EDDI
Electronic Design
Development Institute

에디로봇아카데미
임베디드 마스터 Lv1 과정

제 3기

2022. 02. 22

김원석

CONTENTS

EDDI
Electronic Design
Development Institute

AVR 환경 설정

EDDI
Electronic Design Development Institute

- Code::Blocks 설치

```
sudo apt-get update
sudo apt-get install codeblocks
```

Code::Blocks는 GCC, MSVC를 포함한 여러 컴파일러를 지원하는 자유, 오픈 소스, 크로스플랫폼 IDE이다.

- AVRdude 설치

```
sudo add-apt-repository ppa:pmjdebruijn/avrdude-release
sudo apt-get update
sudo apt-get install libc6-dev-i386
sudo apt-get install binutils-avr gcc-avr avr-libc avrdude libc6-dev-i386
```

avrdude는 AVR 마이크로 콘트롤러에 ROM과 EEPROM 이미지를 업로드하는 유틸리티이다.

AVR 환경 설정

EDDI
Electronic Design Development Institute

- ISP Programmer 검사

```
dmesg
```

- AVR 프로젝트 설정

```
codeblocks 에서
File -> New -> Project -> AVR Project

Next

Project title: blink
Folder to create project in: 프로젝트 경로
Project filename: blink.cbp
Resulting filename: 프로젝트 경로/blink/blink.cbp
Next

Uncheck: Create "Debug" configuration
Next

Please choose a processor for this project...: atmega328p
Uncheck: Create Fuse, Lock, Signature files (.fuse .lock .sig)
Finish
```

AVR 환경 설정

EDDI
Electronic Design Development Institute

- 프로그램 Flash 하기

```
hex 파일이 있는 위치로 이동합니다.

sudo avrdude -c avrisp2 -p m328p -U flash:w:blink.hex
```

LED 예제 – F_CPU

```
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      DDRB = 0x28;
12      // Insert code
13
14      while(1)
15      {
16          PORTB = 0x00;
17          _delay_ms(500);
18          PORTB = 0x20;
19          _delay_ms(500);
20      }
21
22      return 0;
23  }
24
```

- F_CPU는 지연함수에서 반드시 정의되어야 하는 토큰 값(보드 클럭)이다.
- 크로스 컴파일 환경에서 컴파일은 PC에서 실행되기 때문에 컴파일 한 기계어 파일이 실행될 장치의 클럭은 알지 못한다. 따라서 미리 정의해 알려준다.
- 만약 클럭이 잘못 설정되면 지연 함수를 사용했을 때 시간이 달라질 것이다.

LED 예제 – PORT 설정

```
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      DDRB = 0x28;
12      // Insert code
13
14      while(1)
15      {
16          PORTB = 0x00;
17          _delay_ms(500);
18          PORTB = 0x20;
19          _delay_ms(500);
20      }
21
22      return 0;
23  }
24
```

13.2.1 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).

- PortB의 Pin들의 입/출력 설정을 해주는 부분이다.

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	

- 해당 레지스터에 1을 쓰면 출력, 0을 쓰면 입력으로 사용된다.
- 현재 PB3과 PB5를 출력으로 설정해주었다.

LED 예제 – PIN 설정

```
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      DDRB = 0x28;
12      // Insert code
13
14      while(1)
15      {
16          PORTB = 0x00;
17          _delay_ms(500);
18          PORTB = 0x20;
19          _delay_ms(500);
20      }
21
22      return 0;
23  }
24
```

13.2.1 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).

- PortB의 Pin들의 입/출력 설정을 해주는 부분이다.

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	

- 해당 레지스터에 1을 쓰면 출력, 0을 쓰면 입력으로 사용된다.
- 현재 PB3과 PB5를 출력으로 설정해주었다.

LED 예제

```
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      DDRB = 0x28;
12      // Insert code
13
14      while(1)
15      {
16          PORTB = 0x00;
17          _delay_ms(500);
18          PORTB = 0x20;
19          _delay_ms(500);
20      }
21
22      return 0;
23  }
24
```

13.2.1 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).

- PortB에 High를 출력할지 Low를 출력할지 결정한다.

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	

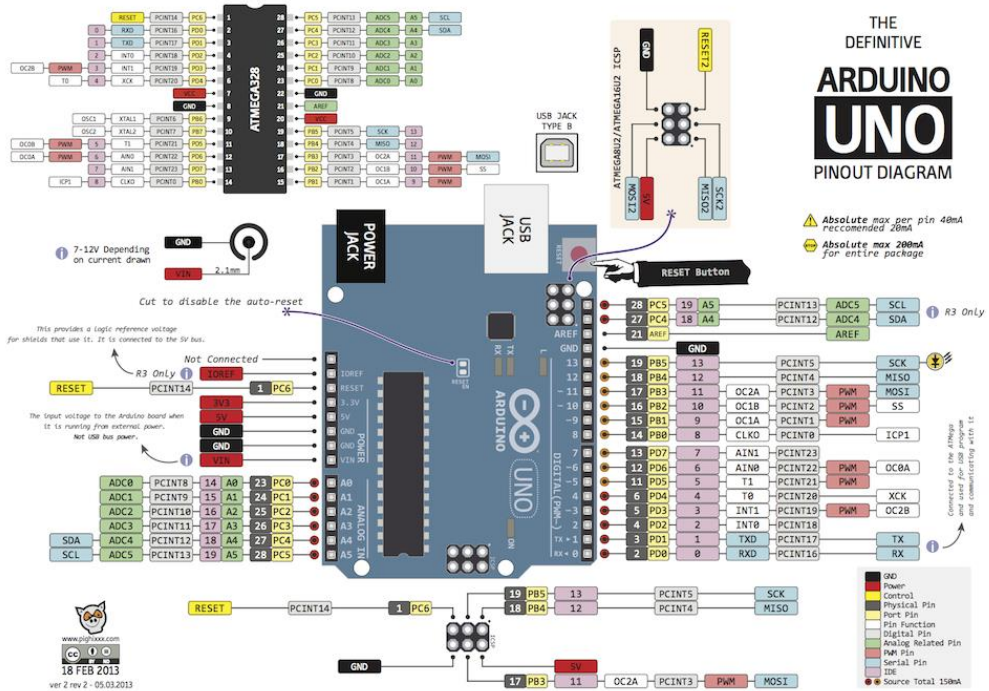
- 해당 레지스터에 1을 쓰면 High, 0을 쓰면 Low가 출력된다.

LED 예제

```
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      DDRB = 0x28;
12      // Insert code
13
14      while(1)
15      {
16          PORTB = 0x00;
17          _delay_ms(500);
18          PORTB = 0x20;
19          _delay_ms(500);
20      }
21
22      return 0;
23  }
24
```

- 0.5초 간격으로 PB5에 High와 Low를 출력하도록 했다.

아두이노 핀맵



- 핀맵을 참조해 PB5핀(13번 핀)에 LED를 연결하면 0.5초 간격으로 토글된다.