Programmentwurf

Systemnahe Programmierung I

Task-Verwaltung

Michael Strobel

Ernesto Elsäßer

TINF11B

4. Halbjahr

2013

Inhalt

[1 Einleitung 3](#_Toc354418281)

[2 Behandlung von doppelten Prozess-Aufrufen 3](#_Toc354418282)

[3 Kontroll-Ausgabe 3](#_Toc354418283)

[4 Beschreibung wichtiger Programmteile 4](#_Toc354418284)

[5 Kommentiertes Listing 5](#_Toc354418285)

# Einleitung

Die Projektaufgabe hat zum Ziel, ein Multiprozess-System auf einem 8051-Mikroprozessor zu realisieren. Die einzelnen Prozesse sollen unabhängig voneinander quasi-parallel ablaufen, wobei sie sich nicht kooperativ verhalten. Ein Scheduler kümmert sich um eine gerechte Verteilung der Prozessorzeit unter den aktiven Prozessen. Als Zuteilungs-Verfahren kommt beispielsweise die Zeitscheiben-Steuerung infrage, bei welcher der aktive Prozess nach einer bestimmten Zeitspanne gewechselt wird.

Folgende Prozesse sind enthalten:

Der Ausgabe-Prozess A gibt ca. einmal pro Sekunde ein 'a' auf der seriellen Schnittstelle aus.

Der Ausgabe-Prozess B gibt einmalig '54321' auf der seriellen Schnittstelle aus. Danach beendet er sich.

Der Konsolenprozess liest Zeichen auf der seriellen Schnittstelle ein. Bei 'a' und 'b' wird der Ausgabe-Prozess A gestartet bzw. beendet. Ein 'c' startet den Ausgabe-Prozess B. Werden andere Zeichen eingegeben, führt der Prozess keine Aktion aus.

# Behandlung von doppelten Prozess-Aufrufen

Wird ein Prozess etwa durch Eingabe von 'aa' mehrfach aufgerufen, wird er zurückgesetzt und neu gestartet.

Das StartProcess-Unterprogramm läuft unabhängig davon ab, ob der zu startende Prozess schon gestartet wurde. Es setzt den Eintrag in der Prozesstabelle auf aktiv, schreibt die Startadresse des Prozesses in dessen Stack-Bereich und setzt dessen gesicherte Register zurück. Dies dient in erster Linie dazu, einen bereits beendeten Prozess beim erneuten Starten wieder komplett zurückzusetzen. Es hat allerdings auch denselben Effekt auf bereits laufende Programme.

# Kontroll-Ausgabe

54321aaaa54321aaa54321aaaaaaa54321aa

# Beschreibung wichtiger Programmteile

# Kommentiertes Listing

Listing 1: main.a51

; C517A-Symbole verfügbar machen

$NOMOD51

#include <Reg517a.inc>

NAME main

; Symbole aus den Modulen importieren

EXTRN CODE (scheduler, startProcess, serialSend, processConsole, processAusgabeA, processAusgabeB)

; Variablen anlegen

dataSegment SEGMENT DATA

RSEG dataSegment

STACK: DS 4

; Interrupt-Routinen definieren

CSEG

ORG 0x0B ; Timer 0

JMP scheduler

; Systemstart-Anweisungen

ORG 0

JMP start

codeSegment SEGMENT CODE

RSEG codeSegment

start:

;

; Prozessor-Konfiguration

;

; Interrupt-Flags

SETB EAL ; Interrupts global aktivieren

SETB ET0 ; Timer 0-Interrupt für den Scheduler

MOV TMOD,#00010000b ; Timer 1: 16 Bit Timer, Timer 2: 8 Bit Timer

; Timer 1 für Prozess A aktivieren

SETB TR1

; Serial Mode 1: 8bit-UART bei Baudrate 9600

CLR SM0

SETB SM1

SETB REN0 ; Empfang ermöglichen

SETB BD ; Baudraten-Generator aktivieren

MOV S0RELL,#0xD9 ; Baudrate einstellen

MOV S0RELH,#0x03 ; 9600 = 03D9H

; Stack Pointer auf reservierten Bereich setzen

MOV SP,#STACK