MIKROPROZESSORPRAKTIKUM

WS2020

Termin 5

Programmierung für eingebettete Systeme: Pointer, Peripherie, USART, SWI

Name, Vorname	Matrikelnummer	Anmerkungen
Datum	Raster (z.B. Mi3x)	Testat/Datum

Legende: V:Vorbereitung, D: Durchführung, P: Protokoll/Dokumentation, T: Testat

Arbeitsverzeichnins:

Kopieren Sie sich aus dem Ordner „/mnt/Originale“ auf dem Laborarbeitsplatzrechner oder aus dem Ordner „sftp://stxxyyyy@userv-shell.fbi.h-da.de/home/groups/LabDisk/MI/“ den Ordner mpsWS2020. Dort finden Sie zu jedem Termin vorgegebene Dateien.

Lernziele:

Die Kenntnis der Basistechnologie zur Implementierung einer Schnittstelle zwischen Anwendungsprogrammen und Betriebssystemen. Die Bedeutung und Anwendung von Supervisor- und User-Mode. Möglichkeiten der Ein- und Ausgabe von Zeichen über eine serielle Schnittstelle. Vorbereiten von Testszenarien bei Zugriff auf ein Zielsystem im Labor.

Aufgabenstellung:

Der SWI Befehl führt zu einer Ausnahmebehandlung im Prozessor die mit einem Wechsel in den Supervisor Mode verbunden ist. Dem SWI Befehl kann beim Aufruf eine bis zu 24 Bit großer Wert übergeben werden, die in einem SWI Handler dazu benutzt werden kann eine gewünschte Funktion/Aktion auszuwählen.

In der Aufgabe soll ein SWI-Handler gezeigt/genutzt werden, um eine Trennung des Low Level IOs vom Anwendungsprogramm zu erreichen. Dazu sind Funktionen/Unterprogramme, die im Supervisor-Mode ausgeführt werden, in Assembler implementiert/zu implementieren. Mit dem Debugger ist die Funktionsweise des SWI-Handlers zu untersuchen.

Aufgabe 1:

Nehmen Sie die zur Verfügung gestellten Dateien in ein neues Projekt auf, diskutieren, verstehen, testen und dokumentieren Sie diese.

Ermitteln Sie welche Parameter in der Terminalemulation (z.B. minicom) einzustellen sind, damit eine Kommunikation über die serielle Schnittstelle funktionieren wird.

_____

Beschreiben Sie den Unterschied der Unterprogramme/Prozeduren init_ser() und inits().

_____

Was passiert oder könnte passieren, wenn Sie direkt nach den Ausgaben von CR und LF durch die Aufrufe *putc(0xd)* und *putc(0xa)* noch weitere Zeichen *putc('a'); putc('b')* (in Echtzeit) auf die gegebene Weise ausgeben und warum?

_____

Die erzeugten Ausgaben von CR (Wagenrücklauf) und LF (Zeilenvorschub) sollten auf einem Terminal an der seriellen Schnittstelle zu sehen sein. Verwenden Sie in einer separaten Shell das Programm ***„minicom“*** als Terminalemulation.

Aufgabe 2:

Erklären Sie den Code des SWI-Handlers (siehe swi.c/swi.s). Debuggen Sie Ihr Programm und lokalisieren Sie die Umschaltung vom User Mode in den Supervisor Mode und zurück.

Woran erkennen Sie, in welchem Mode sich der Prozessor befindet?

_____

An welcher Stelle wird der Supervisor Mode verlassen?

_____

Aufgabe 3:

Die Initialisierungs-Routine *void inits(void)* und die IO-Funktionen char putc(char) und char getc() ermöglichen es, die vorhandenen in C geschriebenen Programme *int init_ser(void)*, *char putch(char)* und *char getch(void)* im Supervisor Modus auszuführen. Die benötigten Rückgabewerte der in C-geschriebenen Funktionen werden entsprechend durch gereicht. Also auch wenn die serielle Schnittstelle nicht bereit wäre, liegt die Entscheidung beim Nutzer, ob auf die gerade nicht sende- oder empfangsbereite Schnittstelle gewartet werden soll.

Wie könnten/müssten die Aufrufe der Funktionen *char putc(char)*, *char getc(void)*, *char putch(char)* und *char getch(void)* verpackt werden, damit gewartet wird, bis ein Zeichen gesendet bzw. empfangen wurde. Testen Sie Ihre Lösung und weisen Sie die Funktionalität nach.

Aufgabe 4:

Können Sie den fehlenden in Assembler zu implementierenden Programmteil für void puts(char*) (siehe ***ser_io.S***) so ergänzen, dass auch ein String im Supervisor Mode auf die serielle Schnittstelle ausgegeben wird?

void puts(char *) Ausgabe eines nullterminierten Strings.

Sie können auch die in C geschriebene Routine void putstring(char*) zur Ausgabe von Strings nutzen.

Aufgabe 5:

Entwickeln und **testen** Sie für Ihr Projekt eine möglichst effektive und speichersparende Routine inttostring, mit der Sie eine vorzeichenbehaftete Integerzahl (int Zahl) in einen String zur Ausgabe auf die Konsole wandeln.

Mit welchen Werten werden Sie Ihre Routine testen, um die volle/richtige Funktionalität nachzuweisen?

Aufgabe 6:

Erstellen Sie zu diesem Termin ein Protokoll mit den Lösungen zu den Aufgaben und Ihren Erkenntnissen. Das Protokoll sollen Sie zu den nächsten Terminen vorlegen können. Denken Sie daran, dass zum letzten (sechsten Termin) auch eine Dokumentation zu Ihrem Projekt (Funktions- und Programmbeschreibungen, Installationsanleitung, Inbetriebnahme, Benutzerhandbuch) vorgelegt werden soll.