Praktikum 25./26. Oktober 2010

Jörn Loviscach

Versionsstand: 28. Oktober 2010, 23:27

In diesem Praktikum soll ein Fotowiderstand durch einen Modellbau-Servo automatisch ins optimale Licht gedreht werden.

Aufgabe zu Hause als Vorbereitung: Wie muss die Funktion für die ersten beiden Aufgaben (siehe nächste Seite) in C aussehen? Schreiben Sie die auf oder programmieren Sie die ohne Servo.

Erzeugen Sie ein neues C-Projekt in der IAR Embedded Workbench (Einstellungen wie bisher, also Device: MSP430G2231, Compiler: keine Optimierung, Driver: FET Debugger). Kopieren Sie die Dateien servo01.h und servo01.c neben die Datei main.c des Projekts. Fügen Sie in der Projektverwaltung der IAR Embedded Workbench die Datei servo01.c per rechtem Mausklick hinzu (Add > Add Files). Ergänzen Sie am Anfang der Datei main.c die Zeile `#include "servo01.h"`. Nun stehen [c1] Ihnen vier Funktionen zur Verfügung:

[c1] *removed text by jl:* vier

- `initialize` muss ganz zu Beginn aufgerufen werden. Es nimmt einige Grundeinstellungen vor. Die erste Zeile in `main` sollte deshalb `initialize();` lauten. Das bisherige `WDTCTL = WDTPW + WDTHOLD;` ist nicht mehr nötig.
- `readAnalog` hat keine Parameter und liefert einen Wert von 0 bis 1023, welcher linear von der Spannung an Pin 1.5 abhängt.
- `setServoTo` stellt einen Servo ein. Dessen Steuereingang muss an Pin 1.2 angeschlossen sein. Als Argument wird die gewünschte Position des Servos als Zahl von 0 bis 100 übergeben. Zahlen außerhalb des Bereichs 0 bis 100 machen nichts kaputt, zumindest nicht sofort. Der Servo dreht aber nicht beliebig weit, sondern erreicht irgendwann seine mechanische Grenze.
- `waitMilliseconds` hält das Programm für die als Argument angegeben Zahl an Millisekunden an.

Schließen Sie den 1 kΩ-Widerstand, den Fotowiderstand und den Servo so an:

- Widerstand und Fotowiderstand bilden einen Spannungsteiler zwischen V_{CC} (positive Versorgungsspannung, hier etwa 3,6 V) und GND (Ground = Masse).

Das eine Bein des Fotowiderstands wird mit V_{CC} verbunden; das eine Bein des Widerstands mit GND. Die Mitte des Spannungsteilers wird mit Pin 1.5 verbunden.

- Der Servo wird mit Spannung versorgt (rot: V_{CC}, braun: Masse) und mit seiner Steuerleitung (orange) an Pin 1.2 angeschlossen. (Diese Farben sind je nach Hersteller leicht verschieden. Die dunkelste ist Masse, die hellste die Steuerleitung.)
- Befestigen Sie den Fotowiderstand am Hebel des Servos.

Programmieraufgaben:

1. Schreiben Sie eine Funktion mit dem Prototypen[c1] `int setServoAndRead(int percentage)`, die den Servo auf die in Prozent angegebene Position stellt, dann 300 Millisekunden wartet, um dem Servo Zeit zum Bewegen zu geben, und zum Schluss die Spannung misst und als Ergebnis zurückliefert. Deklarieren Sie diese Funktion am Anfang der C-Datei nach den `#include`. Definieren Sie diese Funktion am Ende der C-Datei. Schreiben Sie dazwischen eine `main`-Funktion, die einige Aufrufe von `setServoAndRead` testet.

   [c1] jl: der Signatur

2. Speichern Sie in `setServoAndRead` die jeweils zuletzt eingestellte Position in einer `static`-Variablen. Benutzen Sie das, um die Wartezeit für kleine Drehungen zu verkürzen: Für eine sehr kleine Drehung soll nur 200 Millisekunden gewartet werden. Die Wartezeit soll mit der Größe der Drehung linear bis auf 400 Millisekunden wachsen. Testen Sie das mit Aufrufen in `main`.
3. Schreiben Sie eine `main`-Funktion, die den Servo langsam schwenkt und den größten Spannungswert sowie die dazugehörige Servostellung in Variablen speichert [c2]. Nach dem Schwenken soll der Servo auf diese beste Position gestellt werden.

   [c2] *removed text by jl:* werden
4. Nun soll das Programm so erweitert werden, dass es einmal schwenkt, um die aktuell beste Position zu suchen, aber dann die Position langsam nachführt, um Änderungen des Lichts zu berücksichtigen. Dazu dreht es den Servo regelmäßig etwas nach rechts und nach links und misst die Spannungen dort. Sollte die Spannung irgendwo größer sein, gilt das nun als die optimale Position. Versuchen Sie, das so zu programmieren, dass der Servo nie auf Positionen unter 0 oder über 100 gestellt wird.
5. Je nach verbleibender Zeit erweitern Sie das Programm zum Beispiel so:
   - Die Leuchtdioden auf dem Board zeigen an, in welche Richtung der Servo gerade fährt.
   - Das Programm führt regelmäßig wieder einen vollen Schwenk durch, um keine Änderungen zu verpassen, die weit weg von der aktuellen Position passieren.
   - Mehrere Messungen rechts und links werden gemittelt, um zu entscheiden, in welche Richtung der Servo nachgeführt wird.