

Wireless CAN-Debugger

Technische Informatik / Prof. Roger Weber
Experte: Dipl. El.-Ing. HTL Dominique Renevey, Siemens Schweiz AG

In der Industrie werden zur Ansteuerung von Aktoren und Sensoren zunehmend Bus-Systeme eingesetzt. Speziell für die Entwicklung der Hardwareknoten von bewegten Objekten wie z. B. Roboter, fehlt dabei oft eine komfortable Möglichkeit diese ohne die Hauptsteuerung zu testen. Ziel dieser Thesis ist es, ein Gerät zu entwickeln womit CAN-Nachrichten empfangen, angezeigt und gesendet werden können. Über eine Wireless-Verbindung sollen mehrere Benutzer in einem Web-Frontend auf den Bus zugreifen können.



Philipp Haslebacher
1985
079 440 55 35
philipp.haslebacher [at] gmx.ch

Motivation

Während zwei Semesterarbeiten wurde in einem siebenköpfigen Team ein Roboter für den Eurobot-Wettbewerb entwickelt. Die Hauptsteuerung bildete ein Industrierechner, welcher über den CAN-Bus mit zwei Hardwareknoten Sensordaten anforderte und Aktoren ansteuerte. Damit parallel entwickelt werden konnte, wurde die Arbeit in drei Teile zerlegt: Mechanik, Mechatronik und Informatik. Die Schwierigkeit dabei war, dass zum Testen der Funktionalität der CAN-Knoten immer eine lauffähige Softwareversion auf der Hauptsteuerung benötigt wurde. Daraus entstand der Wunsch nach einem Gerät, mit welchem mehrere Benutzer unabhängig voneinander auf den CAN-Bus zugreifen können.

Anforderungen

Der Benutzer soll den CAN-Debugger über ein einfaches Interface direkt auf dem Debugger oder via WLAN in einem Webfrontend bedienen können. Er bietet Funktionen zum senden, empfangen und aufzeichnen von Nachrichten. Um Abläufe zu testen, sollen zudem Sequenzen von Nachrichten programmiert werden können.

Testaufbau

Das Herzstück des CAN-Debuggers bildet ein Single-Board Computer mit einem PXA270 Prozessor. Als Betriebssystem verwenden wir Linux. Alle von uns benötigten Schnittstellen wie USB, Ethernet, CAN, TFT, sind auf dem Board integriert.

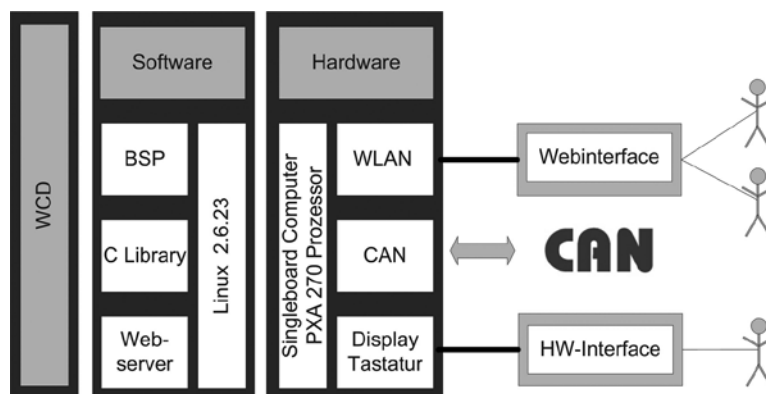
Implementierung

Der Wireless-CAN-Debugger besteht aus mehreren Komponenten, welche wir selber entwickelt und teilweise übernommen haben. Dabei greifen wir auf den bestehenden Lighttpd-Webserver und die FastCGI-Library zurück, welche uns die Entwicklung des Webfrontends deutlich vereinfachen. Weiter konnten wir auf bestehende Software zum Kommunizieren über den CAN-Bus, das Ansteuern vom Display und das Auslesen der Matrixtastatur zurückgreifen. Zu den selbst entwickelten Teilen gehören der CAN-Debug-Daemon, das CAN-Debug-Interface, das CAN-Debug-Web-Interface sowie das CAN-Debug-Communication-Protocol über welches unsere Komponenten kommunizieren.

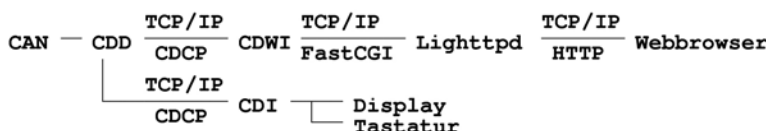
Der CAN-Debug-Daemon arbeitet direkt mit dem CAN-Bus zusammen und lässt sich über TCP/IP mit dem CAN-Debug-Communication-Protocol steuern. Für die direkte Bedienung des Wireless-CAN-Debugger bildet das CAN-Debug-Interface mit Tastatur und Display eine Schnittstelle zum Anwender. Daneben ermöglicht das CAN-Debug-Web-Interface die Bedienung mittels Webbrowser. Dabei ist das CAN-Debug-Web-Interface durch die FastCGI-Library mit dem Webserver verbunden.



Moritz Kobel
1983
079 794 50 40
moritz [at] kobelnet.ch



Aufbau



Kommunikation