Echtzeitbetriebssysteme
Frühlingssemester 2017
Roger Weber
Klassen E3a, E3b, E4p

Übung 1

Grundlagen Echtzeitbetriebssysteme

Aufgabe 1.1 Zyklische Betriebssysteme

Im Unterricht wurde der Ansatz für ein zyklisches Betriebssysteme mit Hilfe einer Tabelle erklärt. In dieser Aufgabe können Sie dieses Thema vertiefen, indem Sie den Code einer bestehenden Applikation nachvollziehen.

Gehen Sie wie folgt vor:

- Importieren Sie das Projekt EZBSY_U1A1 (auf Moodle) in die Attolic-IDE.
- Arbeiten Sie sich in den Demo-Code ein und überprüfen Sie insbesondere:
  - wie die Struktur der Tabelle definiert ist,
  - wie die verschiedenen Applikations-Funktionen in der Tabelle verwaltet werden.
- Debuggen Sie die Applikation.
- Verändern Sie den periodischen Aufruf von Applikations-Funktionen und beobachten Sie die Auswirkungen.
- Optional: Ändern Sie die einzelnen Applikations-Funktionen.

Bemerkungen: Informationen zu den BSP-Funktionen finden Sie auf der CARME-Webseite: carme.bfh.ch

Aufgabe 1.2 Task Starvation

In dieser Aufgabe werden die Auswirkungen der “Task Starvation“ behandelt. Sie können basierend auf einer bestehenden Applikation durch modifizieren von Task-Prioritäten eine “Task-Starvation“ provozieren und nachvollziehen.

Gehen Sie wie folgt vor:

- Importieren Sie das Projekt EZBSY_U1A2 (auf Moodle) in die Attolic-IDE.
- Arbeiten Sie sich in den Demo-Code ein und überprüfen Sie insbesondere:
  - wie die beiden Tasks erzeugt werden (Prioritäten, Stackgrösse, ...),
  - welche Aufgaben die beiden Tasks ausführen,
  - die Wirkung des Compiler-Switches “USE_vWAIT“.
- Debuggen Sie die Applikation mit folgenden Einstellungen:
  a) “USE_vWAIT“ aktiviert, beide Task-Prioritäten 1. Was stellen Sie fest, wie können Sie das begründen?
  b) “USE_vWAIT“ aktiviert, Task zwei mit Priorität 2. Was stellen Sie fest, was ist der Unterschied zu a)?
  c) “USE_vWAIT“ deaktiviert, d.h. vTaskDelay() ist aktiv, Tasks mit Prioritäten wie b). Was stellen Sie fest, was ist der Unterschied zu b)?

Aufgabe 1.3 LED Lauflicht

Auf dem CARME-M4 Kit soll mit den LEDs ein Lauflicht realisiert werden, dessen Geschwindigkeit mit dem Potentiometer eingestellt werden kann. Dazu werden zwei Tasks programmiert:

- Task1 liest den Wert des Potentiometers alle 100ms ein. Den Wert des Potentiometers können Sie mit der Funktion CARME_IO2_ADC_Get() einlesen. Der eingelesene Wert wird in einer globalen Variablen “u16LEDSpeed“ gespeichert.
- Task2 ist für die Ansteuerung der LEDs verantwortlich. Die Geschwindigkeit liest der Task aus der globalen Variablen “u16LEDSpeed“. “u16LEDSpeed“ soll Werte zwischen 100 und 999 annehmen und einer Verzögerung in Millisekunden entsprechen, bevor die nächste LED angesteuert wird.

Verwenden Sie als Ausgangslage für diese und alle folgenden Aufgaben den Source-Code, welchen die Attolic-IDE beim Anlegen eines neuen Projektes generiert. Gehen Sie für das Erzeugen eines neuen Projektes wie folgt vor:

- Menü File > New > C Project
- Dialogbox “ C Project“: Geben Sie einen Projektnamen ein (z.B. EZBSY_U1A3) und wählen Sie unter “Executable“ den Eintrag “Embedded C Project“.
- Next
- Dialogbox “Hardware configuration“: Wählen Sie für “Vendor“ “BFH“ und für “Evaluation board“ “CARME-M4 with FreeRTOS“.
- Next
- Dialogbox “Software configuration“: Hier brauchen Sie keine Einstellung zu machen, somit Next wählen.
- Dialogbox “Debugger configuration“: Ebenfalls keine Änderungen, somit Finish wählen.

Aufgabe 1.4 Ausgabe der Schalter und Taster auf dem LCD

Auf dem LCD sollen die Zustände der Schalter und Taster ausgegeben werden. Dazu sind folgende Tasks zu programmieren:

- Task1 liest periodisch alle 100ms (100 Ticks) die Zustände der Schalter ein und speichert diese in der globalen Variablen “u8SwitchState“. Um eine konstante periodische Ausführung zu erhalten, stellt freeRTOS die Funktion “vTaskDelayUntil()“ zur Verfügung. Angaben zur Funktion “vTaskDelayUntil()“ finden Sie auf der Homepage von freeRTOS www.freertos.org auf der linken Seite unter “FreeRTOS > API Reference > TaskControl“.
- Task2 pollt die Zustände der Taster alle 50ms (50 Ticks) und legt diese in der Variablen “u8ButtonState“ ab.
- Task3 zeigt die Zustände der Schalter und der Taster auf dem LCD an. Um das LCD-Display anzusteuern stehen verschiedene Library-Funktionen zur Verfügung. Informationen finden Sie auf der CARME-Webseite (carme.bfh.ch) unter “Software > Documentation Online > Simple Graphic Library > Online“.

Bemerkung: Die Kommunikation über globale Variablen ist nicht sehr elegant. Wir werden später im Kapitel Message-Queues sehen, wie man dies besser machen kann.