

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



**h\_da**

HOCHSCHULE DARMSTADT  
UNIVERSITY OF APPLIED SCIENCES

**fbi**  
FACHBEREICH INFORMATIK

MIKROPROZESSORPRAKTIKUM

SS2014

Termin 4

C-Programmierung für eingebettete SystemePointer, 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 das Verzeichnis mpsSS2014. 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 können direkt auf die Timereingänge (PA7 und PA4) der Evaluierungsplatine AT91EB63 gegeben werden.

### Aufgabe 1:

Machen Sie sich mit dem Programm aus *Termin4-Aufgabe1.c* vertraut. Übersetzen und testen Sie das Programm. Wie groß ist die gemessene Periodendauer  $T_{PA4}$ ? Verändert sich die Periodendauer bei Belastung der Waage?

**ACHTUNG: Die Waage darf nur mit max. 1000g (1kg) belastet werden!**

### Aufgabe 2:

Erweitern Sie das Programm so, damit auch die Periodendauer  $T_{PA7}$  erfaßt wird. Testen Sie auch hier ob sich die Periodendauer um die 2ms bewegt.

### Aufgabe 3:

Aus dem Verhältnis 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  $16\text{kHz} / 32 = 500\text{Hz}$ . Die beiden Größen  $C1$  und  $C2$  sind wägezellspezifische Konstanten.

Die Frequenz an PA4 wird mit zunehmendem Gewicht kleiner. Die Frequenz an PA7 wird mit zunehmendem Gewicht größer.

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

Berechnen Sie nun noch die Masse  $m$  welche auf der Waage liegt. Setzen Sie für  $C1$  und für  $C2$  die für die Waage angegebenen Werte ein. Überprüfen Sie anhand der gegebenen Gewichte ob Ihre Messwerte 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 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 vorgelegen können.

```
// Lösung zur Aufgabe Termin 4
// Aufgabe 1
//*****
// Messen der Periodendauer einer angelegten Frequenz
// von: Manfred Pester
// vom: 06. August 2003
// Achtung: Programm hat noch Fehler

#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 = 0x06f80;    // Clock PIOA, PIOB, Timer5, Timer4, Timer1 einschalten

    // Periodendauer der Waagensignale messen
    // Signal an TIOA4 ca. 16kHz entspricht ca. einer Periodendauer von 62,5us
    // durch den Teiler von 32 ergeben sich ca. 2ms
    // Zähler mit positiver Flanke starten

    piobaseA->PIO_PDR = 0x090;
    tcbase4->TC_CCR = TC_CLKDIS;
    tcbase4->TC_CMR = TC4_INIT;
    tcbase4->TC_CCR = TC_CLKEN;
    tcbase4->TC_CCR = TC_SWTRG;

    while(!!(piobaseB->PIO_PDSR & KEY3))
    {
        tcbase4->TC_CCR = TC_SWTRG;
        while (!!( tcbase4->TC_SR & 0x40)); // Capture Register B wurde geladen Messung abgeschlossen
            captureRA1 = tcbase4->TC_RA;    //
            captureRB1 = tcbase4->TC_RB;
            capturediff1 = abs(captureRB1) - abs(captureRA1);
            Periodendauer1 = abs(capturediff1) / 12.5;    // Zeit in us
    }

    return 0;
}
```