Softwareoszilloskop zum Debugging von Multiapplikationssystemen unter Linux

Studiengang: BSc in Elektrotechnik und Informationstechnologie | Vertiefung: Embedded Systems
Betreuer: Prof. Dr. Elham Firouzi
Expert: Daniel Rickli (Signal AG)
Industriepartner: Source Engineers, Bern

In Industrieapplikationen werden oft Embedded Linux Systeme mit mehreren Applikationen eingesetzt. Einzelne Applikationen zu debuggen ist gut möglich, jedoch ist das Debugging auf der System-Ebene komplex, da die Daten aus den verschiedenen Applikationen nicht zentral zur Verfügung stehen. Hier schafft das Softwareoszilloskop Abhilfe.

In dieser Arbeit wird auf einem Embedded Linux System eine Multiapplikationslandschaft mit der Server-Applikation **se-scope-server** und mehreren Client-Applikationen, den **se-scope-clients** gestaltet. Die Clients senden ihre Daten mittels Interprozesskommunikation an den Server, welcher diese anhand der C-Library **se-scope-lib** serialisiert und über TCP an den PC schickt. Die Daten werden in der Desktop-Applikation **se-scope-gui** empfangen, deserialisiert und visualisiert. Sowohl die Server- wie auch die Client-Anwendungen werden im Rahmen dieser Arbeit entwickelt. Das GUI wird so erweitert, dass die Kommunikation neu über TCP läuft.

Interprozesskommunikation

Eine Herausforderung der Arbeit ist die Dynamik der Clients: Der Server soll sich mit beliebig vielen Client-Applikationen verbinden können. Der Server muss erkennen, wenn ein neuer Client gestartet wird, welcher ebenfalls Debugging-Daten zu senden hat. Dazu werden **Unix Domain Sockets** eingesetzt. Mit Hilfe einer virtuellen Datei erkennt der Server, wenn ein Client gestartet wird und öffnet den Kommunikationskanal. Der Server koordiniert die Clients und verpackt deren Daten mithilfe der C-Library. Als Protokoll zwischen den Prozessen wird **Google Protocol Buffer** verwendet, mit den Vorteilen, dass die Kommunikation so einen kleinen Overhead hat und die Ergänzung weiterer Clients in anderen Programmiersprachen vereinfacht wird.

Grafische Benutzeroberfläche

Das GUI ist auf dem **Electron**-Framework basiert, welches erlaubt, Desktop-Apps mit Web-Technologien wie HTML, Javascript, css und auch NodeJS zu entwickeln. Neu ist die Option, dass die Kommunikation zwischen dem GUI und dem Embedded System statt über UART über TCP stattfindet, da viele Embedded Linux Systeme nur über das Netzwerk erreichbar sind.

ADC-Client

Um den Praxiswert zu zeigen, wird ein Client erstellt, welcher über SPI aus einem 24-Bit A/D-Wandler Analogwerte ausliest.

Anselm Fuhrer
anselm.fuhrer@gmail.com