

Prof. Dr. Heinz Werntges  
Dipl.-Inform. (FH) Marcus Thoss, M. Sc.  
Marcus Michaely



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

Hochschule RheinMain  
Fachbereich Design Informatik Medien  
Studiengänge  
Angewandte Informatik / Technische Systeme

Praktikumsunterlagen

LV 1122

---

# **Einführung in die Informatik– Praktikum**

## **Laborversuche**

WS 2019/2020

---

Stand 13. November 2019

## Versuch 5

### Raspberry PI

Teilnehmer	
Gruppe (Team)	Versuchstag (Datum)

#### Lernziele

1. Linux-Embedded-PCs als Konzept
2. Verständnis der Rolle von Aktoren und Sensoren
3. I/O-Ansteuerung per Software

#### Was soll man wissen

- Umgang mit einer Linux-Shell
- Grundlegende Rechnerarchitektur
- Repräsentierung von Werten durch Spannungen

#### Geräte

- Gleichspannungs-Netzgerät
- Oszilloskop
- Raspberry PI 3 mit Ruby-Skript und C-Programm zur I/O-Ansteuerung

#### Aufgabenstellung

1. Schließen Sie den Raspberry PI an eine USB-Stromversorgung an. Loggen Sie sich mittels `ssh` auf dem PI ein, die IP-Adresse lautet 192.168.237.9X, wobei die Host-Endziffer X auf dem PI markiert ist (0..9). Username ist `EinfInf`, Passwort `R4s_P1`.<sup>1</sup>
2. Lassen Sie mittels vorgefertigtem Ruby-Skript eine Rechteck-Wellenform erzeugen, und stellen Sie diese auf dem Oszilloskop dar. Variieren Sie die Frequenz der Wellenform durch Veränderung des Quellcodes und beobachten Sie die Ergebnisse auf dem Oszilloskop.
3. Lassen Sie mittels eines C-Programms den Messwert des Helligkeitssensors ausgeben und beobachten Sie mit dem Oszilloskop die daraus erzeugte Ausgangsspannung. Vergleichen Sie diese mit der direkt am Sensor gemessenen Eingangsspannung. Ändern Sie das Programm so ab, dass es je nach Helligkeit das Wort `hell` oder `dunkel` ausgibt.

---

<sup>1</sup>Sollten Sie nach einer Passphrase gefragt werden, geben Sie nur `<Enter>` ein.

## Durchführung

### 5.1 Aufbau und Funktion

Ihr Experimentalsystem besteht aus einem Raspberry PI und aufgestecktem Gertboard (einer Erweiterungsplatine mit diversen Ein-/Ausgabemöglichkeiten). Für die Ausgabe eines digitalen Signals (Schritt 2) ist ein Kabel an Pin GP25 angeschlossen. Ein Helligkeitssensor (LDR, lichtabhängiger Widerstand) ist über einen Spannungsteiler an einem auf dem Gertboard vorhandenen A/D-Wandler an Pin AD1 angeschlossen (Schritt 3). Ein weiteres Kabel ist an den Ausgang eines D/A-Wandlers angeschlossen (ebenfalls Schritt 3).

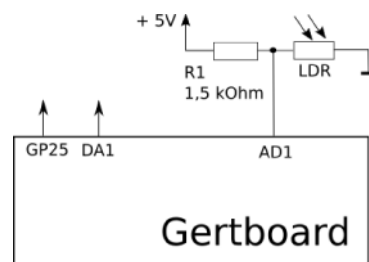


Abbildung 5.1: Schaltbild (Gertboard am Raspberry PI mit verwendeten Anschlüssen)

### 5.2 Vorbereitungen

Nach erfolgreicher Anmeldung befinden Sie sich in Ordner `/home/EinfInf`. Führen Sie das dort abgelegte Shell-Skript `reset_code.sh` aus: `$ ./reset_code.sh`. Damit wird der Ordner von eventuellen Überresten früherer Praktikumsnutzer bereinigt und in einen für Ihre Aufgaben definierten Anfangszustand versetzt. Sie sollten nur noch folgende Dateien vorfinden: `reset_code.sh`, `GP25_rect.rb`, `lichtsensor.c`, `Makefile`.

### 5.3 Rechteckkurve

Das für diesen Versuch verwendete Ruby-Skript ist `GP25_rect.rb`. Für die Ausführung benötigen Sie root-Rechte; starten Sie den Ruby-Interpreter `ruby` daher mit `sudo` oder wechseln Sie mit `sudo su` in eine Root-Shellumgebung.<sup>2</sup> Beenden Sie das Programm mit `Strg-C`.

Schließen Sie Kanal A bzw. 1 des Oszilloskops an Pin GP25 an, verbinden Sie die schwarze Krokodilklemme der Messleitung mit einem Massepunkt (GND) auf dem Gertboard und stellen Sie das Oszilloskop auf 2 V/div Vertikal- und 20 ms/div Horizontalvergrößerung ein.

Für die Veränderung der Frequenz der Rechteckkurve sehen Sie sich das Skript an. Wie muss es dafür angepasst werden? Ändern Sie es auf die halbe bzw. doppelte Frequenz.

### 5.4 Lichtsensor

Die Spannung zwischen dem LDR, einem lichtabhängigen Widerstand, und dem Festwiderstand ändert sich in Abhängigkeit von der Umgebungshelligkeit. Diese Spannung wird an Pin AD1 aufgenommen und in einen Digitalwert gewandelt. Das Programm `lichtsensor` gibt diesen Wert als Text aus und übergibt ihn außerdem an den Digital-Analog-Wandler des Gertboards, das eine dazu proportionale Analogspannung an Pin DA1 ausgibt.

<sup>2</sup>Beispiel-Aufruf: `$ sudo ruby ./GP25_rect.rb`

Schließen Sie Kanal A bzw. 1 des Oszilloskops an den Punkt zwischen den Widerständen und Kanal B bzw. 2 des Oszilloskops an Pin DA1 an, verbinden Sie die schwarze Krokodilklemme der Messleitung mit einem Massepunkt (GND) auf dem Gertboard und stellen Sie das Oszilloskop so ein, dass Sie die Linienposition gut ablesen können.

Für die Textausgabe von `hell` oder `dunkel` ändern Sie den C-Quellcode `lichtsensor.c` entsprechend ab und erzeugen sie `lichtsensor` neu durch Aufruf von `make`. Beobachten Sie zuvor die Werte, um einen geeigneten Umschaltpunkt zu finden.

Hinweis: Für deutliche Änderungen der gemessenen Spannung sollte der LDR einer kräftigen Lichtquelle ausgesetzt werden. Bewährt hat sich dafür z.B. die Taschenlampen-App von Smartphones.

### 5.5 Nachbereitungen

Wenn Ihre Anwendungen sich wie gewünscht verhalten, melden Sie sich für das Testat. Dafür zeigen Sie Ihrer Betreuerin bzw. Ihrem Betreuer die korrekte Funktion Ihrer Programme und beantworten Sie eventuelle Nachfragen. Den ausgefüllten und abgezeichneten Anleitungsbogen geben Sie ab. Erst NACH dieser Abnahme räumen Sie wie folgt auf:

- Dateien bzw. Ordner in Anfangszustand versetzen: `$ ./reset_code.sh`
- Raspi herunterfahren: `$ sudo shutdown now`  
Bem.: Die `ssh`-Verbindung zum Raspi wird dabei automatisch beendet
- Oszilloskop-Sonden abklemmen, Oszilloskop ausschalten
- Raspi-Netzteil vom Board trennen, vom Netz trennen, Kabel aufrollen
- Raspi-Netzteil und Platinen-Set zurück in die kleine graue Kiste legen, dabei die Kisten nicht verwechseln (Nr. der Kiste sollte mit Nr. auf Raspi übereinstimmen).
- Vom Linux-Rechner abmelden (wie immer)

Wer fertig ist, kann diesmal auch schon vor Ende der Praktikumseinheit gehen.