



Aufgabenblatt 7 Termine: 03.06. / 06.06

| | |
|---------|-------------------|
| Gruppe | |
| Name(n) | Matrikelnummer(n) |
| | |

Aufgabe 7.1 (Punkte 40)

Das Ziel dieser Aufgabe ist das Kennenlernen und Verwenden von Timern. Konsultieren Sie das Datenblatt des auf dem Arduino Due verwendeten ARM-Mikrocontrollers (SAM3X8E) unter tams.informatik.uni-hamburg.de/lectures/2014ss/vorlesung/es/doc/sam3x8e_datasheet.pdf. Schauen Sie sich insbesondere das Kapitel 37 (ab Seite 869) an.

Um (unter anderem) die Konfiguration und Verwendung der Timer zu vereinfachen enthält das Arduino Framework eine von Atmel für die SAM-Mikrocontroller zur Verfügung gestellte Funktionsbibliothek: **libsam**. Der Quellcode dieser Bibliothek kann unter github.com/arduino/Arduino/tree/ide-1.5.x/hardware/arduino/sam/system/libsam eingesehen werden. Beachten Sie insbesondere die Dateien `tc.h`/`tc.c`, die entsprechende Funktionen für die Konfiguration und Verwendung der Timer enthalten.

Der Mikrocontroller des Arduino Due besitzt **3 frei nutzbare Timer-Blöcke** (TC0, TC1 und TC2) mit jeweils **3 separaten Kanälen** (0 bis 2) und **32-Bit Zählern**. Somit stehen Ihnen - wie in der folgenden Tabelle dargestellt - 9 Timer zur Verfügung.

| TC | Ch | NVIC "irq" | IRQ handler | PMC id |
|-----|----|------------|-------------|--------|
| TC0 | 0 | TC0_IRQn | TC0_Handler | ID_TC0 |
| TC0 | 1 | TC1_IRQn | TC1_Handler | ID_TC1 |
| TC0 | 2 | TC2_IRQn | TC2_Handler | ID_TC2 |
| TC1 | 0 | TC3_IRQn | TC3_Handler | ID_TC3 |
| TC1 | 1 | TC4_IRQn | TC4_Handler | ID_TC4 |
| TC1 | 2 | TC5_IRQn | TC5_Handler | ID_TC5 |
| TC2 | 0 | TC6_IRQn | TC6_Handler | ID_TC6 |
| TC2 | 1 | TC7_IRQn | TC7_Handler | ID_TC7 |
| TC2 | 2 | TC8_IRQn | TC8_Handler | ID_TC8 |

Verwenden Sie bitte in Ihrer `setup()`-Routine folgenden Basis-Code für die Konfiguration des Timers:

```
pmc_set_writeprotect(false);  
pmc_enable_periph_clk( >> PMC id des Timers << );
```

```
>> Hier erfolgt die Konfiguration des Timers

NVIC_ClearPendingIRQ( >> NVIC irq des Timers << );
NVIC_EnableIRQ( >> NVIC irq des Timers << );

>> Hier wird der konfigurierte Timer gestartet
```

Hinweise zur Konfiguration:

- Konfigurieren Sie den Timer im `waveform mode` (auch wenn das erzeugte Rechtecksignal nicht verwendet werden wird).
- Der Timer soll den Zähler hochzählen und bei positiver Übereinstimmung mit dem Vergleichswert diesen automatisch zurücksetzen können (`automatic RC trigger`).
- Setzen Sie den entsprechenden Vergleichswert um die geforderte Event-Frequenz von 1Hz zu erhalten.
- Aktivieren Sie für den gewählten Timer das `RC-compare-Interrupt`, deaktivieren Sie die restlichen Interrupt-Typen.

Entwerfen Sie ein Programm, das einen der verfügbaren Timer verwendet um eine Event-Behandlung im 1 Sekunden-Takt auszulösen. Implementieren Sie in der Event-Behandlungsroutine entsprechende Funktionalität zur Aktualisierung einer digitalen Uhr-Anzeige (Stunden/Minuten/Sekunden) im LC-Display.

Aufgabe 7.2 (Punkte Bonus)

Implementieren Sie einen Befehlssatz, der Ihnen das Setzen/Modifizieren der angezeigten Uhrzeit über die Eingabezeile des seriellen Monitors ermöglicht.

Aufgabe 7.3 (Punkte 40)

Für die Lösung dieser Aufgabe benötigen Sie ein Wireless SD Shield zuzüglich einer vorbereiteten microSD Speicherkarte. Da die microSD Speicherkarte ebenfalls über **SPI** kommuniziert und das Shield bei direkter Verbindung mit dem Arduino Due Board den **SPI**-Block unzugänglich macht, ist folgendes bei der Verdrahtung zu beachten:

- Verbinden Sie `VCC 3.3V`, `VCC 5V` und `GND` mit den entsprechenden Pins des Wireless SD Shields.
- Verbinden Sie die `MISO`, `MOSI` und `SCK` Pins über das Steckbrett mit den entsprechenden Pins des **SPI**-Blocks.
- Das Wireless SD Shield sieht **Pin 4** exklusiv als `Slave-Select Pin` vor (Aufdruck `SD_CS`). Verbinden Sie diesen Pin mit einem noch freien `Slave-Select Pin` des Arduino Due (4, 10 oder 52). **Wichtig:** Übergeben Sie die von Ihnen gewählte Pin-Nummer als Argument an `SD.begin(pin)`.

Ziel dieser Aufgabe ist primär das Lesen von Dateien von der microSD Speicherkarte. Machen Sie sich zunächst mit dem Funktionsumfang der Arduino SD Bibliothek (-> `#include`

<SD.h>) vertraut arduino.cc/en/Reference/SD. Schauen Sie sich insbesondere die folgenden Funktionen an:

| | |
|--------------------|--|
| * SD.begin() | arduino.cc/en/Reference/SDbegin |
| * SD.exists() | arduino.cc/en/Reference/SDexists |
| * SD.open() | arduino.cc/en/Reference/SDopen |
| * File.available() | arduino.cc/en/Reference/FileAvailable |
| * File.read() | arduino.cc/en/Reference/FileRead |
| * File.close() | arduino.cc/en/Reference/FileClose |

Auf der Ihnen ausgehändigten microSD Speicherkarte befinden sich auf der obersten Ebene des Dateisystems Textdateien mit der Endung `.txt`:

- `text1.txt`
- `text2.txt`

sowie Bilddateien mit der Endung `.img`:

- `tams.img`
- `smile1.img`
- `smile2.img`
- `smile3.img`

Entwerfen Sie einen kleinen Befehlssatz für die Eingabezeile des seriellen Monitors, der Ihnen ermöglicht 1. die Existenz einer Datei anzufragen und 2. die existierende Datei auf dem LC-Display anzeigen zu lassen. Beachten Sie bitte folgende Hinweise bezüglich der Dateikodierung:

- Die Textdateien enthalten nur eine Textzeile, abgeschlossen durch ein newline-Zeichen (`\n`). Beispiel: `This is some text.\n`
- Die Bilddateien enthalten zwei Zeilen ASCII-kodierter Daten. Auch hier sind beide durch ein newline-Zeichen (`\n`) abgeschlossen.
- Die erste Zeile enthält die Dimensionen des Bildes:
 - Beispiel: `10,10\n`
- Die zweite Zeile enthält die Definition des jeweiligen Pixels (**zeilenweise angeordnet**) durch eine `0` oder eine `1`:
 - Beispiel: `0,0,0,0,0,1,1,1,1,1, ... ,1,1,1,1,1\n`

Weitere Hinweise:

- Bitte positionieren Sie die Bilder nach Möglichkeit **horizontal und vertikal zentriert** in der Ausgabe des LC-Displays.
- Verwenden Sie den aus Blatt 6 bekannten ASCII-Datensatz zur Darstellung des Inhalts der Textdateien. Sorgen Sie für den entsprechenden Umbruch des Textes.
- Sollte eine Textdatei mehr Zeichen haben als auf dem LC-Display im 6x8-Format dargestellt werden können, so sollten Sie dieses entsprechend auf dem LC-Display mitteilen (Beispiel: `Too many characters`).

- *Optional* können Sie auch ein fortwährendes Scrollen implementieren, um den Text vollständig anzeigen zu können.