

Praktikum 1

Im Rahmen der Veranstaltung „Mikrorechentchnik“ wird es 10 Praktika von 3x45min Dauer geben. Durch diese Praktika sollen die Teilnehmer in die Lage versetzt werden, mit Mikrocontrollern umzugehen und ein größeres Projekt erfolgreich zu bearbeiten. Diese Ziele können nur erreicht werden, wenn jedes Praktikum gewissenhaft vorbereitet wird.

Gegen eine Teamarbeit bei der Vorbereitung ist nichts einzuwenden. Die eigentliche Lösung der Aufgaben sollte jedoch von jedem verstanden werden. Ohne ein solches Verständnis wird das Lösen (und Verstehen der Lösung!) der späteren Aufgaben schwierig!

In diesem ersten Praktikum¹ soll gezeigt werden, wie man den ersten Schritt tut und einen Controller zum Leben erweckt. Als Controller verwenden wir einen AVR AT90S8515 (den hatten wir in der Krabbelkiste).

Vorbereitung

Schreiben Sie eine Assemblerfunktion für den AVR-Core, die mit Hilfe eines Zählers das Bit 0 des Registers R10 im Sekundentakt setzt und löscht. Gehen Sie davon aus, dass der Prozessor mit 1MHz getaktet wird!

- Implementieren Sie einen Zähler!
- Berechnen Sie die notwendige Bitzahl dieses Zählers, um mit 1MHz Systemtakt die Wartezeit von 1s realisieren zu können.
- Ihnen stehen nur 8-Bit Register zur Verfügung. Wie viele benötigen Sie? Überlegen Sie sich eine Möglichkeit, die notwendige Anzahl Bits mit Hilfe von 8-Bit-Registern zu implementieren.
- Implementieren Sie den Zähler!
- Implementieren Sie das Setzen und Löschen des Bit0 des Registers R10!
- Kommentieren Sie **jede** Zeile!

Während des Praktikums stehen Ihnen PCs zur Verfügung, die über einen USB-Anschluß für Memory-Sticks und über ein Diskettenlaufwerk verfügen. Bringen Sie den vorbereiteten Code auf einem geeigneten Speichermedium mit!

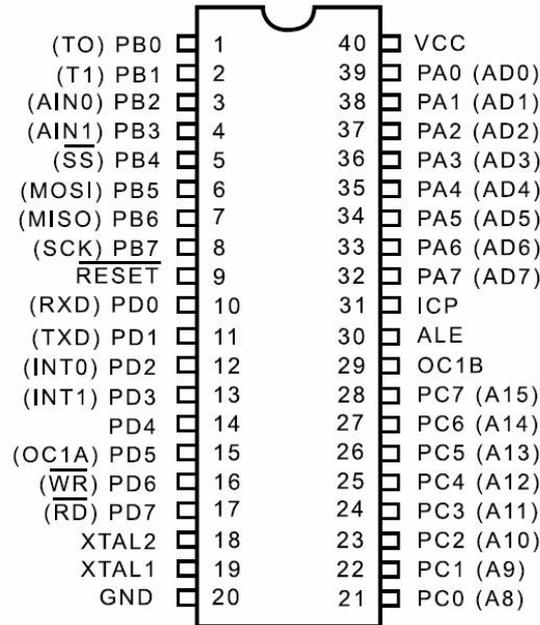
Benutzen Sie folgendes Template:

```
.INCLUDE "8515def.inc"; Deklarations for AT90S8515
.DEF    temp = r16    ; working register
;... further definition follow here...
.CSEG                                ; Program-Flash
rjmp   start           ; Reset-address
.ORG   0x2A            ; don't program interrupt vector table
start: ; put your code in here...
```

¹ Große Teile dieser Anleitung stammen aus dem Online-Tutorial „AVR-Einstieg leicht gemacht“, das unter http://www.roboternetz.de/wissen/index.php/AVR-Einstieg_leicht_gemacht verfügbar ist.

Durchführung

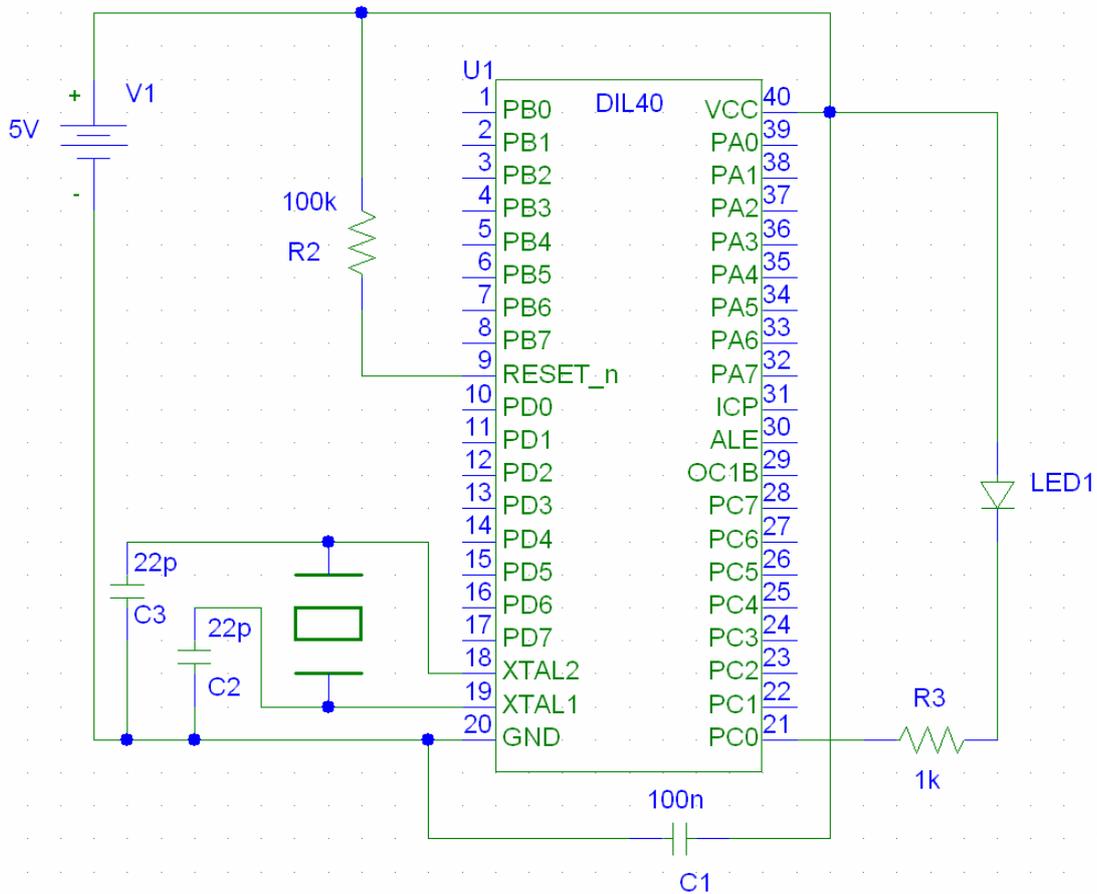
Das Pinout des im ersten Praktikum verwendeten Prozessors ist unten dargestellt.



Damit ein Controller erst mal zum Laufen kommt, benötigt man zumindest einen 10-k-Widerstand, einen 100-nF-Kondensator und eine 5V-Spannungsquelle. Das Ganze muss entsprechend dem unteren Schaltplan verschaltet werden.

Damit wir sehen, ob sich bei unserer Schaltung auch was tut, schließen wir eine LED über einen 1K-Vorwiderstand an einen Port an. Da wir die Kathode (kurzes Bein) an den Controllerpin und die Anode (längeres Bein) an Plus legen, leuchtet die LED immer dann, wenn dieser Pin auf Low geschaltet wird.

Der Quarz sollte genauso wie die beiden 22pF-Kondensatoren, die zum Anschwingen des Quarzes dienen, möglichst nahe am Controller platziert werden. Wichtig ist, dass die Leitungen vom Quarz zum Controller möglichst kurz sind. Auch die einzelnen Bauteilbeine, z.B. der Kondensatoren, sollten so kurz wie möglich sein.



Um den Controller in Betrieb zu nehmen, benötigt man bei dieser Schaltung ein Netzteil, das eine stabilisierte Spannung von 4 bis 5V liefert. Erwähnenswert ist noch, dass der Kondensator als Abblockkondensator zur Unterdrückung von Störungen dient. Er muss so nah wie möglich an die Spannungszuführung am Controller selbst gesetzt werden. Ohne Kondensator entstehen erhebliche Störungen in der Versorgungsspannung. Der Widerstand in der Grundschiung dient dazu, die Reset-Leitung konstant auf definiertem High-Pegel zu halten. Verbindet man diesen RESET-Pin später kurz mit GND (Masse), dann wird das Programm im Controller neu gestartet.



Bauen Sie diese Schaltung auf dem Steckbrett nach! Achten Sie darauf, dass keine Kurzschlüsse entstehen!

Woran merkt man, dass der Controller funktioniert?

Gute Frage! Man merkt es gar nicht. Genau genommen funktioniert ja auch trotz korrekter Schaltung noch immer nichts, denn es muss zuerst ein Programm in den Controller geladen werden, damit er überhaupt weiß, was zu tun ist. Ohne Programm ist der Controller quasi tot.

Deshalb muss jetzt als nächstes das Programm erstellt werden.



- Benutzen Sie AVR-Studio, um Ihr vorbereitetes Programm zu testen!
- Ändern Sie jetzt das Programm dahingehend, dass statt des Bit 0 des Register R10 der Pin 0 des Ports C ein- und ausgeschaltet wird. Dazu muss dieser Pin als Ausgang konfiguriert sein!
- Assemblieren Sie Ihr Programm!

Um das ausführbare Mikrocontrollerprogramm (Firmware, HEX-File) in dem nichtflüchtigen Speicher des Mikrocontrollers speichern zu können, werden so genannte **Programmer** verwendet. Im ersten Praktikum verwenden wir kein so genanntes In-System-Programming (ISP), das von den AVR-Bausteinen unterstützt wird, sondern den klassischen Ansatz, bei dem der Mikrocontroller physikalisch in ein Programmiergerät gesteckt wird, um dann mit dem Programm beschrieben zu werden.



Flashen Sie den Mikrocontroller und nehmen Sie die Schaltung in Betrieb!
Blinkt die LED, haben Sie das Ziel des 1. Praktikums erreicht!

Zusatzaufgabe:

Benutzen Sie alle Bits des Port C, um 8 LEDs anzuschließen! Programmieren Sie ein Lauflicht!