

Batteriebetriebene LabView-Anwendungen auf kundenspezifischen Formfaktoren

Entwicklung eines mobilen industriellen Ultraschall-Handmessgerätes

Die Produktentwicklung mobiler, leistungsfähiger Messgeräte stellt auf allen Ebenen immer höhere Anforderungen an die Flexibilität des Entwicklungsprozesses. Dank off the shelf Hardware, Customizing und grafischer Applikationsentwicklung wurde für ein Ultraschall-Handmessgerätes in Rekordzeit die Serienreife erzielt. Wie können dabei typische Hardware-/Software-Probleme, die sich in der täglichen Praxis stellen, gelöst werden.

Für die Entwicklung eines mobilen Ultraschall-Handmessgerätes (Bild 1) setzte die Sika Technology AG auf den Ansatz grafischer Systemprogrammierung mit dem «LabView Embedded Module», kombiniert mit standardisierter OEM-Hardware. Sika ist im Bereich Prozessmaterialien für das Dichten, Kleben, Dämpfen, Verstärken und Schützen von Tragstrukturen am Bau und in der Industrie tätig. Dank der Effizienz und Systemübersicht der eingesetzten Entwicklungsumgebung liessen sich auch komplexe

Aufgaben und unvorhersehbare Probleme, wie sie während der Entwicklung von Embedded-Systemen unweigerlich auftauchen, mit vernünftigem Aufwand bewältigen.

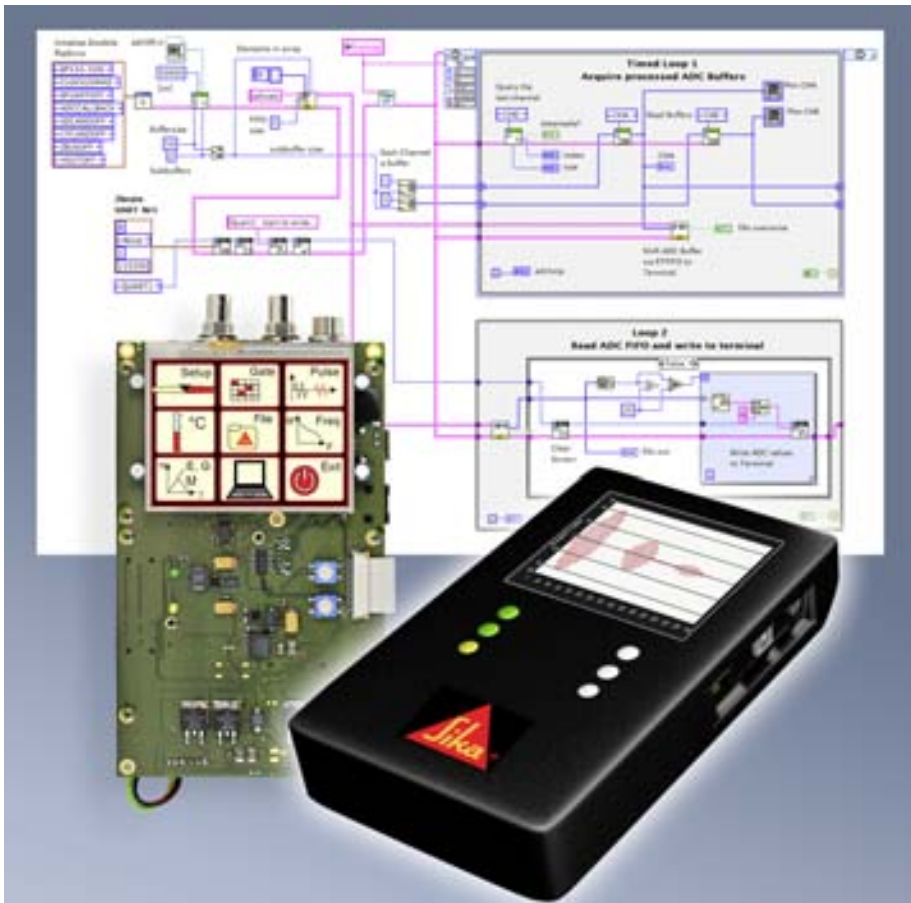
Machbarkeitsprüfung definiert die Anforderungen. Der in Bild 1 gezeigte Messhandheld muss eine Reihe von Anforderungen erfüllen, die nach einer Machbarkeitsprüfung mit LabView/PC-basierender Messtechnik spezifiziert wurden. Die Ultraschallmesstechnik benötigt eine Signalgenerierung in einer

Bandbreite zwischen 10 kHz bis 25 MHz mit einer Signalerfassung von 50 MHz, gekoppelt mit schneller Signalkorrelation und -analyse der gewonnenen Daten.

Für den mobilen Betrieb ausgelegt, musste das Gerät kompakt und ergonomisch sein, über eine lange Batterielaufzeit verfügen, über einen TFT mit Touch bedienbar sein sowie die Daten an einen Computer transferieren können, entweder direkt oder kabellos über eine Speicherkarte. Resultate aus Benchmarks während der Machbarkeitsprüfung führten zur Entscheidung, die komplexe Messaufgabe auf zwei unabhängige CPUs zu verteilen, um die zeitlichen Anforderungen zu erfüllen.

Die Applikation mit der Bedienerschnittstelle und der Hostkommunikation läuft auf dem Hauptprozessor (linke Seite von Bild 3). Er ist mit einem Co-Prozessor (rechte Seite von Bild 3) verbunden, der für die Signalgenerierung und -erfassung zuständig ist und darüber hinaus weitere physikalische Messdaten wie Temperatur, Feuchte

Bild 1: Das batteriebetriebene Ultraschallhandmessgerät konnte dank grafischer Programmierung und der standardisierten OEM-Hardware und Softwareplattform ZBrain in kurzer Zeit realisiert werden.



AUTOREN

Dr. Didier Lootens, Senior Scientist, Sika Technology

Armin Brühwiler
Systemingenieur und
Projektleiter
Schmid Engineering

Marco Schmid,
Geschäftsführer
Schmid Engineering

INFOS

Schmid Engineering AG
9542 Münchwilen
Tel. 071 969 35 90
info@schmid-engineering.ch
www.schmid-engineering.ch

Sika Technology AG
8048 Zürich
Tel. 044 436 42 47
sika@sika.ch
www.sika.com



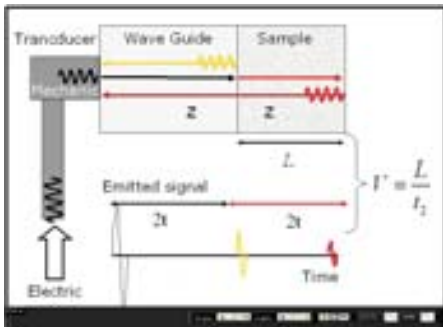


Bild 2: Ein elektrisches Signal wird mittels eines Umformers in ein mechanisches Signal konvertiert. Die so generierten Ultraschallwellen pflanzen sich durch Wellenleiter und Werkstoff fort und werden reflektiert.

und Druck erfasst. Vorverarbeitete Rohdaten und Resultate werden anschliessend synchronisiert an den Hauptprozessor zurücktransferiert.

Das Ultraschall-Messprinzip. Die Hauptfunktion «Ultraschallmessung» folgt dem Ortungsprinzip der Fledermaus: Ein Schwingungspaket wird ausgesandt und die Zeit gemessen, die verstreicht, bis ein Echo zurückkommt (Bild 2). Dieses Verfahren kann, wie im vorliegenden Fall, auch in der Prüftechnik zur Materialanalyse verwendet werden. Es liefert entscheidende Hinweise zur Werkstoffqualität. Im Unterschied zu akustischen Verfahren liegen die Frequenzen hier im MHz-Bereich.

Mit einem von Sika Technology entwickelten Algorithmus wird aus dem Echosignal die genaue Antwortzeit und die Amplitude bestimmt, woraus Rückschlüsse über den zu prüfenden Werkstoff gewonnen werden. Das Gerät verfügt über einen Multiplexer, um die Ultraschallmessung auf mehreren verschiedenen Kanälen durchführen zu können. Zu jeder Messung werden wichtige Zusatzparameter wie Temperatur gespeichert.

Mobilität als Schlüsselfaktor. Mobile, batteriebetriebene Geräte müssen klein und handlich sein und möglichst wenig Strom verbrauchen, damit eine lange Betriebszeit gewährleistet ist. Oftmals scheidet daher schon aus rein technischen Gründen der Einsatz eines Embedded PC aus. Folgende Punkte sind für mobile Geräte im Allgemeinen gültig und treffen auch auf das Ultraschallmessgerät zu:

Geringer Stromverbrauch: Der Stromverbrauch passt sich im Idealfall der jeweiligen Belastung des Gerätes an, das heisst, während komplexer Berechnungen schaltet sich das Gerät in einen leistungsstärkeren Modus mit höherem Stromverbrauch und kehrt danach in einen Sparbetrieb zurück. Dies geschieht zum Beispiel über einen dynamisch veränderten Prozessortakt oder Variation der Core-Spannung. Display und

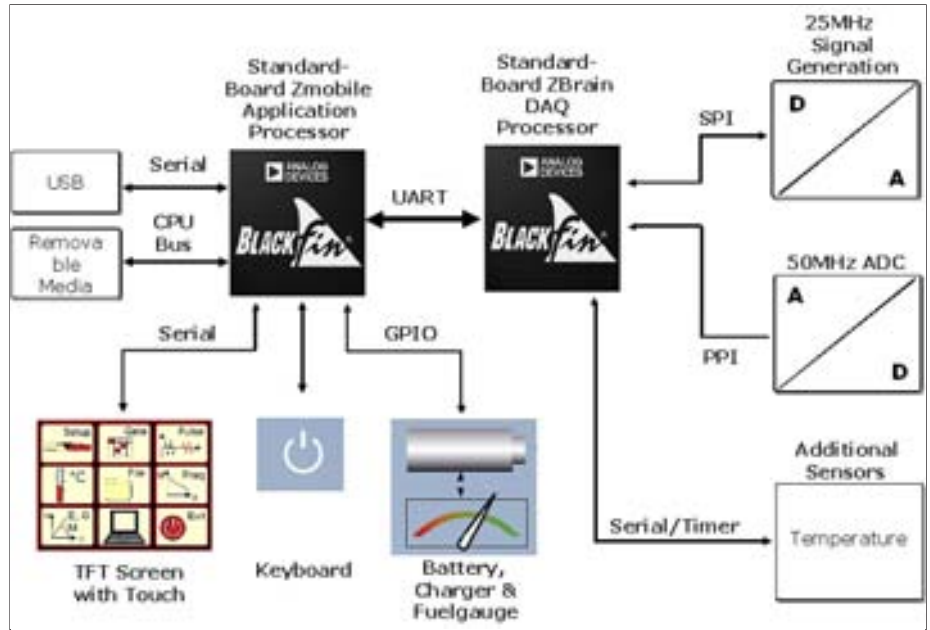


Bild 3: Blockschaltbild des Ultraschall-Handmessgerätes mit Applikationsprozessor (links) und einem zusätzlichen Co-Prozessor (rechts) für deterministische Datenerfassung und Signalanalyse.

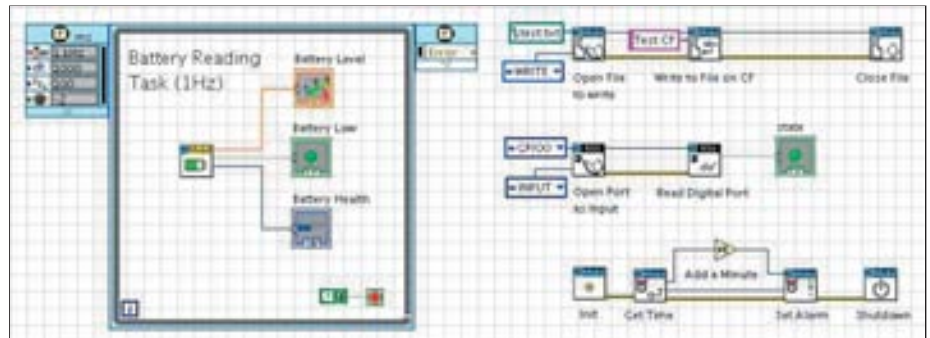


Bild 4: Programmiermodell der Embedded Plattform ZMobile: Batterieüberwachung (links), Alarme setzen und sich selbst ausschalten (unten rechts), Digitaler I/O (Mitte rechts) und lesen/schreiben auf Speichermedien wie CF-Karte (oben rechts).

Messelektronik sollten separat ein- und ausgeschaltet werden können.

Batteriemuschaltung und Akkuladung: Wenn das Gerät am Netz betrieben wird, soll der Akku geladen werden. Wenn das Gerät vom Netz getrennt wird, soll unterbruchfrei auf Akkubetrieb umgeschaltet werden.

Batteriezustandsanzeige: Für mobile Geräte ist es zwingend, den Zustand der Batterie zu kennen und den Benutzer darüber zu informieren. Viele Batteriepacks haben bereits Ladezustandsschaltungen integriert.

Automatisches Power Off: Wenn keine Benutzereingabe erfolgt, soll das Gerät nach einer einstellbaren Zeit in einen Stand-by-Betrieb übergehen.

Automatisches Power On: Für viele Langzeit Datalogging-Anwendungen macht es Sinn, dass sich das Gerät nach einer definierten Zeit automatisch wieder einschalten kann.

User-Interface: Dem Benutzer soll ein ergonomisches Interface zur Verfügung gestellt werden. Hierbei ist ein Touchscreen meist die erste Wahl. Für Anwendungen mit wenig Benutzerinteraktion genügen LEDs

und Knöpfe.

Prozess I/O: Entsprechend der Aufgabenstellung sollen analoge Werte gemessen und ausgegeben oder digitale Ports gelesen und gesetzt werden.

Datenspeicherung: Bei Datalogging-Systemen fallen grössere Datenmengen an, die vorzugsweise auf portable Datenträger wie Compact-Flash-Karten gespeichert werden.

Datentransfer zu einem PC: Für den Benutzer sehr komfortabel ist der Transfer via USB.

Die Realisierung all dieser Punkte stellt den Systementwickler vor aufwändige Detailaufgaben. Sie belegen einen Grossteil seiner kostbaren Entwicklungszeit, die bei der Arbeit am Kernproblem dann fehlt.

Das Ultraschallmessgerät verfügt über verschiedene Betriebszustände, um die Batterie zu schonen und so einen Akkubetrieb von bis zu 24 Stunden zu ermöglichen (Bild 4 links). Für Langzeitmessungen wird das Display komplett ausgeschaltet und über einen Tastendruck wieder aktiviert. Bei abgeschaltetem Display zeigt eine blinkende LED den Zustand des Gerätes an. ➔



Bild 6: Innenansicht des Messhandhelds, der eine kundenspezifische Basisplatine mit einem aufgesteckten OEM-Standardmodul kombiniert.

Bedienschicht per Drag-and-Drop gestalten und optimieren.

Bei der Realisierung der grafischen Bedienschicht (Graphical User Interface, GUI) wurde viel Wert auf hohe Abstraktion und Ergonomie gelegt (Bild 6). Die komplizierte Technologie sollte verborgen und nur die Endresultate dem Benutzer angezeigt werden.

Die üblicherweise sehr aufwändige Gestaltung und Ausprogrammierung eines GUI ist dank Schmid Engineerings «Fast Debug Mode» (FDM, ähnlich wie HIL = Hardware-In-The-Loop) sehr effizient. Es war nicht mehr nötig, das GUI wie bisher in einer Art Simulation am PC zu entwickeln, sondern es konnte online direkt auf dem Zielsystem

gestaltet, implementiert und bearbeitet werden. Damit konnte Sika Technology schon früh Produktdesigner und Anwender in die praxisgerechte und ergonomische Gestaltung mit einbeziehen.

Das komfortable, grafische Debugging (FDM) machte zeitaufwändige C-Code-Generierung für den Grossteil der Applikationsentwicklung überflüssig und beschleunigte so gerade die GUI-Entwicklung um Faktoren.

LabView Embedded macht Signalanalyse einfach.

Klassischerweise wird LabView für eine Vielzahl von Problemstellungen aus der Mess-, Prüf- und Automatisierungstechnik angewendet. Dabei wird meistens eine vom PC gesteuerte Hardware, zum Beispiel eine Einsteckkarte oder ein dezentrales Chassis (PXI, cRIO) eingesetzt. Für mobile Geräte hingegen, bei denen Gewicht, Grösse und Stromverbrauch die limitierenden Faktoren sind, war der Entwickler bisher gezwungen, in die Tiefen der Low-Level-Programmierung zu tauchen und eine textbasierte Programmiersprache zu verwenden.

Neu können Mikroprozessoren direkt mit LabView programmiert werden. Dieser Ansatz ermöglicht die durchgehende grafische Programmierung eines kompletten Embedded-Systems. Der LabView-Anwender erhält damit die Möglichkeit, in seinem vertrauten Umfeld mobile Anwendungen zu entwickeln. Damit wird Embedded-Technologie auch für Wissenschaftler, Systemingenieure und Domänenexperten, denen dieses Know-how häufig weniger geläufig ist, zugänglich.

Der hohe Abstraktionsgrad der grafischen Programmierung eignet sich sowohl für einfache als auch für hochkomplexe Systeme. Die Berechnung einer Fourier-Transformierten wird nun plötzlich so einfach wie das Ansteuern eines digitalen Ports.

Grundsätzlich lässt sich jeder 32-Bit-Mikroprozessor mit LabView programmieren. Allerdings ist ein gewisser Aufwand notwendig, um die jeweilige Prozessor-Toolchain in die LabView-Umgebung zu integrieren. Für Analog-Devices-Blackfin-Prozessoren und ARM/Cortex-Mikrocontroller ist dieses Problem bereits gelöst.

Was unterscheidet sich zur Programmierung auf einem PC?

Dem Programmierer steht die übliche LabView-Umgebung mit Paletten, Blockschaltbild und Frontpanel zur Verfügung. Die grafische Bedienschicht wird mit LabView Frontpanel auf dem PC entworfen, auf dem Embedded TFT getestet und schliesslich als Prozessorcode in die Zielplattform geladen.

Die meisten LabView-VIs (VI: Virtual Instrument = Funktionsblock), auch Mathematik- und Signalverarbeitungsblöcke, stehen weitgehend uneingeschränkt zur Verfügung. Die Board-Spezifischen Funktionen, welche die jeweilige Hardware bietet, können per Drag-and-Drop aus der VI-Palette in die Applikation eingefügt werden (Bild 4). Was neu hinzukommt, sind Target-spezifische Merkmale wie das Erstellen und Ausführen von Real-Time Code oder Laden einer Standalone-Applikation ins Flashmemory des Zielsystems. Ausserdem verfügt ein Embedded-System über eingeschränkte Speicher- und Prozessorressourcen, was betreffend Systemleistung zu Engpässen führen kann. Regelmässige Benchmarks können deshalb vor Überraschungen schützen. Zudem ist es von Vorteil, sich mit Fixed-Point-Arithmetik vertraut zu machen, da vielen Mikroprozessoren und -controller eine FPU (Floating Point Unit) fehlt.

Grafisch schneller zum Produkt. Die grafische Programmierung ermöglicht eine effiziente, flexible und übersichtliche Entwicklung. Ähnlich wie bei Sika Technology ist LabView in vielen Firmen ein verbreitetes und geschätztes Werkzeug. Damit besteht grosses Potenzial, vorhandenes Know-how und bestehende Lösungen in Form mathematischer Algorithmen direkt in ein Serienprodukt einfließen zu lassen. Und mit Wissenschaftlern, Systemingenieuren und Domänenexperten zusätzliche Manpower ins Projektteam zu holen.

Als gemeinsame Sprache zwischen Kunde, Management und Entwicklungsteam verhilft LabView-Embedded-Projekten zu neuer Dynamik und verstärkter interdisziplinärer Zusammenarbeit. Dank OEM-Hard und Software wie dem ZBrain System sowie der Echtzeitexpertise von Drittherstellern gelingt der Einstieg in die grafische Programmierung von mobilen und stationären Geräten mühelos. (pm)

Komfortable Entwicklungstools für Grafisches Embedded Systemdesign



Das ZBrain-System ist ein flexibles Entwicklungspaket mit CPU und Prozess I/O für Embedded Mess- und Regeltechnikaufgaben. Im Kit enthalten sind:

- Entwicklungsumgebung NI LabVIEW «Full»
- Embedded-Modul für ADI Blackfin mit C-Code Generator
- VisualDSP++ Blackfin IDDE
- Lizenzfreier RTOS Realtime-Kernel
- Entwicklungshardware
- ZBrain BSP mit Echtzeitdiensten für [µs]-Betrieb
- +200 Embedded-VI's, zusätzlich zur Standardpalette
- Grafisches Live-Debugging
- High-Speed ICE (Option), Flashtools