moving forward

antrimon group

Software-Entwicklung und Bus-Systeme

Die Software ist massgeblich für die Funktionalität mechatronischer Systeme. Doch kauft man diese besser von der Stange oder entwickelt diese lieber in Eigenregie oder in Kooperation mit einem Dienstleister?

Behalten Sie den Überblick über die unterschiedlichen Technologien. Mit unseren Beiträgen möchten wir Sie dabei unterstützen, bei der Evaluation schnell und effizient vorzugehen und Ihre Kosten und Aufwände zu optimieren.

Die Entwicklung der mobilen Dosiereinheit bei der Karl Schnurr AG* läuft auf Hochtouren. Während die für die Motoransteuerung verantwortlichen Ingenieure sich an der Frage reiben, ob besser Standard-Elektronik oder aber eine passgenaue Lösung zum Tragen kommen soll, generiert die Software-Abteilung bereits fleissig Codes. Und von diesen braucht es viel! Schliesslich soll die Infusionspumpe im Vergleich zum Vorgängermodell alle gängigen Schnittstellenstandards integrieren, damit diese unter anderem an externe Patientendatenmanagementsysteme (PDMS) sowie in klinische Netzwerke eingebunden werden kann.

Insbesondere mit letztem Punkt ist ein starkes Verkaufsargument. So soll sich die Dosiereinheit mithilfe einer speziellen Software von jedem Rechner innerhalb des Krankenhausnetzwerks aus über einen Web-Browser bedienen lassen. Diese Fähigkeit soll es Spitälern ermöglichen, die Patientenüberwachung über alle Ebenen und Stockwerke hinweg zu zentralisieren.

Bedeutung von Librarys

Nun ist es nicht so, dass die Software-Ingenieure bei der Karl Schnurr AG* deutlich entscheidungsfreudiger als ihre Kollegen aus der Antriebsabteilung sind. Allerdings stand im Falle der Dosiereinheit gleich von Beginn an fest, dass diese ihre Funktionalität nicht durch Standard-Software, sondern durch eigenentwickelten Code erhalten soll. Von den Vorgängermodellen der mobilen Infusionspumpe sowie diversen anderen Projekten sind umfangreiche Bibliotheken vorhanden, die sich mit relativ geringem Aufwand implementieren lassen.

Für uns eine nachvollziehbare Entscheidung: Die Karl Schnurr AG* verfügt bereits über Librarys, die sie einfach adaptieren kann. Zudem fallen auch bei der Verwendung von Standard-Software Kosten für die Implementierung und die Lizenz an.

Daraus nun abzuleiten, dass eine Eigenentwicklung günstiger ist, wäre jedoch der falsche Schluss! Letztlich ist es immer eine Abwägungsfrage, bei der unter anderem die Stückzahlen und die Verwendung bereits vorhandener Librarys eine Rolle spielten. Da die Software-Entwickler bei der Karl Schnurr AG* ausserdem klar zwischen hardwareabhängigen und hardwareunabhängigen Modulen trennen, können sie vorhandenen Code ohne grossen Aufwand adaptieren.

Ein weiterer Vorteil dieser sauberen Trennung: Da bei einer Abkündigung von Produkten oder Komponenten nicht immer eine 100-prozentige Kompatibilität gewährleistet ist, muss beim modularen Software-Aufbau nur der entsprechende Treiber ersetzt werden.

SOFTWARE

ANTRIMON Group AG | Gotthardstrasse 3 | CH-5630 Muri AG | Tel. +41 58 330 26 00 | info@antrimon.com | antrimon.com

moving forward

antrimon group

Software-Entwicklung und Bus-Systeme

Die Software ist massgeblich für die Funktionalität mechatronischer Systeme. Doch kauft man diese besser von der Stange oder entwickelt diese lieber in Eigenregie oder in Kooperation mit einem Dienstleister?

Behalten Sie den Überblick über die unterschiedlichen Technologien. Mit unseren Beiträgen möchten wir Sie dabei unterstützen, bei der Evaluation schnell und effizient vorzugehen und Ihre Kosten und Aufwände zu optimieren.

Schnittstellendefinition
Die Möglichkeit, die Dosiereinheit in externe PDMS und klinische Netzwerke einbinden zu können, bedarf einer sauberen Schnittstellendefinition. Dabei stellt sich zunächst die Frage, welche Anforderungen es an die Kommunikation zwischen den einzelnen Modulen braucht? Müsste die mobile Infusionspumpe beispielsweise mehrere Achsen im Mikrosekunden-Takt synchronisieren, kämen die Entwickler an einem echtzeitfähigen Bussystem wie EtherCAT oder Powerlink nicht vorbei. Da jedoch lediglich über die Kolbenbewegung die verabreichte Dosis ermittelt und im Display dargestellt wird, genügt eine einfache CAN-Anbindung.

Vorteil dieses Lösungsansatzes: Die meisten Mikrocontroller integrieren standardmässig die für die CAN-Kommunikation benötigten Treiber und Funktionalitäten.

Offene und geschlossene Bussysteme
Die Geschwindigkeit ist ein Kriterium bei der Wahl des Busses. Ein anderes ist, ob die Kommunikation über ein offenes oder ein geschlossenes Bussystem erfolgen soll? In manchen Anwendungen ist das Bussystem durch das Design vorgegeben. Ist dieser jedoch nicht festgelegt, gibt es verschiedene Kriterien, diesen zu bestimmen. Eines kann beispielsweise der Overhead eines Standard-Bussystems sein, der zu Lasten des Prozessors geht. Ein anderes Kriterium ist der Know-how-Schutz. «Ein proprietäres System macht ein Reverse-Engineering schwierig bis unmöglich.» beschreibt Mo Aakti die Vorzüge eines geschlossenen Bussystems.

Software-Schnittstellen
Softwareseitige Datenschnittstellen sind logische Berührungspunkte in einem Softwaresystem und regeln den Austausch von Kommandos und Daten zwischen verschiedenen Prozessen und Komponenten. Bei den Software-Schnittstellen erfolgt dabei grundsätzliche Unterscheidung:

Datenorientierte Schnittstellen – Diese Schnittstellen werden ausschliesslich zur Kommunikation benutzt und tauschen die Informationen zwischen den beteiligten Systemen aus. Ein Beispiel hierfür wären Adressübergaben mit Verweis auf zu verwendende Daten/Informationen bei Aufruf von Unterprogrammen.

Funktionale Schnittstellen – Diese Schnittstellen führen eine bestimmte Funktionalität aus, um die primär beteiligten Systemteile zu synchronisieren oder zu unterstützen. Ein Beispiel hierfür wären Druckertreiber.

SOFTWARE

ANTRIMON Group AG | Gotthardstrasse 3 | CH-5630 Muri AG | Tel. +41 58 330 26 00 | info@antrimon.com | antrimon.com