Grundlagen hardwarenahe Programmierung (GHP)

Prof. Dr. Manzke – FH-Kiel – robert.manzke@fh-kiel.de

Laboraufgabe 2: MSP430 Microcontroller-Programmierung

Zielsetzung Versuch 2

Einstieg in die Programmierung von einfachen Microcontrollern am Beispiel des TI MSP430 Launchpad. Verwendung der Timer und des ADCs. Programmierung von Interrupt-Handlern und einfache digitale Signalverarbeitung. Untersuchung welchen Maschinencode der C-Compiler erzeugt und wie im Ansatz Quellcode optimiert werden kann, um die begrenzten Resourcen des Chips optimal auszunutzen.

Voraussetzungen / Grundlagen:

Durcharbeiten der Skripte und Web-Quellen (Datenblätter). Aufbau des MSP430, insbesondere Interrupt-Handler, Timer, PWM und Analog Digital Wandler (ADC10).
Der ADC wird angesteuert über ein Spannungsteiler Potentiometer (siehe Schaltplan im Anhang).

Gleitender Mittelwert-Filter, gewohnte Implementation (N ist Anzahl der Mittelungen, $\overline{x}$ ist Mittelwert, i, j sind Indizes):

$$\overline{x}_i = \frac{1}{N} \cdot \sum_{j=i-(N-1)}^{i} x_j$$

Gleitender Mittelwert, Rekursionsgleichung (nicht zu verwechseln mit rekursiven Funktionsaufrufen beim Programmieren!):

$$\overline{x}_{i+1} \cdot N = \overline{x}_i \cdot N - x_{i-N+1} + x_{i+1}$$

Durchführung:

Falls Sie das Launchpad *selbst* gekauft haben (bspw. Farnell, Watterott etc.):

1. Laden Sie Code Composer V5.x+ für das MSP430 Launchpad herunter und installieren Sie die Code Composer Software, sowie die USB Gerätetreiber für das MSP430 Launchpad Board (http://processors.wiki.ti.com/index.php/MSP430_LaunchPad_(MSP-EXP430G2), http://processors.wiki.ti.com/index.php/CCS_for_LaunchPad, TI Account Registrierung notwendig).
2. Verbinden Sie entsprechend des u.a. Schaltplanes ein Potentiometer, sowie eine LED mit bspw. 270 Ohm Vorwiderstand an die entsprechenden Pins. Benutzen Sie die mitgelieferten Steckleisten (bspw. anlöten).

Aufgabenteil A, Analog-Digital-Converter (ADC)

1. Erstellen Sie ein neues MSP430 Grace Projekt im Code Composer. Verwenden Sie den M430G2553 Chip (bzw. den, der auf ihrem Launchpad installiert ist).
2. Im „Grace Welcome Screen" selektieren Sie „Device Overview" und konfigurieren Sie:
    a. 3.6V,
    b. Oszillator (Basic User), High Speed Clock 16MHz, Low Speed Clock 12kHz,
    c. Timer0_A3 (Basic User), PWM Mode, TA0.0 Output Off, 10ms Periode, 50% Duty Cycle (CC Block 1), TA0.1 Output Off, CC Block 2 Off,
    d. ADC10 (Power User), Single Channel A7/ADC Channel 7, System GND, System VCC, Repeated Conversion, 4xADC10CLKs, ADC Clock Source MCLK, ADC Trigger Source Timer_A3 Channel 1, ADC10 Interrupt Enabled,
    e. GPIO, Pinout 20-TSSOP/20-PDIP, P1.0 GPIO Output, P1.6 GPIO Output.
3. Im ADC10 Interrupt Handler (src/grace/InterruptVectors_init.c), implementieren Sie Code, der Bit 6 von Port 1 (angeschlossen an LED Pin Port 1.6) in Abhängigkeit des ADC-Wertes über einen Schwellwert (bspw. 511) High / Low schaltet. Schauen Sie sich das LED-Verhalten an, wenn der ADC-Wert nah am Schwellwert ist. Achten sie darauf, dass eigener Code nur zwischen den von Grace eingefügten Kommentaren platziert wird. Ansonsten wird der Code beim erneuten Aufruf von Grace überschrieben!
4. Im ADC10 Interrupt Handler, implementieren Sie den gleitenden Mittelwert Filter nach der Rekursionsgleichung oben (verwenden sie den *fixed point* Datentyp *unsigned int*). Definieren Sie dabei das „N" über eine Compiler-Direktive „#define MWN 16" (damit kann man dann später unterschiedliche Werte für N testen). Die Messwerte für den Filter sollen die ADC-Werte sein. Vergleichen Sie nun anstelle des direkten ADC-Wertes den Gemittelten mit dem Schwellwert und schalten Sie entsprechend die LED. Schauen Sie sich das LED-Verhalten an, wenn der ADC-Wert nah am Schwellwert ist. Experimentieren Sie mit Ihrem Code so, dass Sie in der Lage sind, die Fragen im nachstehenden Teil zu beantworten.

Aufgabenteil B, PWM-Dimmer

1. Erweitern Sie den Interrupt-Handler des ADC10 um Steuerung des PWM-Verhältnisses (Duty-Cycle) des Timer1_A3.
   Konfigurieren Sie (Grace) den Timer1_A3 auf PWM mit 10ms Periode. Setzen Sie CC Block #2 auf PWM mit Ausgang auf Port 2.4. An Port 2.4 ist eine weitere LED angeschlossen, die Abhängig von Duty-Cycle hell bzw. dunkel leuchtet.
   Im ADC10 Interrupt-Handler soll das CC-Register 2 von Timer1_A3 mit dem gemitteltem ADC Wert so konfiguriert werden, dass man mit dem Potentiometer das PWM-Verhältnis setzten kann. Achten Sie darauf, dass Sie die LED ganz hell / dunkel schalten können (PWM-Verhältnis 0-100%).
2. Implementieren Sie außerdem Code, der den PWM-Glitch beim Übergang des CC-Register 2 von „1 auf 0“ verhindert (durch manuelles Setzen des Ausgangs an Port 2.4, entsprechende Änderungen an der GPIO-Konfiguration sind notwendig im Code, nicht Grace). Experimentieren Sie mit der PWM-Steuerung entsprechend, um die Fragen im Folgenden beantworten zu können.

Aufgabenteil C, Interrupt und Round Robin mit Taster

1. Implementieren Sie einen Interrupt-Handler für den Taster S2 (an Port 1.3) auf dem Entwicklerboard. Beim Drücken des Tasters S2 soll die LED an Port 1.0 toggeln. Achten Sie darauf, dass Port 1.3 auf Input mit Pull-Up und Interrupt Rising-Edge gesetzt ist. Benutzen Sie dazu den GPIO Dialog in Grace.
2. Implementieren Sie nun anstatt des Interrupt-Handlers für den Taster eine Round-Robin Programmstruktur. Das Toggeln soll nun durch Abfragen des Port 1.3 anstelle der Interrupt-Routine erfolgen.
3. Experimentieren Sie mit dem Taster und dem Interrupt Handler bzw. der Round-Robin Implementation, bis Sie die Fragen aus der Fragensektion unten beantworten können.

Spezifische Fragen zum Versuch (vom Laborpersonal werden diese bei der Abnahme überprüft, und die Antworten sind im Laborbericht explizit zu vermerken; die richtige Beantwortung der Fragen ist Voraussetzung für das Testat!):

1. **Fragen zu Teil A, ADC**
   a. Wie verändert sich das Verhalten der LED-Umschaltung beim Überschreiten des Schwellwertes in Abhängigkeit von der Benutzung des
      i. Direkten ADC-Wertes,
      ii. Gleitenden Mittelwertes des ADC-Wertes mit verschiedenen Anzahlen von Mittelungen (bspw. N={4, 8, 16, 32, 64}).
   b. Welche nachteiligen Effekte bestehen bei der Wahl von großen N (bspw. N={128, 1024})?
   c. Was ist zu beachten bei der Wahl von N? Schauen Sie sich an, was mit dem Assemblat passiert, wenn N als Zweierpotenz bzw. nicht als Zweierpotenz gewählt wird, was sind die Unterschiede?

d. Welche Vorteile bestehen bei der Mittelwertberechnung unter Benutzung der rekursiven Formel im Vergleich zur Regulären?
e. Wie könnte man das Verhalten der LED-Umschaltung auch einfacher ohne Filter erreichen?

2. **Fragen zu Teil B, PWM-Dimmer**
   a. Wie verhält sich die PWM Steuerung, wenn die Periodendauer des Timers verändert wird (bspw. von 10ms auf 100ms)?
   b. Warum erscheint uns die LED als gedimmt (bei 10ms Timerperiode), abhängig vom PWM-Verhältnis?
   c. Wie wäre es möglich, mittels PWM Ausgang das Gegenstück zum ADC zu bauen, also einen DAC (Digital-Analog-Converter)?
3. **Fragen zu Teil C, Taster Interrupt und Round-Robin**
   a. Achten Sie beim Drücken des Tasters darauf, wie die LED toggelt. Wie könnte man das Verhalten verbessern (Entprellung)?
   b. Wann wird der Interrupt tatsächlich ausgelöst in Abhängigkeit von Rising bzw. Falling Edge Konfiguration?
   c. Vergleichen Sie die Performance zwischen Round-Robin und Interrupt Implementierung und diskutieren Sie diese (um den ADC Interrupt-Handler langsam zu machen, wählen Sie N nicht als Zweierpotenz).

Optionale Aufgaben:

- Implementieren Sie den Mittelwertfilter nach folgender Näherungsformel, welche Vor- / Nachteile bietet dies?

$$\overline{x}_{i+1} \approx \overline{x}_i + 1/N \cdot (x_{i+1} - \overline{x}_i)$$

- Erstellen Sie ein Assembler Projekt und programmieren Sie eine einfache Schleife direkt in Assembler.
- Erweitern Sie das MSP430 Board um eine Bluetooth Schnittstelle und erschaffen Sie eine Kommunikationsmöglichkeit zwischen Mobiltelefon (bspw. Android Bluetooth API) und Microcontroller bspw. für Heimautomatisierung-Anwendungen.

Anhang:

Datenblätter MSP430 unter:

http://www.ti.com/lit/ug/slau144i/slau144i.pdf

http://www.ti.com/lit/ds/symlink/msp430g2553.pdf

Schaltplan: