

Chapter 3. Mega2560 모듈 - AM-2560USB

학습 보드에는 ATmega2560 칩을 사용하여 만들어진 AM-2560USB 모듈이 포함되어 있다. AM-2560USB 모듈은 학습 보드에 탈부착이 가능하며, AM-2560USB 모듈만으로도 아두이노 메가2560으로 동작하기 위한 모든 기능들이 포함되어 있다.

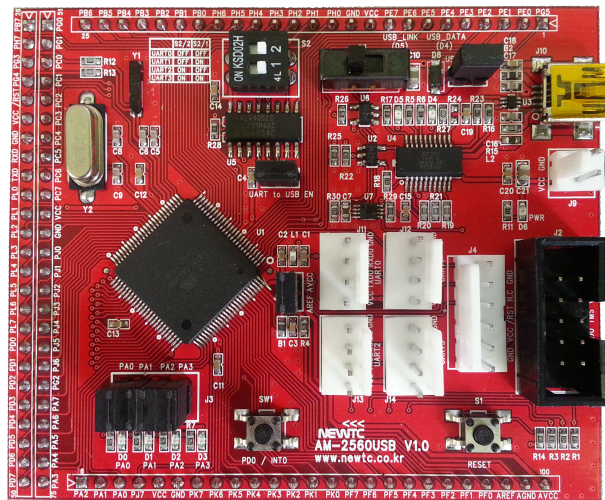


그림 1. AM-2560USB 모듈

이 장에서의 설명은 AM-2560USB 모듈을 분리하여 사용하는 것으로 설명하고 있지만 학습보드에 연결한 상태에서도 모든 내용을 테스트해볼 수 있다. 다만 부트로더를 굽는 경우에는 한 가지 주의해야 할 점이 있다. 학습보드에는 SPI 통신을 위해 아두이노 우노 모듈과 아두이노 메가2560 모듈이 연결되어 있으며 EEPROM이 SPI 통신을 위해 연결되어 있다. 부트로더를 굽기 위한 ISP(In System Programming) 장치 역시 SPI 통신에서와 동일한 핀을 사용하므로 AM-2560USB 모듈이 학습보드에 연결되어 있는 경우 부트로더를 굽기 위해서는 학습보드의 점퍼를 제거하여야 한다. 부트로더를 구운 후에는 점퍼를 다시 연결해야 한다는 점도 잊지 말아야 한다.

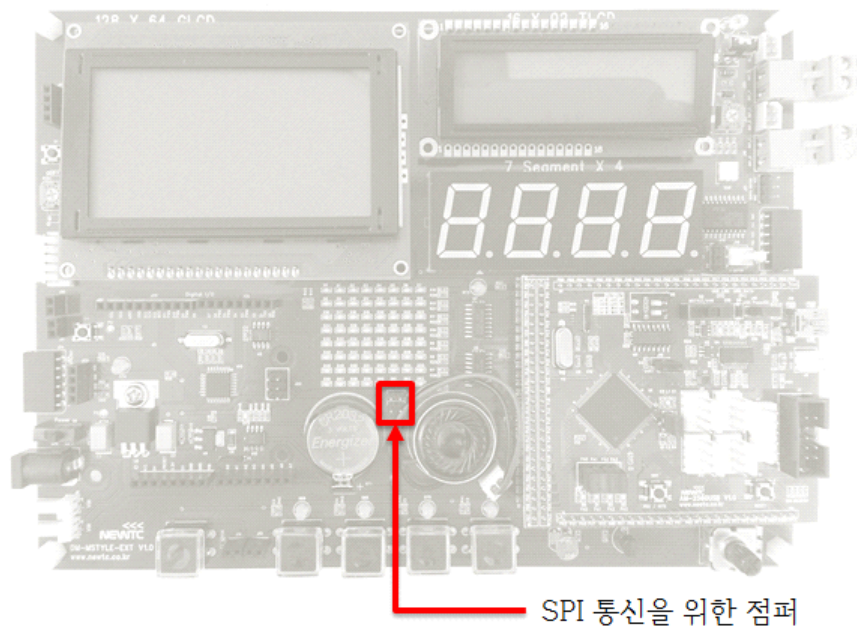


그림 2. 부트로더를 굽기 위한 점퍼 제거

1. 부트로더 굽기

먼저 AM-2560USB 모듈에 부트로더를 구워야 한다. 부트로더를 굽기 위해서는 USB 방식의 ISP 장치인 AD-USBISP를 사용할 수 있다.

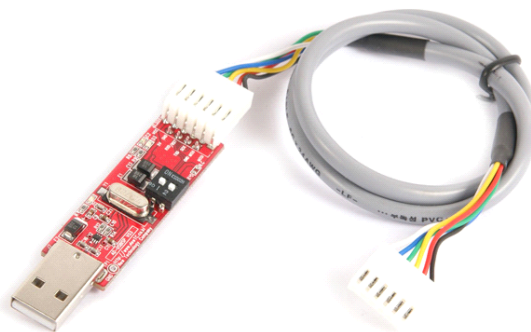
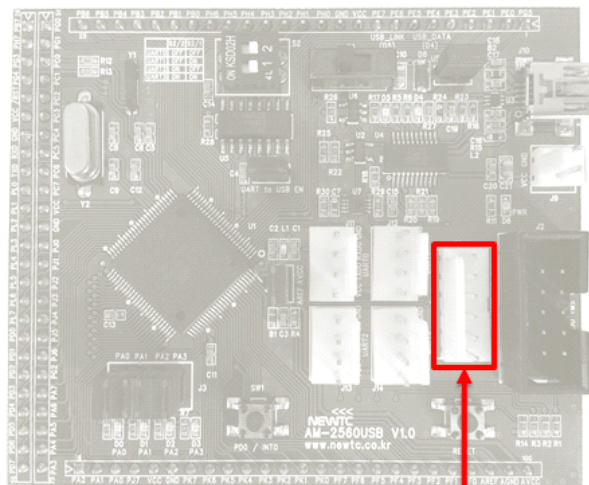


그림 3. AD-USBISP¹⁾

ISP 장치를 AM-2560USB 모듈의 ISP 커넥터에 연결하고 컴퓨터와 연결해 보자.

1) 이미지 출처: <http://www.newtc.co.kr>



ISP 장치 연결 커넥터

그림 4. 부트로더를 굽기 위한 ISP 장치 연결 커넥터

AD-USBIISP 장치를 컴퓨터에 연결하면 AD-USBIISP 장치의 드라이버가 설치되고 장치관리자의 포트 부분에 시리얼 포트가 생성된다. 여기서는 COM9가 할당된 것으로 가정한다.

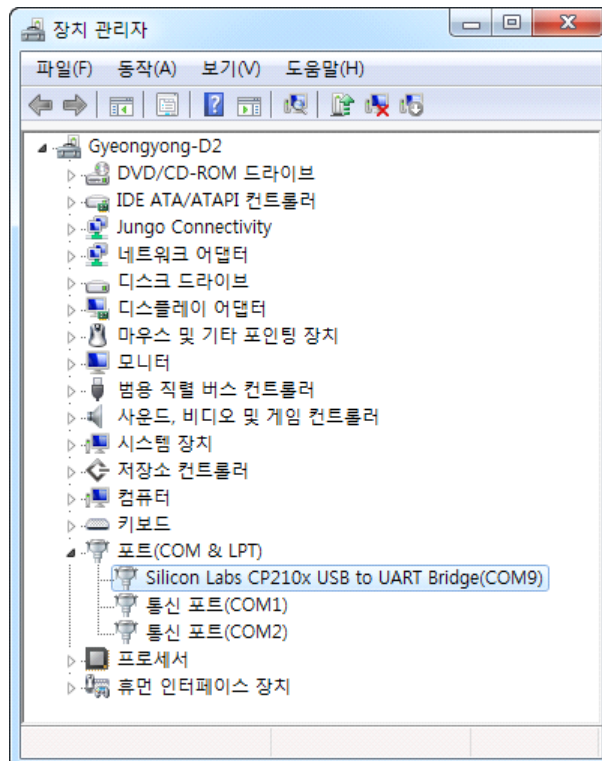


그림 5. ISP 장치 연결 후 포트

AM-2560USB 모듈에 전원이 연결되어 있지 않으므로 ISP 장치에서 USB 전원을 공급하도록 설정하는 것이 편리하다. ISP 장치의 DIP 스위치 중에서 1번 스위치를 ON시키면 USB 전원이 AM-2560USB 모듈에 공급된다.



그림 6. USB 전원 공급을 위한 ISP 장치의 DIP 스위치

부트로더를 굽기 전에 아두이노 프로그램에서 한 가지 설정이 더 필요하다. 아두이노에서 기본적으로 사용할 수 있는 ISP 장치 중에는 AD-USBISP가 포함되어 있지 않다. 따라서 먼저 AD-USBISP를 아두이노에 등록하여야 한다. 아두이노 프로그램이 설치된 디렉터리 아래 'hardware\arduino\avr' 디렉터리에 'programmers.txt' 파일의 가장 아래쪽에 AD-USBISP 장치의 정보를 추가한다.

```
AD_USBISP.name=AD-USBISP
AD_USBISP.communication=serial
AD_USBISP.protocol=stk500v2
AD_USBISP.program.protocol=stk500v2
AD_USBISP.program.tool=avrdude
AD_USBISP.program.extra_params=-P{serial.port}
```

- 아두이노 프로그램을 실행시키고 '도구 → 프로그래머 → AD-USBISP' 항목이 나타나면

ISP 장치가 정상적으로 등록된 것이다. AD-USBISP를 선택한다.

- ‘도구 → 보드’ 메뉴에서는 사용할 아두이노 보드를 선택한다. AM-2560USB 모듈은 아두이노 메가2560 호환 모듈이므로 ‘Arduino/Genuino Mega or Mega 2560’을 선택한다.
- ‘도구 → 프로세서’ 메뉴에서 ‘ATmega2560 (Mega 2560)’을 선택한다.
- ‘도구 → 포트’ 메뉴에서는 ISP 장치에 할당된 시리얼 포트를 선택한다. AD-USBISP에 COM9가 할당된 것으로 가정하였으므로 COM9를 선택한다.

이제 모든 준비는 끝났다. ‘도구 → 부트로더 굽기’ 메뉴를 선택하여 부트로더를 구워보자. 약간의 시간이 흐르면 부트로더 굽기가 성공적으로 끝났다는 메시지가 나타날 것이다.

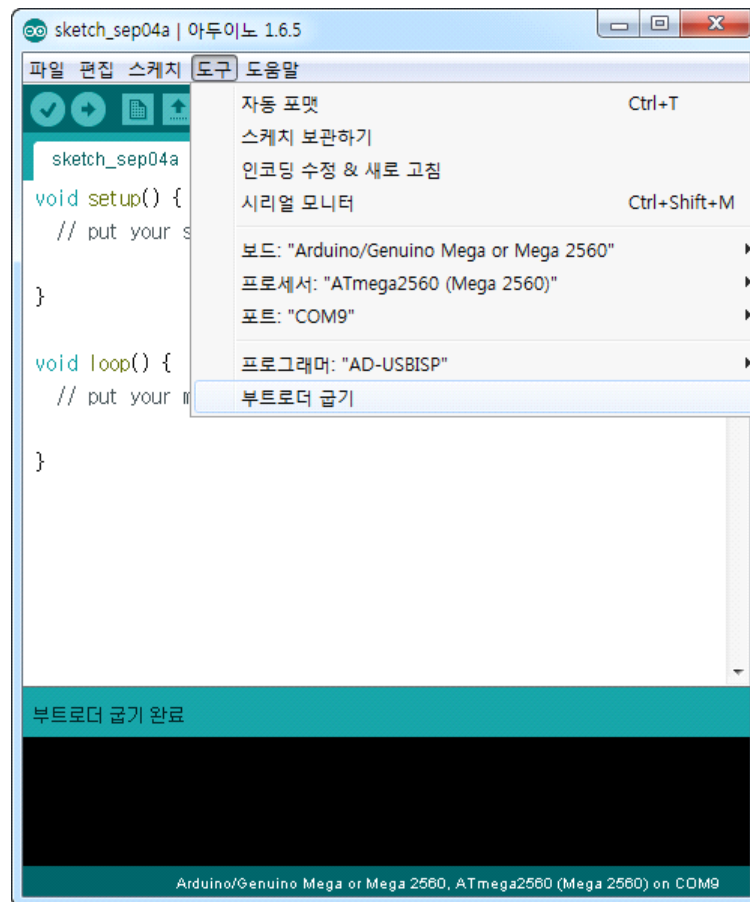


그림 7. 부트로더 굽기

2. 블링크 (Blink)

부트로더가 구워졌으니 아두이노의 기본 프로그램인 블링크 스케치를 업로드 해보자. ISP 장

치로도 스케치 업로드가 가능하지만 아두이노에서는 부트로더를 통해 시리얼 방식으로 스케치를 업로드 하는 것이 일반적이다. AM-2560USB 모듈에는 USB-Serial 변환장치가 포함되어 있으므로 별도의 장치 없이 컴퓨터와의 USB 연결만으로 스케치를 업로드 할 수 있다. AM-2560USB 모듈을 컴퓨터와 USB 케이블로 연결하면 장치관리자에는 ISP 장치를 연결한 경우와 유사하게 USB-Serial 변환장치인 FTDI 칩에 해당하는 시리얼 포트가 생성된다. 여기서 COM10에 할당된 것으로 가정한다.

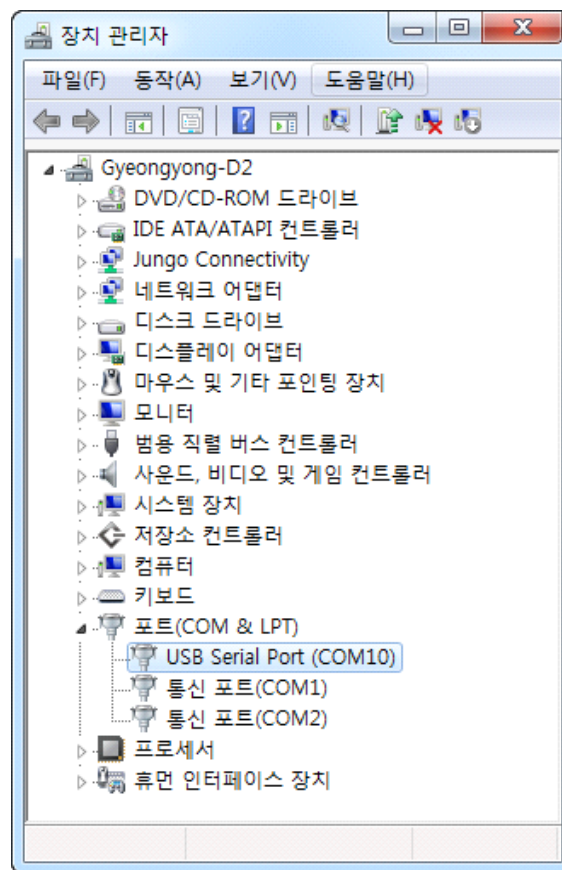


그림 8. AM-2560USB 모듈을 USB로 연결한 경우 포트

그림 9는 USB로 AM-2560USB 모듈을 연결할 때 사용되는 커넥터와 설정을 위한 스위치를 나타낸다.

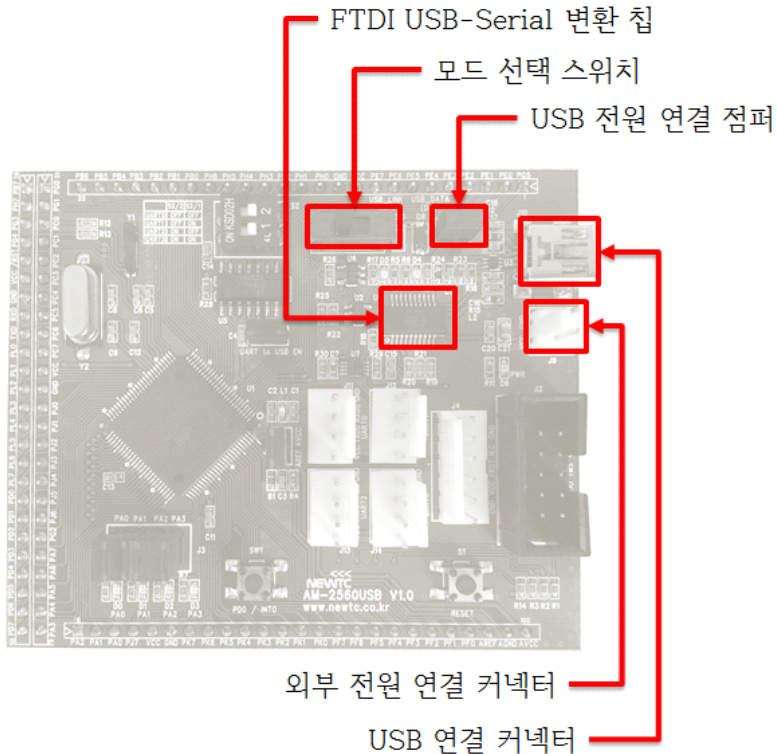


그림 9. 시리얼 방식의 스케치 업로드를 위한 커넥터 및 설정 장치

ISP 장치에는 USB를 통한 전원 공급 여부를 결정하기 위한 DIP 스위치가 있다. USB로 AM-2560USB 모듈을 컴퓨터와 연결하는 경우에는 그림 9에서 ‘USB 전원 연결 점퍼’로 USB 전원 사용 여부를 선택할 수 있다. 점퍼가 연결된 상태에서는 USB 전원이 AM-2560USB 모듈에 공급된다. 별도의 전원을 사용하고자 한다면 USB 전원 연결 점퍼를 제거하고 ‘외부 전원 연결 커넥터’에 5V의 전원을 연결해주면 된다.

‘모드 선택 스위치’는 AM-2560USB 모듈의 프로그래밍 방식을 선택하기 위해 사용하며 ‘MSTYLE’과 ‘AVR’ 중 선택할 수 있다. 아두이노로 사용하고자 한다면 모드 선택 스위치를 MSTYLE에, AVR로 사용하고자 한다면 모드 선택 스위치를 AVR에 연결하여야 한다. 부트로더를 굽는 경우에는 두 가지 모드 모두에서 가능하지만 스케치 업로드를 위해서는 MSTYLE에 위치시켜야 한다.

‘파일 → 예제 → 01.Basics → Blink’를 선택하여 블링크 스케치를 열어보자. 부트로더를 굽기 위한 설정에서 한 가지 변경해야 하는 점은 포트 설정 부분이다. 부트로더를 굽기 위해서는 ISP 장치에 할당된 포트를 선택하였지만 프로그램을 업로드하기 위해서는 USB-Serial 변환 장치에 할당된 포트를 선택해야 한다. ‘도구 → 포트’ 메뉴에서 USB-Serial 변환 장치에 할당된 COM10을 선택하고 업로드 버튼을 눌러 스케치를 업로드 해보자. 업로드에 성공하더

라도 AM-2560USB 모듈에는 아무런 표시가 나지 않을 것이다. 블링크 스케치는 디지털 13번 핀을 사용하지만 AM-2560USB 모듈에는 디지털 13번 핀에 연결된 LED가 포함되어 있지 않다. 대신 디지털 22번부터 25번까지 (PA0 ~ PA3) 4개의 LED가 포함되어 있다. 4개의 LED를 점멸하기 위해 스케치 1을 MyBlink라는 이름으로 스케치북에 저장한 뒤 업로드 해보자.

스케치 1. MyBlink

```
int pins[] = {22, 23, 24, 25};           // LED 연결 핀

void setup(){
  for(int i = 0; i < 4; i++){
    pinMode(pins[i], OUTPUT);           // LED 연결 핀을 출력으로 설정
    digitalWrite(pins[i], HIGH);        // 꺼진 상태에서 시작
  }
}

void loop(){
  for(int i = 0; i < 4; i++){
    digitalWrite(pins[i], LOW);          // LED 켜기
    delay(500);                          // 0.5초 대기

    digitalWrite(pins[i], HIGH);         // LED 끄기
    delay(500);                          // 0.5초 대기
  }
}
```

한 가지 주의할 점은 LED들이 연결된 핀에 LOW를 출력해야 LED가 켜지도록 연결되어 있다는 점이다. AM-2560USB 모듈에는 LED 이외에도 각 LED의 연결을 제어할 수 있는 점퍼가 포함되어 있으므로 다른 장치를 연결하여 사용하고자 하는 경우에는 LED 연결을 해제할 수 있다.

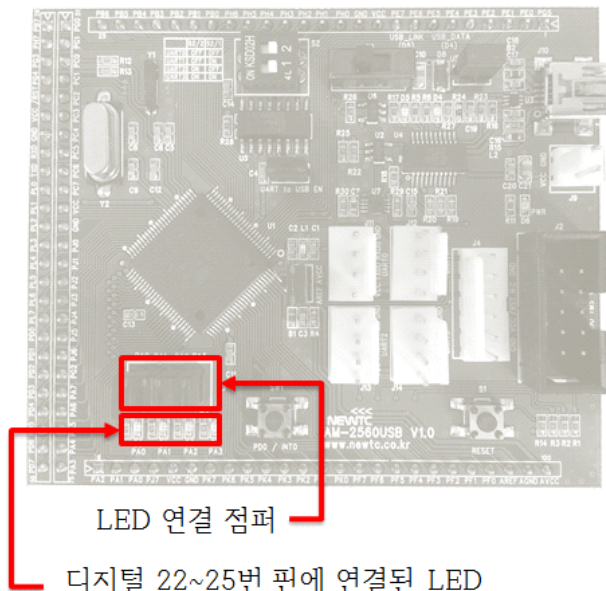


그림 10. 디지털 22~25번 연결 LED와 점퍼

3. UART 시리얼 통신

스케치 업로드를 위해 사용되는 USB-Serial 변환 장치는 스케치 업로드를 위해 사용하지 않는 경우에는 컴퓨터와의 UART 시리얼 통신을 위해 사용할 수 있다. 스케치 2는 시리얼 모니터로 1초에 1씩 증가하는 카운터 값을 출력하는 스케치이다. 스케치를 업로드 하고 시리얼 모니터를 열어 결과를 확인해 보자.

스케치 2. 시리얼 모니터 출력

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

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

void loop(){
  Serial.println(count++);                     // 시리얼 모니터로 출력
  delay(1000);                                // 1초 대기
}
```

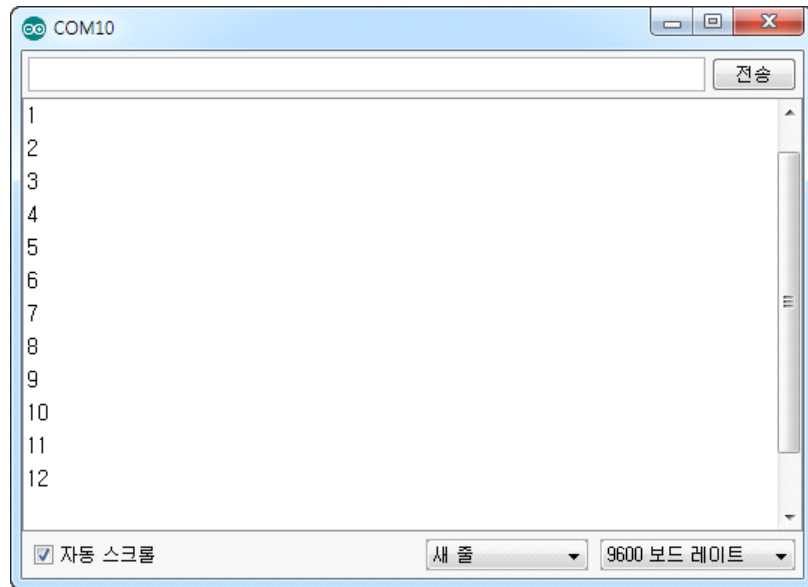


그림 11. 스케치 2 실행 결과

아두이노 우노에 사용된 마이크로컨트롤러인 ATmega328은 하나의 하드웨어 시리얼 포트만 제공하지만 아두이노 메가2560에 사용된 마이크로컨트롤러인 ATmega2560은 4개의 하드웨어 시리얼 포트(UART0 ~ UART3)를 제공하고 있으며 각각 Serial, Serial1, Serial2, Serial3 객체를 통해 UART 시리얼 통신을 수행할 수 있다. 이 중 스케치 업로드에 사용되는 포트는 디폴트로 Serial(UART0)이 연결되어 있지만 AM-2560USB 모듈에는 하드웨어 시리얼 포트를 선택할 수 있는 방법을 제공하고 있다.

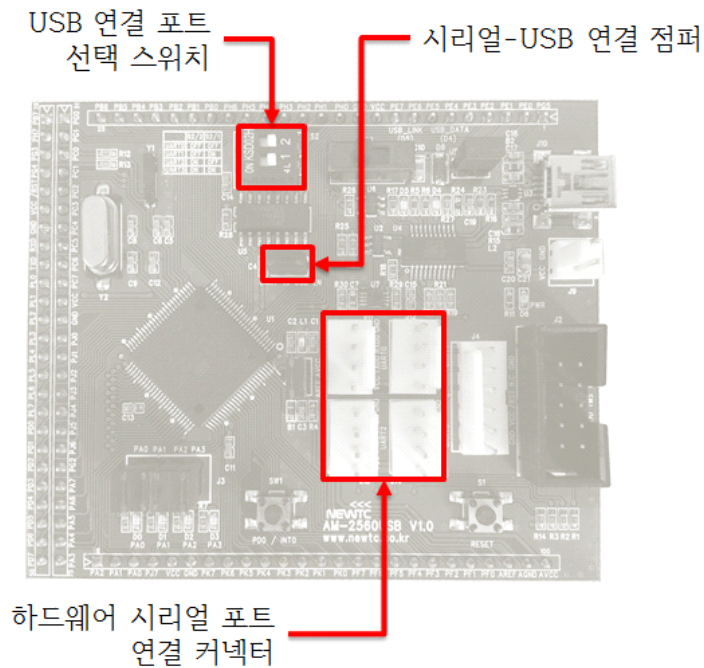


그림 12. 시리얼 포트 선택

‘시리얼-USB 연결 점퍼’는 ATmega2560의 하드웨어 시리얼 포트 중 하나를 USB-Serial 변환 장치로 연결하기 위해 사용된다. 따라서 점퍼가 연결되어 있지 않으면 USB를 통해 스케치를 업로드 할 수 없으며 시리얼 모니터를 통해 결과를 확인할 수 없다. USB-Serial 연결 점퍼가 연결된 경우 USB-Serial 변환 장치로 연결할 하드웨어 시리얼 포트는 ‘USB 연결 포트 선택 스위치’에 의해 결정된다. 디폴트로 0번 하드웨어 시리얼 포트(Serial)가 연결되도록 설정되어 있다.

하드웨어 시리얼 포트가 USB-Serial 변환 장치를 통해 연결되어 있지 않은 경우에도 ‘하드웨어 시리얼 포트 연결 커넥터’를 통해 ATmega2560의 하드웨어 시리얼 포트를 사용할 수 있다. 하드웨어 시리얼 0번은 디폴트로 AM-2560USB 모듈의 USB-Serial 변환 장치로 연결되어 있어 별도의 장치 없이 컴퓨터와 연결할 수 있지만, 다른 포트들을 컴퓨터의 USB 포트와 연결하기 위해서는 USB-Serial 변환 장치에 해당하는 별도의 변환 장치가 필요하다. 별도의 USB-Serial 변환 장치로는 DM-USB2SERIAL을 사용할 수 있다.

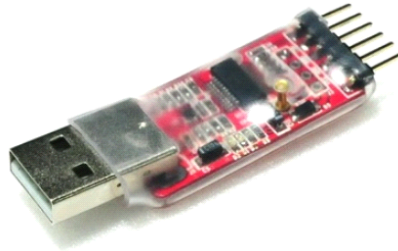


그림 13. DM-USB2SERIAL

DM-USB2SERIAL은 6개의 핀을 가지고 있으며 스케치 업로드에서는 이 중 5개 핀(VCC, GND, RX, TX, DTR)을 사용하고, 시리얼 통신을 위해서는 4개 핀(VCC, GND, RX, TX)만을 사용한다. DM-USB2SERIAL를 하드웨어 시리얼 포트 연결 커넥터 중 UART1(Serial1)에 연결해 보자. 이 때 DM-USB2SERIAL의 RX와 TX는 UART1 커넥터의 TX와 RX로 교차하여 연결하여야 한다는 점에 유의하여야 한다. DM-USB2SERIAL을 컴퓨터와 연결하면 AM-2560USB 모듈은 컴퓨터와 2개의 연결이 설정된다. 장치관리자를 확인해 보면 AM-2560USB 모듈과 DM-USB2SERIAL의 USB-Serial 변환 장치를 위해 별도의 COM 포트가 할당되어 있는 것을 확인할 수 있다.

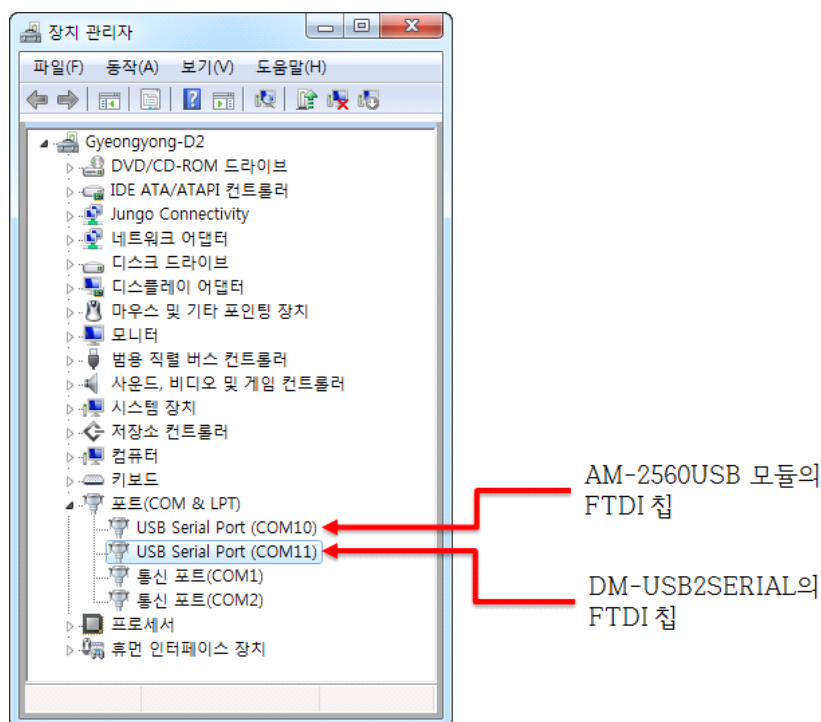


그림 14. 2개의 하드웨어 시리얼 포트 연결

스케치 2의 Serial을 스케치 3과 같이 Serial1으로 수정하자. 이제 출력은 Serial이 아닌 Serial1으로 이루어진다.

스케치 3. 하드웨어 시리얼 1번으로 출력

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

void setup(){
  Serial1.begin(9600);                         // UART 시리얼 통신 초기화
}

void loop(){
  Serial1.println(count++);                   // 시리얼 모니터로 출력
  delay(1000);                                // 1초 대기
}
```

스케치를 업로드하고 시리얼 모니터를 실행시켜 보아도 아무런 메시지가 출력되지 않을 것이다. 시리얼 모니터는 현재 '도구 → 포트'에 선택된 포트와 연결되며 스케치가 업로드 되는 포트에 해당한다. 따라서 Serial1으로 출력되는 내용을 확인하기 위해서는 '도구 → 포트'에서 DM-USB2SERIAL이 연결된 COM11을 선택한 후 시리얼 모니터를 실행시켜야 한다.

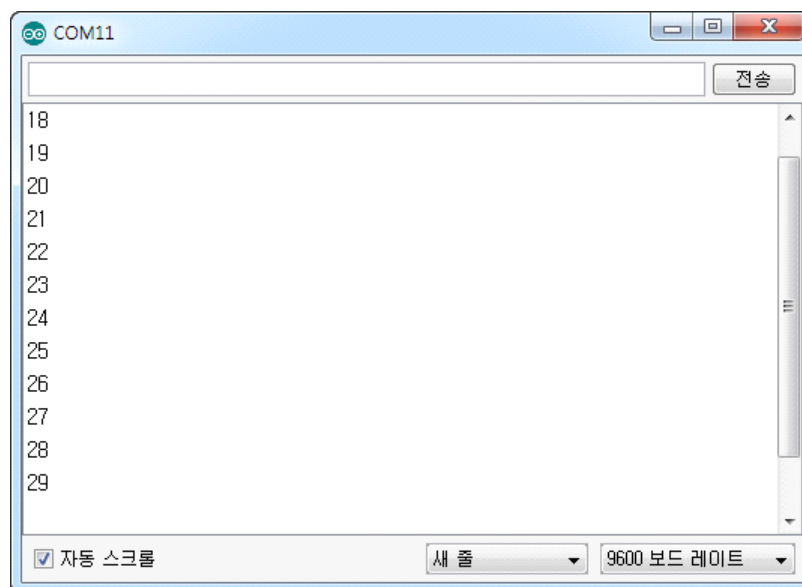


그림 15. 스케치 3 실행 결과

4. 기타

앞에서 설명한 내용 이외에도 AM-2560USB 모듈에는 몇 가지 기능이 더 숨겨져 있다.

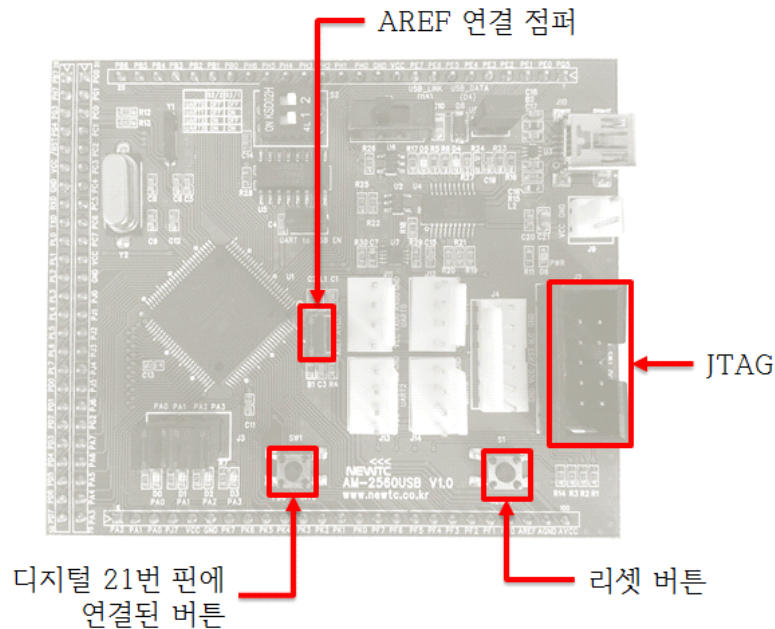


그림 16. 기타 커넥터 및 버튼

‘AREF 연결 점퍼’는 ADC 변환에서 5V 이외의 기준 전압을 사용할 수 있도록 해준다. 아두이노의 아날로그 디지털 변환에서는 디폴트로 AVCC를 기준으로 하므로 연결하고 사용하면 된다. ‘JTAG’은 디버깅(debugging)을 위한 용도로 흔히 사용되지만 아두이노 환경에서는 디버깅이 지원되지 않으므로 사용하지 않는다. 리셋 버튼은 ATmega2560 마이크로컨트롤러의 리셋 용도로 사용되는 버튼이며, 다른 하나의 버튼은 디지털 21번 핀(PD0)에 연결된 버튼이다. 스케치 4는 버튼을 눌렀을 때 디지털 22~25번 핀에 연결된 4개의 LED가 켜지는 스케치의 예이다. 스케치 4에서 주의할 점은 버튼이 내부 풀업 저항을 사용하도록 연결되어 있다는 점이다. 따라서 풀업저항을 사용하지 않고 버튼 상태를 읽는 경우에는 항상 LOW 값을 반환한다.

스케치 4. 버튼 테스트

```
int pin_button = 21;
int pin_LED[] = {22, 23, 24, 25};

void setup(){
  pinMode(pin_button, INPUT_PULLUP); // 내부 풀업저항 사용
  for(int i = 0; i < 4; i++){
    pinMode(pin_LED[i], OUTPUT);
    digitalWrite(pin_LED[i], HIGH); // 꺼진 상태로 시작
  }
}

void loop(){
  boolean state = digitalRead(pin_button); // 버튼 상태 읽기

  for(int i = 0; i < 4; i++) // LED 점멸
    digitalWrite(pin_LED[i], state);
}
```