LumWeb

Universelle Maschinensteuerung mit Webinterface

HTL Paul-Hahn

Informationstechnologie

Diplomarbeit von:

Martin Anzinger

Florian Hahn

Betreut von:

Prof. Mag. Herbert Jachs

In Kooperation mit:

Hainzl Industriesysteme

Linz, 14. April 2010

Version: TMG 0.1

Kurzfassung

Das Ziel von LumWeb ist, die Vereinfachung der Userinterface-Erstellung für Maschinensteuerungsinterfaces, für Heizungen und andere Maschinen.

Erreicht wird das Ziel durch das Zusammenspiel von Embedded Webserver und Embedded WebClient auf einem Microcontroller System.

Das User Interface wird als HTML Seite mit SSI Tags definiert. Die Bedienseiten sind auf einem Serversystem gespeichert. Das Serversystem ist zuständig für die Kommunikation mit der Maschine und mit dem Benutzer. Die Seiten werden über HTTP an die Clients verteilt. Auf PC Systemen übernimmt der Browser die Darstellung. Auf uC Systemen über nimmt der Webclient die Darstellung. Dazu wertet er spezielle HTML Kommentare aus, die für die SSI Tags eingefügt werden.

Um eine Bedienseite zu erstellen werden in den HTML Code Bedienelemente in Form von SSI Tags eingebaut, zum Beispiel der Tag NumberInputField für die Eingabe einer Zahl. Zusätzlich muss eine ID angegeben werden, die das Element mit eine Einstellung der Maschine verbindet. Der ComTask wertet diese ID aus und liefert den entsprechenden Wert. Der Webserver fügt beim Auswerten der Tags außerdem spezielle HTML Kommentare ein, die der Webclient zur Darstellung auswertet. Der Client kann nur solche Tags darstellen, für die er Darstellungsmethoden implementiert.

Abstract

The Luminary Webinterface or LumWeb is a simple solution for making universal user interfaces. These user interfaces can be designed for different heatings or other machines that have any wired interface to access the settings. The major point is that the interface information are broadcasted over the HTTP protocol as a web page with special SSI tags. It's also possible that the user interface can be accessed by any web browser. The client on the micro controller board is also a embedded web browser with a graphical touchscreen interface.

This solution is more efficient because only one source for data and design is needed. It's also possible to do the settings from different user interfaces.

The functional principle is realized by an embedded web server. If a request comes to this server, it fetches the correct design file from the microSD card and begins to parse it. This file is mostly HTML code. In this HTML code are also some SSI tags. The server system is able to parse this tags and fetches the current values from the communication task. This communication task knows how to get the data from the machine and sends it back to the web server. The web server adds the HTML code for this tag and a special tag in a HTML comment. So the second tag is ignored by any web browser. Only the embedded web browser on the micro controller Board is able to parse this tags and shows it content on the touchscreen.

All the tags from the tag-library can be added, removed or expanded dynamically only by writing or editing single source-code file on the system.

Inhaltsverzeichnis

1. Externe Komponenten

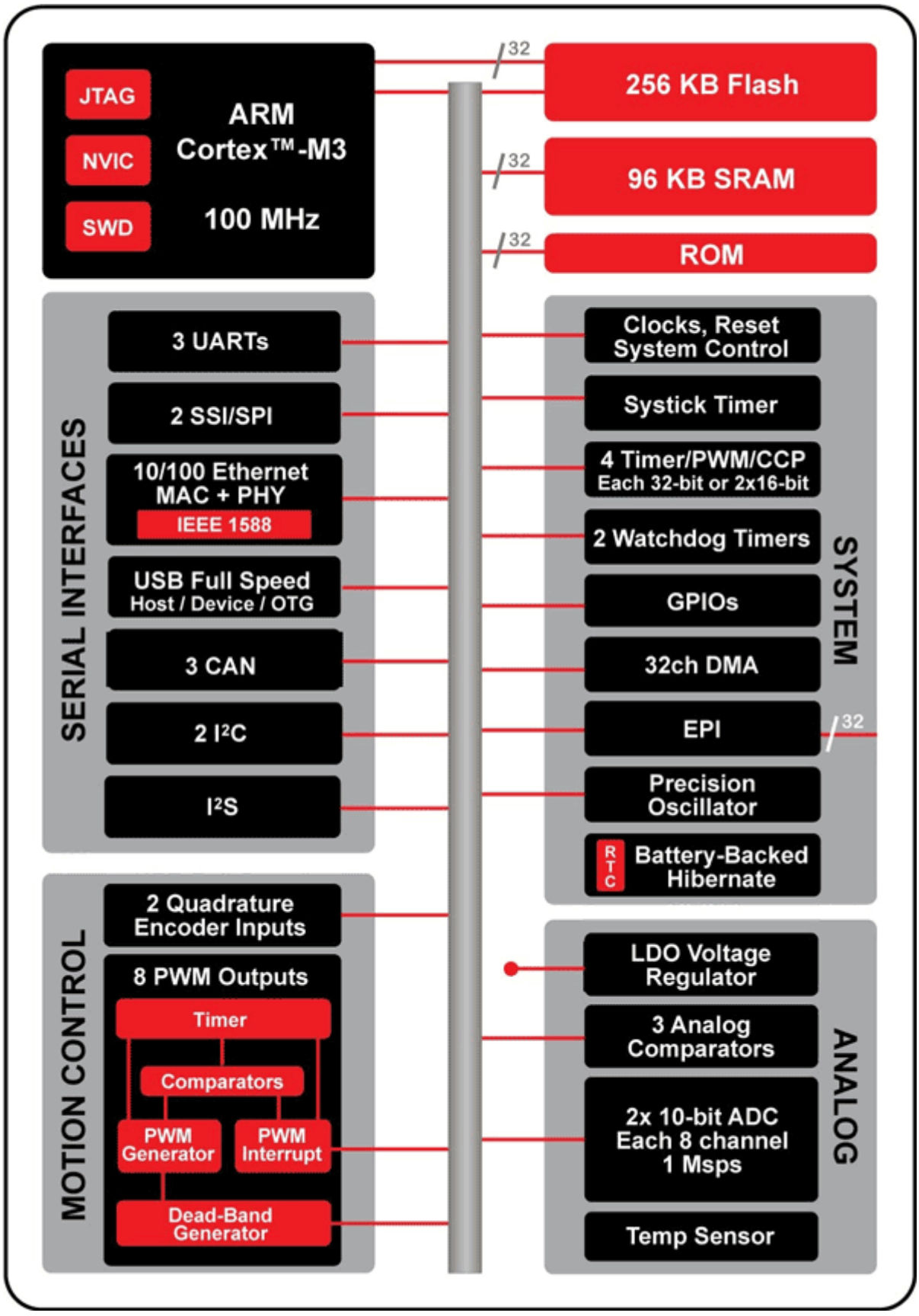
Hardwareumgebung

Als Entwicklungssystem wurde ein LuminaryMicro DK-LM3S9B96 verwendent. Im Zuge der Diplomarbeit wurde zuerst das an der Schule verwendete Luminary Board verwendet, welches dann aber durch das DK-LM3S9B96 ersetzt wurde, das uns von der Firma Hainzl zur Verfügung gestellt worden ist und mit einem Touchscreen ein praxis-näheres Entwicklungssystem bietet.

Wichtige Eckdaten:[1]

- 3.5“ Farb-LCD
    - Auflösung: 320 x 240
    - Resistiver Touchscreen
- 80 MHz LM3S9B9 uC
    - 256kB Flash
    - 96kB Sram
    - integrierter Ethernet Controller
    - Card Reader
    - ARM Cortex 3M CPU

1 Luminary Micro – April 2010, http://www.luminarymicro.com/products/dk-lm3s9b96.html



Luminary Micro Board,

http://www.luminarymicro.com/images/stories/stellarisblockdiagram_large.gif

Softwareumgebung

FreeRTOS

Als Betriebssystem wird das in C geschriebene Open Source Realtime Operatring System FreeRTOS verwendet, da es lizenzfrei verwendet werden kann und als sehr stabil gilt. Als Entwicklungssystem wurde eine ARM Cortext 3M CPU verwendet, FreeRTOS ist aber für viele verschiedene Hardwareplattformen verfügbar:

- Mikrocontroller mit ARM-Cortex-M3-Architektur
- Mikrocontroller mit ARM7-Architektur
- Atmel AVR und Atmel AVR32
- Freescale Semiconductor HCS12-Familie und Coldfire V2
- Xilinx MicroBlaze und PowerPC PPC405
- Texas Instruments MSP430
- Microchip Technology PIC18, PIC24, dsPIC, PIC32
- Renesas H8/S SuperH
- Fujitsu MB91460 32bit und MB96340 16bit
- NEC V850ES 32bit und 78K0R 16bit

Um Netzwerkanfragen, die grafische Darstellung am Display und die Ansteuerung der Maschine quasi-parallel realisieren zu können wird eine Multitaskingumgebung benötigt. FreeRTOS bietet diese Umgebung in Kombination mit Queues zur Interprocesskommunikation.

Funktionsweise[2]

FreeRTOS ist ein Open-Source-Echtzeitbetriebssystem für eingebettete Systeme. Es wurde auf verschiedene Mikrocontroller portiert. Das unter einer leicht modifizierten GPL stehende Microkernelsystem wird zurzeit in der Version 6.0.2 angeboten. Durch die leichte Modifikation der GPL braucht unter FreeRTOS laufende Applikationssoftware nicht auch unter die GPL gestellt zu werden, wodurch sich das Betriebssystem auch für kommerzielle Projekte eignet. Im Gegensatz zu kommerziellen Systemen, wie embOS und CMX-RTX, fehlen FreeRTOS allerdings wichtige Synchronisationsmechanismen wie "Event Flags". Mutexe sind in neueren Versionen vorhanden.

Um eine gute Wartbarkeit zu gewährleisten, wird FreeRTOS weitestgehend in C entwickelt, lediglich wenige Funktionen sind in Assembler realisiert. Der Scheduler ist konfigurierbar, so dass

2 Wikipedia – FreeRTOS, April 2010, http://de.wikipedia.org/w/index.php?title=FreeRTOS&oldid=69842333

präemptiver und kooperativer Betrieb möglich ist. Das Betriebssystem unterstützt seit der Version 4 zwei verschiedene Taskklassen. "Echte" Tasks und Koroutinen, denen nur wenig Speicher zur Verfügung steht.

Multitasking

Der Begriff **Multitasking** bezeichnet die Fähigkeit eines Betriebssystems, mehrere Aufgaben (Tasks) nebenläufig auszuführen. Dabei werden die verschiedenen Prozesse in so kurzen Abständen immer abwechselnd aktiviert, dass der Eindruck der Gleichzeitigkeit entsteht.

Die Nutzung von Multitasking bietet folgende Vorteile:[3]

- Durch die Multitasking und Intertask Kommunikation können komplexe Anwendungen in kleiner, besser wartbare Tasks zerlegt werden
- Die Zerlegung von Anwendungen vereinfacht das Testen, die Wiederverwendbarkeit und die Arbeitsteilung in Teams
- Komplexe Timing Funktionen können von der Anwendung ins Betriebssystem ausgelagert werden

FreeRTOS bietet 2 Möglichkeiten, um Multitasking zu verwenden, Tasks und Co-routines[4].

Eine Real Time Anwendung, die FreeRTOS verwendet kann als Sammlung von seperaten Tasks realisiert werden. Jeder Task wird unabhängig von anderen Tasks oder dem Scheduler[5] in seinem eigenen Kontext[6] ausgeführt. Es kann zu jedem Zeitpunkt nur ein Task ausgeführt werden. Die Zeiteinteilung wird vom Scheduler gemanaget.

Vor- und Nachteile von Tasks:

- einfach
- keine Nutzungsbeschränkungen
- voll priorisierbar

3 RTOS Fundamentals – Multitasking, April 2010, http://www.freertos.org/implementation/index.html

4 FreeRTOS Dokumentation – Tasks and Co-routines, April 2010, http://www.freertos.org/taskandcr.html

5 Ein **Prozess-Scheduler** ist ein Steuerprogramm, das die die zeitliche Ausführung mehrerer Prozesse in Betriebssystemen regelt

6 Als **Prozesskontext** bezeichnet man die gesamte Information, die für den Ablauf und die Verwaltung von Prozessen von Bedeutung ist.

- jeder Task hat einen eigenen Stack → höhere RAM-Nutzung

Auf Co-routines wird hier nicht eingegangen, da LumWeb ausschließlich Tasks verwendet.

Inter-task Kommuniktaion[7]

LumWeb verwendet zur Task Kommunikation ausschließlich Queues, darum wird hier nur auf diese Methode eingegangen.

Queues

Queues stellen die primäre Form der Inter-Task Kommunikation dar. Sie werden zum Senden und Empfangen von Nachrichten zwischen Tasks und zwischen Interrupts[8] und Tasks. In den meisten Fällen werden sie als thread-sichere FIFO[9] Buffer verwendet.
Queues können Elemente mit „fixer“ Größe enthalten. Beim Erstellen einer Queue muss der Datentyp und die maximale Anzahl der Elemente angegeben werden.
Elemente werden als Kopie eingefügt, nicht als Referenz, darum sollten die Elemente nicht zu groß sein, um den Kopieraufwand zu minimieren.
Um große Elemente in Queues einzufügen sollten Zeiger verwendet werden. Dabei muss aber sichergestellt werden, dass klar definiert ist, welchem Task die Daten „besitzt“.

7 FreeRTOS Dokumentation - Inter-task Communication - http://www.freertos.org/Inter-Task-Communication.html

8 Unter **Interrupt** versteht man die kurzfristige Unterbrechung eines Programms, um eine andere, meist kurze, aber zeitkritische Verarbeitung durchzuführen.

9 **First In – First Out** (engl. etwa „Erster rein – Erster raus“)

LWIP

Als Netzwerkstack wird LWIP verwendet, da dieser gegenüber uIP viele Vorteile bietet. Zuerst wurde der uIP Webserver verwendet. Im Zuge des Boardwechsels wurde auch der Wechsel zu LWIP vollzogen. Zu den Vorteilen zählen DHCP Unterstützung und die Möglichkeit mehrere TCP Ports zu verwenden. Dadurch können Web Client und Web Server auf dem selben System implementiert werden.

lwIP (*lightweight IP*) ist ein weit verbreiteter Open Source TCP/IP stack für Microcontroller.

Der Fokus der lwIP TCP/IP Implementierung ist ein reduzierter Ressourcenverbrauch, bei voll Umfang von TCP. Dadurch eignet sich lwIP für Embedded Systems mit wenigen 10 kB RAM Speicher und ca. 40 kB Code Speicher.

LWIP Funktionen[10]

LwIP bietet folgende Funktionen, die von LumWeb Verwendeten werden werden hervorgehoben:

- **IP** (Internet Protocol) inklusive Packet Forwarding über mehrere Network Interfaces
- **ICMP** (Internet Control Message Protocol) für Netzwerkwartung und Debugging
- IGMP (Internet Group Management Protocol) für Multicast Traffic Management
- **UDP** (User Datagram Protocol)
- **TCP** (Transmission Control Protocol)
- Raw/native API für verbesserte Performance
- Optional Berkeley-like socket API
- **DNS** (Domain names resolver)
- SNMP (Simple Network Management Protocol)
- **DHCP** (Dynamic Host Configuration Protocol)
- **AUTOIP** (für IPv4, konform mit RFC 3927)
- PPP (Point-to-Point Protocol)
- **ARP** (Address Resolution Protocol) für Ethernet

10 LWIP Homepage – April 2010, http://lwip.wikia.com/wiki/LwIP_Wiki

2. Lumweb

Lumweb besteht aus 3 Komponenten:

- Webserver:
- Webclient: stellt HTML Interface Seiten auf Touchscreen dar
- ComTask: verwaltet die Kommunikation zwischen Webserver und der Maschine.

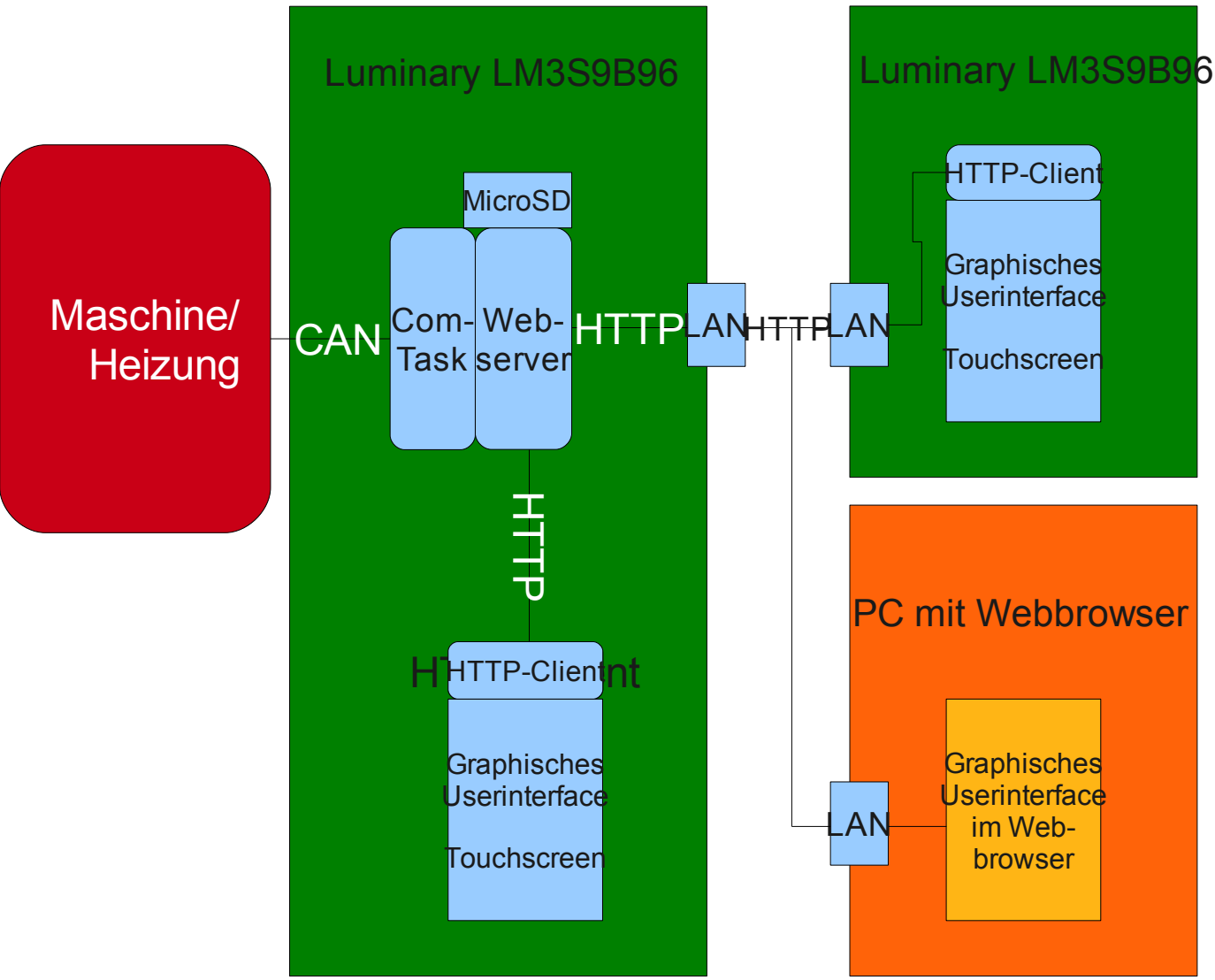
Konzept

Das Ziel von LumWeb ist, die Vereinfachung der Userinterface-Erstellung.

Erreicht wird das Ziel durch das Zusammenspiel der oben genannten Komponenten.

Das User Interface wird als HTML Seite mit SSI Tags definiert. Die Bedienseiten sind auf einem Serversystem gespeichert. Das Serversystem ist zuständig für die Kommunikation mit der Maschine und mit dem Benutzer. Die Seiten werden über HTTP an die Clients verteilt. Auf PC Systemen übernimmt der Browser die Darstellung. Auf uC Systemen über nimmt der Webclient die Darstellung. Dazu wertet er spezielle HTML Kommentare aus, die für die SSI Tags eingefügt werden.

Um eine Bedienseite zu erstellen werden in den HTML Code Bedienelemente in Form von SSI Tags eingebaut, zum Beispiel der Tag NumberInputField für die Eingabe einer Zahl. Zusätzlich muss eine ID angegeben werden, die das Element mit eine Einstellung der Maschine verbindet. Der ComTask wertet diese ID aus und liefert den entsprechenden Wert. Der Webserver fügt beim Auswerten der Tags außerdem spezielle HTML Kommentare ein, die der Webclient zur Darstellung auswertet. Der Client kann nur solche Tags darstellen, für die er Darstellungsmethoden implementiert.


Luminary LM3S9B96
Luminary LM3S9B96
Maschine/
Heizung
CAN
MicroSD
Com-
Task
Web-
server
HTTP
LAN
HTTP
LAN
HTTP-Client
Graphisches
Userinterface
Touchscreen
HTTP
HTTP-Client
Graphisches
Userinterface
Touchscreen
PC mit Webbrowser
LAN
Graphisches
Userinterface
im Web-
browser