



Gute Ideen gibt es viele, aber für die Umsetzung fehlt häufig die Zeit und das Geld. Kommerzielle Entwicklungssysteme können helfen, wenn man bei der Auswahl einige wichtige Punkte beachtet.

Von der Idee zum Prototyp

Fokus auf die Kernkompetenz

Eine gute Idee macht zunächst kaum Aufwand und kostet erst einmal so gut wie nichts. Oftmals scheidet die Realisierung aber am Zeit- und Geldmangel. Viele gute Ideen werden daher schon frühzeitig verworfen, ohne dass ihr tatsächlicher Nutzen und ihr Potential jemals in einem Feldtest geprüft wurde. So manche erfolgversprechende Innovation wird daher nie realisiert. Begründet wird dies häufig mit zu hohen Kosten für die Erstellung einer geeigneten Hardware in kleinen Stückzahlen sowie

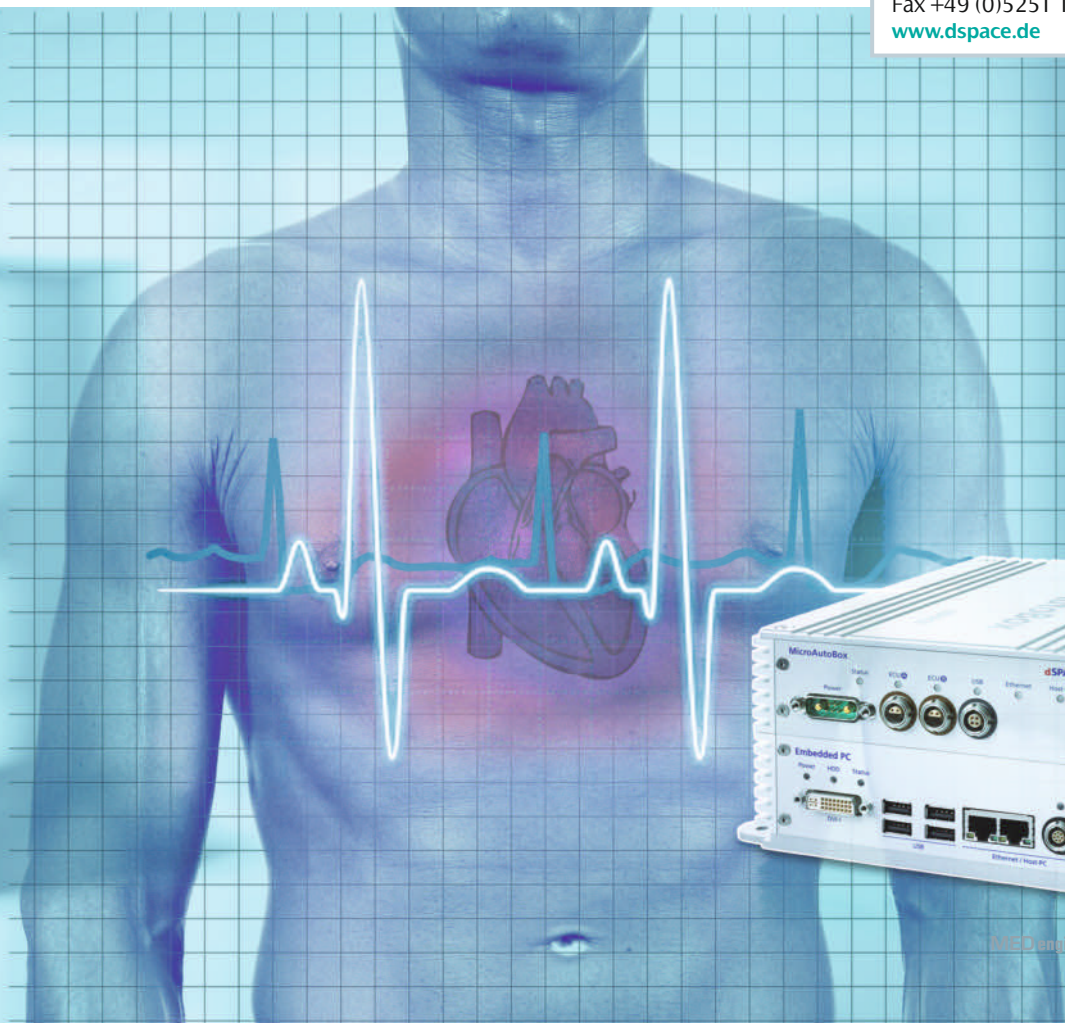
dem erheblichen Zeitaufwand für die manuelle Programmierung eines aussagekräftigen Prototyps.

Beim Erstellen eines Prototyps steht am Anfang die Entscheidung: Make or Buy? Welche Komponenten will oder muss ich selbst entwickeln und welche kaufe ich idealerweise hinzu? Die Antwort ist eigentlich einfach: Zählt die Komponente zu meiner Kernkompetenz, werde ich sie selbst entwickeln. Ist sie kommerziell nicht in der gewünschten Ausprägung verfügbar, muss ich zwischen einer aufwändigen Eigenentwicklung und einer oftmals kostspieligen Beauftragung abwägen. In allen anderen Fällen ist man in der Regel gut beraten, auf kommerzielle Produkte zurückzugreifen, um Kosten, Zeit und Risiko zu minimie-



KONTAKT

dSPACE GmbH
33102 Paderborn
Tel. +49 (0)5251 16380
Fax +49 (0)5251 161980
www.dspace.de





ren und sich auf seine Kernaufgabe konzentrieren zu können. Für die Elektronikentwicklung in den unterschiedlichen Industriezweigen haben sich daher Rapid-Prototyping-Systeme etabliert. Sie bieten die Möglichkeit, ohne viel Hardwareentwicklung und spezielle Programmierkenntnisse mit geringem Zeitaufwand kostengünstig erste Prototypen aufzubauen und unter realen Bedingungen zu testen und zu optimieren. In der Medizintechnik haben Rapid-Prototyping-Systeme von dSPACE bereits ihre Leistungsfähigkeit unter Beweis gestellt. Mit ihnen lassen sich entscheidende Verbesserungen in der Diagnose, Therapie und Reha erreichen. Übliche Aufgaben sind die Positionierung, Akustik, Dosierung, schnelle oder zeitkritische Berechnungsaufgaben sowie die Anlagensteuerung.

Die MicroAutoBox ist ein universell einsetzbares Rapid-Prototyping-System. Aufgrund seiner flexiblen und leistungsfähigen Hardwarearchitektur sowie einer umfangreichen und intuitiven

Platz ist oft Mangelware

Softwareunterstützung ermöglicht es eine schnelle Prototypenentwicklung. Aber was muss man beim Einsatz eines Rapid-Prototyping-Systems und dem Erstellen eines Prototyps beachten? Erst einmal muss ein passender Einbauort im Gerät gefunden werden. Speziell beim Einsatz in mobilen Geräten ist aufgrund von Vibrationen oder

Erschütterungen ein gewisser Grad an Robustheit erforderlich. Zudem ist meist nur wenig Platz verfügbar. Ein möglichst lautloser Betrieb ist häufig eine Grundvoraussetzung und eine aktive Belüftung unerwünscht. Die MicroAutoBox ist daher robust ausgelegt und eignet sich somit für den stationären und den mobilen Einsatz. Ihre kompakte Bauweise und passive Kühlung, kombiniert mit einem erweiterten Temperaturbereich von bis zu +85 °C, ermöglichen den Einsatz auch in Geräten, die nur ein begrenztes Platzangebot zur Verfügung stellen.

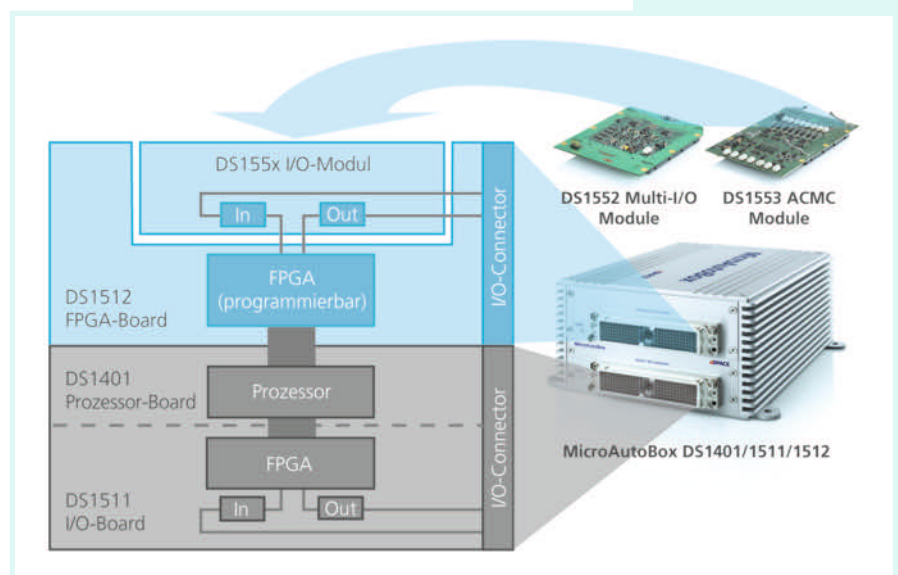
Natürlich erwartet man von einem Rapid-Prototyping-System auch eine gewisse Leistungsfähigkeit. Übliche Aufgabenstellungen in der Medizintechnik sind die Datenerfassung, Steuerung und Regelung so-

wie die Visualisierung. Für jede Aufgabe existiert eine präferierte Art der technischen Umsetzung. So basiert die Datenerfassung und Datenvorverarbeitung idealerweise auf der FPGA-Technik, um ein hohes Maß an Flexibilität und Determinismus bei minimalen Verzögerungszeiten zu erreichen. Steuerungen und Regelungen werden hingegen auf leistungsfähigen Embedded Controllern mit Echtzeitbetriebssystem ausgeführt. Für die Visualisierung selbst nutzt man Windows-basierte PC-Systeme.

Um diese Aufgabenstellungen abdecken zu können, stehen verschiedene Varianten der MicroAutoBox (Bild 1) bereit. Die Basiseinheit für Steuerungs- und Regelungsaufgaben besteht aus einem für Echtzeitanwendungen ausgelegten Embedded Controller ▶

1 Das kompakte und robuste Rapid-Prototyping-System MicroAutoBox ist in verschiedenen Varianten verfügbar. Zum Beispiel mit Erweiterung für die modellbasierte FPGA-Programmierung oder mit Embedded-PC

2 MicroAutoBox-II-Variante DS1401/ 1511/1512 mit programmierbarem FPGA und Steckplatz für I/O-Module





► (PowerPC) mit 900 MHz und einem I/O-Board mit vielen analogen und digitalen I/O-Schnittstellen. Für speziellere I/O-Anforderungen existiert eine Variante, deren Basiseinheit um ein vom Anwender programmierbares FPGA (Xilinx Spartan 6) erweitert wurde (Bild 2). Dadurch können Signalvorverarbeitungen wie Filterungen vorverlagert auf dem FPGA ausgeführt werden, ohne den Embedded Controller zu belasten.

Da das FPGA über einen parallelen, auf niedrige Latenzen optimierten Bus an den Embedded Controller angebunden ist, können die Latenzzeiten für die I/O-Verarbeitung sehr kurz gehalten werden. Die I/O-Wandler sind aus Gründen der Flexibilität auf separate, auf das FPGA-Board steckbare I/O-Module ausgelagert und lassen sich

je nach Anwendung austauschen. Um ein breites Anwendungsspektrum zu adressieren, bietet das universelle DS1552 Multi-I/O-Module eine Vielzahl schnelle und leistungsfähige I/O-Wandler sowie verschiedene serielle Schnittstellen.

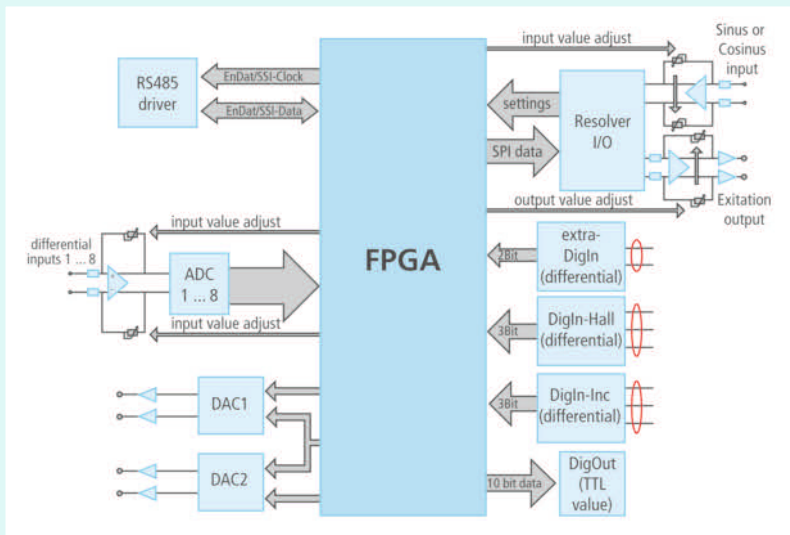
Neben der zuvor beschriebenen universellen Nutzbarkeit kann das FPGA-Board aber auch zur anwendungsspezifischen I/O-Erweiterung verwendet werden. Das AC-Motor-Control-Modul (ACMC) offeriert eine Lösung, die speziell für die Regelung von Elektromotoren ausgelegt ist (Bild 3). Auch dieses Modul verfügt über viele spezifische Schnittstellen für die verschiedenen Arten der Rotor-Positionserfassung

**FPGA-Technik:
flexibel und
deterministisch**

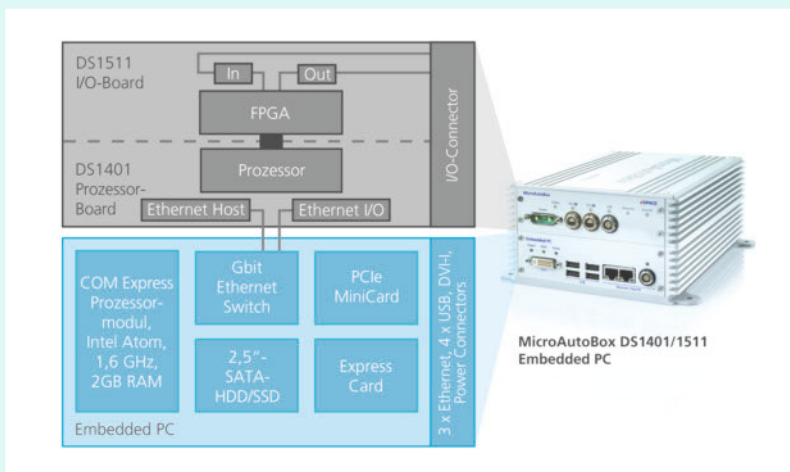
(unter anderem für Hall-Sensoren, Encoder, Resolver, EnDat, SSI) und die Ansteuerung der Leistungsendstufen. In Fällen, in denen eine Anwendung eine darüber hinausgehende Unterstützung spezieller I/O-Schnittstellen erfordert, kann auch eine kundenspezifische Lösung entwickelt und dank des Modulkonzepts einfach integriert werden.

Neben den üblichen I/O-Schnittstellen ist aber auch die Integration des Rapid-Prototyping-Systems in eine vorhandene Infrastruktur wichtig. Dazu gehört unter anderem der Anschluss an Haus-LAN, Monitor, Tastatur, Bedienpanel, Kamerasysteme oder andere existierende Geräte. Häufig erfordern derartige Zusatzgeräte das Vorhandensein eines PCs mit seinen üblichen Schnittstellen sowie ein gängiges Betriebssystem, um entsprechende Treiber und zusätzliche Bediensoftware installieren zu können.

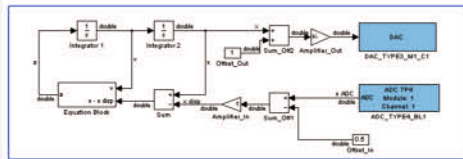
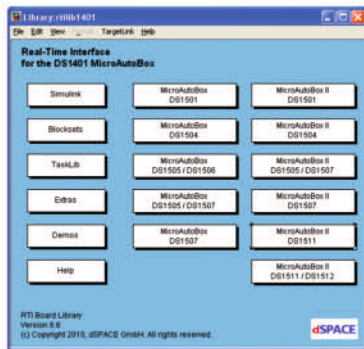
Hier bieten sich die Varianten der MicroAutoBox mit einer integrierten Embedded-PC-Einheit an (Bild 4). Durch die interne Verschaltung sind nur ein Host-Anschluss sowie eine Spannungsversorgung erforderlich, zudem ist ein synchrones Einschalten beim Gerätestart möglich. Die Embedded-PC-Einheit ist mit einem passiv gekühlten Intel-Atom-Prozessor-Modul mit 1,6 GHz Taktfrequenz und 2 GB RAM bestückt. Erweiterungen werden über einen PCIe-Minimicard-Steckplatz sowie einen Express-Card-Einschub ermöglicht. Als Datenträger lassen sich sowohl HDD- als auch SSD-Festplatten im 2,5"-Format integrieren. Zudem stehen frontseitig vier USB-2.0-Schnittstellen und ein DVI-I-Ausgang zur Übertragung von Videodaten zur Verfügung. Als



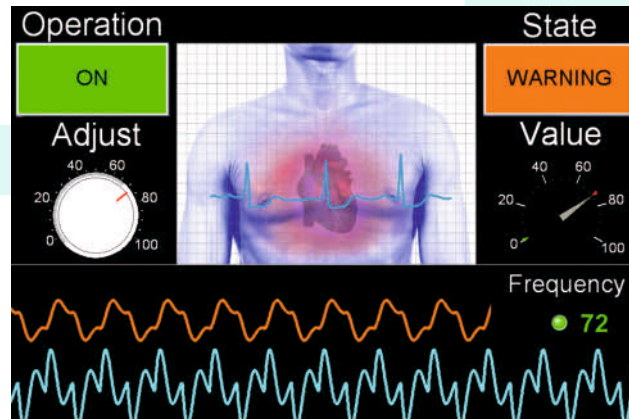
3 Schnittstellen des ACMC-Aufsteckmoduls für die FPGA-Erweiterung zum Anschluss von Elektromotoren



4 MicroAutoBox-Variante mit Embedded-PC-Erweiterung



5 MicroAutoBox-RTI-Blockbibliothek zum Anschluss von I/O-Schnittstellen an ein Simulink-Modell



6 Visualisierung grafischer Instrumente in ControlDesk

Betriebssysteme stehen Windows 7 oder Linux zur Wahl.

Die leistungsfähigste und flexibelste Hardware entfaltet ihren vollen Nutzen nur, wenn sie durch eine umfangreiche und bedienfreundliche Software unterstützt wird. Der modellbasierte Softwareentwurf hat in den letzten Jahren den herkömmlichen Entwicklungsprozess maßgeblich verändert. Anstatt mit Programmiersprachen wie C oder Assembler sind Entwickler heute in der Lage, ihre Mess-, Steuer- und Regelsoftware in einer intuitiven grafischen Umgebung wie MATLAB/Simulink/Stateflow zu erstellen und zu verifizieren. Diese lässt sich später mithilfe einer Implementierungssoftware und automatischer Codegenerierung auf das Rapid-Prototyping-System implementieren und unter realen Bedingungen ausführen.

Die Implementierungssoftware muss zu diesem Zweck Aspekte wie Modell-Code-Generierung, Echtzeit-Framework und I/O-Schnittstellenabdecken. Dabei sollten alle relevanten Punkte möglichst automatisiert und in Form von grafischen Blocksets sowie ohne spürbare manuelle Implementierungsarbeit für den Anwender berücksichtigt sein. Implementierungssoftware wie Real-Time Interface (RTI) von dSPACE bietet eine umfassende und konfigurierbare Blockbibliothek mit I/O-Treibern für Simulink (Bild 5). Zeitintensives manuelles Programmieren von I/O-Treibern ist nicht notwendig.

Schneller am Ziel mit Softwareunterstützung

Um Softwarefunktionalitäten zu optimieren, erlauben dedizierte Experimentierwerkzeuge dem Entwickler die Erfassung von Signalen und das Ändern von Parametern zur Laufzeit. Eine intuitive Arbeitsumgebung kann dabei mithilfe grafischer Instrumente erstellt werden. Experimentierwerkzeuge wie ControlDesk bieten zu diesem Zweck Bibliotheken mit Instrumenten wie Plottern, Drehschaltern, Anzeigen, Schiebern und Knöpfen, die individuell konfiguriert und auf einer Oberfläche platziert werden können (Bild 6). Variablen lassen sich dem jeweiligen Instrument leicht durch Drag & Drop aus einer automatisch generierten Variablenliste zuweisen. Mit diesen Mitteln lassen sich mit nur wenigen Mausklicks verschiedenste aussagekräftige Bedienoberflächen erstellen,

die frühzeitig einen ersten Eindruck zur Bedienbarkeit des Prototyp-Geräts ermöglichen.

Ist der Prototyp fertig, optimiert und wird er für gut befunden, muss sich der Entwickler Gedanken darüber machen, wie er die erstellten Funktionalitäten auf eine Serienplattform übertragen kann. Auch hier hilft die MicroAutoBox, denn aus den Simulink/Stateflow-Modellen lässt sich mithilfe des Seriercode-Generators TargetLink C-Code für beliebige Ziel-Controller generieren. Anteile, die auf dem FPGA entwickelt wurden, können auf andere FPGAs übertragen werden. Gleiches gilt auch für die PC-Software. Jetzt kann die nächste Idee ruhig kommen.

Bibliothek als Basis für die Bedienoberfläche



DIPL.-ING. FRANK MERTENS ist Lead Product Manager für Rapid-Control-Prototyping-Systeme bei dSPACE in Paderborn. fmertens@dspace.de



DIPL.-ING. HOLGER ROSS ist Product Engineer für Rapid-Control-Prototyping-Systeme. hröss@dspace.de

