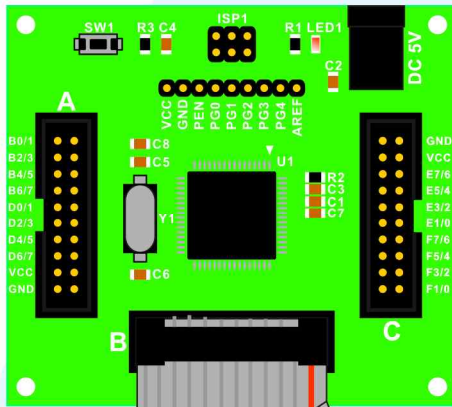
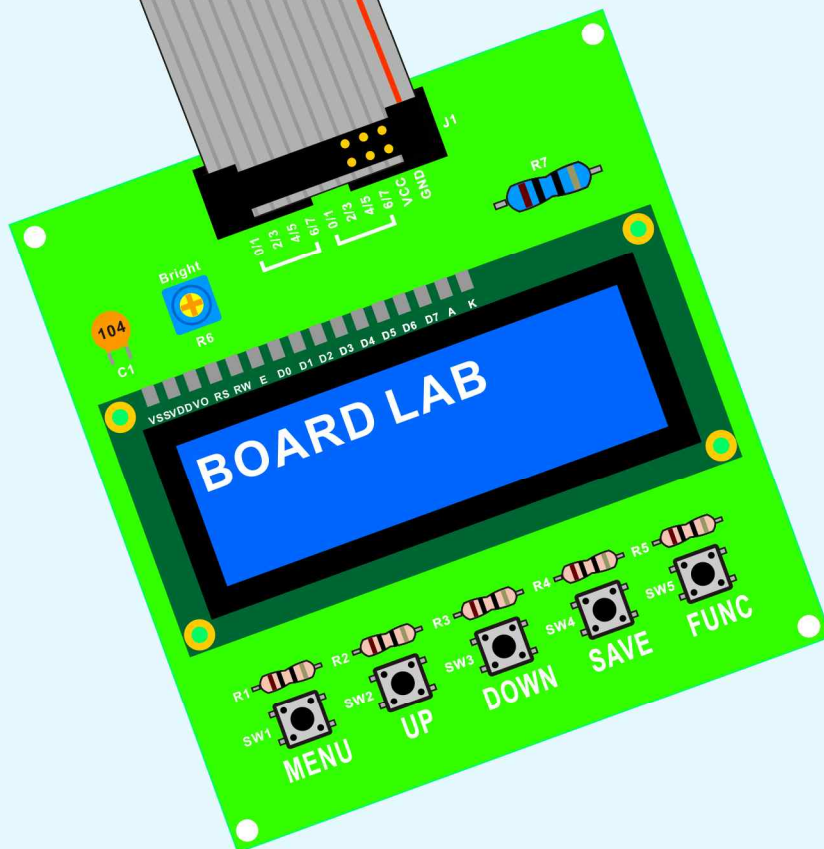


선배가 취업나와 보니까 꼭 필요하더라

전자분야 취준생을 위한 실무 프로젝트



- 실무 프로젝트를 활용한 교육
- 스스로 생각하는힘 형성
- 다양한 예제를 통한 실력향상
- 실무 경험 꿀팁 전달



www.BoardFree.kr

저자:김명수.이소리 공저 / 감수 : 김용필

차례

01. MAIN BOARD

ATmega128 개요
컴파일러
ISP
메인보드
보드 to 보드 연결법

02. LED MATRIX

기본설명, 예제
프로젝트 1 : 미니 전광판
프로젝트 2 : 통신 전광판

03. FND

기본설명, 예제
프로젝트 1 : 전자시계
프로젝트 2 : 모터제어기
프로젝트 3 : 스텝모터제어기
프로젝트 4 : 전자온도계
프로젝트 5 : 온도감지 후드
프로젝트 6 : 온도감지 밸브

04. LCD

기본설명, 예제
프로젝트 1 : 전자시계
프로젝트 2 : 모터제어기
프로젝트 3 : 스텝모터제어기
프로젝트 4 : 전자온도계
프로젝트 5 : 온도감지 후드
프로젝트 6 : 온도감지 밸브

05. 만능기판 사용법

기본설명, 예제
제어 루프 프로그램
Buck타입을 이용한 전압제어
Boost타입을 이용한 전압제어

06. 자료 정리

출처모음

01. MAIN BOARD

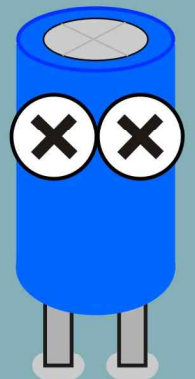
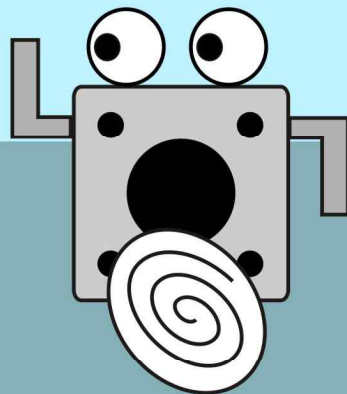
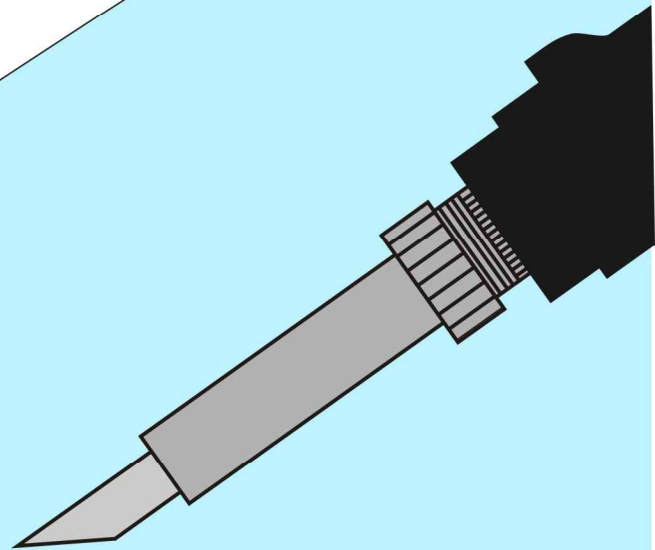
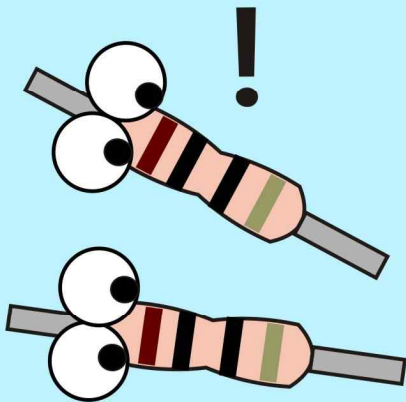
ATmega128 개요

컴파일러

ISP

메인보드

보드 to 보드 연결법



1

ATmega128 개요

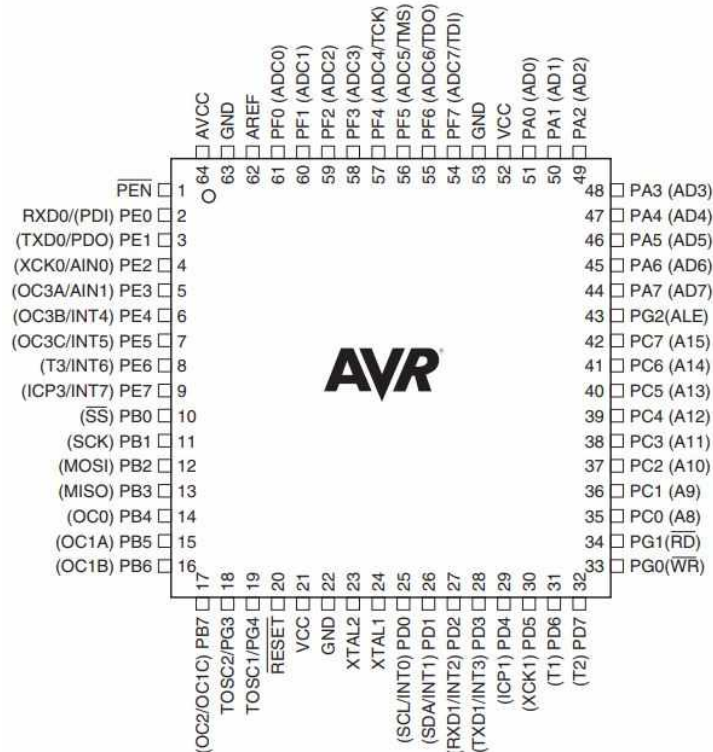
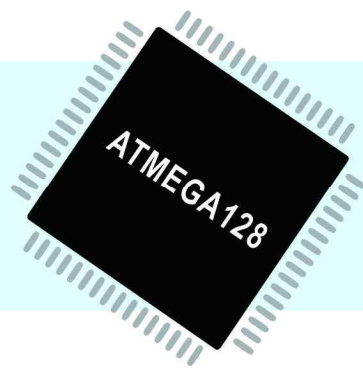


그림1. ATmega128

1.1 ATmega1280이 뭔지 아니?

안녕, 우리는 취업을 나와 직장 생활을 하고 있는 직장인 선배야. 우선 너희가 왜 ATmega128을 사용하는지 알고 있니? 나도 학교 다닐 때는 아무 생각 없이 주는 대로 쓰다가 회사에 입사한 뒤에야 알게 되었어. 그러고선 느낀 게 어떤 칩을 왜 사용하는지, 그 칩에 특징을 알아야 상황에 맞춰서 스스로 사용해볼 수가 있고 그래야 엔지니어의 길에 한 발자국 다가갈 수 있는 거더라고.

칩을 선정하는 방법 중 가장 중요한 것은 자신이 어떤 기능을 구현할 것인지에 따라 어떤 핀이 필요한지를 고려하는 거야. 일단 아무 칩이

나 골라서 납땀도 다 했는데 정작 네가 쓰려는 기능이 들어있지 않은 칩이면 얼마나 바보 같은 상황이겠어? 그런데 왜 하필 ATmega128을 쓰는가 하면 가장 다양한 기능을 효과적으로 구현할 수 있기 때문이야. Tiny시리즈, Mega 시리즈, AT90 시리즈 등 다양하게 있지만, 가격과 기능, 너희가 사용할 코드의 양 등을 고려해서 ATmega128을 사용하게 된 거야.

1.2 ATmega128의 기능 및 성능

AVR은 ATMEL 사의 Micro Control Unit(MCU)으로 Tiny 시리즈, Mega 시리즈, AT90 시리즈를 아우르는 말이야. 이러한 AVR의 특징으로 첫 번째는 저 전력 소모, 두 번째는 저렴한 가격, 세 번째는 높은 성능이 있어 한마디로 효율이 좋다는 말이지.

그중 ATmega128이 속한 Mega 시리즈는 100핀 이하 20핀 이상의 핀 수를 가지고 있고, 메모리도 다른 종류의 AVR보다 크기 때문에 가격도 높지만 다양한 기능을 구현할 수 있어. Mega시리즈 중에서도 ATmega128은 성능이 높은 축에 속해 EEPROM과 SRAM은 4Kbyte로 상당히 크고 Flash메모리도 128KByte나 되기 때문에 다양한 코드를 작성해볼 수 있어.

이런 ATmega128에는 ADC, USART, PWM 등의 여러 가지 기능들이 있는데 너희 혹시 데이터 시트라는 말을 들어봤니? 데이터 시트란 간단하게 말해서 칩 사용설명서야. 칩을 만든 사람이 이 칩은 이렇게 사용해야 한다고 설명해 놓은 거지.

물론 너희가 배우고 직접 사용해보면서 외워지기도 하겠지만 그걸 몇 년이고 계속 기억할 수는 없어. 그렇기에 너희는 Datasheet를 읽고 사용하는 습관을 들여야 해. 어차피 모든 부품은 Datasheet가 있을뿐더러 몇 년이 지나도 데이터시트가 없어지진 않거든. 혹시 그건 알고 있니? 저항, 캐패시터도 데이터시트가 있다는 걸.

www.BoardFree.kr

다음은 코드비전에서 Build 버튼을 누르면 컴파일된 어셈블러가 기계어로 링크 되어 .hex파일인 기계어가 나오게 되는거야. (어셈블러를 기계어로 바꿔주는 장치를 링커라고 해) 이제 이 파일을 ATmega128에 다운로드 할 거야. 프로그램을 다운로드하기 위해선 ISP (In-System Programmable)라는 장치가 필요해 ISP에 대한 설명은 뒤에서 하니깐 지금은 이름만 알아둬.

ATmega128에 다운로드 할 땐 우리는 주로 AVR studio 4를 써 5,6도 있지만, 그 둘에 비해 사용하기가 가벼워 사용하기 편할 거야. AVR studio 4는 STK500, AVR MK2, AVR OSP2 등의 프로그래머 킷(ISP의 종류)을 사용해.

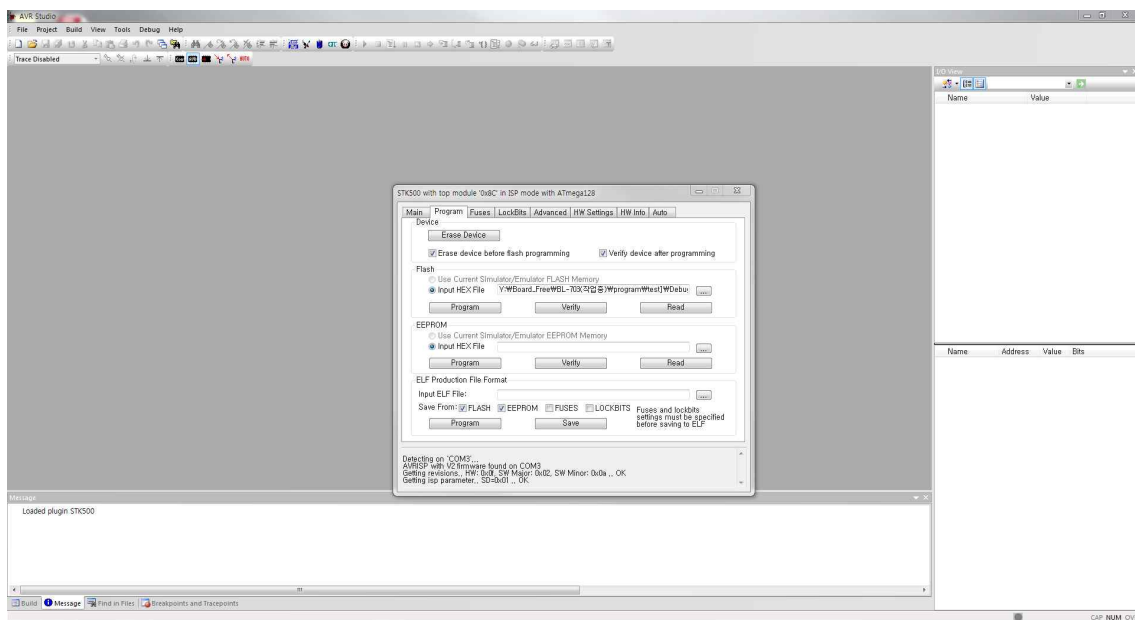
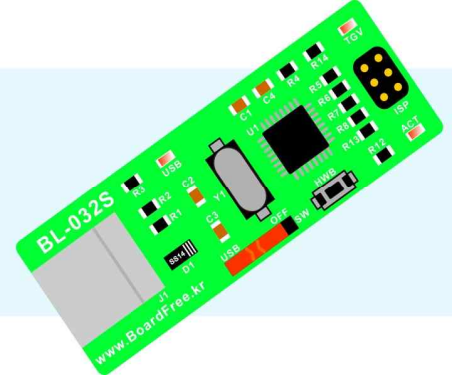


그림2-2. AVR studio 4 화면

3

ISP



3.1 ISP를 본적이 있지 않니?

ISP가 뭐지? 하는 생각이 들을 수도 있겠지만 잘 생각해봐. 프로그램을 짜본 적이 있다면 누구나 본 적이 있을 거야. 혹은 볼 기회는 있었지만 무심코 지나가는 일이 많았을 거야. ISP(In System Programming)는 쉽게 말해 너희가 짠 프로그램을 MCU에 프로그래밍을 해주는 것을 말해. 우리는 Atmel 사에 칩을 사용하니까 그에 맞는 AVR ISP를 사용할 거야. AVR ISP의 종류는 MK-II, STK500, AVR 911 등 여러 가지가 있어. 그중 사용해보고 각자 좋다 생각하는 거로 사용하면 돼.

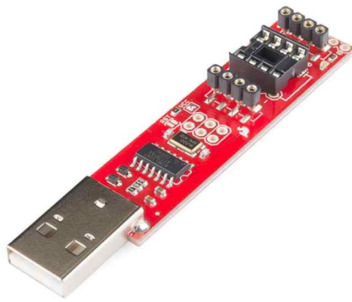


그림3-1. ATtiny45, ATtiny85 - Programmer

3.2 ISP는 꼭 사야하는 걸까?

시중에 파는 MK-II, STK500 등 AVR ISP의 가격은 성인에게도 부담스러울 정도의 가격이야. 그런데 직장인도 아닌 너희보고 사서 쓰라면 더 부담스럽겠지? 그래서 너희를 위해 하나를 소개하려고 해. 너희가 이 책을 보고 있다면 알겠지만 BoardFree 사이트에서는 ISP 보드도 무료로 나눠주고 있어. 너희에게 부담스럽지 않고 직접 만들어보기도 할 수 있는 안성맞춤의 사이트라고 생각해. 사이트 홍보는 짧게 하고 ISP를 어떻게 만들면 되는지 설명해줄게.

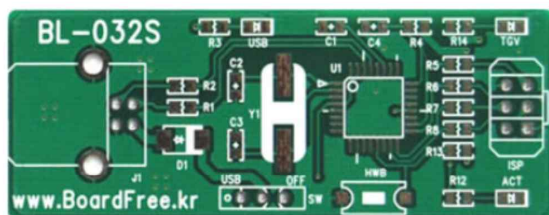
3.3 AVR ISP,어떻게 만들까?

먼저 어떻게 만들지 설명하기 전에 주의사항! 이 사이트는 좋은 목적으로 운영하는 만큼 우리가 지켜줘야 할 몇몇 가지 에티켓이 있어. 홈페이지에 들어가면 가장 먼저 나오는 내용들이 있지? 그걸 꼭 먼저 읽어보고 필요한 내용을 찾아보길 바래. 그래야 서로 간의 예의가 지켜지면서 담당자분들도 기분 좋게 계속해서 사이트를 운영할 수 있고 받는 너희들도 무료로 공부할 기회를 주는 사이트를 계속 이용할 수 있잖아? 자, 이쯤이면 모두 이해했을 거라 생각하고 AVR ISP를 만드는 방법을 알려줄게.

1. 일단 <http://boardfree.kr/> 사이트에 들어가.
2. 그리고 가장 첫 페이지의 내용을 꼭 읽어봐!
3. 다 읽었으면 상단의 ISP,ICSP,JTAG 로 들어가.



4. 들어갔으면 AVR - ISP 중에 너희가 맘에 드는 보드를 고르고 첫 페이지에 양식에 맞춰 보드를 신청해.
5. 신청한 보드를 받았다고 치고 해당 보드 회로도를 열어 조립을 해 (나는 임의로 MK-II로 할게)




품명 : AVR ISP BL-032S
가격 : 무료 입니다.

AVR ISP MK-II 호환 보드로 ATmega 시리즈의 ISP만 지원하는 프로그래머이며, 최소한의 부품으로 쉽게 만들 수 있는 프로그래머입니다. 처음 ISP를 만드는 사용자도 별도의 장치나 프로그래머 없이 USB 케이블 연결만으로 펌웨어를 입력할 수 있으며 5V와 3.3V 로 동작하는 보드를 지원합니다. FLIP을 이용한 펌웨어 입력방법에 대한 설명이 필요 하다면 [여기](#)를 클릭 하시기 바랍니다.

MK-II 드라이버는 AVR Studio를 설치할 때 USB 드라이버 설치옵션을 체크하고 설치 하시면 되고, 자동으로 인식 합니다. 제작이 쉽고, 부품수가 적으며, 펌웨어 다운로드가 쉽고, 퓨즈설정이 필요없고, 저렴하게 만들 수 있는 보드로서 보드프리에서 입문자에게 추천하는 보드 입니다.



6. 조립을 끝냈으면 아래 IC 모양을 눌러 안에 있는 내용을 복사해서 메모장에 붙여 넣거나 오른쪽 마우스를 클릭해 다른 이름으로 저장을 눌러

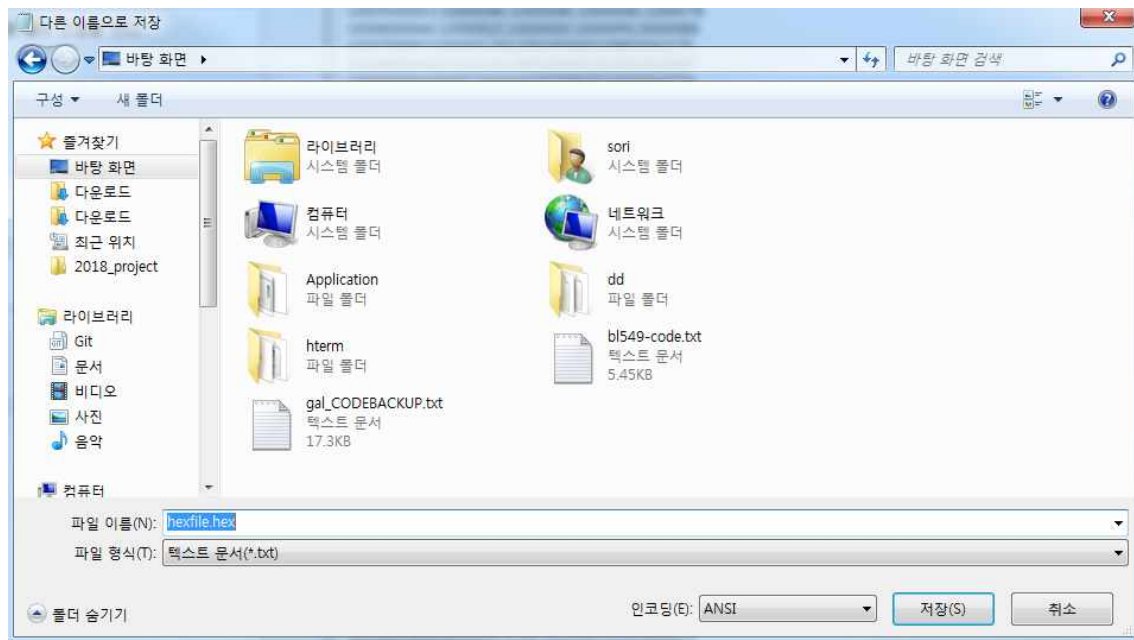


```

101010
01
: 10000001 8C1 000032C1 000030C1 00002EC1 000044
: 1001 00022C1 00002AC1 000028C1 000026C1 000038
: 100220024C1 000022C1 000020C1 0000C942A1 04D
: 100330001 CC1 0000AEC1 000018C1 0000ABC70 00029
: 100440001 4C1 00001 2C1 000010C1 00000EC1 000068
: 100550000CC1 00000AC1 000008C1 000006C1 000078
: 1006600004C1 000002C1 000000C1 0000FEC00 00089
: 10077000FC0000001 201 1001 FF00001 0EB030 421 7E
: 10088000000201 020301 0902200001 01 00C03 2093F
: 1009900004000002FF0000000706820240000A077A
: 100AA0000502024000A040309040C0341 00540045
: 100BB0004D0045004C0000001 80341 00560052005E
: 100CC00049005300500020060006800490049006A
: 100DD000000001 C033 00030003000320030003 0000F
: 100EE000300031 003200300303400350000000 000E1
: 100FF00009000A000A000B000C000D000E0000E00A4
: 1001 00000F001 0001 0001 2001 3001 3001 4001 6005E
: 10011 0001 6001 8001 9001 A001 C001 D001 F002 00006
: 1001200220023002500270029002B002E003 00008C
: 100130003200340037003A003D00400043004600E2
: 100140004A 004E00520055005A005E0063006800ED
: 100150006D00730078007E0084008B0091 00990090
: 10016000A000A900B1 00B900C300CC00D600E1 0096
: 10017000EC00F7000401 1101 1E01 2D01 3B01 4B01 B0
: 100180005B01 6C01 7D01 9101 A501 BA01 CF01 E601 7E
: 10019000FE01 160231 024C026A028C02A802CA025B
: 1001A000ED021 03380360038B03B803E7031 8045F
: 1001B0004B0481 04B904F50434057505B90501 063D
: 1001C0004E069006EF06A807A30704086508D 008F9
: 1001D0004009B6092B0A80A300B0C30B51 0CE0C0C
: 1001E0008F0D400EF60EAF0F681 0471 1141 2F51 266
: 1001F000ED13E214F015F91 61B1 85C1 991 1AE51BA2

```

7. 내용을 붙여 넣었으면 ‘영어로 쓴 파일 이름.hex’의 hex파일로 저장하면 돼.



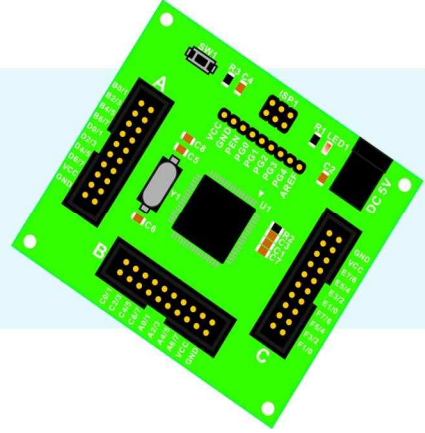
8. **여기**를 들어가면 있는 USB 케이블로 프로그램 입력하기 (FLIP 사용방법)을 따라하면 돼.(7에서 만든 .hex파일은 여기서 사용 됨)

9. AVR-ISP 완성!

만약 이대로 했는데 안된다면 납땜 문제, 드라이버 문제, 시키는 대로

4

메인보드



4.1 회로도

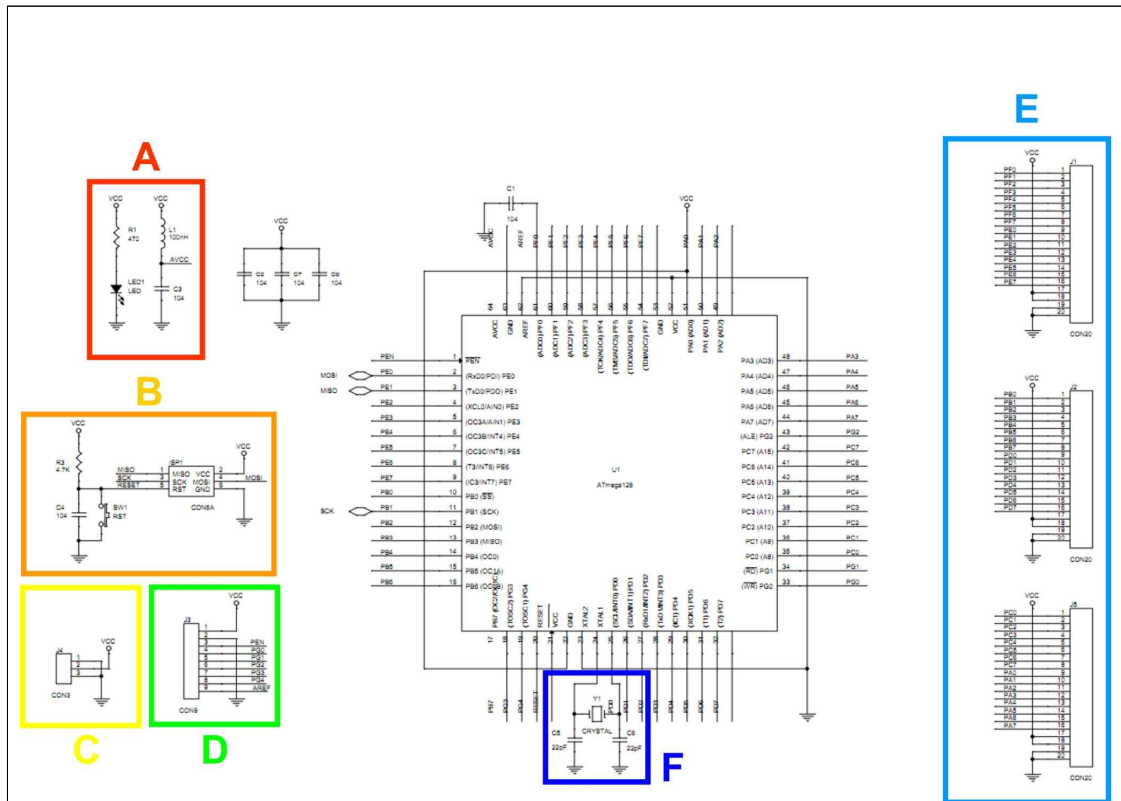


그림 4-1. 메인보드 회로도

회로도라 하니까 ‘어 너무 어려울 것 같은데’라고 생각하는 사람 있을 거야. 그래서 쉽게 이해할 수 있도록 설명해주려고 해. 단! 앞에서 말했었듯이 데이터 시트를 읽었어야 이해하기 쉽다?

A. LED와 AVCC전원

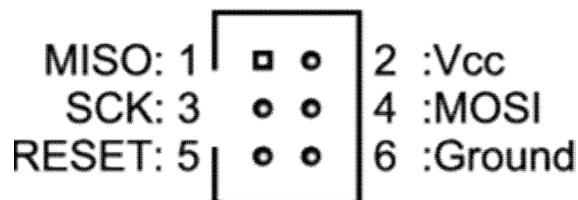
LED는 보통 어떤 신호를 나타내기 위해 많이 사용하지? 예를 들어 뭔가 위험할 때 빨간 LED를 깜빡깜빡, 동작 중일 때 초록 LED를 ON. 이렇듯이 LED1도 전원이 들어왔을 때를 의미해. LED 위쪽에 VCC가 들어가는 게 보이지? VCC 즉, 전원이 들어오지 않으면 LED는 꺼질

데고 전원이 들어오면 LED는 켜지겠지. 쉽지?

AVCC는 뭘까? 데이터 시트를 읽어본 사람이라면 분명 알 거야. AVCC는 나중에 배우게 될 A/D 변환에서 사용될 전원이야. ADC를 사용하거나 혹은 ADC를 사용하지는 않지만, ADC에 해당하는 핀을 사용 (SW or LED 등의 용도)할거면 AVCC에 전원을 꼭 인가해야 해. A/D 변환에서는 전압 변화에 매우 민감하기 때문에 전원을 따로 분리해 놓은 거야. 왜 코일이랑 캐패시터로 회로를 구성했냐고 묻는다면 이것도 데이터 시트에 나와 있어. 어때 데이터시트를 왜 읽어야 하는지 확 와닿지 않아?

B. ISP

ISP가 무슨 용도로 사용하는지는 알고 있을 거라고 믿어. 너희 $\overline{\text{RESET}}$ 처럼 위에 바가 붙으면 무슨 의미인지 알지? 바가 붙으면 0을 줘야 1이고 바가 붙지 않으면 1일 때 1이야. 그래서 그림 좌측에 SW가 눌리면 $\overline{\text{RESET}}$ (바가 붙어있으니까 0일 때 1) 이 접지로 붙어서 0이 들어가는 거지. 옆에 ISP는 핀 맵이 이미 정해져 있어. 정해져 있는 핀 번호대로 MCU랑 이어주면 되는 거지.



4-2. AVR ISP connectors 핀 맵

C. 전원

전원은 그냥 VCC,GND를 이어주면 되는 건데 우리가 쓰려는 JACK이 3핀짜리라 1개는 VCC,2개는 GND로 이은 거야. 혹시 이런 고민을 하는 애가 있니? 몇 V를 인가해야 할까 하고. 데이터 시트 앞장에 MCU 동작 전압이 쓰여 있는데 그냥 그거에 맞춰서 결정하면 돼.

D. 남은 핀들

정말 말 그대로 남은 핀들을 모아 놓은 거야. 아 이 중에 AREF는 AVCC처럼 A/D 변환에서 쓰이는데 이거의 전압은 너희가 바꿔서 사용할 수 있도록 따로 빼놓은 거니 혹시 필요하거든 사용하도록 해!

E. 박스 헤더

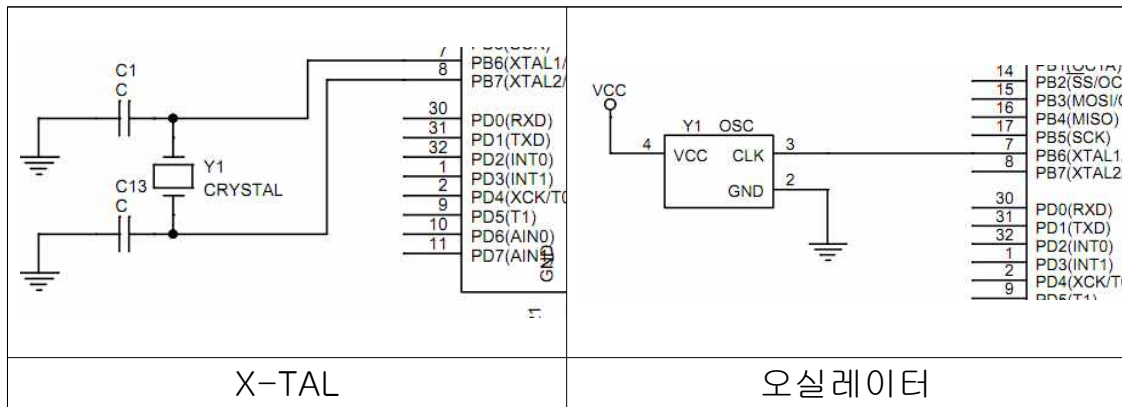
박스 헤더를 20핀짜리로 3개가 있는데 이건 나중에 다른 보드를 이을 때 사용하려고 만들어 놓은 거야. 예를 들면 너희가 A/D 변환을 하려는데 그 핀이 PORTF에 있다. 그럼 그 해당하는 박스 헤더를 이어서 연결하면 되는 거지. 박스헤더 연결은 다시 한 번 뒤에서 설명할 거니 너무 복잡하게 생각하지는 말어.

F. CRYSTAL(X-TAL)

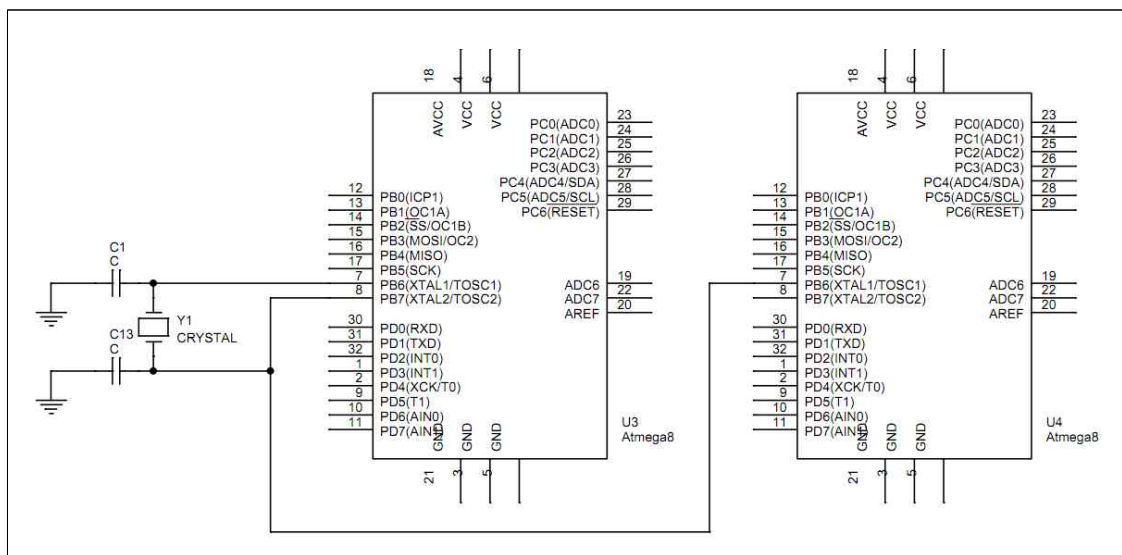
CRYSTAL은 정말 들어본 적 없는 애들이 많을 거라고 생각해. 들어본 적이 없다는 말은 무슨 기능을 하는지도 모르겠지. CRYSTAL은 일정하게 진동하는 크리스탈 판이야. 막상 이게 어디에 쓰이는지 모르겠지? 코드를 짜다 보면 정말 정확한 시간이 필요할 때가 있을 거야. 1초를 센다든가, 특정 주파수마다 뭔가 해야 한다든가 이럴 때 사용하는데 이와 비슷한 거에는 오실레이터라는게 있어. 둘이 모양은 비슷한데 차이점은 확연해.

	X-TAL	오실레이터
그림 4-3, 4-4		
주변 회로	별도의 드라이버가 있고 외부에 크리스탈과 캐패시터를 달아야함.	내부에 크리스탈, 캐패시터, 게이트 또는 X-TAL 드라이버가 이미 들어있어서 주변회로가 간단함.
가격	오실레이터에 비해 가격이 싸지만 고주파일수록 비쌘	크리스탈에 비해 비쌘

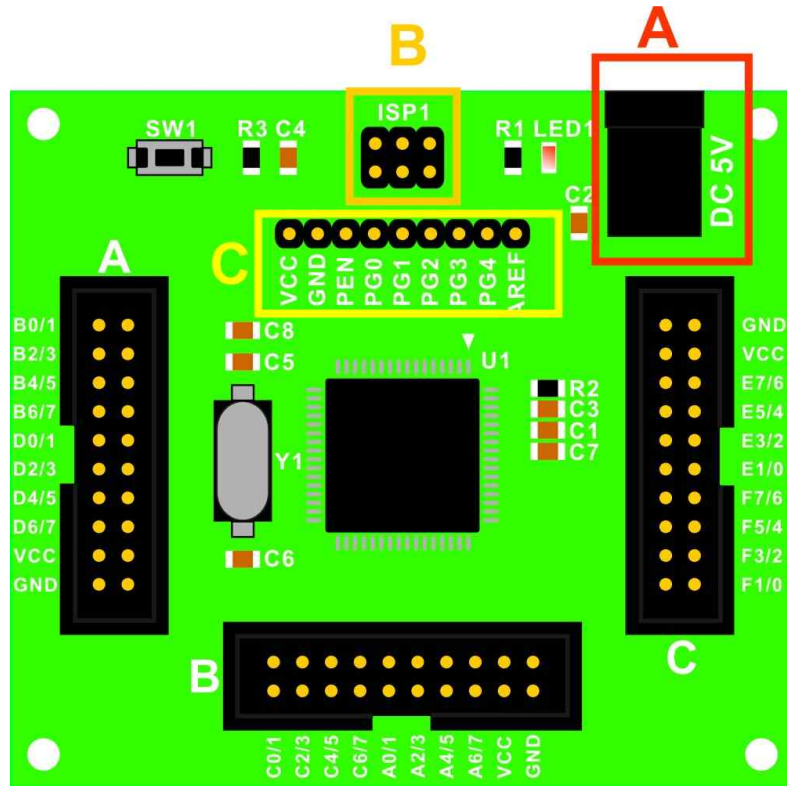
X-TAL은 알다시피 MCU에 XTAL1,XTAL2에 캐패시터와 함께 달면 되고 오실레이터는 어떻게 다는지 궁금하지 않아? MCU에 XTAL1,XTAL2이 각각 INPUT, OUTPUT인 건 알고 있어? 그래서 오실레이터의 출력을 XTAL1으로 입력하면 돼.



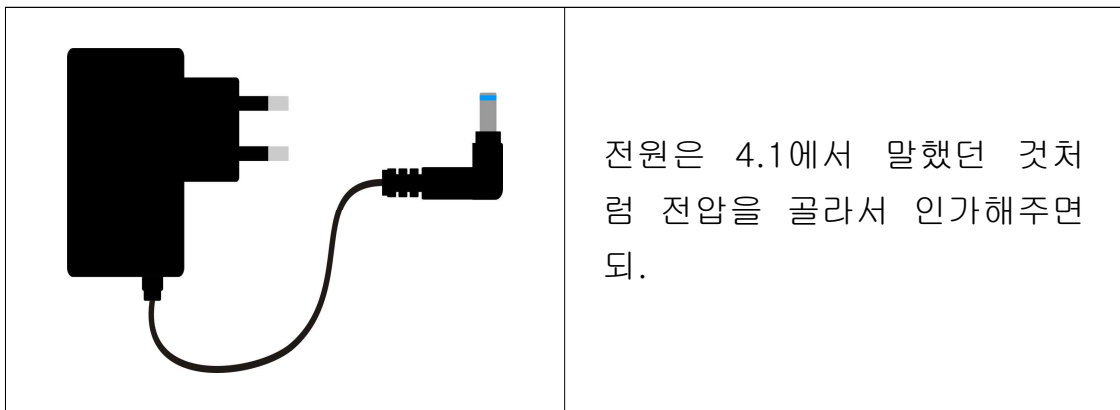
또 이런 건 생각해본 적 있어? 두 개의 MCU 보드를 같은 X-TAL 혹은 오실레이터를 사용하고 싶다. 그러면 어떻게 연결하면 좋을까? 간단하게 X-TAL이나 오실레이터가 들어간 쪽의 XTAL2(출력)를 다른 MCU보드의 XTAL1(입력)에 연결하면 돼.



4.2 실제 보드



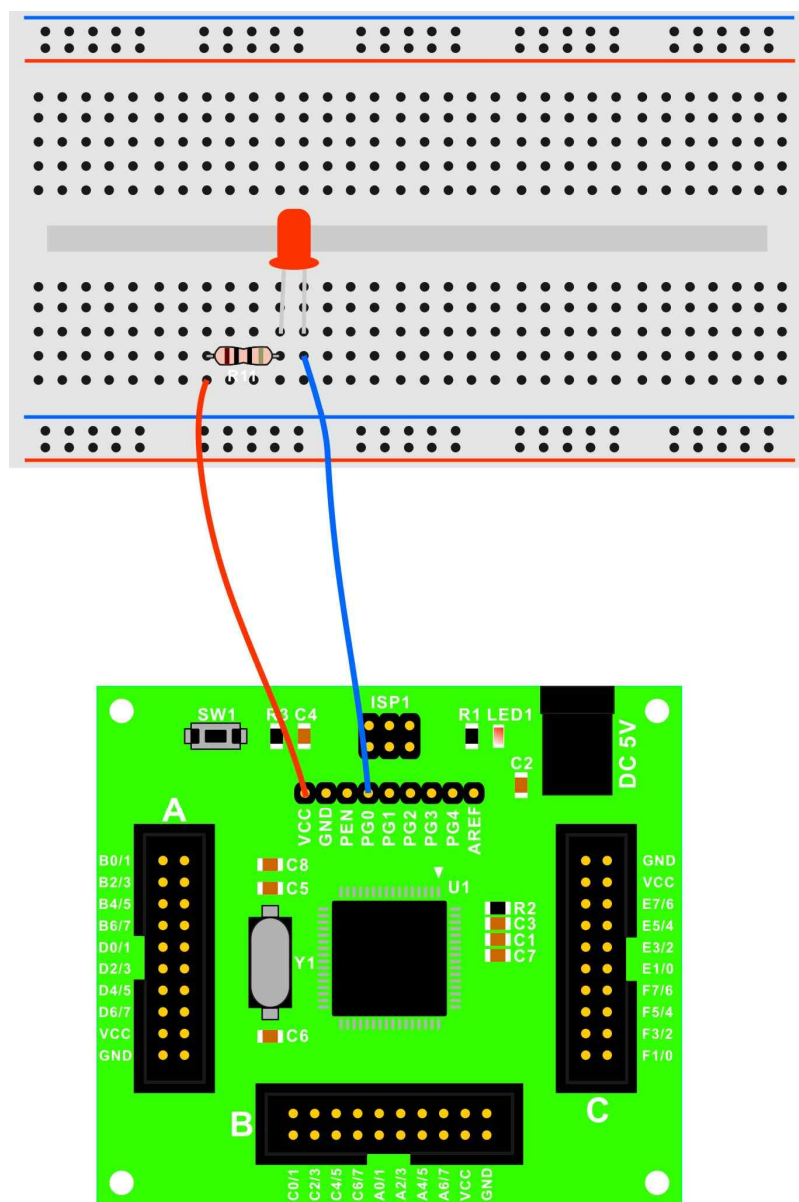
A. 전원



B. ISP

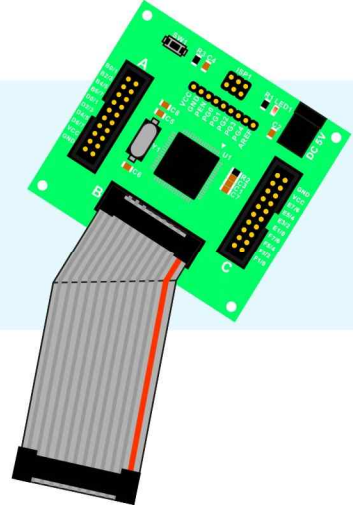
 <p>BL-032S</p> <p>www.BoardFree.kr</p> <p>Components: R3, USB, R2, C2, R1, C3, Y1, D1, USB, OFF, SW, HWB, U1, C1, C4, R4, R14, R5, R6, R7, R8, R13, R12, ACT, TGV, ISP.</p>	<p>ISP에도 비슷한 모양의 커넥터가 있지? 나중에 저 두 부분에 선으로 연결하면 되</p>
---	--

C. 남은 핀



5

보드 투 보드



5.1 보드끼리 아무렇게나 연결하면 되는 건가?

앞에서 설명했듯이 세 개의 박스헤더는 다른 보드와 연결하기 위해 만들어 놓은 거야. 이 책의 순서로는 다음에 배울 내용은 LED_MATRIX잖아? 그럼 그 보드를 조립해서 세 개의 박스헤더에 연결하면 돼. 물론 이제 너가 어떤 기능을 쓸 거냐에 따라 그에 맞는 포트에 연결해야 하지. 예를 들면 아래 그림은 ATmega128의 핀 맵이야. 근데 우리가 통신을 하려고 해. 그럼 TX, RX가 있는 포트를 사용해야겠지? 그럼 RX, TX가 있는 포트가 어딘지 보자. PORTE, PORTD지? 그럼 둘 중에 마음에 드는 PORT로 선택해서 그 PORT가 포함되어 있는 박스헤더를 연결해주면 되는 거야. 간단하지?

