



**h\_da**

HOCHSCHULE DARMSTADT  
UNIVERSITY OF APPLIED SCIENCES

**fbi**  
FACHBEREICH INFORMATIK

MIKROPROZESSORPRAKTIKUM

WS2020

**Termin 4**

C-Programmierung für eingebettete Systeme Pointer, Peripherie, PIO,  
Timer (CAPTURE-Mode)

Name, Vorname	Matrikelnummer	Anmerkungen
Datum	Raster (z.B. Mi3x)	Testat/Datum

Legende: V: Vorbereitung, D: Durchführung, P: Protokoll/Dokumentation, T: Testat

## Lernziele:

Sie sollen heute lernen wie mit Zählern im Capture-Mode Zeitmessungen realisiert werden können.

## Arbeitsverzeichnis:

Kopieren Sie sich aus dem Ordner „./mnt/Originale“ auf dem Laborarbeitsplatzrechner oder aus dem Ordner „sftp://stxxyyy@userv-shell.fbi.h-da.de/home/groups/LabDisk/MI/“ den Ordner mpsWS2020. Dort finden Sie zu jedem Termin vorgegebene Dateien. Ein Schaltplan der Anbindung von Waage und Pumpe finden sie in der Datei **interface.pdf**.

## Weitere Infos:

Ab dem Sommersemester 2011 stehen an jedem Laborarbeitsplatz WaSim (Waagensimulatoren) zur Verfügung. Die WaSim liefern zwei, entsprechend des zu simulierenden Gewichtes, sich verändernde Frequenzen im Bereich von ca. 500Hz. Diese beiden Signale sind direkt auf die Timereingänge (über PA7 und PA4) der Evaluierungsplatine AT91EB63 gegeben.

### Aufgabe 1:

Machen Sie sich mit dem Programm aus **Termin4Aufgabe1.c** vertraut. Übersetzen und testen Sie das Programm. Beseitigen Sie evtl. vorhandene Fehler.

Welche Werte [Zaehlwerte, Zeit] für die zu ermittelnde Periodendauer bei 500Hz erwarten Sie?

..

Welche Frequenzen können wir mit der gewählten Initialisierung erfassen?

..

Ändert sich die Periodendauer bei Belastung der Waage? (Wird gemeinsam im Praktikumstermin ausprobiert.)

..

**ACHTUNG: Die Waage** (gilt nicht für den Simulator) **darf nur mit max. 1000g (1kg) belastet werden!**



### Aufgabe 2:

Erweitern Sie das Programm so, damit auch die Periodendauer  $T_{PA7}$  erfasst werden kann.

Im Praktikum testen wir, ob die zu messenden Werte der Periodendauer bei 500 Hz sich im zu erwartenden Bereich bewegen.

Dokumentieren Sie, welche Frequenz mit zunehmendem Gewicht kleiner und welche Frequenz mit zunehmendem Gewicht größer wird.

### Aufgabe 3:

Aus dem Verhältnis (gilt für  $T_{PA4} > T_{PA7}$ ) der beiden Frequenzen  $f_{PA7}$ ,  $f_{PA4}$  kann die Masse  $m$  nach der Gleichung

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

oder

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

errechnet werden. Die Schwingfrequenzen  $f_{PA7}$ ,  $f_{PA4}$  der Saiten liegen bei etwa 16kHz. Diese Frequenzen werden durch einen Teiler durch 32 auf ca. 500Hz reduziert. Die beiden Größen C1 und C2 sind wägezellenspezifische Konstanten.

**ACHTUNG: Die Waage darf nur mit max. 1000g belastet werden!**

Berechnen Sie die Masse  $m$  aus den zu erfassenden/messenden und gegebenen Werten. Setzen Sie für C1 und für C2 die für die Waage angegebenen Werte ein.

Im Praktikum überprüfen wir anhand eines eingestellten Gewichtes ob Ihre Messwerte für die beiden Periodendauern stimmen.

#### Aufgabe 4:

Schreiben Sie für die Messung der Masse eine Funktion

*int MessungderMasse(void)*

welche die ermittelte Masse als integer Wert in g (Gramm) liefert.

Denken Sie auch diesmal daran, dass Programmteile zur Lösung des Gesamtprojektes (Termin6) am Ende wieder benötigt werden.

#### Aufgabe 5:

Erstellen Sie zu diesem Termin ein Protokoll mit den Lösungen zu den Aufgaben und Ihren Erkenntnissen. Das Protokoll sollen Sie zum nächsten Termin vorlegen können. Teile Ihrer Dokumentation können sicher Bestandteil der Dokumentation des am Ende abzugebenden Projektes sein.

Einige Fragen deren Beantwortung zum Verständnis des Gesamtsystems beitragen:

Wie lange kann es max. dauern, bis eine Messung der Masse abgeschlossen ist?

..

In welcher Reihenfolge sollten die Messungen durchgeführt werden, um möglichst effizient zu sein?

..

Könnten Sie die Aufgabe auch ohne den Einsatz der Bibliothek „libgcc.a“ lösen?

..

Was wäre es für ein Aufwand?

..

Was müssten Sie tun, um das Gewicht in Gramm ohne Floatberechnungen zu erhalten?

..

Könnten Sie auch Milligramm [mg] genau wiegen?

..

Bekommen wir im Simulator konstant stabile Werte?

..

// Loesung zur Aufgabe Termin 4

// Aufgabe 1

//\*\*\*\*\*

// Messen der Periodendauer einer angelegten Frequenz

// von: Manfred Pester

// vom: 06. August 2003

// Achtung: **Programm ist noch nicht ausgereift!**

#include "../h/pio.h"

#include "../h/tc.h"

#include "../h/pmc.h"

// für die Initialisierung des Zähler TC4

#define TC4\_INIT TC\_CLKS\_MCK2 | TC\_LDBSTOP | TC\_CAPT | TC\_LDRA\_RISING\_EDGE | TC\_LDRB\_RISING\_EDGE

int main(void)

{

volatile int captureRA1;  
volatile int captureRB1;  
volatile int capturediff1;  
volatile float Periodendauer1;

StructPMC\* pmcbase = PMC\_BASE;  
StructPIO\* piobaseA = PIOA\_BASE;  
StructTC\* tcbase4 = TCB4\_BASE;  
StructTC\* tcbase5 = TCB5\_BASE;

pmcbase->PMC\_PCER = 0x06C00; // Clock PIOA, PIOB, Timer5 und Timer4 einschalten

// Periodendauer der Waagensignale messen

// Signal an TIOA4 ca. 16kHz entspricht ca. einer Periodendauer von 62,5us

// durch den in der Hardware vorgeschalteten Teiler von 32 ergeben sich ca. 2ms

// Zähler starten und von positiver zu positiver Flanke messen

piobaseA->PIO\_PDR = 0x090;  
tcbase4->TC\_CCR = TC\_CLKDIS;  
tcbase4->TC\_CMR = TC4\_INIT;  
tcbase4->TC\_CCR = TC\_CLKEN;

while(1) //

{

tcbase4->TC\_SR; // Stati durch lesen des Statusregister zurueck setzen  
tcbase4->TC\_CCR = TC\_SWTRG; // Timer fuer aktuelle Messung starten  
while (! (tcbase4->TC\_SR & 0x40)); // Capture Register B wurde geladen -> Messung abgeschlossen  
captureRA1 = tcbase4->TC\_RA; // Capture Register A auslesen  
captureRB1 = tcbase4->TC\_RB; // Capture Register B auslesen  
capturediff1 = captureRB1 - captureRA1; // Ergebnis entspricht der Periodendauer  
Periodendauer1 = capturediff1 / 12.5; // Normierung der Zeit in us

}

return 0;

}