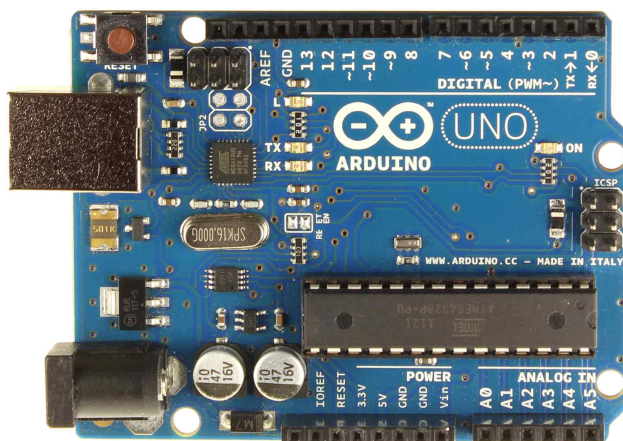


Chapter 300. Arduino Zero Pro

아두이노 제로 프로는 ARM의 Cortex-M0+ 아키텍처를 사용하여 만들어진 아트멜의 ATSAM21 마이크로컨트롤러를 사용하여 제작된 아두이노 보드이다. 아두이노 보드 중 기본 이 라고 할 수 있는 아두이노 우노와 외형을 먼저 비교해 보자.



(a) 아두이노 우노



(b) 아두이노 제로 프로¹⁾

그림 1. 아두이노 우노와 아두이노 제로 프로

아두이노 우노와 아두이노 제로 프로의 외형은 거의 비슷하다. 특히 데이터 입출력과 전원 관

1) 이미지 출처: <http://www.arduino.org>

런 핀 헤더는 두 보드가 동일하다. 하지만 아두이노 우노가 하나의 USB 연결 커넥터를 가지고 있는 반면 아두이노 제로 프로는 두 개의 USB 연결 커넥터를 가지고 있는 점에서 차이가 있다. 또한 아두이노 제로 프로는 디버깅을 지원하기 위해 10핀의 SWD 커넥터와 JTAG 커넥터를 포함하고 있다. 먼저 아두이노 우노에 사용된 ATmega328 마이크로컨트롤러와 아두이노 제로 프로에 사용된 SAMD21 마이크로컨트롤러를 비교하면 표 1과 같다.

항목	아두이노 우노	아두이노 제로 프로
마이크로컨트롤러	ATmega328	ATSAMD21G18
동작 전압	5V	3.3V
플래시 메모리	32 KByte	256 KByte
SRAM	9 KByte	32 KByte
EEPROM	4 KByte	16 KByte (에뮬레이션)
클록	16 MHz	48 MHz
디지털 입출력 핀	14개	14개
PWM 출력 가능 핀	6개	12개
아날로그 입력 핀	6개	6개
ADC 해상도	10비트	12비트
DAC	-	1 (10비트)

표 1. 아두이노 우노와 아두이노 제로 프로

표 1에서 알 수 있듯이 아두이노 제로 프로는 우노와의 호환성을 위해 동일한 개수의 디지털 및 아날로그 핀을 사용할 수 있다. 아두이노 제로 프로는 더 많은 플래시 메모리와 SRAM을 가지고 있어 더 큰 프로그램을 설치하여 실행할 수 있으며, 우노에 비해 3배 빠른 클록을 가지고 있다. 아두이노 제로 프로에는 EEPROM이 포함되어 있지 않지만 에뮬레이션을 통해 16 KByte까지 EEPROM을 사용할 수 있다. 이외에도 아두이노 우노에는 없는 DAC가 포함되어 있어 아날로그 신호를 직접 출력하는 것도 가능하다.

그림 1에서도 알 수 있듯이 아두이노 제로 프로는 프로그래밍 포트와 native 포트, 2개의 USB 연결 커넥터를 가지고 있다. 프로그래밍 포트는 아두이노 우노에서와 마찬가지로 프로그램을 업로드 하고 UART 시리얼 통신을 위해 사용되는 포트이며, native 포트는 USB 연결을 지원하기 위한 포트이다.

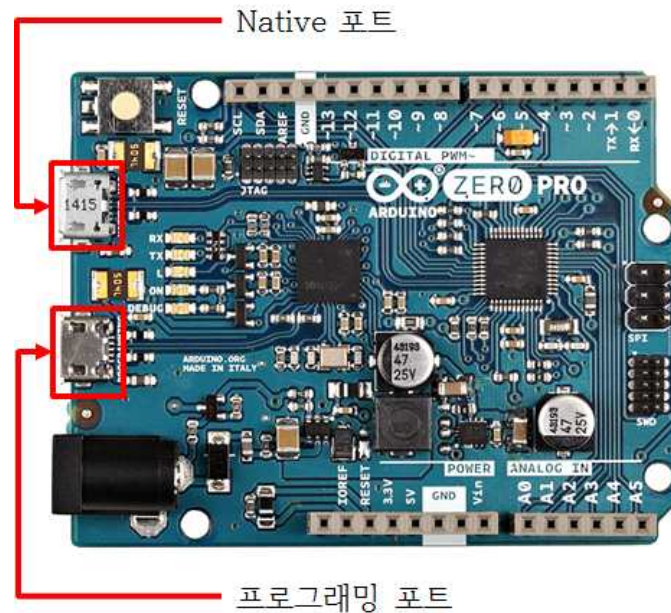


그림 2. 아두이노 제로 프로의 USB 연결 커넥터

프로그래밍 포트를 컴퓨터와 연결해 보자.

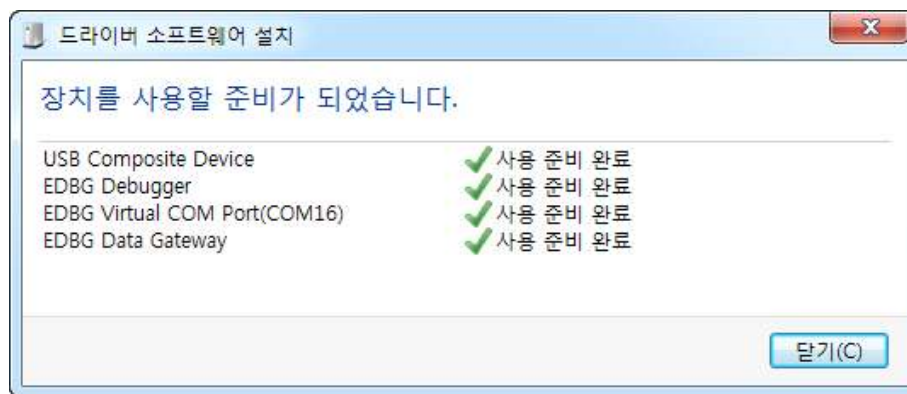


그림 3. 아두이노 제로 프로 프로그래밍 포트 드라이버 설치

아두이노 제로 프로는 디버깅을 지원하고 가상의 COM 포트를 제공하는 USB 복합 장치로 인식된다. 드라이버가 자동으로 설치되고 나면 장치관리자에서 COM 포트가 할당된 것을 확인할 수 있으며, 이 포트를 통해 프로그램을 업로드 하고 시리얼 모니터를 통해 송수신되는 데이터를 확인할 수 있다.

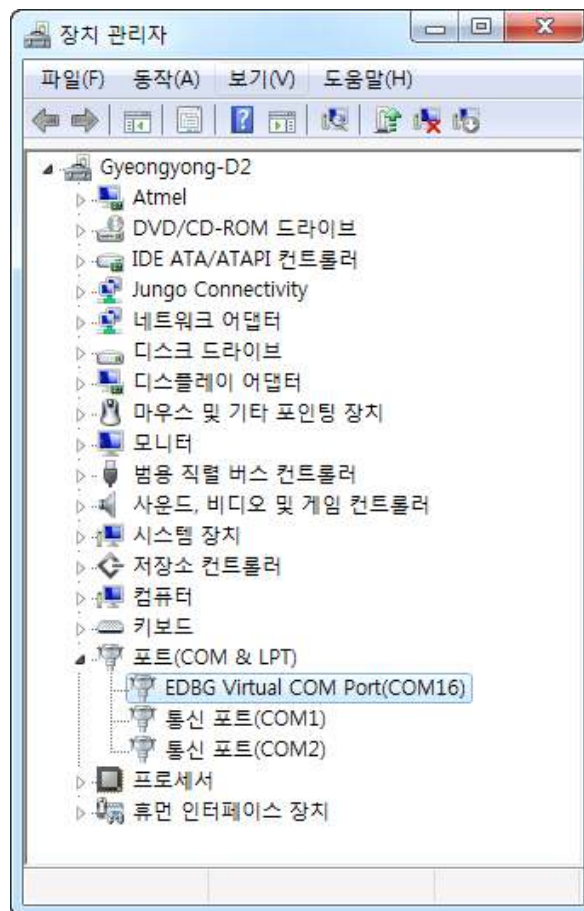


그림 4. 아두이노 제로 프로의 COM 포트

그림 3에서 EDBG는 Atmel Embedded Debugger의 약어로 아트멜에서 만든 키트들의 디버깅을 지원하기 위해 보드에 포함되어 있는 장치를 가리킨다. EDBG가 포함되어 있다면 별도의 외부 장치 없이도 디버깅이 가능하다. 아두이노 제로 보드에는 2개의 비슷한 크기의 칩이 포함되어 있으며 이 중 하나는 아두이노 제로 보드에 사용된 마이크로컨트롤러인 ATSAM21이고 다른 하나가 EDBG를 지원하는 칩이다. EDBG 칩은 아두이노 우노에서 USB-시리얼 변환을 담당하는 ATMEGA16U2 마이크로컨트롤러와 동일한 기능을 수행한다. 물론 ATMEGA16U2 마이크로컨트롤러는 디버깅 기능은 지원하지 않는다.

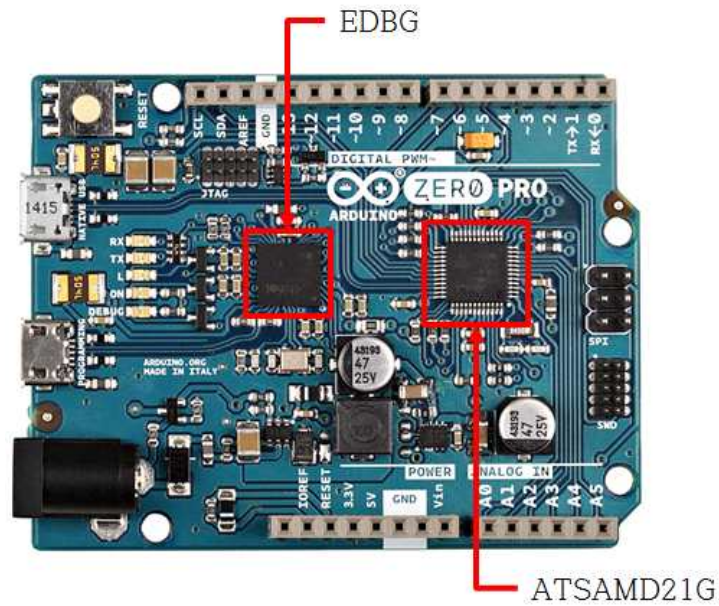


그림 5. 아두이노 제로 프로의 마이크로컨트롤러와 EDBG 칩

아두이노 제로 프로에 블링크 프로그램을 업로드 해보자. 아두이노 제로 프로를 사용하기 위해서는 아두이노 프로그램 중 1.7.3 버전²⁾을 사용하여야 한다. 아두이노 프로그램을 설치하고 ‘도구 → 보드 → Arduino Zero Pro (Programming Port)’를 선택한다.

2) <http://www.arduino.org/downloads>

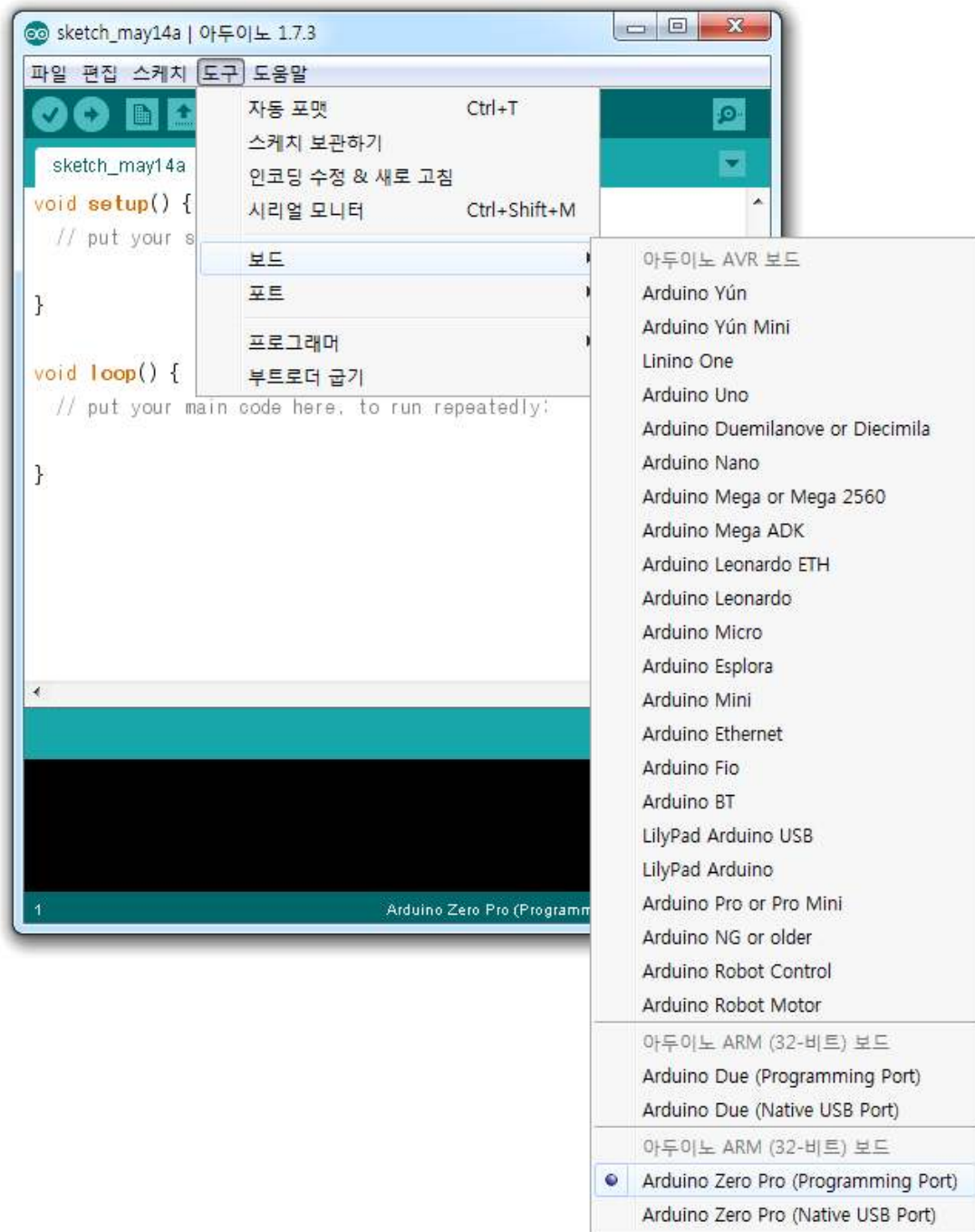


그림 6. 보드 선택 - 프로그래밍 포트

‘도구 → 포트’는 그림 4에서 할당된 포트를 선택하면 된다. ‘파일 → 예제 → 01.Basics → Blink’를 선택하여 블링크 스케치를 열고 업로드 해 보자. 설치 후 처음 스케치를 업로드 한다면 스케치 업로드를 위해 사용되는 프로그램에 대한 보안 경고 창이 나타날 수 있다.

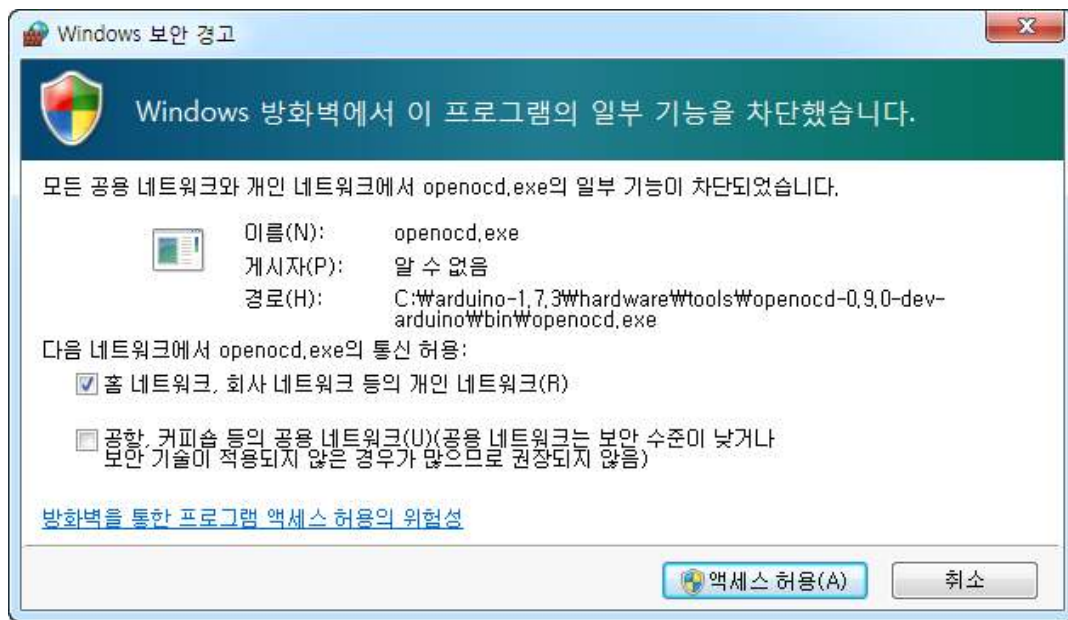


그림 7. 보안 경고 창

‘액세스 허용’을 선택하면 스케치가 업로드 되고 LED가 깜빡이는 것을 확인할 수 있다. 시리얼 모니터로 데이터를 출력해 보자. 아두이노 제로 프로의 경우 프로그래밍 포트와 통신하는 _Serial 클래스의 객체는 Serial5로 정의되어 있으므로 Serial이 아닌 Serial5를 사용하여야 한다. 스케치 1을 업로드 하고 시리얼 모니터로 1초 간격으로 증가하는 카운터 값을 확인해 보자.

스케치 1. 시리얼 모니터 출력 - 프로그래밍 포트

```
int count = 0;

void setup() {
    Serial5.begin(9600);
}

void loop() {
    Serial5.print("Current count is ");
    Serial5.println(count++);

    delay(1000);
}
```

```
}

```

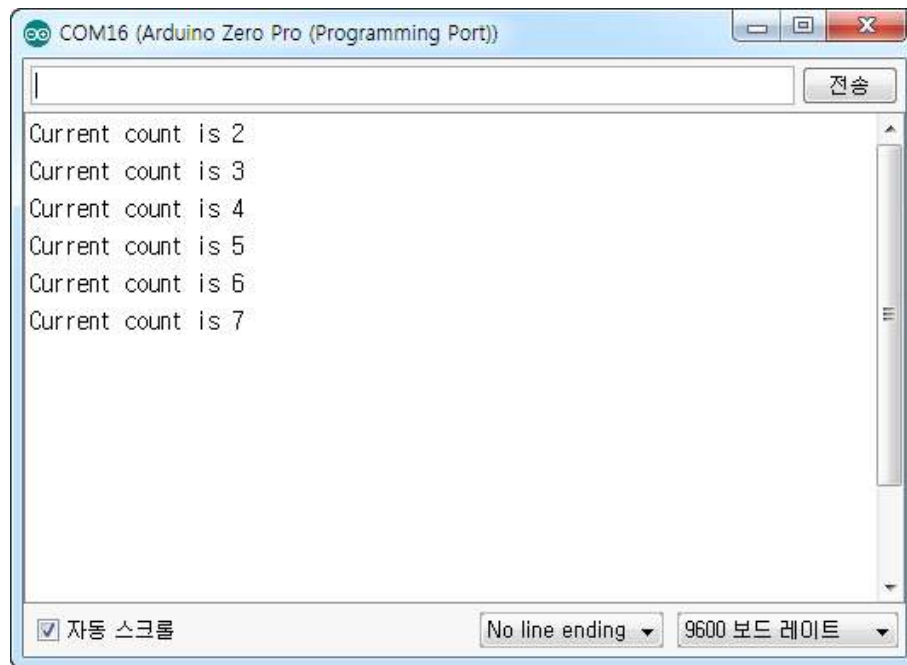


그림 8. 스케치 1 실행 결과

프로그래밍 포트가 EDBG 칩을 통해 ATSAMD21 마이크로컨트롤러와 연결되어 있는 반면, native 포트는 ATSAMD21 마이크로컨트롤러와 직접 연결되어 있다. 프로그래밍 포트 연결을 해제하고 native 포트를 통해 아두이노 제로 프로를 컴퓨터와 연결해 보자. 프로그래밍 포트를 연결했을 때와 마찬가지로 자동으로 드라이버가 설치된다.

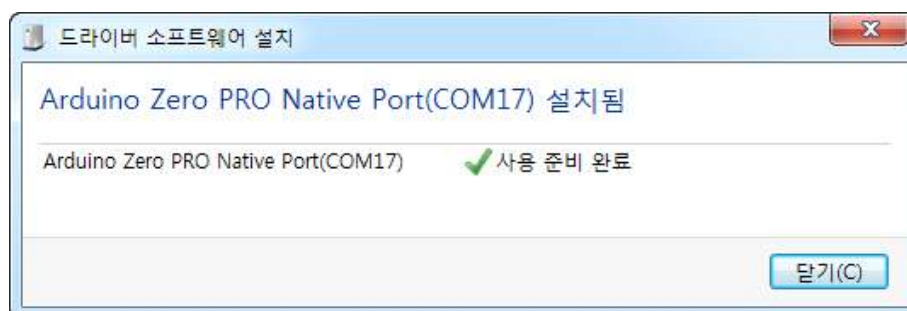


그림 9. 아두이노 제로 프로 native 포트 드라이버 설치

Native 포트를 사용하기 위해서는 그림 6과 달리 '도구 → 보드 → Arduino Zero Pro

(Native USB Port)'를 선택하여야 한다.

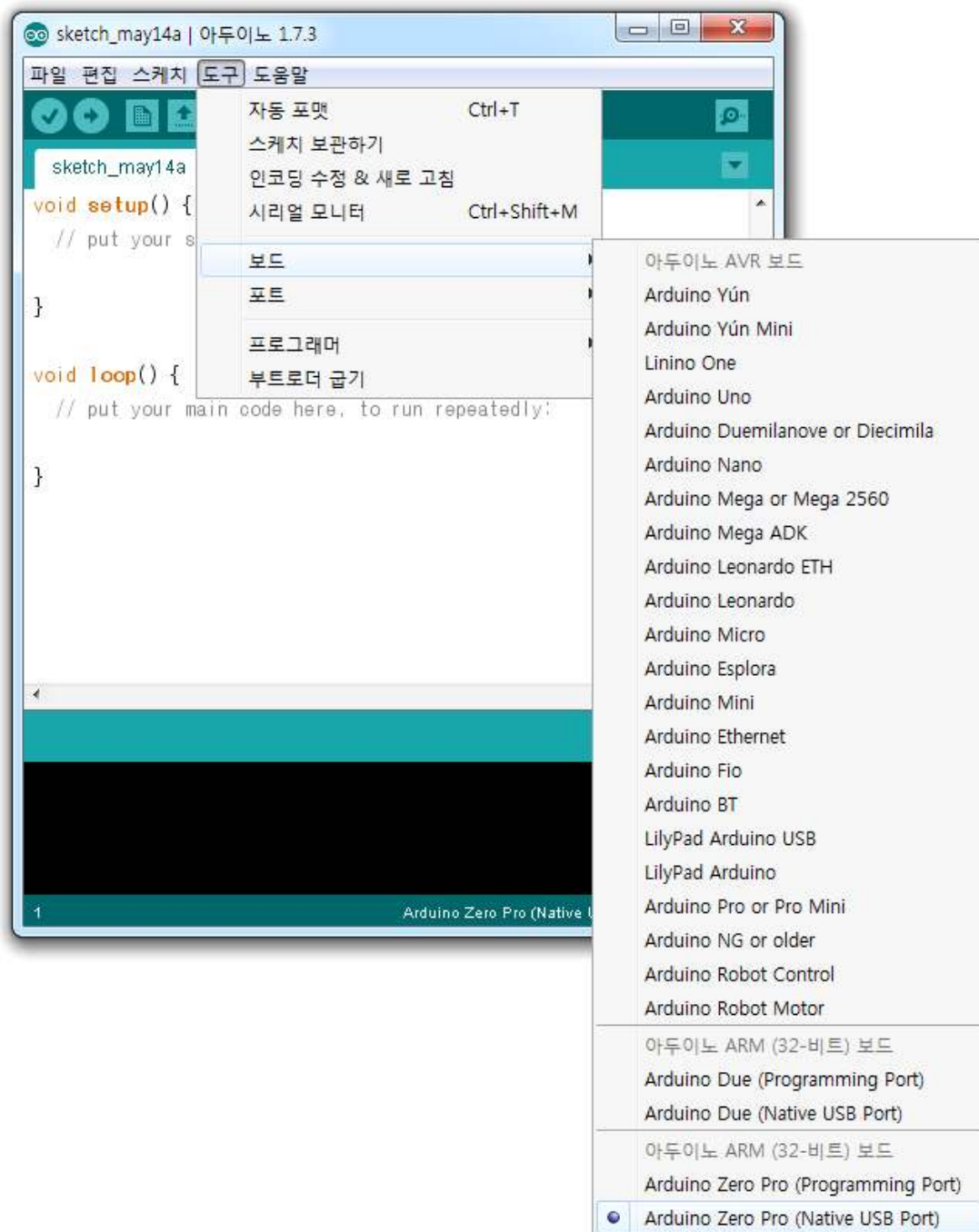


그림 10. 보드 선택 - Native 포트

블링크 예제를 열고 프로그램을 업로드 하면 native 포트를 통해서도 스케치가 업로드 되고 LED가 깜빡이는 것을 확인할 수 있다. 하지만 장치 관리자를 자세히 살펴보면 스케치가 업로

드 되기 시작하는 시점에서 부트로더를 통한 업로드를 위해 포트 번호가 바뀌고, 업로드가 끝난 시점에서 다시 원상태로 되돌아오는 것을 확인할 수 있다.

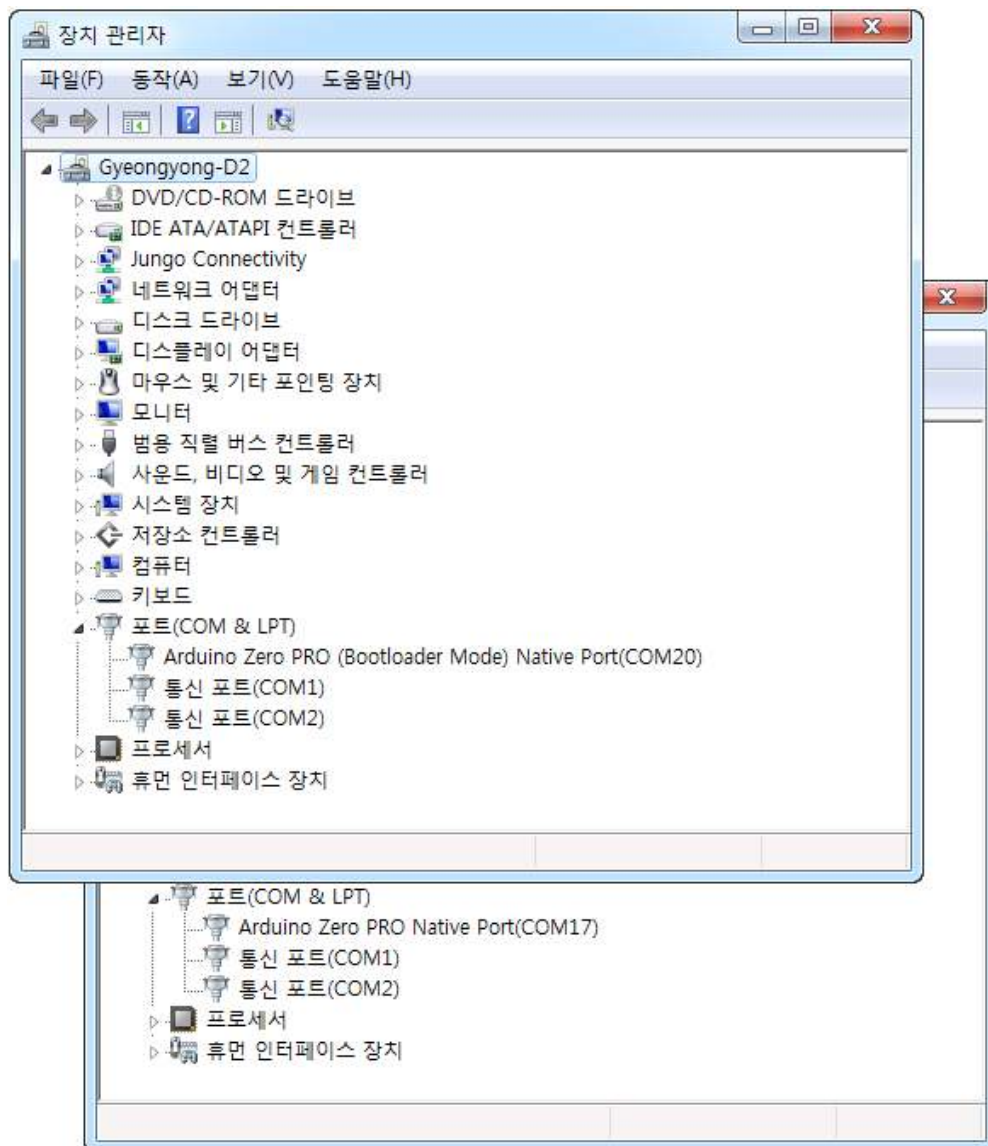


그림 11. native 포트를 통한 업로드 과정에서의 포트 변화

Native 포트 역시 시리얼 모니터로 연결할 수 있다. 하지만 프로그래밍 포트가 하드웨어 시리얼 UART 포트와 연결되어 있는 것과 달리 native 포트는 마이크로컨트롤러의 USB 포트에 연결되어 있으므로 SerialUSB를 사용하여야 한다. 스케치 2를 업로드 하고 시리얼 모니터를 열어 보면 그림 8과 동일한 결과를 확인할 수 있다.

스케치 2. 시리얼 모니터 출력 - native 포트

```
int count = 0;

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

void loop() {
  SerialUSB.print("Current count is ");
  SerialUSB.println(count++);

  delay(1000);
}
```

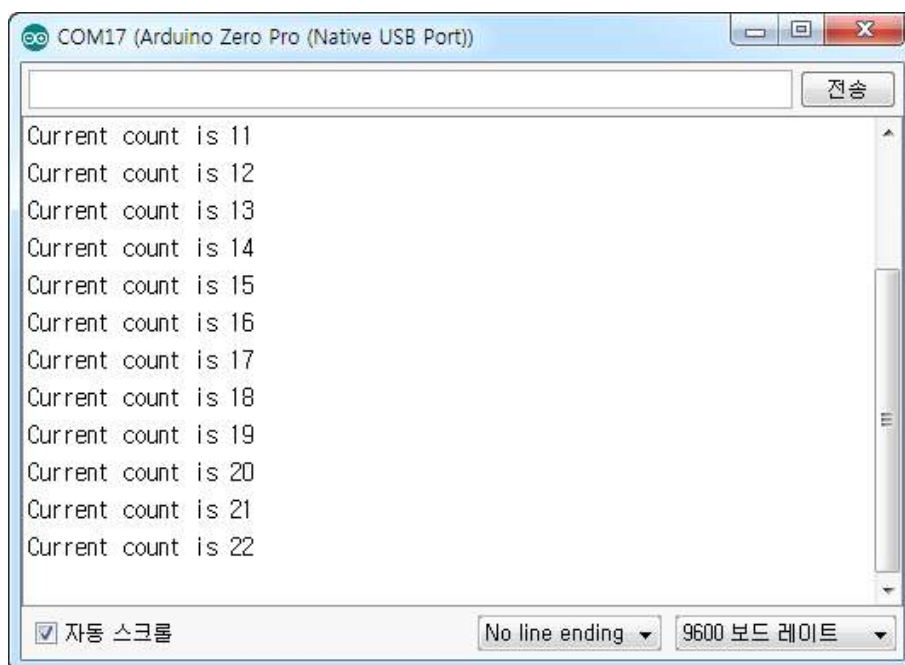


그림 12. 스케치 2 실행 결과

스케치 업로드를 위해 프로그래밍 포트와 native 포트를 모두 사용할 수 있음을 살펴보았다. 하지만 아두이노 제로 프로의 ATSAM21 마이크로컨트롤러는 AVR 계열의 마이크로컨트롤러와 업로드 과정에서 약간의 차이가 있다. 스케치 업로드는 ATSAM21 마이크로컨트롤러의 플래시 메모리 내용이 모두 지워진 경우에만 가능하므로 업로드 이전에 플래시 메모리를 지워

주는 작업이 필요하다. 프로그래밍 포트를 통한 업로드의 경우 플래시 메모리를 지우는 작업이 마이크로컨트롤러에 의해 이루어지며 'hard erase'라고 불린다. 반면 native 포트를 통한 업로드의 경우 플래시 메모리를 지우는 작업은 부트로더 즉, 소프트웨어에 의해 이루어지므로 'soft erase'라고 불린다. 하드웨어에 의한 지우기 작업은 부트로더가 잘못된 경우에도 가능하지만 소프트웨어에 의한 지우기 작업은 불가능하므로 아두이노 제로 프로의 스케치 업로드는 프로그래밍 포트를 사용하는 것이 추천되고 있다. 하지만 native 포트는 USB 호스트로 동작할 수 있으므로 마우스나 키보드와 같은 장치를 연결하여 USB를 통해 통신을 수행하는 것이 가능하다.