



EDDI

Electronic Design  
Development Institute

---

# 에디로봇아카데미 임베디드 마스터 Lv#1 과정

제 #3기

2022. 02. 19

이광효

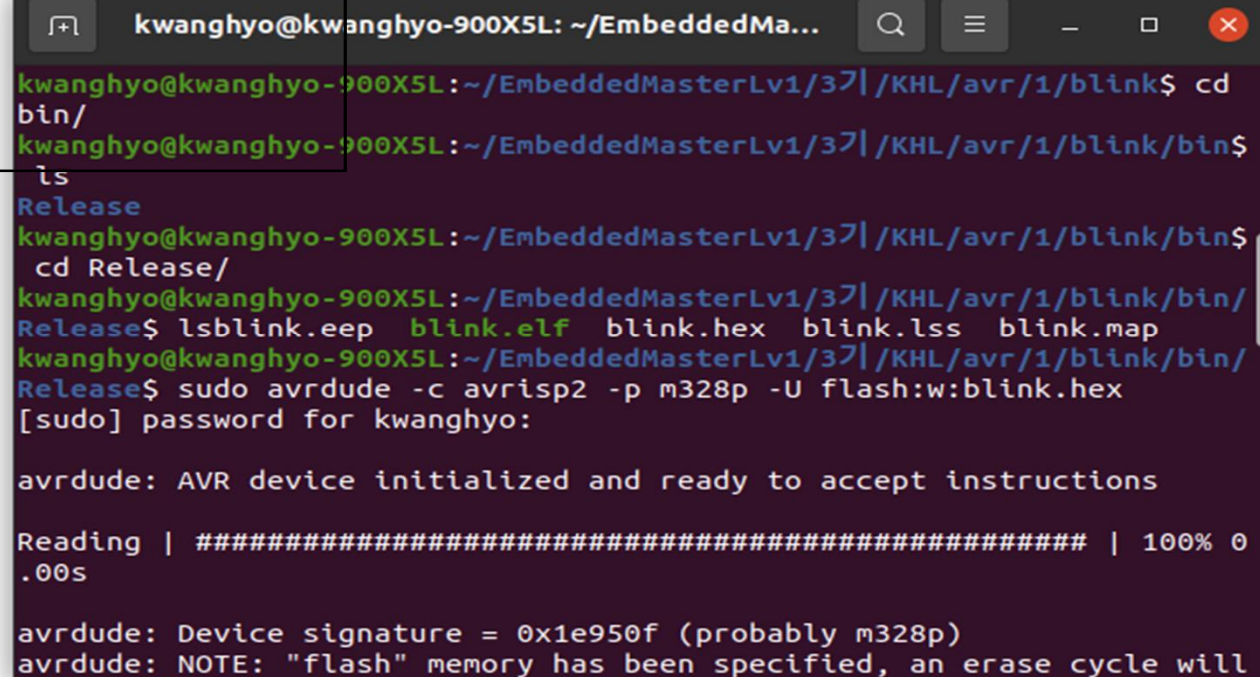
# ATmega328p light 제어 EX1

소스코드를 컴파일하기 전에  
avr 폴더에 있는 io.h를 포함

마이크로 컨트롤러의 주파수  
를 16MHz로 동작  
L은 프로그래밍 단위를 설정:  
LONG

Delay는 설정한 시간 동안 프로그램을  
중지시키는 기능

```
1  /*
2  */
3
4  #include <avr/io.h>
5
6  #define F_CPU 16000000L
7  #include <avr/delay.h>
8
9  int main(void)
10 {
11
12     DDRB = 0x28;
13
14     while(1)
15     {
16         PORTB = 0x00;
17         _delay_ms(500);
18         PORTB = 0x28;
19         _delay_ms(500);
20     }
21
22     return 0;
23 }
24
25
```



```
kwanghyo@kwanghyo-900X5L: ~/EmbeddedMa...
kwanghyo@kwanghyo-900X5L:~/EmbeddedMasterLv1/3기/KHL/avr/1/blink$ cd bin/
kwanghyo@kwanghyo-900X5L:~/EmbeddedMasterLv1/3기/KHL/avr/1/blink/bin$ ls
Release
kwanghyo@kwanghyo-900X5L:~/EmbeddedMasterLv1/3기/KHL/avr/1/blink/bin$ cd Release/
kwanghyo@kwanghyo-900X5L:~/EmbeddedMasterLv1/3기/KHL/avr/1/blink/bin/Release$ lsblink.eep blink.elf blink.hex blink.lss blink.map
kwanghyo@kwanghyo-900X5L:~/EmbeddedMasterLv1/3기/KHL/avr/1/blink/bin/Release$ sudo avrdude -c avrisp2 -p m328p -U flash:w:blink.hex
[sudo] password for kwanghyo:

avrdude: AVR device initialized and ready to accept instructions

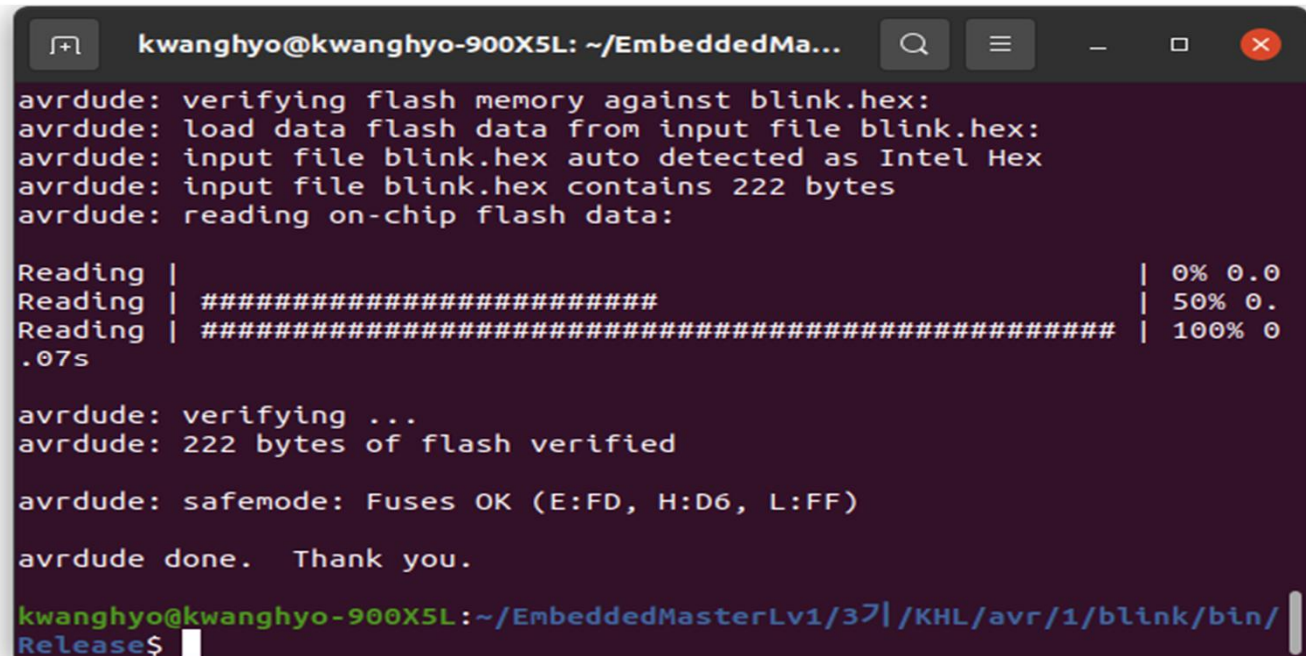
Reading | ##### | 100% 0.00s

avrdude: Device signature = 0x1e950f (probably m328p)
avrdude: NOTE: "flash" memory has been specified, an erase cycle will
```

# ATmega328p light 제어 EX2

```
4  #include <avr/io.h>
5
6  #define F_CPU 16000000L
7  #include <avr/delay.h>
8
9  int main(void)
10 {
11
12     DDRB = 0x28;
13
14     while(1)
15     {
16         PORTB = 0x00;
17         _delay_ms(500);
18         PORTB = 0x08;
19         _delay_ms(500);
20         PORTB = 0x00;
21         _delay_ms(500);
22         PORTB = 0x20;
23         _delay_ms(500);
24     }
25
26     return 0;
27
28
29 }
```

5번 포트:  $2^5$  이므로 16진수로 0X20  
3번 포트:  $2^3$  이므로 16진수로 0X08



```
kwanghyo@kwanghyo-900X5L: ~/EmbeddedMa...
avrdude: verifying flash memory against blink.hex:
avrdude: load data flash data from input file blink.hex:
avrdude: input file blink.hex auto detected as Intel Hex
avrdude: input file blink.hex contains 222 bytes
avrdude: reading on-chip flash data:

Reading |                                     | 0% 0.0
Reading | #####                             | 50% 0.
Reading | #####                             | 100% 0
.07s

avrdude: verifying ...
avrdude: 222 bytes of flash verified

avrdude: safemode: Fuses OK (E:FD, H:D6, L:FF)

avrdude done. Thank you.

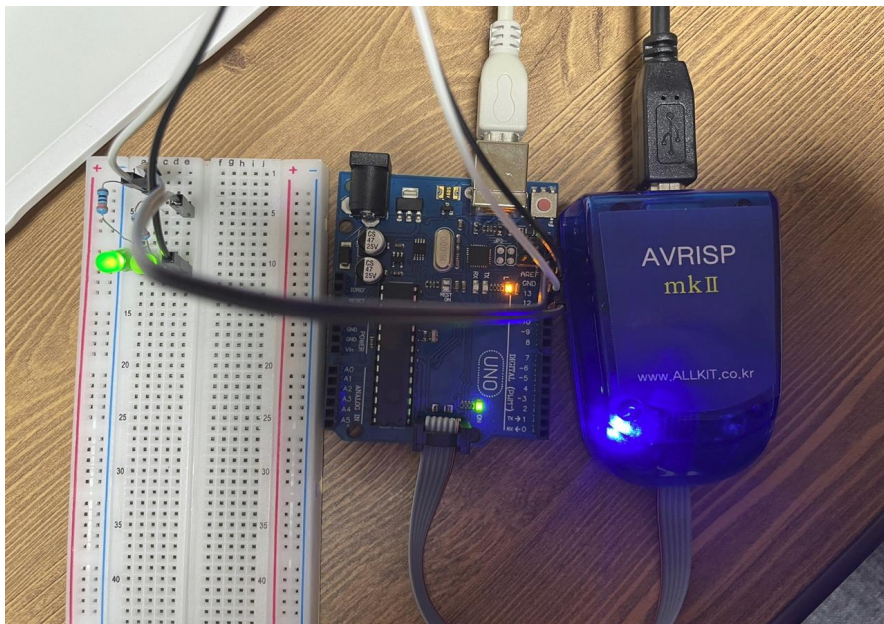
kwanghyo@kwanghyo-900X5L:~/EmbeddedMasterLv1/3기/KHL/avr/1/blink/bin/
Release$
```

0.5초 마다 3번 포트 light  
0.5초 마다 5번 포트 light  
번갈아가며 켜짐

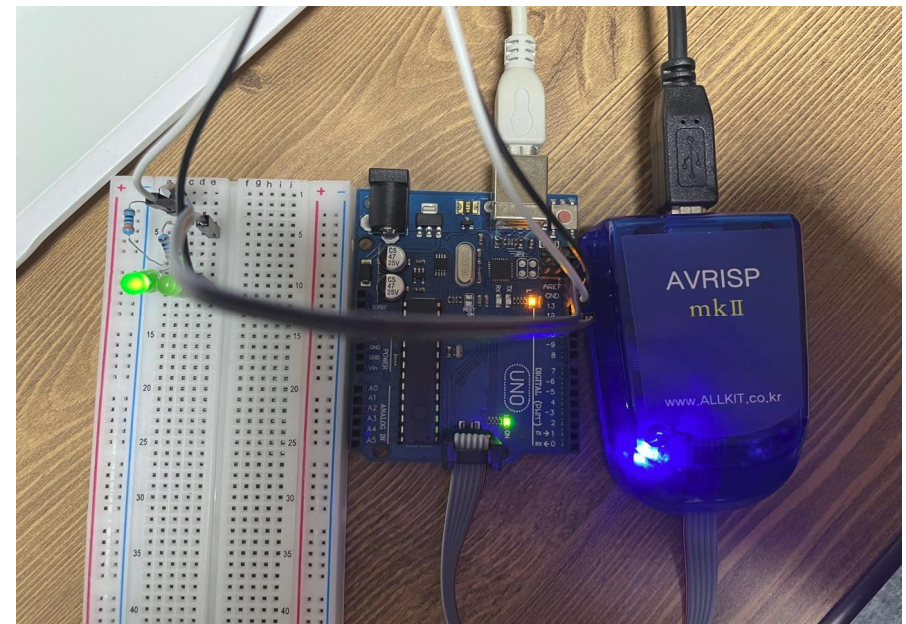


## EX1 및 EX2 하드웨어 실습 예시

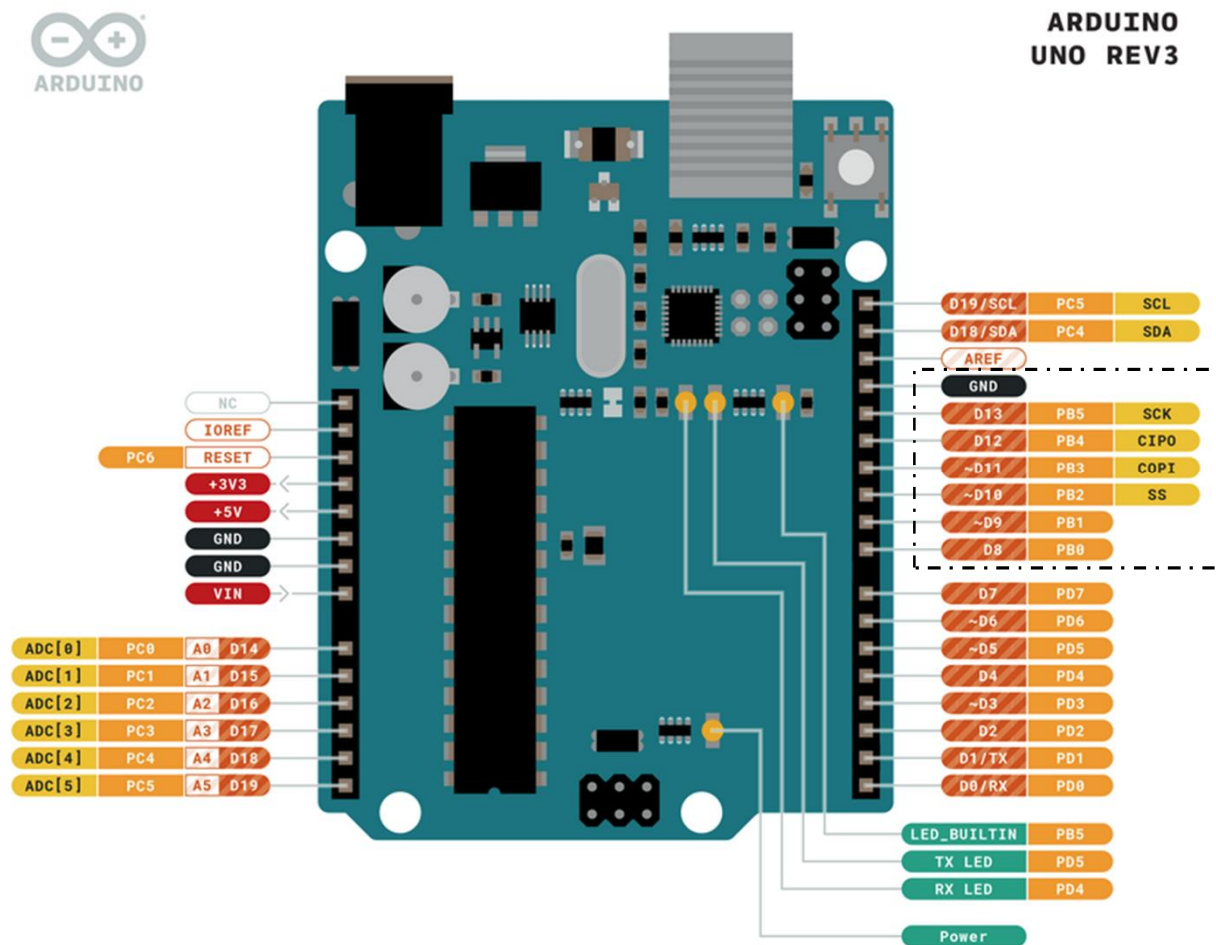
# EX1



# EX2



# Arduino Uno datasheet



1) PB5 : 0001 0000  
• 32이므로 16진수로 0X20

2) PB3 : 0000 0100  
• 8이므로 16진수로 0X08

# ATmega328p Manual



## Configuring the Pin

Each port pin consists of three register bits: DDxn, PORTxn, and PINxn. As shown in [Section 13.4 "Register Description" on page 72](#), the DDxn bits are accessed at the DDRx I/O address, the PORTxn bits at the PORTx I/O address, and the PINxn bits at the PINx I/O address.

The DDxn bit in the DDRx register selects the direction of this pin. If DDxn is written logic one, Pxn is configured as an output pin. If DDxn is written logic zero, Pxn is configured as an input pin.

If PORTxn is written logic one when the pin is configured as an input pin, the pull-up resistor is activated. To switch the pull-up resistor off, PORTxn has to be written logic zero or the pin has to be configured as an output pin. The port pins are tri-stated when reset condition becomes active, even if no clocks are running.

If PORTxn is written logic one when the pin is configured as an output pin, the port pin is driven high (one). If PORTxn is written logic zero when the pin is configured as an output pin, the port pin is driven low (zero).

각 포트 핀은 DDxn, PORTxn 및 PINxn의 세 가지 레지스터 비트로 구성됩니다. 섹션 13.4 "등록 설명"(72페이지)에 표시된 것처럼 DDRx I/O 주소에서 DDxn 비트에 액세스하고 PORTx I/O 주소에서 PORTxn 비트 및 PINx I/O 주소에 액세스합니다. DDRx 레지스터의 DDxn 비트는 이 핀의 방향을 선택합니다. DDxn이 로직 1로 작성된 경우 Pxn은 출력 핀으로 구성됩니다. DDxn이 로직 0으로 작성되면 Pxn이 입력 핀으로 설정됩니다. 핀이 입력 핀으로 구성되었을 때 PORTxn이 기록된 로직 1이면 풀업 저항기가 활성화됩니다. 풀업 저항을 끄려면 PORTxn이 로직 0으로 작성되거나 핀을 출력 핀으로 구성해야 합니다. 실행 중인 클럭이 없는 경우에도 재설정 조건이 활성화되면 포트 핀이 세 개로 정렬됩니다. 핀이 출력 핀으로 구성되었을 때 PORTxn이 기록된 로직 1이면 포트 핀이 하이(1)로 구동됩니다. 핀이 출력 핀으로 구성되었을 때 PORTxn이 로직 0으로 기록되면 포트 핀이 로우(0)로 구동됩니다.



# ATmega328p Manual 및 Q&A

## PORTB – The Port B Data Register

Bit	7	6	5	4	3	2	1	0	
0x05 (0x25)	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

## DDRB – The Port B Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x04 (0x24)	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

## PINB – The Port B Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x03 (0x23)	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

Q1. 번갈아가며 light on을 진행할 때, DDRB = 0X20, DDRB = 0X08로 설정을 하고 while문 안의 내용은 똑같이 했을 경우 왜 한 포트만 깜빡 거리는 것인가요?

Q2. 위의 질문을 진행하면서 DDB5 = 0X20, DDB3 = 0X08로 설정을 하고, while문 안을 PORTB5 = 0X00... 이런 실으로 진행하는 것은 아예 맞지 않는 방법인가요?



감사합니다.