## 실습 제목

Microchip Studio for AVR의 소개 및 개발 도구 사용법

### 실습목적

AVR 계열 마이크로컨트롤러인 ATmeag328PB에 대해 간단히 살펴보고, ATmeag328PB를 이용한 장치 개발에 이용되는 개발 도구의 기본적인 사용법을 익히도록 한다.

## 실습 준비물

실습용 보드 (ATmega328PB Xplained Mini) ATmega328PB datasheet Microchip Studio for AVR

# 실습에 필요한 예비지식

1. ATmega328PB Xplained Mini

- 1) ATmega328PB Microcontroller 특징
  - ATmega328PB 마이크로컨트롤러는 단일 칩으로 구성된 MCU이며 주요 특징은 아래와 같다. Memories

32 kBytes of In-System Self-Programmable Flash Program Memory

1 kBytes EEPROM

2 kBytes Internal SRAM

#### Peripherals

Two 8-bit Timer/Counters Three 16-bit Timer/Counters Eight-channel 10-bit ADC Two USARTs Two SPIs Two TWIs Operating Voltage: 1.8 - 5.5 V Speed Grade: 0 - 4 MHz @ 1.8 - 5.5 V 0 - 10 MHz @ 2.7 - 5.5 V 0 - 20 MHz @ 4.5 - 5.5 V

### 2) ATmega328PB MCU 구성



- 3) 메모리
  - ① 데이터 메모리(data memory) 데이터 저장 메모리로 읽고 쓰기가 가능하며, 용량은 2 kByte.
  - ② 프로그램 메모리(code memory, program memory)
     명령 실행코드(program code)가 저장되는 메모리로서 32 kByte의 Flash 메모리로 구성.

4) ATmega328PB Development Kit (ATmega328PB Xplained Mini)



#### 2. Microchip Studio for AVR 사용법

Microchip사의 Microchip Studio for AVR은 프로젝트 파일 관리, 소스 코드 편집, 프로그램 디버깅, 시뮬레이션 등을 할 수 있는 통합 개발 환경이다. 또한 프로그램을 보다 쉽고 빠르게 만들 수 있으며, 하나의 어플리케이션으로 에디터와 디버거를 모두 사용할 수 있는 장점을 가지고 있다.

# 실습 내용

- 1. Microchip Studio for AVR과 <u>C 언어</u>를 사용하여 ATmega328PB Xplained Mini Board에 설치되어 있는 LED를 점멸시키는 프로그램을 만들고, 디버깅을 해 본다.
  - 1) 프로젝트(project) 생성
    - (1) 파일(F) 새로만들기(N) 프로젝트(P) 항목을 선택한다.

ø	AtmelStudio (관리자)		-		
파일	닐(F) 변십(E) 보기(V) VASSISTX	ASF 프로젝트(P)	L	버그(D) 도구(1) 경	양(W) 노움말(H)
	새로 만들기(N)	•	韵	프로젝트(P)	Ctrl+Shift+N
	열기(O)	•	*ъ	파일(F)	Ctrl+N
	달기(C)			Atmel Start Configura	ator
EB	솔루션 닫기(T)		#	Example Project	Ctrl+Shift+E
	Import	٠			
6	선택한 항목 저장(S)	Ctrl+S			
	다른 이름으로 선택한 항목 저장(A)				
<b>1</b> 2	모두 저장(L)	Ctrl+Shift+S			
	템풀릿 내보내기(E)				
Ð	페이지 설정(U)				
樱	인 쇼핑(P)	Ctrl+P			
	최근에 사용한 파일(F)	•			
	최근에 사용한 프로젝트 및 솔루션(J)	٠			
X	끝내기(X)	Alt+F4			

(2) 왼쪽 메뉴 항목에서 ①C/C++ 항목를 선택하고. 가운데 창에서 ②GCC C Executable Project (C/C++)를 선택한 후, ③찾아보기 버튼을 클릭하여 ④프로젝트 파일이 저장될 폴더의 위치를 지정한 다음, ⑤프로젝트 이름(Blinky)을 입력하고, ⑥'솔류션용 디렉터리 만들기' 항목을 선택한 다음 ⑦확인 버튼을 클릭한다.

새 프로젝트			? <b>**</b>
▷ 최근 항목	정렬 기준: 기본값 🔹 🏢 🔚		설치된 템플릿 검색(Ctrl+E) 🔎 -
▲ 설치됨	GCC C ASF Board Project	C/C++	형식: C/C++ Creater an AVR & hit or AV/P/ARM 22.hit
Assembler (2) AtmelStudio Solution	GCC C Executable Project	C/C++	C project
	GCC C Static Library Project	C/C++	
	GCC C++ Executable Project	C/C++	int main
	GCC C++ Static Library Project	C/C++	printf("Haz
	Create project from Arduino sketch	C/C++	GCC
이름(N): (5) Blinky			3
위치(L): ④ r System Lab II+	2017-2#Lab Manuals#Week_1_Development_Tools	₩Programs₩ -	찾아보기(B) 6
솔루션 이름(M): Blinky			✓ 솔루션용 디렉터리 만들기(D)
			7 확인 취소

디지털 시스템 및 마이크로컴퓨터시스템 실습 I Microchip Studio for AVR 소개 및 사용법

(3) 'Device Selection' 창에 ①'328PB'라고 입력하면 해당 이름을 가진 소자만 선별되어 출력된다. 이 중에서 ②'ATmega328PB' 소자를 클릭하여 선택한 다음, ③OK 버튼을 클릭한다.

Device Selection			×
Device Family:	All	(1) 328PB	×
Name ATmega328PB	App./Boot Memory (Kbytes) Data Memory (bytes) EEPROM (bytes) 32 2048 1024 (2)	Device Info: Device Name: Speed: Vcc: Family: Datasheet (Summary) Device Page Supported Tools Atmel-ICE AVR Dragon AVRISP mkll AVR ONE! X EDBG X EDBG MSD J JAGICE3	ATmega328PB N/A N/A ATmega
			3
			OK Cancel

(4) Microchip Studio for AVR이 자동으로 프로젝트의 골격을 생성하고, 아래 그림과 같이 아주 간단한 main.c 파일을 생성해 준다.



(5) 이 main.c 파일을 다음과 같이 수정한다.

```
1 0/*
     * Blinky.c
 2
 3
    *
4
    * Created: 2017-08-25 오후 5:44:02
 5
    * Author : Administrator
     */
6
 7
8
    #define F_CPU 16000000
9
10
   #include <avr/io.h>
11
   #include <util/delay.h>
12
13 pint main(void)
14
    {
15
        DDRB = 0b00100000;
        while (1)
16
17
        {
18
            PORTB = 0b00100000;
19
            _delay_ms(100);
            PORTB = 0b0000000;
20
21
            delay ms(900);
22
        }
23
   }
```

디지털 시스템 및 마이크로컴퓨터시스템 실습 I Microchip Studio for AVR 소개 및 사용법

(6) Build - Build Solution 메뉴를 선택하여 위 프로그램을 Compile 및 Link 한다.

Blinky - AtmelStudio (관리자)       1         파일(F) 편집(E) 보기(V) VAssistX 프로젝트(P)	빌드	E(B) 디버그(D) 도구	(T) 창(W) 도움
G • O   智 • 智 • 🖕 💾 🔐 🐰 🗗 街 🔰	*	솔루션 빌드(B)	F7
MI B II → II ▶   ↔ : ? * ▶ T   16		솔루션 다시 빌드(R) 솔루션 정리(C)	Ctrl+Alt+F7
main.c* ↔ × → main.c	*	Blinky 빌드(U) Blinky 다시 빌드(E) Blinky 정리(N)	5
2 * Blinky.c 3 * 4 * Created: 2017-08-25		일괄 빌드(T) 구성 관리자(O)	Ctd. 57

(7) 프로그램에 문제가 없고 정상적으로 빌드가 되면 출력창에 아래의 그림과 같은 프로젝트가 정상적으로 빌드 되었다는 메시지가 출력된다.

출력		
출력 보기 선택(S):	빌드	-
<avrgcc.assemb <avrgcc.assemb </avrgcc.assemb </avrgcc.assemb 	, bler.general.includePaths> )ler.debugging.DebugLevel>Default (-Wa,-g) <th>cc.asse</th>	cc.asse
r RunOutputFi	Program Memory Usage : 254 bytes 0.8 % F Data Memory Usage : 0 bytes 0.0 % Ful leVerifyTask" 작업을 실행했습니다.	ull I
"Blinky.cpro)" 프 조건이 잘못되어 " "E:#Lecture#Under "Blinky.cproj" 프 "Blinky.cproj" 프	프로젝트에 "CoreBuild" 대상을 빌드했습니다. "PostBuildEvent" 대상을 건너뜁니다. (`\$(PostBui erGraduate#Digital Circuits and Microcomputer Sy 프로젝트에 "Build" 대상을 빌드했습니다. 프로젝트를 빌드했습니다.	ldEvent stem La
빌드했습니다. ======== <mark>빌드:</mark>	성공 또는 최신 상태 1, 실패 0, 생략 0=======	

(8) USB Cable을 사용하여 ATmega328PB Xplained Mini Board를 PC에 연결하면, 잠시 후에 인식하였다는 화면이 나타난다.



(9) 잠시 더 기다리면 이 Board와 PC가 직렬 통신을 하는데 필요한 Driver가 설치되고 다음과 같이 Virtual COM가 설치되었다는 창이 나타난다. 닫기 버튼을 클릭하여 창을 닫는다.

프라이버 소프트웨어 설치		×
mEDBG Virtual COM Port(CON	₩40) 설치됨	
mEDBG Virtual COM Port(COM40)	✔사용 준비 완료	
		[달기(C)]

디지털 시스템 및 마이크로컴퓨터시스템 실습 I Microchip Studio for AVR 소개 및 사용법

(10) 빌드된 프로그램을 ATmega328PB Xplained Mini Board에 있는 ATmega328PB MCU 내부의 Flash Memory에 써 넣기 위해서 이 프로젝트에 대한 속성의 일부를 변경해 주어야 한다. 프로젝트 메뉴 - Blinky 속성을 선택한다.



(11) 아래 그림과 같이 프로젝트 속성창에서 ①Tool 항목을 선택한 후,
 ②Selected\_debugger/programmer 항목에서 현재 PC에 연결된 ATmega328PB Xplained Mini Board에 해당하는 mEDBG ATML\*\*\*\* 항목을 선택하고, ③Interface 항목에서 debugWIRE 항목을 선택한 후, 다시 빌드를 실행한다.

Blinky* → × ATmega3	28PB Xplained Mini - 1653	main.c	
Build Build Events	구성(C): N/A	▼) 플랫폼(M): N/A	
Toolchain Device 1 Tool* Components Advanced	Selected debugger/program mEDBG • ATML25230302	nmer 00001653 🔻 Interface: debu 3	gWIRE 🗸

(12) 빌드된 프로그램을 ATmega328PB Xplained Mini Board에 있는 ATmega328PB MCU 내부의 Flash Memory에 써 넣은 후, 즉시 프로그램을 실행시키기 위해 디버그 메뉴의 디버그하지 않고 시작 항목을 선택한다.



디지털 시스템 및 마이크로컴퓨터시스템 실습 I Microchip Studio for AVR 소개 및 사용법

(13) 사용하는 환경에 따라 ATmega328PB XMINI board의 Firmware를 Upgrade해야 한다는 창이 나타날 수 있으며, 이 경우에는 Upgrade 버튼을 클릭하여 Firmware를 Upgrade를 시작한 후, Upgrade가 종료되면 Close 버튼을 클릭한다. 위 (12) 항목을 한 번 더 실행한다.

Firmware Upgrade		×	Firmware Upgrade		×
mEDBG firmware mu	ist be updated bef	ore continuing	mEDBG firmware is u	up to date	
	On Tool	On Disk		On Tool	On Disk
Firmware Version	1.13	1.16	Firmware Version	1.16	1.16
Firmware Upgrade	U	ograde Close	mEDBG firmware suc	ccessfully upgrade	grade Close

(14) 프로그램이 다운로드 되는 동안에는 창의 아래쪽에 그림과 같이 진행 상황이 숫자로 나나타고, 다운로드가 완료되는 즉시 프로그램이 실행되어 Board 상의 주황색 LED가 점등되는 것을 볼 수 있다.

빌드했습니다. ====== 빌드: 	성공 또는	최신 상태	1, 실패 0,	생략 0 ===	
· · · · · · · · · · · · · · · · · · ·					
Writing (28 %)					
<u>@</u>		0	9	6	

(15) 프로그램을 디버깅 하기 위해 디버그 메뉴 항목의 Start Debugging and Break 항목을 선택한다.



디지털 시스템 및 마이크로컴퓨터시스템 실습 I Microchip Studio for AVR 소개 및 사용법

(16) 빌드된 프로그램이 PC에서 ATmega328PB XMINI board로 다운로드 되고 나면, 화면이 debugging을 위한 상태로 전환된다. 아래의 그림과 같이 프로그램이 다운로드 되고 난 후, 프로그램의 시작점인 15번 라인에서 디버깅을 위한 명령을 기다리고 있다.



디지털 시스템 및 마이크로컴퓨터시스템 실습 I Microchip Studio for AVR 소개 및 사용법

(17) 이 15번 라인에 있는 statement의 기능은 ATmega328PB MCU 내부에 있는 입출력 설정을 위한 레지스터 가운데 하나인 DDRB 레지스터에 2진수 00100000을 써 넣는 것이다.
Microchip Studio for AVR의 디버깅 기능을 통해 실제로 DDRB 레지스터에 값이 써 넣어지는 것을 확인해 볼 수 있다. 이를 위해 ①디버그 - ②창 - ③I/O 메뉴 항목을 선택하여 입출력 레지스터의 값을 보여주는 나타내는 창을 연다.

	창(W)	•		중단점(B)	Alt+F9
	Start Debugging and Break	Alt+F5		Data Breakpoints	
	Attach to Target			Processor Status	
I,	디버깅 중지(E)	Ctrl+Shift+F5		1/0 (3)	
	디버그하지 않고 시작(H)	Ctrl+Alt+F5	R	Live Watch	
	Disable debugWIRE and Close		9	Program Counter Trace	
	Continue	F5	₽	출력(O)	

(18) I/O 창에서 PORTB 항목을 클릭하면 아래쪽의 창에 PORTB에 관련된 세 개의 레지스터 값들이 출력된다. 여기서 Value 항목은 각 레지스터에 저장되어 있는 값을 16진수로 나타낸 것이고, Bits 항목은 2진 형태의 그림으로 나타낸 것으로 각 비트에 해당하는 사각형이 비어 있으면 0, 청색이나 적색으로 채워져 있으면 1을 나타낸다.

I/O				• 🗆 ×
Filte	er:		- 🔟	
	Name		Value	
CPU     EEPRC     EXINT	M			
VO PORTE	3			
PORTO				-
Name	Address	Value	Bits	
VO PINB	0x23	0x9F		
10 DDRB	0x24	0x00	00000000	0
VO PORTE	3 0x25	0x00	0000000	

(19) 15번 라인의 한 statement만 실행하고 멈추기 위해 ①디버그 - ②프로시저 단위 실행 항목을 선택한다.

디버	1그(D) 도구(T) 창(W)	도움말(H)
	창(W)	٠
011	Start Debugging and Break	Alt+F5
Ē.	Attach to Target	
•	디버깅 중지(E)	Ctrl+Shift+F5
Þ.	디버그하지 않고 시작(H)	Ctrl+Alt+F5
	Disable debugWIRE and Clo	se
•	Continue	F5
4	Execute Stimulifile	
4	Set Stimulifile	
ΰ	다시 시작(R)	
11	모두 중단(K)	Ctrl+F5
69	간략한 조사식(Q)	Shift+F9
*	한 단계씩 코드 실행(I)	F11
?	프로시저 단위 실행(0)	F10
:	프로시저 나가기(T)	Shift+F11
×.	커서까지 실행(N)	Ctrl+F10
Ť	Reset	Shift+F5

(20) 15번 라인의 statement가 실행을 완료한 후, 18번 라인에서 프로그램의 실행은 멈추게 되고, 갱신된 입출력 레지스터의 값들이 적색으로 표시된다. 이런 동작을 반복하여 프로그램의 논리적인 오류를 쉽게 찾아낼 수 있다.



(21) 이후의 모든 statement를 실행한 후 20번 라인에서 실행을 멈추기 위해 20번 라인의 가장 왼쪽 부분을 클릭하면 그림과 같이 20번 라인에 Break Point가 설정된다.

	13 E	⊒int main(void)	1/0 👻	Ξ×			
	14	{	Filter:				
	15	DDRB = 0000100000;	Name Value				
	16	while (1)	🕀 🧰 CPU				
	17	{		T			
	18	PORTB = 0b00100000;					
	19	_delay_ms(100);	VO PORTS				
•	20	PORTB = 0b00000000;	PORTD	*			
	21	deLay ms(900);	Name Address Value Bits				
	22	1	100 PINB 0x23 0x9F	14			
	22	, J	100 DDRB 0x24 0x20 00000000000000000000000000000				
	23	}	PORTB 0x25 0x00	*			

(22) Break Point가 설정되었으면 ①디버그 - ②Continue 항목을 선택한다.

CIH	(D) ニ	도구(T)	창(W)	도움말(H)
	창(W)			•
M	Start D	ebugging	Alt+F5	
ñ	Attach	to Target		
	티버깅	중지(E)		Ctrl+Shift+F5
Þ.	디버그	하지 않고	Ctrl+Alt+F5	
	Disable	e debugWI	RE and Clo	ose
•	Contin	iue		F5
1	Execut	e Stimulifil	e	

(23) 20번 라인에서 프로그램의 실행이 멈추게 되고, 변경된 레지스터의 내용이 I/O 창에 적색으로 표시된다.



(24) 디버깅을 종료하려면 ①디버그 - ②디버깅 중지 항목을 선택한다.

CIE	비그(D)	도구(T)	창(W)	도움말(H)
	창(W)			•
φЩ.	Start D	ebugging	and Break	Alt+F5
Ē	Attach	to Target		
	디버깅	중지(E)		Ctrl+Shift+F5
Þ	디버그	하지 않고	시작(H)	Ctrl+Alt+F5
	Disable	e debugWI	RE and Clo	ose

(25) 위 항목들을 적절하게 조합하여 실행함으로써 프로그램의 실행 상태를 이해할 수 있고, 논리적인 오류도 검출할 수 있다.

디지털 시스템 및 마이크로컴퓨터시스템 실습 I Microchip Studio for AVR 소개 및 사용법

2. Microchip Studio for AVR과 <u>Assembly 언어</u>를 사용하여 ATmega328PB Xplained Mini Board에 설치되어 있는 LED를 점멸시키는 프로그램을 만들고, 디버깅을 해 본다.

프로그램을 만드는 절차는 C언어와 유사하고 이번 주 동영상에 상세하게 설명되어 있으므로 설명은 생략함.

; blinky.asm

•							_	
, lp:	sbi rcall	PORTB,5 dly100ms	; F ; 1	PB5에 100ms	논 의	리값 시간	' <b>1'</b> 이 지연	출력
	cbi rcall	PORTB,5 dly900ms	;F ;9	PB5에 900ms	논 의	리값 시간	'0'이 지연	출력
	rjmp	lp						
;=====================================				=====		===:	=	
	push ldi	r25 r25,9						
a1y900ms_1:	call dec brne pop ret	dly100ms r25 dly900ms_1 r25						
; dly100ms:								
d]v100ms 1.	push ldi	r25 r25,100						
ury100ms_1.	call dec brne pop ret	dly1ms r25 dly100ms_1 r25						
; total # of clk	( = 14 + 4	4*3999 + 3 = 1601	L3					
dly1ms:	.equ push push	count_1ms = 4000 r24 r25	9		;;	2 2	subrou	tine
;	ldi ldi	r24,LOW(count_1 r25,HIGH(count_1	ms) 1ms)	1	;;	1 1		
dly1ms_1:						_		
	sbiw brne	r24,1 dly1ms_1			;;	2 2 /	1	
;	pop pop ret	r25 r24			;;;;	2 2 4		

## 과제

- C 언어에서 사용되는 다음 연산자들의 동작에 대해 예제를 들어 요약 설명하여 제출하시오.
   (1) 산술 연산자 (+, -, \*, /, %)
   (2) 대입 연산자 (+=, -=, \*=, /=)
   (3) 관계(비교) 연산자 (>, <, >=, <=, ==, !=)</li>
   (4) 논리 연산자 (&&, ||, !)
   (5) 비트 연산자 (&, |, ^, ~)
   (6) 시프트 연산자 (<<, >>)
   (7) 기타 연산자 ( sizeof, ?: )
- 2) C 언어에서 사용되는 while loop, for loop 및 if else 문에 대해 예제를 들어 요약 설명하여 제출하시오.
- 3) C 언어에서 사용되는 #include와 #define에 대해 예제를 들어 요약 설명하여 제출하시오.