



Versuch 4: Digital-Analog-Wandler

Zielstellung:

- Kennenlernen von 8-Bit-Mikrocontrollern,
- Arbeit mit einem Mikrocontrollerentwicklungssystem,
- Eingabe und Editieren von Quellprogrammen in der Programmiersprache C,
- Compilieren und Linken von Anwenderprogrammen,
- Download von Anwenderprogrammen in die Zielhardware,
- Starten und praktischer Test von Anwenderprogrammen,
- Realisierung der Ansteuerung von DA-Wandlern mithilfe von Mikrocontrollern,
- Kennenlernen der Interruptbearbeitung von Mikrocontrollern (Timerinterrupt)

Versuchsvorbereitung

Für die Versuchsdurchführung ist es hilfreich, eine knapp gehaltene Befehlsreferenz für die Programmiersprache C, das Datenblatt des eingesetzten Controllers Dallas 80C320 und eine ASCII Tabelle zur Verfügung zu haben.

Aus der Vorlesung und Übung sollte der Umgang mit der Entwicklungsumgebung Keil µVision bekannt sein.

Bereiten Sie sich auf die Beantwortung von Fragen aus der separat verfügbaren Datei „Beispielfragen Praktikum Mikrorechentechnik.pdf“ vor.

Worin besteht bei Mikrocontrollern der Unterschied zwischen Timern und Countern?

Ermitteln Sie den Wert der nachzuladenden Zeitkonstante eines aufwärts zählenden 16Bit-Zählers, der als Timer die quartzgesteuerte Taktfrequenz von 24 MHz des DS80C320 verarbeitet und der bei einem Übergang des Zählerstandes von 0xFFFF nach 0x0000 einen Interrupt auslöst wobei eine Interruptserviceroutine mit einer Zykluszeit von 0.5 ms aufgerufen wird! Berücksichtigen Sie zusätzlich einen internen Vorteiler von 4!

Welche Möglichkeiten des Ladens der Zeitkonstanten lassen die Timer des DS80C320 für die Abarbeitung zeitlich äquidistanter Interruptzyklen zu?

Welche Wertebereiche der digitalen Ausgabevariablen ergeben sich bei 8-Bit, 10-Bit, 12-Bit und 16 Bit D-A-Wandlern?

Die aktuelle Anleitung ist unter: [Fakultät EI, Labore E, Mikrorechentechnik](#) zu finden!

Versuchsaufbau

Die 8 Ausgangsbits des Mikrocontrollersystems P1_0 ... P1_7 sind entsprechend der folgenden Abbildung mit den digitalen Eingängen eines D-A-Wandlers zu verbinden. Der D-A-Wandler wird mit seiner eigenen Referenzspannung REF_OUT von 2.5V gespeist (Verbinden von REF_OUT mit REF_IN). Die Referenzausgangsspannung des D-A-Wandlers hat einen Wert von 2.5V, ein digitaler Eingabewert von 0xFF entspricht 2,5V, ein Wert von 0x00 entspricht 0V. Da nur ein Teilnehmer an die Ausgangsbits des Mikrocontrollers geschaltet wird, kann der CE-Eingang ständig aktiv geschaltet sein und auf Massepotential liegen. Die Ausgangsspannung des D-A-Wandlers am Pin VOUT ist per Oszilloskop zu kontrollieren.

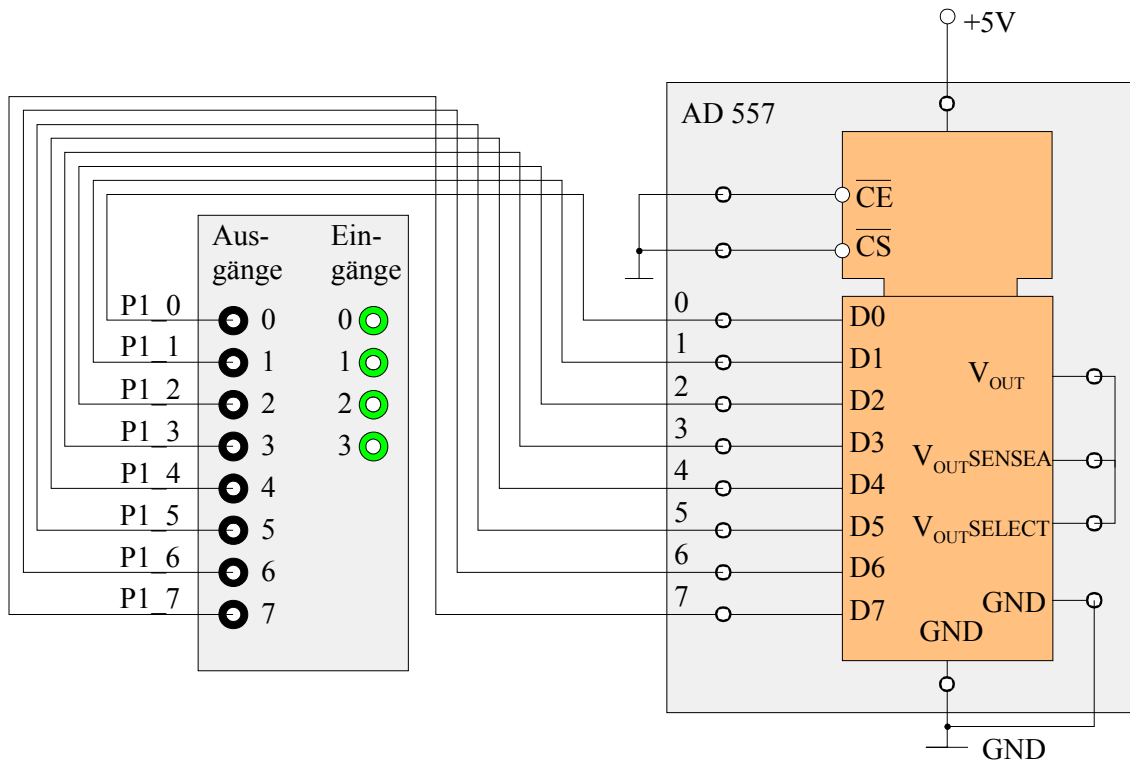


Abb.1: Verdrahtung der Ausgabeleitungen mit der Sieben-Segment-Anzeige

Aufgabenstellung:

Es ist ein Programm für einen Generator zu entwickeln, womit Frequenz und Amplitude vom PC aus eingestellt werden können. Die Ausgangsspannung soll 0V für den low-Bereich und einer einstellbaren Spannung bis max 2,5V für die Highzeit entsprechen. Die Frequenz soll von 1..10kHz mit 1kHz Schrittweite einstellbar sein.

Details zur Umsetzung (jeder Punkt erfordert zur Umsetzung einige wenige Sourcecodezeilen):

1. Innerhalb einer Endlosschleife
2. soll abwechselnd mit 2 aufeinander folgenden Abfragen über RS232 die Frequenz und die Ausgangsamplitude abgefragt werden.
3. Die eingegebenen bzw. daraus abgeleiteten Werte sind in globalen Variablen abzulegen.
4. Die Abfragen sollen durch aussagekräftige Texte (z.B. Bitte Frequenz in kHz eingeben) eingeleitet werden und
5. können mit getchar() oder scanf() durchgeführt werden.
6. Wird als Frequenz die Zahl 1 eingegeben, so ist eine Frequenz von 1 kHz und einem Tastverhältnis von 1, bei Eingabe der Zahl 2 von 2 kHz und einem Tastverhältnis von 1 usw. zu erzeugen.
7. Die Spannung soll als Fließkommazahl eingelesen werden.
8. Frequenzen zwischen 1 und 10 kHz im 1kHz Raster sind gültig, sowie
9. Spannungen zwischen 0 und 2,5Volt.
10. Bei Wertüberschreitung wird mit den bisherigen Werten weiter gearbeitet
11. und eine entsprechende Fehlermeldung ausgegeben.
12. Die zu berechnende Zeitkonstante wird in die vordefinierten Variablen TL0 und TH0 geladen, wobei TL0 die 8 niederwertigen und TH0 die 8 höherwertigen Bits des 16Bit-Zeitkonstantenwertes beinhalten. Diese 16Bit-Zeitkonstante dient als Startwert eines Timers.
13. Die Spannung ist auf den AD-Wandlerbereich zu skalieren (0x00 → 0V, 0xff → 2,5V).
14. Im Programm ist interruptgesteuert entsprechend des Wertes der Zeitkonstante im 0.5ms, 0,25ms, 0,125ms ... der Nachladewert TH0 und TL0 für den Timer 0 einzustellen.
15. Innerhalb der Interruptserviceroutine sind entsprechende Ausgaben auf die digitalen Ausgänge P1_0 ... P1_7 durchzuführen und der D-A-Wandler mit Digitalwerten zu speisen.
16. Mithilfe eines Oszillographen sind Amplitude und die Frequenz der Rechteckschwingung zu überprüfen (vgl. Abb.1).

- Die Software zur Bearbeitung ist Keil µVision. Die Software beinhaltet einen Editor, Compiler, Linker, Simulator und Debugger.

- Auf dem Laborrechner evtl. vorhandene Vorlagen für die Projekte sind nicht direkt zu nutzen. Erstellen Sie sich in jedem Fall eine Kopie der Daten! Eine Sicherung auf das eigene Homelaufwerk (Novell-Login), USB-Stick o.ä. ist in jedem Fall durchzuführen.

Listing des Roh-Programms versuch4.c

```
#include <reg320.h>           /* include 8051 header file      */
#include <stdio.h>             /* Standard I/O functions */
#include "init.h"

int Nachladewert_TH0, Nachladewert_TL0, Nachladewert_Amplitude; // globale Variablen

static void my_timer0_isr(void) interrupt 1 /*Timer 0 Overflow*/ using 1 {
/*****
** Hier eigenen Quelltext
** einfügen
*****/
}

void main (void) {
/*****
** Hier eigene Variablen
** anlegen
*****/

init ();
init_interrupt();

printf(" Praktikumsversuch 4: Digital-Analog-Wandler:\n\n");

while (1) {
/*****
** Hier eigene Quelltext
** einfügen
*****/
}
}
```