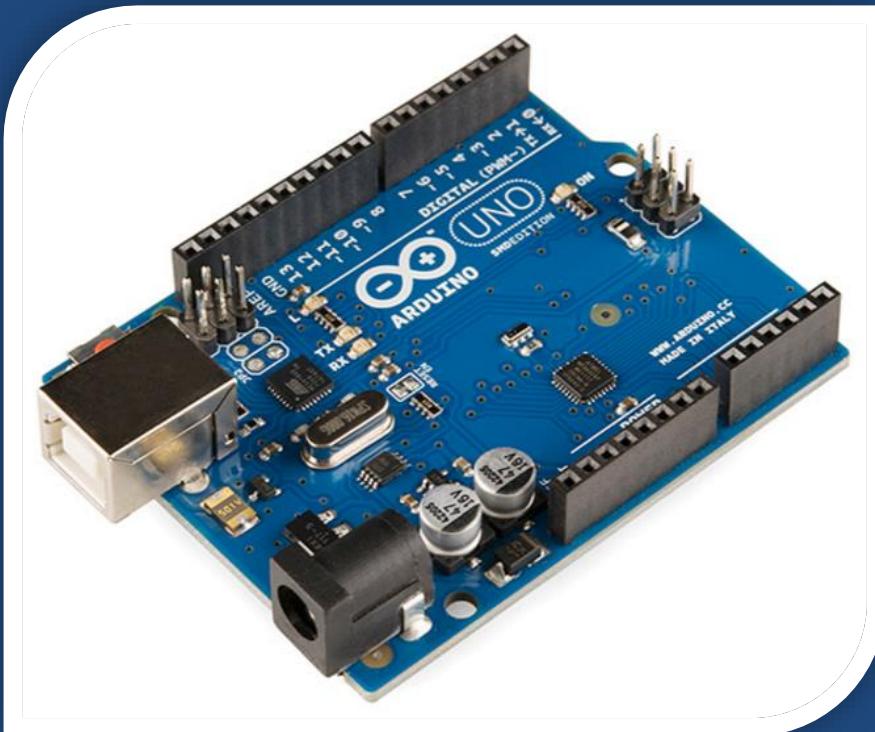
ISCn1	ISCn0	Description
0	0	The low level of INTn generates an interrupt request.
0	1	Any logical change on INTn generates an interrupt request
1	0	The falling edge between two samples of INTn generates an interrupt request.
1	1	The rising edge between two samples of INTn generates an interrupt request.

- EIFR : FLAG 레지스터를 초기화 시킴
- ISR(INT0_vect) : 인터럽트 신호 감지 시 이 함수가 자동 호출됨

실습 문제

- 스위치를 오른쪽으로 옮기면 LED의 불이 켜지게 해보세요.
- => 인터럽트를 사용하세요.

Arduino 프로그래밍



아두이노란?

- 2005년 이탈리아의 한 대학원(IDII)에서 개발된 하드웨어 개발 오픈소스 플랫폼
- 통합 개발 환경(IDE) 사용이 매우 쉬움
- 펌웨어 업로딩이 매우 쉬움(USB)
- 하드웨어 구매비용이 저렴함 (4000원~)
- 다양한 기본 예제 코드 존재
- 커뮤니티 및 라이브러리 생태계가 활성화 되어있음

다운로드 및 설치

- <https://www.arduino.cc/en/main/software>

The screenshot shows the Arduino Software download page. At the top, it says "Download the Arduino IDE". Below that is a large teal circle containing the Arduino logo (an infinity symbol with a minus and plus sign). To the right of the logo, there's a section titled "ARDUINO 1.8.3" with a brief description of what the IDE is used for. Further down, there's a "Support the Arduino Software" section with a call to contribute and some icons of electronic components. At the bottom, there are donation buttons for \$3, \$5, \$10, \$25, \$50, and "OTHER", along with two large red-bordered buttons: "JUST DOWNLOAD" and "CONTRIBUTE & DOWNLOAD". A large blue arrow points from the bottom left towards the "Windows Installer" button.

Download the Arduino IDE

ARDUINO 1.8.3

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac written in Java source software. This software is... Refer to the G instructions.

Windows Installer
WINDOWS ZIP file for non admin install

Windows app Get

Support the Arduino Software

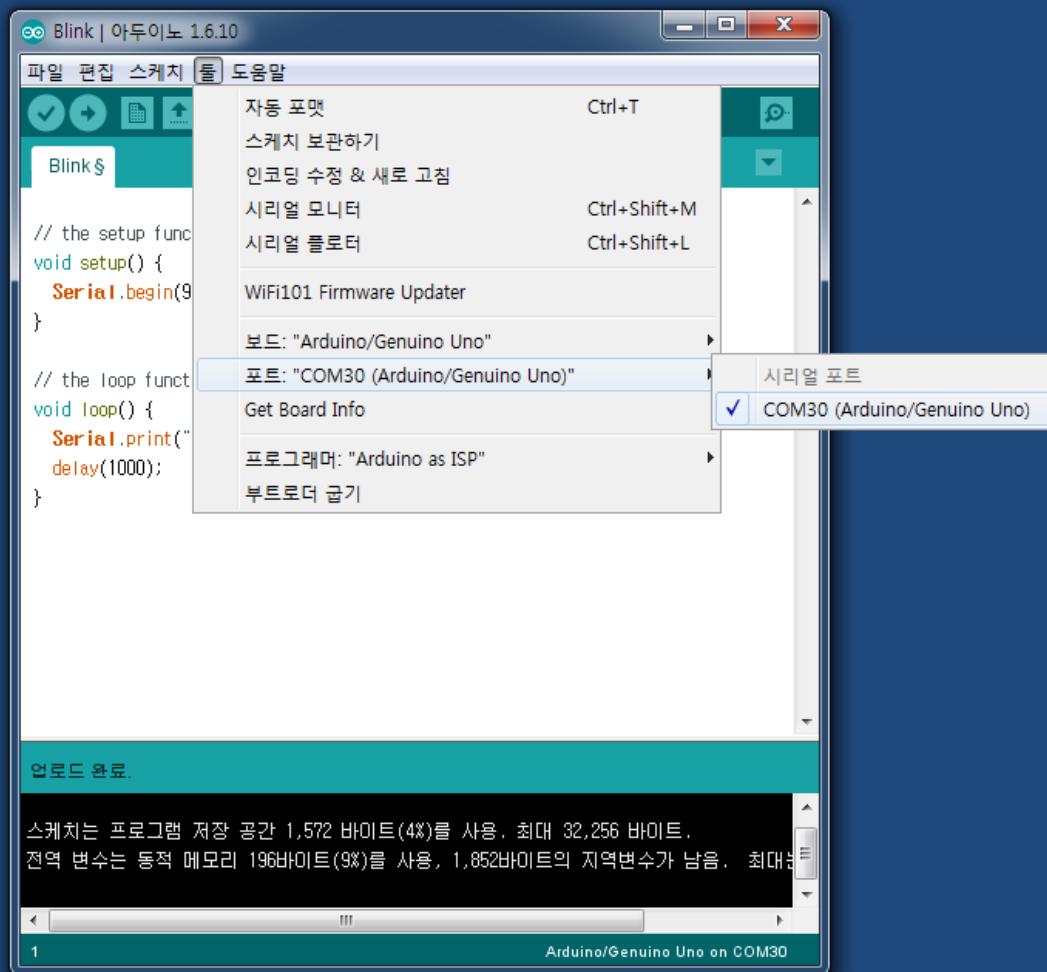
Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)

SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED **16,277,930** 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!

\$3 \$5 \$10 \$25 \$50 OTHER

JUST DOWNLOAD CONTRIBUTE & DOWNLOAD

보드 및 포트 설정

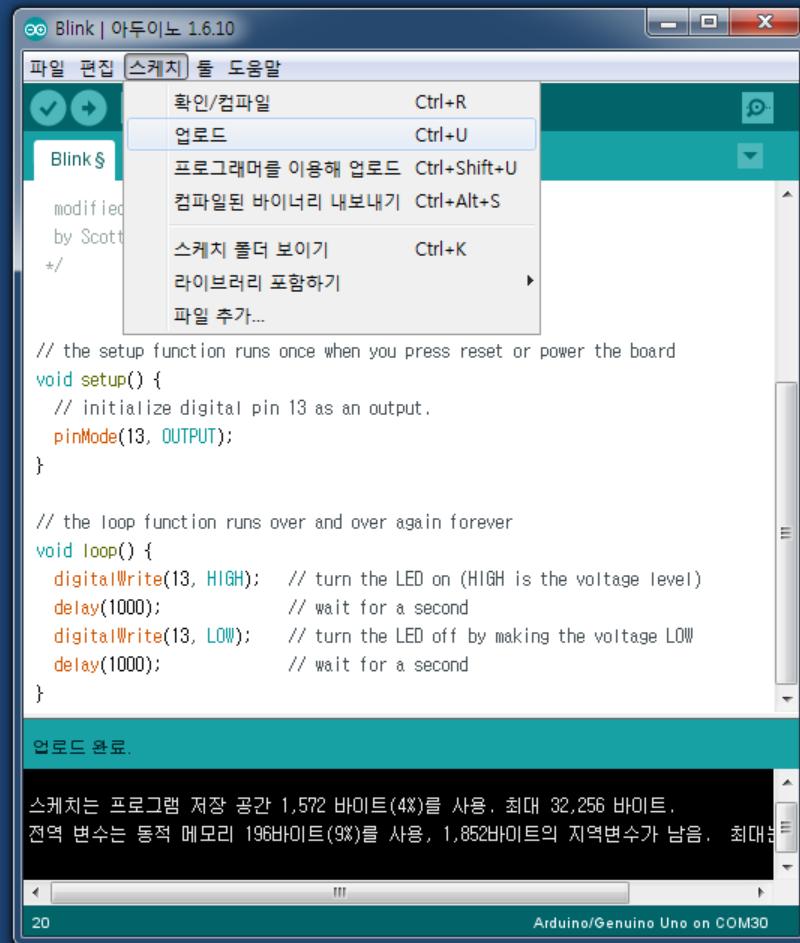


LED 점멸 테스트

```
// 초기화 함수
void setup() {
    // 13번 핀을 출력 모드로 설정
    pinMode(13, OUTPUT);
}

// main loop 함수
void loop() {
    digitalWrite(13, HIGH); // LED-on
    delay(3000);           // 3초 delay
    digitalWrite(13, LOW); // LED-off
    delay(3000);           // 3초 delay
}
```

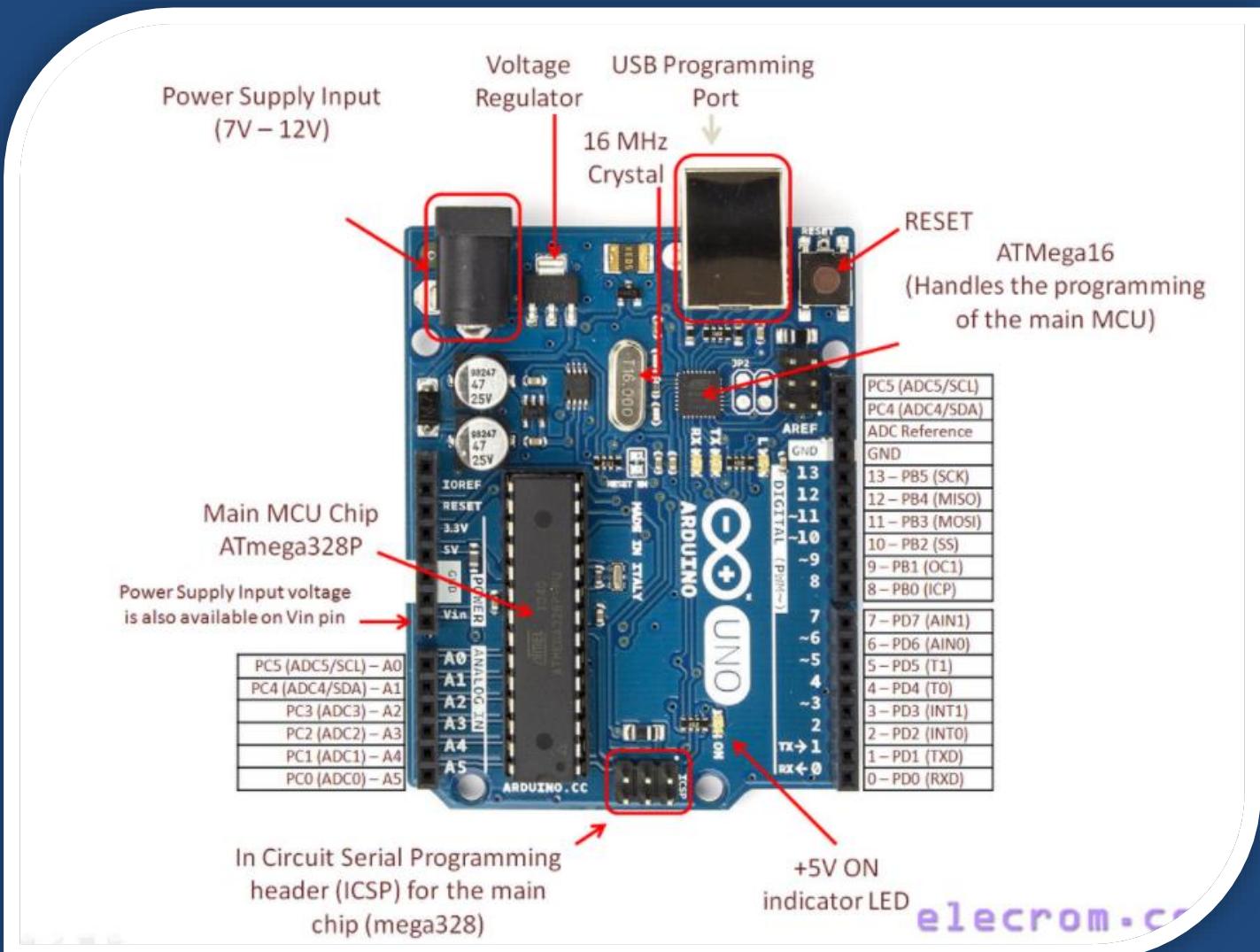
펌웨어 업로딩



핀 변경 실습 (D13 -> D2)

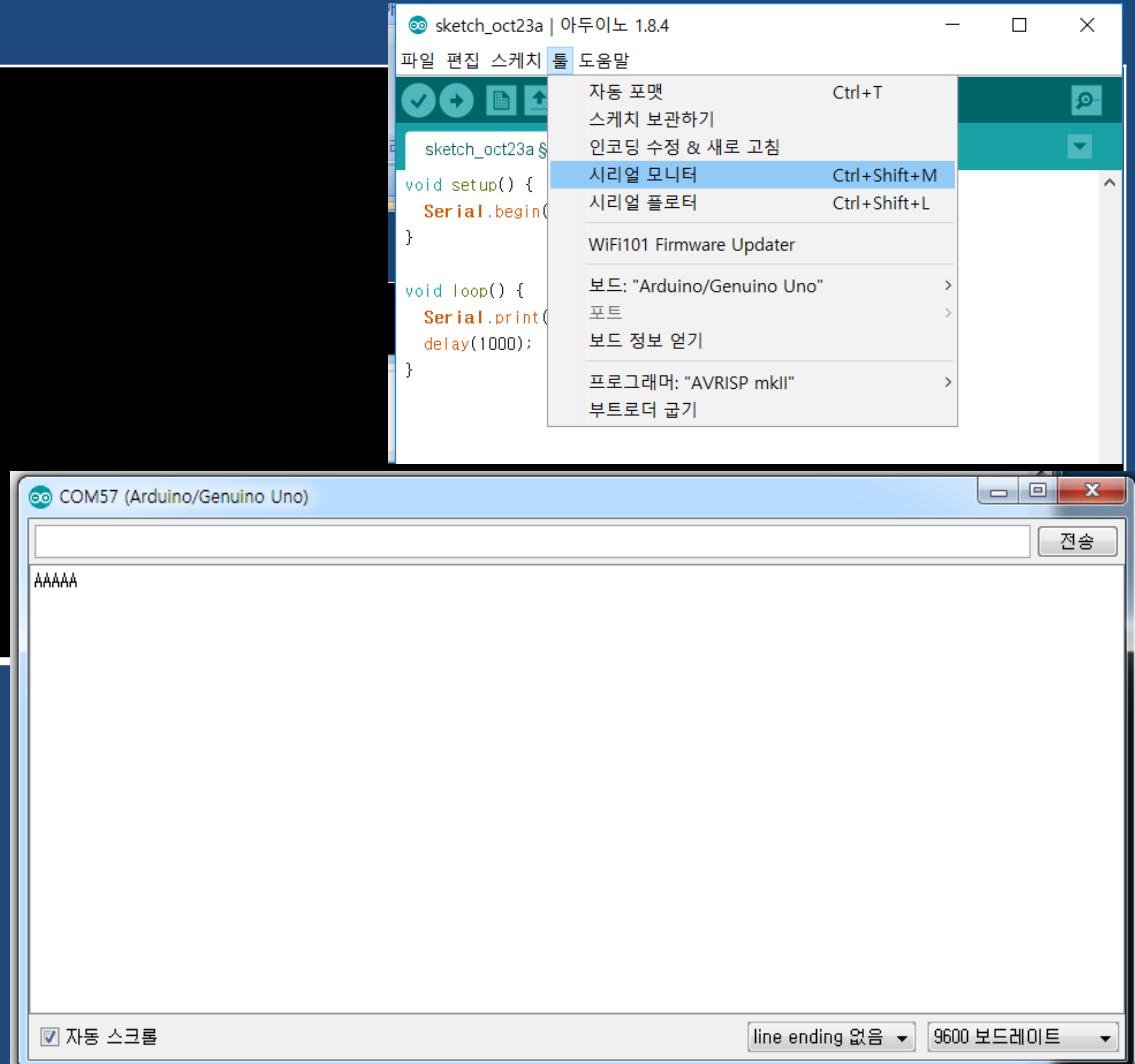
* LED 소자를 연결하여 확인

Arduino Pin map



Serial 출력 실습

```
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    Serial.print("A");  
    delay(1000);  
}
```

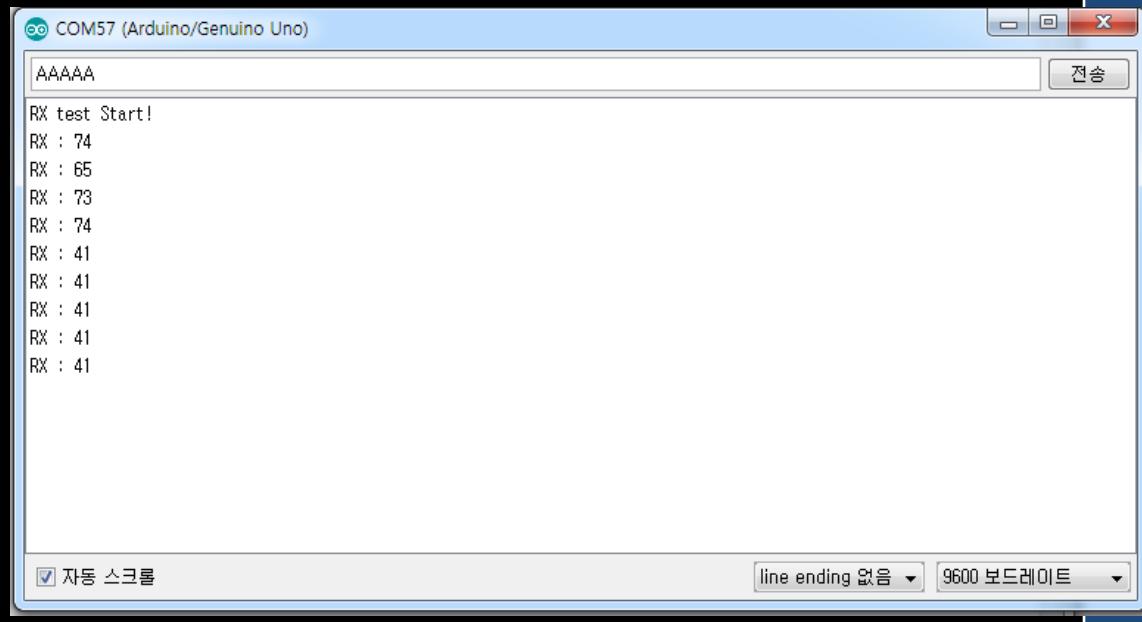


Serial 입력 실습

```
void setup()
{
    Serial.begin(9600);
    Serial.println("RX test Start!");
}

void loop()
{
    char ch;

    if (Serial.available())
    {
        ch = Serial.read();
        Serial.print("RX : ");
        Serial.println(ch, HEX);
    }
}
```



부저 스피커 사용하기

SPE-1404W-48

회로외장형 피에조 부저, 사이즈 : $\Phi 14.0 \times 4.0(h)$, Input : 10V, 크기 : 85db



위 상품 이미지는 참조용 대표 이미지이며,
정확한 사양은 데이터시트에서 확인하셔야 합니다.

큰이미지 보기

> 상품코드	32844
> 판매가격	360원 (부가세 미포함가)
> 제조사	KANGSEO EST
> 적립금	0원
> 평균준비기간	1~2일
> 최소주문수량	1 개
> 수량	<input type="text" value="1"/>

바로구매

장바구니

관심상품

이미지를 확장
을 확인하신 후에는 데이터시트에서 확인하세요.
이 유동 이미지는 참조용 대표 이미지이며,

바로구매

장바구니

관심상품

부저 스피커 사용하기

- 반복되는 전기 신호를 전달하여 진동 생성

```
void setup()
{
    pinMode(2, OUTPUT);
}

void loop()
{
    digitalWrite(2, HIGH);
    delay(1);

    digitalWrite(2, LOW);
    delay(1);
}
```

부저 스피커 사용하기

- 주파수와 음의 관계

	C	C#	D	Eb	E	F	F#	G	G#	A	Bb	B
0	16.35	17.32	18.35	19.45	20.60	21.83	23.12	24.50	25.96	27.50	29.14	30.87
1	32.70	34.65	36.71	38.89	41.20	43.65	46.25	49.00	51.91	55.00	58.27	61.74
2	65.41	69.30	73.42	77.78	82.41	87.31	92.50	98.00	103.8	110.0	116.5	123.5
3	130.8	138.6	146.8	155.6	164.8	174.6	185.0	196.0	207.7	220.0	233.1	246.9
4	261.6	277.2	293.7	311.1	329.6	349.2	370.0	392.0	415.3	440.0	466.2	493.9
5	523.3	554.4	587.3	622.3	659.3	698.5	740.0	784.0	830.6	880.0	932.3	987.8
6	1047	1109	1175	1245	1319	1397	1480	1568	1661	1760	1865	1976
7	2093	2217	2349	2489	2637	2794	2960	3136	3322	3520	3729	3951
8	4186	4435	4699	4978	5274	5588	5920	6272	6645	7040	7459	7902

8 4186 4432 4699 4978 5274 5588 5920 6272 6645 7040 7459 7902
1 5083 5211 5340 5480 5631 5764 5900 3136 3322 3520 3729 3951
0 1047 1109 1175 1245 1319 1397 1480 1568 1661 1760 1865 1976

부저에 “도”음 출력하기

- C(도) = 261.6hz = 대략 262

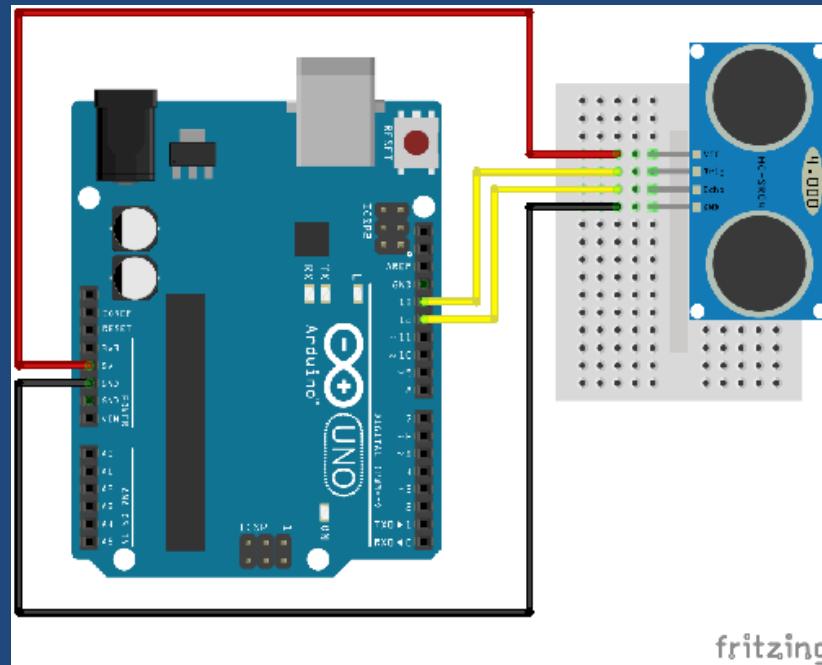
```
void setup() {  
    pinMode(2, OUTPUT); // D2핀 초기화 (OUTPUT)  
}  
  
void loop() {  
    tone(2, 262, 1000); // D2핀에 "도"음을 1초간 출력  
    delay(2000); // 2초 대기  
}
```

부저를 이용한 연주 예제

- 슈퍼마리오
 - <http://www.princtronics.com/supermariotunesong/>
- 스타워즈 다스베이더 테마
 - <https://gist.github.com/nicksort/4736535>

초음파 거리 센서

- 초음파를 발사한 후, 반사되어 돌아오는 시간을 재어 거리 측정
- Echo <-> D12
- Trig <-> D13



초음파 거리 센서

```
// 출처 : https://kocoafab.cc/tutorial/view/65
//초음파 센서의 핀번호를 설정한다.
int echoPin = 12;
int trigPin = 13;

void setup() {
    Serial.begin(9600);
    // trig를 출력모드로 설정, echo를 입력모드로 설정
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
}

void loop() {
    float duration, distance;

    // 초음파를 보낸다. 다 보내면 echo가 HIGH 상태로 대기하게 된다.
    digitalWrite(trigPin, HIGH);
    delay(10);
    digitalWrite(trigPin, LOW);

    // echoPin 이 HIGH를 유지한 시간을 저장 한다.
    duration = pulseIn(echoPin, HIGH);
    // HIGH 였을 때 시간(초음파가 보냈다가 다시 들어온 시간)을 가지고 거리를 계산 한다.
    distance = ((float)(340 * duration) / 10000) / 2;

    Serial.print(distance);
    Serial.println("cm");
    // 수정한 값을 출력
    delay(500);
}
```

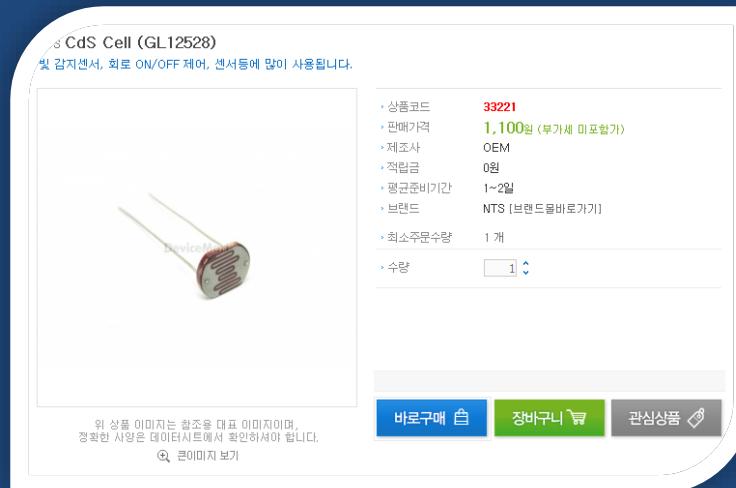
무선통신 실습

무선 통신의 원리

- 빛 감지 센서

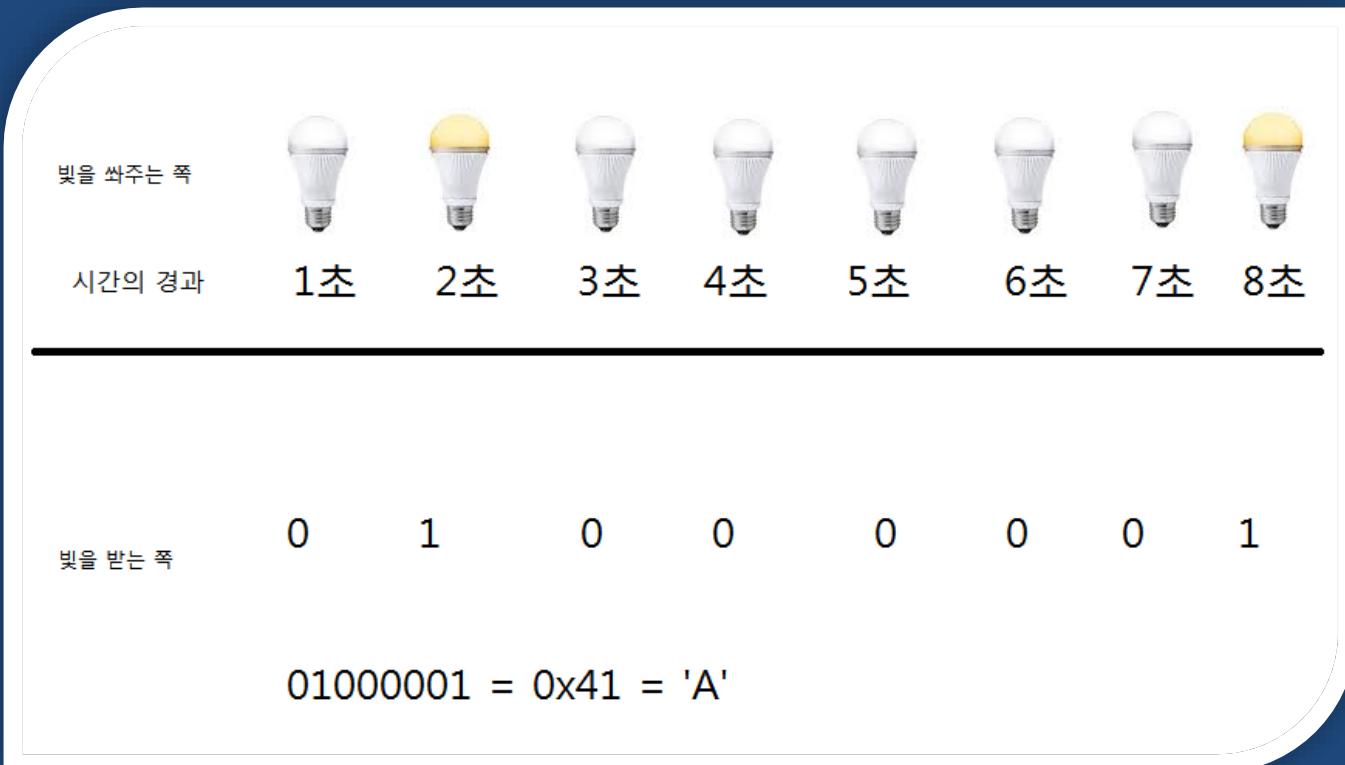
- 빛을 감지하면 저항이 작아져서 회로가 연결된다

- Ex> 빛이 있을 때 : 1 빛이 없을 때 : 0



- 그렇다면 빛의 on/off를 이용하여 무선 통신을 할 수 있지 않을까?

무선 통신의 원리



무선 통신의 원리

- 하지만 실제로 빛(가시광선)을 이용하는 것은 사람의 눈에 영향을 미치며, 항상 빛을 사용하고 있어야 한다는 단점이 있다.
- 가시광선보다 낮은 주파수 대역의 적외선을 이용
 - 사람의 눈에 보이지 않는다
 - 더 먼 곳까지 전파를 보낼 수 있다
- 적외선 송신부와 수신부(수광부)를 이용한다.

수광부 사용 실습

- 앞에서 볼 때 순서대로 OUT, GND, VCC
- 적외선 신호의 세기를 감지하는 것이 아니라 38Khz의 주기로 깜박이는 적외선에 반응하도록 설계되어 있다.
- 즉, 38khz의 적외선이 있다/없다로 반응한다.
- OUT의 값은 평상시에 HIGH, 신호 감지 시 LOW 값을

Any Vendor 603LM
수광부, CL-1L5 과 같이 쓰임



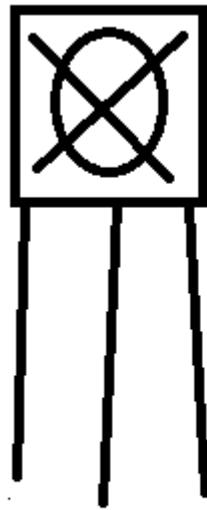
위 상품 이미지는 참조용 대표 이미지이며, 정확한 사양은 데이터시트에서 확인하셔야 합니다.

▶ 상품코드 **3178**
▶ 판매가격 **600원** (부가세 미포함)
1개 ~ 9개 600원
10개 ~ 99개 550원
100개 ~ 199개 520원
200개 이상 490원
▶ 제조사 LONGSAN (중국)
▶ 적립금 0원
▶ 평균준비기간 1~2일
▶ 브랜드 Any Vendor [브랜드몰 바로가기]
▶ 최소주문수량 1 개
▶ 수량 ▾

▣ 구로매장

바로구매  장바구니  관심상품 

수광부 사용 실습



적외선 없음



적외선 있음

적외선 수신

```
#include <IRremote.h>

// receive pin number
int RECV_PIN = 11;
IRrecv irrecv(RECV_PIN);
decode_results results;

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

void loop() {
    if (irrecv.decode(&results)) {
        Serial.println(results.value, HEX);
        irrecv.resume();
    }
}
```

수광부의 신호핀은 11번에 연결

IRremote Library

- <https://github.com/shirriff/Arduino-IRremote/archive/master.zip>

참고

- <https://kocoafab.cc/tutorial/view/78>

적외선 송신부 사용 실습

- VCC/GND
- 38khz의 신호를 만들어서 전달한다

Any Vendor CL-1L5
발광부, 603LM 과 함께 쓰임

> 상품코드	3177
> 판매가격	180원 (부가세 미포함)
> 제조사	DAKWANG (중국)
> 적립금	1원
> 평균준비기간	1~2일
> 브랜드	Any Vendor [브랜드몰 바로가기]
> 최소주문수량	1 개
> 수량	<input type="text" value="1"/> <input type="button" value="▼"/>

위 상품 이미지는 참조용 대표 이미지이며,
정확한 사양은 데이터시트에서 확인하셔야 합니다.

적외선 송신

```
#include <IRremote.h>

IRsend irsend;

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

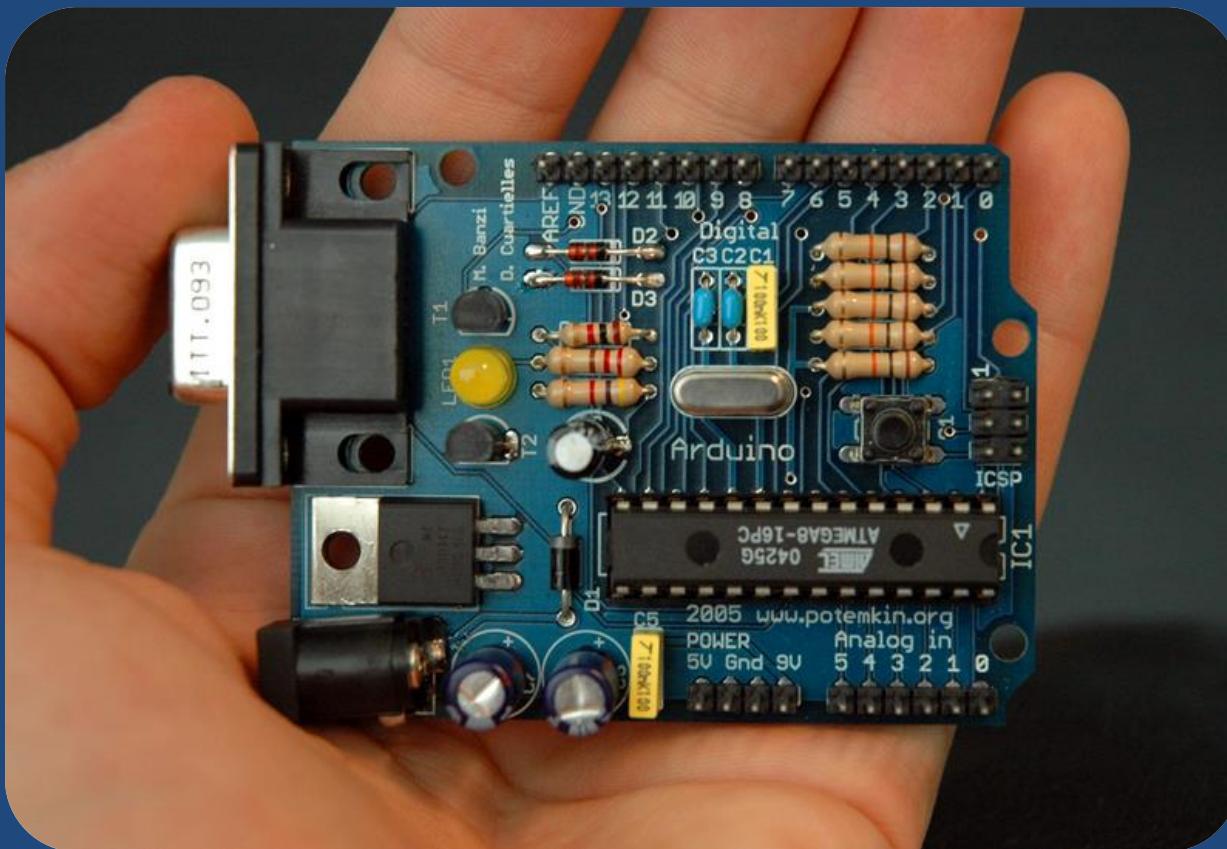
void loop(){
    for(int i = 0; i < 3; i++){
        irsend.sendSony(0x0a90, 16); // Code, Bit length
        delay(40);
    }
    delay(1000);
}
```

적외선 LED 핀은 3번에 연결

참고

- <https://kocoafab.cc/tutorial/view/76>

AVR과 아두이노



AVR과 아두이노



- 요리를 직접 하느냐?
다 되어있는 걸 데워먹느냐의 차이
- 학습 기간 중엔 되도록이면 AVR을 다루길 권장!

감사합니다.