

### Hardwaresimulation einer Waage

Zur Simulation der Hardware mit einem AT91M63200-Mikrocontroller setzen wir eine Softwaresimulation ein. Neben den Peripheriebausteinen des Mikrocontrollers wird eine Waage simuliert, die an einen Timerbaustein angeschlossen ist.

### Einführung in die Simulation CAPTURE-Modus

Verschaffen Sie sich einen Überblick über die näher zu betrachtenden Programmmodule `stb_scales.c`, `stb_tc.c` und `drc_scales.c` in dem Videomodul **PS-WA** des Moodle-Kurses. Die Timerbausteine werden zur Messung von Zeiten im CAPTURE-Modus verwendet.

### Einführung in das Prinzip der Waage

Im Mikroprozessorklabor des Fachbereichs Informatik wird eine Waage für Praktika verwendet, die für ein Gewicht zwei Frequenzen liefert. Die Frequenzen stehen in einem Verhältnis in Abhängigkeit vom Gewicht, so dass hieraus das Gewicht **m** mit der folgenden Formel berechnet werden kann (die zweite Gleichung für Periodendauer  $T=1/f$ ):

$$m = C1 * ((f_{PA7} / f_{PA4}) - 1) - C2$$

oder

$$m = C1 * ((T_{PA4} / T_{PA7}) - 1) - C2$$

Das gemessene Gewicht **m** lässt sich aus Frequenzen  $f_{PA4}$  und  $f_{PA7}$  berechnen.  $C1=2000$  und  $C2=0$  sind dabei Konstanten, die abhängig von der Waage sind. Das Messprinzip entsteht dadurch, dass die Mechanik der Waage die Schwingung von zwei Saiten verändert. Eine Saite wird kürzer, während die zweite Saite länger wird. Entsprechend verändert sich die Schwingung einer Saite um eine Frequenz  $\$f$ , so dass  $f_{PA7} = f + \$f$  und  $f_{PA4} = f - \$f$ . Die Schwingung **f** der Laborwaage liegt bei 0 Gramm ca. bei 16 KHz. Die an den Timerbaustein angelegten Signale haben durch einen Vorteiler 32 eine Frequenz  $f = 16\text{KHz}/32 = 500\text{Hz}$ .

Die Waage ist darauf ausgelegt, dass maximal 1000 Gramm gewogen werden können. Beim Maximalgewicht lässt sich als Frequenzverhältnis folglich  $f_{PA7}/f_{PA4} = 1,5$  oder abgeleitet von der Frequenz  $f=500\text{Hz}$  ein  $\$f = 100\text{Hz}$  messen.

### Der Simulator

Die zip-Datei zu diesem Termin enthält Änderungen in einigen Modulen u.a. in der `sim_config.h`. Ebenso gibt es eine neue Projektdatei für den Termin 4 und 5 mit der Datei `usr_main4.c`. Wenn Sie das Verzeichnis ESSim und seine Inhalte vom letzten Termin sichern wollen, benennen Sie es um. Ansonsten löschen Sie das alte Verzeichnis und extrahieren das Verzeichnis neu. Vermeiden Sie eine Mixtur aus Dateiinhalten unterschiedlicher Termine. Falls Sie nochmal die Originaldateien für die Aufgaben des Termins 3 benötigen, sind diese im Ordner Lab3 zu finden.

## **1. Waagenprinzip und Messgenauigkeit der Hardware**

Berechnen Sie mit dem Systemtakt  $MCKI=25$  MHz des Boards und einem mit dem Prescaler 8 (`TC_CLKS_MCK8`) konfigurierten Timerbaustein Antworten zu folgenden Fragen:

- 1A)** Welche maximale Messdauer  $T_{MAX}$  ergibt sich aus dem gewählten Prescaler?
- 1B)** Welche maximale Periodendauer  $T_{fm}$  ergibt sich bei der Messung des Maximalgewichts für die langsamere Frequenz?
- 1C)** Welche maximale Messdauer  $T_M$  benötigen Sie für den ungünstigsten Fall einer Messung, die Sie zu einem beliebigen Zeitpunkt gestartet haben?
- 1D)** Ist bei dem gewählten Prescaler die Bedingung der maximalen Messdauer eingehalten, so dass es keinen Zählerüberlauf gibt?
- 1E)** Welche Differenz des Zählerstands der Register B-A ergibt sich bei dem Gewicht  $m=0g$  für beide Frequenzen?
- 1F)** Welche Differenz des Zählerstands der Register B-A ergibt sich bei dem Gewicht  $m=1000g$  für die niedrigere der beiden Frequenzen?
- 1G)** Wieviel unterschiedliche Gewichtswerte lassen sich zwischen 0 und 1000g unterscheiden?
- 1H)** Welche Messgenauigkeit ergibt sich durch die Anzahl unterscheidbarer Werte als theoretische Messgenauigkeit der Hardware?

## **2. Konfigurierbarer Prescaler für Timebausteine**

Der Timerbaustein des AT91M63200-Mikrocontrollers bietet die Möglichkeit, den internen Zähler mit unterschiedlicher Geschwindigkeit durch den ausgewählten Prescaler zählen zu lassen.

Machen Sie sich mit der Systemfunktion `ioctl()` in POSIX vertraut (Hinweis: die Simulation verwendet nur eine vereinfachte Version als Beispiel). Erweitern Sie den Waagentreiber und seine `ioctl()`-Funktion, so dass der Prescaler einstellbar wird.

**2A)** Hierfür ist die Funktion `scales_set_prescaler()` in dem Waagentreiber in der Datei `drv_scales.c` bereits vorgesehen. Implementieren Sie diese Funktion, so dass der jeweils ideale Prescaler für die Messung einer gegebenen Frequenz (als Parameter) in den Timerbausteinen gesetzt wird. Beachten Sie, dass der Timerbaustein mit einem Prescaler von 2, 8, 32, 128 und 1024 konfiguriert werden kann. Den Implementierungsansatz können Sie frei wählen.

Testen Sie mit dem Debugger und verschiedenen Frequenzen, ob ihr Programm korrekt implementiert ist.

**2B)** Welcher ideale Prescaler ergibt sich für die langsamste Frequenz bei Messungen mit der Waage aus Aufgabe 1?

**2C)** Welche Messgenauigkeit in Gramm ergibt sich nun, wenn Sie die Berechnung (in Schritten wie in den Aufgaben 1E..1H) ausführen?

### **3. Simulation von Messungen**

Der Simulator enthält die Implementierung einer simulierten Waage in der Datei `stb_scales.c`. Ein in einer Datenreihe gegebenes Gewicht wird in die Periodendauern der zwei Frequenzen umgerechnet und die Register A und B der simulierten Timerbausteine bei den Triggerflanken mit den Zählerständen gesetzt. Machen Sie sich mit der Definition von simulierten Datenserien in der Quelltextdatei vertraut. Hiermit können Sie Messreihen simulieren, die immer wieder periodisch wiederholt werden. Der Simulator berechnet dann aus den gemessenen Zählerständen wiederum entsprechend das Gewicht.

Erzeugen Sie ihre eigene Messdatenreihe. Lassen Sie die Simulation laufen und geben Sie die Gewichte aus. Probieren Sie die Messung mit unterschiedlichen Prescalern aus. Probieren Sie unterschiedliche Zeitdauern aus, wie lange ein Datenpunkt einer Datenreihe als Gewicht simuliert wird. Variieren Sie die Gewichte pro Messpunkt so, so dass es innerhalb sehr kurzer Zeit sehr starke Schwankungen gibt.

Beschreiben Sie ihre Beobachtungen. Welche Verbesserungen könnten Sie für stark schwankende Gewichtswerte als Messwertverarbeitung vornehmen?

### **4. Optional: Zusatzfragen zur Implementierung der Simulation**

**4A)** Welche kleinste Zeiteinheit verwendet die Simulation der Waage bei Berechnungen? Welche Zeitbasis ist in der restlichen Simulation verwendet?

**4B)** Warum verwendet die Simulation unterschiedliche Zeiteinheiten in den Modulen?

**4C)** Wie ist der Übergang von der simulierten Waage in den simulierten Timerbaustein gelöst, wenn eine Triggerflanke aufgetreten ist und der Zählerwert in den Registern A oder B des Timerbausteins gespeichert werden muss?