

CDT Anleitung zum Raspberry Pi

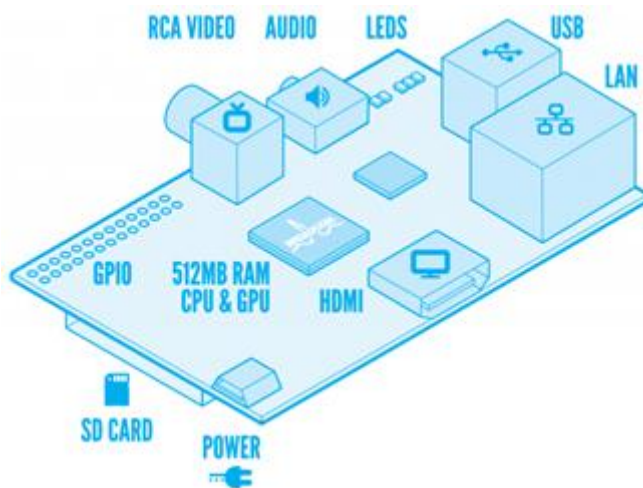
Im Fach CDT wird die C-Programmierung auf dem Einplatinen-Computer *Raspberry Pi* durchgeführt. Mithilfe dieser Anleitung kann das System in Betrieb genommen werden.

1. Einleitung

Für die CDT Praktika wurde ein Image vorbereitet, das die benötigten Anwendungen und Pakete bereits vorinstalliert hat. Um eine Verbindung mit dem Raspberry Pi über LAN aufbauen zu können, wurden die entsprechenden Tools ebenfalls bereits eingerichtet und vorkonfiguriert.

Es wird das Betriebssystem Raspbian verwendet, das auf Debian Wheezy (Debian 7.0) basiert. Diese Distribution wurde speziell für das Raspberry Pi optimiert.

Vorausgesetzt wird, dass ein Raspberry Pi Gerät des Modells B verwendet wird, welches im Gegensatz zum Modell A über einen LAN-Anschluss verfügt. Zudem muss das Board über den Micro-USB-Anschluss mit Strom versorgt werden. Alternativ ist die Stromversorgung auch über die USB-Anschlüsse (Typ A) möglich, wodurch jedoch nur noch einer der beiden Ports verwendet werden kann.



Grafik: raspberrypi.org

Die Informationen zum Standardbenutzerkonto für Raspbian:

Username: pi

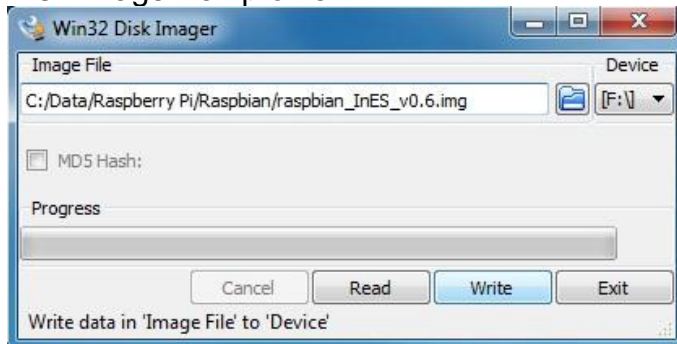
Password: raspberry

Link zum Raspbian Image: [\\pools01.zhaw.ch\\t\\T-STUD-CDT\\Raspbian_for_CDT](http://pools01.zhaw.ch/t/T-STUD-CDT/Raspbian_for_CDT)

2. CDT-Image (SD-Karte einrichten)

Als Erstes wird die SD-Karte mit dem Betriebssystem beschrieben

- Vorausgesetzt wird eine SD-Karte von mindestens 4 GB. Für das Schreiben des Images wird zudem ein SD-Schreibgerät benötigt.
- Als nächstes muss das Raspbian-Image heruntergeladen und dekomprimiert werden. Downloadlink: [\\pools01.zhaw.ch\t\T-STUD-CDT\Raspbian for CDT](http://pools01.zhaw.ch/t/T-STUD-CDT/Raspbian_for_CDT)
- Um das Image auf die SD-Karte zu schreiben, wird unter Windows der „Win32 Disk Imager“ empfohlen:



- Anleitungen, um die SD-Karte unter Linux und Apple zu beschreiben, finden Sie auf http://elinux.org/RPi_Easy_SD_Card_Setup.
- Das Image besteht aus mehreren Partitionen. Sobald der Schreibvorgang abgeschlossen ist, sollte man unter Windows die FAT32 Partition öffnen können. Darin befinden sich Dateien, welche für den Bootvorgang benötigt werden. (mount in raspbian: /boot)

3. Netzwerk-Einstellungen

Das Raspberry Pi soll ohne Bildschirm und Tastatur auskommen. Die Bedienung erfolgt über den eigenen Laptop, welcher mit dem Board über LAN verbunden ist. Um dies zu ermöglichen, müssen die Netzwerkeinstellungen angepasst werden:

- Der Client muss über LAN mit dem Raspberry Pi Board verbunden sein. Es sollte ein Ethernet-Kabel direkt zwischen den beiden Computern angeschlossen werden (Alternativ über einen Hub oder Switch).
- Das Raspberry Pi und der verwendete Computer sollen dieselbe Subnetzwerkmaske verwenden: 255.255.255.0.
- Die IP Adressen müssen sich im selben Subnetzwerk befinden (Subnetzwerkmaske: 255.255.255.0). Es kann entweder die IP-Adresse des Computers (a) oder die vom Raspberry Pi (b) angepasst werden.
 - a) Die Standardadresse in raspbian ist 192.168.137.2. Weist man dem eigenen Ethernet-Adapter eine IP-Adresse im selben Subnetzwerk zu (z.B. 192.168.137.3), so kann die Verbindung aufgebaut werden.
 - b) Dem Raspberry Pi wurde die IP-Adresse 192.168.137.2 zugeteilt. Diese kann jedoch unter /boot/ip_address.txt verändert werden. Da sich dieses File in der FAT32 Partition befindet, kann dieser Wert auch von einem Windows-Rechner aus angepasst werden.

Über einen Ping kann überprüft werden, ob die Verbindung besteht:

`ping 192.168.137.2` (respektive die neugewählte IP des Raspberry Pi Boards)

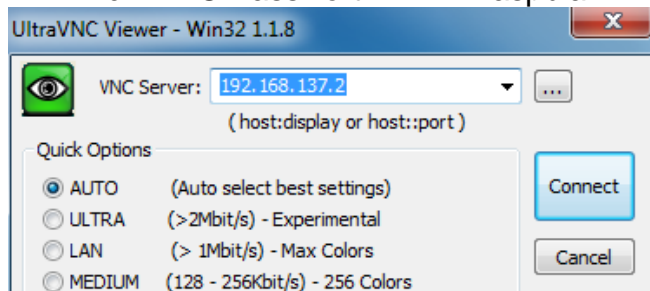
Ist dies der Fall, so kann nun eine VNC- oder SSH-Verbindung aufgebaut werden.

4. VNC Verbindung

Über VNC (Virtual Network Computing) kann auf dem Raspberry Pi gearbeitet werden.

Auf dem Image ist ein VNC Server installiert (tightvncserver), der während dem Bootvorgang gestartet wird.

- Mit einem VNC Client (z.B. UltraVNC Viewer für Windows) kann die Verbindung aufgebaut werden:
 - VNC Server: 192.168.137.2 / 192.168.137.2:0
 - VNC Passwort: raspbian



5. SSH Verbindung

Anstelle von VNC kann auch eine SSH (Secure Shell) Verbindung aufgebaut werden.

Windows

Für die Verbindung wird ein SSH-Client (hier PuTTY) benötigt. Bei Bedarf braucht es für die Darstellung von graphischen Anwendungen und der Benutzeroberfläche LXDE zusätzlich einen X Server. Hierfür kann Xming eingesetzt werden.

1. Installation von Xming und PuTTY auf Windows Rechner.
2. Starten von Xming. Mit der Maus über das Windows-Icon fahren, um die Window Nummer zu ermitteln. (Bei „Xming Server:0.0“ ist dies „:0.0“).
3. PuTTY starten.
4. Unter Session die IP Adresse eingeben (192.168.137.2).
5. Unter Connection → SSH → X11 das Häkchen bei „Enable X11 forwarding“ aktivieren.
6. Unter „X display location“ folgendes eingeben: „localhost:0.0“. „:0.0“ ist jedoch abhängig von Xming (Punkt 2).
7. Die Verbindung öffnen (Open).
8. Im Terminal erscheint „login as:“. Loggen Sie sich ein.
9. Mit dem Befehl „startlxde“ falls erwünscht die graphische Benutzeroberfläche starten.

Linux / Mac OS X

Um den OpenSSH SSH Client zu benutzen, folgenden Befehl eingeben:

```
$ ssh pi@192.168.137.2 -X
```

Die Option `-X` kann weggelassen werden, wenn kein grafischer Client benötigt wird (X11 forwarding).

6. Konfiguration raspi-config

Mit VNC oder SSH kann nun auf raspbian zugegriffen werden. Über die Shell kann der Befehl **sudo raspi-config** eingegeben werden, wodurch man ins Konfigurationsmenü von raspbian gelangt. Falls Sie sich bereits auf der grafischen Benutzeroberfläche befinden, können Sie die Befehle über LXTerminal eingeben.

expand_rootfs	Falls die benutzte SD-Karte grösser als 4GB ist, kann die Partition hier vergrössert werden.
configure_keyboard, change_locale, change_timezone	Tastatur-, Sprach- und Regionseinstellungen können hierrüber vorgenommen werden.
change_pass	Ändert das Passwort des Benutzers „pi“.
overclock	Das Übertakten des ARMs, Cores und SDRAMs kann hier vorgenommen werden. Dies, weitere Einstellungen wie auch das Heruntertakten können auch über /boot/config.txt vorgenommen werden.

7. Samba Server

Auf dem Raspberry Pi wurde ein Samba Server installiert, damit Dateien problemlos ausgetauscht werden können.

Aus Windows kann folgendermassen darauf zugegriffen werden:

file://192.168.137.2/pi/ \\192.168.137.2\pi

Unter Windows kann der Share als Netzlaufwerk verbunden werden („Verbindung bei Anmeldung wiederherstellen“ und „Verbindung mit anderen Anmeldeinformationen herstellen“ anklicken).

8. LXDE – Lightweight X11 Desktop Environment

Raspbian verwendet standardmässig LXDE als grafische Desktopumgebung.

Starten von LXDE:

- startlxde
- startx

Von hier aus kann z.B. der Internetbrowser Midori oder der Texteditor gedit verwendet werden. Aus LXDE heraus kann der Terminal Emulator LXTerminal eingesetzt werden, um Kommandos auszuführen.

9. Linux Basics

In diesem Kapitel werden grundlegende Unix-Befehle beschrieben. Studierende, die sich damit noch nicht auskennen, sollen die einzelnen Punkte auf dem Raspberry Pi austesten.

- Mit „**cd** / “ wechselt man ins Hauptverzeichnis (root). Über „cd ..“ gelangt man ins Überverzeichnis. Nach der Ausführung von „cd /home/pi/ „ befindet man sich im Home-Verzeichnis des Benutzers pi.
- Über den Befehl „**ls**“ können alle Verzeichnisse und Files angezeigt werden. Mit den Zusatzoptionen „ls -la“ erscheint eine detailliertere Ansicht.
- „**mkdir** dirName“ erstellt ein neues Verzeichnis an aktueller Stelle.
- Mit „cd dirName“ kann ins eben erstellte Verzeichnis gewechselt werden. Um herauszufinden, in welchem Verzeichnis man sich befindet, kann mit „**pwd**“ der Pfad des aktuellen Verzeichnisses ausgegeben werden.
- Sie wollen ein neues Textfile erstellen und bearbeiten. Dies können Sie beispielsweise über „**nano** testfile“ öffnen. Dieser Befehl öffnet oder erstellt die Datei testfile mit dem Texteditor nano. Schreiben Sie ein paar Zeilen, damit Sie mit dem Editor vertraut werden und speichern Sie das Dokument ab. Mit Ctrl+o können Sie das File speichern, mit Ctrl+e das Programm beenden, wobei ebenfalls gefragt wird, ob Sie das Dokument abspeichern möchten.
- Um zu überprüfen, ob der Inhalt korrekt gespeichert wurde, können Sie diesen mittels „**cat** testfile“ darstellen lassen.
- Mit „**whatis** cp“ erhalten Sie eine Kurzbeschreibung über den Befehl copy, über „**man** cp“ wird eine detaillierte Anleitung dargestellt. Dies ist für die meisten Befehle möglich und sehr hilfreich.
- Mittels „cp“ können Dateien kopiert werden. „mv“ verschiebt oder nennt Dateien um. „rm“ löscht eine Datei oder ein Verzeichnis. Um diese Befehle korrekt anzuwenden, soll das Manual durchgelesen werden.
- Mit „**sudo**“ kann ein Prozess unter einem anderen Benutzerkonto ausgeführt werden, standardmässig als root. Dadurch können Befehle ausgeführt werden, die eine spezielle Berechtigung verlangen. Beispiele sind „sudo reboot“, „sudo shutdown -h now“ oder „sudo apt-get install packetname“.
- Mit „**top**“ können alle Prozesse aufgelistet werden.
- Über „**grep**“ kann eine Datei nach einem Muster durchsucht werden. Es werden dann alle Zeilen ausgegeben, in welchen das Muster vorkommt. „grep muster“.
- Mittels einer Pipeline („|“) kann der Output eines Prozesses einem anderen Prozess als Input übergeben werden. „ls -la | grep drwx“ liest den Inhalt des aktuellen Verzeichnisses aus und übergibt diesen an grep, der nur die Zeilen ausgibt, welche *drwx* beinhalten (Verzeichnisse, bei denen User die Berechtigung *rw*x haben, sowie sonstige Übereinstimmungen mit *drwx*).
- „**ifconfig**“ listet die Netzwerk-Interfaces auf. Zum Beispiel kann die IP Adresse des Adapters eth0 geändert werden: „sudo ifconfig eth0 192.168.137.100“. Nach einem Reboot werden diese Änderungen jedoch wieder verworfen. Für eine permanente Änderung muss die Textdatei „/etc/network/interfaces“ angepasst werden.

Es gibt online zahlreiche Auflistungen von Befehlen. Eine gute Einführung in die UNIX-Welt bietet <http://www.netzmafia.de/skripten/unix/>

10. *Entwicklungsumgebung und erstes Programm*

Für die Entwicklung der C Praktika wird eine Reihe von Komponenten benötigt. Als Editoren für das Schreiben des Codes eignen sich zum Beispiel nano und vim. Wird LXDE verwendet, so kann auch gedit eingesetzt werden.

Das Fach CDT verwendet im C-Teil die üblichen Linux-Tools, um Programme zu erstellen (GNU Toolchain). Das sind folgende:

- gcc (Compiler und Linker)
- gdb (Debugger)
- make (Build-Script Interpreter)

Verwenden Sie eine andere Plattform, muss die Software gegebenenfalls noch installiert werden. Ob alles vorhanden ist und soweit funktioniert, können Sie in jedem Fall mit folgendem Beispiel überprüfen.

Beispiel „Hello World“

Erstellen Sie ein kleines HelloWorld-Programm mit einem beliebigen Editor und speichern Sie die Datei ab unter dem Namen "hello.c".

```
#include <stdio.h>

int main(int argc, char **argv) {
    printf("HelloWorld\n");
    return 0;
}
```

Das Programm lässt sich nun wie folgt mit gcc (GNU Compiler Collection) kompilieren und linken:

```
gcc -g -c -o hello.o hello.c
gcc -g -o hello hello.o
```

Mit einem Makefile lässt sich diese Abfolge von Befehlen kapseln, so dass das Programm mit dem einfachen Aufruf "make" erstellt und mit "make clean" wieder gelöscht werden kann. Erstellen Sie dafür eine neue Datei mit dem Name "makefile" und folgendem Inhalt:

```
default: hello

hello: hello.o
    gcc -g -o hello hello.c

hello.o: hello.c
    gcc -c -g -o hello.o hello.c

clean:
    rm -f hello.o
    rm -f hello
```

Starten Sie das Programm:

```
./hello
```

Starten Sie das Programm mit dem Debugger:

```
gdb hello
```

Da das Executable mit Debug-Symbols erstellt wurde (-g), können sie mit symbolischen Namen Breakpoints setzen z. B. In der Funktion „main“ mit „break main“. Eine Liste der vorhandenen Haltepunkte zeigt „info breakpoints“ an. Der Befehl „run“ startet das Programm. Sobald ein Breakpoint erreicht wird, haltet das Programm und der Debugger zeigt die aktuelle Zeile an. Mit „list“ kann man die 10 folgenden Zeilen aus dem Quellcode zusätzlich auflisten lassen. Drücken Sie „c“ und Enter, um das Programm weiter laufen zu lassen. Wenn es fertig ist, kann der Debugger mit „quit“ beendet werden. Anstelle von gdb kann auch gdbtui verwendet werden (gdb mit Text User Interface). In LXDE steht Ihnen zudem ddd (Data display debugger) zur Verfügung.

Wenn dieses Beispiel auf Ihrem Rechner läuft, sind die notwendigen Tools für das Praktikum vorhanden, wenn auch noch etwas umständlich zu bedienen. Sie können somit mit den nächsten Aufgaben beginnen.

11. Weiteres

i. Umleiten von Standard-Kanälen (Redirecting)

Ein Prozess kennt drei verschiedene Standard IO Kanäle (streams):

- Eingabekanal (stdin): Eingabe über Tastatur in Terminal
- Ausgabekanal (stdout) Ausgabe auf Bildschirm (echo, printf, ...)
- Error – Ausgabekanal (stderr) Fehlermeldungen ebenfalls auf Bildschirm

Anstatt diese Standardkanäle zu verwenden, kann man diese auch umhängen, ohne dass der Prozess etwas davon wissen muss. Anstatt den Output eines Prozesses auf der Konsole auszugeben, kann dieser auch an eine Datei weitergeleitet werden. „ls“ gibt eine Liste auf dem Screen aus, mit „ls > file“ hingegen wird sie in einem File abgespeichert.

Die folgende Tabelle zeigt die verschiedenen Möglichkeiten auf, wie die Kanäle umgeleitet werden können:

Function	sh
Send stdout to file	prog > file
Send stderr to file	prog 2> file
Send stdout and stderr to file	prog > file 2>&1
Take stdin from file	prog < file
Send stdout to end of file	prog >> file
Send stderr to end of file	prog 2>> file
Send stdout and stderr to end of file	prog >> file 2>&1
Read stdin from keyboard until c	prog <<c
Pipe stdout to prog2	prog prog2
Pipe stdout and stderr to prog2	prog 2>&1 prog2

Quelle: http://www.linuxdevcenter.com/pub/a/linux/lpt/13_01.html

ii. Entfernen des Shell Scripts zur Änderung der IP-Adresse

Ein Script versucht bei jedem Neustart, die IP-Adresse zu ändern. Dies wurde eingerichtet, damit man die Adresse bereits verändern kann, während die SD-Karte noch mit einem Windows Computer verbunden ist. Um dies zu vermeiden, löschen Sie entweder das Textfile „/boot/ip_address.txt“ oder entfernen Sie alle Elemente, die zur Ausführung des entsprechenden Shell-Scripts führen:

- „sudo rm /etc/init.d/change_ip_address.sh“
- „sudo update-rc.d -f change_ip_address.sh remove“
- „sudo rm /boot/ip_address.txt“

Falls die Ausführung unterbunden wird, vergewissern Sie sich vorher, dass in /etc/network/interfaces bei eth0 die gewünschte IP-Adresse eingetragen ist.

iii. Bibliotheken

Auf das Raspberry Pi wurden bereits verschiedene Libraries kopiert, mit welchen auf die Ein- und Ausgänge der Hardware zugegriffen werden können. Dies erleichtert es zum Beispiel, ein Programm zu schreiben, das eine LED steuern kann.

- assembler Ein Hello-World Codebeispiel in ASM.
- bcm2835_lib Library mit verschiedenen Examples in C.
- RPi.GPIO-0.4.1a Source, um die GPIO Pins anzusteuern.
- wiringPi Weitere Sourcen für die Ansteuerung der GPIO Pins.
- wiringPi-Python Bibliothek für die Programmiersprache Python.

Diese Bibliotheken werden erst in späteren Praktika benötigt.