ulm university universität
**u**ulm

Universität Ulm
Institut für Eingebettete Systeme und Echtzeitsysteme

# Laborprojekt Robotik

Benjamin Menhorn

SS 2011

# Kapitel 1

# Einführung

## 1.1 Einordnung der Veranstaltung

Das Laborprojekt Robotik schließt an das Laborprojekt Eingebettete Systeme an, welches auf der Vorlesung Architektur Eingebetteter Systeme aufbaut. Ziel dieser Vorlesung war es, einen ersten Kontakt zu dem Design und Entwurf von Eingebetteten Systemen herzustellen. In den Übungen wurden dazu Grundlagen der Hardwarebeschreibung und das Zusammenspiel mit einem Softcoreprozessor vermittelt. Auf diesen Grundlagen aufbauend wurde in dem Laborprojekt Eingebettete Systeme begleitet durch Übungsblätter zum Abschluss eine Lüfterregelung entwickelt. Dabei wurde insbesondere auf die Aufteilung zwischen den Teilen welche in Hardware umgesetzt wurden und den Teilen, welche in Software umgesetzt wurden Wert gelegt. Dieses soll sich auch in diesem Laborprojekt nicht ändern. Da die Kenntnisse der beiden Veranstaltungen vorausgesetzt werden, besteht Ihre Aufgabe in diesem Projekt unter anderem auch darin eine sinnvolle Partitionierung zu finden.

## 1.2 Grundlagen

Wie bereits angedeutet werden profunde Kenntnisse im Umgang mit der Hardwarebeschreibungssprache VHDL, der Programmiersprache C und den Entwurfswerkzeugen von Altera (Quartus II und NIOS II) vorausgesetzt. Sollten Sie die Veranstaltungen Architektur Eingebetteter Systeme und/oder Laborprojekt Eingebettete Systeme nicht besucht haben, sollten Sie selbständig die Übungen der bei-

den Vorlesungen durchgehen. Diese finden sie auf der Webseite des Institutes http://www.uni-ulm.de/in/esys im Archiv. Vor allem die Übung zur Vorlesung Architektur Eingebetteter Systeme hat ein exzellentes Übungsskript.

Sollten Sie (beispielsweise durch den Wechsel der Universität oder den Besuch gleichwertiger Veranstaltungen) zwar mit den grundlegenden Konzepten vertraut sein, Ihnen aber lediglich die Entwicklungsumgebung oder die Programmiersprache nicht geläufig sein, stellt dieses kein größeres Hindernis dar. Altera bietet auf ihrer Webseite http://www.altera.com ausführliche Tutorials zu ihren Entwurfswerkzeugen an (diese finden Sie unter "Training" → "University Program"), so wie eine ausführliche Dokumentation zu Quartus II und Nios II (diese finden Sie unter "Literature" → "Quartus II Handbook"). Zum Erlernen der Beschreibungssprache VHDL gibt es zahlreiche Tutorials in der Bibliothek oder im Internet. Auch für die Programmiersprache C stehen Ihnen unzählige Quellen zur Verfügung.

Es soll nochmals darauf aufmerksam gemacht werden, dass es nicht zum Kern dieser Veranstaltung gehört, den Umgang mit den Entwurfswerkzeugen, die Programmiersprache C oder die Hardwarbeschreibungssprache VHDL zu lernen. Diese Kenntnisse werden für den Besuch der Veranstaltung vorausgesetzt. Die Vergangenheit hat jedoch gezeigt, dass es engagierten Teilnehmern und Teilnehmerinnen problemlos möglich ist, Wissenslücken selbständig innerhalb kürzester Zeit zu schließen.

## 1.3 Ziele

Nachdem wir Ihnen im vorherigen Abschnitt erklärt haben was wir von Ihnen erwarten, wollen wir kurz darauf eingehen was Sie aus dem Laborprojekt mitnehmen können. Wie der Titel der Veranstaltung bereits andeutet befasst sich die Veranstaltung mit Robotik. Da es sich um ein Laborprojekt handelt soll an die Aufgabenstellung sehr praktisch angegangen werden. Dabei sollen jedoch typische Konzepte eines Eingebetteten Systems keinenfalls vernachlässigt werden.

Die vereinfacht dargestellte Aufgabenstellung ist, einen Laufroboter anzusteuern. Dieser soll bestimme Aufgaben erfüllen können. Die wohl grundlegendste Aufgabe ist das Laufen an sich. Sie werden feststellen, dass dieses bereits eine nicht zu unterschätzende Anforderung darstellt. Weiterhin soll der Roboter Hindernisse durch Dagegenstoßen erkennen und diesen Ausweichen können.

Egal ob Sie sich später im wissenschaftlichen oder industriellen Bereich mit Eingebetteten Systemen beschäftigen, werden Sie feststellen, dass der Umgang mit FPGA zunehmend an Bedeutung gewinnt. Durch immer kostengünstigere Module werden diese heute nicht mehr nur bei Prototypen oder Kleinserien eingesetzt, sondern auch in der Massenfertigung. Einer der wichtigsten Vorteile beim Einsatz von FPGAs ist, dass auch nach Fertigstellung des Produktes das FPGA umkonfiguriert werden kann. Werden beispielsweise weitere Hardwarekomponenten benötigt oder zusätzlicher Speicher, kann dieses durch ein "Softwareupdate" bereitgestellt werden, ohne dass die Komponente ausgetauscht werden muss.

Im Anschluss an diese Veranstaltung wird es Ihnen möglich sein selbständig kleinere Systeme umzusetzen und dabei die Vorteile der jeweiligen Möglichkeiten zu kennen. Sie werden in einem Projektteam darstellen und begründen können, warum es beispielsweise sinnvoller ist, eine weitere Hardwarekomponente zu verwenden anstatt einen größeren Prozessor einzubauen (diese Aussage gilt natürlich nicht pauschal).

# Kapitel 2

# Grundlegendes

## 2.1 Laufroboter

Unsere Laufroboter bestehen aus sechs Servos. Durch die Konstruktion wird ein Teil des menschlichen Bewegungsapparates nachgestellt. Jedoch haben die Roboter nur einen begrenzten Bewegungsablauf. Vor allem der hohe Schwerpunkt des Roboters, der durch die Batterie und den Aufbau gegeben ist, stellt an die Bewegung eine Herausforderung dar. Zudem fehlt die Möglichkeit, durch Oberkörper und Arme das Gleichgewicht zu halten. Allzu ruckartige Bewegungen führen meist zu einem Kippen des Roboters oder zumindest zu einer instabilen Position. Vor allem bei Ihren ersten “Gehversuchen” (aber auch später) achten Sie bitte darauf, dass der Laufroboter sich weit genug vom Rand des Tisches weg befindet und sich eine Unterlage unter dem Roboter befindet. Die Unterlage erhöht auch den Grip des Roboters, da diese auf den Tischen bedingt durch die glatten Metallfüße rutschen.

## 2.2 Analogservos

Analoge Servos, wie sie vor allem im Modellbau üblich sind, sind meist Motoren, die mit Gleichstrom angesteuert werden. Ein Servo hat einen Stecker mit drei Verbindungen: die Versorgungsspannung ($V_{CC}$), die Masse (Gnd) und die Kontrollleitung (PW). Die Steckerbelegung ist in Abbildung 2.1 zu sehen. Da die Versorgungsspannung in der Steckermitte anliegt hat ein versehentliches um 180 Grad gedrehtes Einstecken keine negativen Folgen.


**Abb. 2.1** Servostecker

Zur Ansteuerung der Servomotoren wird eine Art PWM-Signal verwendet, bei dem die Winkelstellung der Servos von der Pulsdauer des Signals abhängig ist. Die Periodendauer dieses Signals liegt üblicherweise bei 20 ms, also 50 Hz. Dieses ist in Abbildung 2.2 dargestellt.

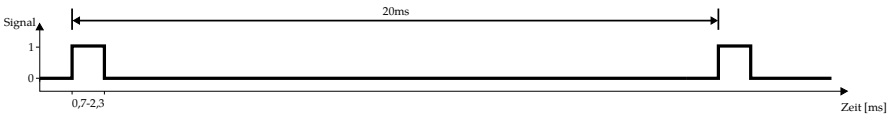


**Abb. 2.2** PWM-Signal zur Servosteuerung

Die Stellung der Servo ist von der Dauer des High-Signales abhängig. Dabei entspricht eine Dauer von 1,5 ms der Mittelstellung. Durch eine Verringerung der Signaldauer auf 0,7 ms bzw. eine Verlängerung auf 2,3 ms wird eine Auslenkung nach links bzw. rechts erreicht. Die verbleibende Zeit der Periodendauer (zwischen 17,7 ms und 19,3 ms) bleibt das Signal auf auf Low. Abbildung 2.3 zeigt das die Abhängigkeit der Servostellung von der Dauer des High-Signals für die Mittelstellung und die Maximalauslenkungen von -90° und 90° bei einem 180° Servo.

Ein Nachteil der Servos ist, dass sich die aktuelle Position der Servos nicht abfragen lässt. Zudem gibt der Servo kein Feedback, ob diese die zu erreichende Position auch erreicht hat bezeihungsweise überhaupt erreichen kann. Dieses ist nur indirekt über die Messung der Stromaufnahme möglich. Mehr dazu später.

## 2.3 Puls-Pausen-Modulation

Damit nicht jeder Servo eine eigene Verbindung zur Signalquelle benötigt wird oft die Puls-Pausen-Modulation (PPM) eingesetzt. Diese bietet sich vor allem im Modellbau mit nur einem Funkkanal an, dort wird das Signal auch Summensignal ge-

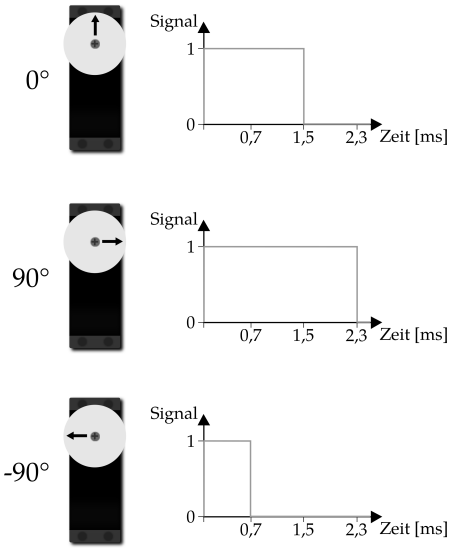


**Abb. 2.3** Servostellung und Signal

nannt. Bei der Puls-Pausen-Modulation handelt es sich um ein zeitmultiplex Verfahren, das die Pausen zwischen den einzelnen Steuerimpulsen ausnutzt.

Nach einer etwas längeren Synchronisationsphase von mehreren Millisekunden High-Pegel, werden die Pulslängen der einzelnen Kanäle nacheinander durch steigende Flanken codiert. Für die „Erzeugung“ der Flanken ist daher jeweils eine kurze Low-Phase von ca. 0,3 ms notwendig. Die resultierende Pulslänge der jeweiligen Kanäle wird also von der Zeit von einer steigenden Flanke bis zur jeweils nächsten erzeugt. Bei der Veränderung des Signals eines Kanals werden daher die Pulse der nachfolgenden Kanäle entsprechend nach vorne bzw. hinten verschoben, auch die Sync-Phase verkürzt oder verlängert sich entsprechend, damit die Länge eines Frames konstant bleibt. Die im Modellbau gängigen Empfänger benötigen ein PPM-Signal mit mindestens vier Kanälen, üblicherweise sind bis zu sieben Kanäle codierbar. Ein Beispiel mit sieben Kanälen sehen Sie in Abbildung 2.4.

Der Laufroborter hat sechs Servos. Das Board an den Robortern stellt eine Dekodierung von sieben Servosingal mit entsprechenden Anschlussmöglichkeiten zur Verfügung. Für den vorliegenden Fall, reicht es aus, wenn ein PPM-Signal zur Ansteuerung von sechs Servos erzeugt wird. Die theoretisch vorgesehen Zeit für den siebten Servo kann als High-Signal in die Synchronisationsphase eingehen. Es ist nicht notwendig, eine weitere kurze Low-Phase zu erzeugen.

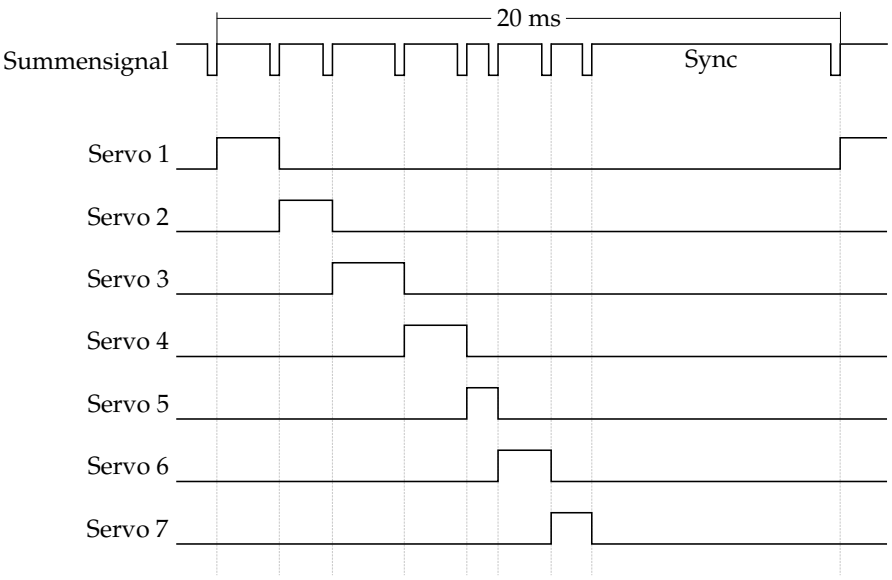


**Abb. 2.4** Dekodierung der Puls-Pausen-Modulation

## 2.4 Kraftfeedback

Die von uns verwendeten einfachen analogen Servomotoren benutzen für die Nachstellung eine PWM-Frequenz welche gleich der von außen über das Steuersignal vorgegeben Frequenz ist.

Der im Servo verbaute Gleichstrommotor wird also mit immer gleicher Spannung betrieben, die Regelung der Kraft erfolgt über die Einschaltdauer zwischen den Steuerpulsen. Dabei wird die Regelabweichung während der Pulse bestimmt und damit die Einschaltdauer bis zum Impuls festgelegt. Der zeitliche Ablauf wird in Abbildung 2.5 verdeutlicht.

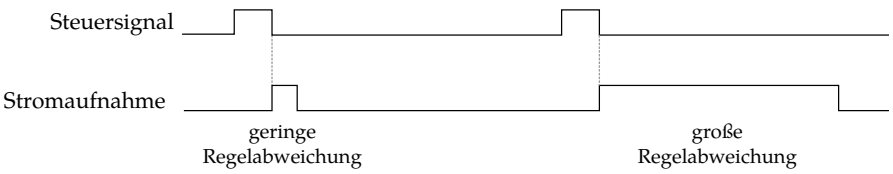


**Abb. 2.5** Zeitverkauf der Regelung und Stromaufnahme

## 2.5 Pinbelegung und Timing

Der Laufroboter ist bereits so vorbereitet, dass er direkt an das DE-2 Board angeschlossen werden kann. Auf dem Board stehen zwei Steckbänke zur Verfügung. Die Pinbelegung der GPIO-Bank (general purpose input output) ist in Abbildung 2.6 dargestellt. Es stehen Ihnen zwei GPIO-Bänke zur Verfügung. Der Stecker eines Laufroborters ist so ausgelegt, dass er eine komplette GPIO-Bank benutzt. Die Pin-Belegung können Sie Tabelle 2.1 entnehmen.

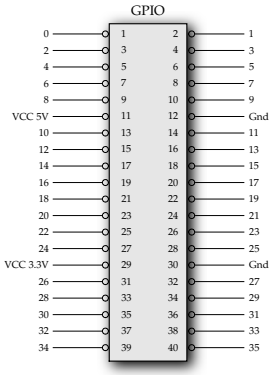


**Abb. 2.6** GPIO

| GIOP-Pin | Belegung | Richtung |
|---|---|---|
| 10 | Summensignal | FPGA → Roboter |
| 12 | Servolast 1 | Roboter → FPGA |
| 13 | Servolast 2 | Roboter → FPGA |
| 14 | Servolast 3 | Roboter → FPGA |
| 15 | Servolast 4 | Roboter → FPGA |
| 16 | Servolast 5 | Roboter → FPGA |
| 17 | Servolast 6 | Roboter → FPGA |
| 18 | Servolast 7 | Roboter → FPGA |

**Tabelle 2.1** Pin-Belegung

Sie können das zweite GPIO-Interface dazu verwenden, um sich beispielsweise Ihr PPM-Signal oder die Stromaufnahmesignale der Servos auf einem Oszilloskop zeigen zu lassen. Hier verbinden Sie einfach Ihr PPM-Signal mit einem Pin der freien GPIO-Bank (z.B. `GPIO_1(0) <= ppm_signal`) und die Eingangssignale

der Stromaufnahmemessung von einem GPIO-Interface mit einem Ausgangssignales auf dem anderen GPIO-Interface (z.B. `GPIO_1(7 downto 1) <= GPIO_0(18 DOWNTO 12)`). Lassen Sie sich von Ihrem Betreuer zuvor eine Einweisung zum verbinden des Oszilloskops mit dem Board sowie zur grundlegenden Benutzung eines solchen Gerätes geben.

# Kapitel 3

# Umsetzung

Wie bereits in der Einleitung erwähnt wird bei der Umsetzung ein hohes Maß von Selbständigkeit verlangt. Daher befasst sich dieser Abschnitt hauptsächlich mit Anregungen zur Umsetzung. Die konkrete Umsetzung soll von Ihnen erarbeitet werden und dokumentiert werden.

## 3.1 Hardware

Wie bereits beim ersten Lesen aufgefallen sein sollte, werden Sie Hardwarekomponenten implementieren müssen. Für die Realisierung der Komponente(n) ist es hilfreich, zuvor ein Blockschaltbild mit dem Aufbau anzufertigen. Planen Sie hierbei notwendige Komponenten und deren Verschaltung. Bei der Umsetzung ist zu beachten, dass das Timing eingehalten wird und die Auflösung groß genug gewählt wird. Überlegen Sie sich, wie Sie möglicherweise ohne Rechnen die variable Länge der Sync-Phase sowie die konstante Länge der Frames erzeugen können. Zum Anderen ist darauf zu achten, dass das Zugriff auf den Avalon-Bus (durch Schreiben und Lesen) synchron zum Systemtakt erfolgt, dort wo es erforderlich ist.

Ihre Hardware sollte am Ende aus einem Softcoreprozessor bestehen, der als Speicher den SDRAM verwendet. Weiterhin sollten sowohl Timer, JTAG-UART, die serielle Schnittstelle RS-232 als auch Ihre Hardwaremodule als Komponenten eingebunden werden. Je nach Umsetzung sind noch weitere Komponenten, wie PIOs für Schalter, Taster oder LEDs notwendig.

## 3.2 Software und Treiber

Eine komplette Umsetzung des Projektes in Hardware scheint wenig sinnvoll zu sein. Daher soll die “Denkarbeit” von Software übernommen werden. Für die Kommunikation zwischen Software und Hardware ist es erforderlich Treiber zur entwickeln. Diese sollen auf der Hardwareseite über den Avalon-Bus mit Ihren Komponenten kommunizieren und auf der Softwareseite über Methoden mit Ihrem Programm. Innerhalb hat der Treiber mehrere Funktionen. Dem Programmierer soll es möglich sein, beispielsweise den Servowinkel in Grad einzustellen und nicht in Anzahl der Taktzyklen oder ähnliches. Gleiches soll auch für die Rückmeldung der Kraft gelten. Hierzu sollten Sie sich einen eigenen Maßstab entwickeln, der die Kraft angibt.

Der Benutzer wird hauptsächlich durch die Steuerung des Roboters mit Ihrer Software in Berührung kommen. Hierzu ist es notwendig, dass der Benutzer sinnvolle und einfach zu verstehende Eingaben machen kann. Entwerfen Sie hierzu eine (einfache) benutzerfreundliche Schnittstelle zur Steuerung des Roboters. Weiterhin soll der Benutzer auch Feedback bekommen, beispielsweise in welchem Zustand der Roboter sich gerade befindet (stehen, laufen, ...).

## 3.3 Anforderungen

Allgemein soll Ihr Roboter am Ende laufen können, Hierzu gehört die Eigenschaft, dass diese sowohl vorwärts als auch rückwärts laufen kann und sich in beide Richtungen drehen kann. Weiterhin soll der Roboter selbständig ein Hindernis erkennen und dieses umgehen. Zuletzt soll der Roboter auf Befehl einen leichten Gegenstand umtreten können. Die genaueren Anforderungen, die erfüllt werden sollen sind:

Kalibrierung: Damit der Laufroboter aufrecht steht muss die genaue Mittelstellung gefunden werden. Baulich bedingt, weicht diese bei den Robotern von der Mittelstellung der Servos ab. Verschiedene Roboter haben jeweils eine eigene “Kalibrierung” für die Mittelstellung. Es sollte Ihnen möglich sein, die Konfiguration für verschiedene Roboter zu speichern und zu laden.

Lernmodus: In der Software soll es möglich sein, dem Roboter neue Bewegungsabläufe beizubringen. Hierzu werden die einzelnen Servos an eine bestimmte Position gefahren, und diese Position dann gespeichert.

Steuerung: Die Steuerung soll über die Tastatur des Rechners erfolgen. Die Datenübertragung zum DE2-Board soll dabei über die serielle Schnittstelle (RS-232) erfolgen. Das Feedback des Roboters wird ebenfalls über ein Terminal ausgegeben.

Betriebssystem: Verwenden sie zur Umsetzung der Software Tasks und das Echtzeitbetriebssystem µC-OS II.

Regelung: Die Servos haben intern nur einen P-Regler. Sobald eine neue Position eingestellt wird, fährt die Servo mit diese so schnell wie möglich an. Dieses entspricht aber nicht dem natürlichen Bewegungsablauf. Sie sollte hierzu eine Rampenfunktionalität bereit stellen. Diese beschleunigt die Servo langsam, hält diese dann auf einer konstanten Geschwindigkeit und bremst diese wieder ab. Die Umsetzung erfolgt dadurch, dass sie die Zielposition nicht sofort einstellen, sondern diese immer wieder in Zwischenschritten aktualisieren und dabei die Schrittweite anpassen. Überlegen Sie sich, wo dieses erledigt wird. Sie müssen hierzu natürlich die Position der Servo speichern. Über das Kraftfeedback können Sie auswerten, ob die gewünschte Position auch erreicht wurde.

Aufschwingen: Sie werden feststellen, dass (spätestens) wenn Ihr Roboter auf einem Bein steht, diese durch leichtes Antippen zu Schwingen beginnt. Dieses soll von Ihnen durch die Ausnutzung des Kraftfeedbacks verhindert werden. Eine Lösung ist den internen Regler der Servo durch einen von ihnen in Hardware oder Software entworfenen I-Regler zu erweitern.

# Kapitel 4

# Ausarbeitung

Der Abschluss Ihres Projektes besteht aus zwei Teilen: zum einem werden Sie die Funktionalität des Laufroboters demonstrieren und zeigen, dass er die Anforderungen erfüllt. Zum Anderen wird eine Ausarbeitung verlangt, die Aufschluss über Ihre Realisierung gibt. Bei der Ausarbeitung ist besonders auf folgendes einzugehen:

Aufteilung: Welche Teiles des Projektes wurden in Hardware und welche Teile wurden in Software umgesetzt. Warum wurde die Aufteilung so gewählt? Welche Konzepte stehen dahinter?

Aufbau der Hardware: Wie hängen der Softcoreprozessor und die Hardwarekomponenten zusammen?

Avalon-Bus: Was ist der Avalon-Bus? Wie funktioniert er? Was ist ein Master und was ein Slave?

PPM und Kraftmessung: Wie sehen Ihre selbst entwickelten Komponenten aus? Wie haben Sie diese realisiert? Wie stellen Sie die Taktung und das Timing sicher?

Betriebssystem: Was ist die Aufgabe eines Echtzeitbetriebssystem? Was sind Task? Wie haben Sie Tasks eingesetzt?

Anforderungen: Wie haben Sie die oben beschriebenen Anforderungen umgesetzt? Wie haben Sie die Herausforderungen gelöst? Gibt es auch andere Lösungen?

Ergebnis: Wie ist das Ergebnis ihres Projektes? Welche Probleme bestanden/ bestehen?