Umstieg von BlackBox auf deep

Martin Züger

22. Februar 2011

Dieses Dokument beschreibt die wichtigsten Unterschiede zwischen den Entwicklungsplattformen BlackBox BB555JOD und deep.

1 Die Entwicklungsumgebung

Als IDE wird anstelle von BlackBox neu Eclipse verwendet. Dadurch werden viele nützliche Funktionen wie z.B. Code-Vervollständigung, Automatische Import-Ergänzung, direktes Anzeigen der JavaDoc, etc. verfügbar. Durch diese Umstellung ändert sich aber auch die Handhabung relativ stark. So werden in Eclipse keine ODC- bzw. OTD-Dateien mehr benötigt (und auch nicht mehr unterstützt). In Eclipse werden deep-Projekte angelegt und in diesen wird festgelegt welches die Rootklassen sind und wie die Targetkonfiguration aussieht. Auf dem Infoportal [1] ist eine Schritt-für-Schritt-Anleitung für ein erstes Projekt zu finden.

2 Das Runtime System

Die meisten spürbaren Änderungen betreffen das Runtime System. Die folgenden Abschnitte geben einen kurzen Überblick über die wichtigsten Änderungen.

2.1 Paket-Struktur

Die Java-Paketstruktur für das Runtime System wurde komplett überarbeitet und verfeinert. Das Basispaket des Runtime Systems für den Freescale MPC555 ist ch.ntb.inf.deep.runtime.mpc555. Es enthält das Paket driver in welchem sämtliche Treiber zu finden sind sowie das das Paket demo in dem Beispiel-Klassen zu finden sind.

[1] http://www.ntb.ch/infoportal/software:eclipse:deep_projekt_anlegen

2.2 Das Tasking-System

Die Klasse ch.ntb.inf.deep.runtime.mpc555.Task ersetzt die Klasse mpc555.Task. Die API dieser Klassen ist jedoch nicht ganz identisch. Die Methode Do() heisst neu action().

2.3 Treiber

Die Treiber wurden ebenfalls angepasst. Die auffälligste Neuerung ist die neue Namensgebung. Die folgende Tabelle zeigt die Änderungen im Detail.

BB555JOD	**deep**
TPU_A	TPUA
TPU_B	TPUB
DIO	TPU_DIO
PWM	TPU_PWM
FQD	TPU_FQD
PPWA	TPU_PPWA
QADC	QADC_AIN
DIO_ADC	QADC_DIO
QSMCM	QSMCM
DIO_PortQS	QSMCM_DIO
SCI1	SCI1
SCI2	SCI2
Mpiosm	MPIOSM_DIO
Mios	MPWMSM_PWM
DIO_Mios	MPWMSM_DIO
RTBoard	RTBoard
Box	RTBox
Robi2	Robi2
StepMotor	StepMotor
DistSensor	HLC1395Pulsed
DS1302Z	DS1302Z
DS1620	DS1620
DAC7614	DAC7614
MAX512	MAX512
TLC549	TLC549
CharLCD	HD44780U
CharLCD2x16	HD44780U

2.4 Textausgaben

Die Klasse mpc555.OutT steht nicht mehr zur Verfügung. Ausgaben können stattdessen wie gewohnt über System.out gemacht werden. Im Unterschied zu einer normalen Java-Anwendung auf dem PC ist es jedoch notwendig festzulegen wo die Ausgaben überhaupt ausgegeben werden sollen. Das folgende Beispiel zeigt, wie die zweite serielle Schnittstelle des MPC555 (SCI2) konfiguriert wird (9600 Bd, 8N1) und System.out anschliessend auf diese Schnittstelle gelegt wird.

Listing 1: Ausgabe über SCI2

```
1  package ch.ntb.inf.deep.runtime.mpc555.demo;
2
3  import java.io.PrintStream;
4  import ch.ntb.inf.deep.runtime.mpc555.Task;
5  import ch.ntb.inf.deep.runtime.mpc555.driver.SCI2;
6
7  public class SystemOutDemoSCI2 extends Task {
8
9          public void action() {
10                 System.out.print('.');
11         }
12
13         static {
14                 // 1) Initialize SCI2 (9600 Bd, 8N1)
15                 SCI2.start(9600, SCI2.NO_PARITY, (short)8);
16
17                 // 2) Use SCI2 for stdout
18                 System.out = new PrintStream(SCI2.out);
19
20                 // 3) Redirect stderr to stdout (optional)
21                 System.err = System.out;
22
23                 // 4) Initialize demo task
24                 Task t = new SCIStreamTest();
25                 t.period = 1000;
26                 Task.install(t);
27         }
28 }
```