EMBEDDED SCHOOL

AVR – HW1

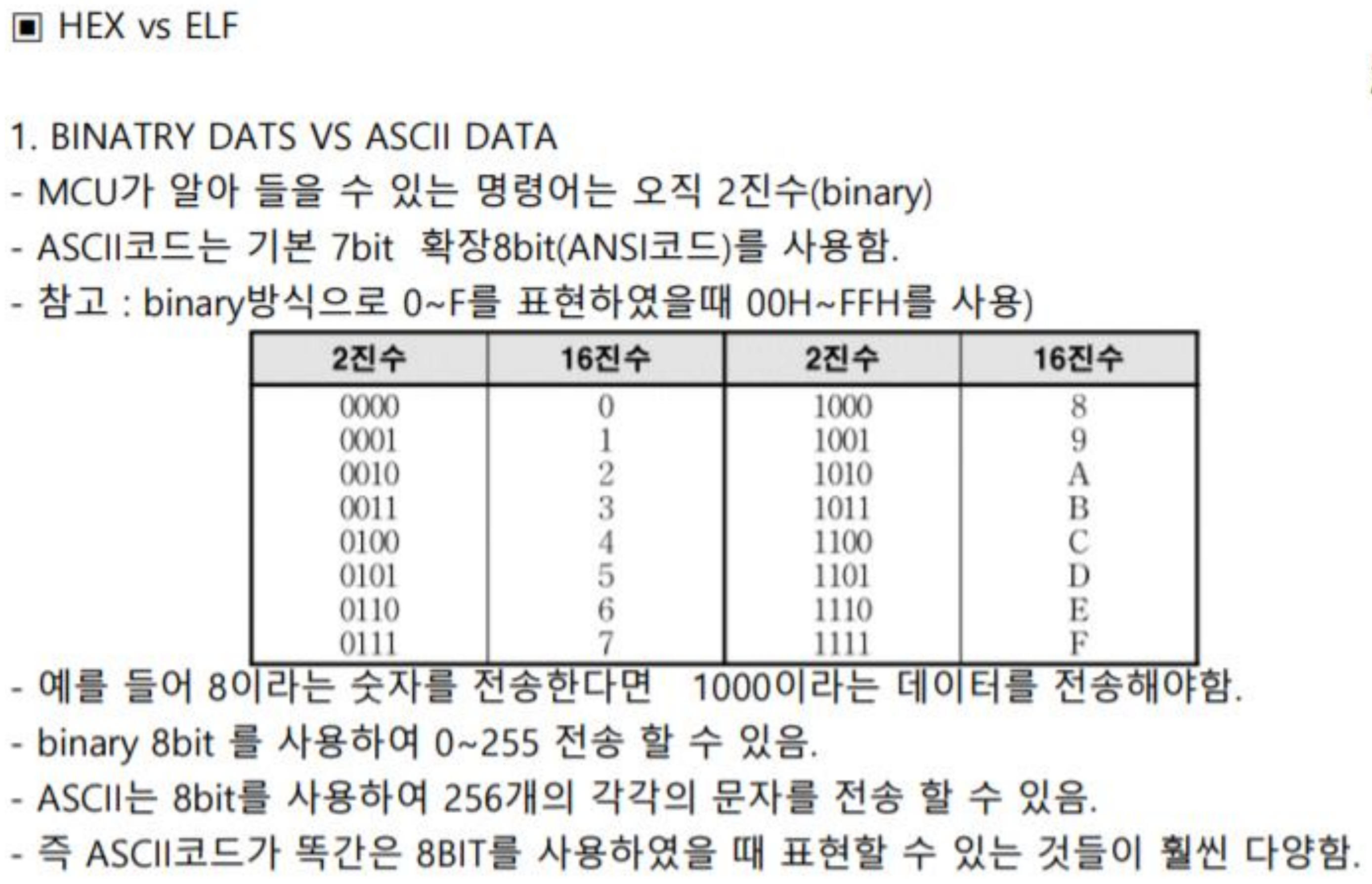
임베디드스쿨1기

Lv1과정

2020. 09. 05

강경수

1. HEX파일



▣ HEX vs ELF

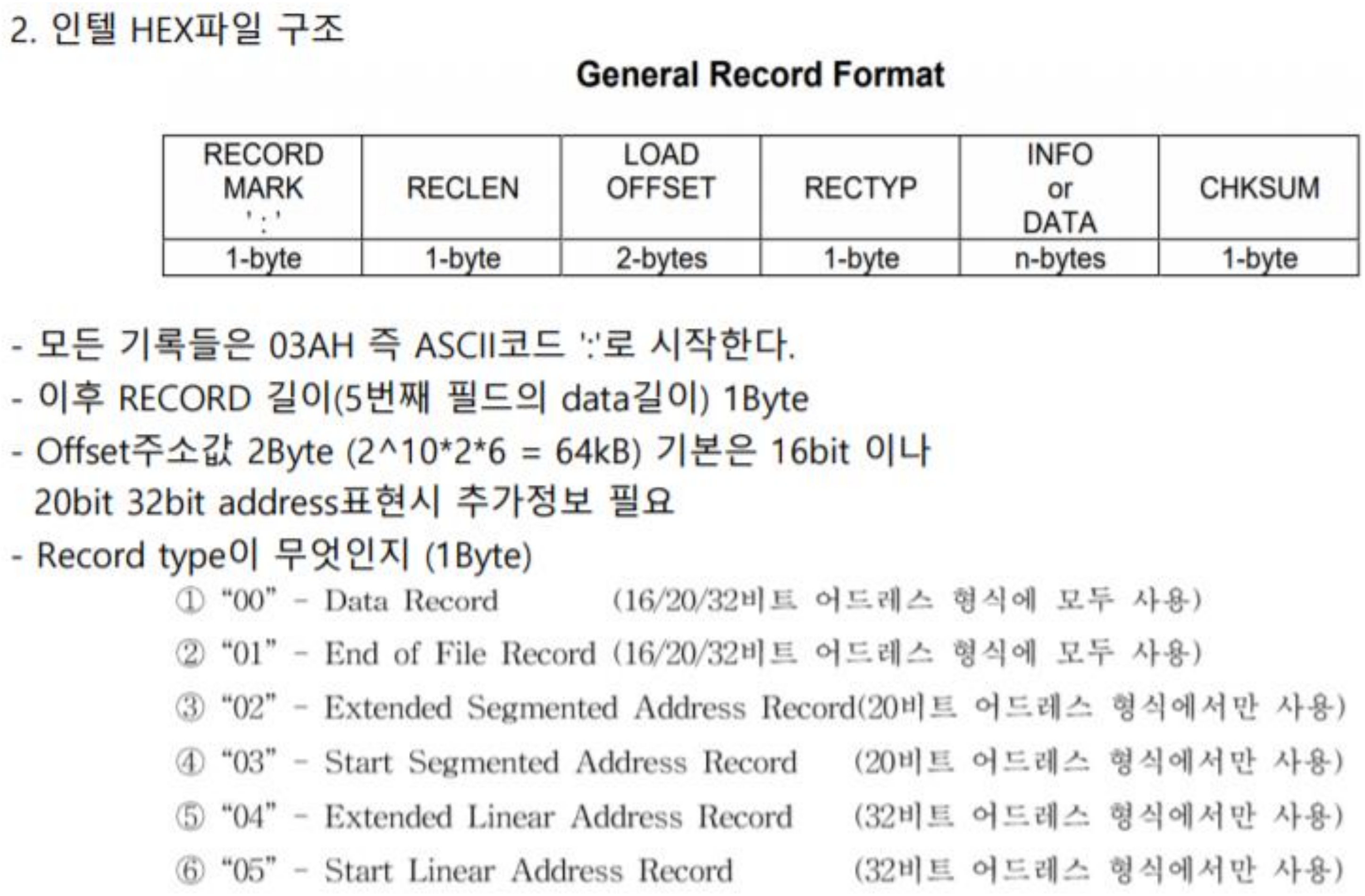
2020.09.03 KKS

1. BINATRY DATS VS ASCII DATA
- MCU가 알아 들을 수 있는 명령어는 오직 2진수(binary)
- ASCII코드는 기본 7bit 확장8bit(ANSI코드)를 사용함.
- 참고 : binary방식으로 0~F를 표현하였을때 00H~FFH를 사용)

2진수	16진수	2진수	16진수
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

- 예를 들어 8이라는 숫자를 전송한다면 1000이라는 데이터를 전송해야함.
- binary 8bit 를 사용하여 0~255 전송 할 수 있음.
- ASCII는 8bit를 사용하여 256개의 각각의 문자를 전송 할 수 있음.
- 즉 ASCII코드가 똑간은 8BIT를 사용하였을 때 표현할 수 있는 것들이 훨씬 다양함.

1. HEX파일



2. 인텔 HEX파일 구조

General Record Format

RECORD MARK ':'	RECLEN	LOAD OFFSET	RECTYP	INFO or DATA	CHKSUM
1-byte	1-byte	2-bytes	1-byte	n-bytes	1-byte

- 모든 기록들은 03AH 즉 ASCII코드 ':'로 시작한다.
- 이후 RECORD 길이(5번째 필드의 data길이) 1Byte
- Offset주소값 2Byte (2^10*2*6 = 64kB) 기본은 16bit 이나
  20bit 32bit address표현시 추가정보 필요
- Record type이 무엇인지 (1Byte)

① "00"	- Data Record	(16/20/32비트 어드레스 형식에 모두 사용)
② "01"	- End of File Record	(16/20/32비트 어드레스 형식에 모두 사용)
③ "02"	- Extended Segmented Address Record	(20비트 어드레스 형식에서만 사용)
④ "03"	- Start Segmented Address Record	(20비트 어드레스 형식에서만 사용)
⑤ "04"	- Extended Linear Address Record	(32비트 어드레스 형식에서만 사용)
⑥ "05"	- Start Linear Address Record	(32비트 어드레스 형식에서만 사용)

포기하면 얻는 건 아무것도 없다.

1. HEX파일

- INFO or DATA '00~01 기본형, 02~03 세그먼트 어드레스, 04~05 32비트 어드레스

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	"02" 2byte
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	LOAD OFFSET

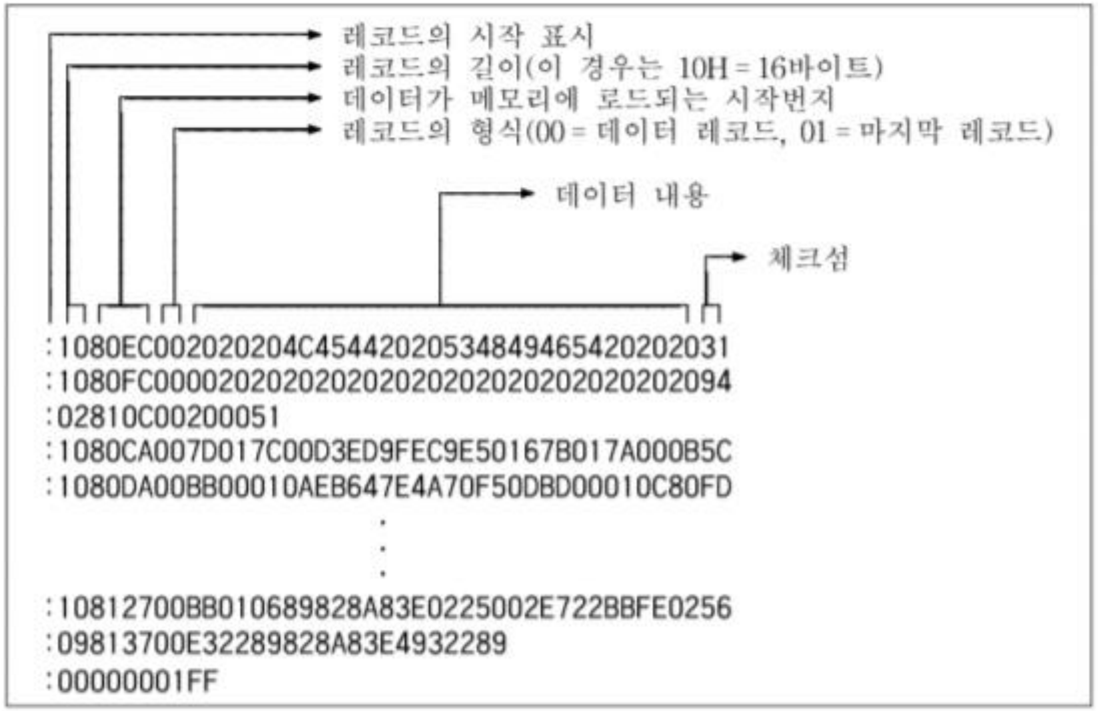
(20bit address)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	5th field 4Byte
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	LOAD OFFSET

(32bit address)

- CHECKSUM ':' 를 제외한 2~5번째 필드에서 차례로 2문자씩을 8bit 2진수로 변환하여 더한값의 2의 보수

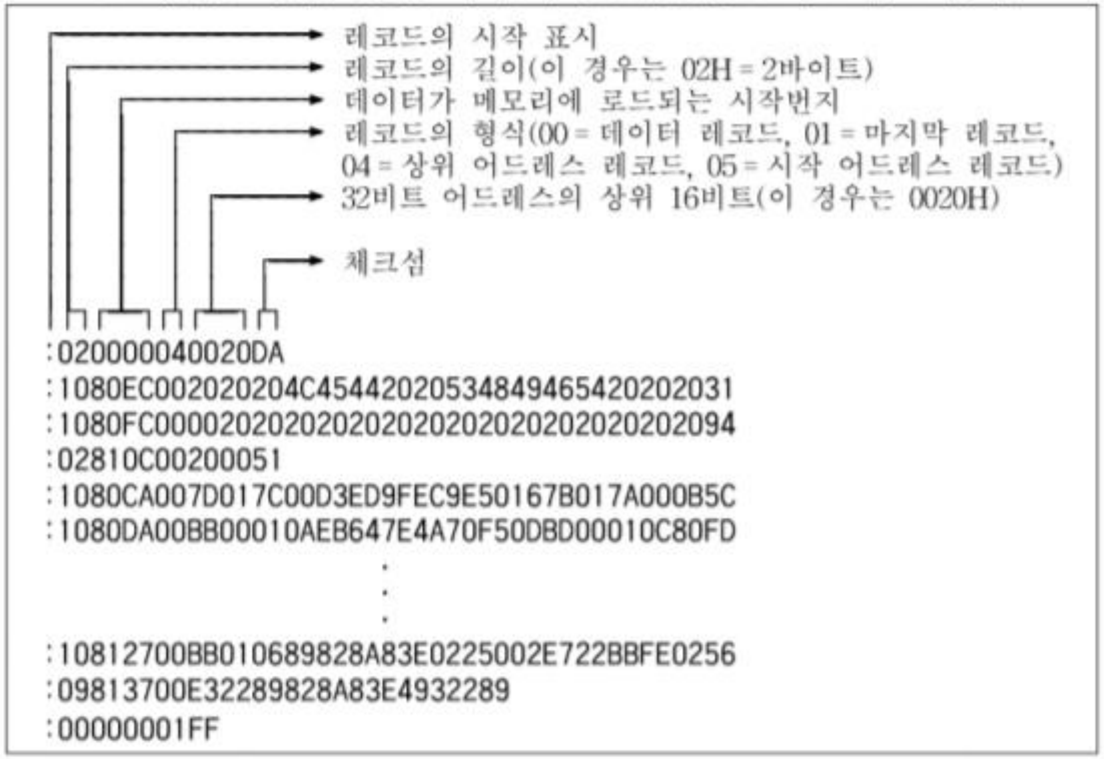
2. HEX파일



레코드의 시작 표시
레코드의 길이(이 경우는 10H = 16바이트)
데이터가 메모리에 로드되는 시작번지
레코드의 형식(00 = 데이터 레코드, 01 = 마지막 레코드)
데이터 내용
체크섬

```
:1080EC002020204C4544202053484946542020203 1
:1080FC0000202020202020202020202020202020 94
:02810C00200051
:1080CA007D017C00D3ED9FEC9E50167B017A000B5C
:1080DA00BB00010AEB647E4A70F50DBD00010C80FD
                    .
                    .
                    .
:10812700BB010689828A83E0225002E722BBFE0256
:09813700E32289828A83E4932289
:00000001FF
```

<그림 2> 인텔 HEX 파일의 구조(16비트 어드레스 형식)

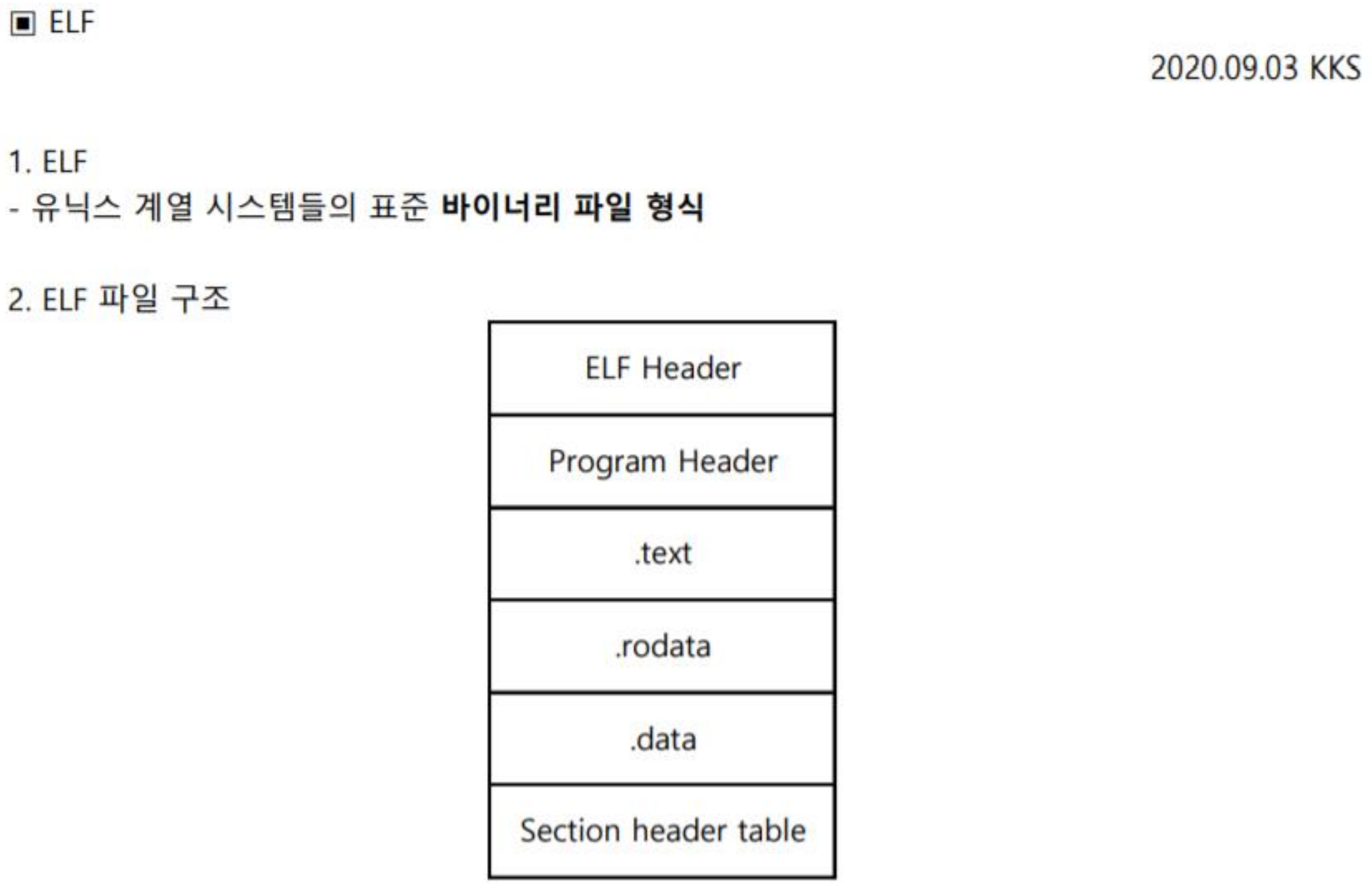


레코드의 시작 표시
레코드의 길이(이 경우는 02H = 2바이트)
데이터가 메모리에 로드되는 시작번지
레코드의 형식(00 = 데이터 레코드, 01 = 마지막 레코드,
04 = 상위 어드레스 레코드, 05 = 시작 어드레스 레코드)
32비트 어드레스의 상위 16비트(이 경우는 0020H)
체크섬

```
:020000040020DA
:1080EC002020204C45442020534849465420202031
:1080FC000020202020202020202020202020202094
:02810C00200051
:1080CA007D017C00D3ED9FEC9E50167B017A000B5C
:1080DA00BB00010AEB647E4A70F50DBD00010C80FD
                    .
                    .
                    .
:10812700BB010689828A83E0225002E722BBFE0256
:09813700E32289828A83E4932289
:00000001FF
```

<그림 3> 인텔 HEX 파일의 구조(32비트 선형 어드레스 형식)

2. ELF



▣ ELF

2020.09.03 KKS

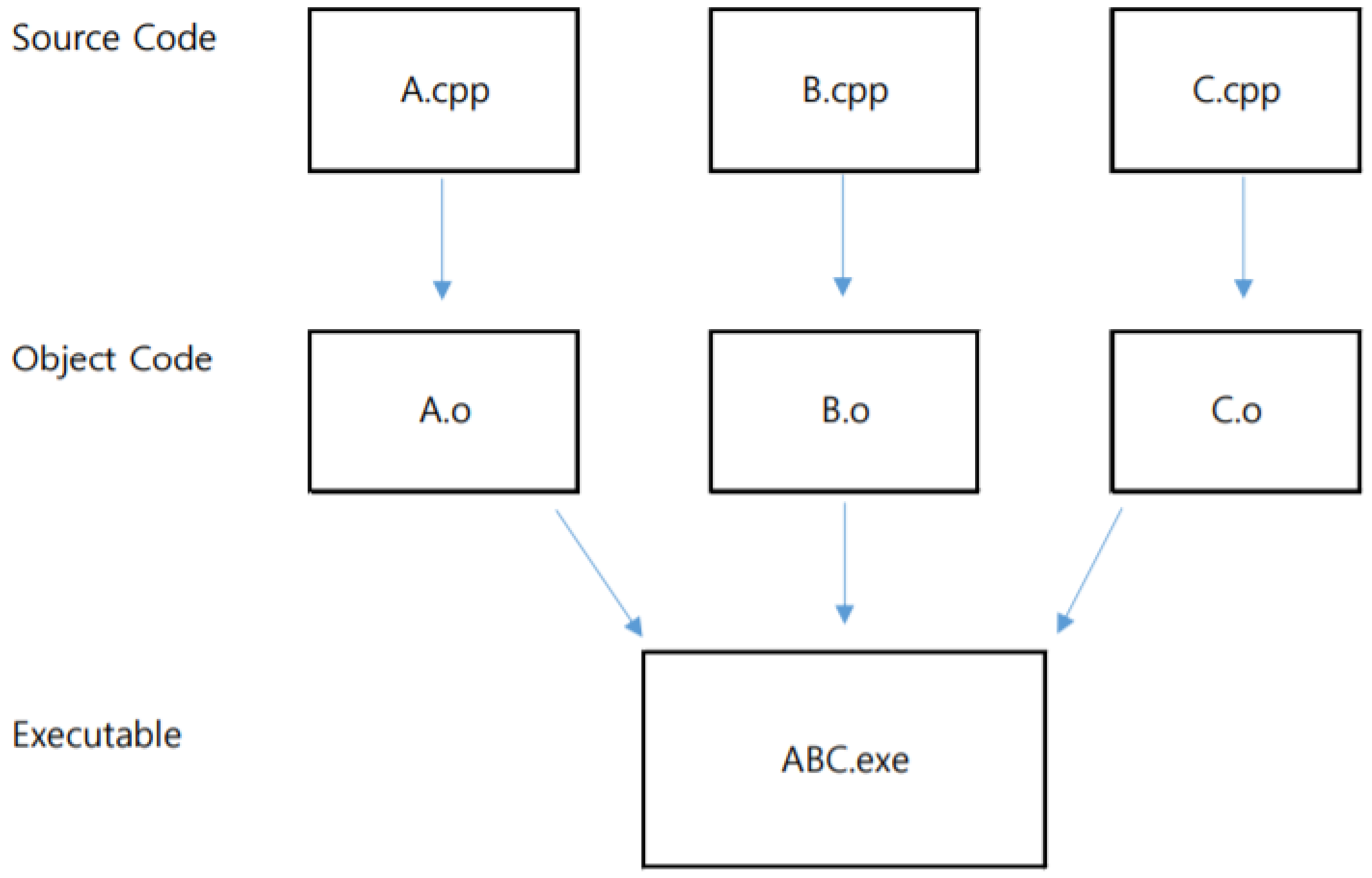
1. ELF
- 유닉스 계열 시스템들의 표준 **바이너리 파일 형식**

2. ELF 파일 구조

ELF Header
Program Header
.text
.rodata
.data
Section header table

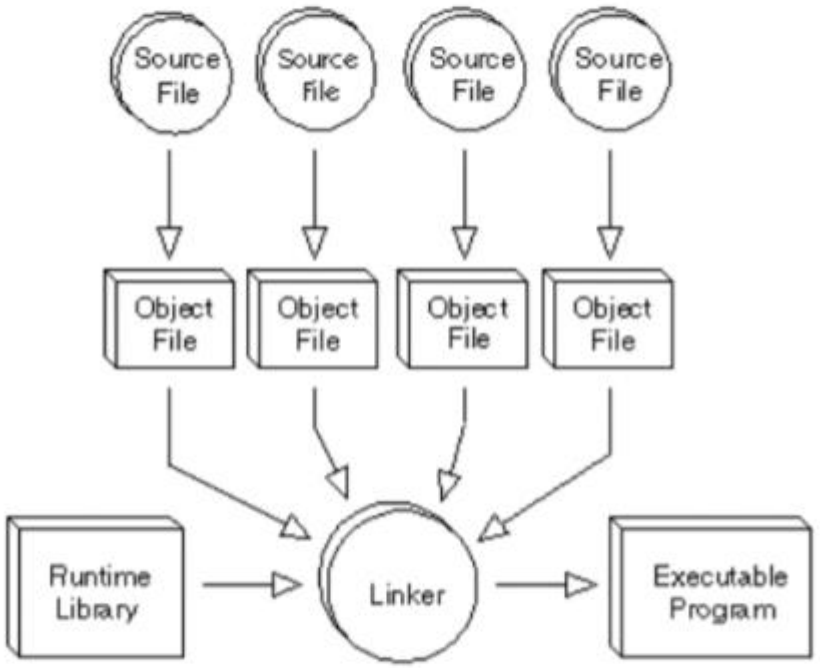
2. ELF

3. OBJECT & LINKING



2. ELF

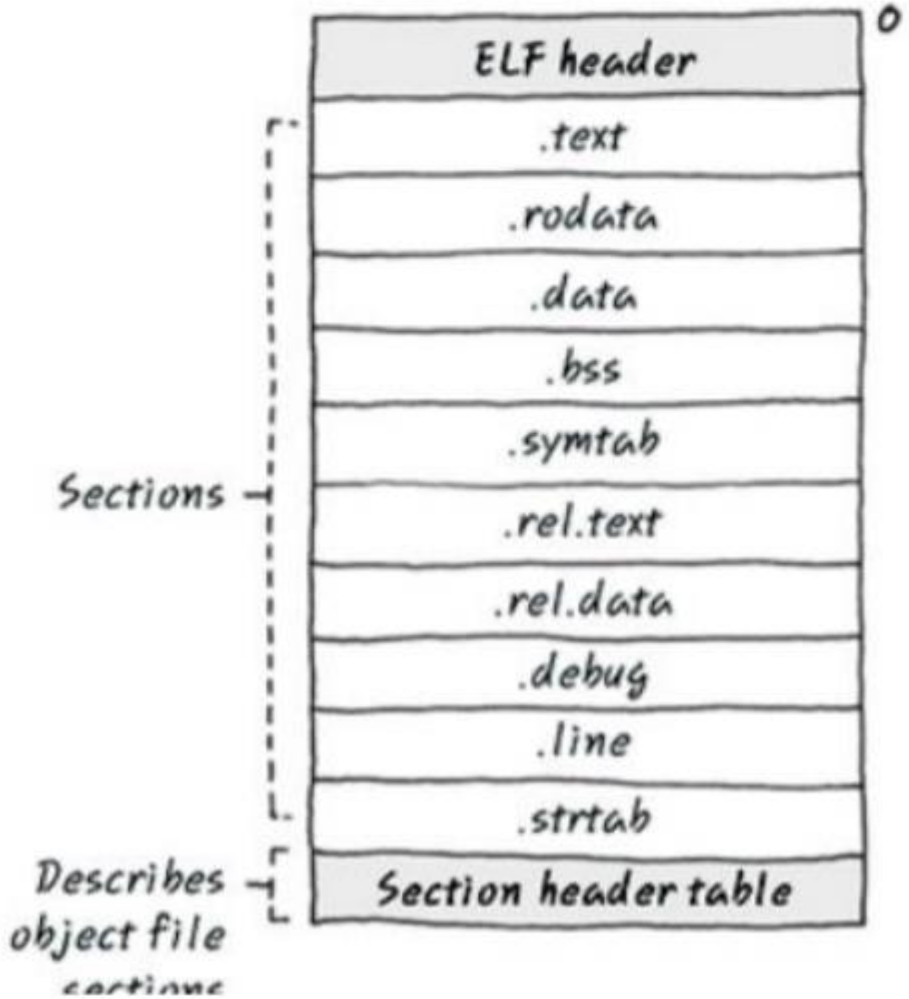
소스코드와 실행파일 중간에 컴파일을 하면 object파일이 생성됨



4. Object 파일 종류
- Relocatable(재배치 가능한 파일)
→ 다른 오브젝트 파일과 링크되어 실행파일과 공유 오브젝트 파일을 만들 수 있음
- Executable(실행파일)
→ 실행파일
- Shared Object File(공유 오브젝트 파일)
→ 다른 공유 오브젝트 파일이나 재배치 가능한 파일과 링크하여 새 오브젝트 파일 생성

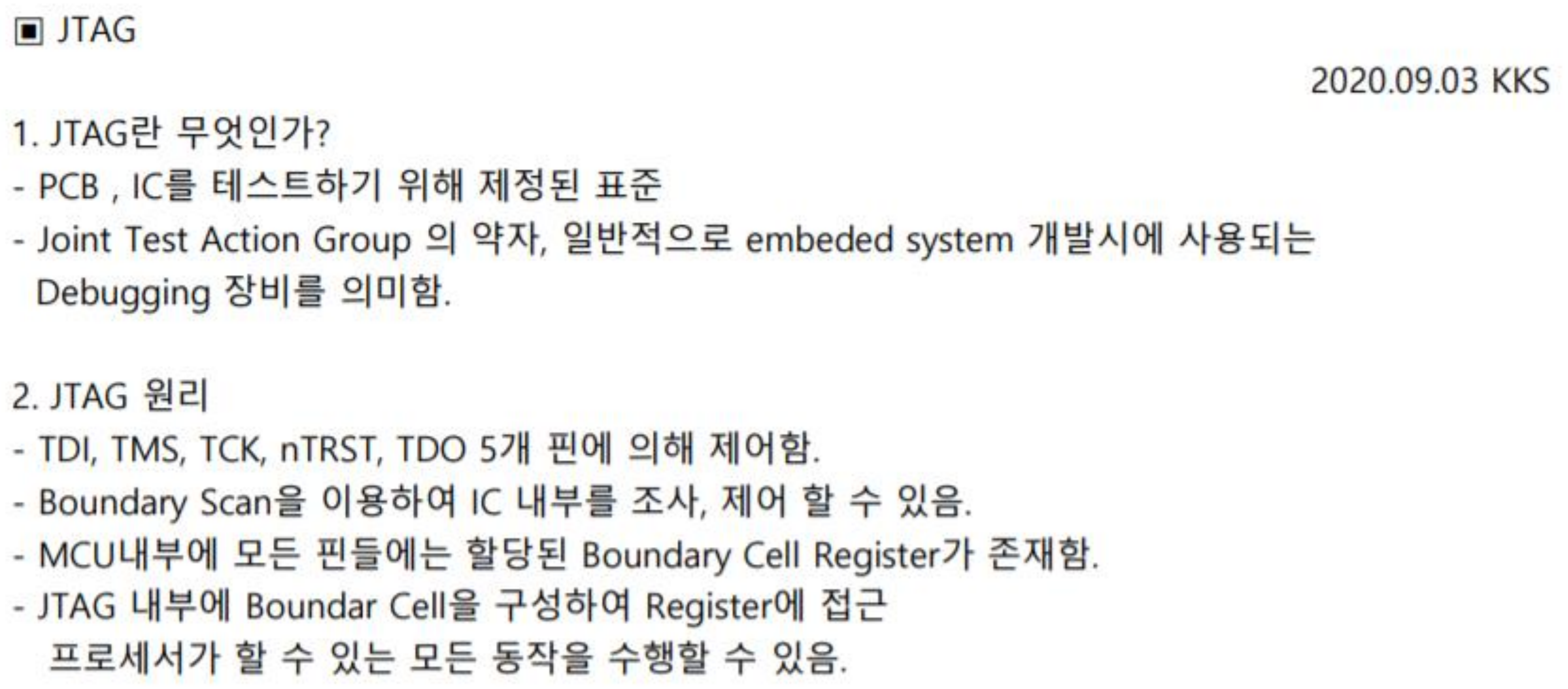
2. ELF

5. elf 파일구조



- ELF HEADER
ELF 파일 포맷임을 표시
실행되는 CPU정보
- rel.text : 각 머신코드의 위치
- rodata : read - only data
    즉 const 변수
- data : 초기화된 전역변수
- bss : 초기화되지 않아 0으로
    초기화되는 전역변수
- symtab : symbol table
    이름을 가지는 단위
    전역변수이름, 함수이름
-section header : 섹션의 이름
섹션의 크기 정보를 포함

3. JTAG



▣ JTAG

2020.09.03 KKS

1. JTAG란 무엇인가?
- PCB , IC를 테스트하기 위해 제정된 표준
- Joint Test Action Group 의 약자, 일반적으로 embeded system 개발시에 사용되는 Debugging 장비를 의미함.

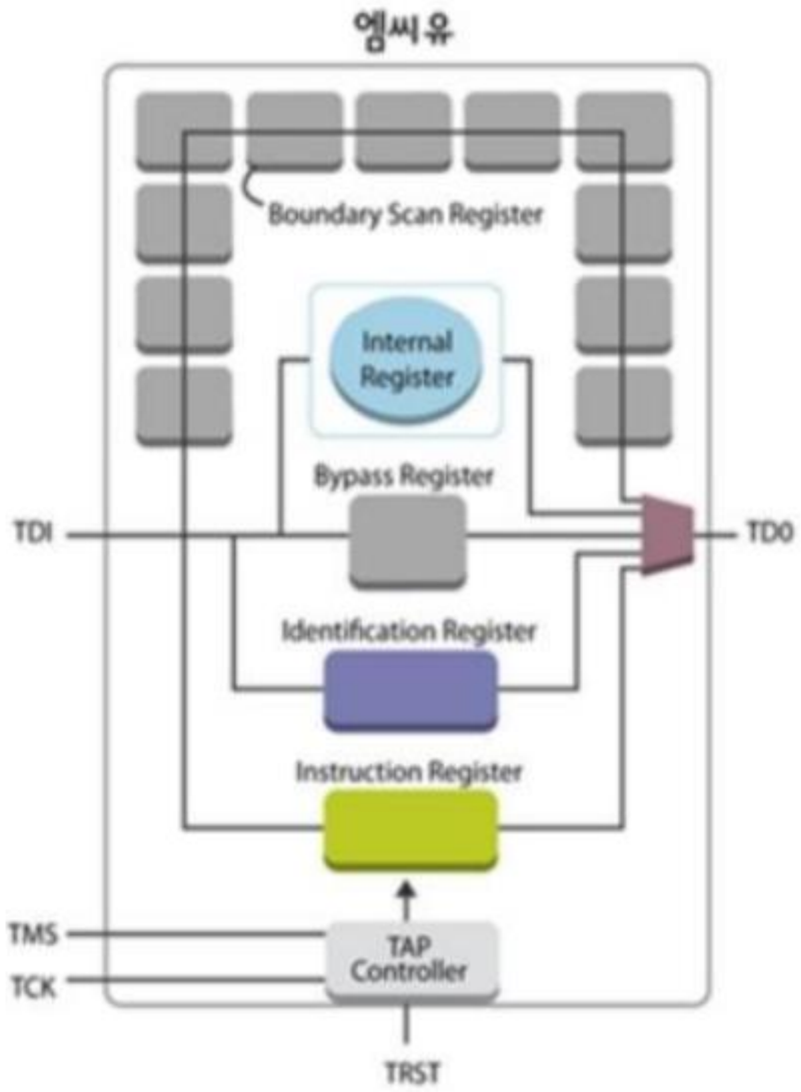
2. JTAG 원리
- TDI, TMS, TCK, nTRST, TDO 5개 핀에 의해 제어함.
- Boundary Scan을 이용하여 IC 내부를 조사, 제어 할 수 있음.
- MCU내부에 모든 핀들에는 할당된 Boundary Cell Register가 존재함.
- JTAG 내부에 Boundar Cell을 구성하여 Register에 접근 프로세서가 할 수 있는 모든 동작을 수행할 수 있음.

3. JTAG

3. Flash & JTAG
- 최초에 Bootloader를 올리는 작업을 하며, 이후 플래쉬 메모리 내용 변경 가능

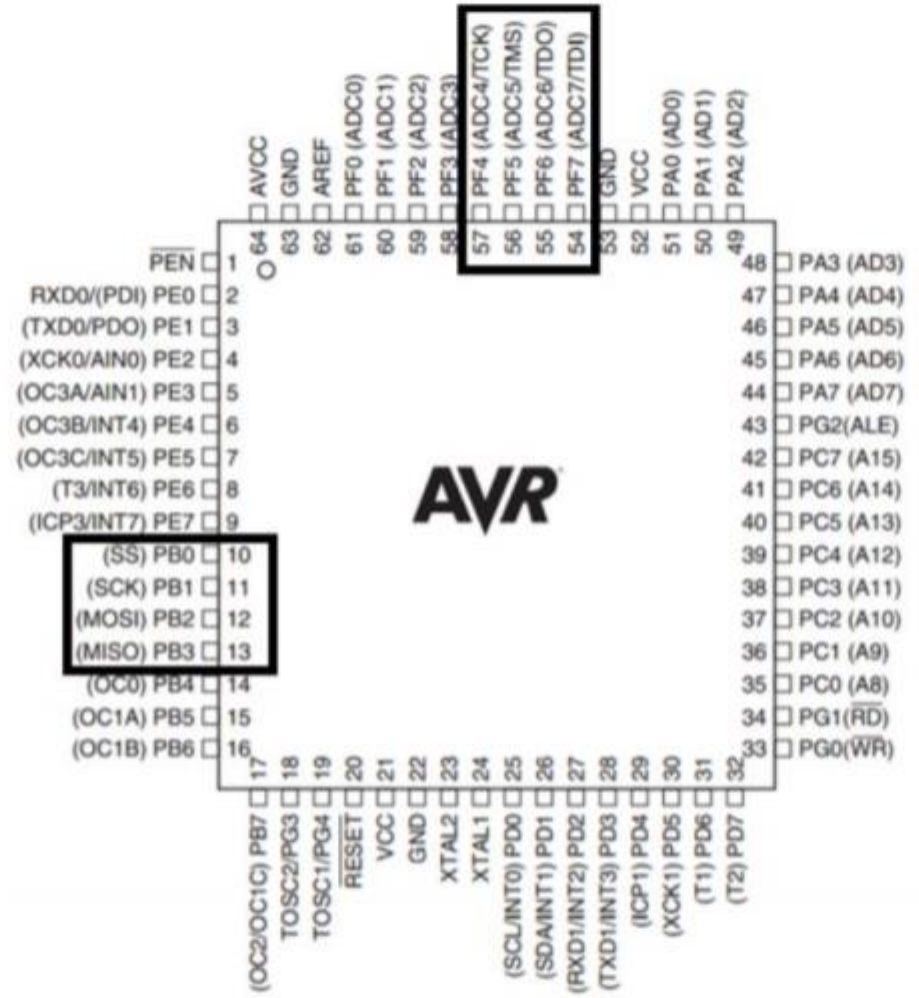
4. 세부동작
- TDI : input 신호 담당
- TDO : Output신호 담당
- TMS : 모드 선택
- TCK : 클럭
- TRST : 리셋

←핀들의 신호를 조합하는 명령어를 JTAG 하드웨어 디버깅 장비측면에서 제공함.



3. JTAG

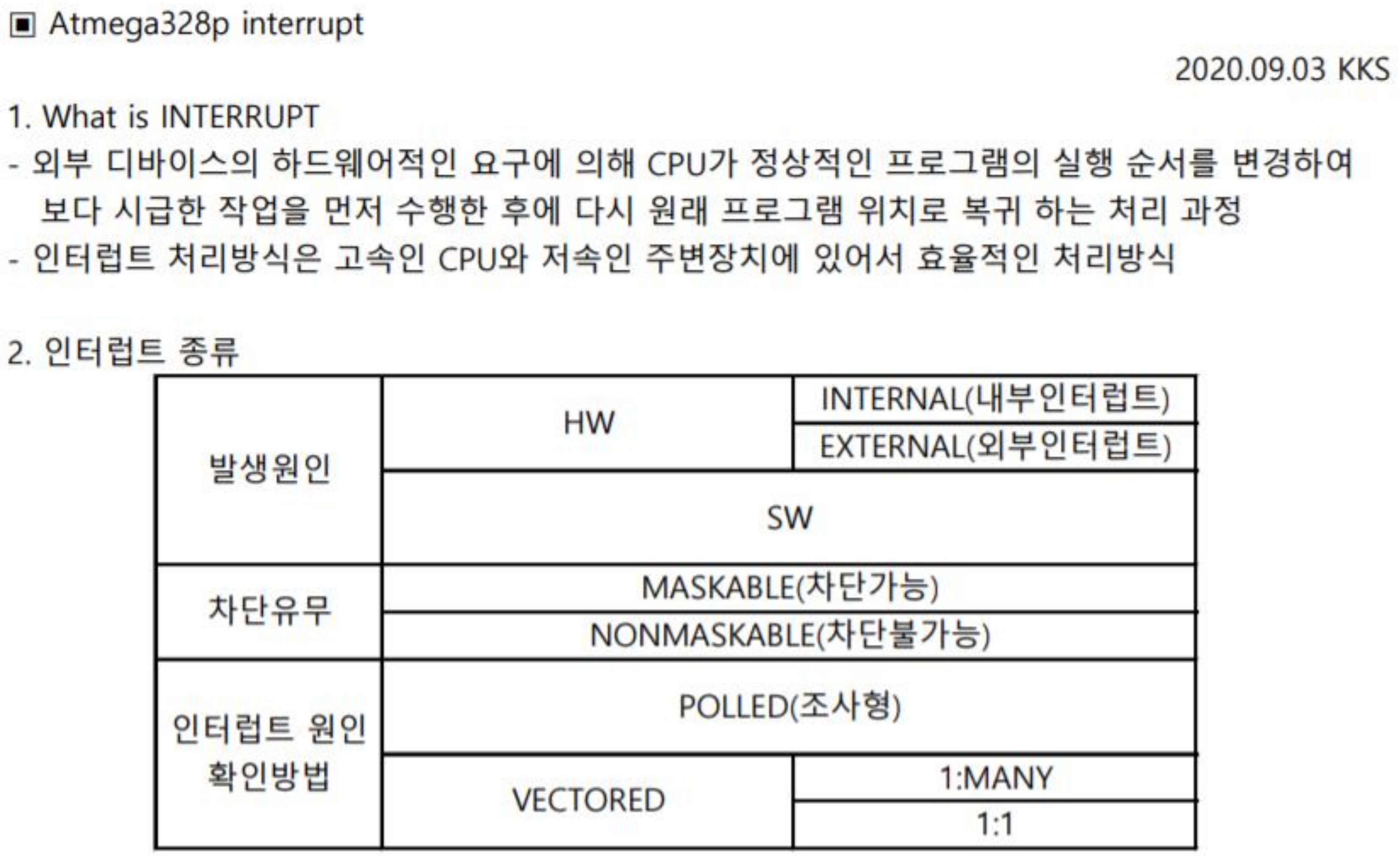
5. JTAG IN AVR



ATMEGA128기준
- 54~57 JTAG핀
- 10~13 ISP핀

ATEMGA328p JTAG지원 안됨

4. ATMEGA 328P INTERRUPT



▣ Atmega328p interrupt

2020.09.03 KKS

1. What is INTERRUPT

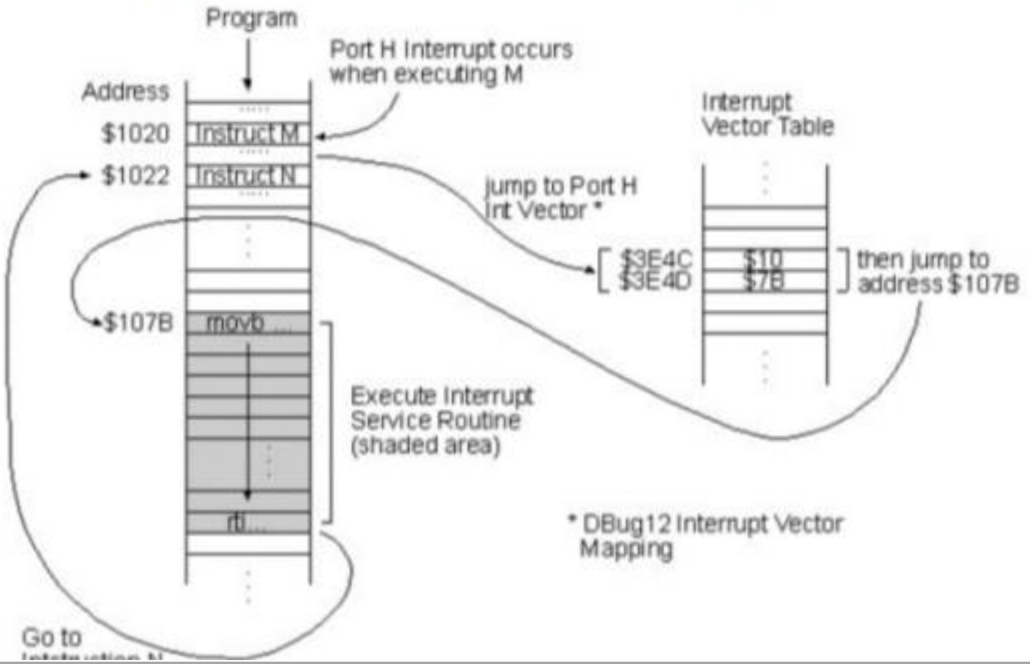
- 외부 디바이스의 하드웨어적인 요구에 의해 CPU가 정상적인 프로그램의 실행 순서를 변경하여 보다 시급한 작업을 먼저 수행한 후에 다시 원래 프로그램 위치로 복귀 하는 처리 과정
- 인터럽트 처리방식은 고속인 CPU와 저속인 주변장치에 있어서 효율적인 처리방식

2. 인터럽트 종류

발생원인	HW	INTERNAL(내부인터럽트)
		EXTERNAL(외부인터럽트)
	SW	
차단유무	MASKABLE(차단가능)	
	NONMASKABLE(차단불가능)	
인터럽트 원인 확인방법	POLLED(조사형)	
	VECTORED	1:MANY
		1:1

4. ATMEGA 328P INTERRUPT

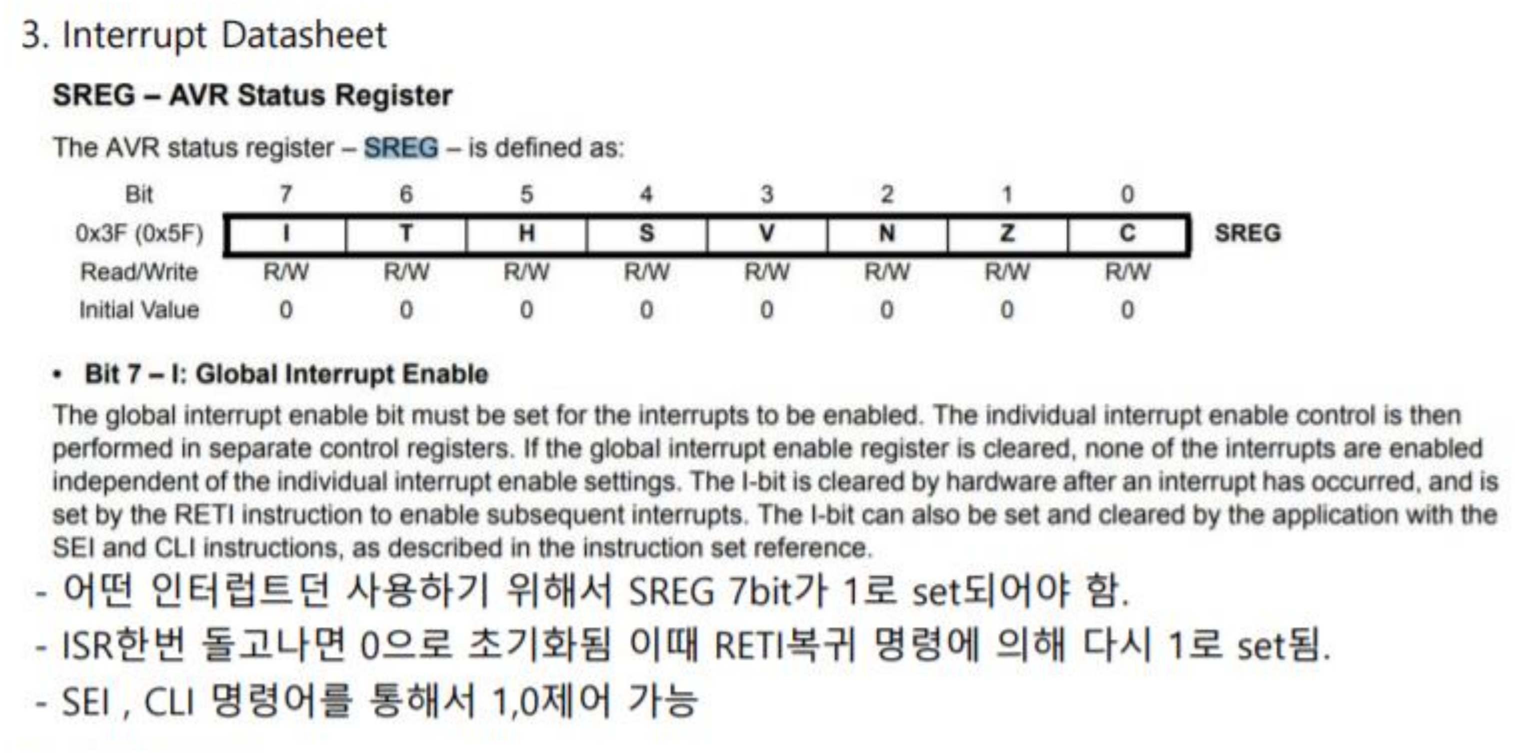
- INTERNAL : 정의되지 않는 명령 실행, 0으로 나눗셈 실행, 메모리 보호 등
  Atmega328에서는 사용하지 않음.
- EXTERNAL : 타이머 시간경과에의한 인터럽트, 입력장치에서의 입력 요청
  출력장치에서의 출력 완료 등 외부 장치에 의해 발생하는 인터럽트
- SW : 소프트웨어상에서 발생하는 인터럽트
- MASKABLE : 차단가능한 인터럽트 Atmega328p 모든 인터럽트가 여기에 해당
- NON-MASKABLE : 긴급한 환경(비상정지 등) 에서의 발생 인터럽트
- POLLED : 인터럽트 원인을 CPU가 소프트웨어적으로 하나하나 확인하여 찾아 내려감
- VECTORED : 인터럽트 발생 주변장치가 CPU에게 벡터를 전송
  CPU는 전송받은 벡터를 이용하여 ISR시작번지 결정
- 1:many : 한 인터럽트 포트에 여러 개 장치 연결(장치는 Status Register 를 확인)
- 1:1 : 한 인터럽트 포트에 한 개의 장치 연결



Program
Port H Interrupt occurs when executing M
Address
$1020
Instruct M
$1022
Instruct N
Interrupt Vector Table
jump to Port H Int Vector *
$3E4C
$3E4D
$10
$7B
then jump to address $107B
$107B
movb ...
Execute Interrupt Service Routine (shaded area)
rti...
* DBug12 Interrupt Vector Mapping
Go to

즉 인터럽트 발생시
STACK에 복귀 주소 PUSH
이후 해당 VECTOR TABLE
주소로 JUMP하여 실행

4. ATMEGA 328P INTERRUPT



3. Interrupt Datasheet

SREG – AVR Status Register

The AVR status register – SREG – is defined as:

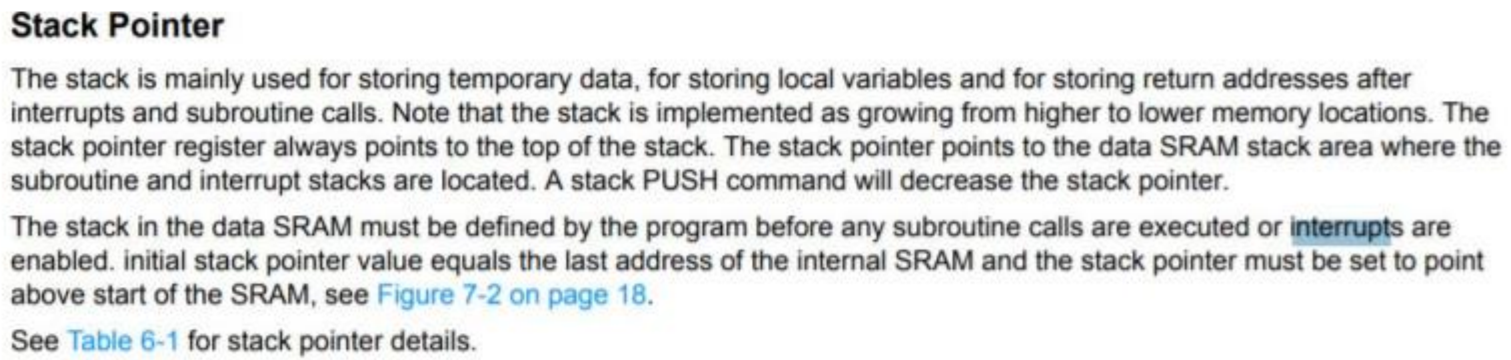
Bit	7	6	5	4	3	2	1	0	
0x3F (0x5F)	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – I: Global Interrupt Enable**

The global interrupt enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the global interrupt enable register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

- 어떤 인터럽트던 사용하기 위해서 SREG 7bit가 1로 set되어야 함.
- ISR한번 돌고나면 0으로 초기화됨 이때 RETI복귀 명령에 의해 다시 1로 set됨.
- SEI , CLI 명령어를 통해서 1,0제어 가능

4. ATMEGA 328P INTERRUPT



Stack Pointer

The stack is mainly used for storing temporary data, for storing local variables and for storing return addresses after interrupts and subroutine calls. Note that the stack is implemented as growing from higher to lower memory locations. The stack pointer register always points to the top of the stack. The stack pointer points to the data SRAM stack area where the subroutine and interrupt stacks are located. A stack PUSH command will decrease the stack pointer.

The stack in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. initial stack pointer value equals the last address of the internal SRAM and the stack pointer must be set to point above start of the SRAM, see Figure 7-2 on page 18.

See Table 6-1 for stack pointer details.

- ISR(interrupt Service Routine) 호출되면 복귀주소가 stack에 저장됨

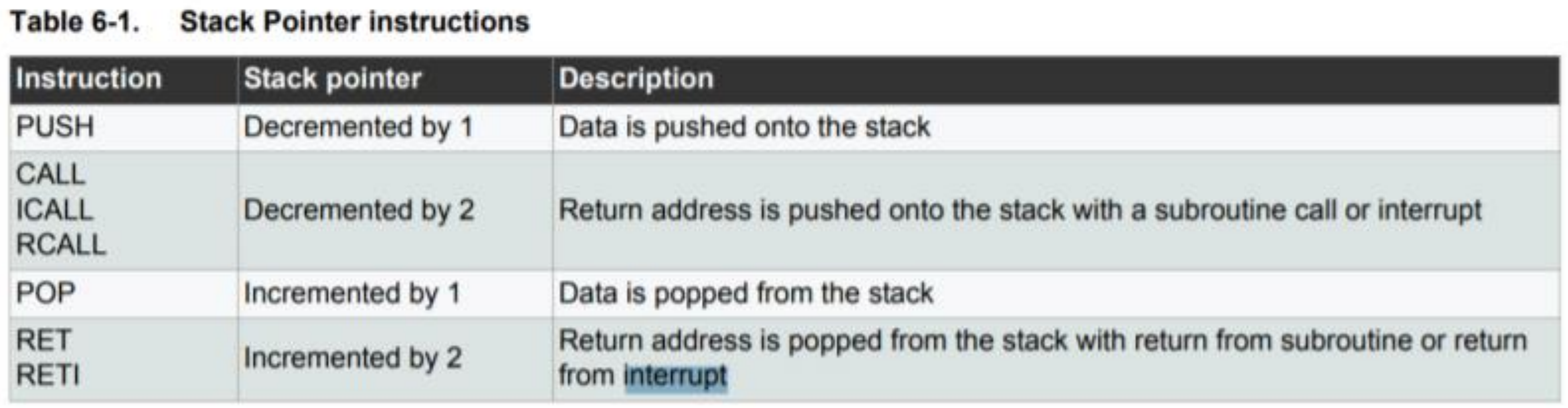
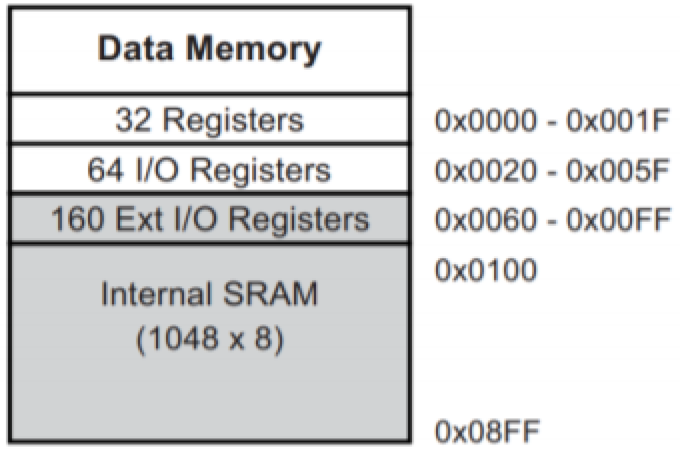


Table 6-1. Stack Pointer instructions

Instruction	Stack pointer	Description
PUSH	Decremented by 1	Data is pushed onto the stack
CALL ICALL RCALL	Decremented by 2	Return address is pushed onto the stack with a subroutine call or interrupt
POP	Incremented by 1	Data is popped from the stack
RET RETI	Incremented by 2	Return address is popped from the stack with return from subroutine or return from interrupt

4. ATMEGA 328P INTERRUPT

INSTRUCTION 명령어 RETI를 통해 ISR혹은 subroutine에서 복귀됨을 확인 할 수 있음.



Stack이 쌓여 나아가는 방향은

HIGH TO LOW

따라서 위 DATASHEET대로라면 0x08FF부터

STACK FRAME이 형성됨

4. ATMEGA 328P INTERRUPT

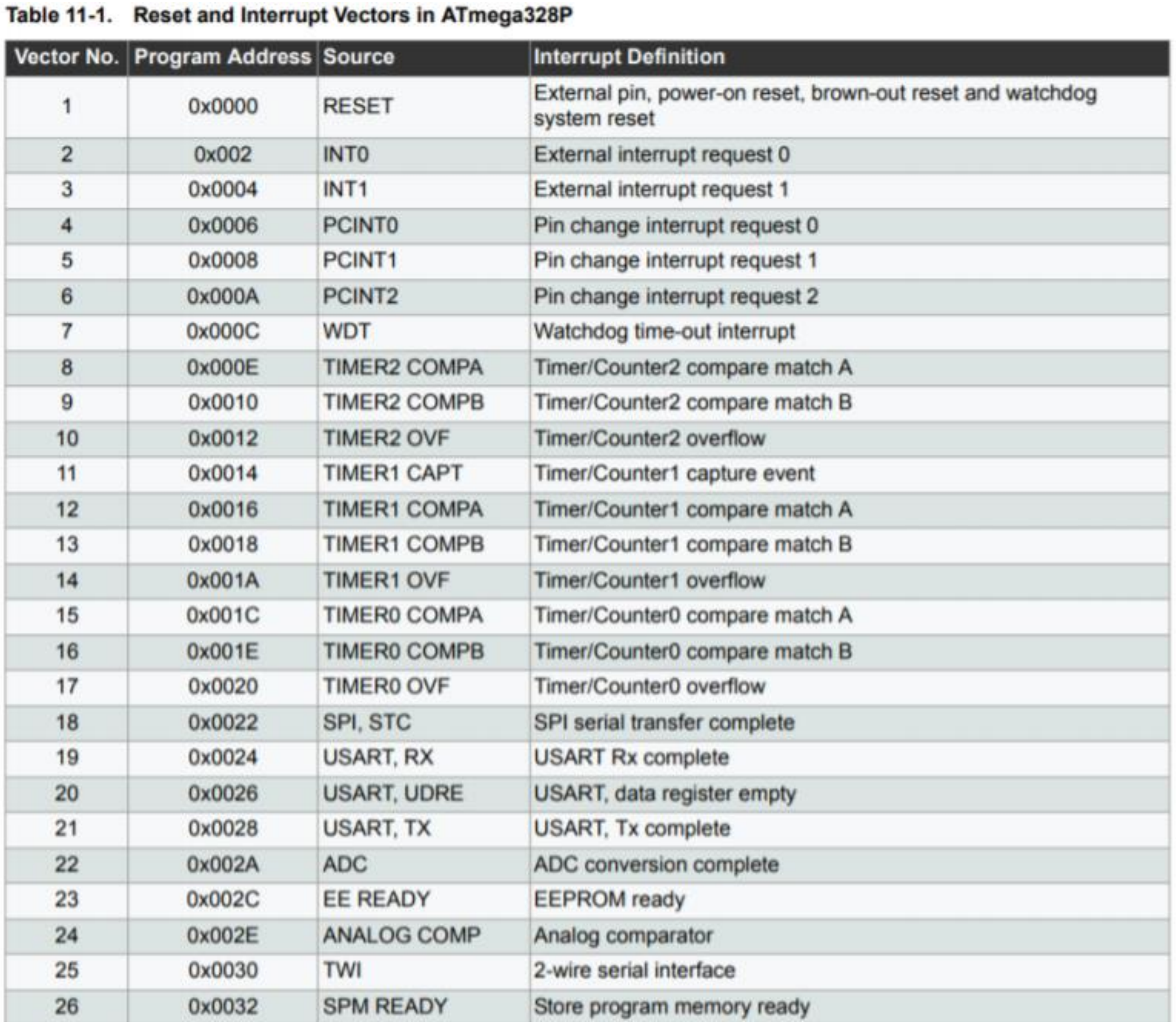
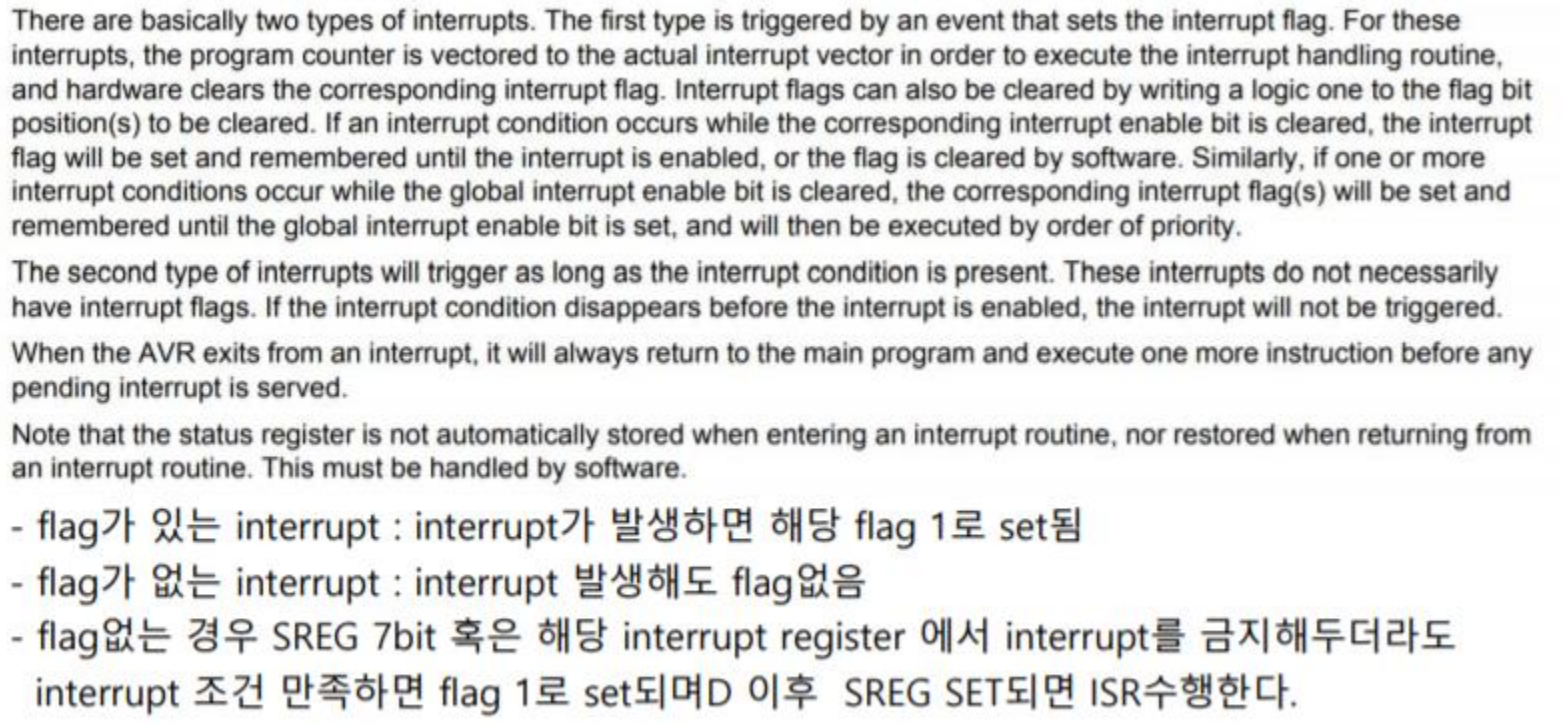


Table 11-1. Reset and Interrupt Vectors in ATmega328P

Vector No.	Program Address	Source	Interrupt Definition
1	0x0000	RESET	External pin, power-on reset, brown-out reset and watchdog system reset
2	0x002	INT0	External interrupt request 0
3	0x0004	INT1	External interrupt request 1
4	0x0006	PCINT0	Pin change interrupt request 0
5	0x0008	PCINT1	Pin change interrupt request 1
6	0x000A	PCINT2	Pin change interrupt request 2
7	0x000C	WDT	Watchdog time-out interrupt
8	0x000E	TIMER2 COMPA	Timer/Counter2 compare match A
9	0x0010	TIMER2 COMPB	Timer/Counter2 compare match B
10	0x0012	TIMER2 OVF	Timer/Counter2 overflow
11	0x0014	TIMER1 CAPT	Timer/Counter1 capture event
12	0x0016	TIMER1 COMPA	Timer/Counter1 compare match A
13	0x0018	TIMER1 COMPB	Timer/Counter1 compare match B
14	0x001A	TIMER1 OVF	Timer/Counter1 overflow
15	0x001C	TIMER0 COMPA	Timer/Counter0 compare match A
16	0x001E	TIMER0 COMPB	Timer/Counter0 compare match B
17	0x0020	TIMER0 OVF	Timer/Counter0 overflow
18	0x0022	SPI, STC	SPI serial transfer complete
19	0x0024	USART, RX	USART Rx complete
20	0x0026	USART, UDRE	USART, data register empty
21	0x0028	USART, TX	USART, Tx complete
22	0x002A	ADC	ADC conversion complete
23	0x002C	EE READY	EEPROM ready
24	0x002E	ANALOG COMP	Analog comparator
25	0x0030	TWI	2-wire serial interface
26	0x0032	SPM READY	Store program memory ready

4. ATMEGA 328P INTERRUPT



There are basically two types of interrupts. The first type is triggered by an event that sets the interrupt flag. For these interrupts, the program counter is vectored to the actual interrupt vector in order to execute the interrupt handling routine, and hardware clears the corresponding interrupt flag. Interrupt flags can also be cleared by writing a logic one to the flag bit position(s) to be cleared. If an interrupt condition occurs while the corresponding interrupt enable bit is cleared, the interrupt flag will be set and remembered until the interrupt is enabled, or the flag is cleared by software. Similarly, if one or more interrupt conditions occur while the global interrupt enable bit is cleared, the corresponding interrupt flag(s) will be set and remembered until the global interrupt enable bit is set, and will then be executed by order of priority.

The second type of interrupts will trigger as long as the interrupt condition is present. These interrupts do not necessarily have interrupt flags. If the interrupt condition disappears before the interrupt is enabled, the interrupt will not be triggered.

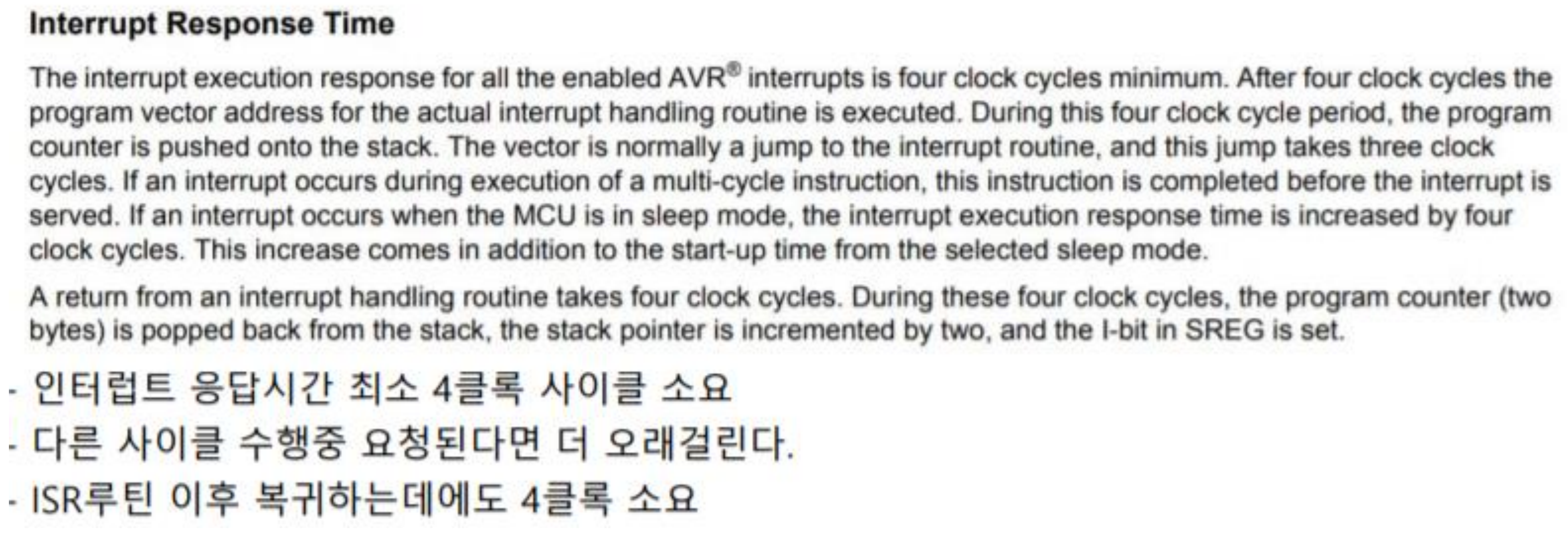
When the AVR exits from an interrupt, it will always return to the main program and execute one more instruction before any pending interrupt is served.

Note that the status register is not automatically stored when entering an interrupt routine, nor restored when returning from an interrupt routine. This must be handled by software.

- flag가 있는 interrupt : interrupt가 발생하면 해당 flag 1로 set됨
- flag가 없는 interrupt : interrupt 발생해도 flag없음
- flag없는 경우 SREG 7bit 혹은 해당 interrupt register 에서 interrupt를 금지해두더라도 interrupt 조건 만족하면 flag 1로 set되며D 이후 SREG SET되면 ISR수행한다.

포기하면 얻는 건 아무것도 없다.

4. ATMEGA 328P INTERRUPT



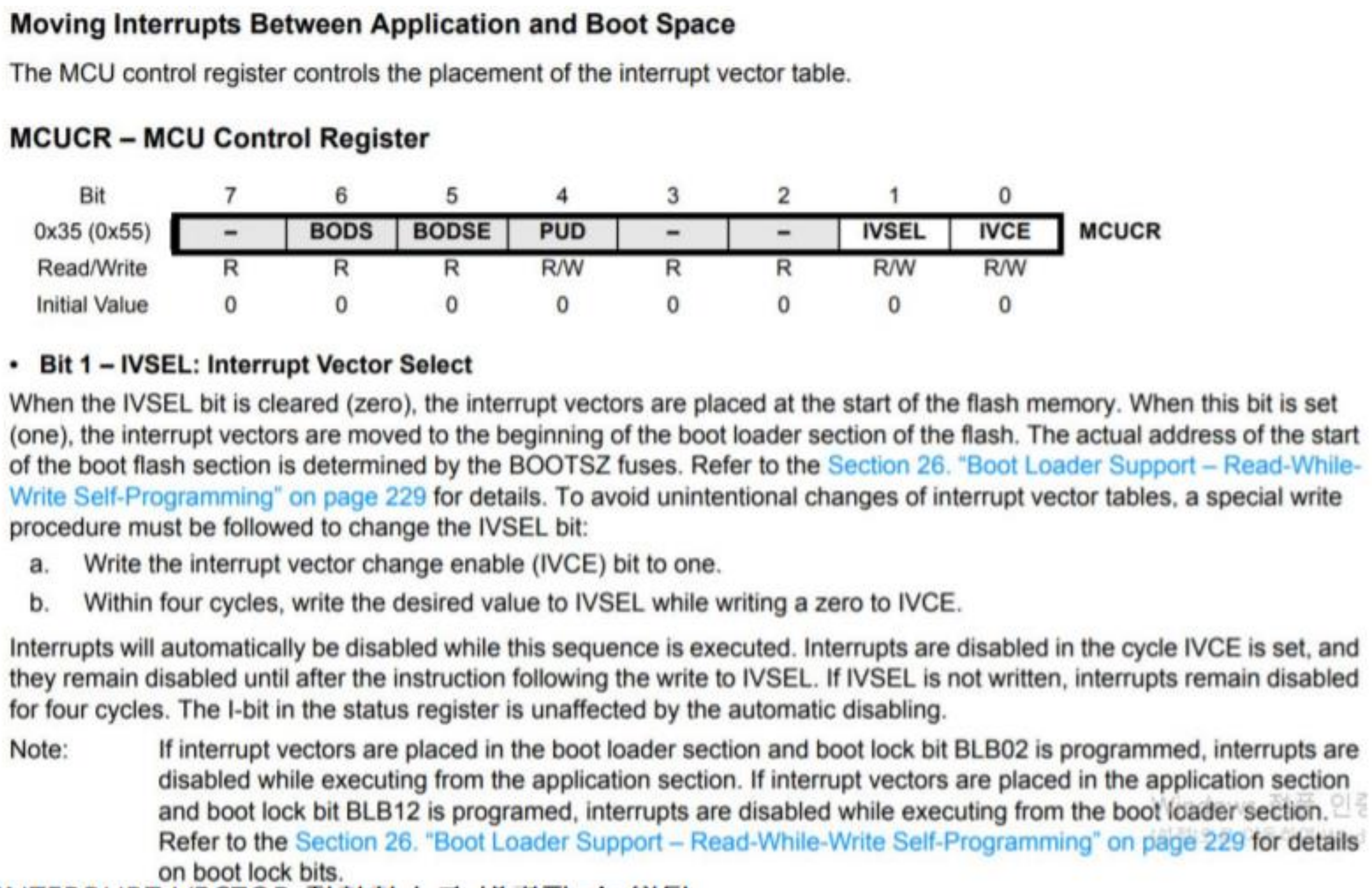
Interrupt Response Time

The interrupt execution response for all the enabled AVR® interrupts is four clock cycles minimum. After four clock cycles the program vector address for the actual interrupt handling routine is executed. During this four clock cycle period, the program counter is pushed onto the stack. The vector is normally a jump to the interrupt routine, and this jump takes three clock cycles. If an interrupt occurs during execution of a multi-cycle instruction, this instruction is completed before the interrupt is served. If an interrupt occurs when the MCU is in sleep mode, the interrupt execution response time is increased by four clock cycles. This increase comes in addition to the start-up time from the selected sleep mode.

A return from an interrupt handling routine takes four clock cycles. During these four clock cycles, the program counter (two bytes) is popped back from the stack, the stack pointer is incremented by two, and the I-bit in SREG is set.

- 인터럽트 응답시간 최소 4클록 사이클 소요
- 다른 사이클 수행중 요청된다면 더 오래걸린다.
- ISR루틴 이후 복귀하는데에도 4클록 소요

4. ATMEGA 328P INTERRUPT



Moving Interrupts Between Application and Boot Space

The MCU control register controls the placement of the interrupt vector table.

MCUCR – MCU Control Register

Bit	7	6	5	4	3	2	1	0	
0x35 (0x55)	–	BODS	BODSE	PUD	–	–	IVSEL	IVCE	MCUCR
Read/Write	R	R	R	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 1 – IVSEL: Interrupt Vector Select**

When the IVSEL bit is cleared (zero), the interrupt vectors are placed at the start of the flash memory. When this bit is set (one), the interrupt vectors are moved to the beginning of the boot loader section of the flash. The actual address of the start of the boot flash section is determined by the BOOTSZ fuses. Refer to the Section 26. "Boot Loader Support – Read-While-Write Self-Programming" on page 229 for details. To avoid unintentional changes of interrupt vector tables, a special write procedure must be followed to change the IVSEL bit:

a. Write the interrupt vector change enable (IVCE) bit to one.

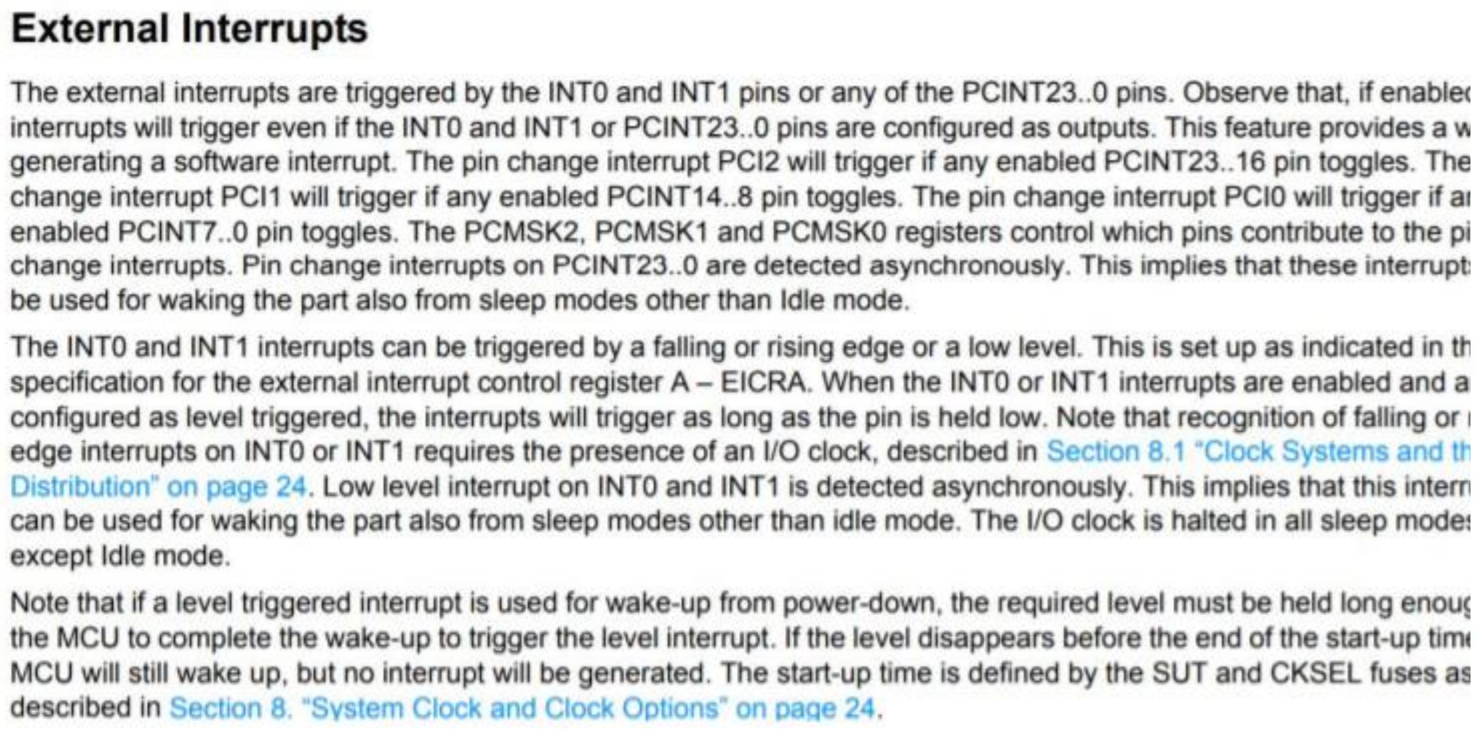
b. Within four cycles, write the desired value to IVSEL while writing a zero to IVCE.

Interrupts will automatically be disabled while this sequence is executed. Interrupts are disabled in the cycle IVCE is set, and they remain disabled until after the instruction following the write to IVSEL. If IVSEL is not written, interrupts remain disabled for four cycles. The I-bit in the status register is unaffected by the automatic disabling.

Note: If interrupt vectors are placed in the boot loader section and boot lock bit BLB02 is programmed, interrupts are disabled while executing from the application section. If interrupt vectors are placed in the application section and boot lock bit BLB12 is programed, interrupts are disabled while executing from the boot loader section. Refer to the Section 26. "Boot Loader Support – Read-While-Write Self-Programming" on page 229 for details on boot lock bits.

- INTERRUPT VECTOR 저상상소를 변경할 수 있다.
- 하지만 ATMEGA328에서는 지원되지 않는다.

4. ATMEGA 328P INTERRUPT



External Interrupts

The external interrupts are triggered by the INT0 and INT1 pins or any of the PCINT23..0 pins. Observe that, if enablec
interrupts will trigger even if the INT0 and INT1 or PCINT23..0 pins are configured as outputs. This feature provides a w
generating a software interrupt. The pin change interrupt PCI2 will trigger if any enabled PCINT23..16 pin toggles. The
change interrupt PCI1 will trigger if any enabled PCINT14..8 pin toggles. The pin change interrupt PCI0 will trigger if ar
enabled PCINT7..0 pin toggles. The PCMSK2, PCMSK1 and PCMSK0 registers control which pins contribute to the pi
change interrupts. Pin change interrupts on PCINT23..0 are detected asynchronously. This implies that these interrupt
be used for waking the part also from sleep modes other than Idle mode.

The INT0 and INT1 interrupts can be triggered by a falling or rising edge or a low level. This is set up as indicated in th
specification for the external interrupt control register A – EICRA. When the INT0 or INT1 interrupts are enabled and a
configured as level triggered, the interrupts will trigger as long as the pin is held low. Note that recognition of falling or
edge interrupts on INT0 or INT1 requires the presence of an I/O clock, described in Section 8.1 "Clock Systems and th
Distribution" on page 24. Low level interrupt on INT0 and INT1 is detected asynchronously. This implies that this interr
can be used for waking the part also from sleep modes other than idle mode. The I/O clock is halted in all sleep modes
except Idle mode.

Note that if a level triggered interrupt is used for wake-up from power-down, the required level must be held long enoug
the MCU to complete the wake-up to trigger the level interrupt. If the level disappears before the end of the start-up time
MCU will still wake up, but no interrupt will be generated. The start-up time is defined by the SUT and CKSEL fuses as
described in Section 8. "System Clock and Clock Options" on page 24.

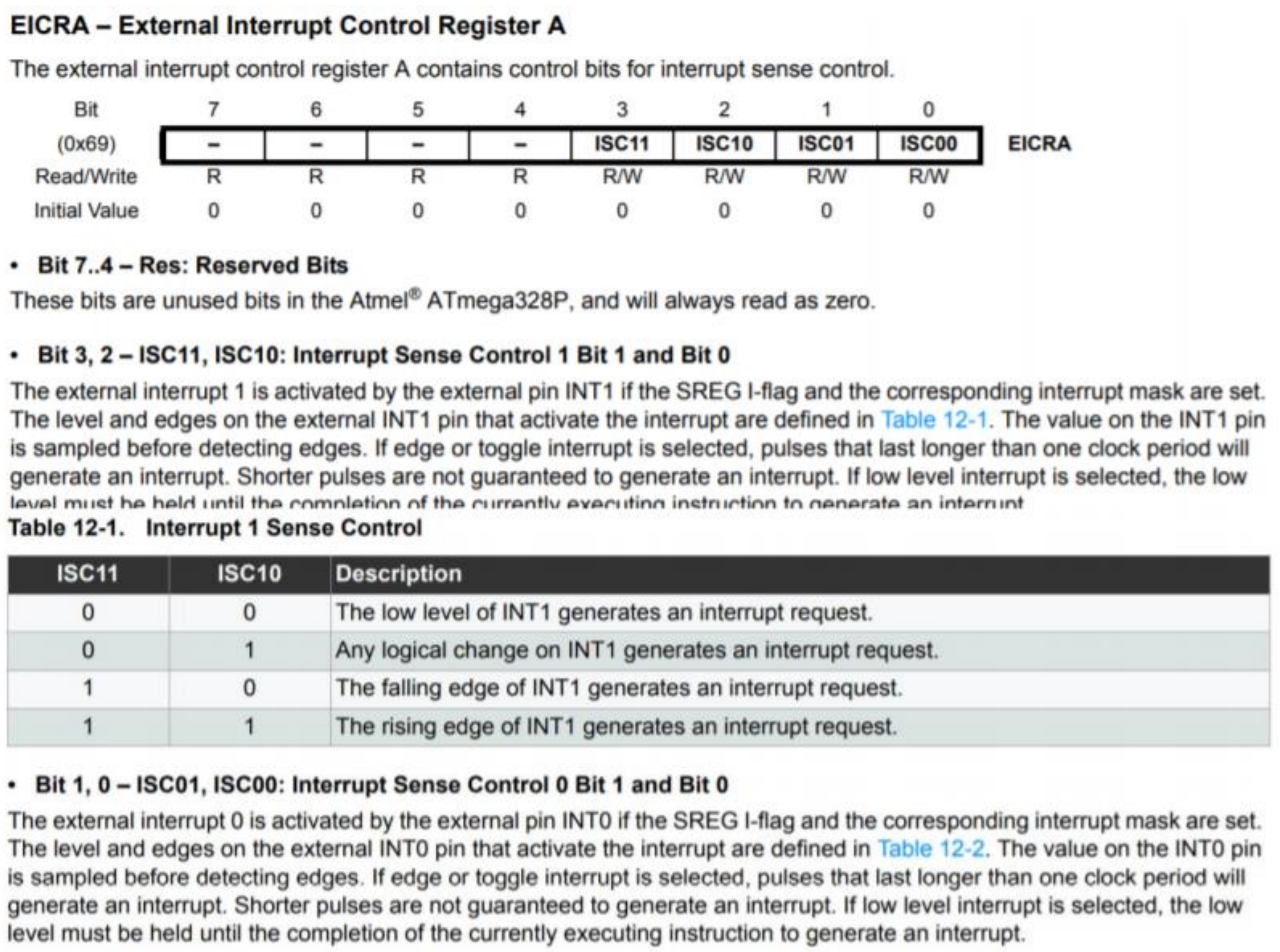
- INT0,INT1,PCINT23..0이 OUTPUT이어도 인터럽트가 활성화 돼 있으면 동작함
- 핀인터럽트 PC12,1,0은 PCINT23..16 14..8,7..0 토글시 발생함.

-PCMSK2,1,0 레지스터를 통해 어떤핀이 Pin change Interrupt에 기여할지 결정함
- INTP0,1은 입력되는 신호 0 또는 상승엣지, 하강엣지에 의하여 활성화
- INTP는 EICRA레지스터를 통해 설정
- 인터럽트 레벨트리거 방식으로 설정시 해당핀에 L입력되는 동안 인터럽트 발생
- 에지트리거 방식으로 설정시 I/O클럭이 필요하므로 I/O클록 차단되는 아이들모드 이외의 SLEEP모드 사용불가

-IDLE모드 이외의 SLEEP모드는 I/O사용 불가능 하기 때문이다

포기하면 얻는 건 아무것도 없다.

4. ATMEGA 328P INTERRUPT



EICRA – External Interrupt Control Register A

The external interrupt control register A contains control bits for interrupt sense control.

Bit	7	6	5	4	3	2	1	0	
(0x69)	–	–	–	–	ISC11	ISC10	ISC01	ISC00	EICRA
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..4 – Res: Reserved Bits**

These bits are unused bits in the Atmel® ATmega328P, and will always read as zero.

- **Bit 3, 2 – ISC11, ISC10: Interrupt Sense Control 1 Bit 1 and Bit 0**

The external interrupt 1 is activated by the external pin INT1 if the SREG I-flag and the corresponding interrupt mask are set. The level and edges on the external INT1 pin that activate the interrupt are defined in Table 12-1. The value on the INT1 pin is sampled before detecting edges. If edge or toggle interrupt is selected, pulses that last longer than one clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt.

Table 12-1. Interrupt 1 Sense Control

ISC11	ISC10	Description
0	0	The low level of INT1 generates an interrupt request.
0	1	Any logical change on INT1 generates an interrupt request.
1	0	The falling edge of INT1 generates an interrupt request.
1	1	The rising edge of INT1 generates an interrupt request.

- **Bit 1, 0 – ISC01, ISC00: Interrupt Sense Control 0 Bit 1 and Bit 0**

The external interrupt 0 is activated by the external pin INT0 if the SREG I-flag and the corresponding interrupt mask are set. The level and edges on the external INT0 pin that activate the interrupt are defined in Table 12-2. The value on the INT0 pin is sampled before detecting edges. If edge or toggle interrupt is selected, pulses that last longer than one clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt.

4. ATMEGA 328P INTERRUPT

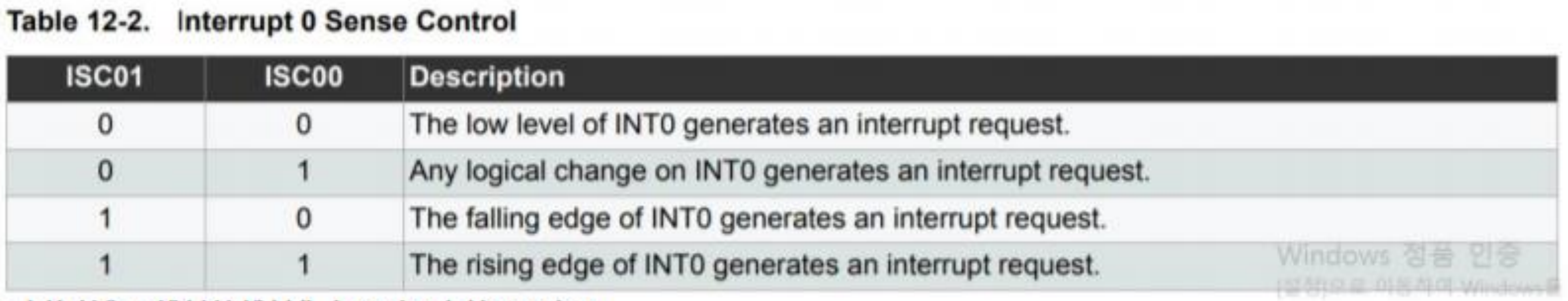


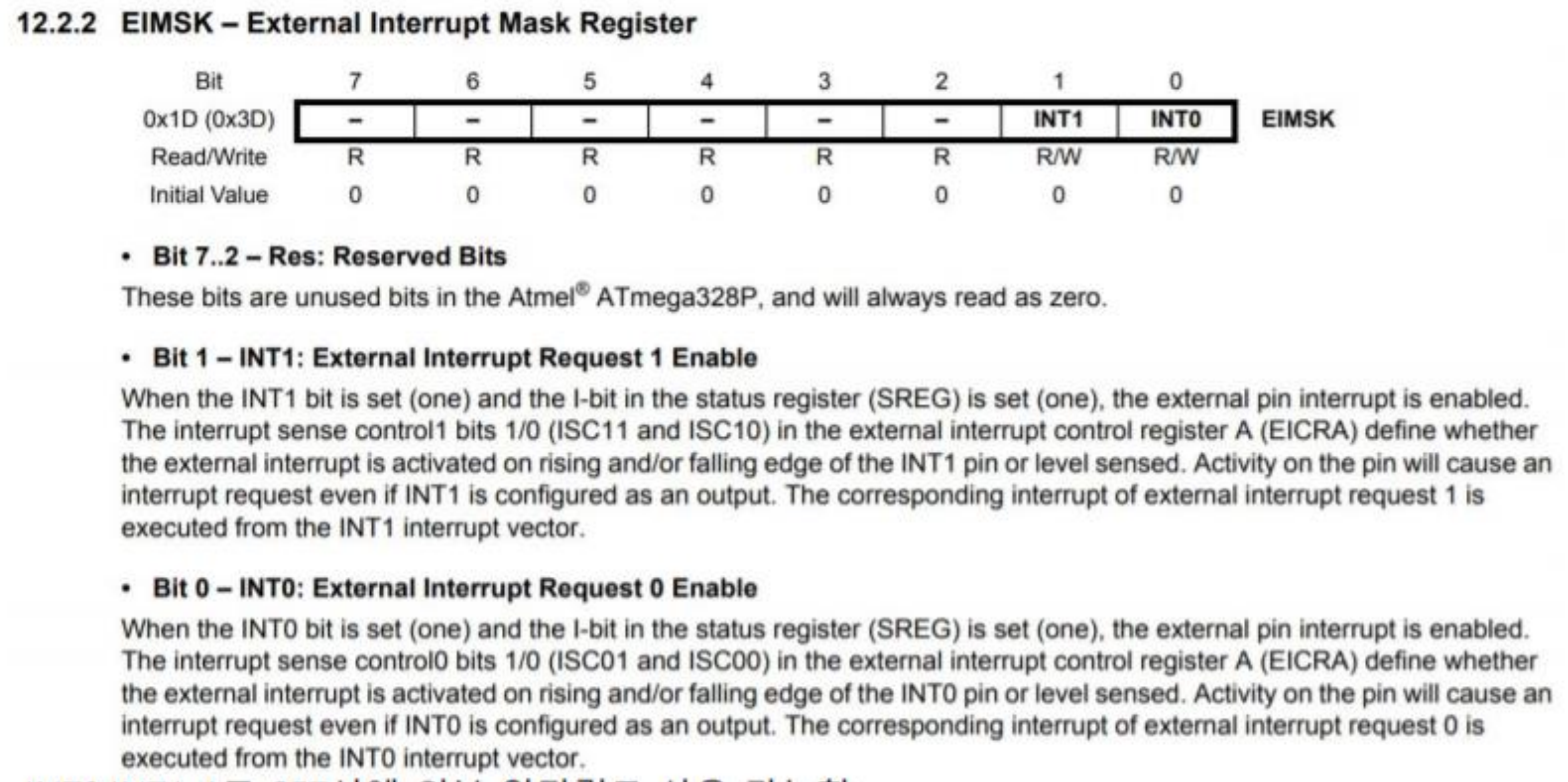
Table 12-2. Interrupt 0 Sense Control

ISC01	ISC00	Description
0	0	The low level of INT0 generates an interrupt request.
0	1	Any logical change on INT0 generates an interrupt request.
1	0	The falling edge of INT0 generates an interrupt request.
1	1	The rising edge of INT0 generates an interrupt request.

- EICRA : INT0,INT1 Control Register

ISCn1	ISCn0	Description
0	0	INTn가 LOW LEVEL이면 인터럽트
0	1	INTn Logic Level이 high or low로 변하면 인터럽트
1	0	INTn 하강에지때 인터럽트
1	1	INTn 상승에지때 인터럽트

4. ATMEGA 328P INTERRUPT



12.2.2 EIMSK – External Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
0x1D (0x3D)	–	–	–	–	–	–	INT1	INT0	EIMSK
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..2 – Res: Reserved Bits**

These bits are unused bits in the Atmel® ATmega328P, and will always read as zero.

- **Bit 1 – INT1: External Interrupt Request 1 Enable**

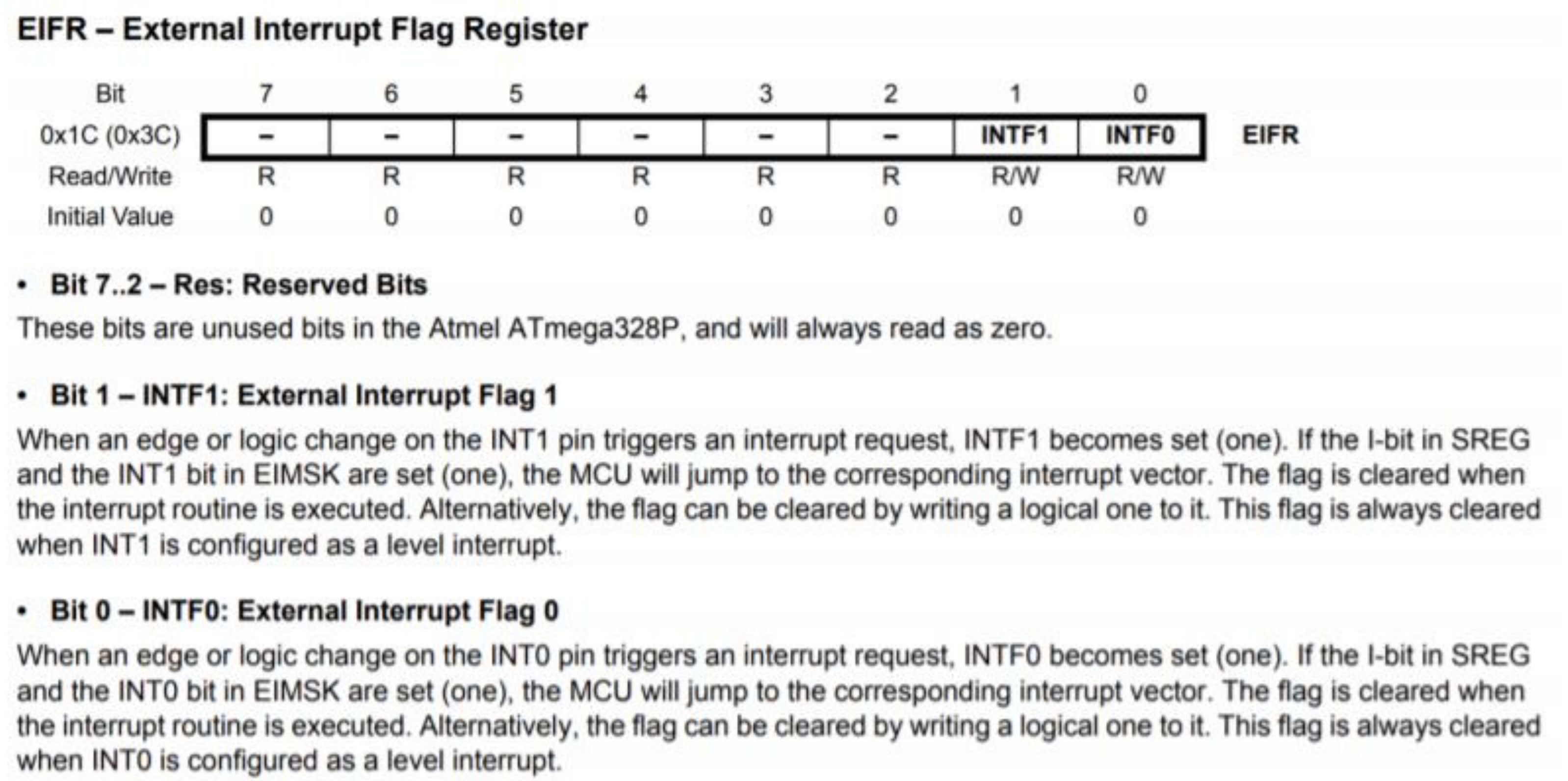
When the INT1 bit is set (one) and the I-bit in the status register (SREG) is set (one), the external pin interrupt is enabled. The interrupt sense control1 bits 1/0 (ISC11 and ISC10) in the external interrupt control register A (EICRA) define whether the external interrupt is activated on rising and/or falling edge of the INT1 pin or level sensed. Activity on the pin will cause an interrupt request even if INT1 is configured as an output. The corresponding interrupt of external interrupt request 1 is executed from the INT1 interrupt vector.

- **Bit 0 – INT0: External Interrupt Request 0 Enable**

When the INT0 bit is set (one) and the I-bit in the status register (SREG) is set (one), the external pin interrupt is enabled. The interrupt sense control0 bits 1/0 (ISC01 and ISC00) in the external interrupt control register A (EICRA) define whether the external interrupt is activated on rising and/or falling edge of the INT0 pin or level sensed. Activity on the pin will cause an interrupt request even if INT0 is configured as an output. The corresponding interrupt of external interrupt request 0 is executed from the INT0 interrupt vector.

- INT0,INT1 1로 SET시에 외부 인터럽트 사용 가능함.

4. ATMEGA 328P INTERRUPT



EIFR – External Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x1C (0x3C)	–	–	–	–	–	–	INTF1	INTF0	EIFR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..2 – Res: Reserved Bits**

These bits are unused bits in the Atmel ATmega328P, and will always read as zero.

- **Bit 1 – INTF1: External Interrupt Flag 1**

When an edge or logic change on the INT1 pin triggers an interrupt request, INTF1 becomes set (one). If the I-bit in SREG and the INT1 bit in EIMSK are set (one), the MCU will jump to the corresponding interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it. This flag is always cleared when INT1 is configured as a level interrupt.

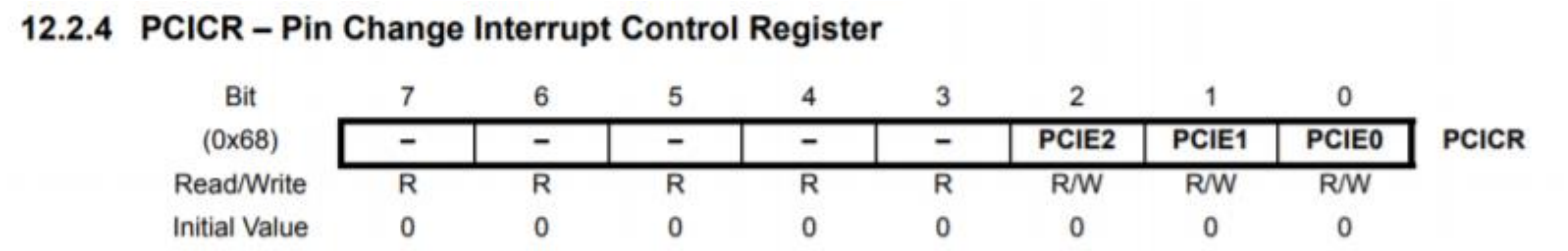
- **Bit 0 – INTF0: External Interrupt Flag 0**

When an edge or logic change on the INT0 pin triggers an interrupt request, INTF0 becomes set (one). If the I-bit in SREG and the INT0 bit in EIMSK are set (one), the MCU will jump to the corresponding interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it. This flag is always cleared when INT0 is configured as a level interrupt.

- EIFR : External Interrupt Flag Status Register

    외부 인터럽트 활성화 되면 flag가 1로 변함. 이때 이 값은 해제는 할 수 있지만
    인위적으로 1로 만드는것은 불가능 하다.

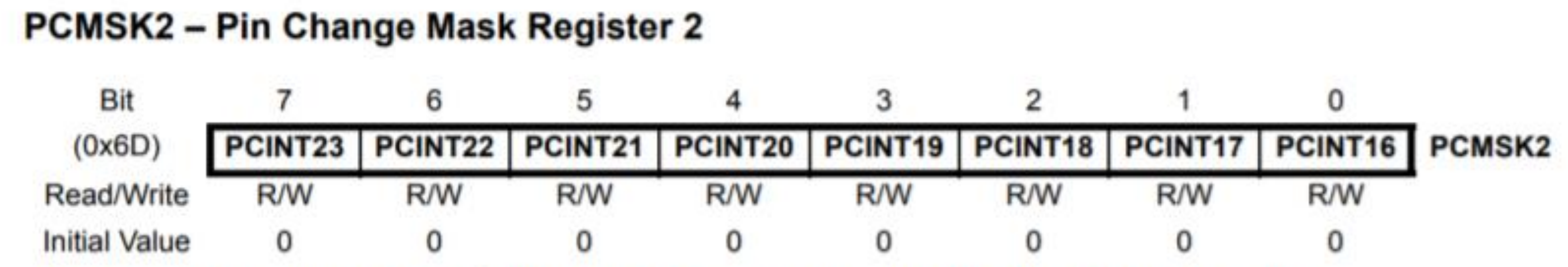
4. ATMEGA 328P INTERRUPT



12.2.4 PCICR – Pin Change Interrupt Control Register

Bit	7	6	5	4	3	2	1	0	
(0x68)	–	–	–	–	–	PCIE2	PCIE1	PCIE0	PCICR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- Pin Change Interrupt 를 enable 한다.
- 각각 PCINT23..16,14..8,7..0에서 변화가 있을시 인터럽트 발생한다.

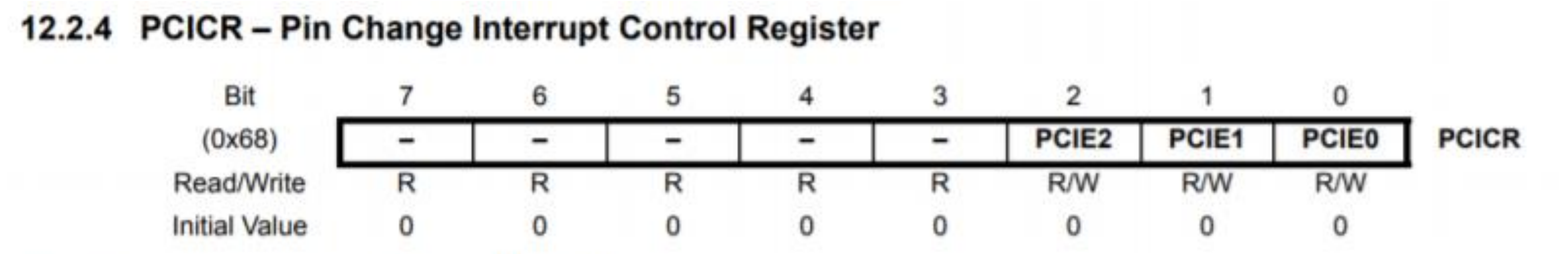


PCMSK2 – Pin Change Mask Register 2

Bit	7	6	5	4	3	2	1	0	
(0x6D)	PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16	PCMSK2
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	

- PCMSK 레지스터에서 사용할 핀에 대하여 SET하면 그 핀의 값이 변화하였을때 인터럽트가 발생한다.

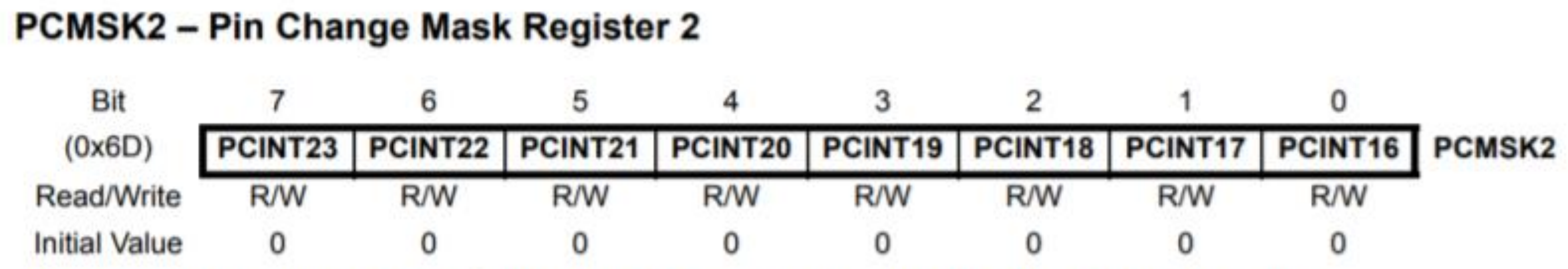
4. ATMEGA 328P INTERRUPT



12.2.4 PCICR – Pin Change Interrupt Control Register

Bit	7	6	5	4	3	2	1	0	
(0x68)	–	–	–	–	–	PCIE2	PCIE1	PCIE0	PCICR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- Pin Change Interrupt 를 enable 한다.
- 각각 PCINT23..16,14..8,7..0에서 변화가 있을시 인터럽트 발생한다.

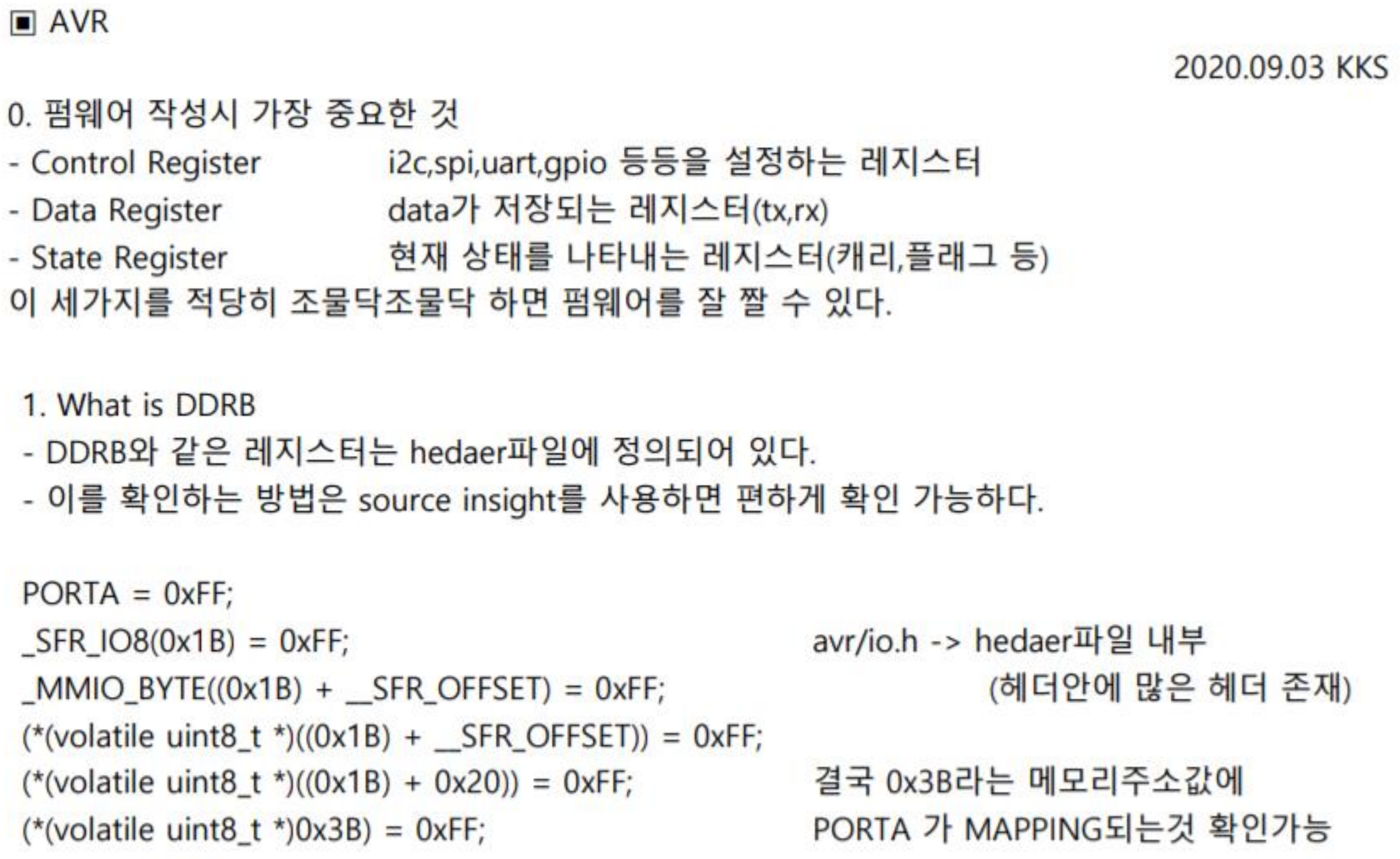


PCMSK2 – Pin Change Mask Register 2

Bit	7	6	5	4	3	2	1	0	
(0x6D)	PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16	PCMSK2
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	

- PCMSK 레지스터에서 사용할 핀에 대하여 SET하면 그 핀의 값이 변화하였을때 인터럽트가 발생한다.

4. ATMEGA 328P INTERRUPT



▣ AVR

2020.09.03 KKS

0. 펌웨어 작성시 가장 중요한 것

- Control Register	i2c,spi,uart,gpio 등등을 설정하는 레지스터
- Data Register	data가 저장되는 레지스터(tx,rx)
- State Register	현재 상태를 나타내는 레지스터(캐리,플래그 등)

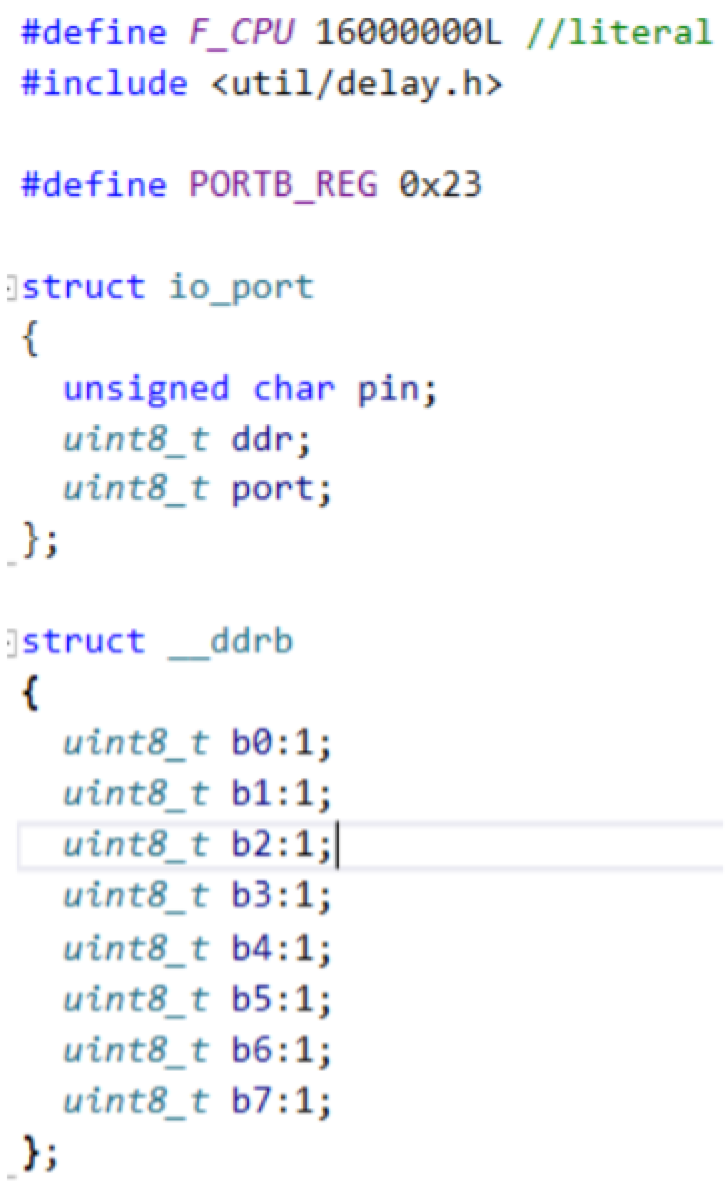
이 세가지를 적당히 조물닥조물닥 하면 펌웨어를 잘 짤 수 있다.

1. What is DDRB
- DDRB와 같은 레지스터는 hedaer파일에 정의되어 있다.
- 이를 확인하는 방법은 source insight를 사용하면 편하게 확인 가능하다.

```
PORTA = 0xFF;
_SFR_IO8(0x1B) = 0xFF;                                avr/io.h -> hedaer파일 내부
_MMIO_BYTE((0x1B) + __SFR_OFFSET) = 0xFF;                      (헤더안에 많은 헤더 존재)
(*(volatile uint8_t *)((0x1B) + __SFR_OFFSET)) = 0xFF;
(*(volatile uint8_t *)((0x1B) + 0x20)) = 0xFF;        결국 0x3B라는 메모리주소값에
(*(volatile uint8_t *)0x3B) = 0xFF;                   PORTA 가 MAPPING되는것 확인가능
```

4. ATMEGA 328P INTERRUPT

2. 레지스터 직접 매핑



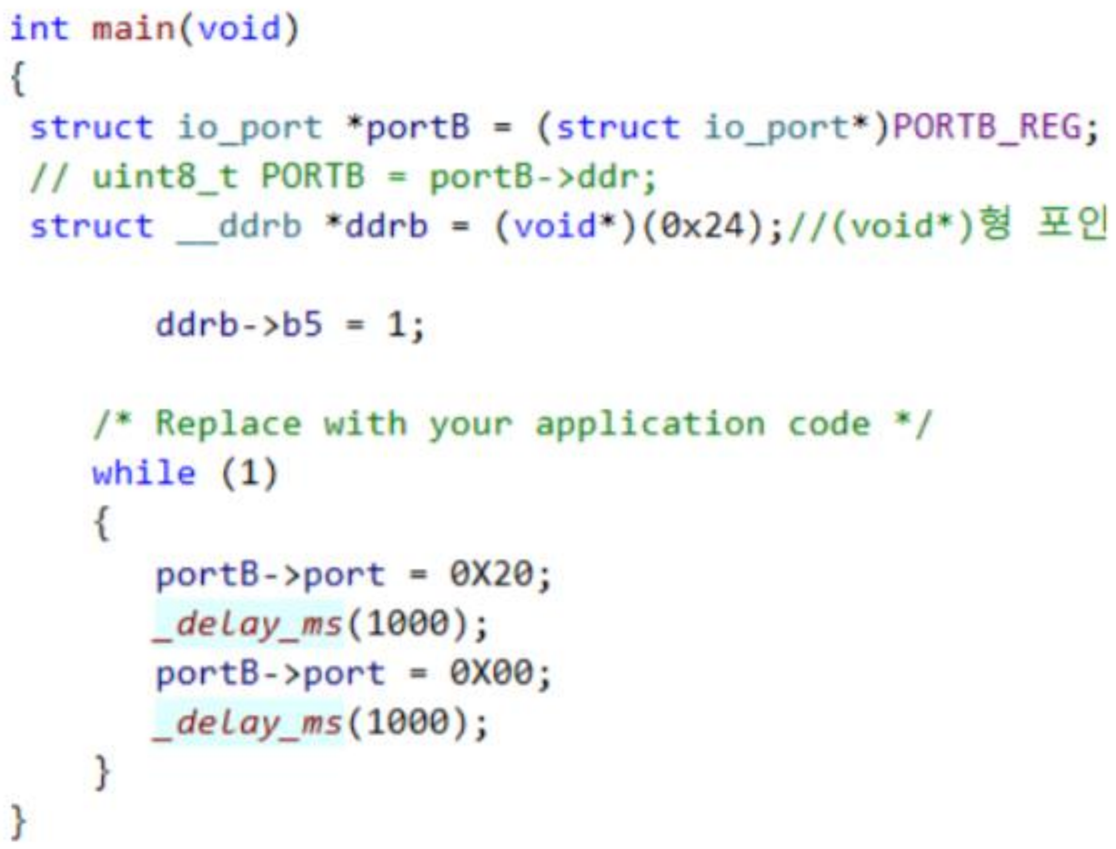
```c
#define F_CPU 16000000L //literal
#include <util/delay.h>

#define PORTB_REG 0x23

struct io_port
{
    unsigned char pin;
    uint8_t ddr;
    uint8_t port;
};

struct __ddrb
{
    uint8_t b0:1;
    uint8_t b1:1;
    uint8_t b2:1;
    uint8_t b3:1;
    uint8_t b4:1;
    uint8_t b5:1;
    uint8_t b6:1;
    uint8_t b7:1;
};
```

//16000000L로 선언한 이유
일반 Literal 상수는 data type이 int형임
이 int형은 16000000을 담지 못하기때문에
L이라는 DATATYPE을 명시해 준것

구조체를 사용하여 메모리 매핑

옆과 같이 비트필드를 이용하여
쪼개어 줄 수 있다.

4. ATMEGA 328P INTERRUPT

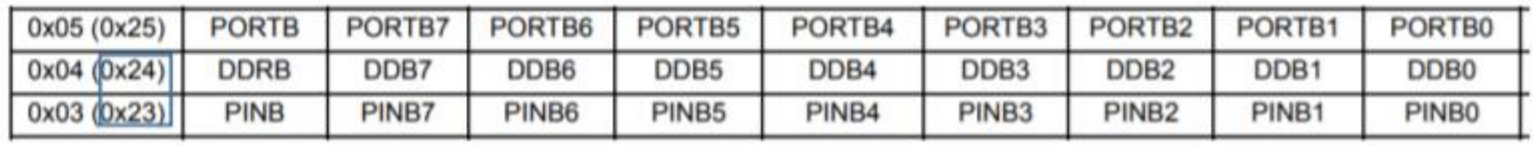


```c
int main(void)
{
 struct io_port *portB = (struct io_port*)PORTB_REG;
 // uint8_t PORTB = portB->ddr;
 struct __ddrb *ddrb = (void*)(0x24);//(void*)형 포인

      ddrb->b5 = 1;

    /* Replace with your application code */
    while (1)
    {
      portB->port = 0X20;
      _delay_ms(1000);
      portB->port = 0X00;
      _delay_ms(1000);
    }
}
```

0x23,4는 DataSheet 참조

(void*) 타입 형변환 이유!
0x24라는 상수를 주소값으로
대입하여야 하기 때문에

위처럼 구조체변수 자체를
포인터 타입으로 형변환
하여도 무방하다.



0x05 (0x25)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
0x04 (0x24)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0
0x03 (0x23)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0