

Chapter 103. Hello Blink

이 장에서는 아트멜 스튜디오를 이용하여 C 스타일과 아두이노 스타일의 프로그램을 작성하는 방법을 알아본다. 아트멜 스튜디오는 기본적으로 C 스타일의 프로그램 작성을 지원하므로 먼저 C 스타일의 프로그램 작성 방법에 대해 알아보고, 이어서 애드온을 설치하여 아두이노 스타일의 프로그램을 작성하는 방법에 대해 알아볼 것이다. 이 장은 아트멜 스튜디오의 사용 방법을 살펴보는 장이므로 프로그램의 내용에 대해서는 상세히 설명하지 않으며 아두이노의 'Hello World'에 해당하는 'Blink' 프로그램을 사용할 것이다.

1. C 스타일

아트멜 스튜디오 설치가 끝났으면 이제 첫 번째 프로그램을 작성해 보자. 아트멜 스튜디오를 실행시키고 '파일 → 새로 만들기 → 프로젝트...'를 선택한다. 'GCC C Executable Project'를 선택하고 프로젝트의 이름과 솔루션의 이름을 'HelloWorld'로 지정해 보자. 솔루션이 저장될 위치도 지정해 준다.

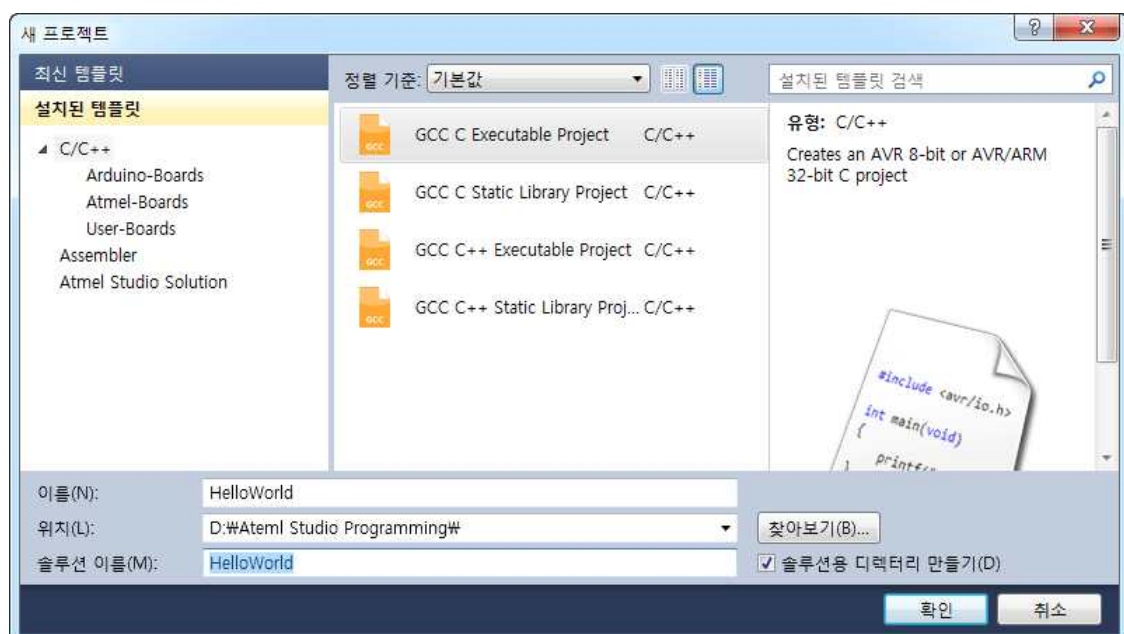


그림 1. 프로젝트 설정

‘확인’ 버튼을 누르면 마이크로컨트롤러를 선택하는 창이 나타난다. 사용할 마이크로컨트롤러로 아두이노 우노에 사용되는 마이크로컨트롤러인 ATmega328P를 선택한다.

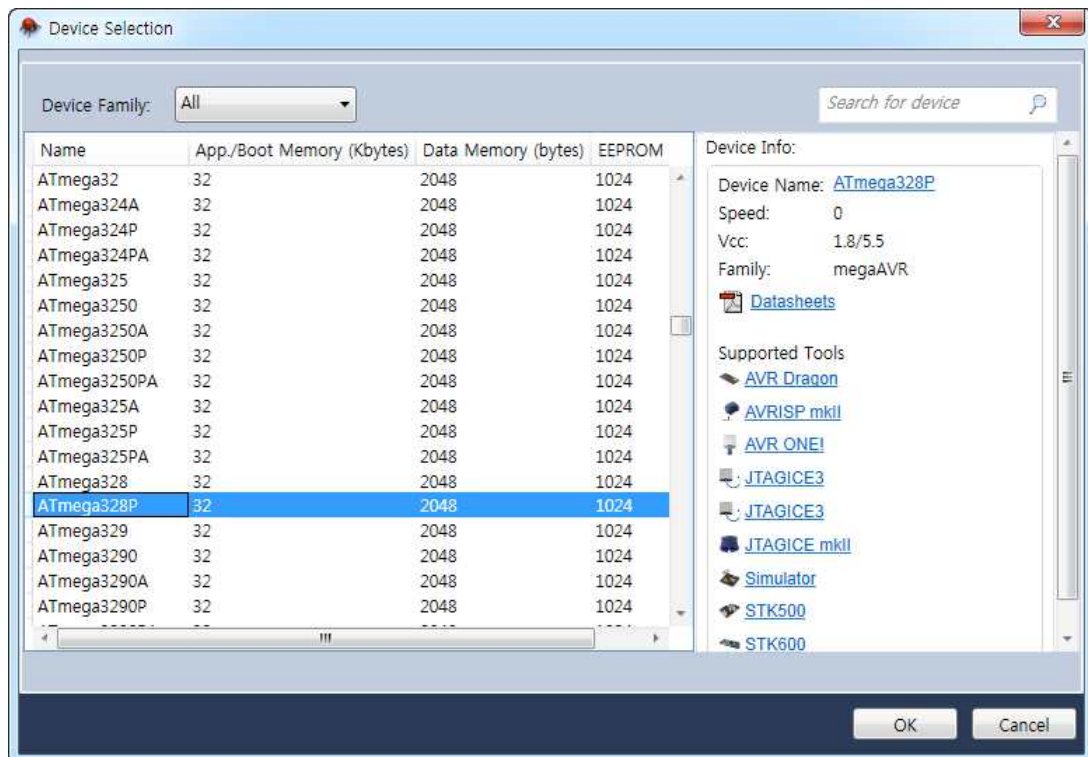


그림 2. 마이크로컨트롤러 선택

‘OK’ 버튼을 누르면 프로젝트가 생성되고 코드 창에 HelloWorld.c가 나타난다.

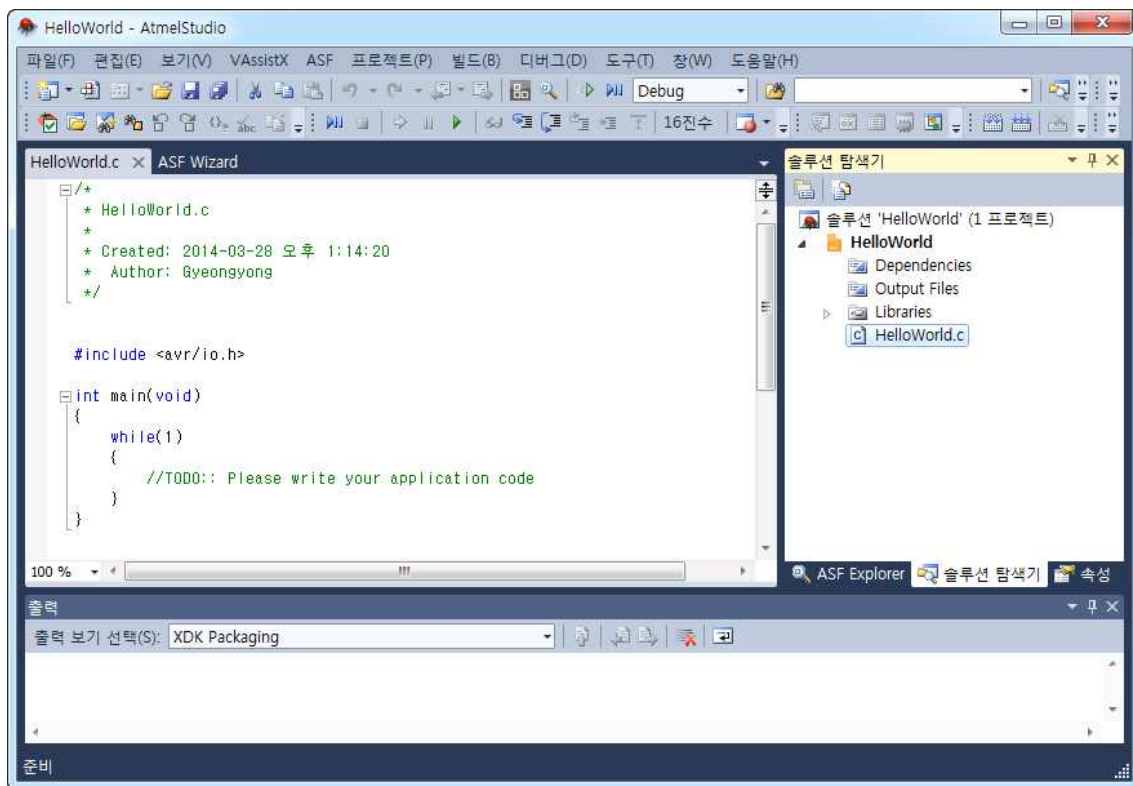


그림 3. 프로젝트 생성 초기 화면

C 스타일의 프로그래밍에서는 ISP 방식으로 프로그램을 다운로드 하며, 아두이노 우노에는 ISP 방식의 프로그램을 다운로드 하기 위한 장치가 포함되어 있지 않으므로 별도로 ISP 장치를 연결해 주어야 한다. 아두이노 우노 보드의 ICSP 커넥터에 USB-ISP 장치를 연결하자.

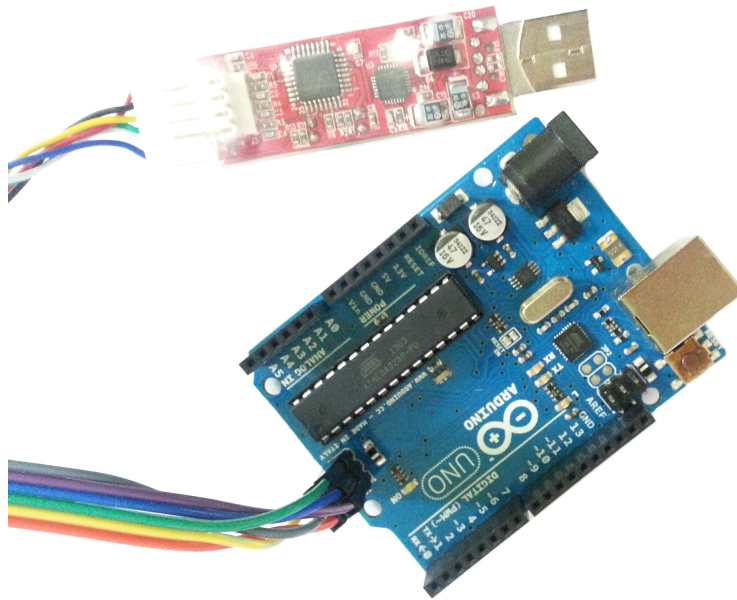


그림 4. 아두이노 우노와 USB-ISP 장치 연결

아트멜 스튜디오의 'HelloWorld.c' 코드 창에 코드 1을 입력한다. 코드 1은 아두이노 우노의 디지털 13번 핀(PB5)에 연결된 LED를 1초 간격으로 점멸하는 코드이다.

코드 1. HelloWorld

```
#define F_CPU 16000000L
#include <avr/io.h>
#include <util/delay.h>

int main(void)
{
    DDRB = 0x20;

    while(1)
    {
        //TODO:: Please write your application code
        PORTB = 0x00;    // 끄기
        _delay_ms(1000);
        PORTB = 0x20;    // 켜기
        _delay_ms(1000);
    }
}
```

```

    }

    return 1;
}

```

코드를 입력한 후 ‘파일 → 저장’을 선택하여 변경된 내용을 저장한다. 이제 ‘빌드 → 솔루션 빌드’ 메뉴, F7 키, 또는 툴바의 솔루션 빌드 버튼()을 눌러 코드를 컴파일 해보자. 코드를 정확히 입력하였다면 에러 없이 성공적으로 컴파일이 되어 기계어 파일이 생성된다.

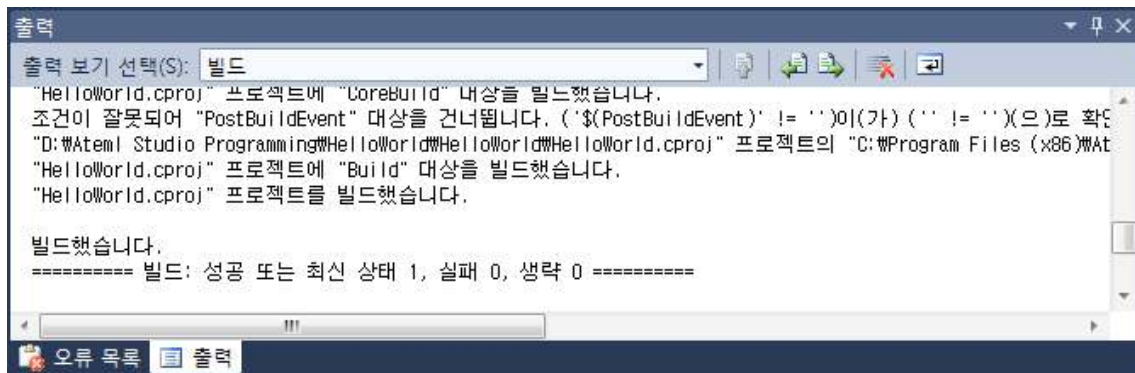


그림 5. 컴파일 메시지

마이크로컨트롤러를 위한 기계어 파일은 ‘HEX’를 확장자로 가진다. 솔루션을 생성한 디렉터리로 가보자. 그림 1에서 지정한 솔루션 이름인 ‘HelloWorld’ 디렉터리가 보이고 그 아래에 프로젝트 이름인 ‘HelloWorld’가 보일 것이다. 프로젝트 디렉터리 아래에는 소스 코드가 위치하고 컴파일된 결과는 그 아래 ‘Debug’ 디렉터리에 생성된다. 그림 1과 같이 프로젝트를 생성한 경우 HEX 파일의 위치는 ‘D:\Atmel Studio Programming\HelloWorld\HelloWorld\Debug\HelloWorld.hex’가 된다.

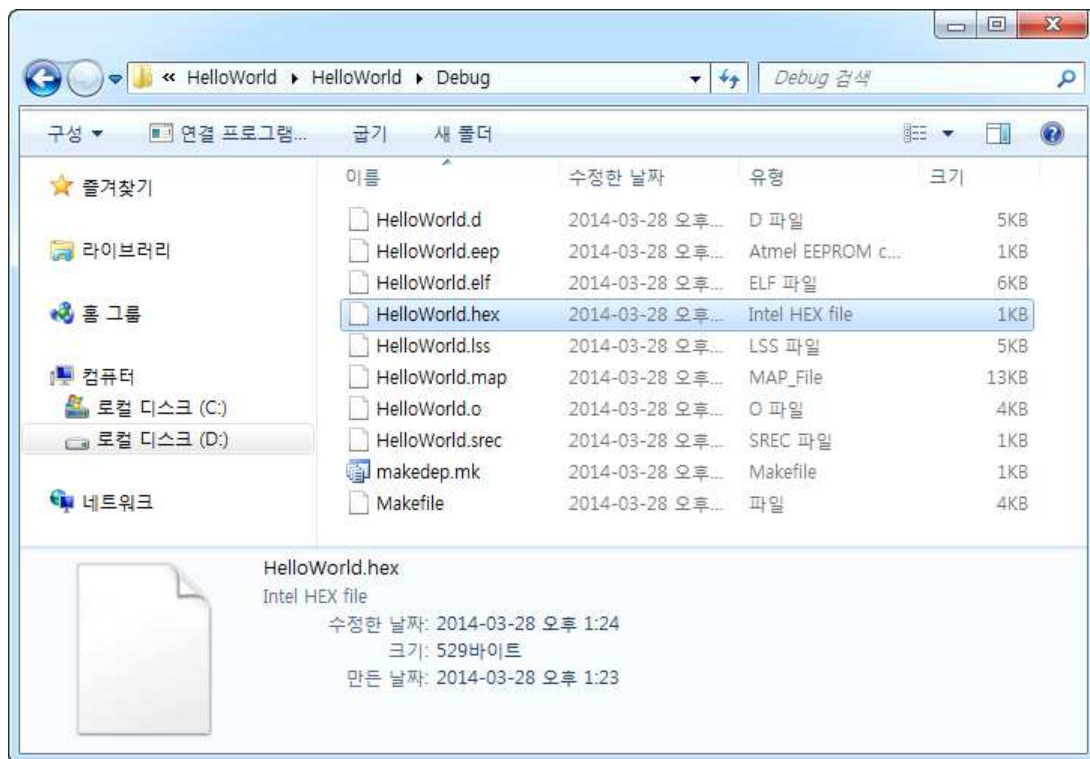


그림 6. HEX 파일 생성 디렉터리

이제 생성된 기계어 파일을 아두이노 우노 보드로 다운로드 할 차례이다. 아트멜 스튜디오에서 '도구 → Add target...'을 선택해 보자. 이 책에서 사용할 USB-ISP 장치는 STK500 프로토콜을 사용하는 장치이므로 USB-ISP 장치에 할당된 COM 포트를 선택하고 'Apply' 버튼을 눌러주면 프로그램을 다운로드 하기 위해 사용할 장치가 아트멜 스튜디오에 등록된다.

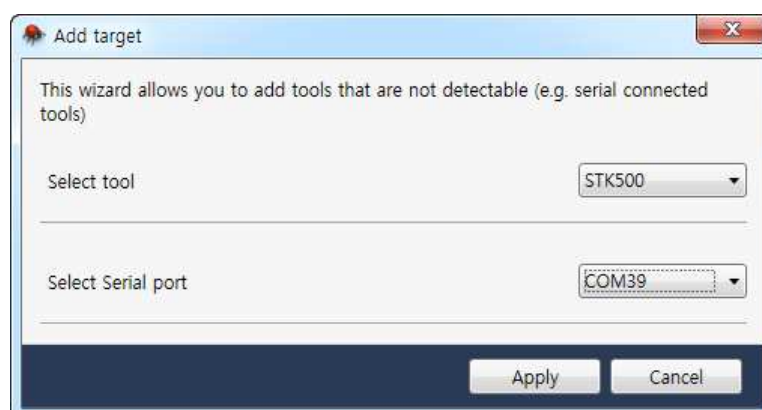


그림 7. Add target 창

사용할 장치가 등록되었으면 ‘도구 → Device Programming’ 또는 툴바의 Device Programming 버튼()을 눌러 보자. 등록된 ‘STK500 COM39’를 선택하고 장치(Device)는 ATmega328P를, 인터페이스는 ISP를 선택한 후 ‘Apply’ 버튼을 누르면 그림 8과 같은 창이 나타난다.

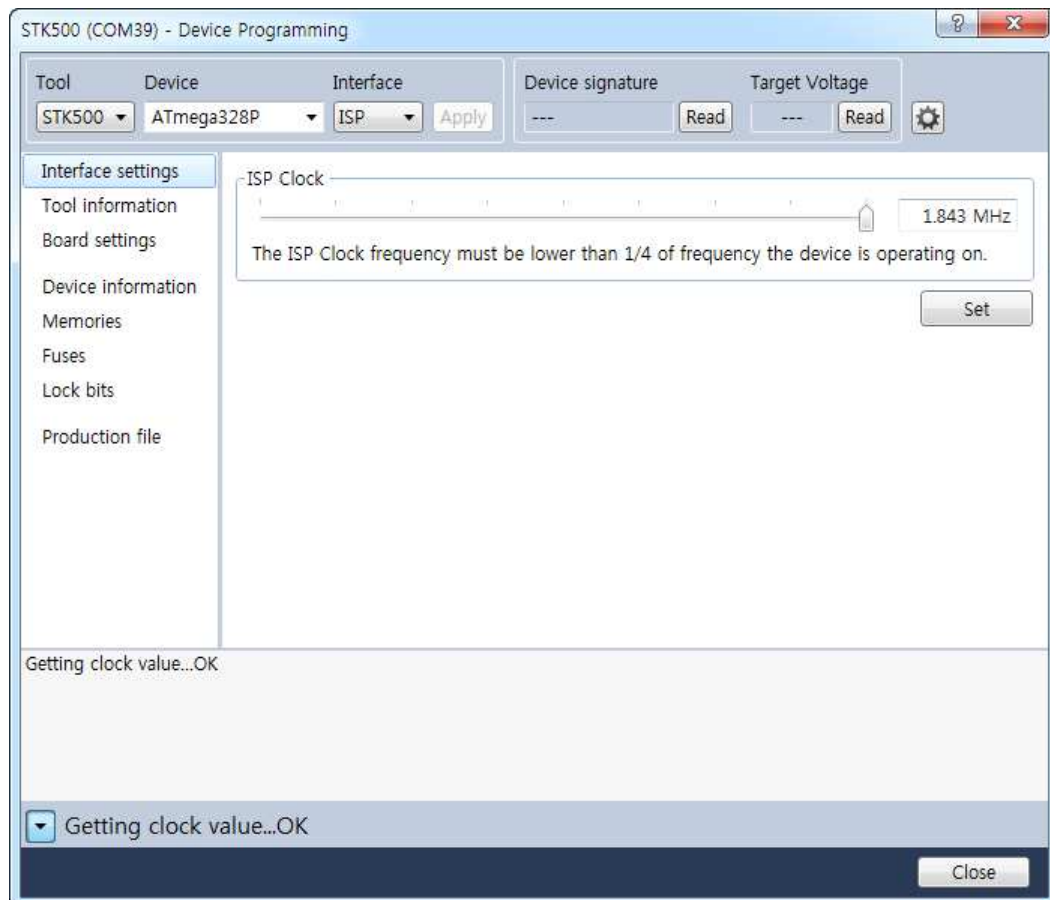


그림 8. Device Programming 창

‘Memories’ 탭을 선택하고 ‘Flash’ 부분에 컴파일로 생성된 HelloWorld.hex 파일을 지정한 후 ‘Program’ 버튼을 눌러보자.

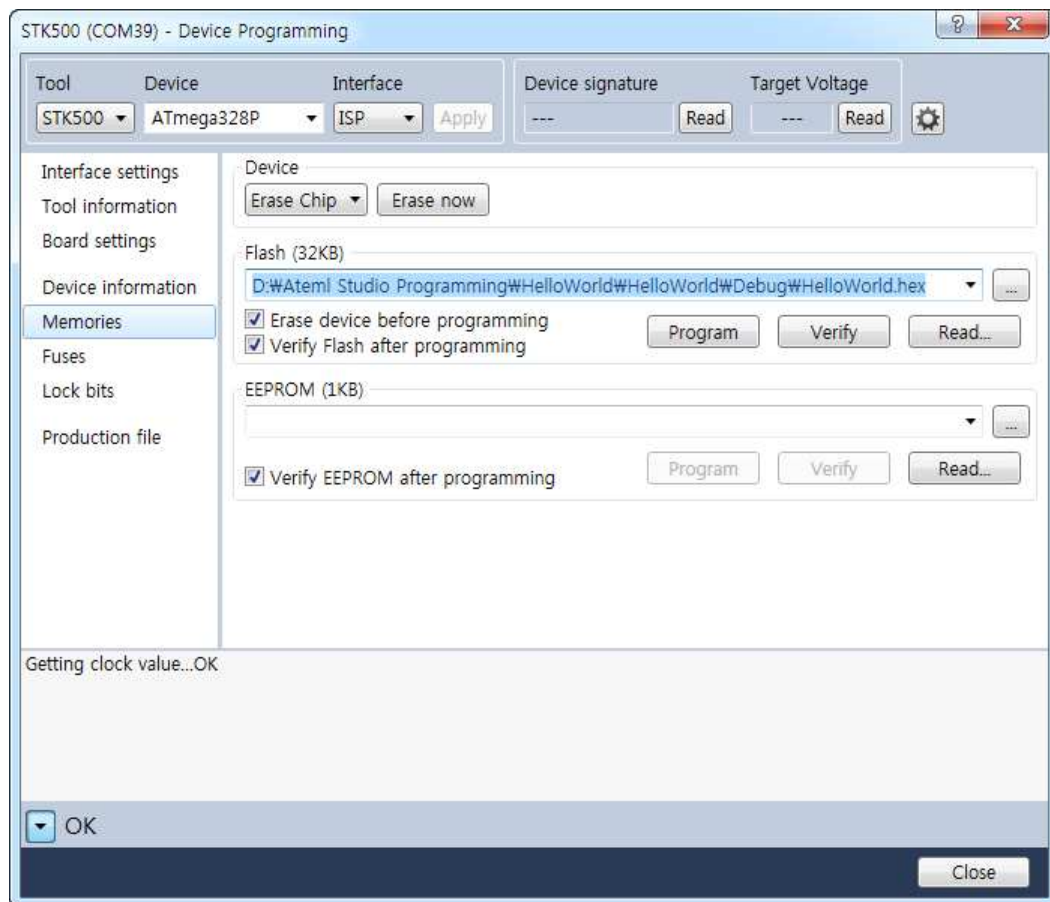


그림 9. HEX 파일 선택

‘Program’ 버튼을 누르면 기계어 파일이 다운로드 되고, 다운로드가 성공적으로 이루어지면 아두이노 우노의 디지털 13번 핀(PB5)에 연결된 LED가 1초 간격으로 깜빡이는 것을 확인할 수 있다.

그림 8의 ‘Device Programming’ 윈도우에서 ‘Fuses’ 탭을 선택해 보자. ‘Fuse Register’를 살펴보면 ‘LOW’, ‘HIGH’, ‘EXTENDED’라는 3개의 항목을 확인할 수 있다. 그림 10은 아두이노 우노에 사용된 ATmega328의 퓨즈 값이다.

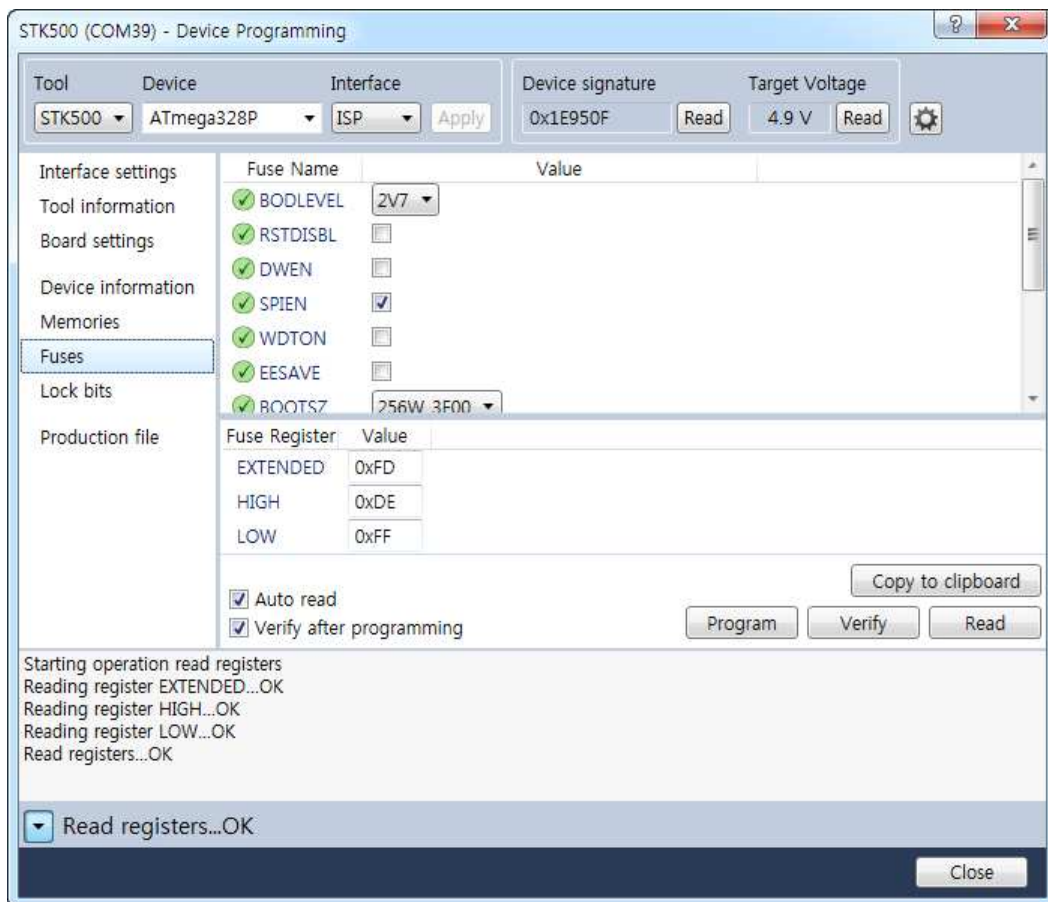


그림 10. 아두이노 우노에 사용된 ATmega328의 퓨즈 값

이전 장에서 새로 구입한 ATmega328은 내부 1MHz 발진기를 사용하도록 설정되어 있어 1초 간격이 아니라 16초 간격으로 LED가 깜빡였던 것을 기억하는가? 브레드보드에 구성한 기본 회로를 컴퓨터와 연결하고 그림 10과 같이 퓨즈 값을 확인해 보자. 이 때 주의할 점은 ‘Interface settings’ 탭에서 ISP 클록을 설정하는 부분이다. 클록 설정의 아래쪽 설명을 보면 “ISP 클록은 ATmega328 동작 클록의 1/4보다 작아야 한다.”고 되어 있다. 즉, ATmega328의 초기 설정은 1MHz로 동작하도록 설정되어 있으므로 ISP 클록은 250KHz보다 작은 값으로 설정하여야 한다. ISP 클록을 설정하고 퓨즈 값을 확인해 보면 그림 10과는 다른 값을 가짐을 알 수 있다.

	아두이노 우노 보드의 ATmega328	공장 출하시의 ATmega328
EXTENDED	0xFD	0xFF
HIGH	0xDE	0xD9
LOW	0xFF	0x62

표 1. 퓨즈 값 비교

퓨즈 값에 아두이노 우노 보드에 사용된 값을 입력하고 ‘Program’ 버튼을 누르면 퓨즈 설정이 완료되며 외부 16MHz 클록을 사용하도록 변경된다. 다시 ‘Memories’ 탭에서 HEX 파일을 업로드 해보자. 디지털 13번 핀(PB5)에 연결된 LED가 1초 간격으로 깜빡거리는 것을 확인할 수 있을 것이다.

‘Device Programming’ 다이얼로그에서 ‘Memories’ 탭을 선택하면 플래시 메모리에 업로드 할 실행 파일이 기본적으로 ELF 파일이 선택되어 있음을 볼 수 있다. ELF는 Executable and Linking Format의 약어로 아트멜 스튜디오에서 사용하는 avr-gcc에서 컴파일의 결과로 생성하는 파일 형식을 나타낸다. avr-objcopy 프로그램은 생성된 ELF 파일을 읽어 최종 HEX 파일로 변환한다.

기계어 파일이 생성되는 디렉터리를 살펴보면 ELF 파일에는 디버깅 정보 등이 포함되어 있어 HEX 파일에 비해 크기가 크다는 사실을 알 수 있다. ELF 파일을 마이크로컨트롤러로 업로드 해도 실행에는 문제가 없지만 크기가 큰 실행 파일의 경우 HEX 파일을 사용하는 것이 좋다.

2. 아두이노 스타일

아트멜 스튜디오는 다양한 애드온을 설치할 수 있어 아트멜 스튜디오에서 지원하지 않는 기능들을 사용할 수 있으며 아두이노 환경 지원 역시 그 중 하나이다. 아트멜 스튜디오를 실행하고 ‘도구 → 확장 관리자...’ 메뉴를 선택해 보자.

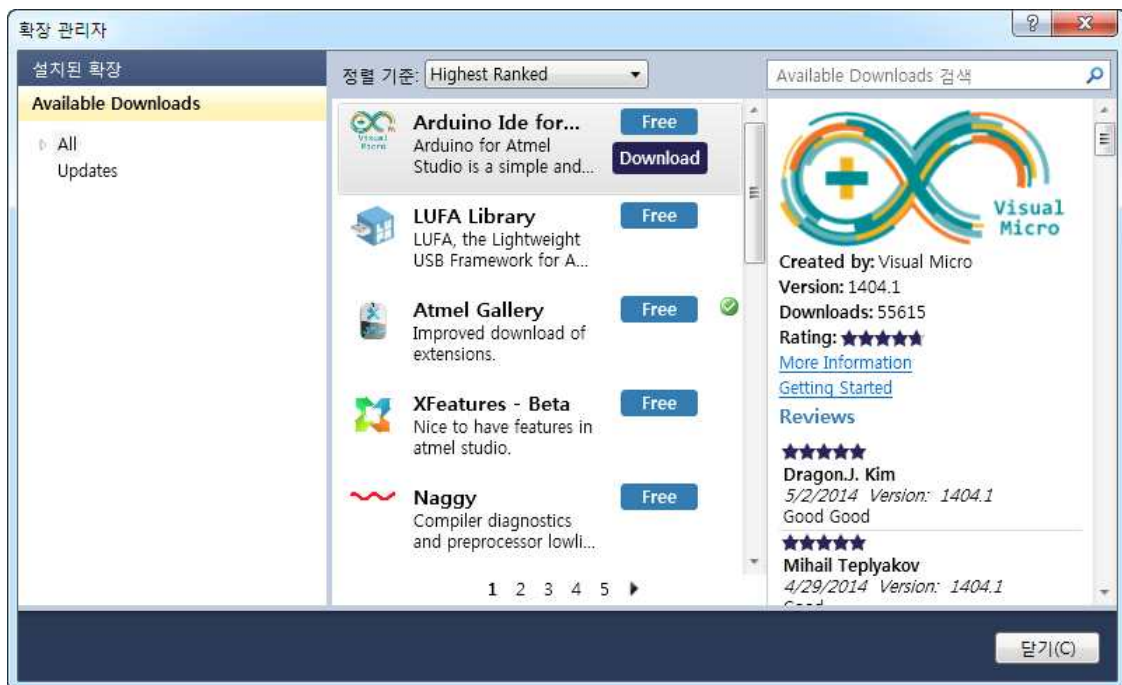


그림 11. 확장 관리자

확장 관리자에서 'Arduino Ide for Atmel Studio'를 선택하여 설치하자. 아트멜 스튜디오를 다시 실행하면 아두이노를 위한 설정 창이 나타난다. 아두이노 프로그램이 설치된 디렉터리와 스케치북 디렉터리를 지정한다. 아두이노 프로그램의 버전 역시 선택할 수 있으므로 설치된 아두이노 프로그램 버전을 선택하고 'OK' 버튼을 누른다.

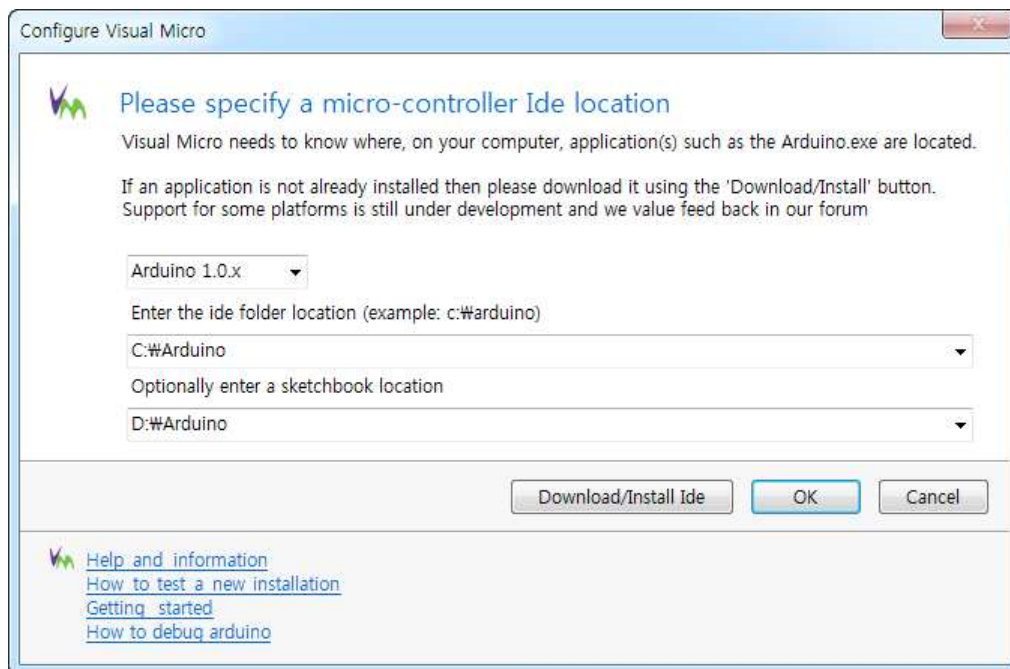


그림 12. Arduino Ide for Atmel Studio 설정

‘파일 → 새로 만들기’를 선택하면 ‘Sketch Project’라는 새로운 메뉴가 추가된 것을 확인할 수 있다. ‘파일 → 새로 만들기 → Sketch Project’ 메뉴를 선택하면 스케치 프로젝트의 이름을 입력하는 창이 표시된다.

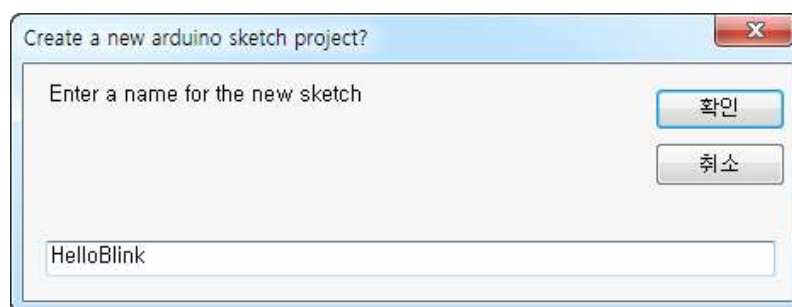


그림 13. 스케치 프로젝트 이름

‘HelloBlink’라고 입력하고 ‘확인’ 버튼을 누르면 그림 12에서 입력한 스케치 디렉터리에 새로운 프로젝트가 생성된다. HelloBlink.ino 파일에는 자동으로 setup 함수와 loop 함수의 정의가 나타난다.

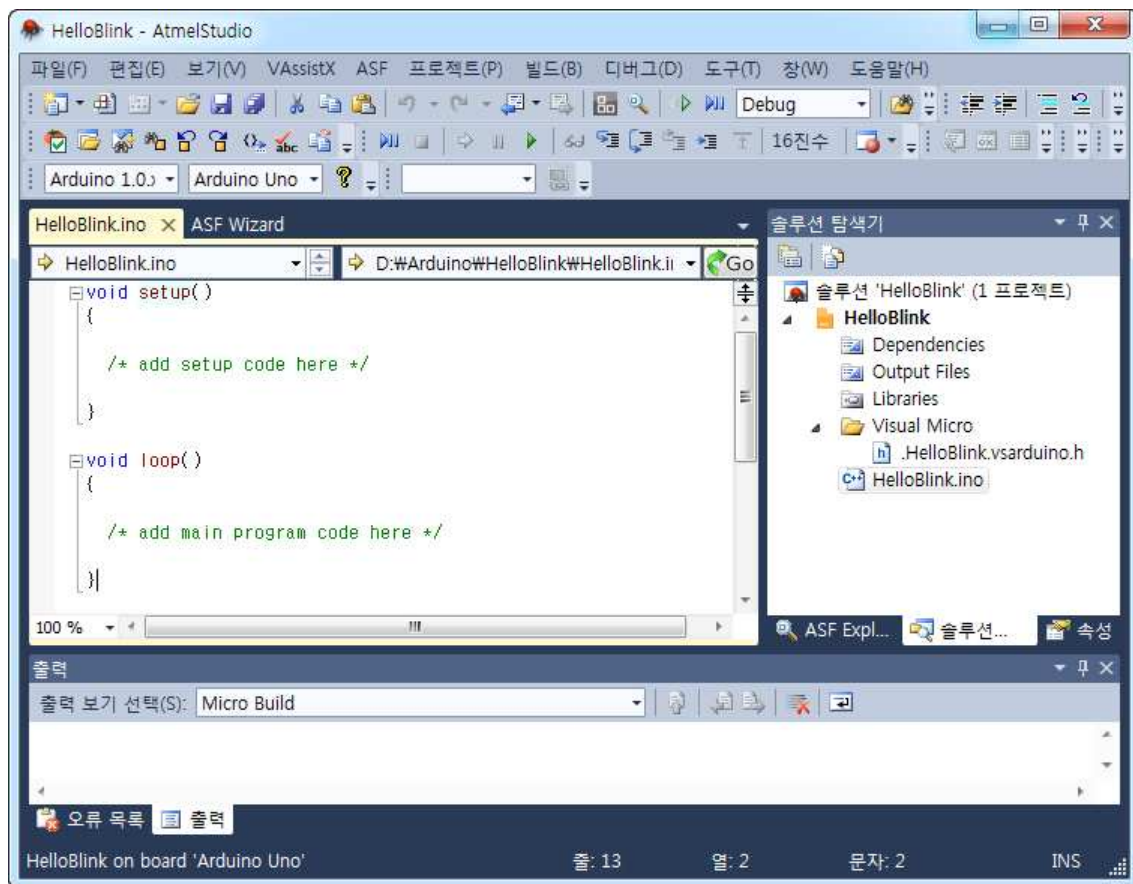


그림 14. HelloBlink 프로젝트 생성

아트멜 스튜디오에의 툴바 영역에도 새로운 항목이 추가되어 있다. 툴바 영역에서 마우스 오른쪽 버튼을 눌러보면 'Arduino Boards'와 'Arduino Serial Communication'이 추가된 것을 확인할 수 있다.



그림 15. 아트멜 스튜디오 툴바 메뉴

‘Arduino Boards’ 툴바 메뉴는 아두이노 프로그램 버전을 나타내는 콤보 박스와 아두이노 보드의 종류를 선택하는 콤보 박스로 구성되어 있다. 프로그램 버전은 1.0.x 버전을 선택하고, 아두이노 보드는 ‘Arduino Uno’를 선택한다. ‘Arduino Serial Communication’ 툴바 메뉴는 아두이노와 연결된 시리얼 포트를 선택하는 콤보 박스와 아두이노의 시리얼 모니터 프로그램과 유사한 시리얼 모니터를 실행시키는 버튼으로 구성된다. 아두이노 연결 포트는 현재 아두이노와 USB를 통해 컴퓨터와 연결된 시리얼(COM) 포트를 선택해 준다.

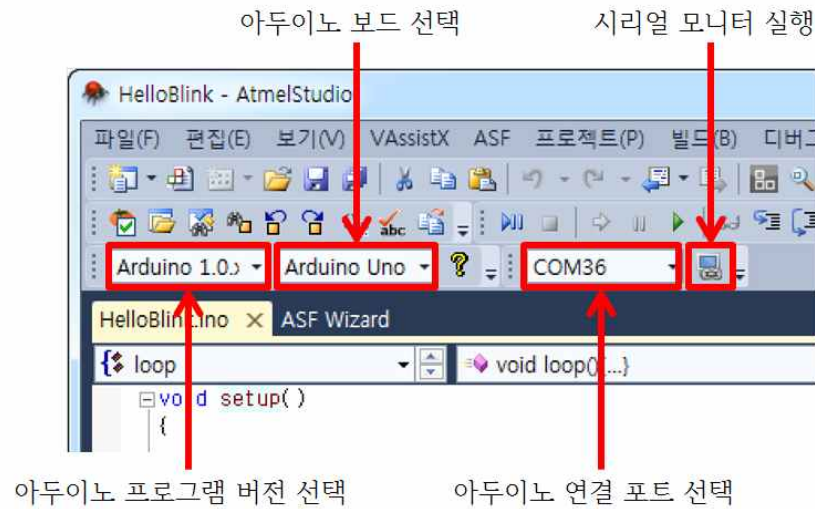


그림 16. Arduino Ide for Atmel Studio의 툴바 버튼

코드 2를 입력해 보자. 코드 2는 0.5초 간격으로 아두이노 우노의 디지털 13번 핀(PB5)에 연결된 LED를 점멸하는 코드이다.

코드 2. HelloBlink

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

void loop()
{
    digitalWrite(13, HIGH);       // LED 켜기
    delay(500);
    digitalWrite(13, LOW);        // LED 끄기
    delay(500);
}
```

코드를 입력한 후 '파일 → 저장' 메뉴 또는 'Ctrl + S' 키를 눌러 변경된 내용을 저장하고 '빌드 → 솔루션 빌드' 메뉴, F7 키, 또는 툴바의 솔루션 빌드 버튼()을 눌러 프로그램을 컴파일 해보자. 오류가 없다면 아두이노 IDE에서와 마찬가지로 프로그램의 크기와 프로그램이

전체 플래시 메모리에서 차지하는 비율을 출력해 준다.

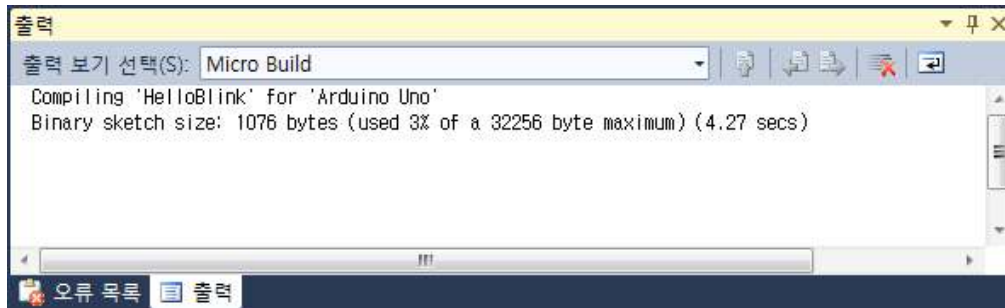


그림 17. 컴파일 메시지

아두이노 스타일의 프로그램 역시 컴파일이 성공적으로 끝나면 'HEX' 확장자를 가지는 기계어 파일이 생성된다. 기계어 파일은 임시 디렉터리에 생성되며 스케치북 디렉터리에 생성되지 않는다. 오류가 없다면 아두이노 우노 보드에 프로그램을 업로드 해 보자. 프로그램을 업로드를 위해서는 시리얼 방식과 ISP 방식의 두 가지를 사용할 수 있다. 시리얼 방식으로 업로드하기 위해서는 '디버깅 시작 (F5)'이나 툴바의 디버깅 시작 버튼(▶) 또는 '디버깅하지 않고 시작 (Ctrl + Alt + F5)'이나 툴바의 디버깅하지 않고 시작 버튼(▶)을 사용하면 된다. 확장 관리자를 통해 설치한 아두이노 애드온은 무료 버전으로 디버깅을 지원하지 않으므로 두 가지 방법 사이에 아무런 차이가 없다. 아두이노가 USB를 통해 컴퓨터와 연결된 포트를 툴바 콤보 박스에서 확인하고 F5 키를 눌러보자. LED가 깜빡거리는가?

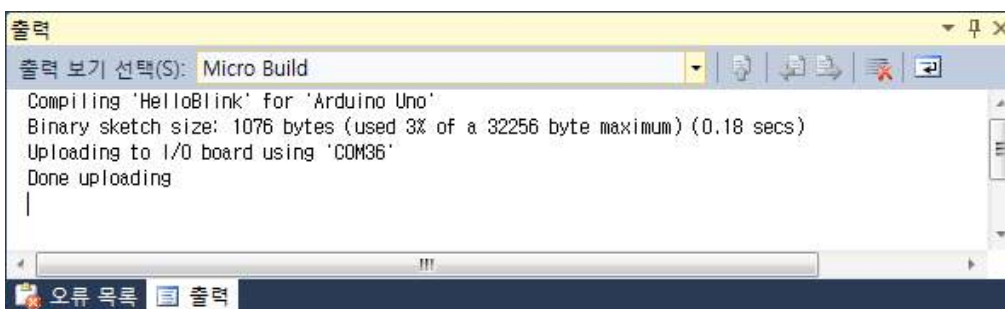


그림 18. USB를 통한 프로그램 업로드 메시지

ISP 장치를 이용하여 프로그램을 업로드 해보자. Arduino Ide for Atmel Studio는 아두이노 프로그램의 정보를 사용하므로 ISP를 통해 프로그램을 업로드 하기 위해서는 먼저 아두이노 프로그램에서 ISP 장치를 '도구 → 프로그래머' 메뉴에 등록시켜야 한다. 사용할 ISP 장치가

아두이노 프로그램에서 'My AVRISP'라는 이름으로 등록되어 있다고 가정하자. 등록 방법은 아두이노 프로그래밍 장을 참고하면 된다.

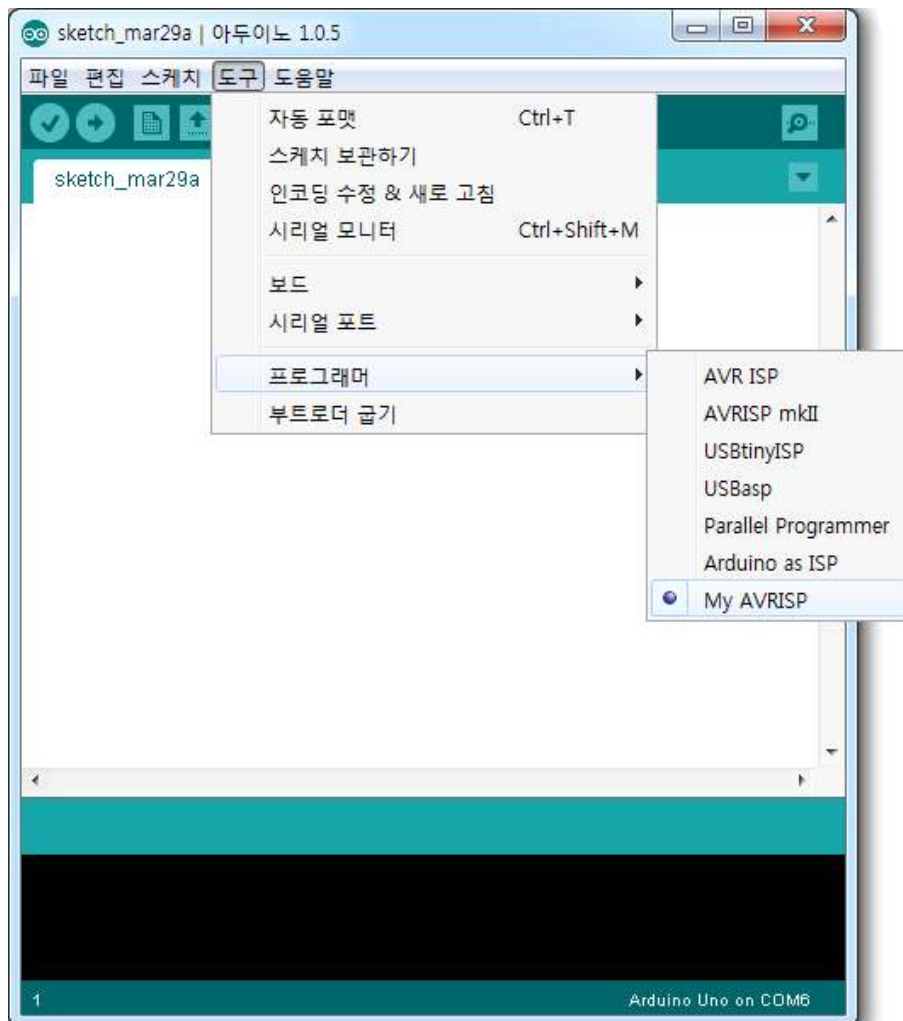


그림 19. 아두이노 프로그램에서 ISP 장치 등록

아트멜 스튜디오에서 '도구 → Visual Micro (Arduino 1.0.x) → Programmers' 메뉴를 선택하고 사용할 ISP 장치인 'My AVRISP'를 선택한다.

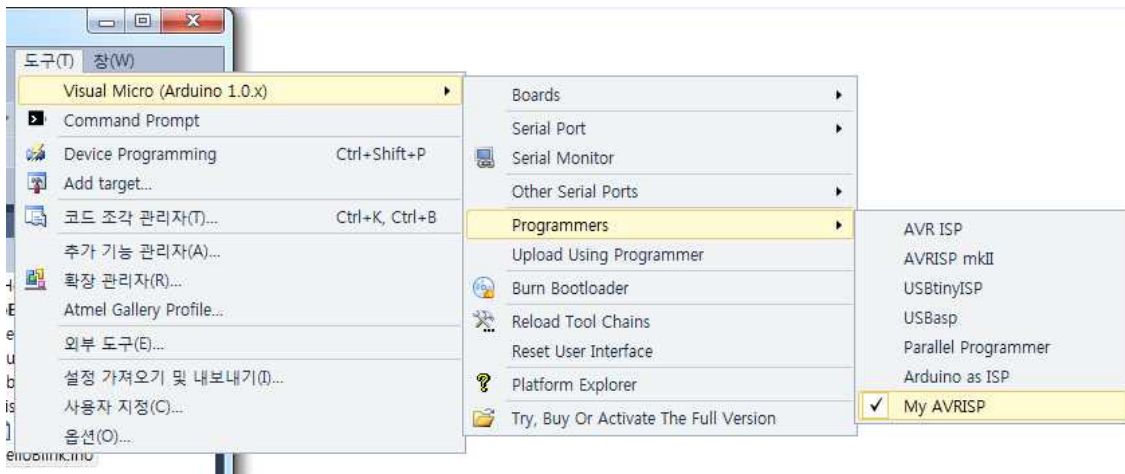


그림 20. ISP 장치 선택

다음은 '도구 → Visual Micro (Arduino 1.0.x) → Upload Using Programmer'를 선택한다. Upload Using Programmer는 토클 메뉴로 시리얼과 ISP 방식을 선택할 수 있도록 해준다.

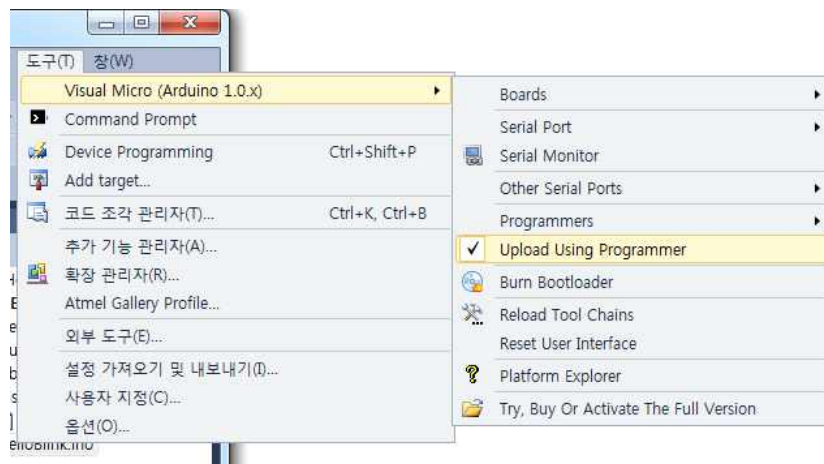


그림 21. Upload Using Programmer 메뉴

ISP 장치가 컴퓨터와 연결된 포트를 툴바 콤보박스에서 확인하고 F5 키를 눌러보자. ISP를 통해 프로그램을 업로드 하는 경우에는 아두이노의 부트로더를 사용하지 않으며 부트로더가 구워져 있는 상태라면 프로그램을 업로드 할 때 부트로더가 지워진다는 점을 기억하여야 한다.

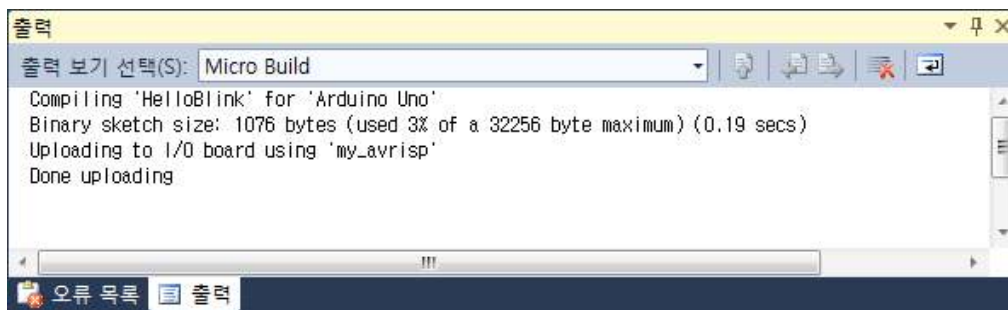


그림 22. ISP를 통한 프로그램 업로드 메시지

부트로더가 지워졌다고 해도 걱정할 필요는 없다. ‘도구 → Visual Micro (Arduino 1.0.x) → Burn Bootloader’ 메뉴를 선택함으로써 부트로더를 다시 구울 수 있다. 부트로더를 구울 때는 ‘Upload Using Programmer’ 메뉴가 선택되어 있어야 한다.

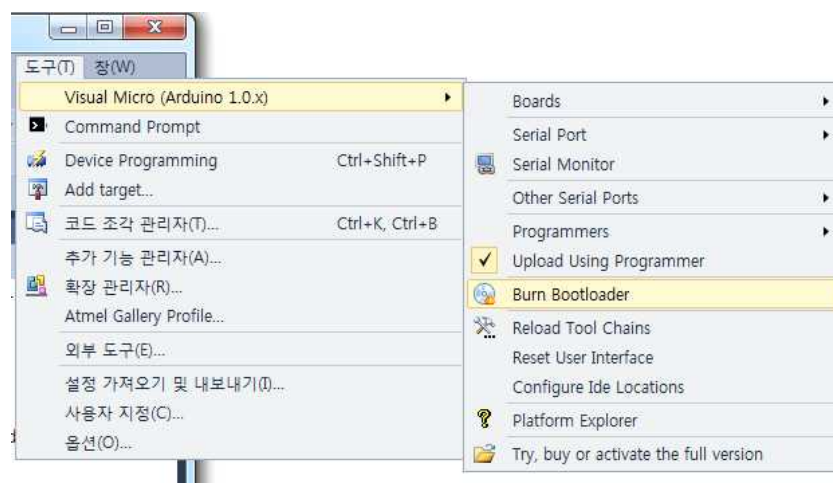


그림 23. Burn Bootloader 메뉴

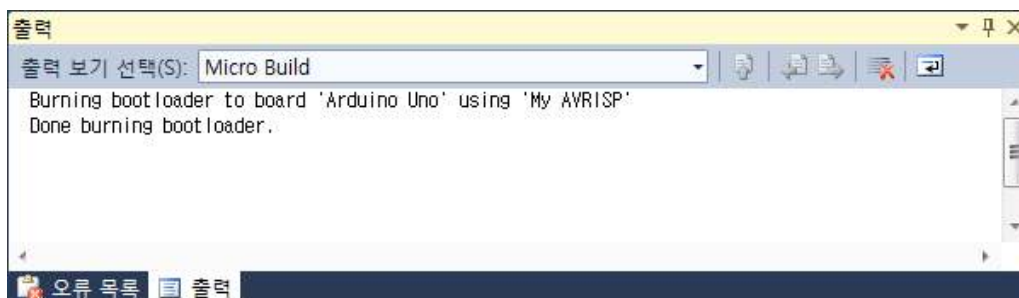


그림 24. 부트로더 굽기 메시지

아두이노의 프로그램의 장점 중 한 가지는 시리얼 모니터가 포함되어 있어 시리얼 포트에 송수신되는 메시지를 손쉽게 확인할 수 있다는 점이다. Arduino Ide for Atmel Studio에도 이와 동일한 기능을 하는 시리얼 모니터를 제공하고 있다. 먼저 코드 3을 업로드 해 보자. 코드 3은 1초 간격으로 카운터를 증가시키고 이를 시리얼 모니터로 출력하는 프로그램이다.

코드 3. 시리얼 모니터 출력

```
int count = 0;                // 카운터 변수 및 초기화

void setup()
{
    Serial.begin(9600);        // 시리얼 연결 초기화
}

void loop()
{
    count++;                   // 카운터 증가
    Serial.println(count);     // 시리얼 모니터로 출력
    delay(1000);
}
```

프로그램을 업로드 하고 툴바에서 시리얼 모니터 버튼을 눌러보자. 시리얼 모니터는 아두이노의 시리얼 모니터와 비슷해 보이지만 더 많은 옵션을 제공하고 있다. 시리얼 모니터의 우측 하단에 있는 속도 설정 부분을 9600 보율로 조절하면 시리얼 모니터로 1초에 한번씩 카운터가 증가하여 출력되는 것을 확인할 수 있다.



그림 25. 프로그램 3 실행 결과

아두이노 IDE에서와 마찬가지로 아트멜 스튜디오에서도 시리얼 모니터는 아두이노 우노의 하드웨어 시리얼 연결, 즉, USB를 통해 연결된 포트와 연결되어 있으므로 ISP 장치로 프로그램을 업로드 한 경우에는 별도로 연결이 필요하다는 점에 유의하여야 한다.

아트멜 스튜디오의 시리얼 모니터에는 아두이노의 시리얼 모니터에는 없는 여러 가지 기능이 추가되어 있다. 'Connect'는 지정한 직렬 포트에 연결 여부를 나타낸다. 실행 도중 체크를 해제하면 연결이 끊어지고 체크하면 다시 연결된다. 'Dtr'은 연결된 아두이노 보드를 리셋 하기 위해 사용되며 체크를 해제하였다가 다시 체크하면 자동으로 보드가 리셋된다. 'Clear'는 현재 수신된 데이터 창을 지우기 위해 사용되며 체크를 해제하였다가 다시 체크하면 현재 출력된 데이터가 모두 지워진다. 'Reconnect'는 연결이 끊어지면 자동으로 다시 연결을 시도하도록 지정하기 위해 사용된다. 'Echo'는 시리얼 모니터에서 입력하여 아두이노로 전송한 내용을 지정한 포트로도 전송하도록 설정하기 위해 사용되며, 전송한 메시지를 시리얼 모니터로 나타내기 위해 사용할 수 있다. 아두이노 IDE에서는 시리얼 모니터에 입력한 내용을 시리얼 모니터로 에코해주는 기능은 없다. 마지막 두 개의 콤보 박스는 아두이노의 시리얼 모니터와 동일하게 전송되는 메시지의 끝에 추가할 문자열과 전송 속도를 설정하기 위해 사용된다.

이 장에서는 아트멜 스튜디오를 사용하여 C 스타일과 아두이노 스타일의 프로그램을 작성하고 이를 업로드하여 실행하는 방법에 대해 알아보았다. C 스타일의 경우 부트로더를 사용하지 않는 ISP 방식으로만 프로그램을 업로드 할 수 있는 반면, 아두이노 스타일의 경우 부트로더를 사용하지 않는 ISP 방식과 부트로더를 사용하는 시리얼 방식 모두를 사용할 수 있다. ISP 방식을 사용하는 경우 부트로더가 지워져 버리므로 아두이노 스타일의 프로그래밍에서 시리얼 방식의 업로드를 사용하면 매번 부트로더를 다시 구워야 하는 번거로움이 있으므로 이 책에서

는 프로그램 업로드를 위해서는 ISP 방식만을 사용한다. 그리고 시리얼 모니터를 위해 USB-시리얼 변환 장치 역시 컴퓨터와 연결하여 사용한다. 즉, ISP 장치와 USB-시리얼 변환 장치를 모두 컴퓨터와 연결하여 사용할 것이다. 본격적인 프로그래밍에 들어가기 전에, 아두이노 스타일로 부트로더를 굽거나 C 스타일로 퓨즈를 설정하여 ATmega328이 16MHz 외부 클록을 사용하도록 준비되어 있는지 확인하도록 하자.