



So sparsam kann PROFINET sein.

Klein aber konform

Die Anforderungen an Ressourcen hinsichtlich Prozessor und Speicher sind bei PROFINET höher als beim klassischen Feldbus. Mit den richtigen Werkzeugen lässt sich PROFINET jedoch auch auf sehr kleinen Prozessoren erschließen.

Die standardisierte Spezifikation von PROFINET erlaubt die individuelle Anpassung von Komponenten eines Stacks auf die jeweiligen Bedürfnisse des Zielgeräts. Hier helfen die jeweiligen Realtime Classes sowie die unterschiedlichen Conformance Classes, je nach gewünschter Funktionalität und den gestellten Anforderungen auch kleine und kostengünstige PROFINET-Implementierungen zu realisieren. Der Industrial Ethernet und auch PROFINET anhaftende Ruf, äußerst ressourcenhungrig zu sein, ist somit nicht gerechtfertigt. Lässt man die einzelnen Elemente der Spezifikation Revue passieren, so zeigt sich, dass sich ein PROFINET-Stack z. B. für ein einfaches Ein-Port-Gerät auch in 96kByte RAM unterbringen lässt. Zunächst sollte der Stack modular aufgebaut sein, damit einzelne nicht zwingend benötigte Features zur Compile-Zeit deaktivierbar sind. Dies gilt insbesondere für komplexe und umfangreiche Features SNMP oder IRT. Weiterhin sollte der Stack allgemein auf Embedded System hin optimiert sein. Dies schließt eine effiziente Speicherverwaltung und die platzsparende Speicherung von Objektstrukturen mit ein.

Eine weitere Überlegung gilt den Application Relationship (AR, Applikationsbeziehungen). Hier sieht die Spezifikation in der Regel entweder den Controller AR (IOC-AR) für die Übertragung zyklischer Ein-/Ausgangsdaten zwischen IO-Controller und IO-Device vor oder den SupervisorAR (IOS-AR) für den Datenaustausch zwischen IO-Supervisor und IO-Device. In der Regel genügt es, Datenstrukturen für diese zwei AR statisch vorzuhalten. Die Daten hierfür sind überschaubar, es ist lediglich darauf zu achten, dass der Speicherbereich für einige Strings (Initiator Name, AR Identifier etc.) ausreichend groß entsprechend der Spezifikation gewählt wird.

Ein weiterer interessanter Punkt bezüglich Speicherverbrauch stellt eine optimierte Implementierung der Remote Procedure Calls (RPC) dar. PROFINET verwendet eine RPC-Kommunikation entsprechend der Spezifikation DCE 1.1 der Open Group. Da pro AR in der Regel nur eine RPC-Session aufgebaut wird, genügt auch hier das Vorhalten von Daten für zwei Sessions. Für die Implementierung der zyklischen IO-Daten (aber nicht nur hier) sollte das Grundprinzip eines Zero-Copy-Stacks gelten. Dabei wird bei der Verarbeitung von Daten weitestgehend auf das Kopieren von Datenpuffern verzichtet.

PROFINET verwendet zur automatischen Topologieerkennung von Automatisierungnetzwerken das Link Layer Discovery Protokoll (LLDP). Dies sendet in regelmäßigen Abständen Informationen über seinen Netzwerkport an sein Gegenüber. Auf diese Weise kennen alle Geräte im Netzwerk ihren jeweils direkt angeschlossenen Nachbarn. Die Abfrage der Informationen erfolgt über SNMP von einer zentralen Managementstation aus. In dieser Betrachtung von PROFINET wurde von einem einfachen Ein-Port-Gerät ausgegangen. Es stellt somit immer den Endpunkt sowohl in einer Linien- als auch in einer Sterntopo-

logie dar. Daher genügt es für das Gerät, lediglich Informationen über sich selbst per LLDP zu versenden. Das Sammeln von Information von gegenüberliegenden Geräten ist somit nicht nötig, auch auf die Implementierung eines SNMP-Stacks kann verzichtet werden.

Fazit

PROFINET lässt sich durchaus auch auf kleinen Prozessoren integrieren. Mit dem entsprechenden Protokoll und Konfigurationswerkzeugen bzw. Designtools erschließen sich PROFINET somit noch weitere Anwendungsgebiete, womit diese Technologie eine sichere Investition in die Zukunft darstellt.

Marcus Tangermann, port GmbH