

## 실습 제목

Microchip Studio for AVR의 소개 및 개발 도구 사용법

## 실습목적

AVR 계열 마이크로컨트롤러인 ATmega328PB에 대해 간단히 살펴보고, ATmega328PB를 이용한 장치 개발에 이용되는 개발 도구의 기본적인 사용법을 익히도록 한다.

## 실습 준비물

실습용 보드 (ATmega328PB Xplained Mini)

ATmega328PB datasheet

Microchip Studio for AVR

## 실습에 필요한 예비지식

### 1. ATmega328PB Xplained Mini

#### 1) ATmega328PB Microcontroller 특징

ATmega328PB 마이크로컨트롤러는 단일 칩으로 구성된 MCU이며 주요 특징은 아래와 같다.

##### Memories

32 kBytes of In-System Self-Programmable Flash Program Memory

1 kBytes EEPROM

2 kBytes Internal SRAM

##### Peripherals

Two 8-bit Timer/Counters

Three 16-bit Timer/Counters

Eight-channel 10-bit ADC

Two USARTs

Two SPIs

Two TWIs

Operating Voltage: 1.8 - 5.5 V

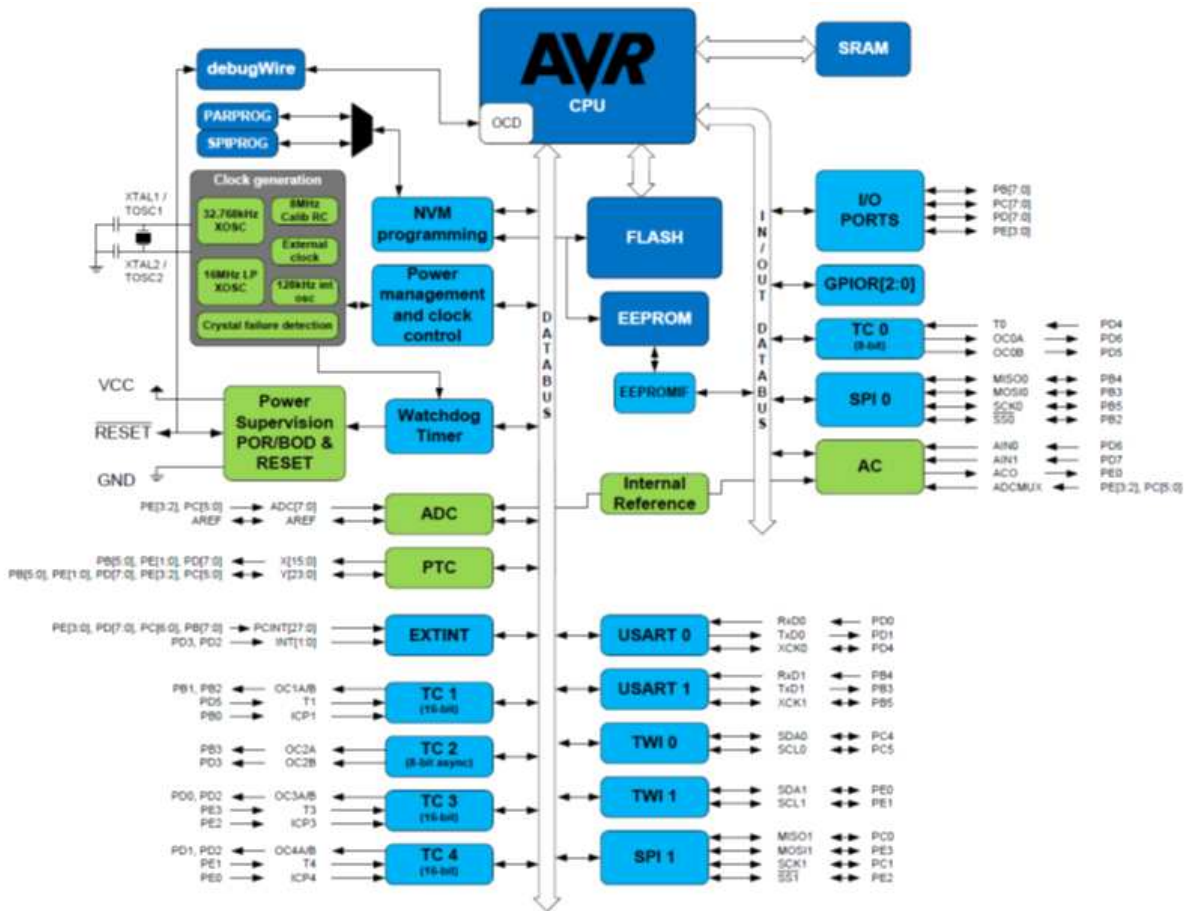
Speed Grade:

0 - 4 MHz @ 1.8 - 5.5 V

0 - 10 MHz @ 2.7 - 5.5 V

0 - 20 MHz @ 4.5 - 5.5 V

2) ATmega328PB MCU 구성



3) 메모리

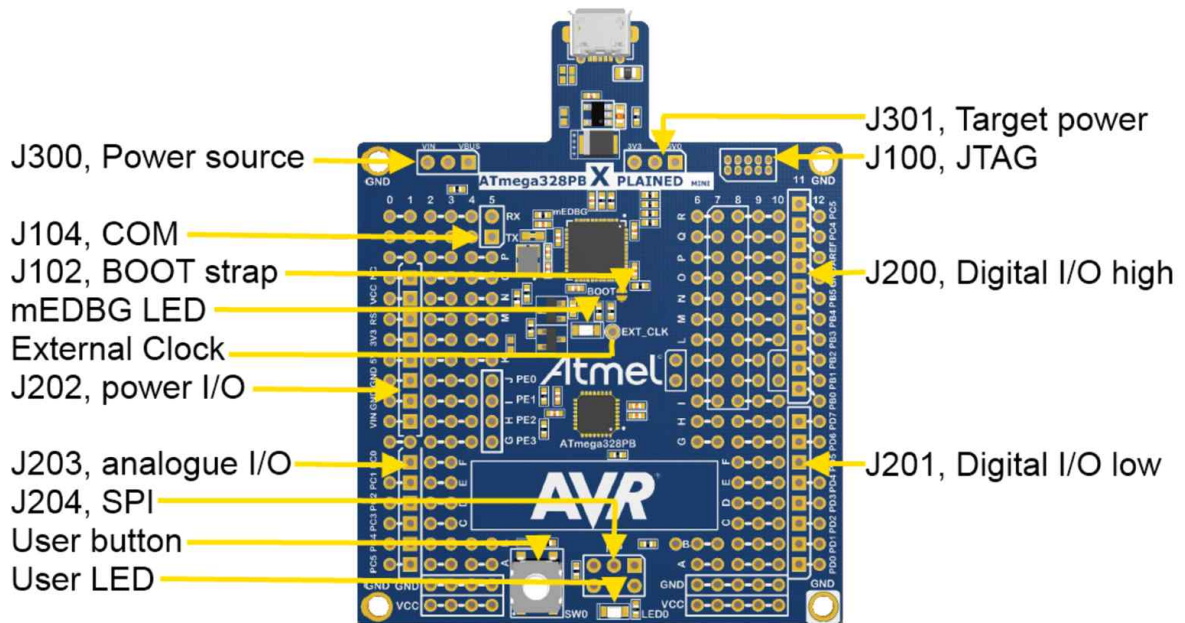
① 데이터 메모리(data memory)

데이터 저장 메모리로 읽고 쓰기가 가능하며, 용량은 2 kByte.

② 프로그램 메모리(code memory, program memory)

명령 실행코드(program code)가 저장되는 메모리로서 32 kByte의 Flash 메모리로 구성.

4) ATmega328PB Development Kit (ATmega328PB Xplained Mini)



## 2. Microchip Studio for AVR 사용법

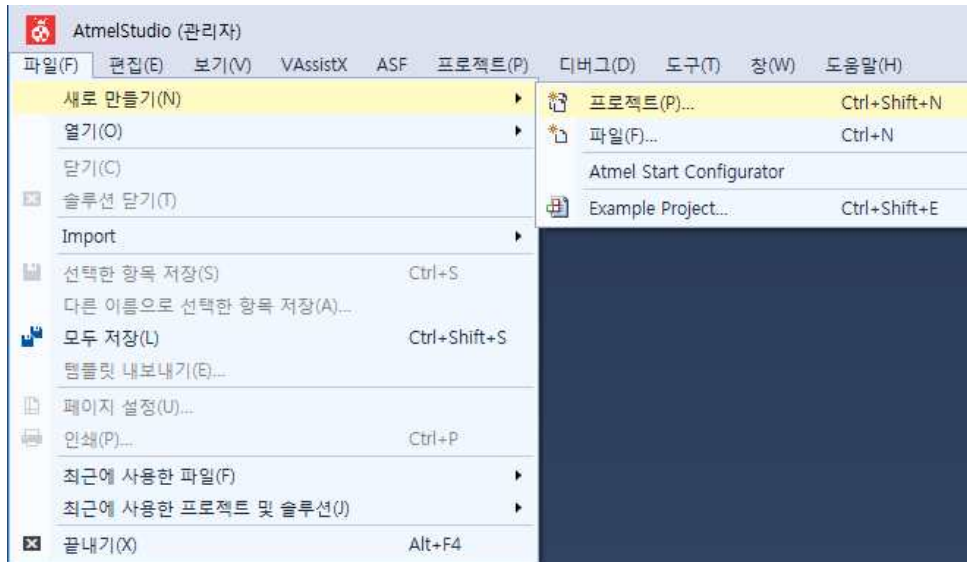
Microchip사의 Microchip Studio for AVR은 프로젝트 파일 관리, 소스 코드 편집, 프로그램 디버깅, 시뮬레이션 등을 할 수 있는 통합 개발 환경이다. 또한 프로그램을 보다 쉽고 빠르게 만들 수 있으며, 하나의 어플리케이션으로 에디터와 디버거를 모두 사용할 수 있는 장점을 가지고 있다.

## 실습 내용

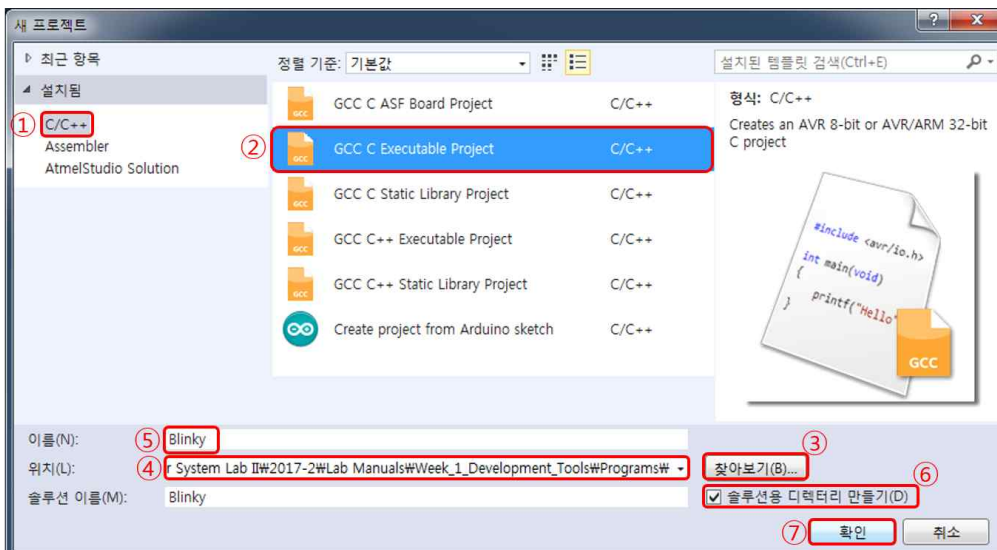
1. Microchip Studio for AVR과 C 언어를 사용하여 ATmega328PB Xplained Mini Board에 설치되어 있는 LED를 점멸시키는 프로그램을 만들고, 디버깅을 해 본다.

### 1) 프로젝트(project) 생성

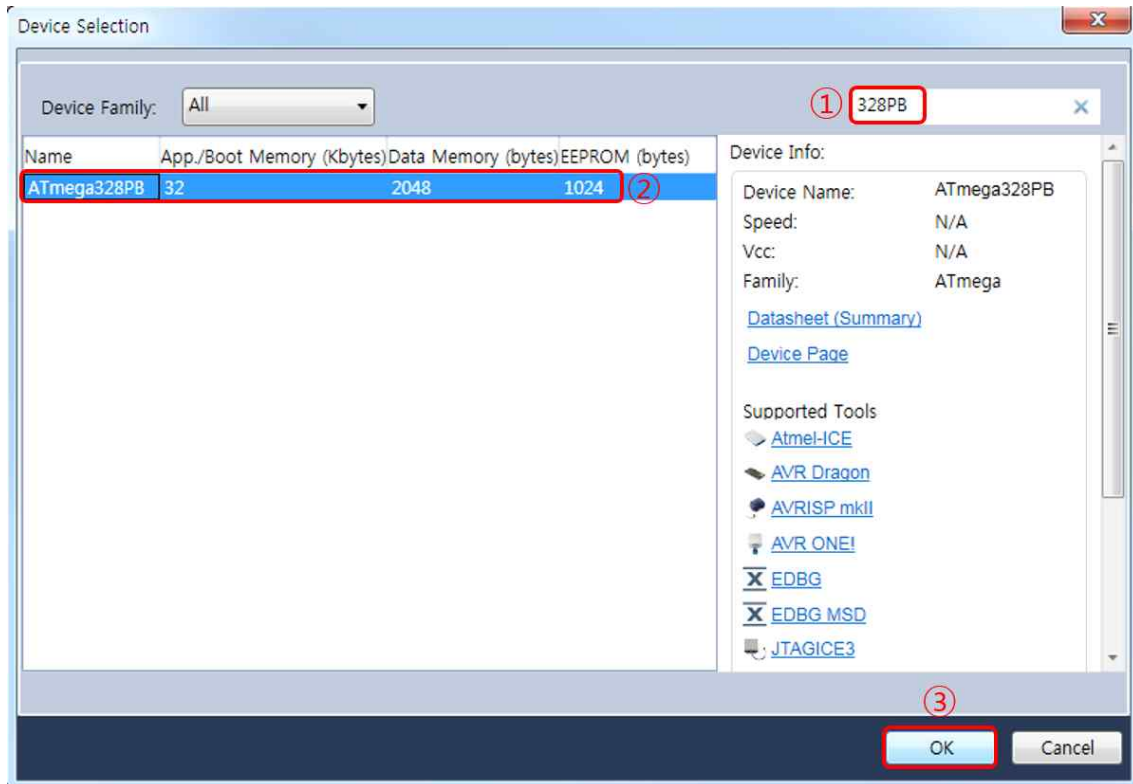
(1) 파일(F) - 새로만들기(N) - 프로젝트(P) 항목을 선택한다.



(2) 왼쪽 메뉴 항목에서 ①C/C++ 항목을 선택하고, 가운데 창에서 ②GCC C Executable Project (C/C++)를 선택한 후, ③찾아보기 버튼을 클릭하여 ④프로젝트 파일이 저장될 폴더의 위치를 지정한 다음, ⑤프로젝트 이름(Blinky)을 입력하고, ⑥'솔루션용 디렉터리 만들기' 항목을 선택한 다음 ⑦확인 버튼을 클릭한다.



- (3) 'Device Selection' 창에 ①'328PB'라고 입력하면 해당 이름을 가진 소자만 선별되어 출력된다. 이 중에서 ②'ATmega328PB' 소자를 클릭하여 선택한 다음, ③OK 버튼을 클릭한다.



(4) Microchip Studio for AVR이 자동으로 프로젝트의 골격을 생성하고, 아래 그림과 같이 아주 간단한 main.c 파일을 생성해 준다.

```

1  /*
2  * Blinky.c
3  *
4  * Created: 2017-08-25 오후 5:44:02
5  * Author : Administrator
6  */
7
8  #include <avr/io.h>
9
10
11 int main(void)
12 {
13     /* Replace with your application code */
14     while (1)
15     {
16     }
17 }
18
19

```

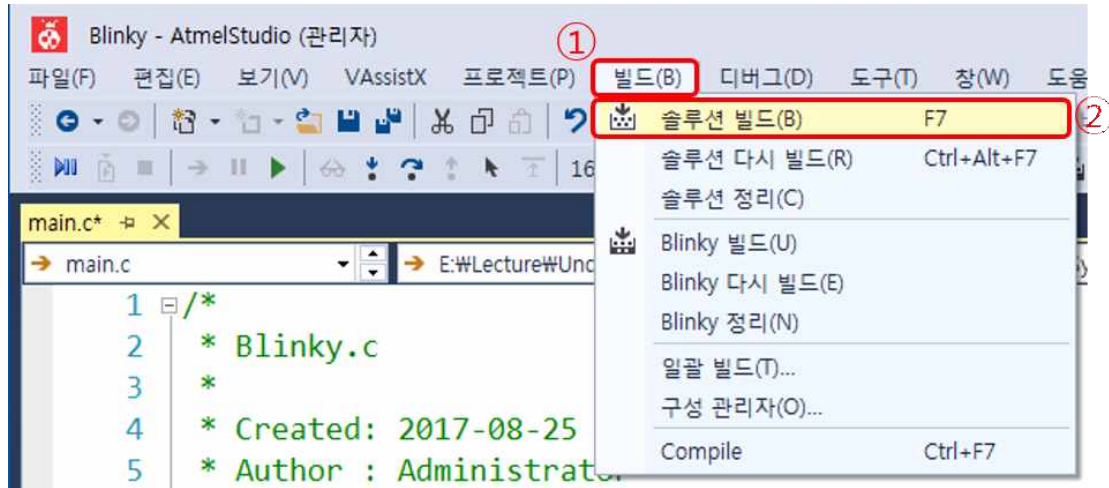
(5) 이 main.c 파일을 다음과 같이 수정한다.

```

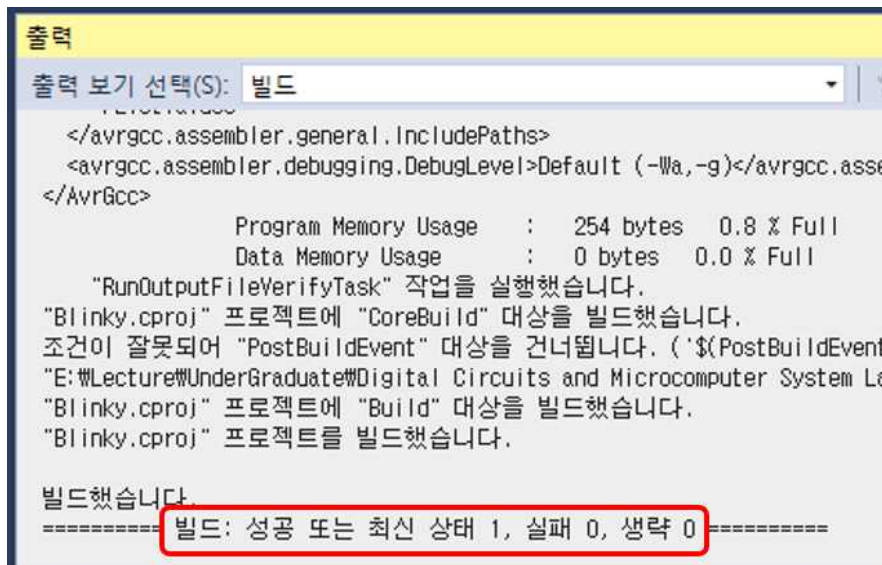
1  /*
2  * Blinky.c
3  *
4  * Created: 2017-08-25 오후 5:44:02
5  * Author : Administrator
6  */
7
8  #define F_CPU 16000000
9
10 #include <avr/io.h>
11 #include <util/delay.h>
12
13 int main(void)
14 {
15     DDRB = 0b00100000;
16     while (1)
17     {
18         PORTB = 0b00100000;
19         _delay_ms(100);
20         PORTB = 0b00000000;
21         _delay_ms(900);
22     }
23 }

```

(6) Build - Build Solution 메뉴를 선택하여 위 프로그램을 Compile 및 Link 한다.

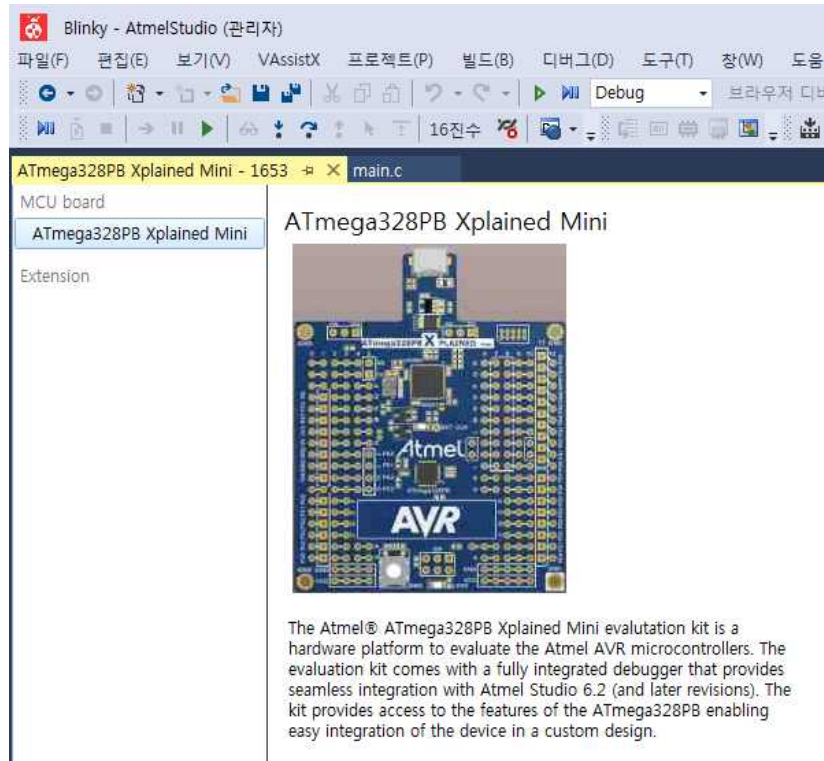


(7) 프로그램에 문제가 없고 정상적으로 빌드가 되면 출력창에 아래의 그림과 같은 프로젝트가 정상적으로 빌드 되었다는 메시지가 출력된다.





- (8) USB Cable을 사용하여 ATmega328PB Xplained Mini Board를 PC에 연결하면, 잠시 후에 인식하였다는 화면이 나타난다.



- (9) 잠시 더 기다리면 이 Board와 PC가 직렬 통신을 하는데 필요한 Driver가 설치되고 다음과 같이 Virtual COM가 설치되었다는 창이 나타난다. 닫기 버튼을 클릭하여 창을 닫는다.



- (10) 빌드된 프로그램을 ATmega328PB Xplained Mini Board에 있는 ATmega328PB MCU 내부의 Flash Memory에 써 넣기 위해서 이 프로젝트에 대한 속성의 일부를 변경해 주어야 한다. 프로젝트 메뉴 - Blinky 속성을 선택한다.



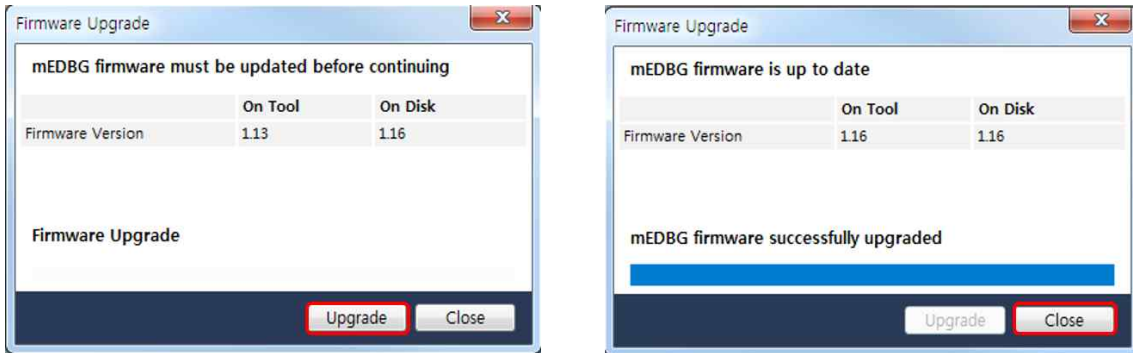
- (11) 아래 그림과 같이 프로젝트 속성창에서 ①Tool 항목을 선택한 후, ②Selected\_debugger/programmer 항목에서 현재 PC에 연결된 ATmega328PB Xplained Mini Board에 해당하는 mEDBG ATML\*\*\*\* 항목을 선택하고, ③Interface 항목에서 debugWIRE 항목을 선택한 후, 다시 빌드를 실행한다.



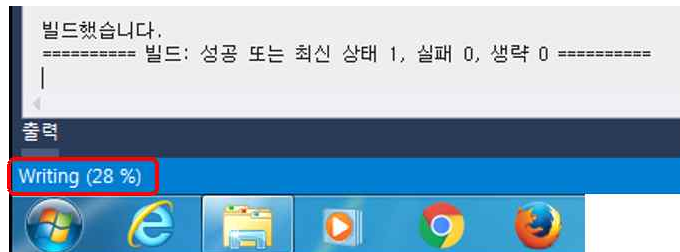
- (12) 빌드된 프로그램을 ATmega328PB Xplained Mini Board에 있는 ATmega328PB MCU 내부의 Flash Memory에 써 넣은 후, 즉시 프로그램을 실행시키기 위해 디버그 메뉴의 디버그하지 않고 시작 항목을 선택한다.



- (13) 사용하는 환경에 따라 ATmega328PB XMINI board의 Firmware를 Upgrade해야 한다는 창이 나타날 수 있으며, 이 경우에는 Upgrade 버튼을 클릭하여 Firmware를 Upgrade를 시작한 후, Upgrade가 종료되면 Close 버튼을 클릭한다. 위 (12) 항목을 한 번 더 실행한다.



- (14) 프로그램이 다운로드 되는 동안에는 창의 아래쪽에 그림과 같이 진행 상황이 숫자로 나타나고, 다운로드가 완료되는 즉시 프로그램이 실행되어 Board 상의 주황색 LED가 점등되는 것을 볼 수 있다.

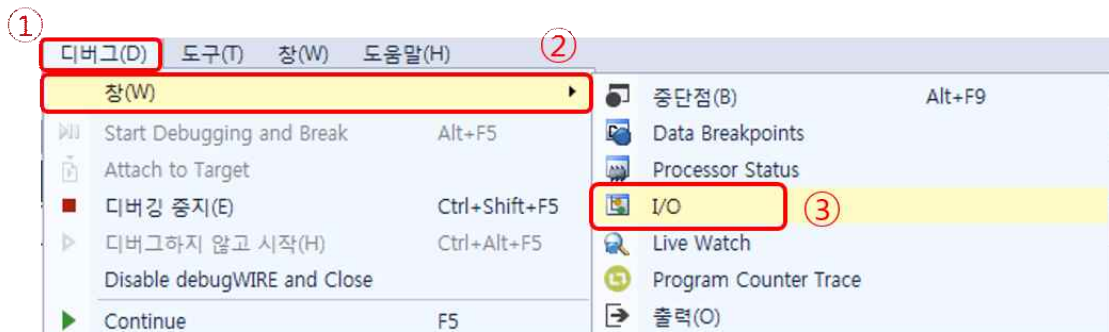


- (15) 프로그램을 디버깅 하기 위해 디버그 메뉴 항목의 Start Debugging and Break 항목을 선택한다.

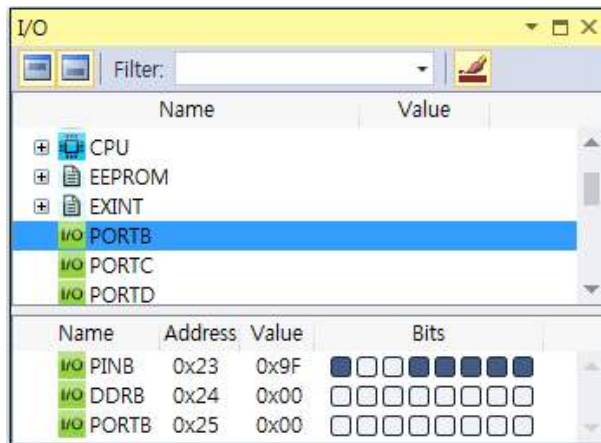




(17) 이 15번 라인에 있는 statement의 기능은 ATmega328PB MCU 내부에 있는 입출력 설정을 위한 레지스터 가운데 하나인 DDRB 레지스터에 2진수 00100000을 써 넣는 것이다. Microchip Studio for AVR의 디버깅 기능을 통해 실제로 DDRB 레지스터에 값이 써 넣어지는 것을 확인해 볼 수 있다. 이를 위해 ①디버그 - ②창 - ③I/O 메뉴 항목을 선택하여 입출력 레지스터의 값을 보여주는 나타내는 창을 연다.



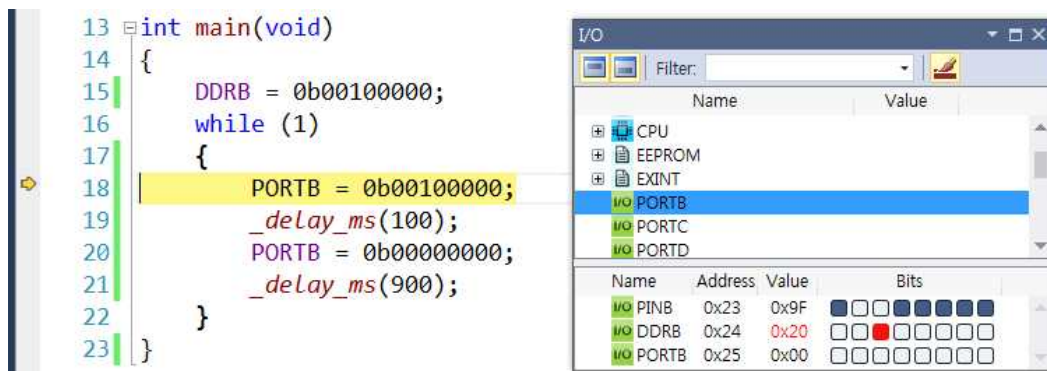
(18) I/O 창에서 PORTB 항목을 클릭하면 아래쪽의 창에 PORTB에 관련된 세 개의 레지스터 값들이 출력된다. 여기서 Value 항목은 각 레지스터에 저장되어 있는 값을 16진수로 나타낸 것이고, Bits 항목은 2진 형태의 그림으로 나타낸 것으로 각 비트에 해당하는 사각형이 비어 있으면 0, 청색이나 적색으로 채워져 있으면 1을 나타낸다.



(19) 15번 라인의 한 statement만 실행하고 멈추기 위해 ①디버그 - ②프로시저 단위 실행 항목을 선택한다.

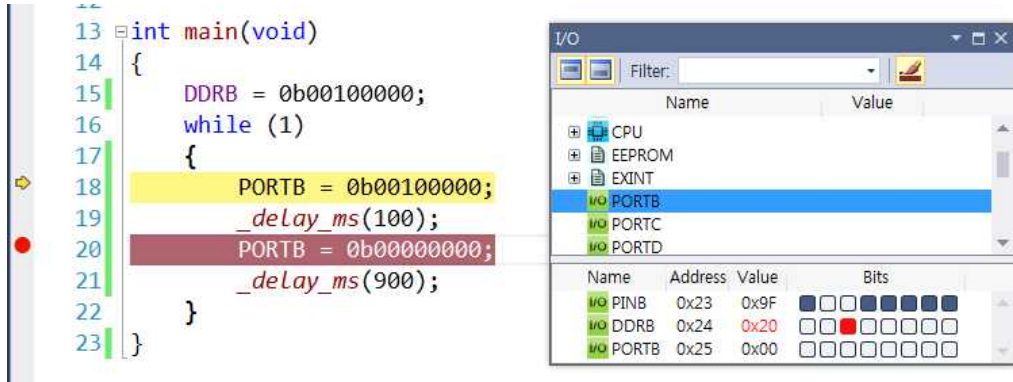


(20) 15번 라인의 statement가 실행을 완료한 후, 18번 라인에서 프로그램의 실행은 멈추게 되고, 갱신된 입출력 레지스터의 값들이 적색으로 표시된다. 이런 동작을 반복하여 프로그램의 논리적인 오류를 쉽게 찾아낼 수 있다.





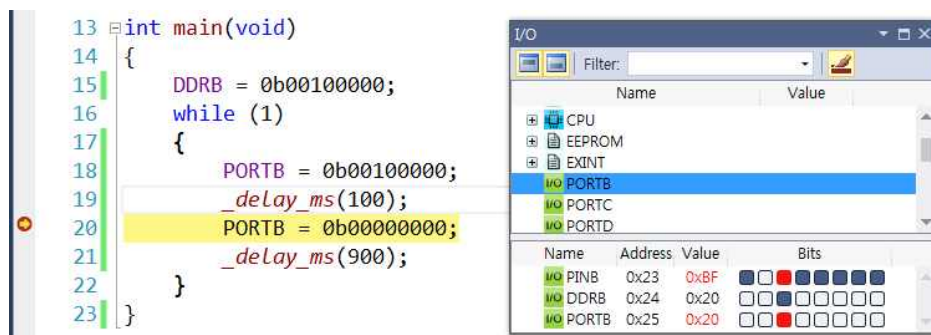
(21) 이후의 모든 statement를 실행한 후 20번 라인에서 실행을 멈추기 위해 20번 라인의 가장 왼쪽 부분을 클릭하면 그림과 같이 20번 라인에 Break Point가 설정된다.



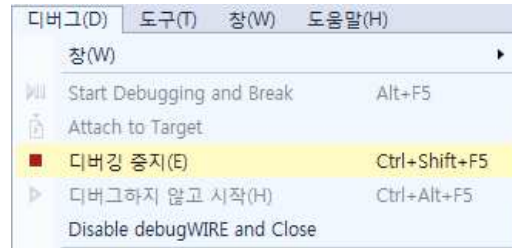
(22) Break Point가 설정되었으면 ①디버그 - ②Continue 항목을 선택한다.



(23) 20번 라인에서 프로그램의 실행이 멈추게 되고, 변경된 레지스터의 내용이 I/O 창에 적색으로 표시된다.



(24) 디버깅을 종료하려면 ①디버그 - ②디버깅 중지 항목을 선택한다.



(25) 위 항목들을 적절하게 조합하여 실행함으로써 프로그램의 실행 상태를 이해할 수 있고, 논리적인 오류도 검출할 수 있다.



2. Microchip Studio for AVR과 Assembly 언어를 사용하여 ATmega328PB Xplained Mini Board에 설치되어 있는 LED를 점멸시키는 프로그램을 만들고, 디버깅을 해 본다.

프로그램을 만드는 절차는 C언어와 유사하고 이번 주 동영상에 상세하게 설명되어 있으므로 설명은 생략함.

```

; blinky.asm

start:      sbi      DDRB,5          ; PB5를 출력 모드로 설정
;
lp:         sbi      PORTB,5         ; PB5에 논리값 '1'이 출력
           rcall    dly100ms        ; 100ms의 시간 지연

           cbi      PORTB,5         ; PB5에 논리값 '0'이 출력
           rcall    dly900ms        ; 900ms의 시간 지연

           rjmp     lp

;=====
dly900ms:
           push     r25
           ldi      r25,9
dly900ms_1:
           call     dly100ms
           dec      r25
           brne     dly900ms_1
           pop      r25
           ret

;-----
dly100ms:
           push     r25
           ldi      r25,100
dly100ms_1:
           call     dly1ms
           dec      r25
           brne     dly100ms_1
           pop      r25
           ret

;-----
; total # of clk = 14 + 4*3999 + 3 = 16013
           .equ     count_1ms = 4000
dly1ms:    push     r24                ; 2  subroutine
           push     r25                ; 2
;
           ldi      r24,LOW(count_1ms) ; 1
           ldi      r25,HIGH(count_1ms) ; 1
dly1ms_1:
           sbiw     r24,1                ; 2
           brne     dly1ms_1            ; 2 / 1
;
           pop      r25                ; 2
           pop      r24                ; 2
           ret                          ; 4
;=====

```

## 과제

- 1) C 언어에서 사용되는 다음 연산자들의 동작에 대해 예제를 들어 요약 설명하여 제출하시오.
  - (1) 산술 연산자 (+, -, \*, /, %)
  - (2) 대입 연산자 (+=, -=, \*=, /=)
  - (3) 관계(비교) 연산자 (>, <, >=, <=, ==, !=)
  - (4) 논리 연산자 (&&, ||, !)
  - (5) 비트 연산자 (&, |, ^, ~)
  - (6) 시프트 연산자 (<<, >>)
  - (7) 기타 연산자 ( sizeof, ?: )
  
- 2) C 언어에서 사용되는 while loop, for loop 및 if else 문에 대해 예제를 들어 요약 설명하여 제출하시오.
  
- 3) C 언어에서 사용되는 #include와 #define에 대해 예제를 들어 요약 설명하여 제출하시오.