

Chapter 4. 우노 호환 모듈

학습보드에는 메가2560 호환 이외에도 우노 호환 모듈이 포함되어 있다. 메가2560 호환 모듈은 탈착이 가능한 반면 우노 호환 모듈은 보드에 포함된 형태로 차이가 있다. 학습보드는 메가2560을 기본으로 하고 아두이노 사이의 통신을 위해 우노 호환 모듈이 사용된다.

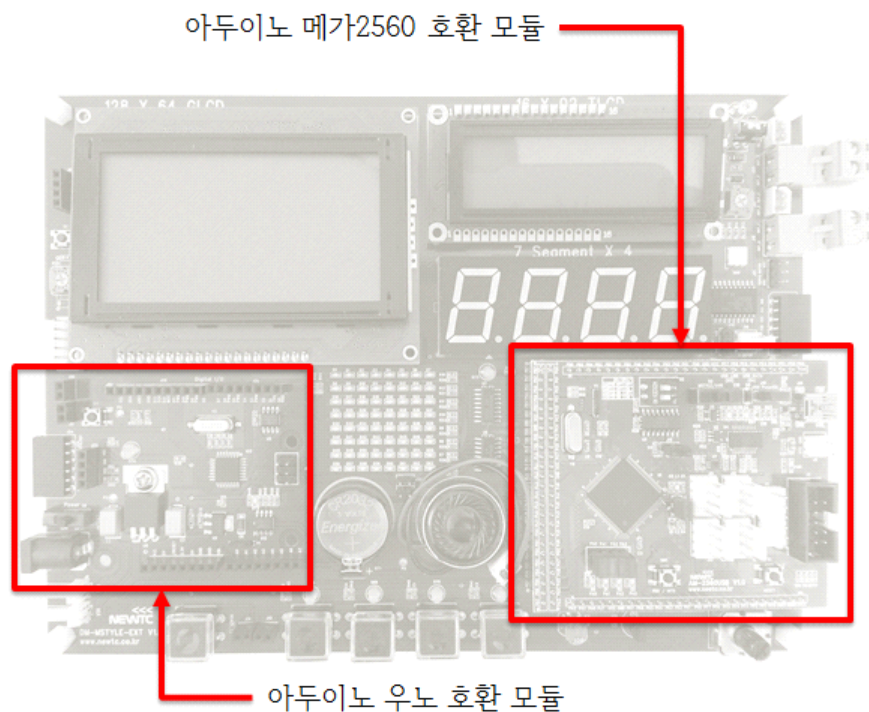


그림 1. 학습 보드의 우노 호환 모듈 부분

1. 부트로더 굽기

메가2560 모듈과 마찬가지로 우노 호환 모듈 역시 먼저 부트로더를 구워야 아두이노 환경에서 사용할 수 있다. 부트로더를 굽기 위해서는 AM-2560USB 모듈에 부트로더를 굽기 위해 사용한 AD-USBISP를 사용한다. ISP 장치를 우노 호환 모듈을 위한 ISP 커넥터에 연결하자. 학습보드에는 2×3 커넥터와 1×6 커넥터가 준비되어 있으며 AD-USBISP는 1×6 커넥터에 연결한다.

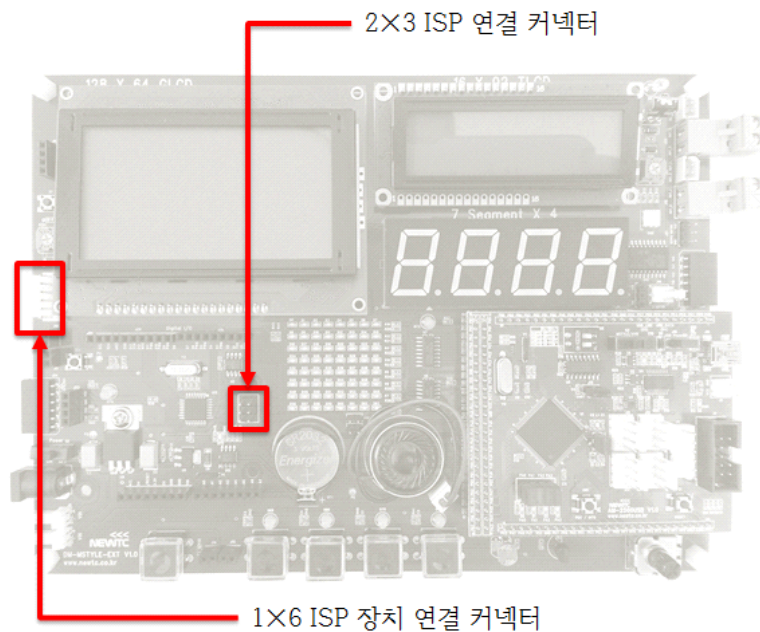


그림 2. 부트로더를 굽기 위한 ISP 장치 연결 커넥터

메가2560 호환 모듈에 부트로더를 굽기 위해 사용한 장치와 동일한 장치를 사용하므로 컴퓨터에서는 COM9가 할당된다. 아두이노 프로그램을 실행시키고 다음 옵션을 확인해 보자.

	우노 호환 모듈	메가2560 호환 모듈
도구 → 보드	Arduino/Genuino Uno	Arduino/Genuino Mega or Mega 2560
도구 → 프로세서	-	ATmega2560 (Mega 2560)
도구 → 포트	COM9	
도구 → 프로그래머	AD-USBISP	

표 1. 부트로더를 굽기 위한 옵션

옵션이 설정되었으면 ‘도구 → 부트로더 굽기’ 항목을 선택하여 부트로더를 굽는다. 한 가지 주의할 점은 메가2560 호환 모듈에서와 마찬가지로 SPI 점퍼를 제거하여 한다는 점이다.

2. LED와 버튼

부트로더를 구웠으므로 스케치는 시리얼 방식으로 업로드 한다. 우노 호환 모듈에는 메가 호

환 모듈과 달리 USB-시리얼 변환 장치가 포함되어 있지 않으므로 스케치 업로드를 위해 별도의 장치를 사용하여야 한다. USB-시리얼 변환 장치는 메가2560 호환 모듈에서 UART1과 컴퓨터를 연결하기 위해 사용하였던 DM-USB2SERIAL을 사용하면 된다.

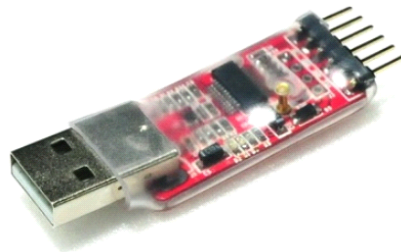


그림 3. DM-USB2SERIAL, USB-시리얼 변환 장치

학습보드에는 우노 호환 모듈을 위한 USB-시리얼 변환 장치 연결 커넥터가 준비되어 있다.



그림 4. 스케치 업로드를 위한 USB-시리얼 변환 장치 연결 커넥터

‘파일 → 예제 → 01.Basics → Blink’ 메뉴를 선택하자. 블링크 스케치를 업로드하기 전에 한 가지 주의할 점이 있다. 블링크 스케치는 디지털 13번 핀에 연결된 LED를 사용한다. 하지만 디지털 13번 핀은 SPI 통신을 위해 사용되는 핀이므로 우노 호환 모듈의 경우 13번 핀이 아닌 9번 핀에 LED가 연결되어 있다. 따라서 블링크 스케치의 핀 번호를 9번으로 변경하고 스

케치를 업로드 하여야 한다.

스케치 1. 9번 핀 LED 블링크

```
int LED_pin = 9;

void setup() {
  pinMode(LED_pin , OUTPUT);
}

void loop() {
  digitalWrite(LED_pin , HIGH);
  delay(1000);
  digitalWrite(LED_pin , LOW);
  delay(1000);
}
```

우노 호환 모듈에는 디지털 2번 핀에 풀다운 저항을 통해 연결된 버튼도 포함되어 있다.

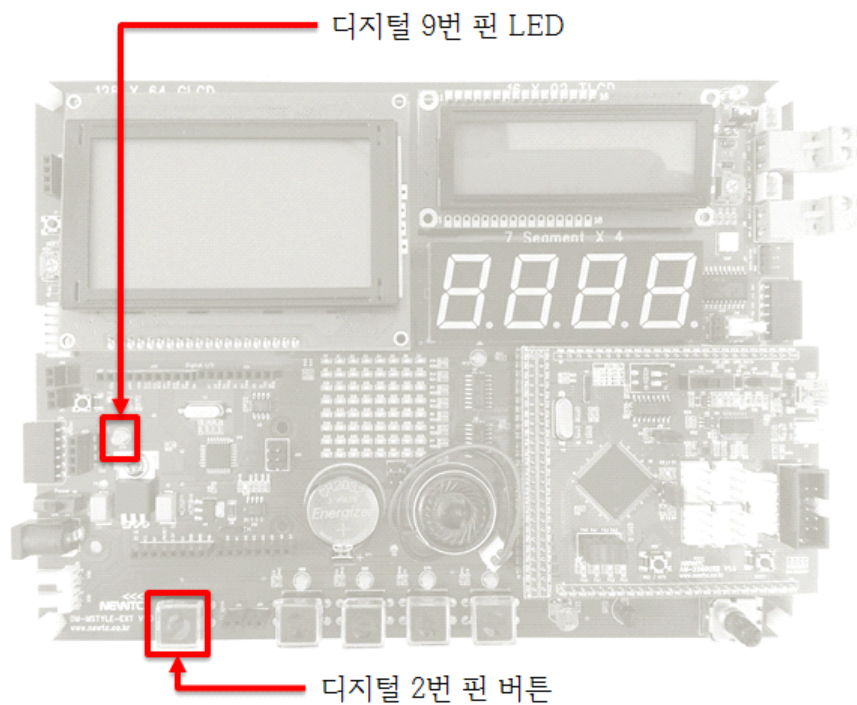


그림 5. 학습보드에 포함된 우노 호환 모듈을 위한 LED와 버튼

스케치 2는 디지털 2번 핀의 버튼 상태를 디지털 9번 핀의 LED에 표시하는 스케치의 예이다. 버튼을 누르면 LED에 불이 켜지고, 누르지 않으면 LED에 불이 켜지지 않는 것을 확인할 수 있다.

스케치 2. LED와 버튼

```
int button = 2;
int LED = 9;

void setup() {
  pinMode(button, INPUT);
  pinMode(LED, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  boolean state = digitalRead(button);    // 버튼 상태 검사

  // 버튼 상태에 따라 시리얼 모니터와 LED에 상태 표시
  Serial.println(state ? "Button is ON." : "Button is OFF.");
  digitalWrite(LED, state);

  delay(1000);
}
```