

# Zwanglose Geräteplattform

Bilder: Photo-X, Fotolia

Bei Firmen, die jahrelang gewachsene Produktfamilien entwickeln, entsteht mit der Zeit ein Sammelsurium von Controllern, Entwicklungsumgebungen und Software-Versionen. Qiagen hat für seine Laborgeräte ein „Generic Electronic System“ entworfen, das für Vereinheitlichung sorgt, trotzdem aber ein hohes Maß an Flexibilität wahrt.

Von Roger Hüppi, Dr. Claus Kühnel und Daniel Zwirner

Qiagen Instruments AG Generic Electronic System (QI GES) ist ein Konzept zur Steuerung von Laborgeräten bestehend aus unterschiedlich ausgestatteten Controller-Modulen mit entsprechender Firmware. Bei diesen zur Automation von Probenvorbereitung und Probenanalyse eingesetzten Laborgeräten handelt es sich um unterschiedlich komplexe mechanische Systeme, die entsprechend skalierbare Steuerungskonzepte erfordern. Der aktuelle Systemzustand wird über verschiedene Sensoren erkannt, mit dem geforderten Systemzustand bzw. den Anforderungen des abzuarbeitenden Workflow verglichen und Aktoren wie Heiz-/Kühlstationen, Motoren, Halbleiterrelais u.a.m. angesteuert.

Dieses Gesamtsystem ist für die fehlerfreie Verarbeitung biologischer Proben von der Eingabe der Probe bis hin zur Ausgabe von Analyseergebnissen verantwortlich. Hinzu kommt bei moder-

nen Laborgeräten die Integration in ein Labornetzwerk bzw. die Anbindung an ein Laborinformationssystem (LIMS), um von dort die Aufträge zur Probenbearbeitung zu erhalten und/oder die Resultate der Tests abzulegen. Der gleiche Zugang wird für den Fernzugriff zu Zwecken der vorbeugenden Wartung genutzt.

## Automatisierung in der Life Science

Die Entwicklung automatisierter Systeme für die gesamte Bandbreite der Life Sciences erfordert die Zusammenarbeit von Fachleuten verschiedener technologischer Disziplinen:

- Biologen – Entwicklung der zu automatisierenden Prozesse und die Prozessintegration,
- Systemingenieure – Anforderungsanalyse und Systemkonzeption,
- Maschinenbauingenieure – Entwicklung der Mechanik der gesamten Geräteplattform,

- Elektronikingenieure – Entwicklung der Steuerungselektronik inkl. Sensorik und Antrieben,
- Software-Ingenieure – Entwicklung der Hardware-nahen Schichten der Software (Firmware),
- Software-Ingenieure und Informatiker – Entwicklung der prozessnahen Schichten (die eigentliche Applikation) und des GUI,
- Kunststoffingenieure – Entwicklung der Labware (Verbrauchsmaterialien).

Die gelisteten Disziplinen zeigen, dass interdisziplinäre Entwicklungsarbeit eine Grundvoraussetzung bei solch komplexen Vorhaben ist. Gleichzeitig ist die technologische Basis für Probenaufbereitung und -analyse ständig zu erweitern, um eine erfolgreiche Entwicklung des Unternehmens im hochdynamischen Life-Science-Markt zu sichern. Die Verbreiterung der technologischen Basis erfolgt heute bei den Life-Science-Unternehmen in den meisten Fällen über Akquisitionen von spezialisierten Unternehmen oder Spinoffs von Universitäten.

Durch diese Strategie entfallen praktisch die Kosten für Forschung und Entwicklung, und die Time to Market kann für das entsprechende Produkt maßgeblich reduziert werden [1]. Schattenseite der akquisitionsbedingten Erweiterung des Technologieportfolios sind Integrationskosten, uneinheitliche Kon-

zepte und Implementierungen der Geräteplattformen und damit erhöhte Aufwendungen im Produktlebenszyklus. Ziel der Überlegungen war es deshalb, die vorliegenden Konzepte von Elektronik-Hardware und zugehöriger Software im Rahmen der Weiterentwicklung der Produkte zu konsolidieren.

Bei der Untersuchung des den Autoren vorliegenden Produktportfolios zeigte sich folgende Situation: Das Elektronikdesign war durch eine breite Vielfalt gekennzeichnet; unterschiedliche Mikrocontroller mit ihren jeweils eigenen Entwicklungsumgebungen mischten sich mit den verschiedenen Programmiersprachen. In einigen Fällen war die technologische Umsetzung nicht mehr zeitgemäß. Diese Gegebenheiten haben folgenreiche Auswirkungen. Die frühzeitige Abkündigung von Komponenten führt zu schwierigem Ersatz und/oder kostenineffizienter Beschaffung auf Vorrat (life time buy). Verschärfte EMV-Anforderungen sorgen für zusätzliche Schwierigkeiten. Die Wissensbasis zur Firmware-Entwicklung

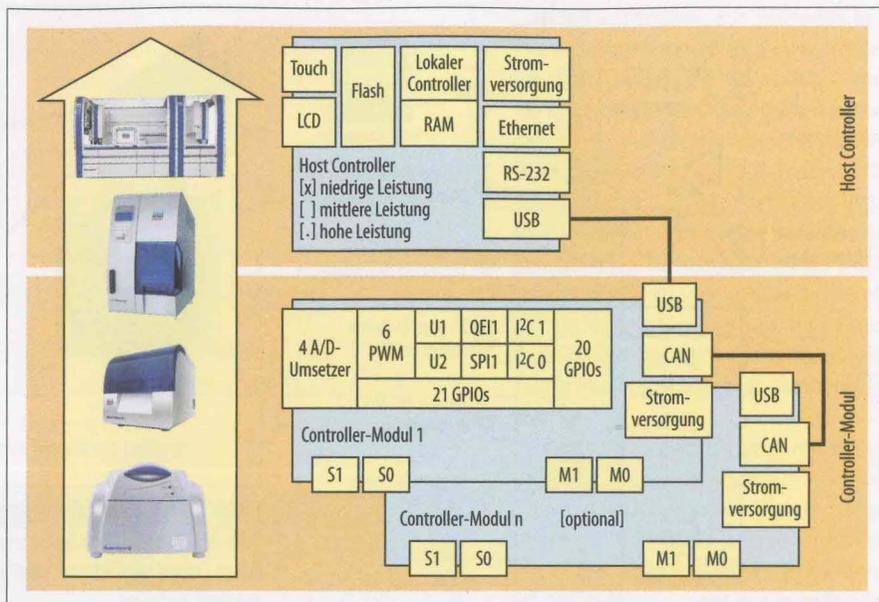


Bild 1. Standardisiertes Gerätekonzept für Laborautomaten. Zwei Hardware-Plattformen decken unterschiedliche Anforderungen an Geräte ab.

bindung – sollen zwei Konzepte für Elektronik und Firmware erarbeitet werden. Einfache Laborgeräte sollen durch ein einfaches standardisiertes Controller-Modul gesteuert werden. Eine Skalierung kann hier durch mehrfaches Verwenden des Moduls und Vernetzen via CAN erreicht werden. Zusätzliche Anforderungen wie Datenspeicherung, Netzwerkanbindung und

ge vorhält. Dieses Controller-Modul muss in einer Hardware-nahen Schicht über CAN vernetzbar sein, um die anstehenden Aufgaben auf mehrere lokale Controller verteilen zu können. Um das System in den oberen Schichten erweitern zu können, müssen die vernetzten lokalen Controller in der Lage sein, mit einem Host Controller in geeigneter Weise kommunizieren zu können.

	Gerät #1	Gerät #2	Gerät #3	Gerät #4	Gerät #5
CPU	68HC11	68HC11	Intel Celeron M	68HC11	Cortex-M3, ATmega128, MSP 430
Optics	–	PMT	Videocamera	PMT	PMT
LED	–	12	–	6	6
Stepper	5	2	2	3	5
DC-Motor	–	2	1	2	2

Tabelle 1. Analyse der bestehenden Kleingeräte mit verwendeter CPU und Peripherie.

ist inkonsistent. Eine Firmware-Dokumentation, wenn überhaupt vorhanden, ist nicht konform zu aktuellen Standards (z.B. EN 62304), woraus zusätzlicher Validierungsaufwand resultiert. Die Controller-spezifischen Entwicklungsumgebungen bieten für solche Validierungen schlechte Voraussetzungen.

### Einheitliche Geräteplattform

Um das breite Anforderungsspektrum abzudecken – vom simplen Laborgerät bis zum Laborautomaten mit grafischem User Interface und Netzwerkan-

grafisches User Interface werden durch ein zusätzliches, skalierbares Modul (Host Controller) abgedeckt. Bild 1 zeigt die beiden Konzepte.

Um ein einheitliches Systemkonzept entwickeln zu können, war eine technische Anforderungsanalyse über die unterschiedlichen Plattformen erforderlich, die für eine Reihe von Kleingeräten zum Ergebnis geführt hat, das in Tabelle 1 zusammengefasst ist. Als Minimalsystem konnte aus dieser Analyse ein Controller-Modul abgeleitet werden, das in der Lage ist, zwei Schrittmotoren anzusteuern, und ausreichend digitale und ggf. analoge Ein-/Ausgän-

Für einfache Laborgeräte kann unter den genannten Bedingungen dann eine Systemarchitektur abgeleitet werden, die in Bild 2 für ein fiktives Gerät dargestellt ist. Der lokale Controller steuert zwei Schrittmotoren direkt an und ist in der Lage, weitere Schrittmotoren anzusteuern, deren Treiber dann allerdings nicht mehr auf dem Modul, sondern auf dem Base Board angeordnet sind, auf dem sich auch die Steckverbindungen zur Peripherie befinden. LEDs und leistungsmäßig abzugrenzende Elektromagnete oder Relais können ebenfalls direkt getrieben werden. Digitale Eingänge dienen der Abfrage von Schaltern und digitalen Sensoren. Ana-

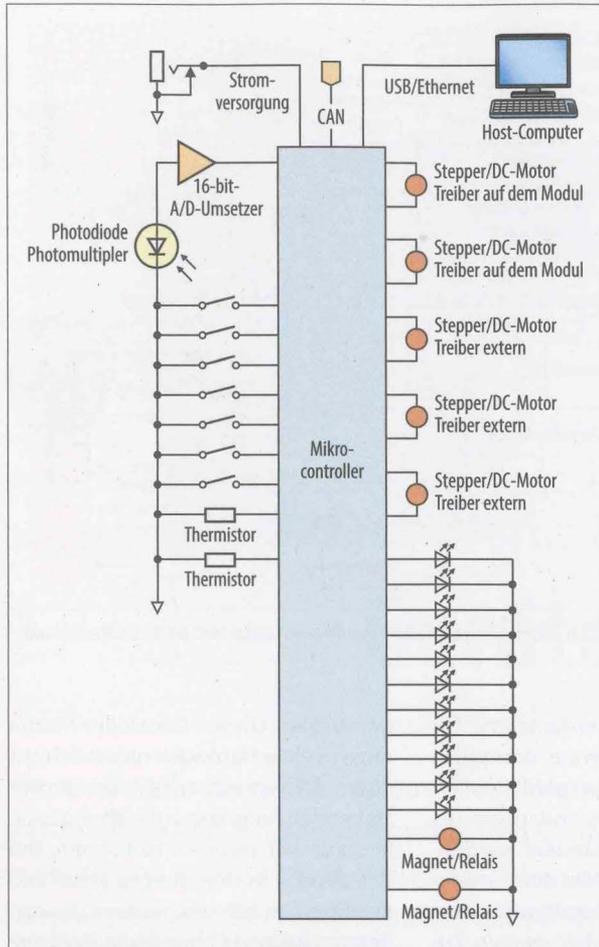


Bild 2. Systemarchitektur eines einfachen Laborgeräts.

loge Eingänge haben i.d.R. 10 bit Auflösung und sind auf dem Modul vorhanden. Sie können beispielsweise Thermistoren o.ä. auswerten. Werden hingegen hochauflösende A/D-Umsetzer für hochdynamische Signale wie Fotodioden oder Photomultiplier benötigt, dann wird man diese ebenfalls auf dem Base Board oder nahe dem Empfänger als externen Baustein mit Auflösungen zwischen 12 und 16 bit anordnen.

Den lokalen Controller, der aus der Systemarchitektur (Bild 2) abgeleitet wurde, haben wir TwinStep-M3-Controller genannt. Der Name ist Programm: Auf der Basis eines Cortex-M3 wurde ein Mikrocontrollermodul entwickelt,

ein Stellaris LM3S2965 ausgewählt, für die Treiberstufen ein Trinamic TMC429 Motion & Interface Controller und zwei Trinamic TMC262 Stepper Power Drivers.

Die insgesamt 36 außen verfügbaren GPIOs weisen Alternativfunktionen wie I<sup>2</sup>C, SPI, SSI, UART, JTAG, CAN, Quadratur-Encoder, Capture/Compare/PWM und Analog Comparator auf und bieten damit ausreichend Flexibilität für die Systemgestaltung auf der Ebene der lokalen Controller. Für die Inbetriebnahme der TwinStep-M3-Module wurde gleichzeitig ein Evaluation Board entwickelt. Dieses Board führt alle

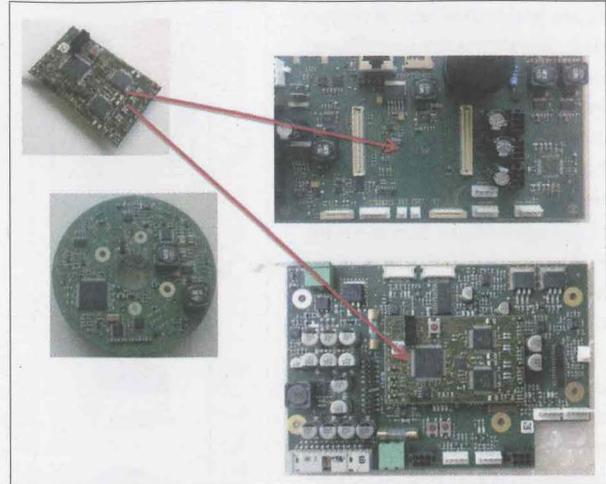


Bild 3. Module eines Laborgeräts: Durch mehrfachen Einsatz des TwinStep-M3-Moduls (oben links) wird der Funktionsumfang erhöht. Auch die runde Leiterplatte ist ein TwinStep-M3-Modul, die mit identischen Design Patterns entstand. (Bild: Qlagen)

welches neben der recht komfortablen Peripherie des Cortex-M3 zusätzlich zwei Treiberstufen für den direkten Anschluss von zwei Schrittmotoren zur Verfügung stellt. Für den Cortex-M3 wurde

Funktionen des TwinStep-M3-Moduls an Steckverbinder nach außen. Ein Lochrasterfeld kann für schaltungstechnische Erweiterungen im Nachhinein verwendet werden. Neben der Verwendung zur Inbetriebnahme kann das Evaluation Board auch als Prototypen-Board für Geräteerweiterungen dienen.

Für die rasche Inbetriebnahme des Moduls und die Integration der Software wurden Bibliotheken (Libraries) erstellt, die die Grundfunktionen des TwinStep-M3-Moduls abdecken. In Tabelle 2 sind die heute existierenden Bibliotheken gelistet.

Ein Beispiel für die Skalierung mit Hilfe mehrerer TwinStep-M3-Module zeigt Bild 3. Die komplexeren Anforderungen wurden auf drei lokale Controller verteilt. Zwei lokale Controller arbeiten mit dem TwinStep-M3-Modul, wie bereits beschrieben. Ein weiteres Modul entstand aus konstruktiven Gründen als kreisförmiger Print, welcher den Einsatz des TwinStep-M3-Moduls nicht zuließ. Hier wurde allerdings mit identischen Design Patterns gearbeitet,



Firmware Component	Bemerkung
Bootloader	für RS-232 oder CAN
Drive Control Library	Trinamic Stepper Motor Control
SD Card Library	für FAT-Dateisystem
CAN Communication	Layer 2, Layer 3 proprietär
Serial Communication	RS-232, Serial-USB, Layer 3, CRC, Time-Out etc.
Interfaces	I <sup>2</sup> C, SPI, ADC, DAC, NTC
Real Time OS FreeRTOS (royalty-free)	Upgrade zu SaferRTOS (SIL3 TÜV certified, not royalty-free) ist möglich

Tabelle 2. Firmware-Bibliotheken für das TwinStep-M3-Modul.

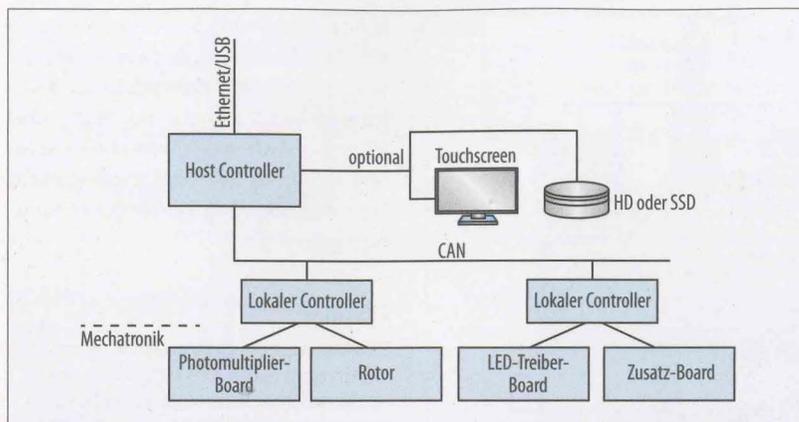


Bild 4. Erweiterbare Systemarchitektur eines komplexen Laborgerätes.

	phyCARD-S	phyCARD-L	phyCARD-XL3
Prozessor	Freescale i.MX27	OMAP 35xx	i.MX 6
Architektur	ARM9	Cortex-A8	Cortex-A9
Taktfrequenz	400 MHz	720 MHz	bis 4 × 1 GHz
RAM	bis 256 MB	bis 1 GB	bis 4 GB
Performance	ca. 1.000 MIPS	ca. 1.500 MIPS	ca. 4.500 MIPS
Preis (ca.)	50 Euro	70 Euro	100 Euro

Tabelle 3. Leistungsmerkmale verschiedener phyCARDS.

wodurch Wiederverwendbarkeit auf einem abstrakteren Level dennoch möglich war.

### Host Controller steuert große Laborgeräte

Bei komplexeren Laborgeräten kann die Ansteuerung der mechatronischen Komponenten auf verschiedene lokale Controller verteilt werden. Die Koordination der einzelnen Tasks übernimmt ein leistungsfähiger Host Controller, der dazu weitere Standardaufgaben, wie die Kommunikation mit einem übergeordneten System, die Ansteuerung eines grafischen User Interface, die Speicherung von Daten auf einem externen Datenträger (Flash Disk, Hard Disk, SSD o.ä.), übernimmt. Während sich die lo-

kalen Controller durch ihren Hardware-Bezug einer Standardisierung entziehen, arbeitet der Host Controller durch die Verwendung eines Betriebssystems (nach Anpassung an die Hardware durch ein sogenanntes Board Support Package) praktisch ohne Hardware-Bezug und über Standard-Schnittstellen mit seiner Umwelt. Bild 4 zeigt die Systemarchitektur für ein komplexeres Laborgerät unter den geschilderten Vorgaben.

Bei der Auswahl eines einzusetzenden Mikrocontrollers ist man wegen des fehlenden Hardware-Bezugs dann wesentlich freier und kann sich im Wesentlichen auf die Performance konzentrieren. In der Vergangenheit wurden

Controller-Module von Microsys und Phytec eingesetzt (Bild 5). Die beiden PowerPC-Module MPX5200 und MPX1022 von Microsys erfüllen Anforderungen auf zwei unterschiedlichen Leistungsebenen, was durch den UNIX-Bench-Index summarisch zum Ausdruck kommt. Für einen belastbaren Vergleich der Performance ist der UNIX-Bench allerdings wenig brauchbar und ein Blick in die Einzelergebnisse dieses Tests liefert deutlich mehr Aussagen. Dieser Punkt soll aber hier nicht vertieft werden.

Phytec Messtechnik setzt mit seinen phyCARD-Modulen [2] den Ansatz des skalierbaren Rechnermoduls um, indem ein phyCARD-Modul auf den 100 Pins seiner Steckerleiste definierte Standard-schnittstellen zur Verfügung stellt. Diese Schnittstellen stehen auf jeder phyCARD garantiert zur Verfügung und befinden sich stets an der gleichen Steckerposition. Das sorgt für Design-Sicherheit und die Skalierbarkeit der Rechenleistung über Architekturen hinweg ist gewährleistet.

### Skalierung auf Basis von Modulen

Einige Mitglieder der phyCARD-Modulfamilie und die damit sichergestellte Skalierbarkeit zeigt Tabelle 3. Das tatsächliche Spektrum der eingesetzten Prozessoren ist noch umfangreicher als in der Tabelle dargestellt. Es reicht vom AM335x Sitara (ARM-Cortex-A8-Mikroprozessoren von Texas Instruments) über die Cortex-A8-Prozessoren der OMAP35xx-Familie bis hin zu den i.MX6-Controllern von Freescale, die intern mit einem (1×1 GHz), zwei (2×1 GHz) oder vier Cortex-A9-Kernen (4×1 GHz) ausgestattet sein können. Daraus ergibt sich eine Skalierbarkeit innerhalb des gleichen Prozessors in Bezug auf die Rechenleistung.

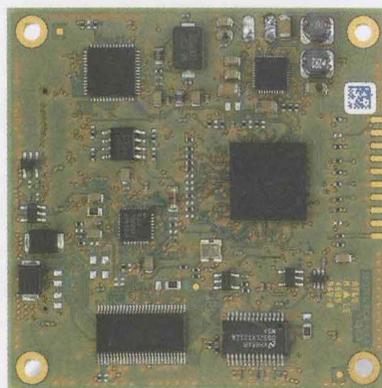
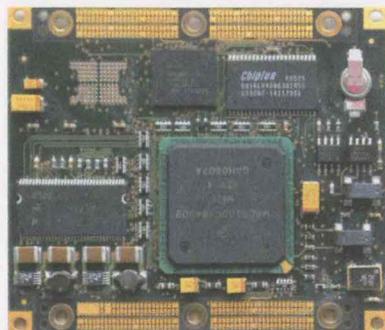


Bild 5. Links: PowerPC MPX5200, 400 MHz, 128 MB RAM, UNIXBench-Index 56,9. Mitte: phyCARD-L OMAP35xx, Cortex-A8, 720 MHz, 1 GB RAM, 1 GB Flash, UNIXBench-Index 69,7. Rechts: PowerPC MPX1022, Dual Core, 1 GHz, 2 GB RAM, UNIXBench-Index 306,7.

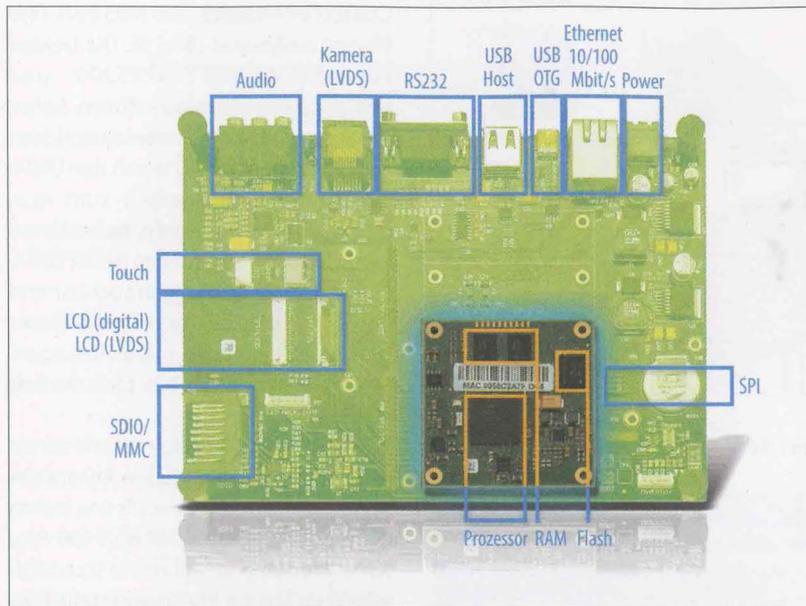


Bild 6. Architektur der Carrier Boards für die phyCARD-Module.

Die Module der phyCARD-Familie decken durch ihre Skalierbarkeit die Ebene des Host Controller in unseren Laborautomaten sehr gut ab und können damit als integrierter Bestandteil betrachtet werden. Wenn sich die Anforderungen an die Performance nicht mehr mit den Modulen aus Tabelle 3 erfüllen lassen, dann gibt es weitere Plattformen bis hin zum Cortex-A15 [3]. Selbst die Verwendung eines PC mit einem Intel-Core-i7-Prozessor ist möglich, da das QI-GES-Konzept beim Host Controller nur Standardschnittstellen (USB, Ethernet u.a.) vorsieht.

Passend zum Konzept der phyCARD-Modulfamilie kann das Phyttec-OEM-Konzept für die Umsetzung des Carrier Board herangezogen werden [4]. Das Konzept beschreibt die Vorgehensweise in drei Schritten:

→ Zuerst wird das Carrier Board anhand der OEM-Design-Library und der Anforderungen spezifiziert.

→ Dann werden notwendige Display- und eine Touch-Anforderung spezifiziert.

→ Zum Schluss werden Vorgaben für das individuelle Gehäusedesign gemacht, wenn eines benötigt wird.

Die Vorteile des OEM-Konzepts liegen im Zugriff auf die Verwendung vorgefertigter Funktionsblöcke auf Ebene von Schaltplan und Layout. Die „OEM Feature Blocks“ enthalten ausschließlich von Phyttec erprobte und im Einsatz bewährte Funktionsblöcke. Die Phyttec-Funktionsblöcke sorgen auch für Schutzmaßnahmen für einen störungs-

freien und sicheren Betrieb. Das aus den OEM Feature Blocks entwickelte Carrier Board ist konform zur EMV-Richtlinie 2004/108/EG und das resultierende OEM-Produkt erfüllt die Normen EN 55022 und EN 55024.

Bild 6 zeigt die Architektur der phyCARD Carrier Boards und eine Auswahl der möglichen Funktionsblöcke. Anhand der ausgewählten Funktionen entsteht auf der Basis der in der Library vorhandenen Funktionsblöcke (Schema- und/oder Layout Patterns) ein kundenspezifisches Carrier Board. Das Phyttec-OEM-Konzept verfolgt damit eine identische Umsetzung, wie sie im QI GES formuliert wurden.

## Grundlage für Wiederverwendung

Das hier vorgestellte Generic Electronic System bedeutet Standardisierung von Hard- und Software-Komponenten für eine ganze Klasse von Laborgeräten und bietet damit die Grundlage für die Wiederverwendung stabiler Hardware-Komponenten und Software-Bibliotheken. Durch die Wiederverwendung erhält man eine steigende Zahl gleichartiger Hardware-Komponenten. Ein einheitliches Entwicklungsumfeld bietet darüber hinaus gute Vorbedingungen für spätere Validierungen. Alle diese Maßnahmen bewirken eine Reduktion von Projektrisiken und Entwicklungszeiten, was schlussendlich zu einer reduzierten Time to Market führt. Für die Planung der Ressourcen des Entwicklerteams bekommt man wegen

nahezu identischer Skill-Anforderungen erhöhte Flexibilität. Auf einer bekannten, stabilen und erweiterbaren Plattform lassen sich neue Konzepte und Ideen außerdem sehr effektiv evaluieren, wodurch für Produktpflege und Weiterentwicklungen eine gute Grundlage gelegt ist. *jk*

## Literatur

- [1] Wachstum durch Innovation. <http://de.creo.ptc.com/2011/06/01/capitalizing-on-growth-through-acquisitions/>
- [2] phyCARD-Konzept. <http://www.phyttec.de/de/produkte/produktphilosophie/1-phyocard.html>
- [3] EPP-Pico-OMAP5430. [http://www.embedded.rs/sites/default/files/EPP-Pico-OMAP5430\\_Datasheet.pdf](http://www.embedded.rs/sites/default/files/EPP-Pico-OMAP5430_Datasheet.pdf)
- [4] Das Phyttec-OEM-Konzept. <http://www.phyttec.de/de/produkte/produktphilosophie/phytec-oem-konzept.html>



### Roger Hüppi

studierte an der HSR Hochschule für Technik Rapperswil (CH) Elektrotechnik mit den fachlichen Vertiefungen Embedded Systems, Informatik und Regelungstechnik. Er arbeitete mehrere Jahre als Entwickler und Projektleiter für Hardware-nahe Software. Seit 2007 ist er bei der QIAGEN Instruments als Associate Director Embedded Systems für die Entwicklung von Hardware-naher Software verantwortlich.



### Daniel Zwirner

studierte an der HSR Hochschule für Technik Rapperswil (CH) Elektrotechnik mit den fachlichen Vertiefungen Informatik und Hochfrequenztechnik. Seit 2006 ist er bei der QIAGEN Instruments AG als Senior-Software-Ingenieur Embedded Systems für die Entwicklung von Hardware-naher Software verantwortlich.



### Dr.-Ing. Claus Kühnel

studierte und promovierte an der Technischen Universität Dresden auf dem Gebiet der Informationselektronik und bildete sich später in Biomedizintechnik weiter. Seit 2004 ist er bei der QIAGEN Instruments AG in Hombrechtikon (CH) als Director Electronic Engineering & Embedded Systems für die Entwicklung von Elektronik-Hardware und Hardware-naher Software verantwortlich.

[info@ckuehnel.ch](mailto:info@ckuehnel.ch)