

Embedded Systems

Zusammenfassung

Wise24/25

Inhaltsverzeichnis

1	I/O Ports	2
1.1	DDRn	2
1.2	Digitale Outports	3
1.2.1	PORTn	3
1.3	Digitale Inports	4
1.3.1	PINn	4

1 I/O Ports

Ports werden durch 8 Bit Register konfiguriert.

Data Direction Register (DDR)

0 = Input

1 = Output

Default Wert ist **0** bzw. **Input**. Damit wird verhindert, dass ein Verbraucher ausversehen zu viel Strom zieht und damit den Mikrocontroller beschädigt.

1.1 DDRn

DDRB - 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	

→ **DDRn** Register legt die Richtung fest (Input oder Output)!

1.2 Digitale Outputs

Wurde ein Pin als Output definiert, so kann man ihn über das **PORTn** Register steuern.

1.2.1 PORTn

PORTB - 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	

→ **PORTn** Register definiert, ob Output high oder low!

Beispiel: High Pegel auf PORTB3

```
#include <avr/io.h>
```

```
int main(void){  
    DDRB |= (1<<DDB3); //Pin 3 in DDRB auf Output setzen  
    PORTB |= (1<<PORTB3); //High Pegel auf Pin 3 definieren  
    while(1){  
        }  
}
```

Hinter **DDB3** und **PORTB3** steckt nur eine Zahl, um die eine 1 nach links verschoben wird im Register.

1.3 Digitale Inports

Wurde ein Pin als Input in DDRn definiert, so kann man ihn über das **PINn** Register abfragen.

1.3.1 PINn

PINB - Port B Input Pins Address

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	

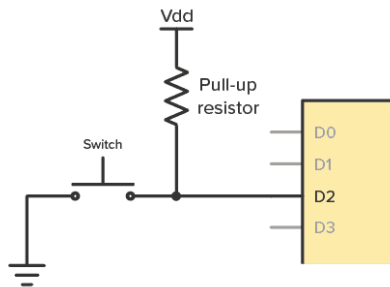
Beispiel: Pin 3 von DDRB auslesen

```
#include <avr/io.h>
```

```
int main(void){
    DDRB |= (0<<DDB0); //Pin 0 in DDRB auf Input setzen (Default state)

    while(1){
        if (PINB & (1<<PINB0)){ //Prüfen, ob 0 an PINB0 anliegt
            //Code ...
        }
    }
}
```

ATmega328p hat bei allen Ports einen interenen Pull-Up Widerstand!



Taster brückt auf Masse! Deshalb wird im Code nach 0 abfragen!