

## AUFGABE 1: HALLO WELT

Diese Aufgabe dient dem initialen Einrichten Ihrer Entwicklungsumgebung und dem ersten Kontakt mit dem Echtzeitbetriebssystem *eCos* und dem Triboard. Ziel ist es einen ersten Einblick in die Möglichkeiten der Entwicklungsumgebung zu erhalten.

*Fragestellungen, die für die Abgabe der Aufgabe von besonderem Interesse sind, setzen wir in dieser und den folgenden Übungsaufgaben kursiv.*

### *Aufgabenstellung*

1. *Vorbereitung*: Kopieren und entpacken Sie die Vorgabe in ein beliebiges Arbeitsverzeichnis und setzen Sie die nötigen Umgebungsvariablen durch den Aufruf von `source ecosenv.sh`. Sie können den Inhalt der Datei auch in Ihre `~/ .bashrc` einfügen, um den Vorgang zu automatisieren. Nun können Sie die Makefiles generieren und die noch funktionslose Anwendung erstmals kompilieren. Sie können die erzeugte Anwendung mit Hilfe des Lauterbach-Debuggers in den SRAM des Triboards laden und starten. Eine doxygen Dokumentation kann man mittels `make doc` erzeugen.

```
❯ source ecosenv.sh
❯ cmake ..
❯ make
❯ make flash
```

2. *Fadensystem*: Die Erzeugung eines geeigneten Threadsystems ist Bestandteil aller folgenden Übungsaufgaben.

Implementieren Sie in `hello.c` die `cyg_user_start()`-Funktion, in der Sie einen Thread mit Hilfe von `cyg_thread_create()` erzeugen. Die Threadfunktion soll zunächst periodisch, jede Sekunde, mittels `diag_printf()` die Zeichenkette „Hallo Welt!\n“ auf der seriellen Schnittstelle ausgeben. Sobald Sie den RS232-Ausgang des Triboards mit der seriellen Schnittstelle Ihres Rechners verbunden haben, können Sie sich die serielle Ausgabe folgendermassen anzeigen lassen: `screen /dev/ttyS0 115200`. Um einen periodischen Thread zu simulieren, bietet sich die Funktion `cyg_thread_delay()` an. Bei dem zu übergebenden Wert handelt es sich um Millisekunden. Die kleinstmögliche Verzögerung beträgt 2 ms

*Welchen Nachteil hat es, periodische Ausführung auf diese Art umzusetzen? Haben Sie eine Idee für einen sinnvolleren Ansatz?*

3. *Debugging*: Für das Echtzeitverhalten ist neben der eigenen Implementierung immer das Gesamtsystem zu betrachten. Um ein Gefühl hierfür zu bekommen, setzen Sie im Lauterbach-Debugger einen Breakpoint auf die Funktion `cyg_hal_plf_serial_putc()` und setzen Sie die Ausführung fort.

Sobald die Ausführung an dem gesetzten Breakpoint gestoppt hat, können Sie die Aufrufhierarchie über den Befehl `backtrace` ausgeben. *Erkunden Sie nun mit Hilfe des Debuggers den Aufrufgraphen der `diag_printf()`-Funktion. Welche von Ihnen benutzte Funktion ist für die Ausführung von `cyg_hal_plf_serial_putc()` verantwortlich? Lassen Sie sich den Quellcode von `.._putc()` im Debugger anzeigen. Welche Zweck erfüllt sie? Welche aus Echtzeitsicht kritischen Stellen fallen Ihnen hier bei der Interaktion mit der Hardware besonders auf?*

4. *Signalerzeugung*: Mittels des periodischen Threads kann nun ein erstes Signal erzeugt werden. Sie können den GPA des Tricore-Prozessors mit Hilfe der Funktion `ezs_dac_write()` als Digital-Analog-Wandler verwenden (Wertebereich 0 bis  $2^8 - 1$ ) und sich das erzeugte Signal am Oszilloskop anschauen. Verbinden Sie hierzu die Masseklemme des Tastkopfes mit dem Massepin der Filterplatine. Nun können sie mithilfe der Klemmprüfspitze das ungefilterte und das gefilterte Signal an den entsprechenden Pins ansehen. Erzeugen Sie ein Sinussignal und verwenden Sie für diesen Zweck die Funktion `sinf()`. Verfolgen Sie hierbei einen systematischen Ansatz bei der Erzeugung des Signals und halten Sie Ihre Implementierung frei parametrierbar.<sup>1</sup> *Verringern Sie hierfür die Periode des Fadens drastisch (z.B. 2 ms). Wie müssen Sie Abtastrate und Parameter des Sinus wählen um ein Signal einer bestimmten Frequenz zu erzeugen? Was passiert, wenn Sie die Verzögerung des Threads verändern? Betrachten Sie abschließen die Funktion `ezs_dac_write()` im Debugger. Wie unterscheidet sie sich von `cyg_hal_plf_serial_putc()` im Bezug auf das Echtzeitverhalten?*

\* math.h

## Hinweise

- Bearbeitung: Gruppe mit je drei Teilnehmern.
- Abgabezeit: 20.–24.10.2014
- Fragen bitte an [i4ezs@lists.cs.fau.de](mailto:i4ezs@lists.cs.fau.de)

<sup>1</sup>[https://de.wikipedia.org/wiki/Schwingung#Harmonische\\_Schwingung](https://de.wikipedia.org/wiki/Schwingung#Harmonische_Schwingung)