

## Praktikumsplatz Labor MS OS-9 Beckhoff PC

### 1. Beschreibung:

Der Praktikumsplatz (PP) besteht aus einem Windows **PC**, einem **OS9-PC**, einem **Terminal** und dem LMS-HW-Board.

Das OS-9-System bootet nach dem Einschalten von der Festplatte. Im Praktikum wird OS9000 geladen, im Weiteren wird dieses System OS-9 genannt.

Das OS-9 System am Praktikumsplatz hat die IP-Adresse 192.168.5.100 und ist via Ethernet ausschließlich und direkt mit dem Windows PC am Praktikumsplatz verbunden. Der Windows PC besitzt zwei Netzwerkkarten:

1. eine ist mit dem OS-9 des Praktikumsplatzes verbunden und hat die IP-Adresse 192.168.5.1 und
2. die andere ist mit dem Hochschulnetz verbunden, deren IP-Adresse wird zur Boot-Zeit ermittelt.

Die OS-9-Systeme sind ‚unsichtbar‘ aus dem Hochschulnetz, die Windows PCs routen keine Pakete vom OS-9-System ins Hochschulnetz und umgekehrt.

### OS-9-Bedienung und Allgemeines:

- Die lokale RAM-Disk hat die Laufwerksbezeichnung **/dd (default-disk)**. **Nur auf diese RAM-Disk können Sie schreiben!!**. **Vorsicht: die RAMdisk verliert durch das Ausschalten oder Reboot des OS-9-Systems ihren Inhalt!!**
- **Terminalbedienung:** Beim Booten des Systems erscheinen am Terminal die einzelnen Bootmeldungen. Das Booten des OS-9 Systems ist nach c.a. 5-10 Sekunden abgeschlossen und es erscheint der login Eingabeprompt:  
**→ OS9-Username: super Passwort: user**
- Damit bei Kommandos der Bildlauf nicht seitenweise erfolgt, sondern kontinuierlich, geben sie am Prompt das Kommando: ***tmode nopause*** ein.
- **Weitere interaktive Terminals via Telnet-Verbindung vom Windows PC aus:** Um interaktiv vom PC aus mit dem OS-9 System zu arbeiten, können Sie eine **telnet** Verbindung zum OS-9-System öffnen. Kommando am PC (→Ausführen wenn kein Telnet-Icon vorhanden ist) : **telnet 192.168.5.100**
- **Die dann laufende Shell besitzt eine Startpriorität von 128.** Prozesse mit dieser Priorität werden im Timesharing bearbeitet.

### 2. Manuals zur C-Bibliothek/OS9 nur online

Siehe: [http://lms.ee.hm.edu/~seck/AlleDateien/OS9HILFE/C-Lib-Ref/ultrac\\_lib\\_ref.pdf](http://lms.ee.hm.edu/~seck/AlleDateien/OS9HILFE/C-Lib-Ref/ultrac_lib_ref.pdf)

### 3. Windows PC-Systeme (Anmeldung und Verzeichnisse)

- **Anmeldung PDT Praktikum:**  
PC-Username: PDT01 – PDT16                      Passwort: pdt
- **Anmeldung Echtzeitsysteme-Praktikum:**  
PC-Username: echt01 – echt16                      Passwort: echt
- **PC-Arbeitsverzeichnisse für beide Praktika:** L:\ Projekte

### 4. Windows PC-Systeme als OS-9-Cross-Entwicklungsplätze

- Sie erstellen am Windows PC ihre Programme mittels der integrierten Entwicklungsumgebung „**Microware Hawk IDE**“
- **(hawk) Kurzanleitung:**
  - hawk integriert Editor/Compiler/RemoteDebugger für das OS-9-System.
  - **hawk strukturiert den Entwicklungsvorgang in folgende Ebenen:**
    - 1. oberste Ebene: **Projectspace** entspricht einer Sammelmappe für **ein oder mehrere Projekte**  
→ In unserem Fall entweder: **Projectspace: L:\Projekte\PDTPraktikum oder L:\Projekte\OS9Praktikum**
    - 2. Ebene: **Projekt**, ein Projekt ist ein in sich abgeschlossenes Ganzes, das aus einer oder mehreren **Components** besteht.  
→ In unserem Fall: **Projekt: Versuch1, Versuch2 ...Versuch3...**
    - 3. Ebene: **Component**, der Name einer Component ist bei der Erstellung von Programmen gleichzeitig der Name der ausführbaren **OS9-Programmmodul- Datei**, eine Component enthält alle Source- und Header-**Dateien (Units)** zu einem lauffähigen Programm
    - 4. Ebene: **Units** : einzelne **Dateien entweder .c oder .h**
  - hawk merkt sich i.d.R. den **letzten geöffneten Projectspace** und **öffnet diesen bei Wiederstart**
- **Erster Praktikumstag:**
  - **Starten von hawk und Anlegen eines Projectspaces** → Menu: **projects/projectspace/new**
    - **Current Directory L:\Projekte**, Filename entweder **PDTPraktikum oder OS9Praktikum**, Die Aufforderung **Projekte hinzuzufügen verneinen**, **Warnung ignorieren**
    - **Damit wurde für das gesamte Praktikum ein Projectspace angelegt**
- **Bei jedem neuen Praktikums-Versuch: Anlegen eines neuen Projektes** → **Versuch1 ....Versuch3**
  - **Menu: projects/projectspace/Add New Project**
  - **Projectname: Versuch<x>**, **Folder: L:\Projekte**, **Chip M505**
  - **Component: Name des zu erstellenden ausführbaren OS9-Programmmoduls z.B. hallo oder systask**, nächstes Menü **Units** mit **finish** beenden

- **Anlegen einer Component innerhalb eines bestehenden, neuen Projects:**
  - **Project** auswählen im Projectfenster und <rechte>-Maustaste drücken und 'new-component' wählen
  - **Component:** Name des zu erstellenden ausführbaren OS9-Programmmoduls z.B. hallo oder systask, Chip M505 wählen, nächstes Menü Units mit finish beenden
- **In jedem (Teil-)Versuch für jede neue .c oder .h Datei (Unit):**
  - In der Toolbar 'create new empty document' auswählen, sofort Menu: File/Save as/ auswählen und <filmanen>.c oder .h eingeben, dabei achten ob die Datei auch in L:\Projekte oder einem Unterverzeichnis davon angelegt wird.
  - **Wichtig:** Danach im Editorfenster <rechte>-Maustaste drücken und 'add File to Project' auswählen → 'Into Component': die Component auswählen in die die Datei aufgenommen werden soll und mit <ok> den Vorgang abschliessen. Jetzt gehört die Datei zur Component dazu.
  - **hawk-Project-Ansicht:**
    - Fenster links oben: Components-Baum → Ansicht Wurzel entspricht Projektebene Versuch1,Versuch2...  
Jeder **Praktikumsteilversuch** ist je als eine neue Componente anzulegen
    - Fenster links unten: zeigt alle Components wenn im Fenster links oben die ‚Wurzel‘ angewählt ist, ansonsten die zu einer Component gehörigen Dateien
    - Fenster rechts oben: Editor Fenster
    - Fenster rechts unten: Fehler Fenster für Compiler
- **Source-Dateien (Programme) übersetzen/laden/debuggen:**
  - **Vor jedem Übersetzen <save-all> im Filemenü auswählen.** Übersetzen:→ Componente anwählen, rechte-Maustaste, <build>/<rebuild> auswählen → Fehlerausgabe im Build-Fenster beachten.
  - Soll das fertige fehlerfrei übersetzte Programm ausprobiert werden, gibt es zwei Möglichkeiten: Laden/und am Terminal manuell starten **oder** vom PC aus debuggen.
    - Debuggen → Componente anwählen, rechte-Maustaste, <debug> auswählen, beim ersten Mal die Targetfrage mit 192.168.5.100 beantworten (→Bedienung Debugger intuitiv).
    - Laden und manuell starten: → Componente anwählen, rechte-Maustaste, <load> auswählen, beim ersten Mal die Targetfrage mit 192.168.5.100 beantworten; dann am Terminal/Telnetverbindung den Programmnamen eintippen und das Programm startet ;-)).
    - **Achtung : Nach jedem fehlerfreien Neuübersetzen und vor dem Laden/Debuggen mittels procs sich vergewissern ob noch das Programm läuft. Wenn ja dann mit kill <taskid> beenden. Danach mittels unlink <Programmname> sicherstellen, dass das Programmmodul auch aus dem Moduldirectory entfernt wurde.**
- **PC-FTP** dient zum Übertragen ihrer am PC erstellten Dateien an Ihr OS-9-System. Nach dem Starten des Programms wird mittels **connect** Verbindung aufgebaut und eine bestehende Verbindung mittels **close** abgebaut. Ein **FTP** -connect funktioniert nur wenn das OS-9-System gebootet ist und nicht 'hängt' (→Taskprioritäten beachten!!). Unter den genannten positiven Umständen braucht die Verbindung für mehrere Filetransfers nur einmal hergestellt werden.
  - **connect**-Aufruf: Suchen sie in der Listbox die Verbindung zu Ihrem OS-9-System →falls nicht vorhanden  
Aufsicht fragen → starten der Verbindung und das PC- FTP-Programm loggt sich auf Ihrem OS-9-System ein und 'spricht' mit dem sog. FTP-Dämonen (Server) Ihres OS-9-Systems→ es erscheinen zwei Boxen →links das PC-Filesystem und rechts das OS-9-Filesystem an der root von /dd → Stellen Sie links ihr Arbeitsdirectory ein → es erscheinen ihre Dateien, markieren Sie nun die zu übertragenden Dateien → mittels => -Button übertragen Sie nun von PC nach OS-9-System die markierten Dateien. → Kontrollieren Sie mittels dir-Kommando am Terminal.

## 5. Beschreibung der Software zur Anbindung des LMS-HW-Boards an eine Anwendungstask

- Es existiert ein Treiber **lmsboard** und der dazugehörige Descriptor **dlmsboard**. Das für den Anwenderprogrammierer (Praktikumsteilnehmer) nutzbare OS9-Gerät **heisst: /dlmsboard**. Der dazugehörige **Treiber lmsboard** darf auf die Hardware direkt zugreifen und bedient exklusiv die oben unter 1. beschriebenen Hardwareschnittstellen. **Erst durch** das Kommando **iniz dlmsboard** wird **die Hardware und die Zugriffe darauf durch den Treiber lmsboard initialisiert** (Auch die Events existieren erst ab dem Zeitpunkt).
- **Das Gerät /dlmsboard** wird vom sog. SCF-Filemanager verwaltet und aufgerufen. Der SCF-Filemanager verwaltet auch das angeschlossene Gerät Terminal /term an dem Sie interaktiv arbeiten. Das **Gerät /dlmsboard** arbeitet demnach **wie ein seriell angeschlossenes Gerät** auf das man **byteweise schreiben** oder von dem man **byteweise lesen** kann.
- **Öffnen/Schliessen des Gerätezugangs auf /dlmsboard:** **Bevor** die Schalterstellung gelesen und die LED-Belegung geschrieben werden kann, muss der Programmierer, wie bei jedem Datei-/Gerätezugriff, das Gerät mittels **→ \_os\_open(...,S\_IREAD | S\_IWRITE,...)** öffnen und nach Gebrauch mittels **→ \_os\_close(...)** wieder schliessen. Das Gerät ist **jederzeit** bereit **ein Byte** zu lesen und **ein Byte** zu schreiben. (*#include modes.h*)
- **Lesen der Schalterstellung:** Mittels **\_os\_read(...)** mit Lesen **eines Bytes** wird die aktuelle Schalterstellung eingelesen.
- **Schreiben der LED-Belegung:** Mittels **\_os\_write(...)** mit Schreiben **eines Bytes** wird der Wert an die LEDs ausgegeben.
- **lmsboard:** lmsboard legt **nach Initialisieren mittels iniz dlmsboard zwei** Events an, die von jedem gelesen werden können: **LMS\_H1 für Taste T0** und **LMS\_H2 für Taste T1**. Die Events sind mit **sinc=1** und **winc=0** angelegt. Mit jedem Tastendruck T0/T1 wird durch die Interruptserviceroutine des Treibers lmsboard das entsprechende Event um eins hochgezählt.
- Soll das Gerät /dlmsboard wieder **endgültig geschlossen und aus dem System, entfernt werden**, so geht das nur wenn kein Programm dies Gerät gerade benutzt, mittels des Kommandos **deiniz /dlmsboard**.
- Mittels des Kommandos **irqs** und **devs** kann man sich einen Überblick über **initialisierte Geräte** und **Treiber** und deren Ressourcenbelegung verschaffen.