

Lowestcost Wireless Webserver

Prof. Alexander Klapproth, Daniel Käslin, Thomas Peter
Hochschule für Technik und Architektur Luzern
CH-6048 Horw - Schweiz

Einführung

Die Wireless-Welt ist in immer schnelleren Aufwinden. Dank neuen Standards wie IEEE 802.11 (WLAN) und 802.15.1 (Bluetooth) konnte die Funktechnik während den letzten Jahren stetig neue Gebiete erschliessen und sich bei einem breiten Publikum etablieren. Die fortschreitende Standardisierung erreichte mit IEEE 802.15.4 nun auch den Low-Rate und Low-Cost Bereich und setzt hier zum Siegeszug an.

Unternehmen, die bei der Erschliessung dieses Sektors mithelfen, sind vor die strategisch-technologische Frage gestellt, wie der neue IEEE Standard eingesetzt und integriert werden soll. Der IEEE Standard deckt lediglich die Layer 1 (PHY) und Layer 2 (MAC) des OSI Referenzmodells und lässt die Spezifikation der höheren Layer offen. Nebst der altbekannten Möglichkeit, proprietäre Protokolle auf den höheren Schichten einzusetzen, bleiben im Wesentlichen zwei mögliche Optionen: ZigBee und TCP/IP. Diese zwei Protokoll-Stacks unterscheiden sich aber wesentlich, sowohl in Anwendung, als auch in Funktionalität und Integrationsfähigkeit.

Zigbee hat den klaren Vorteil, direkt auf IEEE 802.15.4 aufzubauen und für ressourcenarme Embedded Systeme designed zu sein. Anders TCP/IP, das inzwischen oft etwas überladen und auf Embedded Systemen schwer integrierbar erscheint. Dem gegenüber stehen jedoch die weltweit eingesetzten IT-Systeme, die inzwischen fast alle TCP/IP als Com-Stack benutzen. Zukunftsweisend könnte sich hier der Einsatz von TCP/IPv6 erweisen. IPv6 beinhaltet verschiedene Mechanismen, wie z.B. Headerkomprimierung, die den Einsatz auf ressourcenarmen Systemen erleichtern.

Das Center of Excellence for Embedded Systems Applied Research (CEESAR) der HTA Luzern beschäftigt sich schon seit längerer Zeit mit dem Downsizing von Softwarepaketen für den Einsatz auf kostenminimierten Embedded Systemen. Dabei ist unter anderem ein Wireless Webserver Modul entstanden. Dieses Lowest-Cost Modul integriert verschiedene Internet-Technologien wie TCP/IPv4, HTTP, SOAP, Java und Ethernet sowie IEEE 802.15.4 auf einem 8-Bit Mikrocontroller mit sehr beschränkten Ressourcen. Zurzeit findet eine Integration eines ZigBee Stacks auf den Wireless Webserver statt. Eine Integration von IPv6 wird momentan geprüft.

Die folgenden Ausführungen vermitteln detaillierte Informationen zum Wireless Webserver Modul und zeigen mögliche Anwendungsgebiete.

Hardware

Das Webserver Modul basiert auf einem kostengünstigen 8-bit Mikrocontroller von Freescale. Über eine serielle Schnittstelle (SPI) kann auf zusätzlichen Flash-Speicher zugegriffen werden. Über die selbe Schnittstelle wird auch der Funkchip von Chipcon angesteuert, der im 2.4GHz Band nach der IEEE 802.15.4 Norm kommuniziert. Die Leiterplatte benötigt nur zwei Layer und ist daher kostengünstig herzustellen. Auch bei der Auswahl der Komponenten wurde auf minimale Kosten geachtet.

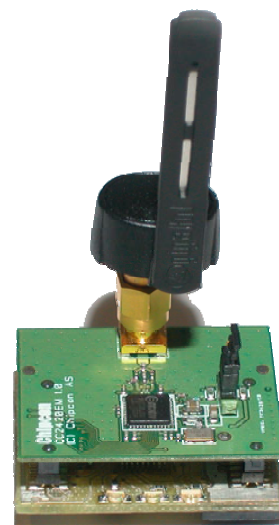


Abbildung 1 - Wireless Webserver Modul
(Quelle: Fachhochschule Luzern)

Die Tabelle fasst die verwendete Hardware zusammen:

Produkt	Daten	
Mikrocontroller <i>Freescale MC9S08</i>	Takt	0 bis 20 MHz (PLL-Synthesizer)
	Stromverbrauch	16 MHz: 5mA, 2 MHz: 1mA, Stop mode: < 1 µA
	Spannung	1.8 - 3.6 V
	Speicher	60 KB Flash, 4 KB SRAM
RF Transceiver <i>Chipcon CC2420</i>	Frequenz	2.4 GHz
	Datenrate	250 kbs
	Reichweite	30 m
	Stromverbrauch	Empfangen: 19.7 mA, Senden: 17.4 mA, Powerdown: 20 µA, Idlemode 426 µA
	Spannung	1.6 - 3.6 V
	Schnittstelle	SPI
	Datenpuffer	Empfangen: 128 Byte, Senden: 128 Byte
	Technologie	DSSS, O-QPSK
Flashmemory <i>Atmel DataFlash</i>	Stromverbrauch	Standby: 2 µA, Lesen: 10 µA, Schreiben: 15 µA
	Spannung	2.5 - 3.6 V
	Schnittstelle	SPI (max. 20 MHz)
	Speicher	4 MBit - 32 MBit
	Datenpuffer	264 Byte

Tabelle 1 - Übersicht Hardware

Software

Die an der HTA Luzern entwickelte Software implementiert eine Vielzahl von international verbreiteten Standards. Trotz der sehr eingeschränkten Ressourcen der Hardware konnte ein Webserver, die Unterstützung von Web Services und eine Java Virtual Machine realisiert werden.

Webserver

Der Webserver kann statische und dynamische Webseiten zur Verfügung stellen. Bei den dynamischen Webseiten werden Templates eingesetzt. Bei einem Aufruf eines solchen Templates, ersetzt der Webserver Schlüsselwörter durch dynamische Werte und liefert erst dann die Seite an den Client. Die statischen Webseiten und die Templates werden wahlweise im RAM, im Programmspeicher (readonly) oder im externen Flash Speicher abgelegt. Dank eines Dateisystems können die Webseiten bequem ausgetauscht oder ergänzt werden. Hierfür stellt der Webserver eine Upload Seite zur Verfügung.

Web Services

Die Software unterstützt SOAP-RPC. Damit können über das Netzwerk Funktionen (Web Services) auf einem "entfernten" System aufgerufen werden (remote procedure call). Da SOAP ein offener W3C Standard ist und auf bewährten Technologien wie HTTP und XML basiert, ist eine hohe Interoperabilität gewährleistet. Es ist daher problemlos möglich, mit der Software über ein Netzwerk auf einen Windows PC zuzugreifen und dort einen Web Service aufzurufen. Umgekehrt können auf dem Modul selbst Web Services implementiert werden, die dann von anderen Systemen aufgerufen werden.

Java Virtual Machine (JVM)

Java Programme zeichnen sich durch ihre Plattformunabhängigkeit und ihre hohe Sicherheit aus. Zudem lassen sich Java Klassen dynamisch laden. Dies ist vor allem für eingebettete Systeme interessant, die so über ein Netzwerk einfach aktualisiert werden können. Damit Java Programme auf einer Plattform ausgeführt werden können, muss eine Java Virtual Machine (JVM) installiert sein.

Die Java Klassen können auch in den externen Flashspeicher mit Dateisystem gespeichert werden und von dort direkt von der JVM interpretiert werden.

Trotz der sehr geringen Codegrösse von weniger als 30 Kilobyte, bietet die JVM die wichtigsten Funktionen. Die Tabelle gibt einen kurzen Überblick:

Modul	Funktion
Java Bytecode Interpreter	Führt die Java Klassen (resp. den Bytecode) aus, auch aus dem Flash-Speicher
Native Interface	Aufrufen von C Code innerhalb der Java Anwendung
ClassLoader	Lädt dynamisch Klassen über das Netzwerk oder aus dem lokalen Dateisystem, respektive aus dem bis zu 32 MBit grossen Flash-Speicher.
Garbage Collector	Entfernt nicht mehr benötigte Objekte aus dem Speicher, automatisch oder durch expliziten Aufruf.

Tabelle 2 - Module der Java Virtual Machine

Die folgende Tabelle fasst die Module und deren Funktionsumfang zusammen:

Modul	Funktionsumfang
Webserver	Statische und dynamische Webseiten Upload von Dateien
Web Services	Bereitstellen von Web Services (Server) Aufrufen von Web Services (Client)
Java Virtual Machine	Ausführen von Java Anwendungen Dynamisches Laden von Java Klassen (lokal und über Netzwerk) Garbage Collector

Tabelle 3 - Funktionen der Software

Architektur

Die Software basiert auf Modulen, die sich je nach Bedarf einbinden lassen. So ist es z.B. möglich, die JVM wegzulassen um mehr Speicher für die Anwendung zur Verfügung zu haben.

Abbildung 2 **Fehler! Verweisquelle konnte nicht gefunden werden.** zeigt die Architektur der Software. Alle Softwarekomponenten zusammen benötigen weniger als 60 Kilobyte an Programmspeicher. Eine Zusammenstellung der Grössen der einzelnen Software-Komponenten zeigt Abbildung 3.

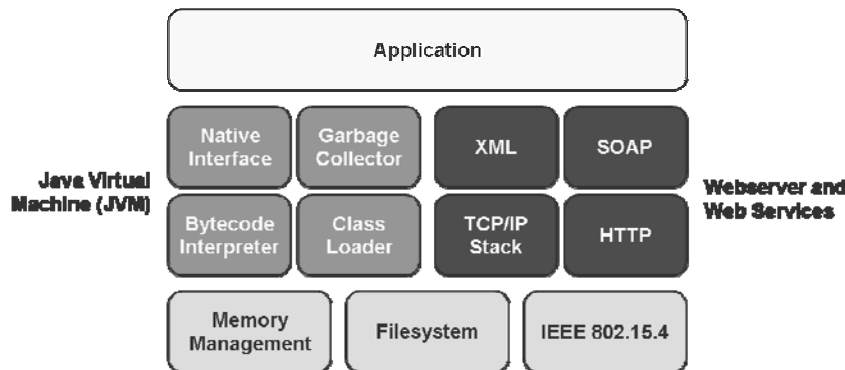


Abbildung 2 - Architektur Software
(Quelle: Fachhochschule Luzern)

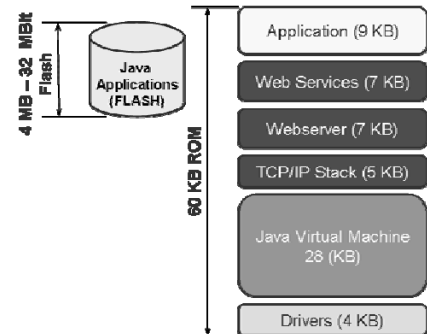


Abbildung 3 - Grösse der Software Module
(Quelle: Fachhochschule Luzern)

Anwendung

Das Webserver Modul lässt sich äussert vielseitig einsetzen. Wo immer ein LR-WPAN (*Low Rate Wireless Personal Area Network*) zum Einsatz kommt und sowohl die Produktkosten, als auch die Entwicklungszeiten kostensensitiv betrachtet werden, führt das Webserver Modul schneller und günstiger zum Ziel. Das Webserver Modul ist nicht nur kostengünstig im Gestehungspreis (5 Euro), sondern dank integriertem TCP/IPv4, HTTP und SOAP schnell, einfach und damit kostengünstig im Engineering in bestehende IT-Strukturen einzubinden. Je nach Aufgabenstellung kann das Modul um Technologien (z.B. ZigBee) erweitert werden. Anhand von zwei konkreten Anwendungsbeispielen werden im Folgenden mögliche Einsatzszenarios von Webserver Modulen aufgezeigt und deren Vorteile erläutert.

Web Enabled Wireless Sensor Networks

Der LR-WPAN Standard IEEE 802.15.4 ermöglicht Sensor Netzwerke auf Basis von standardisierten Protokollen und Geräten, was sich positiv in den Kosten niederschlägt. Durch die Reduktion der Kosten lassen sich neue Anwendungsgebiete erschliessen. Ungelöst, da nicht standardisiert, bleibt dabei allerdings der Zugriff auf die im WSN (*Wireless Sensor Netzwerk*) gewonnen Daten. Hier bietet das Webserver Modul eine bewährte, standardisierte Schnittstelle für die Integration eines WSN in bestehende IT-Systeme. Das Webserver Modul kann dabei je nach Anforderungsprofil als Sensor-Endknoten oder als Koordinator eingesetzt werden. Massgebend ist, dass ein möglicher Client über das HTTP-Protokoll auf das WSN zugreift. Die Daten werden vom Webserver gesammelt und zur Präsentation und weiteren Verarbeitung aufbereitet. Ist die abfragende Instanz ein Mensch, so geschieht der Zugriff auf die aufbereiteten Daten über einen Standardbrowser, ist die Instanz ein IT-System, kommt SOAP über HTTP zum Einsatz. Der Zugriff kann ortsunabhängig über das Internet erfolgen.

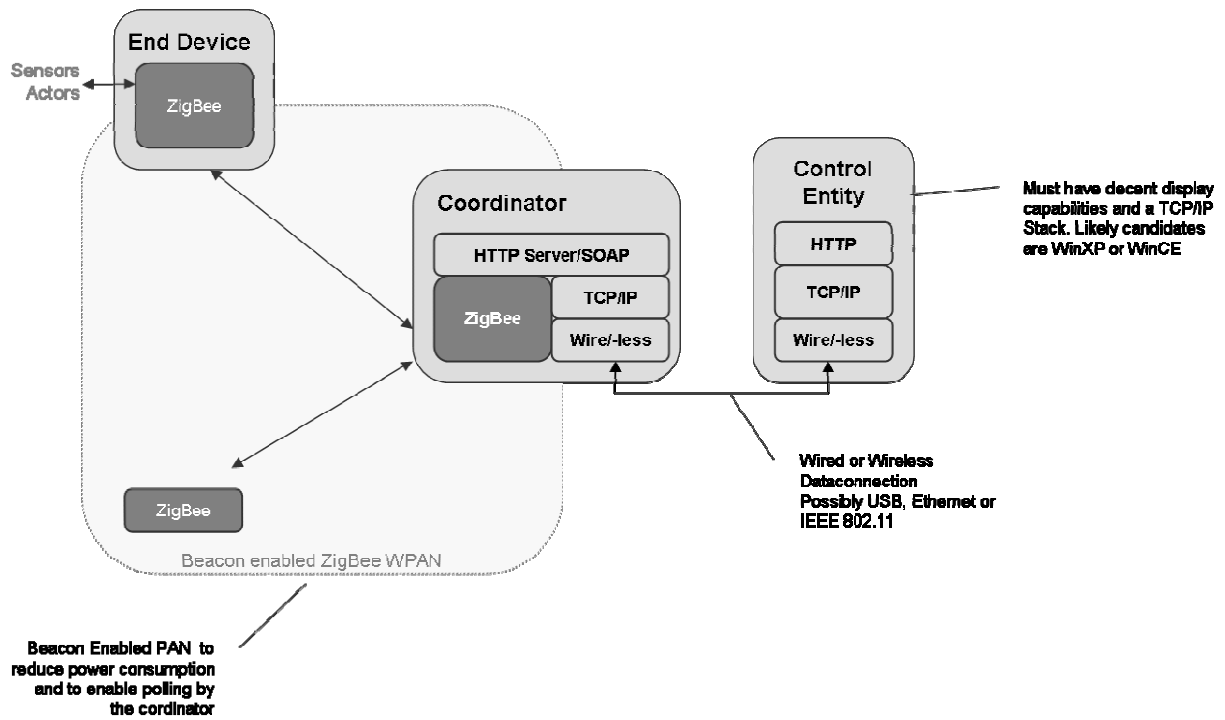


Abbildung 4 - Wireless Sensor Netzwerk mit Webserver Modul als Koordinator und Gateway
(Quelle: Fachhochschule Luzern)

Die Platzierung des Webservers im WSN erfolgt je nach Anforderung und Problemstellung auf dem Koordinator oder direkt auf den Endknoten.

Universal Fernsteuerung

Der Wunsch nach einer allumfassenden, portablen Fernsteuerung für sämtliche elektronischen Geräte in einem Gebäude existiert bereits seit einiger Zeit. Dass noch keine gängigen Lösungen vorhanden sind liegt nicht zuletzt an der Unmenge von heterogenen Geräten, die kontrolliert werden müssen und dem langzeitigen Ausbleiben praktikabler Standards für das Management solch dynamischer Umgebungen. Prinzipiellerweise muss eine Universal Fernsteuerung zwingend auf Standards basieren. Zudem muss sie möglichst wenig Zusatzaufwand für die Bereitstellung der Schnittstelle bei den Geräteherstellern erfordern und sowohl einfach als auch günstig in Betrieb zu nehmen sein. Dies sind zwingende Voraussetzungen, damit sich eine solche universelle Fernsteuerung überhaupt durchsetzen kann.

Gängige Lösungen bieten meist gute Ansätze für viele der Anforderungen, können aber praktisch nie alle zusammen erfüllen. Dank den neuen Standards für LR-WPANS (IEEE 802.15.4 und ZigBee) und SOAP-fähigen, kostengünstigen Funk-Modulen wie dem Webserver Modul sehen wir einen Lösungsansatz für diese Problematik. Basierend auf der UPnP-Technologie und portiert auf Webserver Module, lassen sich kostengünstige, standardisierte, selbst-vermittelnde Funk-Endknoten für Geräte realisieren, die über Browser-Interfaces ferngesteuert werden können.

Autoren



Prof. Alexander Klapproth
Leiter CEESAR
Hochschule für Technik+Architektur Luzern, Schweiz
Mail: ceesar@hta.fhz.ch
Web: www.ceesar.ch



Daniel Käslin
Wissenschaftlicher Mitarbeiter CEESAR
Hochschule für Technik+Architektur Luzern, Schweiz
Mail: dkaeslin@hta.fhz.ch
Web: www.ceesar.ch



Thomas Peter
Wissenschaftlicher Mitarbeiter CEESAR
Hochschule für Technik+Architektur Luzern, Schweiz
Mail: tpeter@hta.fhz.ch
Web: www.ceesar.ch