

Zeit als neue Währung

Ausführbare Rechenmodelle im heterogenen Aktorframework LabVIEW unterstützen unsere Denkweise und beschleunigen die Entwicklung von Timing in Embedded-Software



Bild 1: Verschiedene Rechenmodelle passen sich flexibel der Aufgabenstellung an und lassen sich in ein heterogenes Framework einbetten, verbinden und auf Embedded-Hardware ausführen. Dank dieses Entwicklungsbeschleunigers schaffen auch kleine Teams in kurzer Zeit große Ergebnisse.

Was Embedded-Programmiersprachen angeht, gibt es unterschiedliche Vorstellungen und Meinungen. Oft konzentrieren sie sich auf den Vergleich zwischen Notationen wie „C“, Datenfluss, Model-Based-Design oder UML. Warum nicht aus diesem Raster ausbrechen? Warum nicht Brücken bauen, die jeweilige Stärke einer bestimmten Notation zum Vorteil nutzen und sie kombinieren? Dann besteht das Potential, die Entwicklungszeit und das Timing zukünftiger smarter Embedded Systeme oder sogar Cyber-Physical-Systems in den Griff zu kriegen.

Der Software-Ingenieur wird so in seiner Denkweise unterstützt und wird dabei produktiver. Das verspricht angesichts zunehmender Relevanz und Wertschöpfung in Embedded-Software Erfolg und Profit.

Zeit und Parallelität ist die Herausforderung Nummer 1

In der physikalischen Welt passiert vieles auf einmal und es sind Echtzeit, schnelle Reaktionszeiten, 24/7-Betrieb und Kommunika-


tion gefragt. Embedded-Systeme treffen als zeitdiskrete, künstliche Recheneinheiten auf zeitkontinuierliche, natürliche Prozesse. Sie sind darüber hinaus heterogene Gebilde aus Mikroprozessoren mit analogen und digitalen Sensoren und Aktoren. Das macht sie von Natur aus komplex und deren Entwicklung fehleranfällig. Smarte Embedded-Systeme oder sogar Cyber-Physical-Systems gehen in drei Aspekten den entscheidenden Schritt weiter und

lassen die virtuelle mit der physikalischen Welt nahtlos verschmelzen. Erstens sind Parallelität und Timing das Maß für formale Korrektheit der Software. Zweitens sind dezentrale Regeltechnik und Synchronisierung stark ausgeprägt. Drittens kommunizieren die Systeme in Echtzeit über ein Netzwerk. Genau dieses Verständnis zu gemeinsamer Dynamik ist es, was diese neue Disziplin vom traditionellen Ansatz fundamental unterscheidet. Zeit und Parallelität sind die wichtigsten zwei Merkmale zukünftiger, smarter Embedded-Systeme.

Neue Methode unterstützt unsere Denkweise

Also muss eine neue Methode her, die unsere Denkweise besser unterstützt. So ließe sich die Komplexität beherrschen und das Projekt bleibt auf Kurs. Das System soll von Beginn an richtig entworfen werden, schon lange vor der Aufteilung in Hardware und Software. Auch Co-Design wird nie dieselbe Beschleunigung erreichen wie ein System, das von Beginn an das tut, was es soll und dies unabhängig von seiner Implementierung, nämlich korrektes Verhalten bezogen auf Funktion, Kommunikation und Zeit. Der Schlüssel dazu sind flexible Rechenmodelle, welche uns ein Verständnis für physikalische, zeitliche Zusammenhänge geben. Sie werden in ein heterogenes Framework eingebettet und auf verschie-

Autor:



Marco Schmid,
Ingenieur Systemtechnik FH

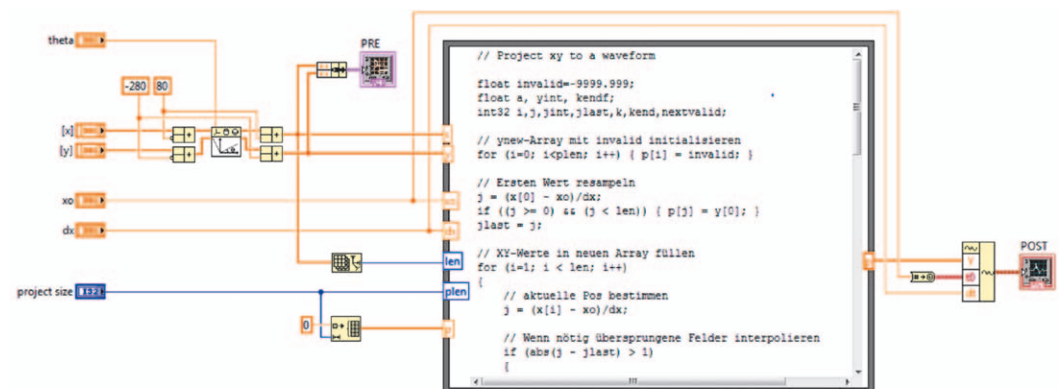


Bild 2: Das textbasierte Rechenmodell (C-Algorithmus) ist optimiert für diskrete, sequentielle Zeitdarstellung und ist hier verknüpft mit dem parallel funktionierenden Datenflussmodell (LabVIEW)

denen Zielplattformen in Echtzeit ausgeführt:

- Diese formalen Modelle beschreiben das Systemverhalten auf hohem Abstraktionsgrad, weitgehend unabhängig von Programmiersprache und unterlegter Hardware. Es handelt sich um Rechenmodelle (engl. Models of Computation). Sie abstrahieren Syntax und Semantik und lenken unseren Fokus auf Parallelität, Zeit, Kommunikation und Synchronisierung.
- Rechenmodelle können beliebig kombiniert werden (Bild 2). Voraussetzung dazu ist ein heterogenes Entwicklungsframework, worin sich die Modelle einbetten und ausführen lassen.
- Dank plattformbasiertem, hardwareunabhängigem Design erhält der Ingenieur die Freiheit, Entscheidungen zu Hardware und Implementierungsdetails hinauszuzögern. Er weiß, dass sich das Framework mit den Modellen auf mehreren Zielsystemen unterschiedlicher Funktions- und Leistungsklassen (Bild 4) ausführen lässt und sich zeitlich und funktional immer korrekt verhält.

Modelle ins heterogene Aktor-Framework einbetten

Die grafische Systemdesign-Umgebung LabVIEW von National Instruments ist ein heterogenes „Gefäß“, in das sich die verschiedenen Rechenmodelle einbetten, verbinden und parallel ausführen lassen (Bild 3). Außerdem bietet sich damit die „richtige“ Abstrak-

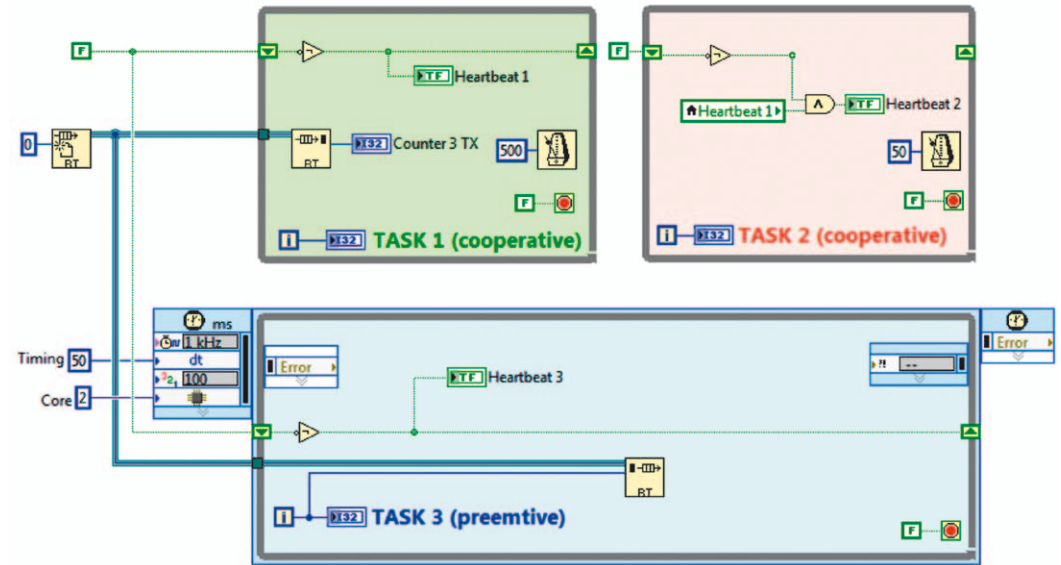


Bild 3: Das Multitasking-Modell abstrahiert die diskrete, parallele Zeitdarstellung und unterstützt synchrone und asynchrone Prozesse. Task 1 (Grün, 500-ms-Takt) und Task 2 (Rot, 50-ms-Takt) arbeiten kooperativ auf dem 1. Rechenkern. Task 3 (Blau, 50-ms-Takt) hingegen läuft preemptiv auf dem 2. Rechenkern

tion zur „richtigen“ Zeit und eignet sich so als Wirt für die verschiedenen Rechenmodelle:

- Plattformbasierter Ansatz ermöglicht gleichzeitig formale Verifikation und Validation auf Modellebene
- Support von heterogenen Rechenmodellen
- Hybrid aus diskretem und kontinuierlichem Zeitmodell
- Parallelität und Kommunikation sind natürlicher Bestandteil der Sprache
- Abstrakt und trotzdem hardwarenah und damit ausführbar auf verschiedenen Hardwarearchitekturen

- Kommerziell verfügbar, weltweite Akzeptanz, verbreitetes Ökosystem

Software mit Hardware zusammenführen

Dieses Framework mit den eingebetteten Rechenmodellen ist ein mächtiges Instrument, dank dem sich der Entwickler in der Embedded-Software auf verschiedenen Abstraktionsniveaus bewegen kann und trotzdem immer die globale Systemsicht vor Augen hat. Über zwei Wege lässt sich dieses Framework in eine Echtzeitapplikation überführen und auf verschiedenen Embedded-Targets ausfüh-

ren (Bild 4). Der erste führt über Echtzeit-Linux, indem die LabVIEW-Anwendung entsprechend dem POSIX-Standard 1:1 auf das Betriebssystem abgebildet und auf einem Dual-Core-ARM9-Prozessor ausgeführt wird. Der zweite funktioniert über einen ANSI-C-Code-Generator. Damit lässt sich LabVIEW auf einen beliebigen Mikrocontroller außerhalb des NI Standards portieren und letztendlich auch kundenspezifische Kompletthardware realisieren.

■ Schmid Elektronik AG
www.schmid-elektronik.ch

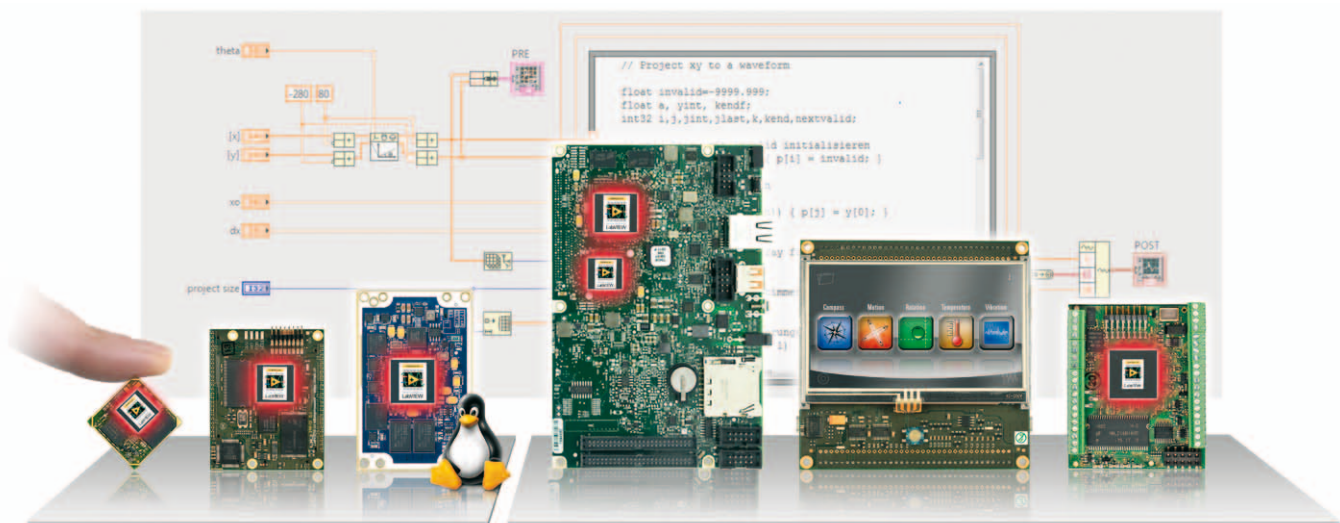


Bild 4: LabVIEW auf eigener Standard-Hardware in unterschiedlichen Formfaktoren und Leistungsklassen. Links: Einsteckmodule als Briefmarken-Coremodul, Scheckkarten-COM oder -SOM. Rechts: Singleboard-Computer als Europakarte, im PC104-Format oder auf der Hutschiene.