

National Instruments über grafische Programmierung im Embedded-Bereich

PLATTFORM- ÜBERGREIFEND

Plant ein Anwendungsentwickler seine Embedded-Applikation auf einer Standalone-Hardware zu realisieren und hat sich wegen hoher Komplexität, engem Terminplan oder knappen Ressourcen für »LabVIEW« als Entwicklungsumgebung entschieden, stehen ihm zwei Plattformen in verschiedenen Formfaktoren und Leistungsklassen mit all ihren Vor- und Nachteilen zur Verfügung. Es gilt die Richtige auszuwählen. Die zwei möglichen Lösungswege sind: die drei standardisierten NI-Plattformen »PXI«, »CompactRIO« und »Single-Board RIO« und die zwei Microcontrollertargets ARM und Analog Devices Blackfin. Wie können Projektanforderungen mit einer von NI LabVIEW unterstützten Hardware gezielt kombiniert werden, ohne diese Entscheidung später revidieren zu müssen? Und falls doch, wie lässt sich die Portierbarkeit von Software so steigern, damit diese ohne massives Umcodieren plattformübergreifend eingesetzt werden kann?

Embedded-Systeme unterscheiden sich fundamental von PC-Applikationen, denn sie sind äußerst restriktiv im Bezug auf Verfügbarkeit, Zuverlässigkeit, Determinismus, Stromverbrauch und Kosten. Dabei ist weniger der Formfaktor der Hard- und Software-Kombination ausschlagge-

bend, sondern es ist die Autonomie, die ein Embedded-System einzigartig macht. Wird die Anwendung gestartet, ist die Software nach einem kurzen Bootprozess sofort betriebsbereit und funktioniert unter allen Umständen fehlerfrei und deterministisch. Weil ein Embedded-System autark funktionieren

muss, kommt selbstständiger Fehlererkennung und -behebung eine zentrale Bedeutung zu. Schlanke Echtzeitbetriebssysteme (RTOS) sorgen für sofortige Reaktion auf Ereignisse, minimalen Stromverbrauch, skalierbare CPU-Leistung und vor allem speicherschonenden Betrieb.

mit einem Funktionsumfang vom elementaren logischen Inverter bis zur Fehlerbehandlung und konfigurierbaren Task. Viertens ermöglicht die durchgängige Entwicklungsumgebung das Arbeiten im Team. Fünftens steht eine breite Palette fertiger IP in Form von Bibliotheken zur Lösung verschiedener

chen (GUI) passen in die Welt der High-End-PXI-Mess- und Regelsysteme. Für den Anwendungsprogrammierer interessant ist die nahezu beliebige Skalierung von Rechenleistung und Prozess-I/O. Angefangen beim Standalone Embedded Controller mit robustem Echtzeitbetriebssystem über den Hybrid »Windows + LabVIEW-Realtime« bis zur Möglichkeit, einen FPGA für sehr hartes Timing einzubinden, passt sich das System nahtlos an die Projektanforderungen an. Zusammen mit flexibler Stromversorgung von 10–30 V_{DC} bis 230 V_{AC}, einem für industrielle Umgebungen konzipierten Lüftersystem und zertifizierten Grund- und Funktionsmodulen lässt diese »Spielwiese« bezüglich Funktionalität, Leistung und Qualität wenige Wünsche offen. Im Hybridmodus kombiniert das PXI die Vorteile vom Windows-Standard mit den Vorteilen eines robusten Echtzeitbetriebssystems. Ersteres bietet Möglichkeiten wie komfortable GUIs, Datenbankanbindungen oder Zugriff auf die Cloud. Letzteres erlaubt robusten 24/7-Dauerbetrieb sowie harte (LabVIEW Realtime) als auch sehr harte Echtzeit (FPGA). Die zwei unterschiedlichen »Welten« werden auf einer Multicore-CPU in je einem Core abgearbeitet. Der Datenaustausch erfolgt in gewohnter Weise über Shared-Memory oder TCP/IP. Eine für Datastreaming oder -recording interessante Option ist die Anbindung von Solid-State-Disks im oberen Gigabyte- oder Festplatten im Terrabyte-Bereich.



Warum Grafische Programmierung?

Der Wunsch nach mehr Funktionalität und Komfort nimmt bei Embedded-Systemen generell zu. Damit steigt exponentiell mit den Anforderungen die technische Komplexität der Software. Für die Entwicklung dringender benötigte Ressourcen wie Geld oder Personal sind meist begrenzt. Das Killerkriterium im Entwicklungsgeschäft aber ist Zeit, die verloren geht und nicht mehr eingeholt werden kann. Damit punkten funktionelle, grafische Programmiersprachen wie »LabVIEW«, da sich mit ihnen im Vergleich zu traditionellen Ansätzen sogar komplexe Probleme im Bruchteil der üblichen Zeit lösen lassen. Deshalb dringen sie vom klassischen PC-Bereich zunehmend in die Welt der Embedded Systeme.

Warum »LabVIEW Embedded«?

Folgende Eigenschaften prädestinieren LabVIEW zum Einsatz in Embedded-Systemen, die bislang vom Platzhirsch »C/C++« dominiert worden ist. An erster Stelle steht die enorme Produktivität, mit der sich Ideen umsetzen lassen. Zweitens ermöglicht die inhärente Parallelität der Sprache Multitasking- und Multicorebetrieb. Drittens ist LabVIEW eine vollumfängliche Programmiersprache

Signalverarbeitungsprobleme zur Verfügung. Sechstens lassen sich »trockene« Algorithmen schnell per Drag&Drop mit live I/O verknüpfen. Und siebentens ist LabVIEW ein Crosscompiler, der verschiedene Programmiermodelle unterstützt und diesen Sprachmix auf verschiedene Hardware-Systeme verteilen kann. Diese zwei in traditionellen Embedded-Lösungen üblichen Software-Techniken sind bei LabVIEW direkt von unterlegter Embedded-Hardware abhängig: effizienter Umgang mit CPU- und Speicherressourcen und unmittelbare Reaktionen auf Ereignisse. Die Wahl der Hardware ist also ein wichtiger Meilenstein, denn die damit verbundene Leistungsreserve hat einen direkten Einfluss, ob sich die Projektanforderungen wie gewünscht umsetzen lassen, ohne diese Entscheidung später revidieren zu müssen.

Der PXI als Industrierechner der oberen Leistungsklasse

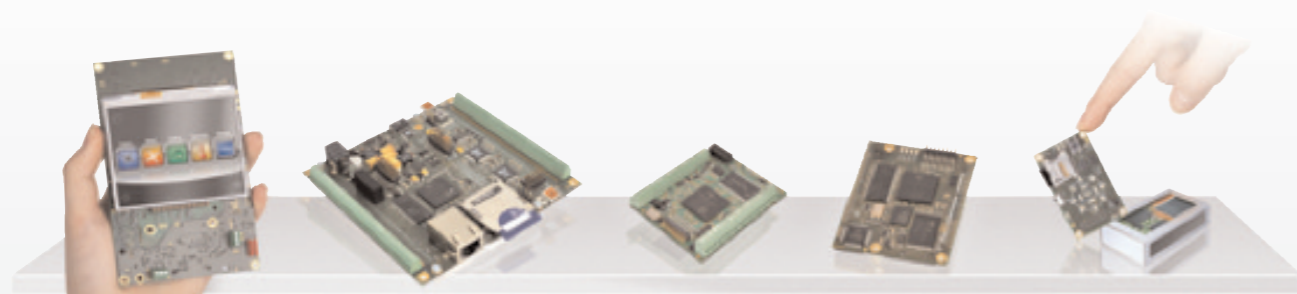
Data-, Sound- und Videostreaming, kontinuierliche High-Speed-Datenerfassung und -verarbeitung im MHz-Bereich, das Einbinden mehrerer Tausend Messkanäle, komplexe Kommunikationsanwendungen und komfortable grafische Bedienoberflä-

Die modularen Messrechner

CompactRIO und Single-Board RIO

Kompakte Standalonesysteme für Datenerfassung, Steuerung und Regelung mit hohen Ansprüchen an zuverlässigen Dauerbetrieb und einem breiten Spektrum an Prozesssignalen lassen sich mit den CompactRIO/cRIO und Single-Board RIO/SBRIO-Plattformen realisieren. Das cRIO steht für robuste Gehäuse mit Echtzeitcontroller, FPGA-Backplane für rekonfigurierbaren I/O und verschiedene Hotplug-I/O-Module. Das SBRIO





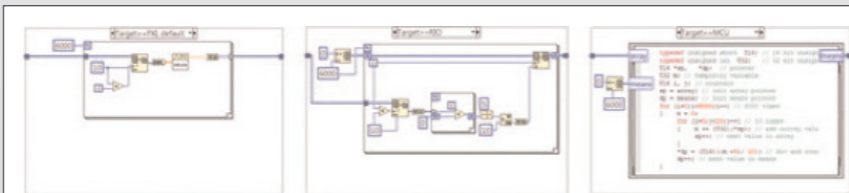
Fällt die Wahl aufgrund zwingender Vorgaben wie Formfaktor oder Stromverbrauch auf ein kundenspezifisches ARM/Blackfin-Microcontroller-Target, müssen Portierbarkeit, Speicherbedarf und Ausführungsgeschwindigkeit genau evaluiert werden.

ist die »Boardvariante« derselben Hard- und Software-Architektur, konzipiert für den Serienseinsatz. Die im Vergleich zum PXI deutlich tiefere Verlustleistung zwischen 10–20 W, eine 10–30 V_{DC}-Speisung, der kompakte Formfaktor und zahlreiche industrietaugliche I/O-Module prädestinieren die Plattform für den typischen harten SPS-ähnlichen Industrie-Einsatz. Die Verfügbarkeit des Realtime-Codes (Booten) in einer Größenordnung von 20 Sekunden und ein Software-Timing auf dem Echtzeit-Controller im Millisekundenbereich sind für viele industrielle Anwendungen ausreichend. Hardware-seitig kombinieren die Plattformen einen schnellen (typischerweise bis zu 800 MHz getakteten) Floating-Point-Applikationsprozessor mit einem parallel arbeitenden FPGA. Dieser verarbeitet I/O und logische Operationen parallel und deterministisch im Nanosekundenbereich und ermöglicht parallele I/O-Verarbeitung und äußerst schwierige Timings auch unter harten Echtzeitbedingungen. Damit lässt sich Jitter, der aufgrund der Architektur des Echtzeitbetriebssystems üblicherweise vorhanden und abhängig von parallelen Threads sind, kompensieren.

Grafische Programmierung von FPGAs

Ein FPGA ist eine im Kern software-rekonfigurierbare, parallel arbeitende Hardware. An ihn werden zeitkritische Aufgaben und I/O delegiert, um die CPU des Hauptcontrollers zu entlasten. Dabei gibt es zwei entscheidende Eckdaten: die realisierbare Funktionalität und deren Timing. Bislang war FPGA-Programmierung Spezialisten vorbehalten. Dank LabVIEW erhalten Ingenieure aus allen Sparten Zugang zur mächtigen Technologie rekonfigurierbarer Logik. Genauer betrachtet kommt die echte Parallelität dem LabVIEW-Datenflussparadigma am nächsten. Über intuitive Funktionsblöcke lässt sich ein breites Spektrum analoger, digitaler und serieller Prozesssignale einbinden, verknüpfen und parallel vorverarbeiten, bevor sie in die CPU weitergeleitet werden. Die Funktionalität der abzubildenden Anwendung auf einem FPGA ist direkt durch die zur Verfügung stehenden Gatter begrenzt. Bekannte Kennwerte, welche Operation (Addition, Filter, FFT) wieviele Gatter benötigt, liefern deshalb wertvolle

Entscheidungsgrundlagen bei der Auswahl der FPGA-Leistungsklasse. Beim Timing »denkt« der Anwendungsprogrammierer in »Ticks«, dem kleinsten Zeitinkrement in der Größenordnung von Nanosekunden auf dem FPGA. Dabei ist definiert, wieviel Zeit von bestimmten logischen und mathematischen Operationen und I/O-Zugriffen benötigt wird. Dies gibt Richtwerte für das Systemtiming. Beim Software-Design gilt es, die Embedded-Anwendung sorgfältig auf CPU und FPGA aufzuteilen. Der Hauptcontroller ist verantwortlich für die High-Level-Hauptfunktionen. Low-Level Details wie zeitkritischer Code, digitale Filter, kombinatorische und sequentielle Logik, Skalierungen, Fixed-Point- und Integer-Arithmetik werden auf den FPGA ausgelagert. Das LabVIEW-FPGA-Diagramm wird schlussendlich in VHDL-Code synthetisiert, mit den



Über targetspezifische Definitionen passt sich dieser Mittelwert-Algorithmus der Leistung und Speicherbedarf der Zielhardware an. Von Floating-Point-Arithmetik (Links, PXI) über Integerarithmetik (Mitte, cRIO) zum Inline-C-Node (Rechts, Microcontroller).

FPGA-Tools in eine Firmware kompiliert und in den FPGA geladen. Der gesamte Workflow wird dank den LabVIEW-Tools komplett abstrahiert und erfolgt auf Knopfdruck. Der »Build« mit Synthetisieren, Kompilieren und Flashen kann je nach Komplexität der FPGA-Anwendung mehrere Stunden dauern, was beim FPGA- und Hardware-Design jedoch üblich ist.

LabVIEW auf ARM-Microcontroller

Verlangt die Anwendung nach stromsparenden, kleinsten Formfaktoren im Scheckkartenformat und Boardpreisen im zweistelligen Bereich, ist ein ARM-Cortex-M3-Microcontroller (32 Bit) mit skalierbarer On-chip-Peripherie die erste Wahl. Den Vorteilen großer Peripherievielfalt und interessanten Komponentenpreisen stehen im Hinblick auf Labview-Applikationen Nachteile wie limitierte CPU-Leistung (50 MHz Prozessortakt)

und geringer Speicher (64 kByte RAM, 256 kByte Flash) gegenüber. Obwohl Compiler und Linker über Optimierungsroutinen für Codelänge und -performance verfügen, entsteht bei grafischer Programmierung in Labview wegen der hohen Abstraktion immer ein Overhead gegenüber manueller C-Programmierung. Das geht zu Lasten des Speicherbedarfs und der Ausführungszeit. Damit stehen der Anwendung die im Datenblatt des Prozessors spezifizierten Ressourcen nicht komplett zur Verfügung. Eine frühzeitige Festlegung auf diese Hardware-Plattform könnte deshalb schmerzhaft Konsequenzen haben. Dann nämlich, wenn die Anforderungen die Möglichkeiten der ARM-Plattform so übersteigen, dass nicht einmal Optimierungsmaßnahmen zum Ziel führen oder so stark umcodiert werden muss, dass sich die Aufwände nicht mehr rechnen.

LabVIEW auf Blackfin Mikroprozessor

Ist der Bedarf an Rechenleistung und Speicher deutlich höher, bietet sich der Blackfin-Prozessor an. Aufgrund seiner CPU (500 MHz/32-Bit), flexiblen I/O-Ports, High-Speed-Prozessorbussen mit DMA-Support (Direct Memory Support), extern ausbaubarem DDR-RAM und schnellem Flash-Interface, lassen sich damit Anwendungen mit hoher Komplexität und nahezu beliebigen Prozess-I/Os realisieren. Die Plattform ermöglicht analoge Signalerfassung im dreistelligen kHz-Bereich, kontinuierliche Regelungen im zweistelligen kHz-Bereich sowie grafische Benutzerschnittstellen mit Touch-Bedienung. Ebenso stehen mobile und Solid-State-Speichermedien, Kommunikation über Ethernet und Wireless bis hin zum integrieren Webserver zur Verfügung, die Entwickler sofort in Projekten nutzen können. Für Maschinen mit Stand-by-Betrieb oder batterie-

gestützte Messgeräte interessant ist der skalierbare Stromverbrauch bis in den zweistelligen Milliwattbereich. Mit Bootzeiten unter einer Sekunde sind auch Systeme realisierbar, die schnell betriebsbereit sein müssen. Mit Mechanismen zur Fehlererkennung und -behebung unterstützt der Prozessor ebenso den 24/7-Dauerbetrieb.

Die Wahl der Hardware

Die häufige Folge einer Fehlentscheidung bei der Hardware kann zu kostspieligen und zeitraubenden Redesigns führen. Die größte Konsequenz hat die Entscheidung »PXI, cRIO, SBRIO, FPGA« oder »ARM-Microcontroller/Blackfin-Microprozessor«, denn hier spaltet sich das LabVIEW-Paradigma. Bei vielen Embedded-Anwendungen kann diese Wahl anhand der Projektanforderungen sofort und eindeutig getroffen werden. Passen Formfaktor, Leistungsaufnahme, Timing und Systemkosten zu PXI, cRIO oder SBRIO, sind diese Plattformen die richtige Wahl. Der Entwickler genießt dann den Komfort einer durchgängigen Toolchain, die nahtlos auf die unterlegte Hardware abgestimmt ist. Zudem lässt sich die Entscheidung zur definitiven Zielhardware in der Regel fast beliebig hinausschieben, ohne dass dies signifikante Auswirkungen auf die Embedded-Software hat.

Konsequenzen für die Software

Je nach Anforderungen und Hardware-Wahl kann sich der Anwendungsentwickler Freiheitsgrade schaffen, um die Embedded-LabVIEW-Software notfalls plattformübergreifend einsetzen zu können. Beschränkt sich die Auswahl auf je eine Plattform, dann gilt es, diese drei Richtlinien zu beachten:

- State machines einsetzen und objektorientierte Ansätze verfolgen
- Über den limitierten Speicher verantwortungsvoll verfügen und den LabVIEW-Memorymanager weitgehend entlasten (Arrays, Strings, Cluster).
- Der schwächeren CPU-Leistung Rechnung tragen und Integer- oder Fixed-Point-Arithmetik der Floating-Point-Alternative vorziehen.

Werden zusätzlich diese vier Regeln konsequent eingehalten, lässt sich eine überblickbare Anwendung mit minimalen Anpassungen zwischen allen Plattformen portieren:

- Auf exotische Funktionen verzichten und nur die Kernfunktionen von LabVIEW verwenden. Für maximale Portierbarkeit dient die von LabVIEW am schwächsten unterstützte Plattform: der ANSI-C-Code-Generator
- Signalverarbeitungs- und Mathematik-Bi-

bliotheken vermeiden, die nicht von allen Hardware-Systemen unterstützt werden

- Jeden Zugriff auf Hardware abstrahieren und zeitkritische Algorithmen über target-spezifische Definition auswählen
- Die Vielzahl an möglichen Software-Architekturen auf den Single-Loop mit einfacher State machine beschränken.

Sobald die fehlende CPU-Leistung und der limitierte Speicher nach tiefgreifendem Code-Optimieren verlangt, kommt irgendwann der »Point-Of-No-Return« und ein Plattformwechsel ist unmöglich oder nur mit sehr viel Aufwand verbunden.

Fazit

Neben der Funktionalität einer Embedded Applikation stehen vor allem die Qualitätsmerkmale wie Stromverbrauch, Bootzyklen, Determinismus, stabiler Dauerbetrieb, Formfaktor und Kosten im Vordergrund. Die Wahl einer von LabVIEW unterstützten Hardware-Plattform beeinflusst direkt diese Parameter und in welchem Maß die Software gegebenenfalls optimiert werden muss. Ist eine Unterscheidung zwischen NI-Plattformen (PXI, cRIO, SBRIO) und NI-unabhängigen Mikroprozessorplattformen (ARM, Blackfin) kritisch, gilt es das Pflichtenheft zu hinterfragen und die Spezifikationen notfalls zu ändern, Kosten neu zu kalkulieren und die Lösungsalternativen abzusichern. Für Entwickler und Entscheidungsträger zählt bei solchen Machbarkeitsprüfungen wie auch beim Rapid-Prototyping und in der Entwicklung von Serengeräten dann vor allem der Zeitgewinn, der sich mittels grafischer Programmierung erzielen lässt.

Für die Themen LabVIEW, PXI, CompactRIO, SingleBoardRIO ist National Instruments der richtige Ansprechpartner (www.austria.ni.com), bei ARM und Blackfin ist das Marco Schmid, Schmid Elektronik zu kontaktieren.

Ihr persönlicher Ansprechpartner

Marco Schmid
Geschäftsführer



Schmid Elektronik AG

Tel.: +41 71/969 35 90
Fax: +41 71/969 35 98

info@schmid-elektronik.ch
www.schmid-elektronik.ch



Sensortechnik

Tapeswitch macht sicher



- Schaltmatten
- Bandschalter
- Schaltweiser
- Lichtvorhänge
- Sonderlösungen

Anwendungsbereiche:

- In der Produktion
- An Maschinen und Anlagen
- Zutritts- und Anwesenheitskontrolle
- Zeitmessung bei Wettkämpfen
- Und für weitere Anwendungen im Bereich Schaltlogik und Sicherheit

Fragen Sie nach Ihrer Sicherheitslösung!

Bernstein GmbH Österreich
Kurtze Gasse 3, A-3544 Leobersdorf
Telefon +43 102 22 56/620 70
Telefax +43 102 22 56/626 38
E-mail office@bernstein.at

Alles über Bernstein:

www.bernstein.at