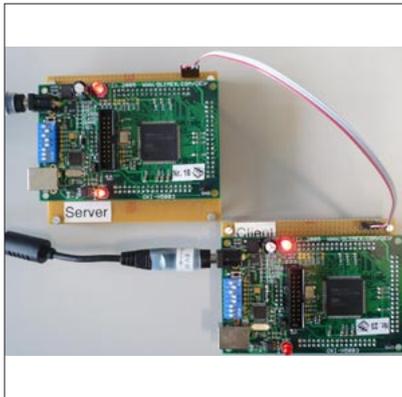




Bernhard  
Willi

## Generisches Protokoll für Embedded Systems

Diplomand	Bernhard Willi
Examinator	Prof. Reto Bonderer
Experte	Theo Scheidegger, swens GmbH, Schänis SG
Themengebiet	Embedded Systems
Projektpartner	Mettler-Toledo GmbH, Greifensee ZH



Implementation auf zwei ARM7-Prozessoren

Die Firma Mettler-Toledo entwickelt verschiedene Wägesysteme für unterschiedliche Anwendungsbereiche. Ein solches Wägesystem besteht aus zwei Teilen, aus einem Bedienpanel und aus einem Wägesensor, wobei beide Teilsysteme mit einem eigenen Mikroprozessor arbeiten. Je nach Aufgabenbereich der Waage wird ein Bedienpanel mit einem geeigneten Wägesensor kombiniert. Das Bedienpanel sollte nun mit dem Wägesensor respektive dessen Mikroprozessor kommunizieren können. Die Kommunikation funktioniert nach dem Client-(Panel)-Server-(Sensor)-Prinzip. Dies bedeutet, dass der Wägesensor nur dann kom-

muniziert, wenn er vom Panel dazu aufgefordert wird. Die Kommunikation zwischen diesen beiden Teilsystemen erfolgt über eine Schnittstelle (RS-232, RS-485, USB, Ethernet, CAN, ...). Zurzeit wird für diese Kommunikation ein binäres Protokoll eingesetzt, welches ganze Objekte überträgt. Diese Art von Kommunikation setzt aber voraus, dass das Client- und das Server-System auf einer identischen Plattform und mit der gleichen Programmiersprache arbeiten. Gerne würde man die beiden Systeme unabhängig voneinander herstellen und so für jedes System die ideale Plattform und Programmiersprache verwenden.

Protokolle, welche eine Kommunikation zwischen verschiedenen Plattformen ermöglichen, werden Middleware genannt. Für Webanwendungen gibt es solche Middleware-Protokolle, diese sind aber sehr ressourcenintensiv und können deshalb nur schlecht auf Embedded-System-Plattformen eingesetzt werden.

Ziel dieser Diplomarbeit ist es, die bereits existierenden Protokolle zu analysieren und ein Protokoll zu entwickeln, mit welchem zwei unterschiedliche Embedded Systems miteinander kommunizieren und gegenseitig Funktionen aufrufen können. Es soll möglich sein, wie im Baukastensystem verschiedene Bedienpanels mit verschiedenen Wägesensoren ohne grosse Anpassungen am Programmcode zu kombinieren.

Die Grundidee dieses Protokolls besteht darin, im Clientsystem Funktionen des entfernten

Serversystems aufzurufen. Der entfernte Funktionsaufruf wird in diesem Protokoll folgendermassen umgesetzt: Die Schnittstelle der entfernten Funktion wird lokal auf dem Client-system dargestellt und kann von diesem auch aufgerufen werden. Beim Aufrufen dieser lokal dargestellten, aber in Wirklichkeit entfernten Funktion werden die Parameter (Name der Funktion und Attribute) in eine Nachricht verpackt und an den entfernten Prozessor gesendet. Dieser empfängt die Nachricht, interpretiert sie und ruft nun mit den gesendeten Parametern die richtige Funktion auf. Die Antwort der ausgeführten Funktion wird wiederum zu einer Nachricht verpackt und zurückgesendet. Dort wird diese wieder ausgepackt und als Antwort der lokalen Schnittstelle übergeben. Ausser der längeren Bearbeitungszeit soll es keinen Unterschied zwischen dem Aufrufen einer lokalen und einer entfernten Prozedur geben.