Praktikumsaufgabe: Interrupts

Prof. Dr. Ralf Gerlich
Hochschule Furtwangen
Fakultät Informatik
Robert-Gerwig-Platz 1
D-78120 Furtwangen
ralf.gerlich@hs-furtwangen.de

7. März 2025

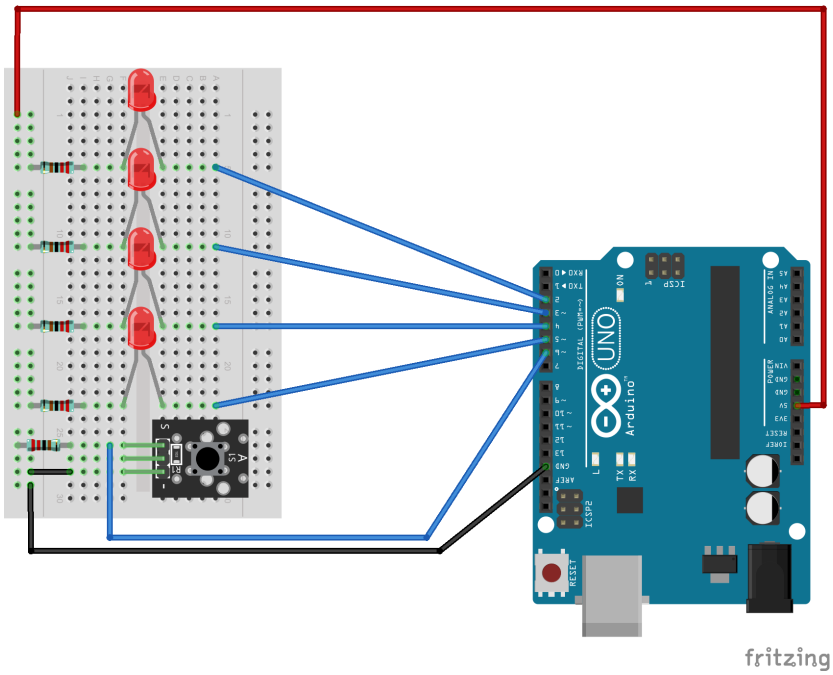
Zusammenfassung

Aktualisieren Sie auf Tastendruck den Status einer LED, während gleichzeitig eine Lichtorgel läuft.

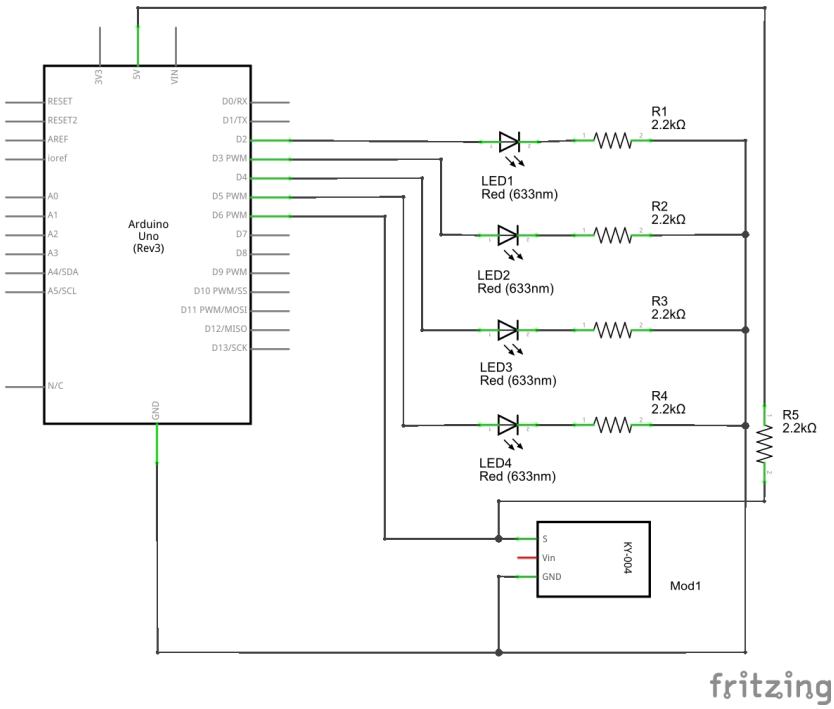
Inhaltsverzeichnis

1 Schaltungsaufbau

Bauen Sie die im folgenden Aufbauplan gezeigte und schon aus einer früheren Aufgabe bekannte Schaltung auf:



Als Referenz sei hier noch der Schaltplan gegeben:



2 Aufgabenstellung

Das Ziel dieser Aufgabe ist es, die auf dem Arduino integrierte LED auf Tastendruck einzuschalten, während gleichzeitig mit den 4 LEDs auf der Steckplatine eine kontinuierlich in eine Richtung laufende Lichtorgel gezeigt wird. Wenn die Lichtorgel ein Ende erreicht hat, soll die integrierte LED wieder ausgeschaltet werden, bevor die Lichtorgel wieder am anderen Ende beginnt.

Die Reaktion auf den Tastendruck soll dabei in einem Interrupt-Handler erfolgen. Die integrierte LED soll erst eingeschaltet werden, wenn der Taster losgelassen wird.

Ein möglicher Ablauf sieht also so aus:

1. Die Lichtorgel beginnt durch das Einschalten der ersten LED. Die integrierte LED ist aus.
2. Die Lichtorgel durchläuft die LEDs der Reihe nach.
3. Der Taster wird gedrückt und losgelassen und die integrierte LED leuchtet auf.
4. Die Lichtorgel erreicht die letzte LED der Reihe. Die integrierte LED wird ausgeschaltet.
5. Der Ablauf beginnt von vorn.

Die zu modifizierenden Stellen sind in der Projektvorlage mit `TODO` und einem entsprechenden Kommentar versehen.

Das C-Makro `sei()` fügt die Assembler-Instruktion `sei` an der entsprechenden Stelle ein. Gleiches gilt für das C-Makro `cli()` und die Assembler-Instruktion `cli`. Diese Makros sind in der Header-Datei `<avr/interrupt.h>` definiert, die in der Projektvorlage bereits eingebunden wird.

Beachten Sie, dass laut Anschlussplan die LEDs an den Pins 2 bis 5 und der Taster an Pin 6 des Port D angeschlossen sind. Die integrierte LED ist an Pin 5 des Port B angeschlossen.

Im Anhang finden Sie einen Auszug aus dem Datenblatt des ATMega328P mit den relevanten Kapiteln „Interrupts“, „External Interrupts“ und „I/O-Ports“. Dort finden Sie auch Hinweise, wie die Aufgabe gelöst werden kann.

Die an Pin 6 des Port D angeschlossene Platine mit Taster hat einen eingebauten Pull-Up-Widerstand, der mit dem mittleren Pin der Tasterplatine verbunden ist. Implementieren Sie das Programm deshalb ohne den internen Pull-Up-Widerstand des AVR zu nutzen.

A Datenblattauszug ATmega328P – Interrupts

12. Interrupts

This section describes the specifics of the interrupt handling as performed in ATmega48A/PA/88A/PA/168A/PA/328/P. For a general explanation of the AVR interrupt handling, refer to ”Reset and Interrupt Handling” on page 23.

The interrupt vectors in ATmega 48A/48PA, ATmega88A/88PA, ATmega168A/168PA and ATmega328/328P are generally the same, with the following differences:

- Each Interrupt Vector occupies two instruction words in ATmega168A/168PA and ATmega328/328P, and one instruction word in ATmega 48A/48PA and ATmega88A/88PA.
- ATmega 48A/48PA does not have a separate Boot Loader Section. In ATmega88A/88PA, ATmega168A/168PA and ATmega328/328P, the Reset Vector is affected by the BOOTRST fuse, and the Interrupt Vector start address is affected by the IVSEL bit in MCUCR.

12.1 Interrupt Vectors in ATmega48A and ATmega48PA

Table 12-1. Reset and Interrupt Vectors in ATmega48A and ATmega48PA

Vector No.	Program Address	Source	Interrupt Definition
1	0x000	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset
2	0x001	INT0	External Interrupt Request 0
3	0x002	INT1	External Interrupt Request 1
4	0x003	PCINT0	Pin Change Interrupt Request 0
5	0x004	PCINT1	Pin Change Interrupt Request 1
6	0x005	PCINT2	Pin Change Interrupt Request 2
7	0x006	WDT	Watchdog Time-out Interrupt
8	0x007	TIMER2_COMPA	Timer/Counter2 Compare Match A
9	0x008	TIMER2_COMPB	Timer/Counter2 Compare Match B
10	0x009	TIMER2_OVF	Timer/Counter2 Overflow
11	0x00A	TIMER1_CAPT	Timer/Counter1 Capture Event
12	0x00B	TIMER1_COMPA	Timer/Counter1 Compare Match A
13	0x00C	TIMER1_COMPB	Timer/Counter1 Compare Match B
14	0x00D	TIMER1_OVF	Timer/Counter1 Overflow
15	0x00E	TIMER0_COMPA	Timer/Counter0 Compare Match A
16	0x00F	TIMER0_COMPB	Timer/Counter0 Compare Match B
17	0x010	TIMER0_OVF	Timer/Counter0 Overflow
18	0x011	SPI_STC	SPI Serial Transfer Complete
19	0x012	USART_RX	USART Rx Complete
20	0x013	USART_UDRE	USART, Data Register Empty
21	0x014	USART_TX	USART, Tx Complete
22	0x015	ADC	ADC Conversion Complete
23	0x016	EE_READY	EEPROM Ready

Table 12-1. Reset and Interrupt Vectors in ATmega48A and ATmega48PA (Continued)

Vector No.	Program Address	Source	Interrupt Definition
24	0x017	ANALOG_COMP	Analog Comparator
25	0x018	TWI	2-wire Serial Interface
26	0x019	SPM_Ready	Store Program Memory Ready

The most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega 48A/48PA is:

```
Address  Labels  Code              Comments
0x000            rjmp   RESET       ; Reset Handler
0x001            rjmp   EXT_INT0    ; IRQ0 Handler
0x002            rjmp   EXT_INT1    ; IRQ1 Handler
0x003            rjmp   PCINT0      ; PCINT0 Handler
0x004            rjmp   PCINT1      ; PCINT1 Handler
0x005            rjmp   PCINT2      ; PCINT2 Handler
0x006            rjmp   WDT         ; Watchdog Timer Handler
0x007            rjmp   TIM2_COMPA  ; Timer2 Compare A Handler
0x008            rjmp   TIM2_COMPB  ; Timer2 Compare B Handler
0x009            rjmp   TIM2_OVF    ; Timer2 Overflow Handler
0x00A            rjmp   TIM1_CAPT   ; Timer1 Capture Handler
0x00B            rjmp   TIM1_COMPA  ; Timer1 Compare A Handler
0x00C            rjmp   TIM1_COMPB  ; Timer1 Compare B Handler
0x00D            rjmp   TIM1_OVF     ; Timer1 Overflow Handler
0x00E            rjmp   TIM0_COMPA   ; Timer0 Compare A Handler
0x00F            rjmp   TIM0_COMPB   ; Timer0 Compare B Handler
0x010            rjmp   TIM0_OVF     ; Timer0 Overflow Handler
0x011            rjmp   SPI_STC      ; SPI Transfer Complete Handler
0x012            rjmp   USART_RXC    ; USART, RX Complete Handler
0x013            rjmp   USART_UDRE   ; USART, UDR Empty Handler
0x014            rjmp   USART_TXC    ; USART, TX Complete Handler
0x015            rjmp   ADC          ; ADC Conversion Complete Handler
0x016            rjmp   EE_RDY       ; EEPROM Ready Handler
0x017            rjmp   ANA_COMP     ; Analog Comparator Handler
0x018            rjmp   TWI          ; 2-wire Serial Interface Handler
0x019            rjmp               ; SPM_RDYStore Program Memory Ready
Handler
;
0x01A RESET:     ldi    r16, high(RAMEND); Main program start
0x01B            out    SPH,r16      ; Set Stack Pointer to top of RAM
0x01C            ldi    r16, low(RAMEND)
0x01D            out    SPL,r16
0x01E            sei                 ; Enable interrupts
0x01F            <instr>  xxx
  ...   ...      ...     ...
```

12.2 Interrupt Vectors in ATmega88A and ATmega88PA

Table 12-2. Reset and Interrupt Vectors in ATmega88A and ATmega88PA

Vector No.	Program Address(2)	Source	Interrupt Definition
1	0x000(1)	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset
2	0x001	INT0	External Interrupt Request 0
3	0x002	INT1	External Interrupt Request 1
4	0x003	PCINT0	Pin Change Interrupt Request 0
5	0x004	PCINT1	Pin Change Interrupt Request 1
6	0x005	PCINT2	Pin Change Interrupt Request 2
7	0x006	WDT	Watchdog Time-out Interrupt
8	0x007	TIMER2_COMPA	Timer/Counter2 Compare Match A
9	0x008	TIMER2_COMPB	Timer/Counter2 Compare Match B
10	0x009	TIMER2_OVF	Timer/Counter2 Overflow
11	0x00A	TIMER1_CAPT	Timer/Counter1 Capture Event
12	0x00B	TIMER1_COMPA	Timer/Counter1 Compare Match A
13	0x00C	TIMER1_COMPB	Timer/Counter1 Compare Match B
14	0x00D	TIMER1_OVF	Timer/Counter1 Overflow
15	0x00E	TIMER0_COMPA	Timer/Counter0 Compare Match A
16	0x00F	TIMER0_COMPB	Timer/Counter0 Compare Match B
17	0x010	TIMER0_OVF	Timer/Counter0 Overflow
18	0x011	SPI_STC	SPI Serial Transfer Complete
19	0x012	USART_RX	USART Rx Complete
20	0x013	USART_UDRE	USART, Data Register Empty
21	0x014	USART_TX	USART, Tx Complete
22	0x015	ADC	ADC Conversion Complete
23	0x016	EE_READY	EEPROM Ready
24	0x017	ANALOG_COMP	Analog Comparator
25	0x018	TWI	2-wire Serial Interface
26	0x019	SPM_Ready	Store Program Memory Ready

Notes:
1. When the BOOTRST Fuse is programmed, the device will jump to the Boot Loader address at reset, see "Boot Loader Support – Read-While-Write Self-Programming" on page 272.
2. When the IVSEL bit in MCUCR is set, Interrupt Vectors will be moved to the start of the Boot Flash Section. The address of each Interrupt Vector will then be the address in this table added to the start address of the Boot Flash Section.

Table 12-3 on page 69 shows reset and Interrupt Vectors placement for the various combinations of BOOTRST and IVSEL settings. If the program never enables an interrupt source, the Interrupt Vectors are not used, and regular program code can be placed at these locations. This is also the case if the Reset Vector is in the Application section while the Interrupt Vectors are in the Boot section or vice versa.

Table 12-3. Reset and Interrupt Vectors Placement in ATmega88A and ATmega88PA(1)

BOOTRST	IVSEL	Reset Address	Interrupt Vectors Start Address
1	0	0x000	0x001
1	1	0x000	Boot Reset Address + 0x001
0	0	Boot Reset Address	0x001
0	1	Boot Reset Address	Boot Reset Address + 0x001

Note: 1. The Boot Reset Address is shown in Table 27-7 on page 284. For the BOOTRST Fuse “1” means unprogrammed while “0” means programmed.

The most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega88A/88PA is:

```
Address  Labels  Code             Comments
0x000           rjmp   RESET      ; Reset Handler
0x001           rjmp   EXT_INT0   ; IRQ0 Handler
0x002           rjmp   EXT_INT1   ; IRQ1 Handler
0x003           rjmp   PCINT0     ; PCINT0 Handler
0x004           rjmp   PCINT1     ; PCINT1 Handler
0x005           rjmp   PCINT2     ; PCINT2 Handler
0x006           rjmp   WDT        ; Watchdog Timer Handler
0x007           rjmp   TIM2_COMPA ; Timer2 Compare A Handler
0X008           rjmp   TIM2_COMPB ; Timer2 Compare B Handler
0x009           rjmp   TIM2_OVF   ; Timer2 Overflow Handler
0x00A           rjmp   TIM1_CAPT  ; Timer1 Capture Handler
0x00B           rjmp   TIM1_COMPA ; Timer1 Compare A Handler
0x00C           rjmp   TIM1_COMPB ; Timer1 Compare B Handler
0x00D           rjmp   TIM1_OVF   ; Timer1 Overflow Handler
0x00E           rjmp   TIM0_COMPA ; Timer0 Compare A Handler
0x00F           rjmp   TIM0_COMPB ; Timer0 Compare B Handler
0x010           rjmp   TIM0_OVF   ; Timer0 Overflow Handler
0x011           rjmp   SPI_STC    ; SPI Transfer Complete Handler
0x012           rjmp   USART_RXC  ; USART, RX Complete Handler
0x013           rjmp   USART_UDRE ; USART, UDR Empty Handler
0x014           rjmp   USART_TXC  ; USART, TX Complete Handler
0x015           rjmp   ADC        ; ADC Conversion Complete Handler
0x016           rjmp   EE_RDY     ; EEPROM Ready Handler
0x017           rjmp   ANA_COMP   ; Analog Comparator Handler
0x018           rjmp   TWI        ; 2-wire Serial Interface Handler
0x019           rjmp   SPM_RDY    ; Store Program Memory Ready Handler
;
0x01A    RESET: ldi    r16, high(RAMEND); Main program start
0x01B           out    SPH,r16    ; Set Stack Pointer to top of RAM
0x01C           ldi    r16, low(RAMEND)
0x01D           out    SPL,r16
0x01E           sei               ; Enable interrupts
0x01F           <instr>  xxx
```

When the BOOTRST Fuse is unprogrammed, the Boot section size set to 2Kbytes and the IVSEL bit in the MCUCR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega88A/88PA is:

```
Address  Labels  Code             Comments
0x000    RESET: ldi    r16,high(RAMEND); Main program start
0x001           out    SPH,r16    ; Set Stack Pointer to top of RAM
```

```
0x002           ldi    r16,low(RAMEND)
0x003           out    SPL,r16
0x004           sei                ; Enable interrupts
0x005           <instr>  xxx
;
.org 0xC01
0xC01           rjmp   EXT_INT0    ; IRQ0 Handler
0xC02           rjmp   EXT_INT1    ; IRQ1 Handler
...             ...    ...         ;
0xC19           rjmp   SPM_RDY     ; Store Program Memory Ready Handler
```

When the BOOTRST Fuse is programmed and the Boot section size set to 2Kbytes, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega88A/88PA is:

```
Address                          LabelsCodeComments
.org 0x001
0x001           rjmp   EXT_INT0    ; IRQ0 Handler
0x002           rjmp   EXT_INT1    ; IRQ1 Handler
...             ...    ...         ;
0x019           rjmp   SPM_RDY     ; Store Program Memory Ready Handler
;
.org 0xC00
0xC00    RESET: ldi    r16,high(RAMEND); Main program start
0xC01           out    SPH,r16     ; Set Stack Pointer to top of RAM
0xC02           ldi    r16,low(RAMEND)
0xC03           out    SPL,r16
0xC04           sei                ; Enable interrupts
0xC05           <instr>  xxx
```

When the BOOTRST Fuse is programmed, the Boot section size set to 2Kbytes and the IVSEL bit in the MCUCR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega88A/88PA is:

```
Address  Labels  Code              Comments
;
.org 0xC00
0xC00           rjmp   RESET       ; Reset handler
0xC01           rjmp   EXT_INT0    ; IRQ0 Handler
0xC02           rjmp   EXT_INT1    ; IRQ1 Handler
...             ...    ...         ;
0xC19           rjmp   SPM_RDY     ; Store Program Memory Ready Handler
;
0xC1A    RESET: ldi    r16,high(RAMEND); Main program start
0xC1B           out    SPH,r16     ; Set Stack Pointer to top of RAM
0xC1C           ldi    r16,low(RAMEND)
0xC1D           out    SPL,r16
0xC1E           sei                ; Enable interrupts
0xC1F           <instr>  xxx
```

12.3 Interrupt Vectors in ATmega168A and ATmega168PA

Table 12-4. Reset and Interrupt Vectors in ATmega168A and ATmega168PA

VectorNo.	Program Address(2)	Source	Interrupt Definition
1	0x0000(1)	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset
2	0x0002	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

Notes:
1. When the BOOTRST Fuse is programmed, the device will jump to the Boot Loader address at reset, see "Boot Loader Support – Read-While-Write Self-Programming" on page 272.
2. When the IVSEL bit in MCUCR is set, Interrupt Vectors will be moved to the start of the Boot Flash Section. The address of each Interrupt Vector will then be the address in this table added to the start address of the Boot Flash Section.

Table 12-5 on page 72 shows reset and Interrupt Vectors placement for the various combinations of BOOTRST and IVSEL settings. If the program never enables an interrupt source, the Interrupt Vectors are not used, and

regular program code can be placed at these locations. This is also the case if the Reset Vector is in the Application section while the Interrupt Vectors are in the Boot section or vice versa.

Table 12-5. Reset and Interrupt Vectors Placement in ATmega168A and ATmega168PA(1)

BOOTRST	IVSEL	Reset Address	Interrupt Vectors Start Address
1	0	0x000	0x002
1	1	0x000	Boot Reset Address + 0x0002
0	0	Boot Reset Address	0x002
0	1	Boot Reset Address	Boot Reset Address + 0x0002

Note: 1. The Boot Reset Address is shown in Table 27-7 on page 284. For the BOOTRST Fuse “1” means unprogrammed while “0” means programmed.

The most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega168A/168PA is:

```
Address  Labels  Code              Comments
0x0000           jmp    RESET       ; Reset Handler
0x0002           jmp    EXT_INT0    ; IRQ0 Handler
0x0004           jmp    EXT_INT1    ; IRQ1 Handler
0x0006           jmp    PCINT0      ; PCINT0 Handler
0x0008           jmp    PCINT1      ; PCINT1 Handler
0x000A           jmp    PCINT2      ; PCINT2 Handler
0x000C           jmp    WDT         ; Watchdog Timer Handler
0x000E           jmp    TIM2_COMPA  ; Timer2 Compare A Handler
0x0010           jmp    TIM2_COMPB  ; Timer2 Compare B Handler
0x0012           jmp    TIM2_OVF    ; Timer2 Overflow Handler
0x0014           jmp    TIM1_CAPT   ; Timer1 Capture Handler
0x0016           jmp    TIM1_COMPA  ; Timer1 Compare A Handler
0x0018           jmp    TIM1_COMPB  ; Timer1 Compare B Handler
0x001A           jmp    TIM1_OVF    ; Timer1 Overflow Handler
0x001C           jmp    TIM0_COMPA  ; Timer0 Compare A Handler
0x001E           jmp    TIM0_COMPB  ; Timer0 Compare B Handler
0x0020           jmp    TIM0_OVF    ; Timer0 Overflow Handler
0x0022           jmp    SPI_STC     ; SPI Transfer Complete Handler
0x0024           jmp    USART_RXC   ; USART, RX Complete Handler
0x0026           jmp    USART_UDRE  ; USART, UDR Empty Handler
0x0028           jmp    USART_TXC   ; USART, TX Complete Handler
0x002A           jmp    ADC         ; ADC Conversion Complete Handler
0x002C           jmp    EE_RDY      ; EEPROM Ready Handler
0x002E           jmp    ANA_COMP    ; Analog Comparator Handler
0x0030           jmp    TWI         ; 2-wire Serial Interface Handler
0x0032           jmp    SPM_RDY     ; Store Program Memory Ready Handler
;
0x0034   RESET:  ldi    r16, high(RAMEND); Main program start
0x0035           out    SPH,r16     ; Set Stack Pointer to top of RAM
0x0036           ldi    r16, low(RAMEND)
0x0037           out    SPL,r16
0x0038           sei                ; Enable interrupts
0x0039           <instr>  xxx
  ...    ...     ...     ...
```

When the BOOTRST Fuse is unprogrammed, the Boot section size set to 2Kbytes and the IVSEL bit in the MCUCR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega168A/168PA is:

```
Address  Labels  Code              Comments
0x0000   RESET:  ldi    r16,high(RAMEND); Main program start
0x0001           out    SPH,r16     ; Set Stack Pointer to top of RAM
0x0002           ldi    r16,low(RAMEND)
0x0003           out    SPL,r16
0x0004           sei                ; Enable interrupts
0x0005           <instr>  xxx
;
.org 0x1C02
0x1C02           jmp    EXT_INT0    ; IRQ0 Handler
0x1C04           jmp    EXT_INT1    ; IRQ1 Handler
...              ...    ...         ;
0x1C32           jmp    SPM_RDY     ; Store Program Memory Ready Handler
```

When the BOOTRST Fuse is programmed and the Boot section size set to 2Kbytes, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega168A/168PA is:

```
Address  Labels  Code              Comments
.org 0x0002
0x0002           jmp    EXT_INT0    ; IRQ0 Handler
0x0004           jmp    EXT_INT1    ; IRQ1 Handler
...              ...    ...         ;
0x0032           jmp    SPM_RDY     ; Store Program Memory Ready Handler
;
.org 0x1C00
0x1C00   RESET:  ldi    r16,high(RAMEND); Main program start
0x1C01           out    SPH,r16     ; Set Stack Pointer to top of RAM
0x1C02           ldi    r16,low(RAMEND)
0x1C03           out    SPL,r16
0x1C04           sei                  ; Enable interrupts
0x1C05           <instr>  xxx
```

When the BOOTRST Fuse is programmed, the Boot section size set to 2Kbytes and the IVSEL bit in the MCUCR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega168A/168PA is:

```
Address  Labels  Code              Comments
;
.org 0x1C00
0x1C00           jmp    RESET       ; Reset handler
0x1C02           jmp    EXT_INT0    ; IRQ0 Handler
0x1C04           jmp    EXT_INT1    ; IRQ1 Handler
...              ...    ...         ;
0x1C32           jmp    SPM_RDY     ; Store Program Memory Ready Handler
;
0x1C3    RESET:  ldi    r16,high(RAMEND); Main program start
0x1C35           out    SPH,r16     ; Set Stack Pointer to top of RAM
0x1C36           ldi    r16,low(RAMEND)
0x1C37           out    SPL,r16
0x1C38           sei                ; Enable interrupts
0x1C39           <instr>  xxx
```

12.4 Interrupt Vectors in ATmega328 and ATmega328P

Table 12-6. Reset and Interrupt Vectors in ATmega328 and ATmega328P

VectorNo.	Program Address(2)	Source	Interrupt Definition
1	0x0000(1)	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset
2	0x0002	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

Notes:
1. When the BOOTRST Fuse is programmed, the device will jump to the Boot Loader address at reset, see "Boot Loader Support – Read-While-Write Self-Programming" on page 272.
2. When the IVSEL bit in MCUCR is set, Interrupt Vectors will be moved to the start of the Boot Flash Section. The address of each Interrupt Vector will then be the address in this table added to the start address of the Boot Flash Section.

Table 12-7 on page 75 shows reset and Interrupt Vectors placement for the various combinations of BOOTRST and IVSEL settings. If the program never enables an interrupt source, the Interrupt Vectors are not used, and

regular program code can be placed at these locations. This is also the case if the Reset Vector is in the Application section while the Interrupt Vectors are in the Boot section or vice versa.

Table 12-7. Reset and Interrupt Vectors Placement in ATmega328 and ATmega328P(1)

BOOTRST	IVSEL	Reset Address	Interrupt Vectors Start Address
1	0	0x000	0x002
1	1	0x000	Boot Reset Address + 0x0002
0	0	Boot Reset Address	0x002
0	1	Boot Reset Address	Boot Reset Address + 0x0002

Note: 1. The Boot Reset Address is shown in Table 27-7 on page 284. For the BOOTRST Fuse “1” means unprogrammed while “0” means programmed.

The most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega328/328P is:

```
Address   Labels  Code            Comments
0x0000            jmp   RESET     ; Reset Handler
0x0002            jmp   EXT_INT0  ; IRQ0 Handler
0x0004            jmp   EXT_INT1  ; IRQ1 Handler
0x0006            jmp   PCINT0    ; PCINT0 Handler
0x0008            jmp   PCINT1    ; PCINT1 Handler
0x000A            jmp   PCINT2    ; PCINT2 Handler
0x000C            jmp   WDT       ; Watchdog Timer Handler
0x000E            jmp   TIM2_COMPA; Timer2 Compare A Handler
0x0010            jmp   TIM2_COMPB; Timer2 Compare B Handler
0x0012            jmp   TIM2_OVF  ; Timer2 Overflow Handler
0x0014            jmp   TIM1_CAPT ; Timer1 Capture Handler
0x0016            jmp   TIM1_COMPA; Timer1 Compare A Handler
0x0018            jmp   TIM1_COMPB; Timer1 Compare B Handler
0x001A            jmp   TIM1_OVF  ; Timer1 Overflow Handler
0x001C            jmp   TIM0_COMPA; Timer0 Compare A Handler
0x001E            jmp   TIM0_COMPB; Timer0 Compare B Handler
0x0020            jmp   TIM0_OVF  ; Timer0 Overflow Handler
0x0022            jmp   SPI_STC   ; SPI Transfer Complete Handler
0x0024            jmp   USART_RXC ; USART, RX Complete Handler
0x0026            jmp   USART_UDRE; USART, UDR Empty Handler
0x0028            jmp   USART_TXC ; USART, TX Complete Handler
0x002A            jmp   ADC       ; ADC Conversion Complete Handler
0x002C            jmp   EE_RDY    ; EEPROM Ready Handler
0x002E            jmp   ANA_COMP  ; Analog Comparator Handler
0x0030            jmp   TWI       ; 2-wire Serial Interface Handler
0x0032            jmp   SPM_RDY   ; Store Program Memory Ready Handler
;
0x0034  RESET:  ldi   r16, high(RAMEND); Main program start
0x0035          out   SPH,r16     ; Set Stack Pointer to top of RAM
0x0036          ldi   r16, low(RAMEND)
0x0037          out   SPL,r16
0x0038          sei               ; Enable interrupts
0x0039          <instr>  xxx
  ...     ...   ...     ...
```

When the BOOTRST Fuse is unprogrammed, the Boot section size set to 2Kbytes and the IVSEL bit in the MCUCR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega328/328P is:

```
Address  Labels  Code              Comments
0x0000   RESET:  ldi    r16,high(RAMEND); Main program start
0x0001           out    SPH,r16     ; Set Stack Pointer to top of RAM
0x0002           ldi    r16,low(RAMEND)
0x0003           out    SPL,r16
0x0004           sei                ; Enable interrupts
0x0005           <instr>  xxx
;
.org 0x3C02
0x3C02           jmp    EXT_INT0    ; IRQ0 Handler
0x3C04           jmp    EXT_INT1    ; IRQ1 Handler
...              ...    ...         ;
0x3C32           jmp    SPM_RDY     ; Store Program Memory Ready Handler
```

When the BOOTRST Fuse is programmed and the Boot section size set to 2Kbytes, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega328/328P is:

```
Address  Labels  Code              Comments
.org 0x0002
0x0002           jmp    EXT_INT0    ; IRQ0 Handler
0x0004           jmp    EXT_INT1    ; IRQ1 Handler
...              ...    ...         ;
0x0032           jmp    SPM_RDY     ; Store Program Memory Ready Handler
;
.org 0x3C00
0x3C00   RESET:  ldi    r16,high(RAMEND); Main program start
0x3C01           out    SPH,r16     ; Set Stack Pointer to top of RAM
0x3C02           ldi    r16,low(RAMEND)
0x3C03           out    SPL,r16
0x3C04           sei                ; Enable interrupts
0x3C05           <instr>  xxx
```

When the BOOTRST Fuse is programmed, the Boot section size set to 2Kbytes and the IVSEL bit in the MCUCR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega328/328P is:

```
Address  Labels  Code              Comments
;
.org 0x3C00
0x3C00           jmp    RESET       ; Reset handler
0x3C02           jmp    EXT_INT0    ; IRQ0 Handler
0x3C04           jmp    EXT_INT1    ; IRQ1 Handler
...              ...    ...         ;
0x3C32           jmp    SPM_RDY     ; Store Program Memory Ready Handler
;
0x3C34   RESET:  ldi    r16,high(RAMEND); Main program start
0x3C35           out    SPH,r16     ; Set Stack Pointer to top of RAM
0x3C36           ldi    r16,low(RAMEND)
0x3C37           out    SPL,r16
0x3C38           sei                ; Enable interrupts
0x3C39           <instr>  xxx
```

12.5 Register Description

12.5.1 Moving Interrupts Between Application and Boot Space, ATmega88A/88PA, ATmega168A/168PA and ATmega328/328P

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[1]	BODSE[1]	PUD	–	–	IVSEL	IVCE	MCUCR
Read/Write	R	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Note: 1. BODS and BODSE only available for picoPower devices ATmega48PA/88PA/168PA/328P

- **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 ”Boot Loader Support – Read-While-Write Self-Programming” on page 272 for details. To avoid unintentional changes of Interrupt Vector tables, a special write procedure must be followed to change the IVSEL bit:

1. Write the Interrupt Vector Change Enable (IVCE) bit to one.
2. 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 ”Boot Loader Support – Read-While-Write Self-Programming” on page 272 for details on Boot Lock bits.

- **Bit 0 – IVCE: Interrupt Vector Change Enable**

The IVCE bit must be written to logic one to enable change of the IVSEL bit. IVCE is cleared by hardware four cycles after it is written or when IVSEL is written. Setting the IVCE bit will disable interrupts, as explained in the IVSEL description above. See Code Example below.

Assembly Code Example

```
Move_interrupts:
       ; Enable change of Interrupt Vectors
       ldi         r16, (1<<IVCE)
       out         MCUCR, r16
       ; Move interrupts to Boot Flash section
       ldi         r16, (1<<IVSEL)
       out         MCUCR, r16
       ret
```

C Code Example

```
void Move_interrupts(void)
{
       /* Enable change of Interrupt Vectors */
       MCUCR = (1<<IVCE);
       /* Move interrupts to Boot Flash section */
       MCUCR = (1<<IVSEL);
}
```

B Datenblattauszug ATmega328P – External Interrupts

13. External Interrupts

The External Interrupts are triggered by the INT0 and INT1 pins or any of the PCINT23...0 pins. Observe that, if enabled, the interrupts will trigger even if the INT0 and INT1 or PCINT23...0 pins are configured as outputs. This feature provides a way of generating a software interrupt. The pin change interrupt PCI2 will trigger if any enabled PCINT[23:16] pin toggles. The pin change interrupt PCI1 will trigger if any enabled PCINT[14:8] pin toggles. The pin change interrupt PCI0 will trigger if any enabled PCINT[7:0] pin toggles. The PCMSK2, PCMSK1 and PCMSK0 Registers control which pins contribute to the pin change interrupts. Pin change interrupts on PCINT23...0 are detected asynchronously. This implies that these interrupts can be used for waking the part also from sleep modes other than Idle mode.

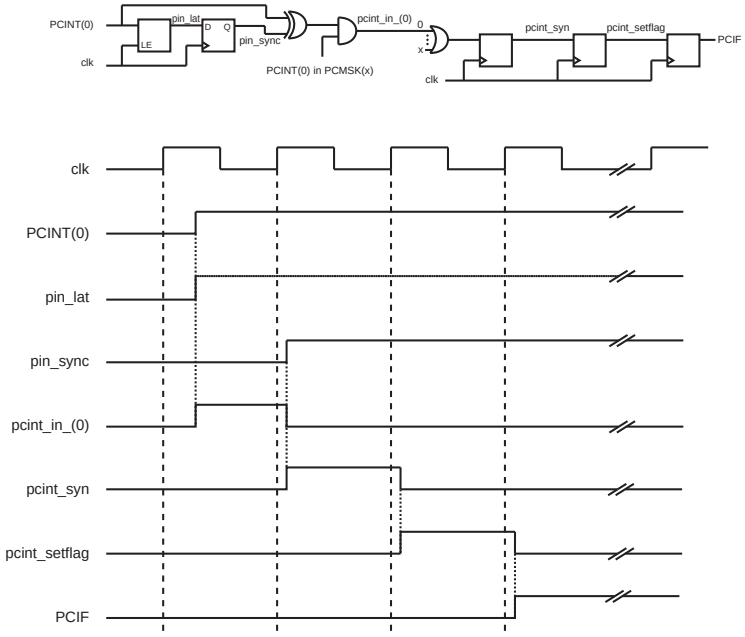
The External Interrupts can be triggered by a falling or rising edge or a low level. This is set up as indicated in the specification for the External Interrupt Control Registers – EICRA (INT2:0). When the external interrupt is enabled and is configured as level triggered, the interrupt will trigger as long as the pin is held low. Low level interrupts and the edge interrupt on INT2:0 are detected asynchronously. This implies that these interrupts 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: Note that if a level triggered interrupt is used for wake-up from Power-down, the required level must be held long enough for the MCU to complete the wake-up to trigger the level interrupt. If the level disappears before the end of the Start-up Time, the 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 ”System Clock and Clock Options” on page 36.

13.1 Pin Change Interrupt Timing

An example of timing of a pin change interrupt is shown in Figure 13-1.

Figure 13-1. Timing of pin change interrupts



13.2 Register Description

13.2.1 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 – Reserved**

These bits are unused bits in the ATmega48A/PA/88A/PA/168A/PA/328/P, 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 13-1. 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 ensured 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 13-1. 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 13-2. 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 ensured 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 13-2. 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.

13.2.2 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 – Reserved**

These bits are unused bits in the ATmega48A/PA/88A/PA/168A/PA/328/P, 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.

13.2.3 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 – Reserved**

These bits are unused bits in the ATmega48A/PA/88A/PA/168A/PA/328/P, 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.

13.2.4 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	

- **Bit 7:3 – Reserved**

These bits are unused bits in the ATmega48A/PA/88A/PA/168A/PA/328/P, and will always read as zero.

- **Bit 2 – PCIE2: Pin Change Interrupt Enable 2**

When the PCIE2 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), pin change interrupt 2 is enabled. Any change on any enabled PCINT[23:16] pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from the PCI2 Interrupt Vector. PCINT[23:16] pins are enabled individually by the PCMSK2 Register.

- **Bit 1 – PCIE1: Pin Change Interrupt Enable 1**

When the PCIE1 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), pin change interrupt 1 is enabled. Any change on any enabled PCINT[14:8] pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from the PCI1 Interrupt Vector. PCINT[14:8] pins are enabled individually by the PCMSK1 Register.

- **Bit 0 – PCIE0: Pin Change Interrupt Enable 0**

When the PCIE0 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), pin change interrupt 0 is enabled. Any change on any enabled PCINT[7:0] pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from the PCI0 Interrupt Vector. PCINT[7:0] pins are enabled individually by the PCMSK0 Register.

13.2.5 PCIFR – Pin Change Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x1B (0x3B)	–	–	–	–	–	PCIF2	PCIF1	PCIF0	PCIFR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:3 – Reserved**

These bits are unused bits in the ATmega48A/PA/88A/PA/168A/PA/328/P, and will always read as zero.

- **Bit 2 – PCIF2: Pin Change Interrupt Flag 2**

When a logic change on any PCINT[23:16] pin triggers an interrupt request, PCIF2 becomes set (one). If the I-bit in SREG and the PCIE2 bit in PCICR 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.

- **Bit 1 – PCIF1: Pin Change Interrupt Flag 1**

When a logic change on any PCINT[14:8] pin triggers an interrupt request, PCIF1 becomes set (one). If the I-bit in SREG and the PCIE1 bit in PCICR 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.

- **Bit 0 – PCIF0: Pin Change Interrupt Flag 0**

When a logic change on any PCINT[7:0] pin triggers an interrupt request, PCIF0 becomes set (one). If the I-bit in SREG and the PCIE0 bit in PCICR 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.

13.2.6 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	

- **Bit 7:0 – PCINT[23:16]: Pin Change Enable Mask 23...16**

Each PCINT[23:16]-bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT[23:16] is set and the PCIE2 bit in PCICR is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT[23:16] is cleared, pin change interrupt on the corresponding I/O pin is disabled.

13.2.7 PCMSK1 – Pin Change Mask Register 1

Bit	7	6	5	4	3	2	1	0	
(0x6C)	–	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8	PCMSK1
Read/Write	R	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 – Reserved**

This bit is an unused bit in the ATmega48A/PA/88A/PA/168A/PA/328/P, and will always read as zero.

- **Bit 6:0 – PCINT[14:8]: Pin Change Enable Mask 14...8**

Each PCINT[14:8]-bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT[14:8] is set and the PCIE1 bit in PCICR is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT[14:8] is cleared, pin change interrupt on the corresponding I/O pin is disabled.

13.2.8 PCMSK0 – Pin Change Mask Register 0

Bit	7	6	5	4	3	2	1	0	
(0x6B)	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	PCMSK0
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 7:0 – PCINT[7:0]: Pin Change Enable Mask 7...0**

Each PCINT[7:0] bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT[7:0] is set and the PCIE0 bit in PCICR is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT[7:0] is cleared, pin change interrupt on the corresponding I/O pin is disabled.

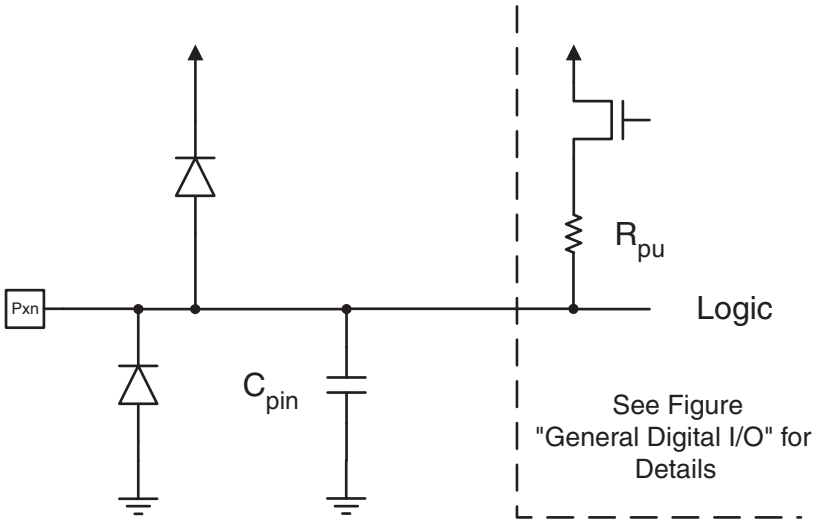
C Datenblattauszug ATmega328P – I/O-Ports

14. I/O-Ports

14.1 Overview

All AVR ports have true Read-Modify-Write functionality when used as general digital I/O ports. This means that the direction of one port pin can be changed without unintentionally changing the direction of any other pin with the SBI and CBI instructions. The same applies when changing drive value (if configured as output) or enabling/disabling of pull-up resistors (if configured as input). Each output buffer has symmetrical drive characteristics with both high sink and source capability. The pin driver is strong enough to drive LED displays directly. All port pins have individually selectable pull-up resistors with a supply-voltage invariant resistance. All I/O pins have protection diodes to both V_{CC} and Ground as indicated in Figure 14-1. Refer to ”Electrical Characteristics – (TA = -40°C to 85°C)” on page 308 for a complete list of parameters.

Figure 14-1. I/O Pin Equivalent Schematic



All registers and bit references in this section are written in general form. A lower case “x” represents the numbering letter for the port, and a lower case “n” represents the bit number. However, when using the register or bit defines in a program, the precise form must be used. For example, PORTB3 for bit no. 3 in Port B, here documented generally as PORTxn. The physical I/O Registers and bit locations are listed in ”Register Description” on page 100.

Three I/O memory address locations are allocated for each port, one each for the Data Register – PORTx, Data Direction Register – DDRx, and the Port Input Pins – PINx. The Port Input Pins I/O location is read only, while the Data Register and the Data Direction Register are read/write. However, writing a logic one to a bit in the PINx Register, will result in a toggle in the corresponding bit in the Data Register. In addition, the Pull-up Disable – PUD bit in MCUCR disables the pull-up function for all pins in all ports when set.

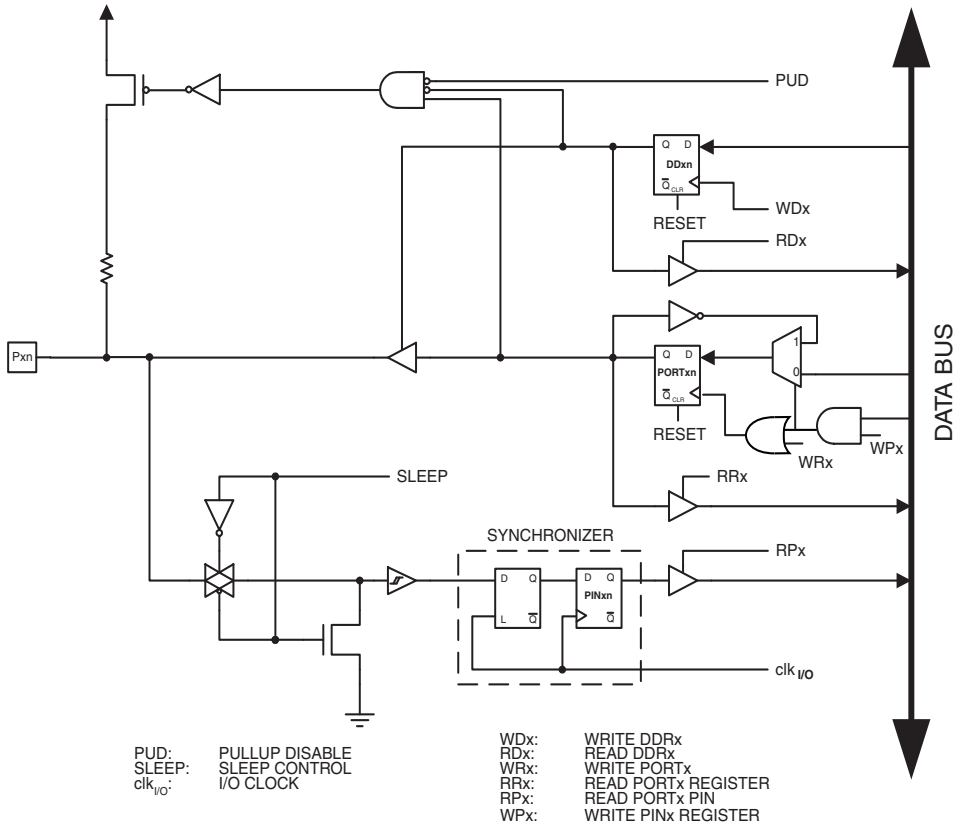
Using the I/O port as General Digital I/O is described in ”Ports as General Digital I/O” on page 85. Most port pins are multiplexed with alternate functions for the peripheral features on the device. How each alternate function interferes with the port pin is described in ”Alternate Port Functions” on page 89. Refer to the individual module sections for a full description of the alternate functions.

Note that enabling the alternate function of some of the port pins does not affect the use of the other pins in the port as general digital I/O.

14.2 Ports as General Digital I/O

The ports are bi-directional I/O ports with optional internal pull-ups. Figure 14-2 shows a functional description of one I/O-port pin, here generically called Pxn.

Figure 14-2. General Digital I/O(1)



Note: 1. WRx, WPx, WDx, RRx, RPx, and RDx are common to all pins within the same port. $clk_{I/O}$, SLEEP, and PUD are common to all ports.

14.2.1 Configuring the Pin

Each port pin consists of three register bits: DDxn, PORTxn, and PINxn. As shown in ”Register Description” on page 100, 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).

14.2.2 Toggling the Pin

Writing a logic one to PINxn toggles the value of PORTxn, independent on the value of DDRxn. Note that the SBI instruction can be used to toggle one single bit in a port.

14.2.3 Switching Between Input and Output

When switching between tri-state ({DDxn, PORTxn} = 0b00) and output high ({DDxn, PORTxn} = 0b11), an intermediate state with either pull-up enabled {DDxn, PORTxn} = 0b01) or output low ({DDxn, PORTxn} = 0b10) must occur. Normally, the pull-up enabled state is fully acceptable, as a high-impedance environment will not notice the difference between a strong high driver and a pull-up. If this is not the case, the PUD bit in the MCUCR Register can be set to disable all pull-ups in all ports.

Switching between input with pull-up and output low generates the same problem. The user must use either the tri-state ({DDxn, PORTxn} = 0b00) or the output high state ({DDxn, PORTxn} = 0b11) as an intermediate step.

Table 14-1 summarizes the control signals for the pin value.

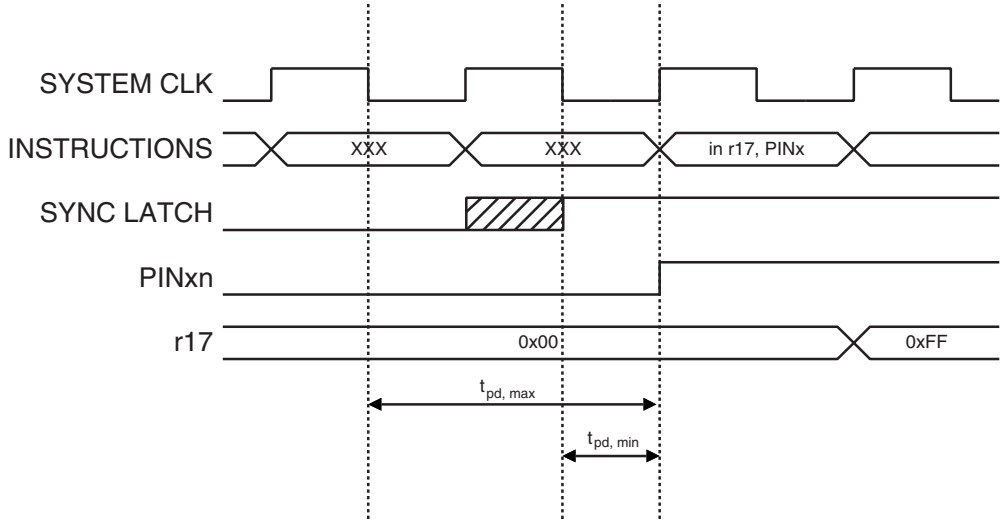
Table 14-1. **Port Pin Configurations**

DDxn	PORTxn	PUD (in MCUCR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

14.2.4 Reading the Pin Value

Independent of the setting of Data Direction bit DDxn, the port pin can be read through the PINxn Register bit. As shown in Figure 14-2, the PINxn Register bit and the preceding latch constitute a synchronizer. This is needed to avoid metastability if the physical pin changes value near the edge of the internal clock, but it also introduces a delay. Figure 14-3 shows a timing diagram of the synchronization when reading an externally applied pin value. The maximum and minimum propagation delays are denoted $t_{pd,max}$ and $t_{pd,min}$ respectively.

Figure 14-3. Synchronization when Reading an Externally Applied Pin value

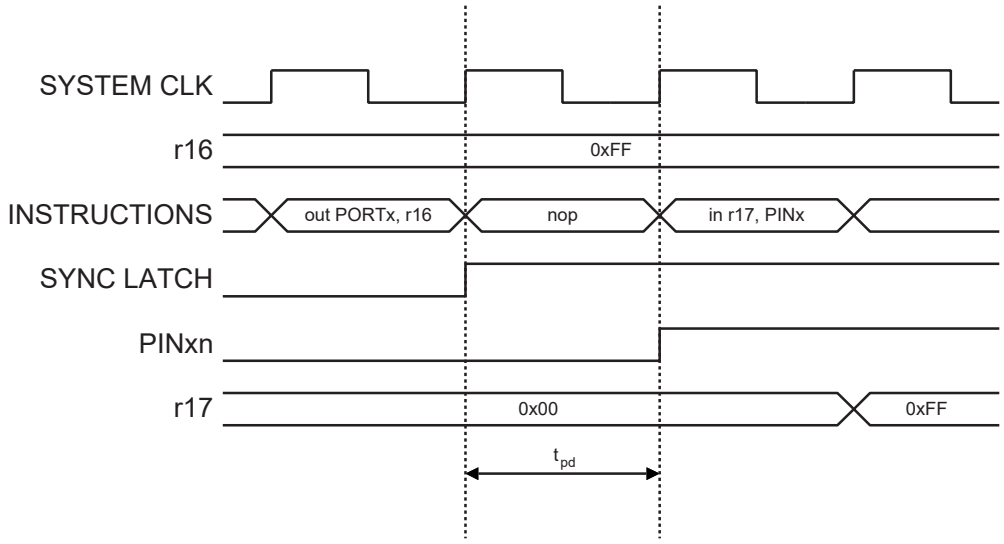


Consider the clock period starting shortly after the first falling edge of the system clock. The latch is closed when the clock is low, and goes transparent when the clock is high, as indicated by the shaded region of the “SYNC LATCH” signal. The signal value is latched when the system clock goes low. It is clocked into the PINxn Register at the succeeding positive clock edge. As indicated by the two arrows tpd,max and tpd,min, a single

signal transition on the pin will be delayed between ½ and 1½ system clock period depending upon the time of assertion.

When reading back a software assigned pin value, a nop instruction must be inserted as indicated in Figure 14-4. The out instruction sets the “SYNC LATCH” signal at the positive edge of the clock. In this case, the delay tpd through the synchronizer is 1 system clock period.

Figure 14-4. Synchronization when Reading a Software Assigned Pin Value



The following code example shows how to set port B pins 0 and 1 high, 2 and 3 low, and define the port pins from 4 to 7 as input with pull-ups assigned to port pins 6 and 7. The resulting pin values are read back again, but as previously discussed, a nop instruction is included to be able to read back the value recently assigned to some of the pins.

Assembly Code Example[1]

```
        ...
        ; Define pull-ups and set outputs high
        ; Define directions for port pins
        ldi             r16,(1<<PB7)|(1<<PB6)|(1<<PB1)|(1<<PB0)
        ldi
r17,(1<<DDB3)|(1<<DDB2)|(1<<DDB1)|(1<<DDB0)
        out             PORTB,r16
        out             DDRB,r17
        ; Insert nop for synchronization
        nop
        ; Read port pins
        in              r16,PINB
        ...
```

C Code Example

```
unsigned char i;
        ...
        /* Define pull-ups and set outputs high */
        /* Define directions for port pins */
        PORTB = (1<<PB7)|(1<<PB6)|(1<<PB1)|(1<<PB0);
        DDRB = (1<<DDB3)|(1<<DDB2)|(1<<DDB1)|(1<<DDB0);
        /* Insert nop for synchronization*/
        __no_operation();
        /* Read port pins */
        i = PINB;
        ...
```

Note: 1. For the assembly program, two temporary registers are used to minimize the time from pull-ups are set on pins 0, 1, 6, and 7, until the direction bits are correctly set, defining bit 2 and 3 as low and redefining bits 0 and 1 as strong high drivers.

14.2.5 Digital Input Enable and Sleep Modes

As shown in Figure 14-2, the digital input signal can be clamped to ground at the input of the Schmitt Trigger. The signal denoted SLEEP in the figure, is set by the MCU Sleep Controller in Power-down mode, Power-save mode, and Standby mode to avoid high power consumption if some input signals are left floating, or have an analog signal level close to $V_{CC}/2$.

SLEEP is overridden for port pins enabled as external interrupt pins. If the external interrupt request is not enabled, SLEEP is active also for these pins. SLEEP is also overridden by various other alternate functions as described in ”Alternate Port Functions” on page 89.

If a logic high level (“one”) is present on an asynchronous external interrupt pin configured as “Interrupt on Rising Edge, Falling Edge, or Any Logic Change on Pin” while the external interrupt is *not* enabled, the corresponding External Interrupt Flag will be set when resuming from the above mentioned Sleep mode, as the clamping in these sleep mode produces the requested logic change.

14.2.6 Unconnected Pins

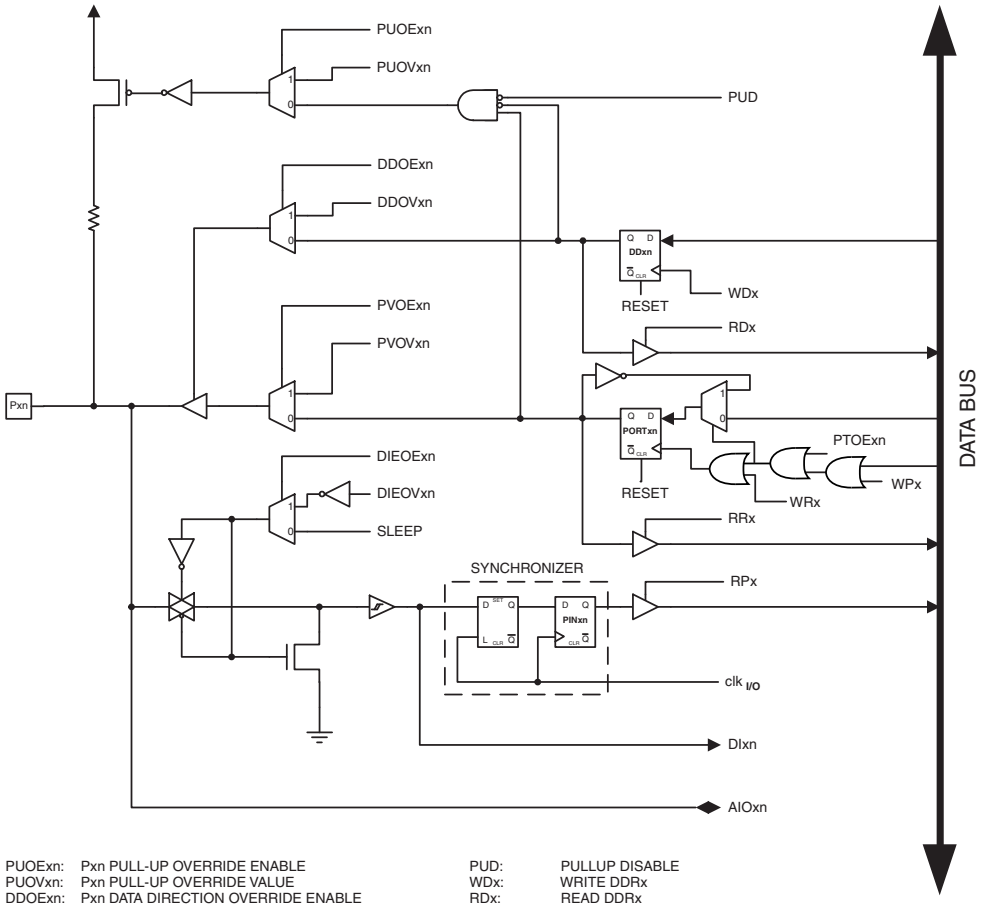
If some pins are unused, it is recommended to ensure that these pins have a defined level. Even though most of the digital inputs are disabled in the deep sleep modes as described above, floating inputs should be avoided to reduce current consumption in all other modes where the digital inputs are enabled (Reset, Active mode and Idle mode).

The simplest method to ensure a defined level of an unused pin, is to enable the internal pull-up. In this case, the pull-up will be disabled during reset. If low power consumption during reset is important, it is recommended to use an external pull-up or pull-down. Connecting unused pins directly to V_{CC} or GND is not recommended, since this may cause excessive currents if the pin is accidentally configured as an output.

14.3 Alternate Port Functions

Most port pins have alternate functions in addition to being general digital I/Os. Figure 14-5 shows how the port pin control signals from the simplified Figure 14-2 on page 85 can be overridden by alternate functions. The overriding signals may not be present in all port pins, but the figure serves as a generic description applicable to all port pins in the AVR microcontroller family.

Figure 14-5. Alternate Port Functions(1)



PUOExn:	Pxn PULL-UP OVERRIDE ENABLE	PUD:	PULLUP DISABLE
PUOVxn:	Pxn PULL-UP OVERRIDE VALUE	WDx:	WRITE DDRx
DDOExn:	Pxn DATA DIRECTION OVERRIDE ENABLE	RDx:	READ DDRx
DDOVxn:	Pxn DATA DIRECTION OVERRIDE VALUE	RRx:	READ PORTx REGISTER
PVOExn:	Pxn PORT VALUE OVERRIDE ENABLE	WRx:	WRITE PORTx
PVOVxn:	Pxn PORT VALUE OVERRIDE VALUE	RPx:	READ PORTx PIN
DIEOExn:	Pxn DIGITAL INPUT-ENABLE OVERRIDE ENABLE	WPx:	WRITE PINx
DIEOVxn:	Pxn DIGITAL INPUT-ENABLE OVERRIDE VALUE	$clk_{I/O}$:	I/O CLOCK
SLEEP:	SLEEP CONTROL	DIxn:	DIGITAL INPUT PIN n ON PORTx
PTOExn:	Pxn, PORT TOGGLE OVERRIDE ENABLE	AIOxn:	ANALOG INPUT/OUTPUT PIN n ON PORTx

Note: 1. WRx, WPx, WDx, RRx, RPx, and RDx are common to all pins within the same port. $clk_{I/O}$, SLEEP, and PUD are common to all ports. All other signals are unique for each pin.

Table 14-2 summarizes the function of the overriding signals. The pin and port indexes from Figure 14-5 on page 89 are not shown in the succeeding tables. The overriding signals are generated internally in the modules having the alternate function.

Table 14-2. Generic Description of Overriding Signals for Alternate Functions

Signal Name	Full Name	Description
PUOE	Pull-up Override Enable	If this signal is set, the pull-up enable is controlled by the PUOV signal. If this signal is cleared, the pull-up is enabled when {DDxn, PORTxn, PUD} = 0b010.
PUOV	Pull-up Override Value	If PUOE is set, the pull-up is enabled/disabled when PUOV is set/cleared, regardless of the setting of the DDxn, PORTxn, and PUD Register bits.
DDOE	Data Direction Override Enable	If this signal is set, the Output Driver Enable is controlled by the DDOV signal. If this signal is cleared, the Output driver is enabled by the DDxn Register bit.
DDOV	Data Direction Override Value	If DDOE is set, the Output Driver is enabled/disabled when DDOV is set/cleared, regardless of the setting of the DDxn Register bit.
PVOE	Port Value Override Enable	If this signal is set and the Output Driver is enabled, the port value is controlled by the PVOV signal. If PVOE is cleared, and the Output Driver is enabled, the port Value is controlled by the PORTxn Register bit.
PVOV	Port Value Override Value	If PVOE is set, the port value is set to PVOV, regardless of the setting of the PORTxn Register bit.
PTOE	Port Toggle Override Enable	If PTOE is set, the PORTxn Register bit is inverted.
DIEOE	Digital Input Enable Override Enable	If this bit is set, the Digital Input Enable is controlled by the DIEOV signal. If this signal is cleared, the Digital Input Enable is determined by MCU state (Normal mode, sleep mode).
DIEOV	Digital Input Enable Override Value	If DIEOE is set, the Digital Input is enabled/disabled when DIEOV is set/cleared, regardless of the MCU state (Normal mode, sleep mode).
DI	Digital Input	This is the Digital Input to alternate functions. In the figure, the signal is connected to the output of the Schmitt Trigger but before the synchronizer. Unless the Digital Input is used as a clock source, the module with the alternate function will use its own synchronizer.
AIO	Analog Input/Output	This is the Analog Input/output to/from alternate functions. The signal is connected directly to the pad, and can be used bi-directionally.

The following subsections shortly describe the alternate functions for each port, and relate the overriding signals to the alternate function. Refer to the alternate function description for further details.

14.3.1 Alternate Functions of Port B

The Port B pins with alternate functions are shown in Table 14-3.

Table 14-3. Port B Pins Alternate Functions

Port Pin	Alternate Functions
PB7	XTAL2 (Chip Clock Oscillator pin 2) TOSC2 (Timer Oscillator pin 2) PCINT7 (Pin Change Interrupt 7)
PB6	XTAL1 (Chip Clock Oscillator pin 1 or External clock input) TOSC1 (Timer Oscillator pin 1) PCINT6 (Pin Change Interrupt 6)
PB5	SCK (SPI Bus Master clock Input) PCINT5 (Pin Change Interrupt 5)
PB4	MISO (SPI Bus Master Input/Slave Output) PCINT4 (Pin Change Interrupt 4)
PB3	MOSI (SPI Bus Master Output/Slave Input) OC2A (Timer/Counter2 Output Compare Match A Output) PCINT3 (Pin Change Interrupt 3)
PB2	$\overline{SS}$ (SPI Bus Master Slave select) OC1B (Timer/Counter1 Output Compare Match B Output) PCINT2 (Pin Change Interrupt 2)
PB1	OC1A (Timer/Counter1 Output Compare Match A Output) PCINT1 (Pin Change Interrupt 1)
PB0	ICP1 (Timer/Counter1 Input Capture Input) CLKO (Divided System Clock Output) PCINT0 (Pin Change Interrupt 0)

The alternate pin configuration is as follows:

- **XTAL2/TOSC2/PCINT7 – Port B, Bit 7**

XTAL2: Chip clock Oscillator pin 2. Used as clock pin for crystal Oscillator or Low-frequency crystal Oscillator. When used as a clock pin, the pin can not be used as an I/O pin.

TOSC2: Timer Oscillator pin 2. Used only if internal calibrated RC Oscillator is selected as chip clock source, and the asynchronous timer is enabled by the correct setting in ASSR. When the AS2 bit in ASSR is set (one) and the EXCLK bit is cleared (zero) to enable asynchronous clocking of Timer/Counter2 using the Crystal Oscillator, pin PB7 is disconnected from the port, and becomes the inverting output of the Oscillator amplifier. In this mode, a crystal Oscillator is connected to this pin, and the pin cannot be used as an I/O pin.

PCINT7: Pin Change Interrupt source 7. The PB7 pin can serve as an external interrupt source.

If PB7 is used as a clock pin, DDB7, PORTB7 and PINB7 will all read 0.

- **XTAL1/TOSC1/PCINT6 – Port B, Bit 6**

XTAL1: Chip clock Oscillator pin 1. Used for all chip clock sources except internal calibrated RC Oscillator. When used as a clock pin, the pin can not be used as an I/O pin.

TOSC1: Timer Oscillator pin 1. Used only if internal calibrated RC Oscillator is selected as chip clock source, and the asynchronous timer is enabled by the correct setting in ASSR. When the AS2 bit in ASSR is set (one) to enable asynchronous clocking of Timer/Counter2, pin PB6 is disconnected from the port, and becomes the input of the inverting Oscillator amplifier. In this mode, a crystal Oscillator is connected to this pin, and the pin can not be used as an I/O pin.

PCINT6: Pin Change Interrupt source 6. The PB6 pin can serve as an external interrupt source.

If PB6 is used as a clock pin, DDB6, PORTB6 and PINB6 will all read 0.

- **SCK/PCINT5 – Port B, Bit 5**

SCK: Master Clock output, Slave Clock input pin for SPI channel. When the SPI is enabled as a Slave, this pin is configured as an input regardless of the setting of DDB5. When the SPI is enabled as a Master, the data direction of this pin is controlled by DDB5. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTB5 bit.

PCINT5: Pin Change Interrupt source 5. The PB5 pin can serve as an external interrupt source.

- **MISO/PCINT4 – Port B, Bit 4**

MISO: Master Data input, Slave Data output pin for SPI channel. When the SPI is enabled as a Master, this pin is configured as an input regardless of the setting of DDB4. When the SPI is enabled as a Slave, the data direction of this pin is controlled by DDB4. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTB4 bit.

PCINT4: Pin Change Interrupt source 4. The PB4 pin can serve as an external interrupt source.

- **MOSI/OC2/PCINT3 – Port B, Bit 3**

MOSI: SPI Master Data output, Slave Data input for SPI channel. When the SPI is enabled as a Slave, this pin is configured as an input regardless of the setting of DDB3. When the SPI is enabled as a Master, the data direction of this pin is controlled by DDB3. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTB3 bit.

OC2, Output Compare Match Output: The PB3 pin can serve as an external output for the Timer/Counter2 Compare Match. The PB3 pin has to be configured as an output (DDB3 set (one)) to serve this function. The OC2 pin is also the output pin for the PWM mode timer function.

PCINT3: Pin Change Interrupt source 3. The PB3 pin can serve as an external interrupt source.

- **$\overline{\text{SS}}$/OC1B/PCINT2 – Port B, Bit 2**

$\overline{\text{SS}}$: Slave Select input. When the SPI is enabled as a Slave, this pin is configured as an input regardless of the setting of DDB2. As a Slave, the SPI is activated when this pin is driven low. When the SPI is enabled as a Master, the data direction of this pin is controlled by DDB2. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTB2 bit.

OC1B, Output Compare Match output: The PB2 pin can serve as an external output for the Timer/Counter1 Compare Match B. The PB2 pin has to be configured as an output (DDB2 set (one)) to serve this function. The OC1B pin is also the output pin for the PWM mode timer function.

PCINT2: Pin Change Interrupt source 2. The PB2 pin can serve as an external interrupt source.

- **OC1A/PCINT1 – Port B, Bit 1**

OC1A, Output Compare Match output: The PB1 pin can serve as an external output for the Timer/Counter1 Compare Match A. The PB1 pin has to be configured as an output (DDB1 set (one)) to serve this function. The OC1A pin is also the output pin for the PWM mode timer function.

PCINT1: Pin Change Interrupt source 1. The PB1 pin can serve as an external interrupt source.

- **ICP1/CLKO/PCINT0 – Port B, Bit 0**

ICP1, Input Capture Pin: The PB0 pin can act as an Input Capture Pin for Timer/Counter1.

CLKO, Divided System Clock: The divided system clock can be output on the PB0 pin. The divided system clock will be output if the CKOUT Fuse is programmed, regardless of the PORTB0 and DDB0 settings. It will also be output during reset.

PCINT0: Pin Change Interrupt source 0. The PB0 pin can serve as an external interrupt source.

Table 14-4 and Table 14-5 on page 94 relate the alternate functions of Port B to the overriding signals shown in Figure 14-5 on page 89. SPI MSTR INPUT and SPI SLAVE OUTPUT constitute the MISO signal, while MOSI is divided into SPI MSTR OUTPUT and SPI SLAVE INPUT.

Table 14-4. Overriding Signals for Alternate Functions in PB7...PB4

Signal Name	PB7/XTAL2/ TOSC2/PCINT7(1)	PB6/XTAL1/ TOSC1/PCINT6(1)	PB5/SCK/ PCINT5	PB4/MISO/ PCINT4
PUOE	$\overline{INTRC}$ • $\overline{EXTCK}$+ AS2	$\overline{INTRC}$ + AS2	SPE • $\overline{MSTR}$	SPE • MSTR
PUOV	0	0	PORTB5 • $\overline{PUD}$	PORTB4 • $\overline{PUD}$
DDOE	$\overline{INTRC}$ • $\overline{EXTCK}$+ AS2	$\overline{INTRC}$ + AS2	SPE • $\overline{MSTR}$	SPE • MSTR
DDOV	0	0	0	0
PVOE	0	0	SPE • MSTR	SPE • $\overline{MSTR}$
PVOV	0	0	SCK OUTPUT	SPI SLAVE OUTPUT
DIEOE	$\overline{INTRC}$ • $\overline{EXTCK}$ + AS2 + PCINT7 • PCIE0	$\overline{INTRC}$ + AS2 + PCINT6 • PCIE0	PCINT5 • PCIE0	PCINT4 • PCIE0
DIEOV	(INTRC + EXTCK) • $\overline{AS2}$	INTRC • $\overline{AS2}$	1	1
DI	PCINT7 INPUT	PCINT6 INPUT	PCINT5 INPUT SCK INPUT	PCINT4 INPUT SPI MSTR INPUT
AIO	Oscillator Output	Oscillator/Clock Input	–	–

Notes: 1. INTRC means that one of the internal RC Oscillators are selected (by the CKSEL fuses), EXTCK means that external clock is selected (by the CKSEL fuses)

Table 14-5. Overriding Signals for Alternate Functions in PB3...PB0

Signal Name	PB3/MOSI/ OC2/PCINT3	PB2/$\overline{SS}$/ OC1B/PCINT2	PB1/OC1A/ PCINT1	PB0/ICP1/ PCINT0
PUOE	SPE • $\overline{MSTR}$	SPE • $\overline{MSTR}$	0	0
PUOV	PORTB3 • $\overline{PUD}$	PORTB2 • $\overline{PUD}$	0	0
DDOE	SPE • $\overline{MSTR}$	SPE • $\overline{MSTR}$	0	0
DDOV	0	0	0	0
PVOE	SPE • MSTR + OC2A ENABLE	OC1B ENABLE	OC1A ENABLE	0
PVOV	SPI MSTR OUTPUT + OC2A	OC1B	OC1A	0
DIEOE	PCINT3 • PCIE0	PCINT2 • PCIE0	PCINT1 • PCIE0	PCINT0 • PCIE0
DIEOV	1	1	1	1
DI	PCINT3 INPUT SPI SLAVE INPUT	PCINT2 INPUT SPI $\overline{SS}$	PCINT1 INPUT	PCINT0 INPUT ICP1 INPUT
AIO	–	–	–	–

14.3.2 Alternate Functions of Port C

The Port C pins with alternate functions are shown in Table 14-6.

Table 14-6. Port C Pins Alternate Functions

Port Pin	Alternate Function
PC6	$\overline{RESET}$ (Reset pin) PCINT14 (Pin Change Interrupt 14)
PC5	ADC5 (ADC Input Channel 5) SCL (2-wire Serial Bus Clock Line) PCINT13 (Pin Change Interrupt 13)
PC4	ADC4 (ADC Input Channel 4) SDA (2-wire Serial Bus Data Input/Output Line) PCINT12 (Pin Change Interrupt 12)
PC3	ADC3 (ADC Input Channel 3) PCINT11 (Pin Change Interrupt 11)
PC2	ADC2 (ADC Input Channel 2) PCINT10 (Pin Change Interrupt 10)
PC1	ADC1 (ADC Input Channel 1) PCINT9 (Pin Change Interrupt 9)
PC0	ADC0 (ADC Input Channel 0) PCINT8 (Pin Change Interrupt 8)

The alternate pin configuration is as follows:

- **$\overline{\text{RESET}}$/PCINT14 – Port C, Bit 6**

$\overline{\text{RESET}}$, Reset pin: When the RSTDISBL Fuse is programmed, this pin functions as a normal I/O pin, and the part will have to rely on Power-on Reset and Brown-out Reset as its reset sources. When the RSTDISBL Fuse is unprogrammed, the reset circuitry is connected to the pin, and the pin can not be used as an I/O pin.

If PC6 is used as a reset pin, DDC6, PORTC6 and PINC6 will all read 0.

PCINT14: Pin Change Interrupt source 14. The PC6 pin can serve as an external interrupt source.

- **SCL/ADC5/PCINT13 – Port C, Bit 5**

SCL, 2-wire Serial Interface Clock: When the TWEN bit in TWCR is set (one) to enable the 2-wire Serial Interface, pin PC5 is disconnected from the port and becomes the Serial Clock I/O pin for the 2-wire Serial Interface. In this mode, there is a spike filter on the pin to suppress spikes shorter than 50 ns on the input signal, and the pin is driven by an open drain driver with slew-rate limitation.

PC5 can also be used as ADC input Channel 5. Note that ADC input channel 5 uses digital power.

PCINT13: Pin Change Interrupt source 13. The PC5 pin can serve as an external interrupt source.

- **SDA/ADC4/PCINT12 – Port C, Bit 4**

SDA, 2-wire Serial Interface Data: When the TWEN bit in TWCR is set (one) to enable the 2-wire Serial Interface, pin PC4 is disconnected from the port and becomes the Serial Data I/O pin for the 2-wire Serial Interface. In this mode, there is a spike filter on the pin to suppress spikes shorter than 50 ns on the input signal, and the pin is driven by an open drain driver with slew-rate limitation.

PC4 can also be used as ADC input Channel 4. Note that ADC input channel 4 uses digital power.

PCINT12: Pin Change Interrupt source 12. The PC4 pin can serve as an external interrupt source.

- **ADC3/PCINT11 – Port C, Bit 3**

PC3 can also be used as ADC input Channel 3. Note that ADC input channel 3 uses analog power.

PCINT11: Pin Change Interrupt source 11. The PC3 pin can serve as an external interrupt source.

- **ADC2/PCINT10 – Port C, Bit 2**

PC2 can also be used as ADC input Channel 2. Note that ADC input channel 2 uses analog power.

PCINT10: Pin Change Interrupt source 10. The PC2 pin can serve as an external interrupt source.

- **ADC1/PCINT9 – Port C, Bit 1**

PC1 can also be used as ADC input Channel 1. Note that ADC input channel 1 uses analog power.

PCINT9: Pin Change Interrupt source 9. The PC1 pin can serve as an external interrupt source.

- **ADC0/PCINT8 – Port C, Bit 0**

PC0 can also be used as ADC input Channel 0. Note that ADC input channel 0 uses analog power.

PCINT8: Pin Change Interrupt source 8. The PC0 pin can serve as an external interrupt source.

Table 14-7 and Table 14-8 relate the alternate functions of Port C to the overriding signals shown in Figure 14-5 on page 89.

Table 14-7. Overriding Signals for Alternate Functions in PC6...PC4[1]

Signal Name	PC6/$\overline{\text{RESET}}$/PCINT14	PC5/SCL/ADC5/PCINT13	PC4/SDA/ADC4/PCINT12
PUOE	RSTDISBL	TWEN	TWEN
PUOV	1	PORTC5 • $\overline{\text{PUD}}$	PORTC4 • $\overline{\text{PUD}}$
DDOE	RSTDISBL	TWEN	TWEN
DDOV	0	SCL_OUT	SDA_OUT
PVOE	0	TWEN	TWEN
PVOV	0	0	0
DIEOE	RSTDISBL + PCINT14 • PCIE1	PCINT13 • PCIE1 + ADC5D	PCINT12 • PCIE1 + ADC4D
DIEOV	RSTDISBL	PCINT13 • PCIE1	PCINT12 • PCIE1
DI	PCINT14 INPUT	PCINT13 INPUT	PCINT12 INPUT
AIO	RESET INPUT	ADC5 INPUT / SCL INPUT	ADC4 INPUT / SDA INPUT

Note: 1. When enabled, the 2-wire Serial Interface enables slew-rate controls on the output pins PC4 and PC5. This is not shown in the figure. In addition, spike filters are connected between the AIO outputs shown in the port figure and the digital logic of the TWI module.

Table 14-8. Overriding Signals for Alternate Functions in PC3...PC0

Signal Name	PC3/ADC3/ PCINT11	PC2/ADC2/ PCINT10	PC1/ADC1/ PCINT9	PC0/ADC0/ PCINT8
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	0	0	0	0
PVOV	0	0	0	0
DIEOE	PCINT11 • PCIE1 + ADC3D	PCINT10 • PCIE1 + ADC2D	PCINT9 • PCIE1 + ADC1D	PCINT8 • PCIE1 + ADC0D
DIEOV	PCINT11 • PCIE1	PCINT10 • PCIE1	PCINT9 • PCIE1	PCINT8 • PCIE1
DI	PCINT11 INPUT	PCINT10 INPUT	PCINT9 INPUT	PCINT8 INPUT
AIO	ADC3 INPUT	ADC2 INPUT	ADC1 INPUT	ADC0 INPUT

14.3.3 Alternate Functions of Port D

The Port D pins with alternate functions are shown in Table 14-9.

Table 14-9. Port D Pins Alternate Functions

Port Pin	Alternate Function
PD7	AIN1 (Analog Comparator Negative Input) PCINT23 (Pin Change Interrupt 23)
PD6	AIN0 (Analog Comparator Positive Input) OC0A (Timer/Counter0 Output Compare Match A Output) PCINT22 (Pin Change Interrupt 22)
PD5	T1 (Timer/Counter 1 External Counter Input) OC0B (Timer/Counter0 Output Compare Match B Output) PCINT21 (Pin Change Interrupt 21)
PD4	XCK (USART External Clock Input/Output) T0 (Timer/Counter 0 External Counter Input) PCINT20 (Pin Change Interrupt 20)
PD3	INT1 (External Interrupt 1 Input) OC2B (Timer/Counter2 Output Compare Match B Output) PCINT19 (Pin Change Interrupt 19)
PD2	INT0 (External Interrupt 0 Input) PCINT18 (Pin Change Interrupt 18)
PD1	TXD (USART Output Pin) PCINT17 (Pin Change Interrupt 17)
PD0	RXD (USART Input Pin) PCINT16 (Pin Change Interrupt 16)

The alternate pin configuration is as follows:

- **AIN1/OC2B/PCINT23 – Port D, Bit 7**

AIN1, Analog Comparator Negative Input. Configure the port pin as input with the internal pull-up switched off to avoid the digital port function from interfering with the function of the Analog Comparator.

PCINT23: Pin Change Interrupt source 23. The PD7 pin can serve as an external interrupt source.

- **AIN0/OC0A/PCINT22 – Port D, Bit 6**

AIN0, Analog Comparator Positive Input. Configure the port pin as input with the internal pull-up switched off to avoid the digital port function from interfering with the function of the Analog Comparator.

OC0A, Output Compare Match output: The PD6 pin can serve as an external output for the Timer/Counter0 Compare Match A. The PD6 pin has to be configured as an output (DDD6 set (one)) to serve this function. The OC0A pin is also the output pin for the PWM mode timer function.

PCINT22: Pin Change Interrupt source 22. The PD6 pin can serve as an external interrupt source.

- **T1/OC0B/PCINT21 – Port D, Bit 5**

T1, Timer/Counter1 counter source.

OC0B, Output Compare Match output: The PD5 pin can serve as an external output for the Timer/Counter0 Compare Match B. The PD5 pin has to be configured as an output (DDD5 set (one)) to serve this function. The OC0B pin is also the output pin for the PWM mode timer function.

PCINT21: Pin Change Interrupt source 21. The PD5 pin can serve as an external interrupt source.

- **XCK/T0/PCINT20 – Port D, Bit 4**

XCK, USART external clock.

T0, Timer/Counter0 counter source.

PCINT20: Pin Change Interrupt source 20. The PD4 pin can serve as an external interrupt source.

- **INT1/OC2B/PCINT19 – Port D, Bit 3**

INT1, External Interrupt source 1: The PD3 pin can serve as an external interrupt source.

OC2B, Output Compare Match output: The PD3 pin can serve as an external output for the Timer/Counter0 Compare Match B. The PD3 pin has to be configured as an output (DDD3 set (one)) to serve this function. The OC2B pin is also the output pin for the PWM mode timer function.

PCINT19: Pin Change Interrupt source 19. The PD3 pin can serve as an external interrupt source.

- **INT0/PCINT18 – Port D, Bit 2**

INT0, External Interrupt source 0: The PD2 pin can serve as an external interrupt source.

PCINT18: Pin Change Interrupt source 18. The PD2 pin can serve as an external interrupt source.

- **TXD/PCINT17 – Port D, Bit 1**

TXD, Transmit Data (Data output pin for the USART). When the USART Transmitter is enabled, this pin is configured as an output regardless of the value of DDD1.

PCINT17: Pin Change Interrupt source 17. The PD1 pin can serve as an external interrupt source.

- **RXD/PCINT16 – Port D, Bit 0**

RXD, Receive Data (Data input pin for the USART). When the USART Receiver is enabled this pin is configured as an input regardless of the value of DDD0. When the USART forces this pin to be an input, the pull-up can still be controlled by the PORTD0 bit.

PCINT16: Pin Change Interrupt source 16. The PD0 pin can serve as an external interrupt source.

Table 14-10 and Table 14-11 relate the alternate functions of Port D to the overriding signals shown in Figure 14-5 on page 89.

Table 14-10. Overriding Signals for Alternate Functions PD7...PD4

Signal Name	PD7/AIN1 /PCINT23	PD6/AIN0/ OC0A/PCINT22	PD5/T1/OC0B/ PCINT21	PD4/XCK/ T0/PCINT20
PUOE	0	0	0	0
PUO	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	0	OC0A ENABLE	OC0B ENABLE	UMSEL
PVOV	0	OC0A	OC0B	XCK OUTPUT
DIEOE	PCINT23 • PCIE2	PCINT22 • PCIE2	PCINT21 • PCIE2	PCINT20 • PCIE2
DIEOV	1	1	1	1
DI	PCINT23 INPUT	PCINT22 INPUT	PCINT21 INPUT T1 INPUT	PCINT20 INPUT XCK INPUT T0 INPUT
AIO	AIN1 INPUT	AIN0 INPUT	–	–

Table 14-11. Overriding Signals for Alternate Functions in PD3...PD0

Signal Name	PD3/OC2B/INT1/ PCINT19	PD2/INT0/ PCINT18	PD1/TXD/ PCINT17	PD0/RXD/ PCINT16
PUOE	0	0	TXEN	RXEN
PUO	0	0	0	PORTD0 • $\overline{PUD}$
DDOE	0	0	TXEN	RXEN
DDOV	0	0	1	0
PVOE	OC2B ENABLE	0	TXEN	0
PVOV	OC2B	0	TXD	0
DIEOE	INT1 ENABLE + PCINT19 • PCIE2	INT0 ENABLE + PCINT18 • PCIE1	PCINT17 • PCIE2	PCINT16 • PCIE2
DIEOV	1	1	1	1
DI	PCINT19 INPUT INT1 INPUT	PCINT18 INPUT INT0 INPUT	PCINT17 INPUT	PCINT16 INPUT RXD
AIO	–	–	–	–

14.4 Register Description

14.4.1 MCUCR – MCU Control Register

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

Notes: 1. BODS and BODSE only available for picoPower devices ATmega48PA/88PA/168PA/328P

- **Bit 4 – PUD: Pull-up Disable**

When this bit is written to one, the pull-ups in the I/O ports are disabled even if the DDxn and PORTxn Registers are configured to enable the pull-ups ({DDxn, PORTxn} = 0b01). See ”Configuring the Pin” on page 85 for more details about this feature.

14.4.2 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	

14.4.3 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	

14.4.4 PINB – The Port B Input Pins Address[1]

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

14.4.5 PORTC – The Port C Data Register

Bit	7	6	5	4	3	2	1	0	
0x08 (0x28)	–	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	PORTC
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

14.4.6 DDRC – The Port C Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x07 (0x27)	–	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

14.4.7 PINC – The Port C Input Pins Address[1]

Bit	7	6	5	4	3	2	1	0	
0x06 (0x26)	–	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	PINC
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

14.4.8 PORTD – The Port D Data Register

Bit	7	6	5	4	3	2	1	0	
0x0B (0x2B)	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
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	

14.4.9 DDRD – The Port D Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x0A (0x2A)	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
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	

14.4.10 PIND – The Port D Input Pins Address[1]

Bit	7	6	5	4	3	2	1	0	
0x09 (0x29)	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

Note: 1. Writing to the pin register provides toggle functionality for IO (see ”Toggling the Pin” on page 85)