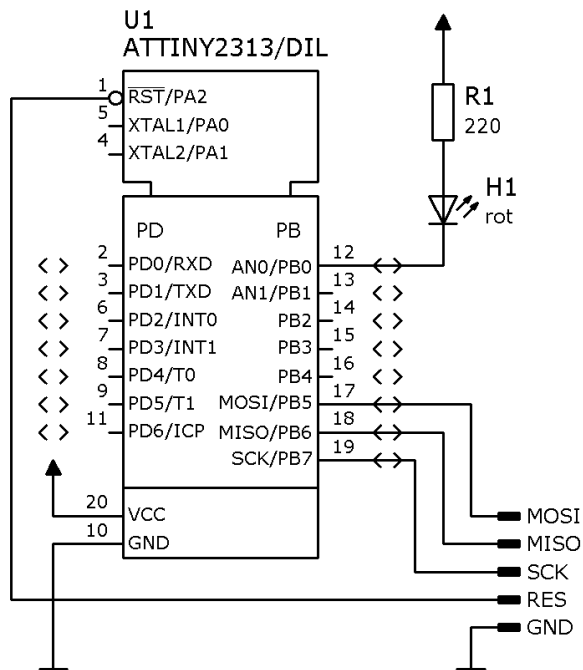


Aufgabe 1

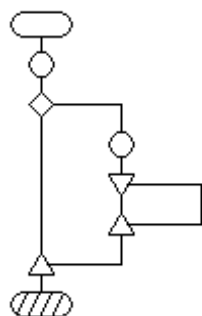
Eine Led blinkt

Schaltplan Blinker



Ablaufdiagramm Blinker 1.0

Dieses Ablaufdiagramm zeigt die grundsätzliche Struktur eines Programms um eine Led blinken zu lassen.



Blink1.0

Variablen (Portpin, i) initialisieren

solange wahr

PortPin invertieren

wiederhole i mal

nächstes i

wiederhole

Ende

Dabei sind einige Vorüberlegungen notwendig.

- Wie greife ich auf ein einzelnes Bit zu?
- Wie wird ein Portpin initialisiert?
- Wie lässt sich ein Portpin invertieren?
- Wie weit muss "i" gezählt werden, damit eine Wartezeit entsteht, die gut sichtbar ist?

Schreiben Sie das Programm um die Led blinken zu lassen.

Bitmaske

An Port B soll das Bit Null gesetzt werden. Die anderen Bits an diesem Port sollen dabei nicht verändert werden.

| X | Y | X oder Y | |
|---|---|----------|-------------------|
| 1 | 1 | 1 | immer gesetzt |
| 0 | 1 | 1 | |
| 1 | 0 | 1 | keine Veränderung |
| 0 | 0 | 0 | |

Wenn Bit Y Eins ist ergibt die ODER- Verknüpfung mit Bit X immer eine Eins. Wenn das Bit Y dagegen Null ist, ändert die Verknüpfung Bit X nicht.

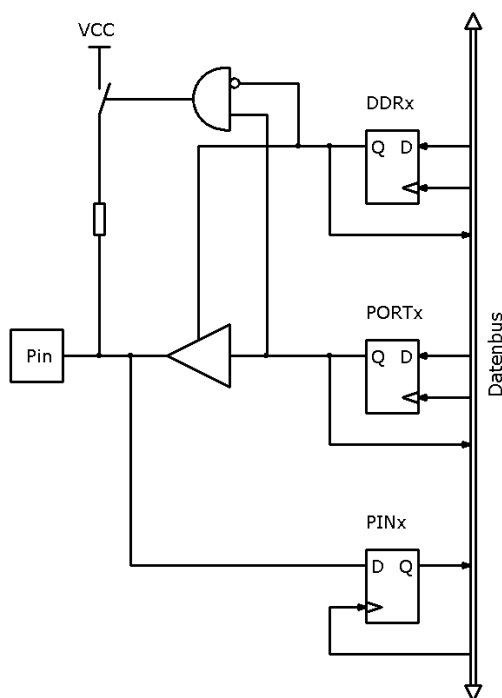
Port B ist ein ganzes Byte (8 Bit) in dem nur ein Bit verändert werden soll. Um dieses zu erreichen ist eine Maske notwendig in der für nicht zu verändernde Bits eine Null und für das zu setzende Bit eine Eins steht.

| | | | | | | | | | |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|---|
| PORTB | PB7 | PB6 | PB5 | PB4 | PB3 | PB2 | PB1 | PB0 | |
| | X | X | X | X | X | X | X | V | X = unverändertes Bit, V = Bit bearbeiten |
| Maske | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | |
| Oder verknüpft | X | X | X | X | X | X | X | 1 | 1 = Bit gesetzt |

Wie kann ein Bit gelöscht werden?

Konfiguration eines Portpins

Jeder Portpin wird von drei Registern kontrolliert.



Da ist es zum einen das Daten- Richtungs-Register (Data Direction Register, DDRx), in dem angegeben wird ob der Pin Eingang ('0') oder Ausgang ('1') ist.

Das Port- Register (PORTx), in dem der logische Wert des Pins steht, wenn der Port auf Ausgang geschaltet ist. Oder, wenn der Port ein Eingang ist, ob ein interner Pullup- Widerstand eingeschaltet wird.

Diese beiden Register können geschrieben und gelesen werden, geben aber nicht an welcher logische Wert tatsächlich am Portpin anliegt. Dazu gibt es das dritte Register, das Port- Input-Register (PINx). Dieses Register liest den Portpin ein, und ein lesen dieses Registers gibt den Zustand des Portpins zurück. Wird in das PINx-Register geschrieben landet der Wert im Port-Register.