

Chapter 152. 디지털 출력

1. STM32F030R8T6 마이크로컨트롤러

NUCLEO-F030R8 보드는 64핀의 STM32F030R8T6 마이크로컨트롤러 사용하고 있다. STM32F030R8T6 마이크로컨트롤러 주요 사양은 표 1과 같다.

주변 장치		설명	비고
플래시 메모리		64 KByte	
SRAM		8 KByte	
타이머		6개	16 bit
통신 인터페이스	SPI	2개	
	I2C	2개	
	USART	2개	
ADC		18 channel	12 bit
GPIO		55개	
동작 전압		2.4V ~ 3.6V	
동작 온도		-40°C ~ 85°C	

표 1. STM32F030R8T6 마이크로컨트롤러 사양

STM32F030R8T6 64핀 중 55개 핀은 범용 입출력 (GPIO, General Purpose Input Output) 핀으로 사용할 수 있으며 핀 배치는 그림 1과 같다.

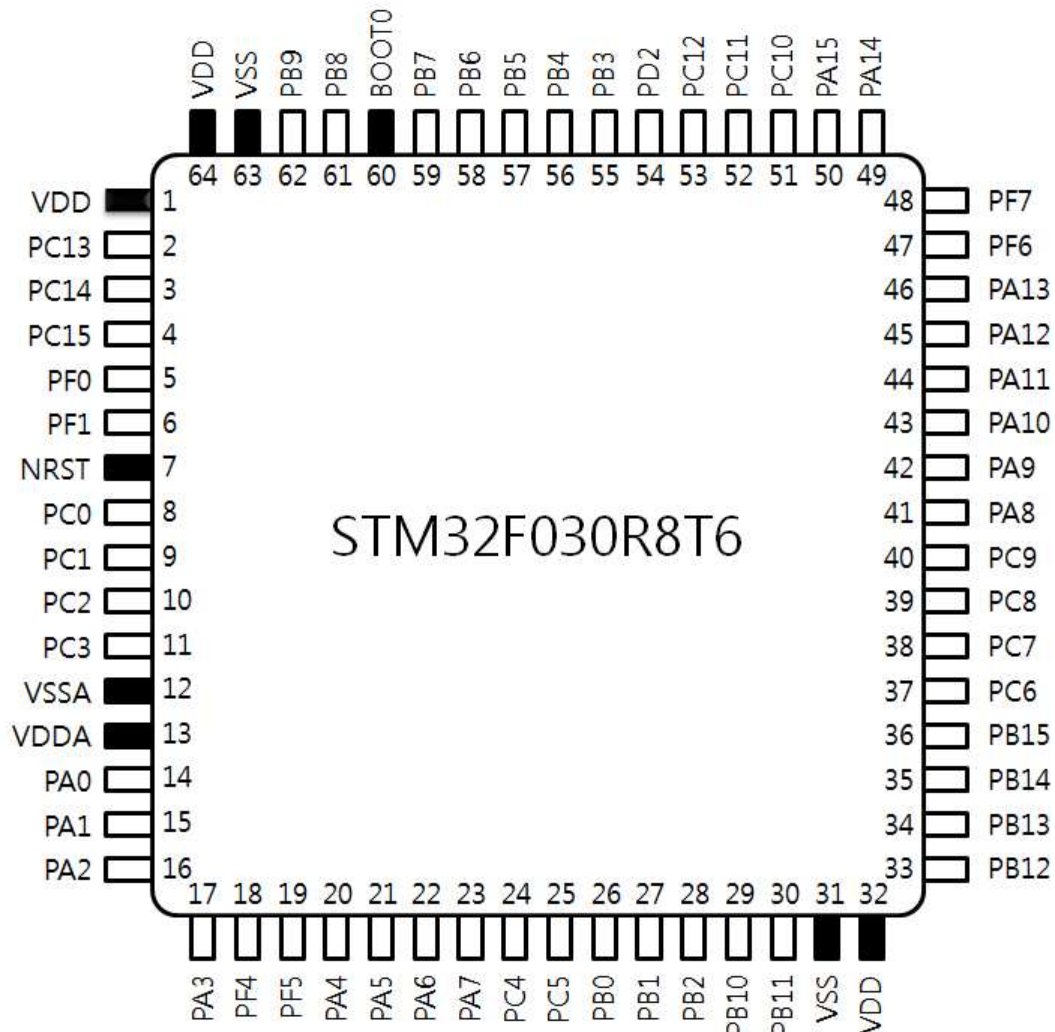


그림 1. STM32F030R8T6 범용 입출력 핀 배치

표 2는 범용 입출력 핀 이외의 특수 기능 핀을 요약한 것이다.

핀 번호	핀 이름	설명
1, 32, 64	VDD	전원
7	NRST	리셋
12	VSSA	아날로그 그라운드
13	VDDA	아날로그 전원
31, 63	VSS	그라운드
60	BOOT0	부트 메모리 선택

표 2. STM32F030R8T6 마이크로컨트롤러의 특수 기능 핀

2. 디지털 출력

STM32F030R8T6 마이크로컨트롤러의 각 핀은 누클레오 보드의 female 및 male 헤더 핀과 연결되어 있다. female 헤더 핀은 아두이노 보드와 호환되는 핀 배열을 가지고 있어 아두이노의 쉴드를 연결해서 사용할 수 있다. 반면 male 헤더 핀은 STM32F030R8T6 마이크로컨트롤러의 모든 범용 입출력 핀을 사용할 수 있도록 해준다.

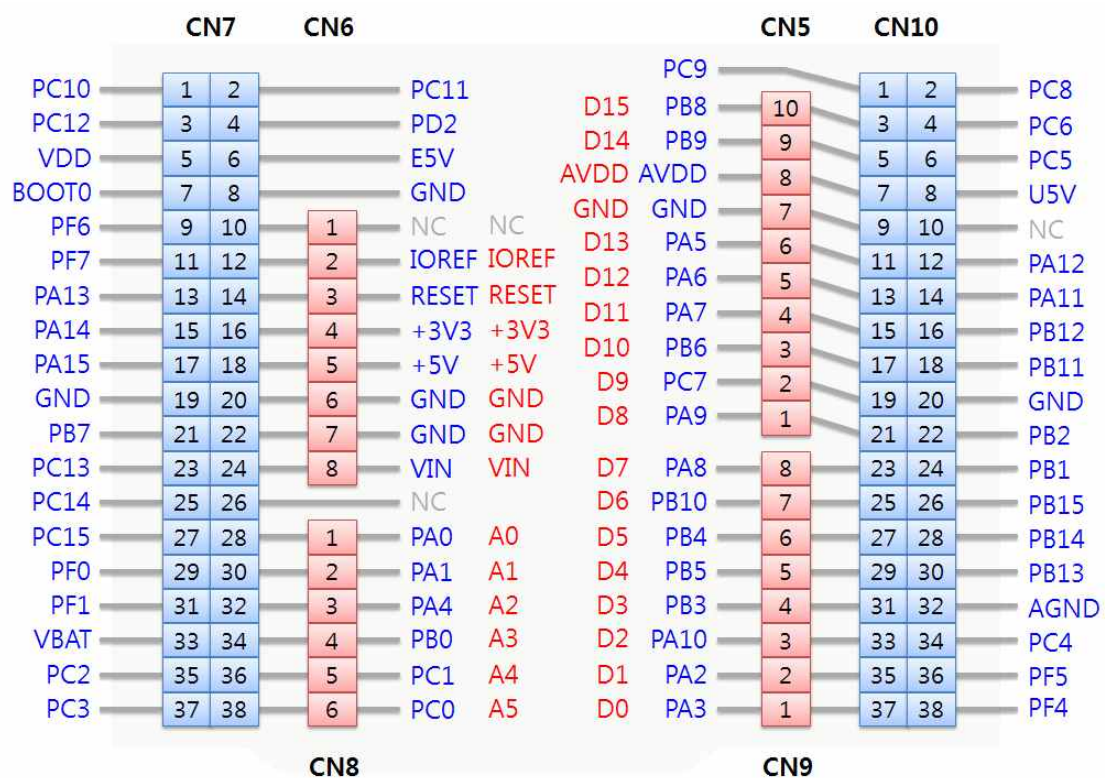


그림 2. 핀 헤더 배열

디지털 출력을 위해 먼저 mbed 환경에서 새로운 프로그램을 생성해 보자. 템플릿으로는 'Blinky LED test for the ST Nucleo boards'를 선택한다.

코드 1. Blink 프로그램

```
#include "mbed.h"

DigitalOut myled(LED1);

int main() {
    while(1) {
        myled = 1;          // LED is ON
        wait(0.2);          // 200 ms
        myled = 0;          // LED is OFF
        wait(1.0);          // 1 sec
    }
}
```

디지털 출력을 위해 사용되는 클래스는 DigitalOut으로 인자인 'LED1'은 LED(LED2)가 연결되어 있는 핀으로 D13 또는 PA5 핀을 가리킨다. 아두이노 보드의 경우 디지털 13번 핀에 연결된 내장 LED가 포함되어 있으므로 누클레오 보드에서도 이와 동일하게 D13번 핀에 LED가 연결되어 있다. GPIO 핀의 이름은 아두이노 호환 핀 헤더의 경우 A0 ~ A5까지, D0 ~ D15까지 이름이 정의되어 있으므로 핀의 이름으로 사용할 수 있다. 반면 PA5는 PORT A의 5번 핀을 가리키며 'PA_5'라는 이름으로 정의되어 있다. 즉, LED1, D13, PA_5는 모두 동일한 핀을 가리킨다. 아두이노 호환 핀을 통해 사용할 수 있는 범용 입출력 핀의 이름은 그림 3과 같다. 한 가지 추가하자면, 누클레오 보드의 중앙에 파란색 버튼(B1)에는 'USER_BUTTON'이라는 이름이 정의되어 있으며 PC13번 핀에 연결되어 있다.

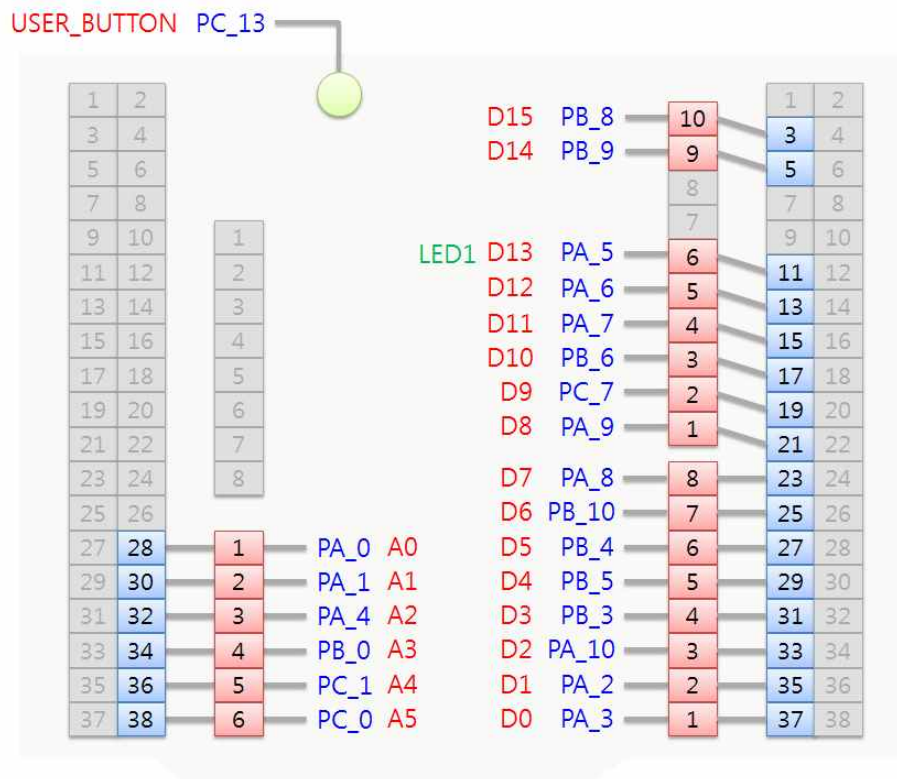


그림 3. 아두이노 호환 핀에 정의된 핀 이름

나머지 male 헤더 핀에 정의된 핀의 이름은 그림 4와 같다.

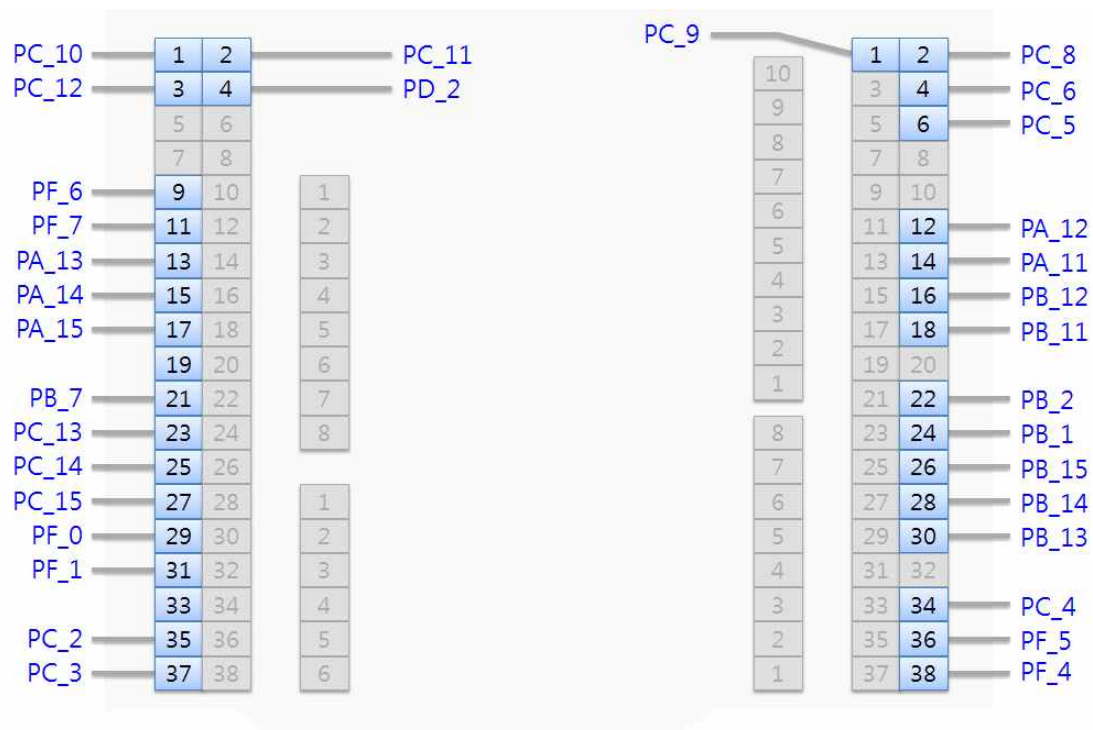


그림 4. male 헤더 핀에 정의된 핀 이름

Workspace에서 프로젝트 아래쪽을 펼쳐 보면 'mbed'라는 이름으로 mbed의 기본 클래스들이 나열되어 있음을 확인할 수 있다. 이 중 디지털 출력을 위한 클래스인 DigitalOut을 선택하면 클래스에 대한 설명을 볼 수 있다.

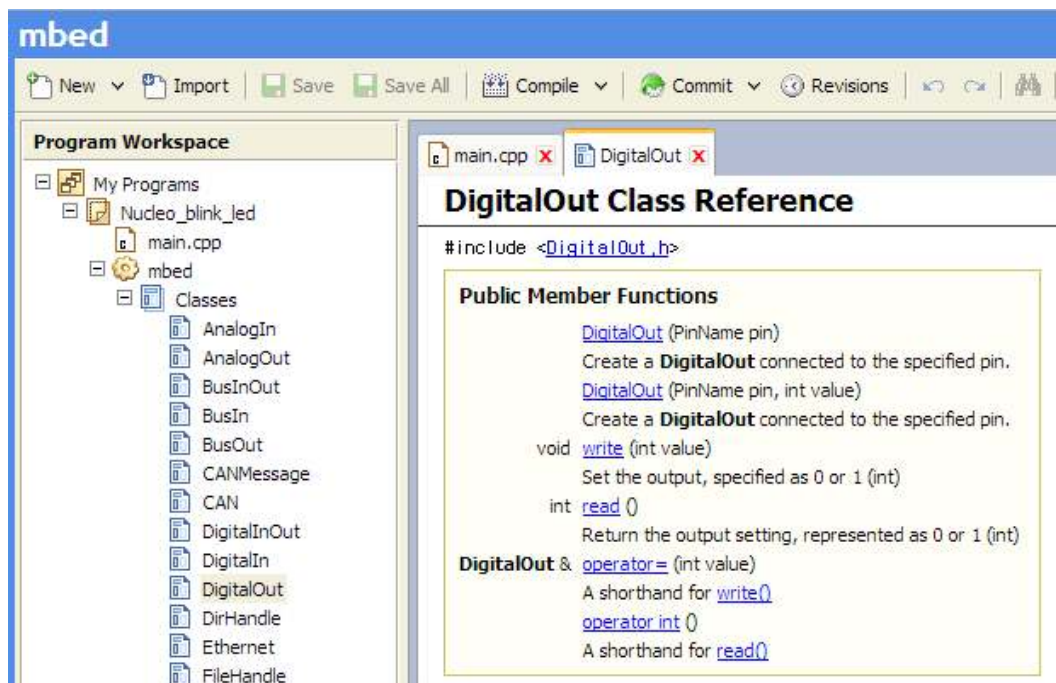


그림 5. DigitalOut 클래스

DigitalOut 클래스는 지정한 핀으로의 디지털 출력을 제어하기 위해 사용하며, 초기값을 지정하지 않는 생성자와 초기값을 지정하는 생성자 2개가 정의되어 있다.

```
DigitalOut(PinName pin)
DigitalOut(PinName pin, int value)

매개변수
    pin : 핀의 이름 (그림 3이나 그림 4에 정의된 이름을 사용)
    value : 초기값
반환값 : 없음
```

지정된 핀으로는 0이나 1의 데이터를 출력할 수 있으며, 현재 출력 상태를 얻어올 수도 있다. 이를 위해 각각 write와 read 함수가 준비되어 있다.

```
void write(int value)
    매개변수
        value : 0 또는 1의 출력값
    반환값 : 없음
```

```
int read()
    매개변수 : 없음
    반환값 : 0 또는 1의 현재 핀 상태
```

매번 write 함수나 read 함수를 사용하는 것이 번거롭다면 write 함수 대신 연산자 '='를 사용하여 값을 대입함으로써 출력할 수 있고, 캐스팅 연산자 'int'를 사용하여 현재 상태를 읽어오는 것도 가능하다.

```
myled = 1;                // myled.write(1); 와 동일
int currentValue = myled;  // int currentValue = myled.read(); 와 동일
                           // 캐스팅 연산자 없이도 자동 형변환이 이루어짐
```