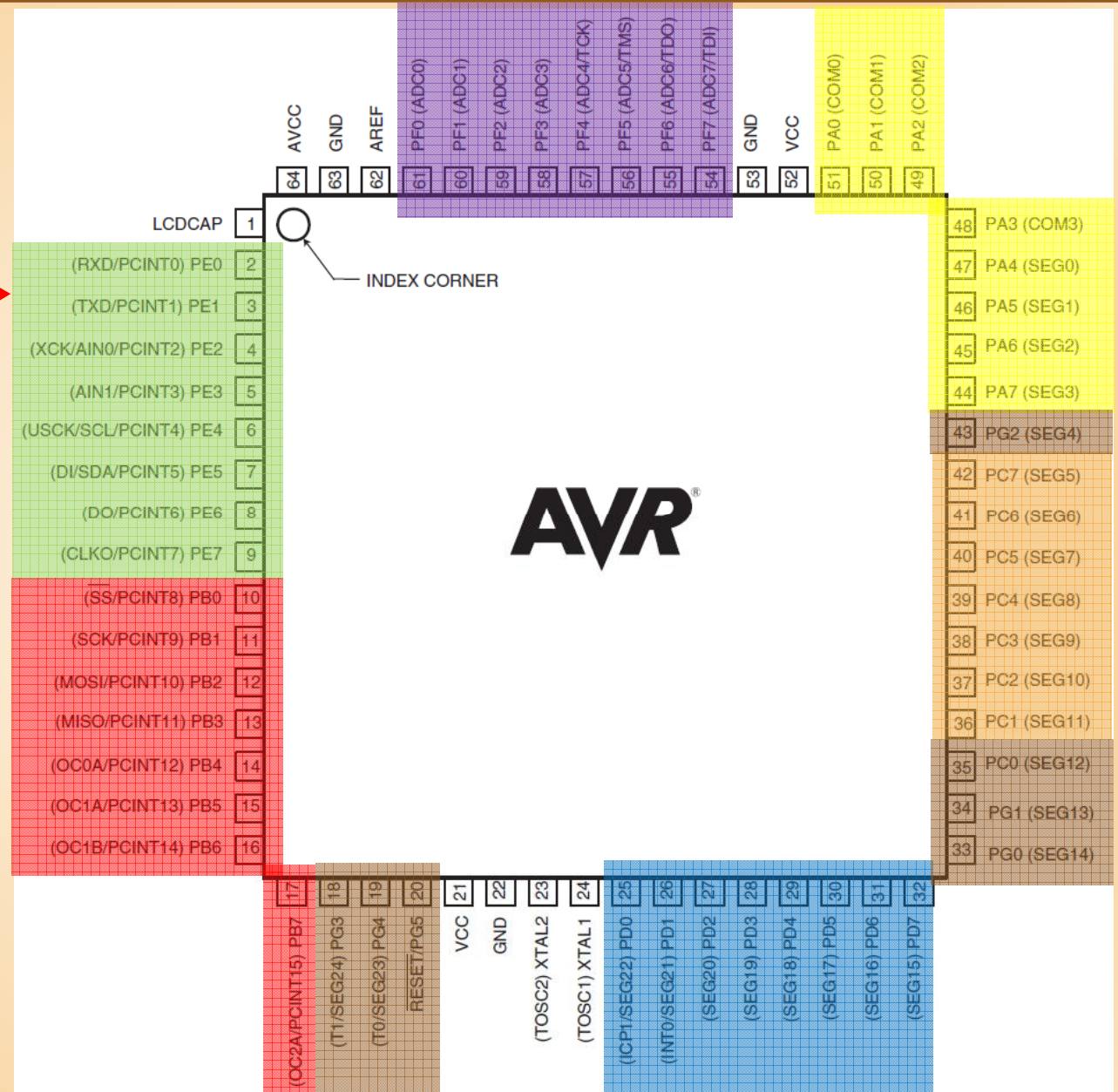


AVR IO Ports: Chip Pinout

- 2nd page of the data sheet; AVR IO pin map is shown here:
- Note, all IO functionality discussed on the following slides is described in the Resource directory on Blackboard: “AVR Butterfly-ATMega169P Datasheet.pdf”
- See Section #13 starting on page 65 through page 67. Alternate Pin functions are described on pages 73 - 87.
- Also – review Resources



AVR IO Ports – Notes (see diag Slide #4):

- All AVR ports have true Read-Modify-Write functionality
- This means each pin on a port can be modified without unintentionally changing any other pin (using CBI and SBI)
- Three I/O memory address locations are allocated for each port.
 - Data Register –PORTx (Read/Write)
 - Data Direction Register –DDRx (Read/Write)
 - Port Input Pins –PINx (Read only)
- ATmega169P has 7 IO ports (A,B,C, D,E,F,G)
- Pxn is representing nth bit in Port x
 - (e.g PA6 == 6 th bit of PORT A)

Exerpt from

"m169Pdef.inc"

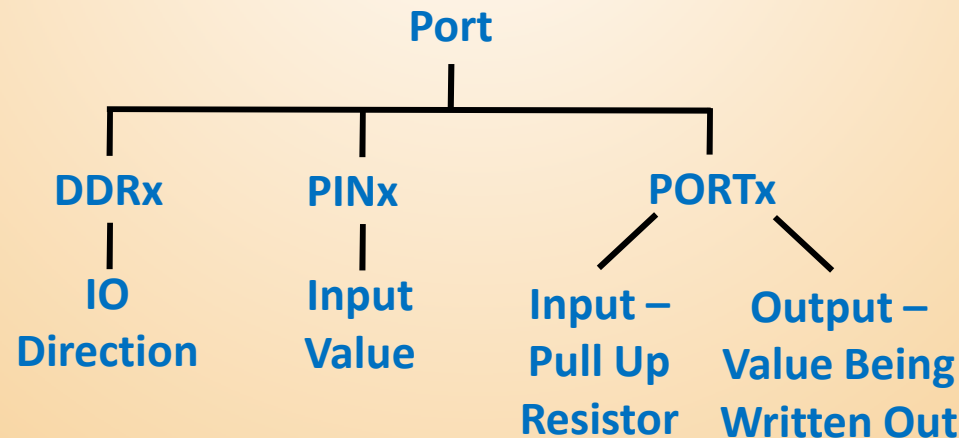
```
.equ  PINB = 0x03
```

```
.equ  DDRB = 0x04
```

```
.equ  PORTB = 0x05
```

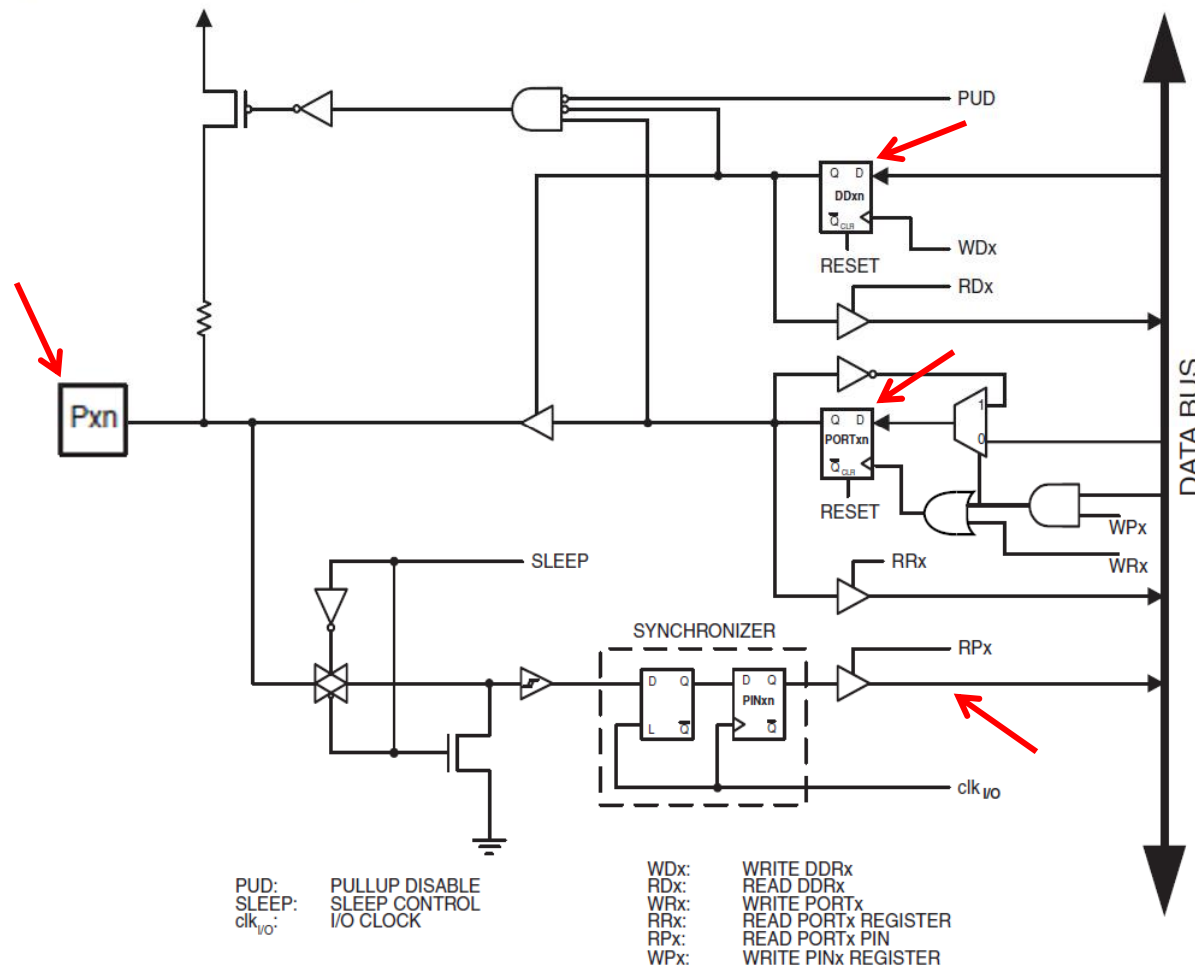
AVR IO Ports – Notes (cont - see diag Slide #4):

- 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 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).
- If **PORTxn** is written logic one when the pin is configured as an input pin, the pull-up resistor is activated.
- **Special Feature note:** writing a logic one to a bit in the **PINx** Register, will result in a toggle in the corresponding bit in the Data



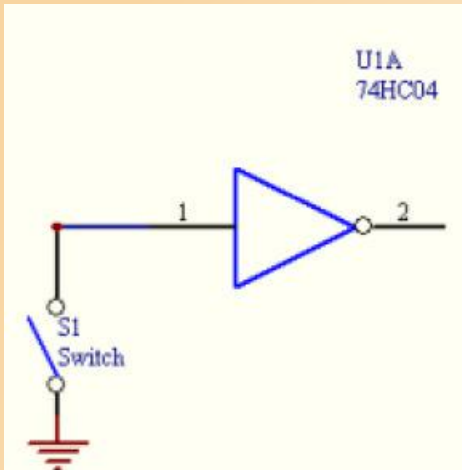
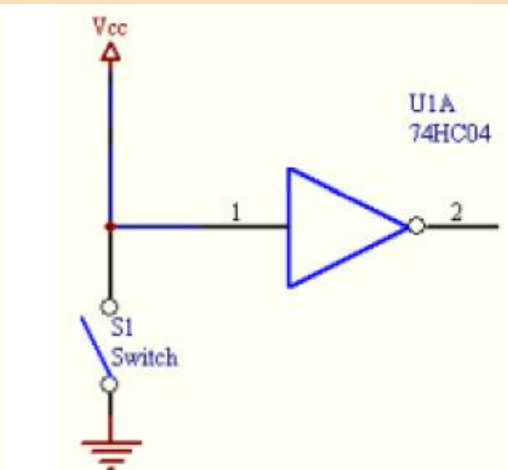
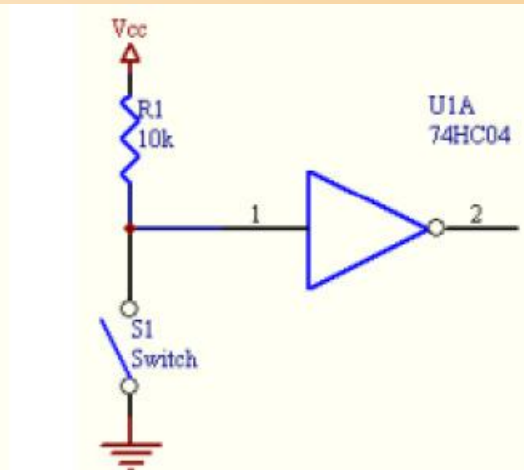
AVR IO Ports – General Digital I/O:

Figure 13-2. General Digital I/O

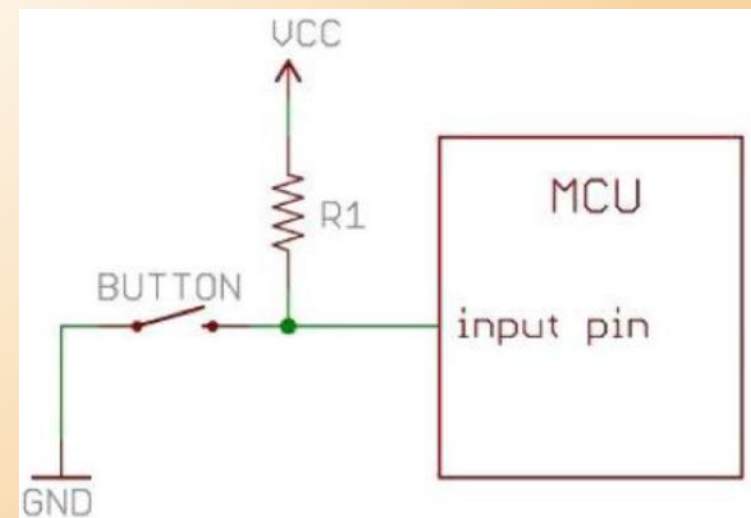


Note: 1. WRx, WPx, WDx, RRx, RPx, and RDx are common to all pins within the same port. clk_{I/O}, SLEEP, and PUD are common to all ports.

AVR IO Ports – Pull Up Resistor

| | | |
|---|--|---|
|  |  |  |
| Off: High Impedance | Off: One | Off: One |
| On: Zero | On: Zero | On: Zero |
| Potential Problem? | Bad Idea: why? | Good! |

- With a pull-up resistor, the input pin will read a high state when the button is open. When the button closes, it connects the input pin directly to ground, thus reading a low state.
- AVR has internal pull-up –no need to implement it outside of MCU (see next slide)



Slide: 6

AVR IO Ports – Programming I/O Ports –Assembly - pg1

;Using CBI and SBI to write to ports

```
SBI DDRB, 3 ;make bit 3 as output bit on PORTB
CBI PORTB, 7 ;make PORTB bit 7 as "0"
SBI PORTB, 4 ;make PORTB bit 4 as "1"
```

;Using OUT instruction to write to ports

```
LDI R18, 0b00100000
OUT DDRB, R18 ;make bit 5 as output bit on PORTB
LDI R18, 0b00000000
OUT PORTB, R18 ;make PORTB bit 5 as "0"
LDI R18, 0b00100000
OUT PORTB, R18 ;make PORTB bit 5 as "1"
```

; INPUT EXAMPLE

```
IN R18, PINB
```

A common error here is that OUT DDRB, R18 doesn't just set bit 1 to a "1", it also sets all of the other bits to "0", so please use the following method to set bits instead

AVR IO Ports – Programming I/O Ports –Assembly – pg2

Toggling

```
;set pin 4 of B port as output  
; without affecting other bits
```

```
IN R18,DDRB  
ORI R18, 0b00010000  
OUT DDRB, R18
```

```
;set pin 5 of B port to 1  
; without affecting other bits
```

```
IN R18,PORTB  
ORI R18, 0b00100000  
OUT PORTB, R18
```

```
;clear pin 4 of B port to 0  
; without affecting other bits
```

```
IN R18,PORTB  
ANDI R18, 0b11101111  
OUT PORTB, R18
```

```
;toggle pin 1 of B (no eori available)  
; without affecting other bits
```

```
IN R18,PORTB  
LDI R19,0b00000010  
EOR R18, R19  
OUT PORTB, R18
```

```
;toggle pin 1 of B using PINB "input  
; write trick"
```

```
OUT PINB, 0b00000010
```

Changing Multiple Bits

So, **sbi** and **cbi** are more convenient and allowed if only changing one bit at a time. If multiple bits need to be set at the same you can't use **sbi,cbi**

```
;set pin 7,3 of B port to 1 at same time  
; without affecting other bits
```

```
IN R18,PORTB  
ORI R18, 0b10001000  
OUT DDRB, R18
```


AVR IO Ports – Ports in Butterfly Board

- Doc4271.pdf : The AVR Butterfly user guide
 - On the web at: <http://www.atmel.com/Images/doc4271.pdf>
 - On Blackboard under the 'Resources' Folder: AVR Butterfly-Butterfly User Guide.pdf

Figure 3-1. Connectors

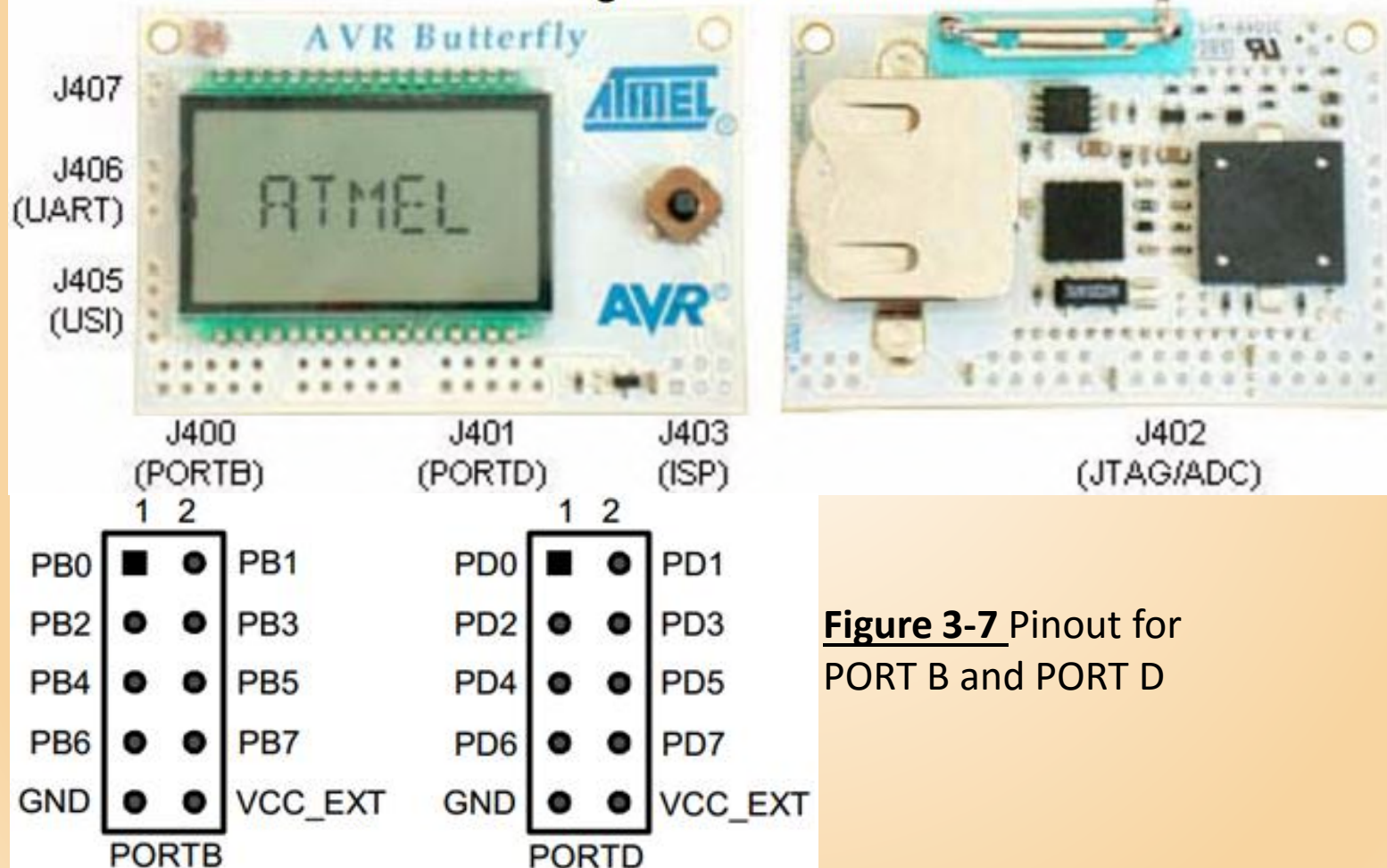
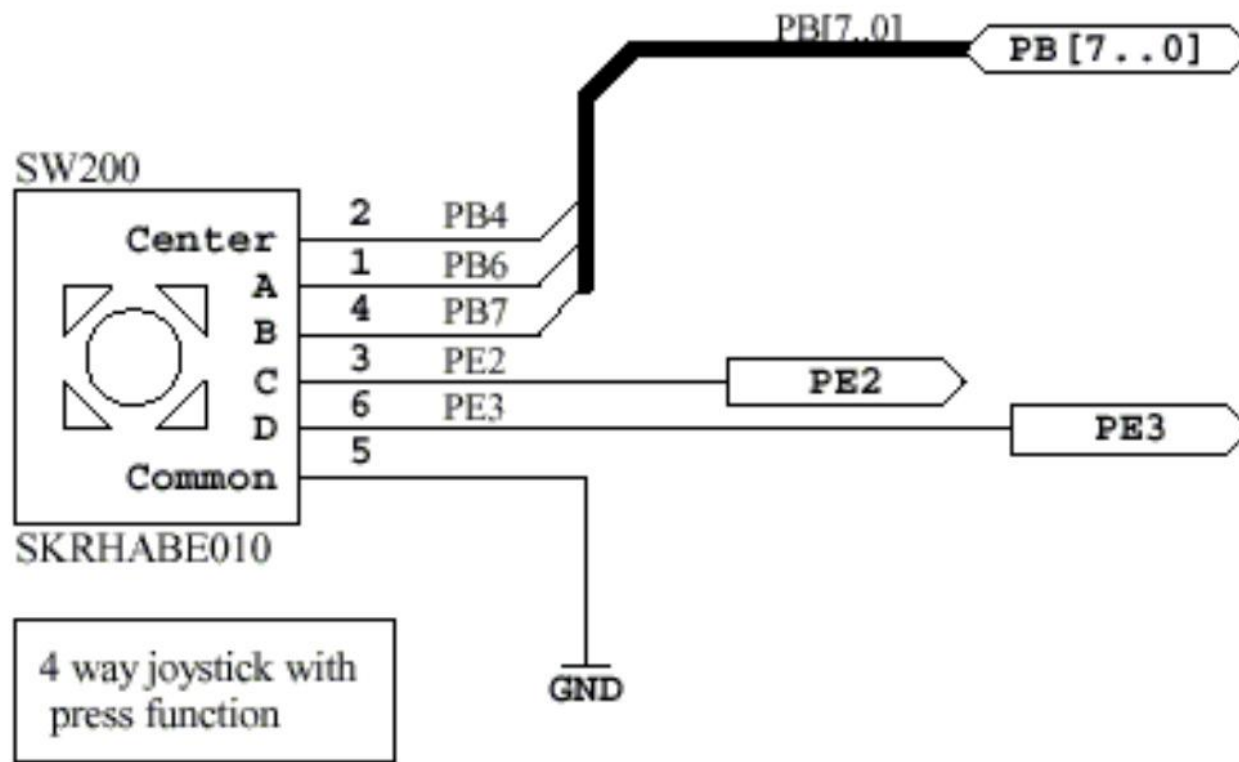


Figure 3-7 Pinout for
PORT B and PORT D

AVR IO Ports – Butterfly Ports (cont)

- The common line of all directions is GND. This means internal pull-up must be enabled on ATmega169P to detect the input.

Figure 3-11. Joystick Schematic



AVR IO Ports – SKRHABE010

- Googled “SKRHABE010”
- Mouser page has the data sheet from ALPS:

