

Praktikum 18. Oktober 2010

Jörn Loviscach

Versionsstand: 26. Oktober 2010, 18:45



This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Germany License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/de/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

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

Aufgabe zu Hause als Vorbereitung: Skizzieren Sie die Programme der ersten beiden Aufgaben (siehe nächste Seite) jeweils als Flussdiagramm oder Struktogramm.

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 neue Befehle zur Verfügung:

^{c1} removed text by jl: vier

- `initialize` muss ganz zu Beginn aufgerufen werden. Es nimmt einige Grundeinstellungen vor. Die erste Zeile im Programm sollte deshalb `initialize();` lauten. Das bisherige `WDTCTL = WDTPW + WDTHOLD;` ist nicht mehr nötig.
- `readAnalog` liefert einen Wert von 0 bis 1023, welcher linear von der Spannung an Pin 1.5 abhängt. Sie können diesen Wert zum Beispiel einer Variablen zuweisen: `int value = readAnalog();`
- `setServoTo` stellt einen Servo ein. Dessen Steuereingang muss an Pin 1.2 angeschlossen sein. Den üblichen Schwenkbereich des Servos erreicht man mit Zahlen von 0 bis 100. Das geht direkt wie `setServoTo(50);` oder mit Ausdrücken wie `setServoTo(3*x+42);`. Zahlen außerhalb des Bereichs 0 bis 100 machen nichts kaputt; der Servo dreht aber nicht beliebig weit, sondern erreicht etwa bei 0 bzw. bei 100 seine mechanische Grenze.
- `waitMilliseconds` hält das Programm eine Weile an, zum Beispiel mit `waitMilliseconds(500);` für eine halbe Sekunde. In den Klammern kann wieder ein berechneter Ausdruck stehen: `waitMilliseconds(3*x+42);`

Schließen Sie den 1k Ω -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.^{c1} 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.

^{c1} text added by jl

Programmieraufgaben:

1. Schreiben Sie ein Programm, das den Servo langsam schwenkt und dann schnell zurückfährt.
2. Lassen Sie das Programm nun beim Schwenken die Spannungswerte einlesen. Der größte Spannungswert und die dazugehörige Servostellung sollen in Variablen gespeichert werden. Nach dem Schwenken soll der Servo auf diese beste Position gestellt werden.
3. Nun soll das Programm so erweitert werden, dass es erst die beste Position sucht, aber dann bei Änderungen des Lichts die Position langsam nachführt. Dazu dreht es den Servo regelmäßig etwas nach rechts und nach links und misst die Spannungen dort. Sollte die Spannung dort größer sein, gilt das nun als die optimale Position.
4. 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.
 - Alle zehn Sekunden führt das Programm 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.