AVR 쉽게 사용해보기 <1> 리얼시스 www.realsys.co.kr 2009.5.26 KCO

AVR 강좌를 시작하면서.....

여러 회사에서 값싸고 다양한 기능을 가진 각종 마이크로프로세서, ARM, DSP 등이 출시되고 있습니다.

옛날에는 Z80, 8051, 8086 등을 많이 사용하였는데, 요즈음에는 이러한 것들은 기능이나 가격 면에서 조금씩 밀리고, AVR, PIC, ARM, DSP 등이 많이 사용되고 있습니다.

AVR은 그 중에서 구입하기 쉽고, C 언어를 사용하여 쉽게 접근하기 좋아서 처음 마이크로프로세서를 사용하는 입문 자 여러분에게 권장할 만합니다.

더욱 빠른 속도를 요구하는 분야에는 DSP 등의 사용이 권장되며, 저 소비 전력, LCD 표시, 복잡한 통신 등과 같은 OS 사용을 요구하는 분야에는 ARM 시리즈를 권장합니다.

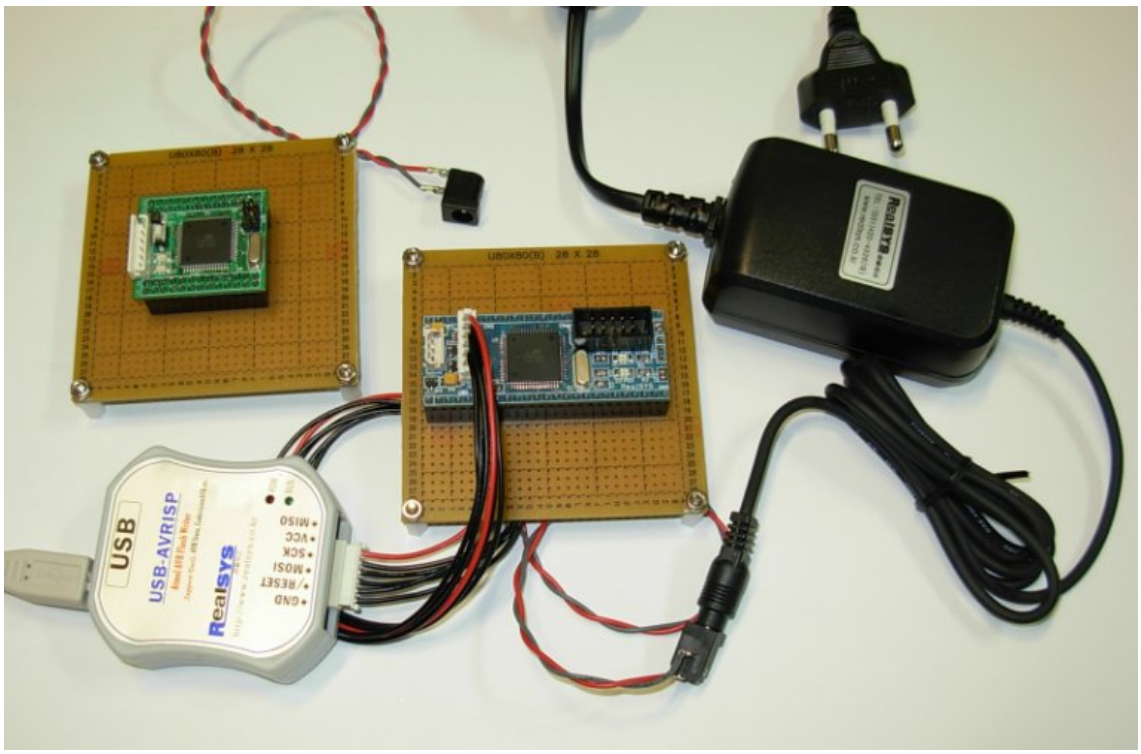
점차 저가격 ARM 시리즈가 많이 출시되어 확대될 거라 생각되지만 그래도 초보자에게 쉽게 간단한 응용을 구현하는 데에는 AVR이 좋을 것 같습니다.

본 강좌에서는 쉽게 사용하는 것을 목표로 합니다.

따라서 단계별로 점진적으로 진행하겠습니다.

준비물:

AVR 모듈(Atmega128 사용), ISP 프로그래머(USB_AVRISP), 전원 장치(5V)



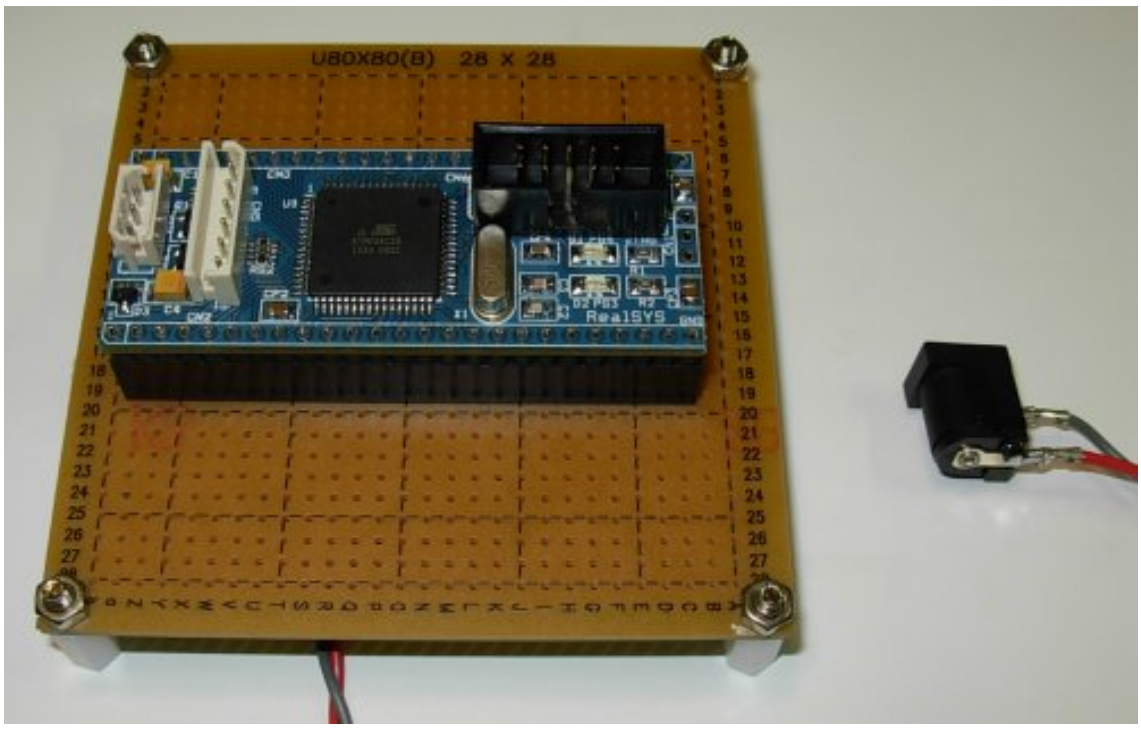
컴파일러는: CodeVisionAVR C Compiler를 사용합니다.

준비물 살펴보기

RCM128 모듈:

- Atmega128 사용 : 16MHz동작 (16MIPS)
- Flash-ROM(128K 바이트), SRAM(4K 바이트), EEPROM(4K 바이트)
- SCI(2개), SPI, TWI(I2C) 통신
- A/D 변환기: 10비트, 8채널
- 타이머 기능, PWM 출력 , I/O 포트, 외부 인터럽트, 외부 버스 확장 가능 등
- 모듈(25핀 1열 커넥터 2개) 기능: RS232 통신, ISP, JTAG 커넥터, 2개의 상태 LED

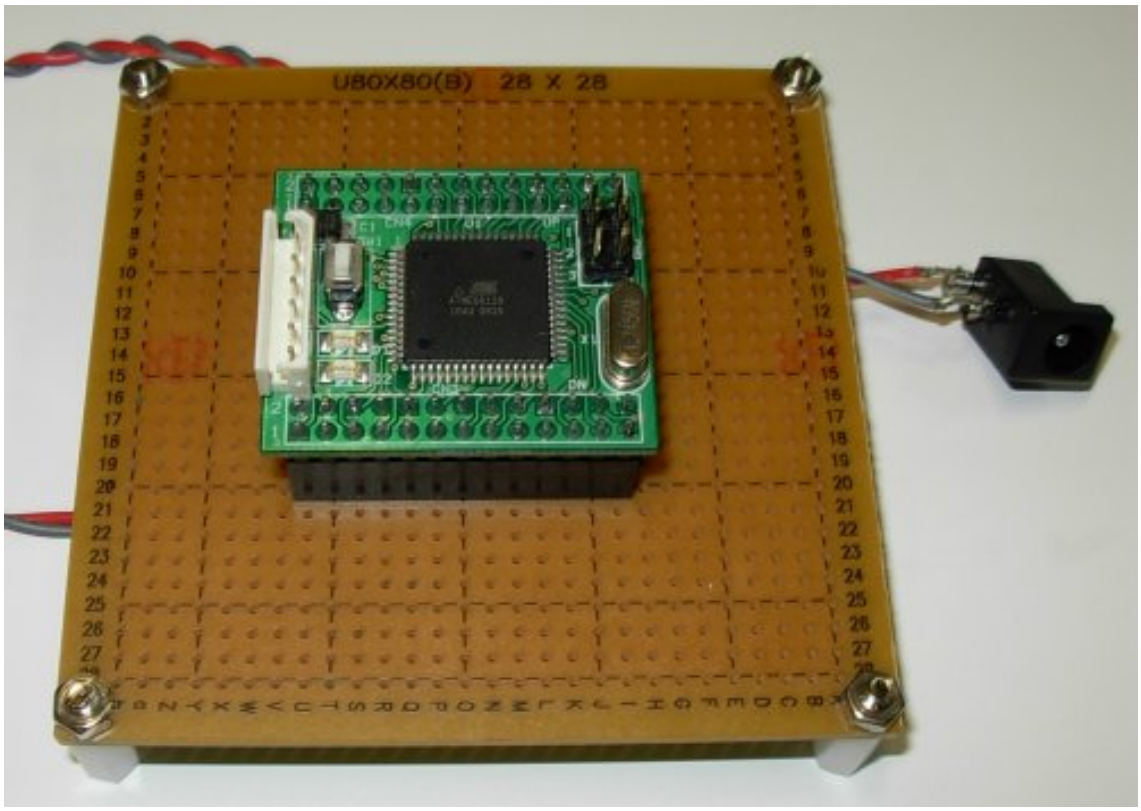
공 PCB에 RCM128 모듈 장착 모습 (공급 전압은 5V 사용)



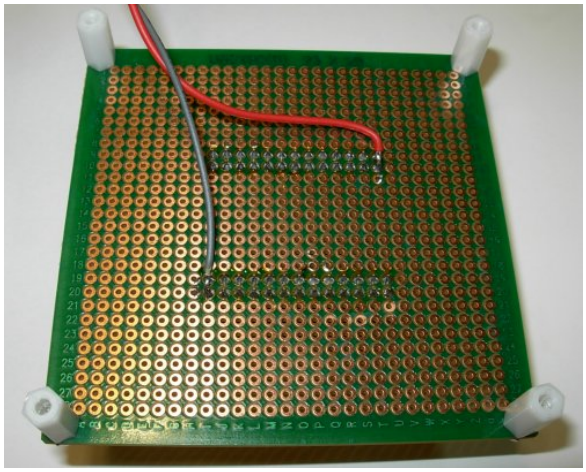
RCM128A 모듈:

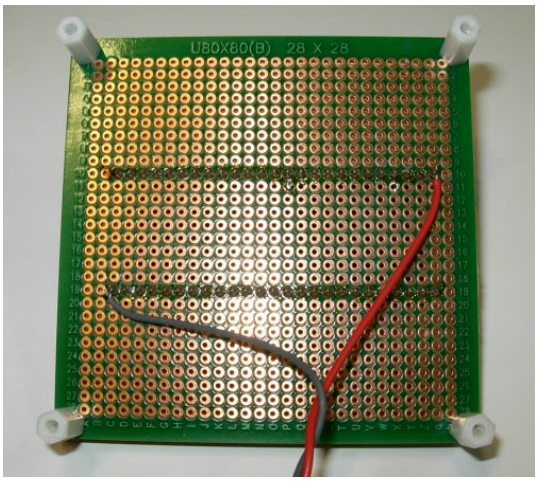
- Atmega128 사용 : 16MHz동작 (16MIPS)
- Flash-ROM(128K 바이트), SRAM(4K 바이트), EEPROM(4K 바이트)
- SCI(2개), SPI, TWI(I2C) 통신
- A/D 변환기: 10비트, 8채널
- 타이머 기능, PWM 출력 , I/O 포트, 외부 인터럽트, 외부 버스 확장 가능 등
- 모듈(26핀 커넥터 2개) 기능: 리세트 버튼, ISP, JTAG 커넥터, 2개의 상태 LED

공 PCB에 RCM128A 모듈 장착 모습(공급 전압은 5V 사용)



뒷면 모습: 처음이라 5V 전원 전원 케이블만 납땜 되어 있음





플래시 롬 프로그램머는 USB-AVRISP 사용

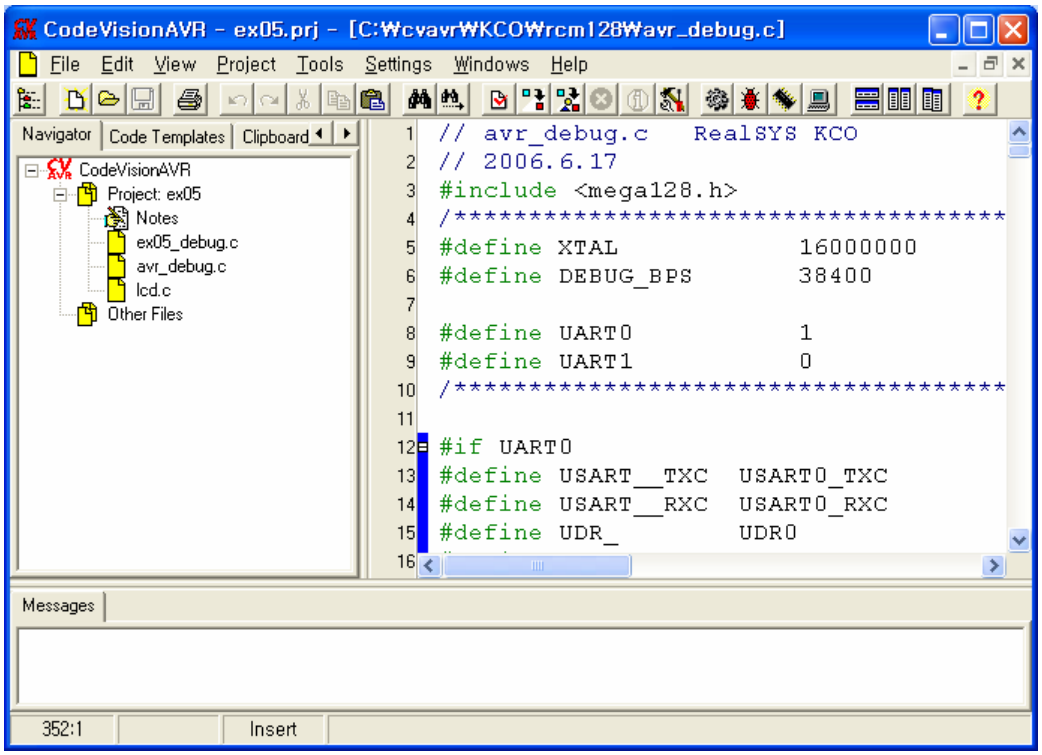
요즈음 컴퓨터에는 프린터 포트가 없는 경우가 많으므로 USB 통신 포트로 접속

플래시 프로그래머는 왜 필요한가?

>>> C 언어로 작성한 실행 프로그램을 전원이 OFF 되어도 지원지지 않는 Flash-ROM에

써 넣어야 전원이 ON 되었을 때 제대로 프로그램을 실행할 수가 있으며, 이 기능을 하는 장치가 플래시 프로그래머입니다.

컴파일러는 CodeVisionAVR C 컴파일러 사용



잠깐 살펴보기:

C 컴파일러는 무엇이고 왜 필요한가???

>>> C 컴파일러는 우리가 C 언어로 작성한 프로그램을 마이크로프로세서가 실행할 수 있는 기계 언어로 변환해주는 프로그램입니다. 마이크로프로세서마다 내부 구조가 각각 다르므로, 여러 가지 다른 컴파일러가 있으며 사용자는 사용하는 컴파일러에 익숙하도록 연습이 필요합니다.

여기에서 사용하는 CodeVisionAVR은 AVR 시리즈만 지원합니다.

C 언어는 꼭 배워야 하는가???

>>> 그렇습니다. 미국에 가서 자유롭게 좀 살려면 영어를 배워야 하고, 한국에 살려면 한국어를 배워야 하듯이 여러 가지 MCU, DSP 등을 사용하려면 C 언어를 배워야 합니다.

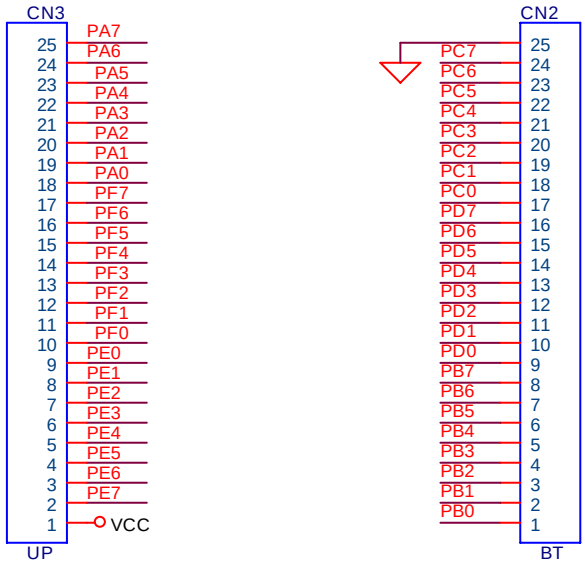
난 베이직이 좋은데 베이직으로 사용하면 안될까요? 조금은 되지만 불편함이 많습니다.

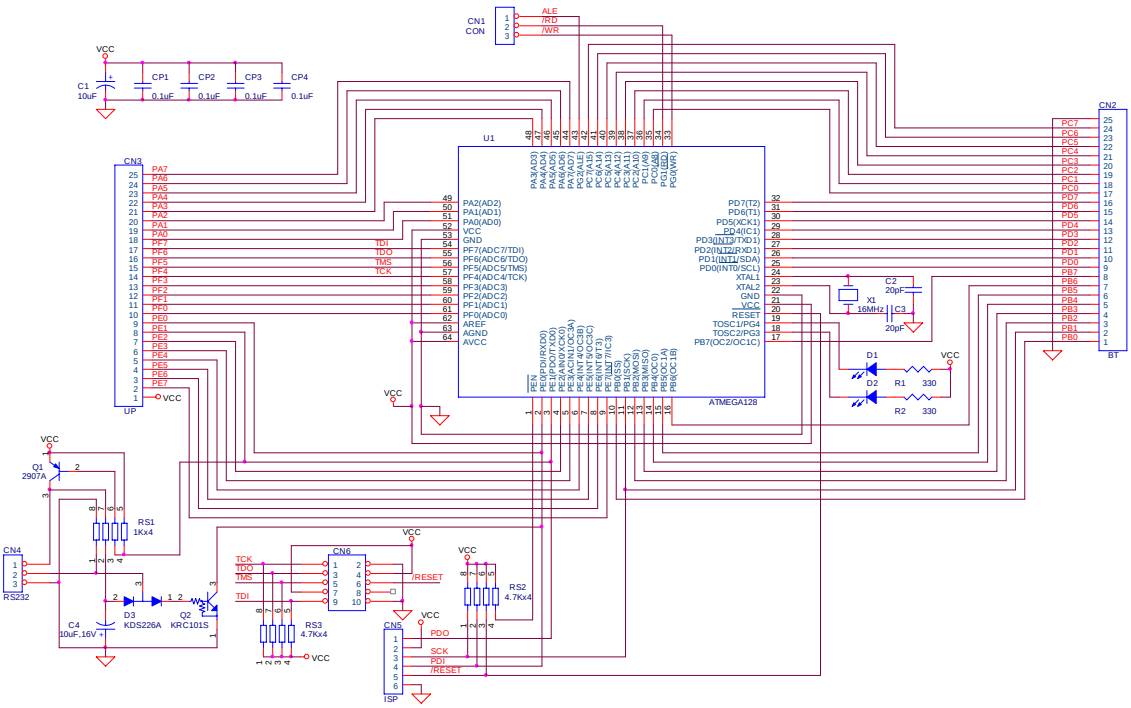
꼭 CodeVisionAVR 컴파일러를 사용해야 되는가???

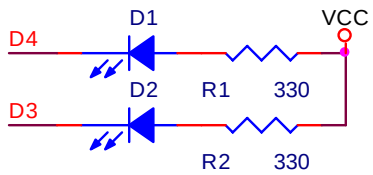
>>> 아닙니다. IAR, ICC, WINAVR를 사용해도 됩니다. CodeVisionAVR를 권장하는 이유는 사용이 조금 편리하고 우리가 대리점(판매업체)이어서 그렇습니다. ㅎㅎ

RCM128 모듈 회로 살펴보기

위쪽 커넥터 아래 쪽 커넥터



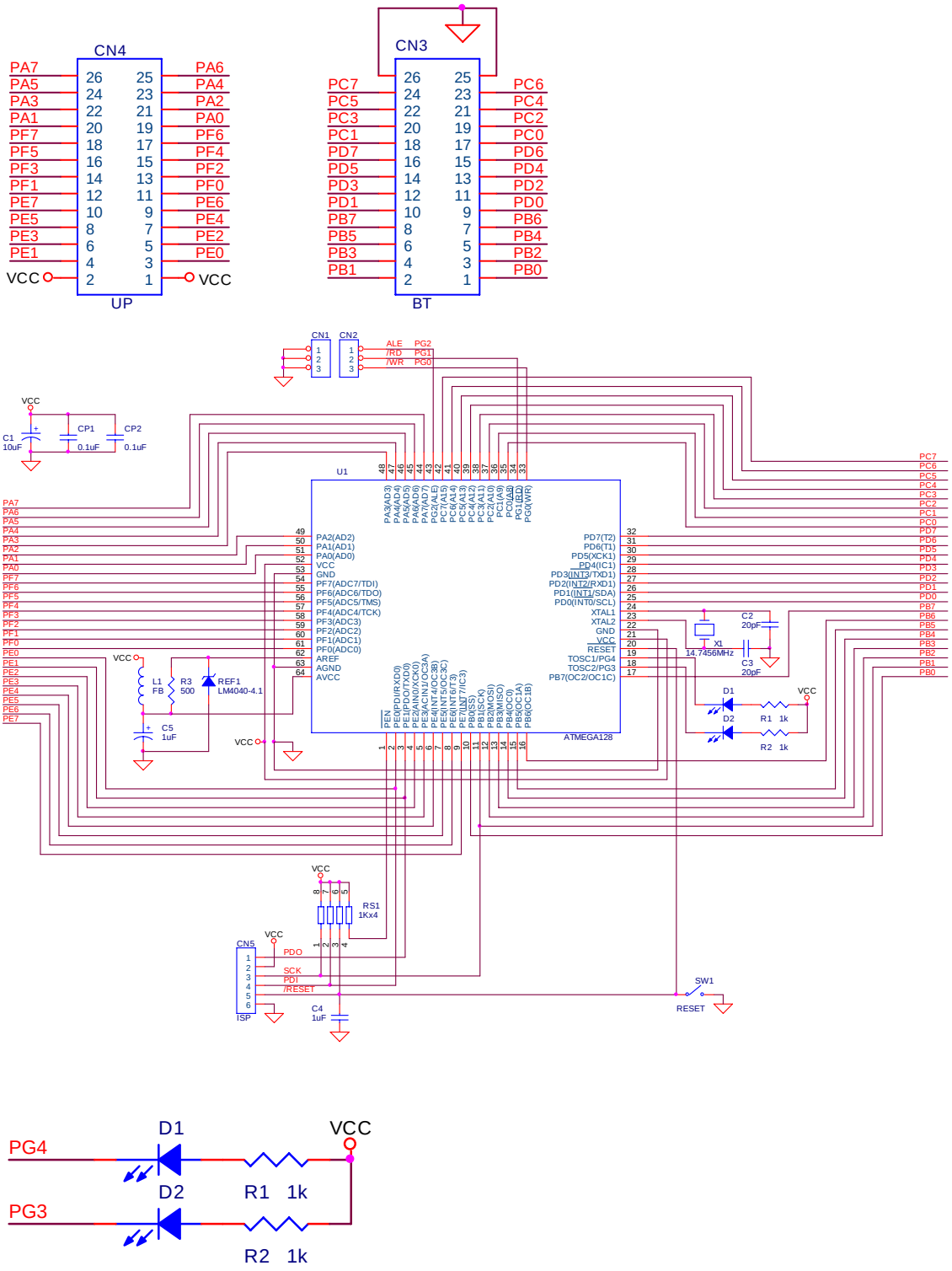




LED D1은 PG4 포트에, LED D2는 PG3 포트에 접속, 포트 값이 0일 때 LED가 켜 짐.

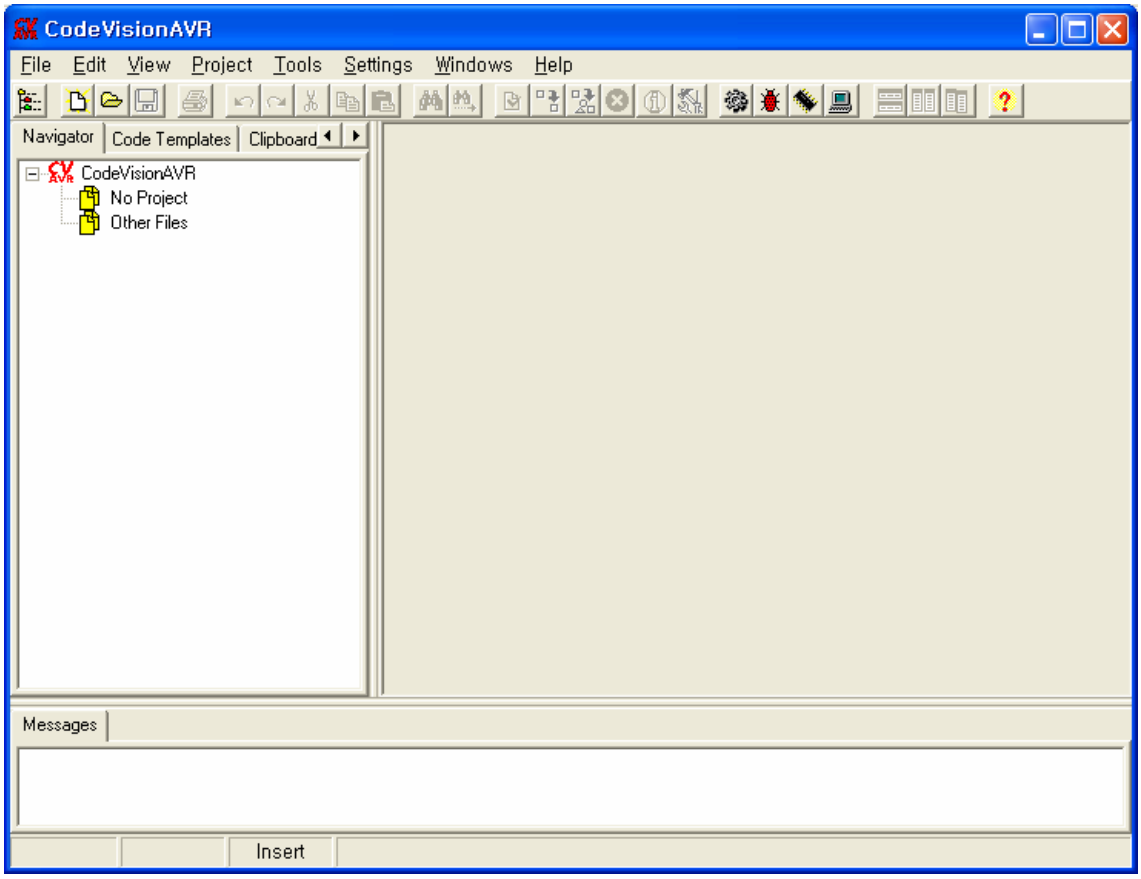
RCM128A 모듈 회로 살펴보기

위쪽 커넥터 아래 쪽 커넥터

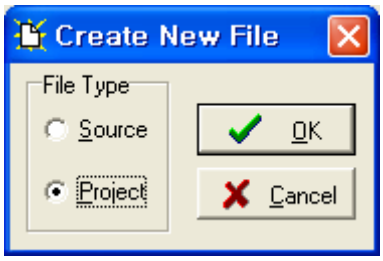


LED D1은 PG4 포트에, LED D2는 PG3 포트에 접속, 포트 값이 0 일 때 LED가 켜 짐.

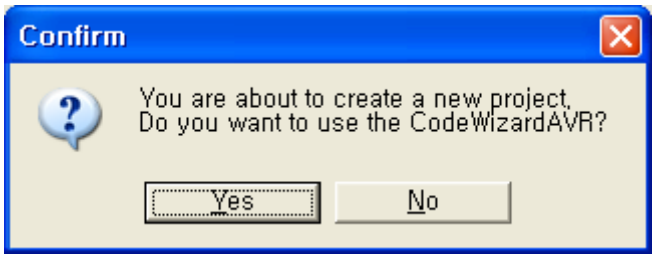
난 ~~~ 아직 C도 잘 모르고, 컴파일러도 잘 모르는데, 일단 한번 LED를 구동하고 싶다.
어떻게 해야 하나???
>>> 가능합니다. 일단 한번 해봅시다.



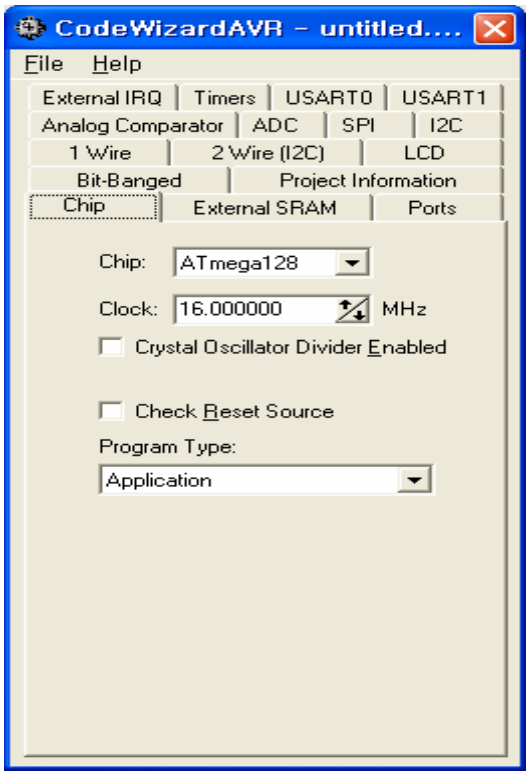
<1> 새로운 프로젝트를 만듭니다.



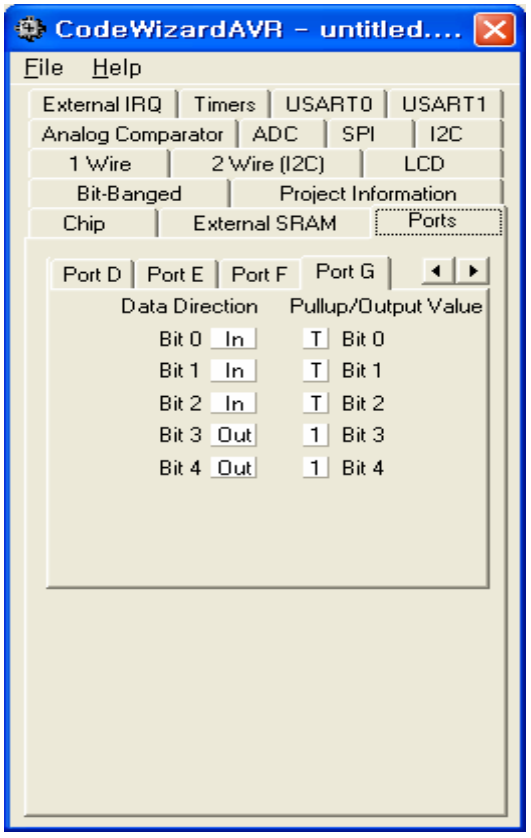
<2> CodeWizardAVR 기능을 사용합니다. - 코드 자동 생성 기능



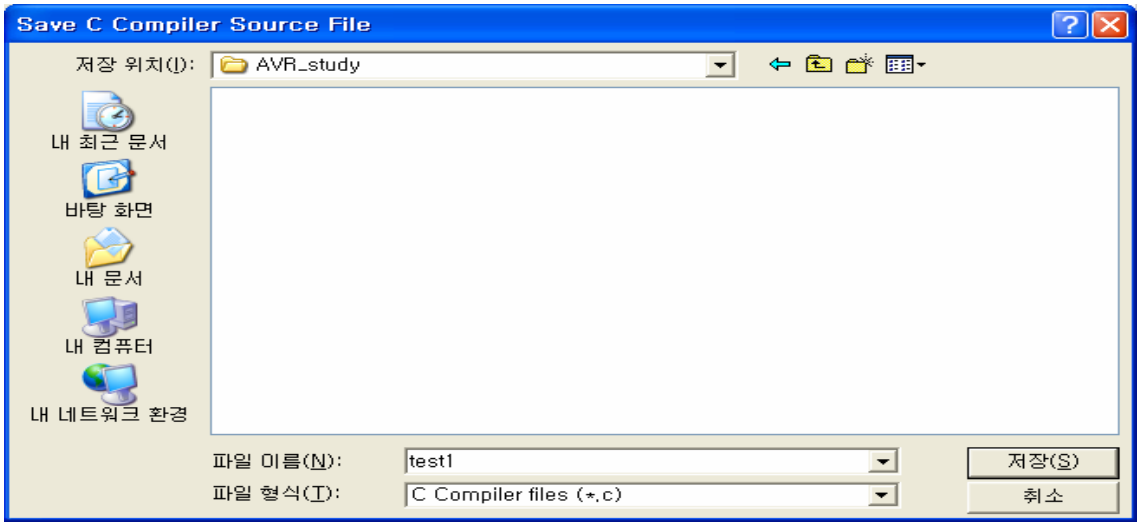
<3> Chip 종류, Clock 값 등을 설정합니다.

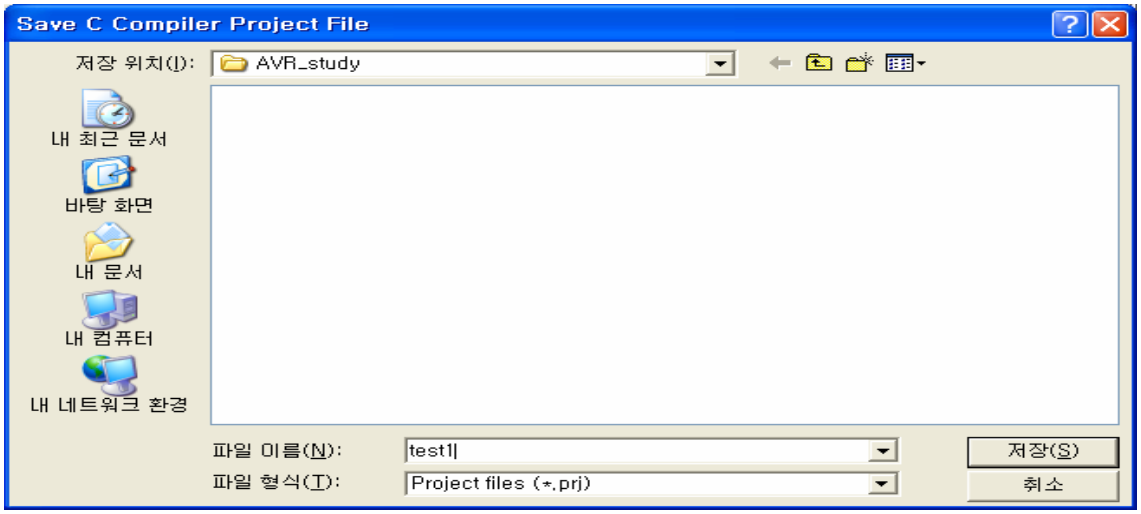


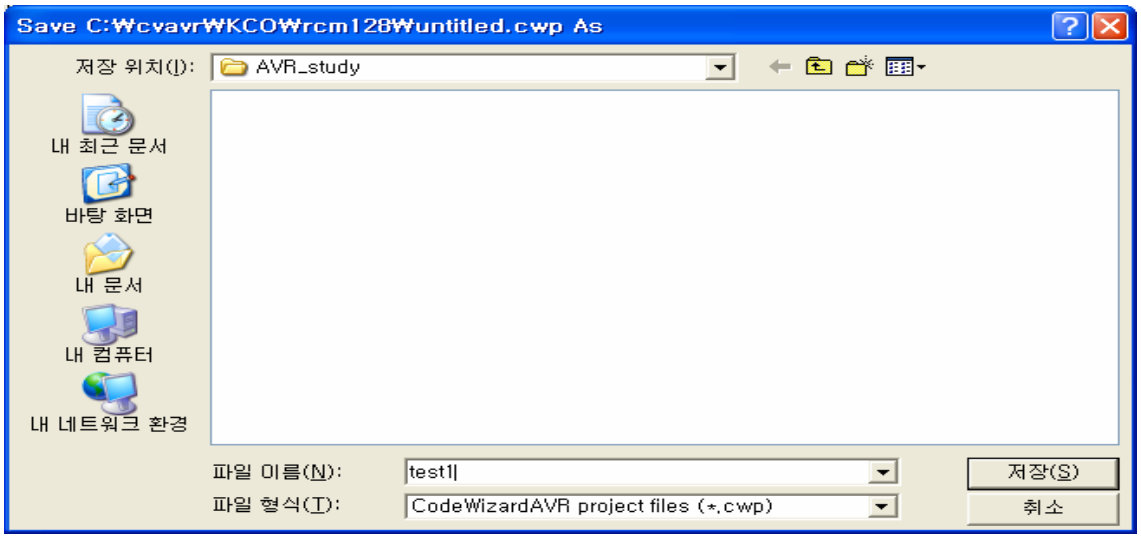
<4> 모듈에 있는 LED를 동작 시키기 위해 포트 G의 Bit3과 Bit4를 출력으로 설정합니다.



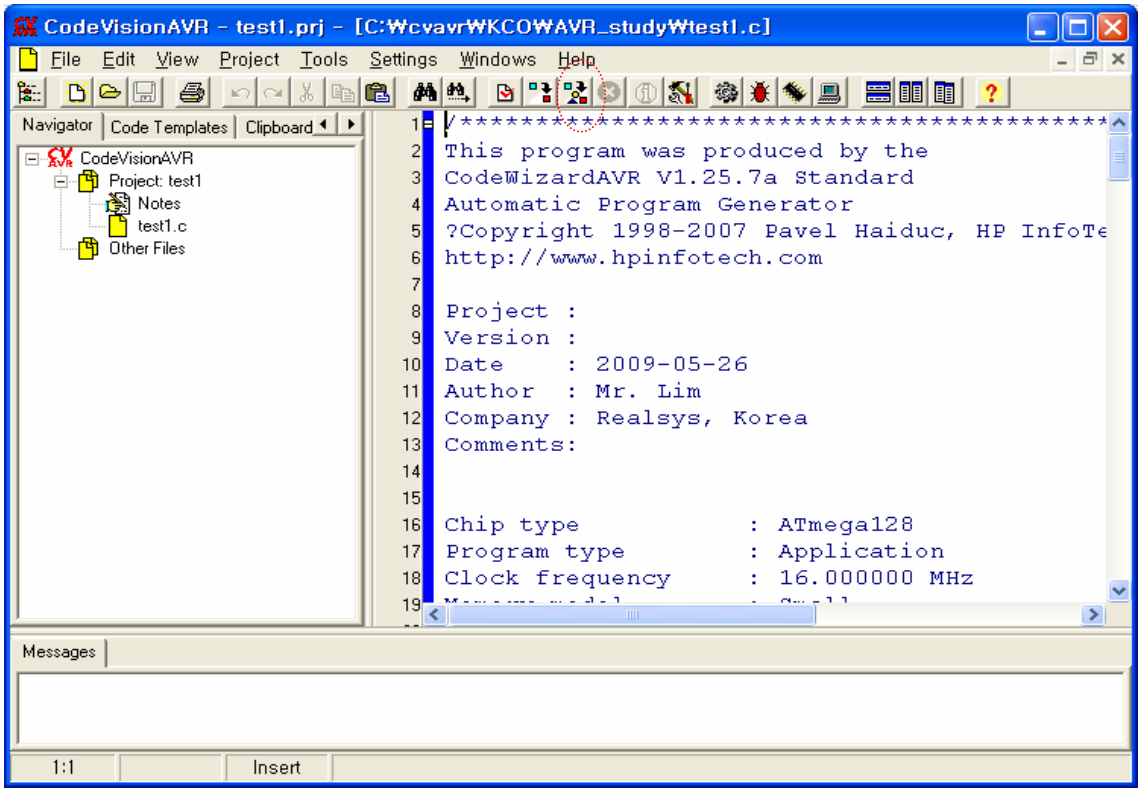
<5> 계속 test1 만 타이프 해봅니다.



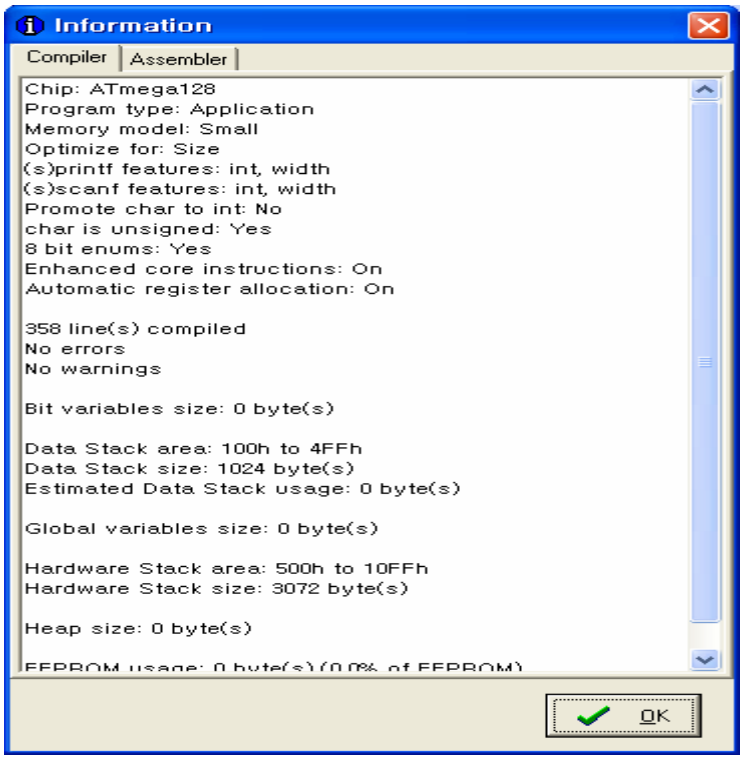




<6> 드디어 소스 파일이 생성되었습니다.

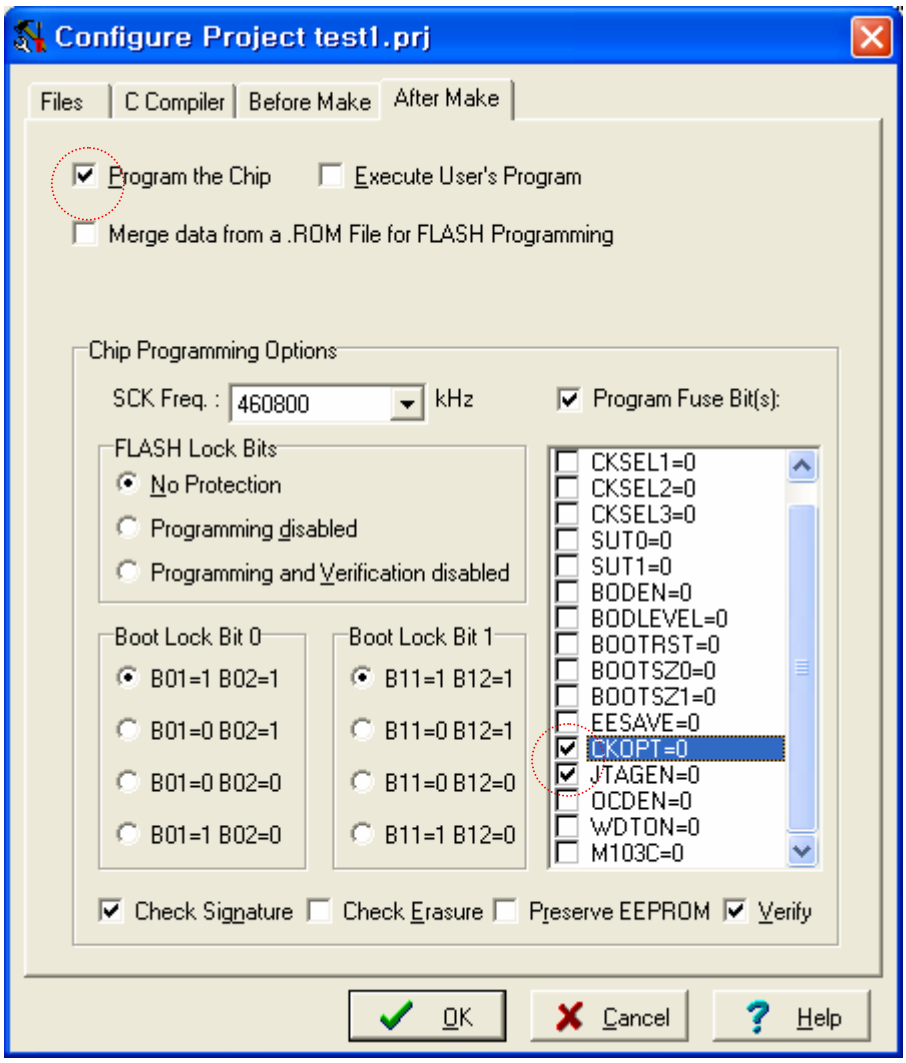


<7> Project > Make를 실행해봅니다.

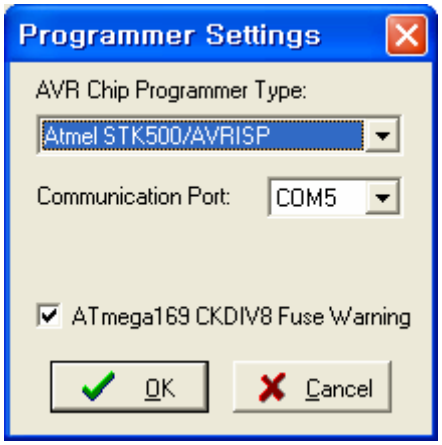


어 ~~ 무엇이 OK 야??? 무언가 동작을 기대했는데…

<8> Project > Configure 를 실행하고, After Make를 선택하여 아래와 같이 체크해줍니다.

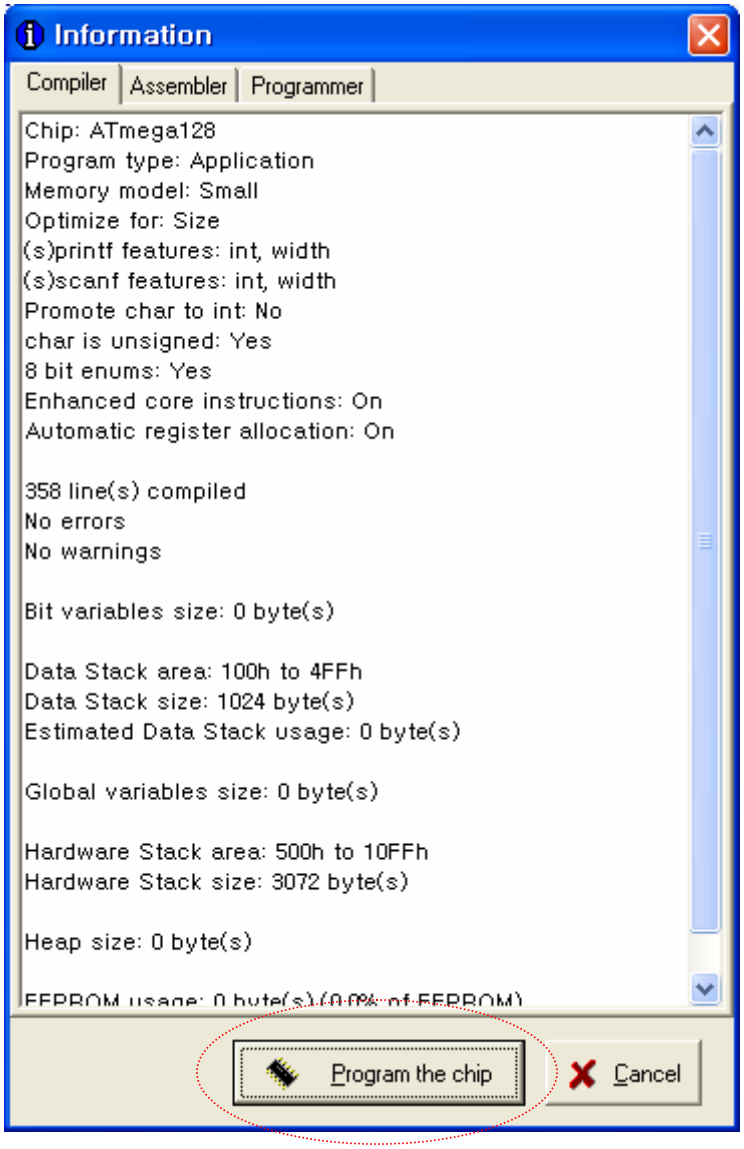


<9> Settings > Programmer를 선택하여 Type 및 COM 포트를 적절히 설정합니다.



<10> Settings > Programmer를 선택하여 Type 및 COM 포트를 적절히 설정합니다.

<10> 다시 Project > Make를 실행해봅니다.



이번에는 Program the chip 버튼이 생성되었습니다.

버튼을 눌러봅니다.

- 무언가가 실행된 것 같습니다. 하지만 LED는 여전히 OFF입니다.

<11> 소스 파일을 변경해봅니다.

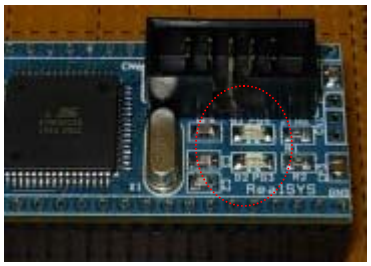
```
#include <mega128.h>
#include <delay.h>
...
while (1)
      {
      // Place your code here
```

```
        PORTG ^= 0x18;
        delay_ms(100);
    };
}
```

<12> 다시 Project > Make를 실행해봅니다.

>>> 2개의 LED가 번쩍 번쩍합니까?

안되면 될 때까지 하세요……



오늘의 강좌는 여기까지……

(2009.5.26. 10:58PM)