

ARBEITSBERICHT

ANWENDUNG DRAHTLOSER KOMMUNIKATIONSTECHNOLOGIEN ZUM AUFBAU EINES SENSORNETZWERKES FÜR VERTEILTE MESSAUFGABEN

Application of wireless communication technologies to build a sensor network for distributed measurement

Henri Schwarz, Hannes Heimbach genannt Göthe & Uwe Heuert*

Hochschule Merseburg | Geusaer Str. 88 | D-06217 Merseburg

Eingegangen am 13.03.2012; Überarbeitet eingereicht am 02.06.2012; nicht peer-reviewed

Lektorat: Nikola Wiegeler

Zusammenfassung

Ziel der Arbeit sind Entwicklung und Aufbau eines auf Funktechnologie basierenden Sensornetzwerkes, das verschiedene physikalische Messgrößen determiniert erfassen, speichern und darstellen kann. Die zugrunde liegende Sensormodulhardware wird dabei so ausgelegt, dass sie möglichst klein, robust und energiesparend ist. Die Kommunikation zwischen einzelnen Knoten des Sensornetzwerkes wird über einen Protokollstapel nach dem TCP/IP-Referenzmodell gewährleistet, dessen physikalische Schicht auf die verwendete Hardware angepasst ist. In einer zukünftigen Evolutionsstufe des Projektes ist geplant, weitere TCP/IP basierende Protokolle (z.B. Universal Protocol, siehe z. B. Punk & Heuert (2011), LXI) in den Protokollstapel zu integrieren, damit die gewonnenen Messdaten von Endnutzeranwendungen wie MATLAB oder LabVIEW entgegengenommen und verarbeitet werden können.

Schlüsselwörter: eingebettete Systeme | drahtlose Kommunikation | Measurement Cloud

Abstract

Modern sciences have a high demand for precise quantitative measurement techniques to analyse natural phenomena. Therefore, the projects main task is the development of a sensor network based on wireless communication technologies to measure, store and display physical variables. The sensor nodes' hardware should feature a small form factor, robustness and low energy consumption. The nodes' communication within the wireless network should be handled by a TCP/IP similar protocol stack with adaptations to the devices' physical layer. This requirement comes due to intentional future use of other TCP/IP based high level protocols like the Universal Protocol [Punk & Heuert (2011) or the Lan Extensions for Instrumentation (LXI) protocol. These protocols enable software like MATLAB and LabVIEW to retrieve data directly from the measurement cloud.

keywords: Embedded System | Wireless Communication | Measurement Cloud

Die modernen Ingenieur- und Naturwissenschaften sind neben theoretischen Modellen zur Beschreibung von physikalischen Phänomenen immer stärker auf präzise Messungen angewiesen, die eine möglichst genaue Quantifizierung der Messdaten erlauben. Eine große Herausforderung für die Messtechnik sind dabei verteilte Messaufgaben, die notwendig werden, sobald z. B. Daten an großen Objekten erhoben oder weite Flächen mit Sensoren überwacht werden sollen. Funkbasierte Sensornetze, deren Sensorknoten auf niedrigen Energieverbrauch, Robustheit und kompakte Bauweise ausgelegt sind, können in diesen und ähnlichen Szenarien ihre Stärken ausspielen.

Neben den bereits genannten Randbedingungen spielt der Kommunikationsablauf zwischen den Sensorknoten eine entscheidende Rolle. Da Funkübertragungen stör anfällig sind, müssen Codierungs- und Prüfverfahren implementiert werden, mit denen die Datenintegrität sichergestellt werden kann.

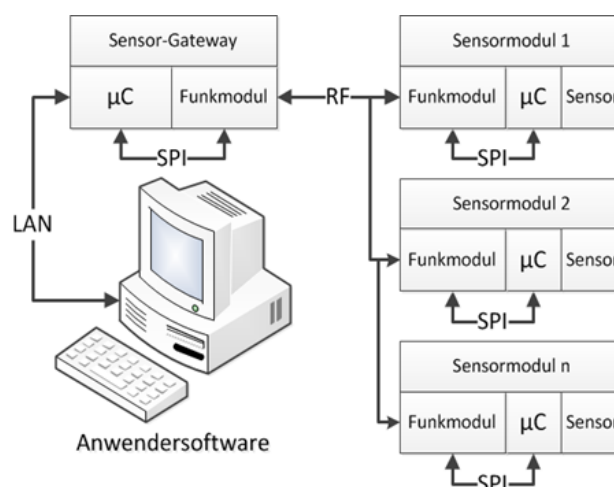


Abb. 1: Infrastruktur eines Sensornetzwerkes

*Kontakt: henri.schwarz@googlemail.com

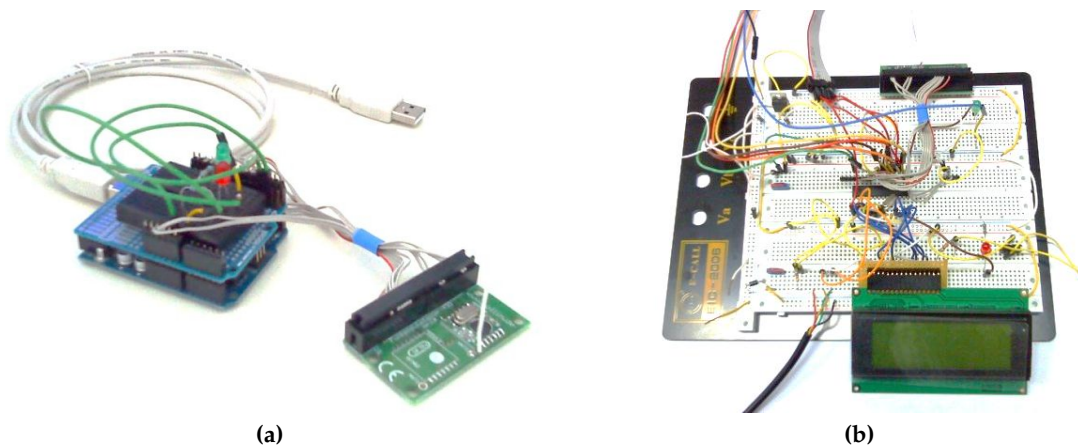


Abb. 2: Aufbau der (a) Sende - und (b) Empfangsschaltung

Darüber hinaus sollten standardisierte Kommunikationsprotokolle, wie z.B. das Transmission Control Protocol (TCP) und das Internet Protokoll (IP), zur Übertragung der gewonnenen Messdaten genutzt werden. Die Zuverlässigkeit und Übertragungssicherheit des Systems können z.B. durch die Verbindungsorientiertheit des TCP-Protokolls noch zusätzlich gesteigert werden (3-Way-Handshake). Ziel der vorliegenden Arbeit ist die Entwicklung eines derartigen Systems.

MATERIAL UND METHODEN

Allgemeines

Ein Sensornetzwerk lässt sich grundlegend als Netzwerk aus Rechnern oder Mikrorechnern, den so genannten Sensorknoten, beschreiben (siehe Abb. 1). Die einzelnen Knoten besitzen in der Regel Schnittstellen für terrestrische Funknetzwerke und gliedern sich in deren Infrastruktur ein. Im Gegensatz dazu stehen sich selbst organisierende Ad-hoc-Netzwerke. Im Rahmen dieses Projektes wird der Ansatz infrastrukturbasierter Drahtlosnetzwerke verfolgt, bei dem die Daten mithilfe eines Gateways (Router) vom Funknetz in Ethernet basierte Netzwerke umgesetzt werden können. Das motiviert den Einsatz des TCP/IP Protokollstapels auf der Funkseite zusätzlich.

Das Herzstück eines jeden Sensorknotens ist ein Mikroprozessor, dessen Eigenschaften den Anforderungen an Energieverbrauch, Robust- sowie Kompaktheit gerecht werden sollen. Die Hardwareplattform soll daher mit Einhausung einen möglichst kleinen Formfaktor aufweisen und für den Batteriebetrieb geeignet sein. Die Entscheidung für einen bestimmten Mikroprozessor als Basis des Systems hängt neben den elektrischen Eigenschaften und anderen Parametern, wie dem verfügbaren Speicher, der möglichen Taktrate und der internen Busbreite, maßgeblich von den zur Programmierung benötigten Werkzeugen ab. Viele Mikroprozessoren sind meist über einfache serielle Schnittstellen, wie dem Serial Peripheral Interface (SPI), direkt im Einsatzsystem programmierbar. In der Regel erfordert die In-System-Programmierung (ISP) jedoch einen speziellen Programmieradapter, dessen Anschaffung

einen erheblichen Teil der Hardwarekosten ausmachen kann. Der ISP-Adapter verbindet Entwicklungs- mit Zielplattform und ermöglicht das Laden des Programmcodes in den Speicher des Controllers. Die wesentlichen Programmteile der Sensorknotensoftware aggregieren und verarbeiten Sensordaten, steuern die Schnittstellen des Mikrocontrollers und organisieren die Netzwerkkommunikation nach den Vorgaben des Protokollstapels.

Hardware

Im Rahmen dieses Projektes werden Funkmodule eingesetzt, die mit einer Basisfrequenz von 868 MHz arbeiten und als Sender und Empfänger agieren können. Die Basisfrequenz wird den Industrial, Scientific and Medical (ISM) Frequenzbändern zugeordnet, weshalb unter Beachtung der Allgemeinzuteilungsregulierungen (siehe Bundesnetzagentur, 2003) keine

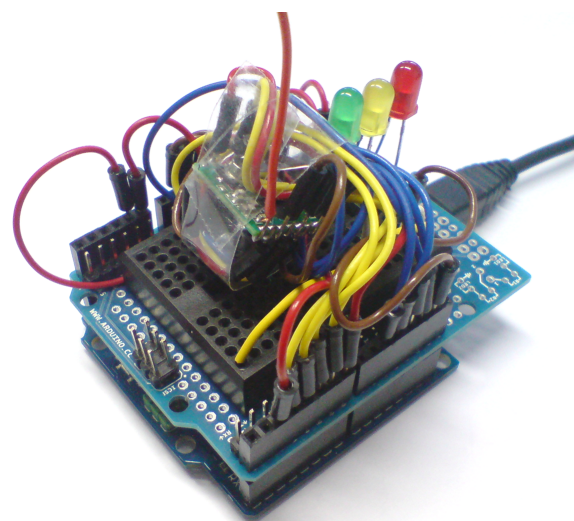


Abb. 3: Hardwareplattform Arduino Pro

weiteren Lizenzkosten für deren Nutzung anfallen. In den bisherigen Evolutionsstufen des Projektes hat sich die Mikroprozessorplattform mehrfach geändert, wie in den folgenden Abschnitten beschrieben wird.

Sensorknoten v1 Um erste Erfahrungen mit den Funkmodulen (Produktübersicht, 2011) zu sammeln, sind diese auf speziellen Evaluationsboards getestet worden. Die Datenübertragung erfolgte hier unidirektional zwischen einem Sender und Empfänger. Der Programmcode ist in der Programmiersprache C geschrieben. Die Datenübertragung erfolgt gepuffert, d. h. zu sendende Zeichen werden vor der Übertragung in einen Sendespeicher auf dem Funkmodul geladen. Der Mikroprozessor sendet die Zeichen über eine durch Software emulierte SPI Schnittstelle. Die maximale Zeichenlänge je Übertragung ist durch diesen Puffer auf acht Bit limitiert. Kommen auf dieser Ebene keine zusätzlichen Steuerzeichen zum Einsatz, kann mit jedem Sendevorgang ein acht Bit langes Zeichen z.B. aus dem American Standard for Information Interchange (ASCII) Blockcode übertragen werden. Auf Empfängerseite ist es so jedoch nicht möglich, Übertragungsfehler zu erkennen oder zu korrigieren. Da beim Feldeinsatz eines solchen Systems nicht davon ausgegangen werden kann, dass die Übertragung störungsfrei abläuft, muss für Sicherungsmaßnahmen gesorgt werden.

Sensorknoten v2 Durch das vorgegebene Layout der Evaluationsboards fehlt die Möglichkeit, die Hardware an unterschiedliche Einsatzbedingungen im Sensornetzwerk anzupassen. Dies begründet den Wechsel zu modularisierten Prototyping Plattformen, wie Arduino (Arduino (2011)), und die Entwicklung prototypischer Schaltungen (siehe Abb. 2b).

Beide Plattformen verwenden einen Atmel Atmega168 Mikrocontroller im Dual-in-Line (DIL) Package. In diesem Stadium besteht das Netzwerk ebenfalls aus einem Sender und einem Empfänger (siehe Abb. 2), wobei der Sender eine durch Analog-Digital-Wandlung gemessene Spannung an den Empfänger übermittelt. Dieser liest die empfangenen Daten und stellt den Spannungswert über das angeschlossene LCD Display grafisch dar. Weiterhin wird eine am Empfänger angeschlossene LED in Abhängigkeit des übertragenen Spannungswertes durch Pulsweitenmodulation zum Leuchten angeregt. Der Aufbau zeigt die Eignung des Systems für Mess- und Steueraufgaben, wobei die Datenübertragung auch hier noch ungesichert erfolgt.

Der Programmcode in diesem Entwicklungsstadium befindet sich sowohl für Sender als auch Empfänger in einer einzigen C-Quelldatei, was für Testzwecke ausreicht, aber nicht den Maßstäben moderner Softwareentwicklung gerecht wird. Aus diesem Grund wird die Software im weiteren Verlauf modularisiert.

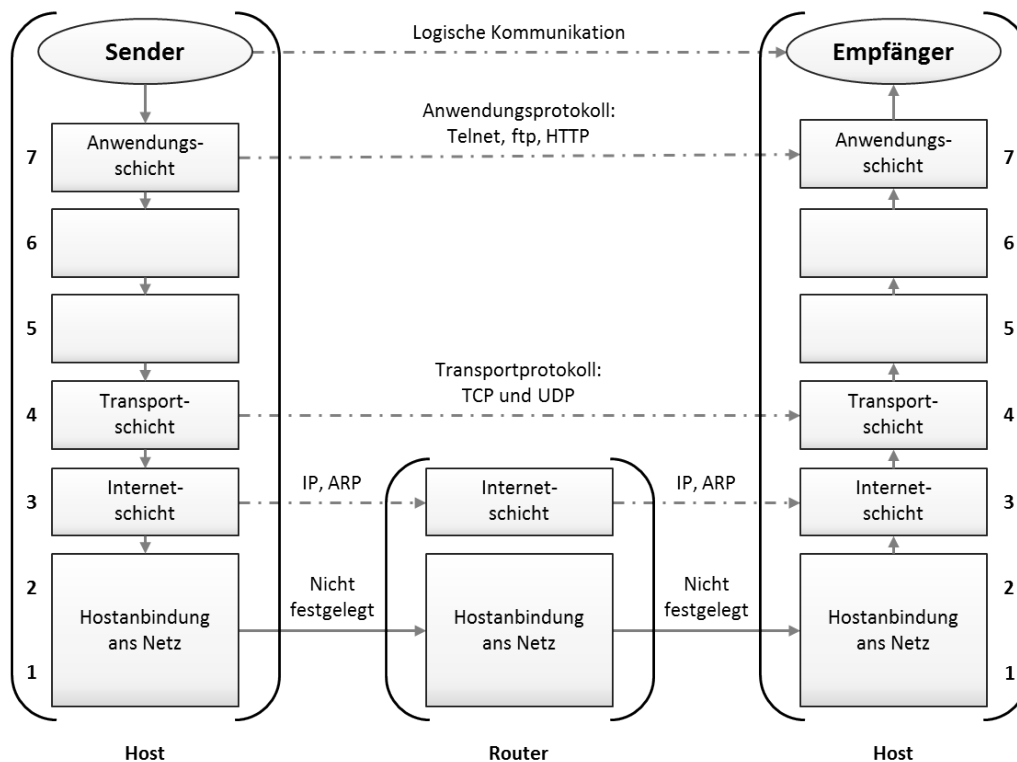


Abb. 4: TCP/IP Referenzmodell

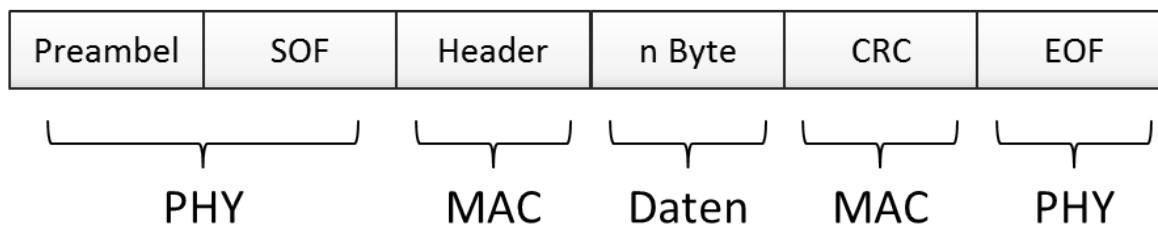


Abb. 5: Struktur von Datenpaketen im physikalischen Netz

Sensorknoten v3 In der aktuellen Entwicklungsstufe wird die Prototyping Plattform Arduino Pro (Abb. 3) und das AVR Net-IO Board (Pollin-Electronic, 2011) verwendet. Die Arduino Pro Plattformen, bestückt mit einem Low-Power Atmel Atmega328P, werden als Sensorknoten eingesetzt. Dadurch können diese batteriebetrieben werden. Das AVR Net-IO Board ist mit einem Atmel Atmega644 bestückt, der mehr Speicher zur Pufferung der Datenpakete bereitstellt. Das Net-IO Board verfügt dazu neben einer nachgerüsteten Drahtloschnittstelle über einen Anschluss für Ethernet basierte Netzwerke. Der Netzwerkcontroller ist ein ENC28J60 von Microchip, der den 10Base-T Standard mit 10 MBit/s unterstützt.

Die aktuellen Entwicklungen konzentrieren sich auf die Implementierung des Programmcodes für die hardwarenahen Schichten des TCP/IP Protokollstapels im Drahtlosnetzwerk. Ihre Aufgabe ist es, Mechanismen zur Absicherung gegen Störsignale zu implementieren (Kanalcodierung). Im Besonderen handelt es sich dabei um die Bitübertragungsschicht (PHY) und die Sicherungsschicht (MAC) (Gessler & Krause, 2009), die speziell auf die verwendete Hardware angepasst werden müssen. Im TCP/IP Referenzmodell werden beide Schichten unter „Hostbindung ans Netz“ zusammengefasst (Abb. 4).

Werkzeugkette

Die Entwicklung des Programmcodes erfolgt in einer integrierten Entwicklungsumgebung, die einen für ATMEL Mikroprozessoren entwickelten Compiler, Assembler sowie einen Simulator mitbringt. Die verfügbaren Programmiersprachen sind Assembler und C/C++. Der geschriebene Programmcodes wird in Projekten angelegt und zusätzlich im Versionskontrollsystem der Forschergruppe um Prof. Heuert verwaltet. Dadurch lassen sich Softwarefehler in neuen Programmversionen schneller verfolgen und beseitigen. Beim Erstellen des Programms wird eine sog. hex-Datei erzeugt, die mithilfe von Programmieradaptern in den Speicher des Mikroprozessors geladen wird. Um während der Laufzeit nach Fehlern im Programm zu suchen, werden fortgeschrittene Programmieradapter benötigt, andernfalls kann sich der Entwickler das korrekte Programmverhalten durch z. B. aufleuchtende LEDs oder durch Ausgaben auf seriellen Schnittstellen wie Universal Asynchron Receiver Transmitter (UART) anzeigen lassen. Die mittels UART übertragenen Daten können auf Rechnerseite auch per Software entgegengenommen und protokolliert werden. Die Kommunikation zwischen Mikroprozessor und Funkmodul kann durch die Überwachung der

Signalpegel auf den Leitungen des eingesetzten Datenbusses mithilfe eines Logikanalysators sichtbar gemacht werden.

ERGEBNISSE

Die sichtbaren Veränderungen innerhalb der einzelnen Entwicklungsstufen beziehen sich auf die eingesetzten Hardwareplattformen. Hier sind die gesteigerte Kompaktheit durch kleinere Formfaktoren und möglicher Batteriebetrieb der Sensorknoten durch geringeren Energieverbrauch zu nennen. Ein kleiner Nachteil, der durch die auf 3 V reduzierte Versorgungsspannung der AVR ATmega328P Mikroprozessoren auf den Arduino Pro Plattformen entsteht, ist der verringerte maximale Prozessortakt von 8 MHz gegenüber den bei 5 V Versorgungsspannung möglichen 20 MHz. Auf Softwareseite konnten die Protokollschichten, die den Host ans physikalische Netz anschließen (PHY, MAC) mit entsprechenden Verfahren zur Sicherstellung der Datenintegrität implementiert werden. Zu diesen Mechanismen gehören zyklische Redundanzprüfungen mittels Prüfsumme sowie die Hamming-Codierung und Decodierung der übertragenen Datenbytes. Abbildung 5 zeigt den Aufbau eines Datenpaketes im physikalischen Netz. An Stelle der Daten treten im weiteren Verlauf protokollspezifische Informationen höherer Schichten, die ihre eigenen Nutzdaten verwalten.

LITERATUR

- Arduino (2011):** *Arduino-Homepage*. <http://www.arduino.cc/>.
- Bundesnetzagentur (2003):** <http://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/BNNetzA/Sachgebiete/Telekommunikation/Regulierung/Frequenzordnung/Allgemeinzuteilung/FundstelleId298pdf.pdf>.
- Gessler R & Krause T (2009):** *Wireless-Netzwerke für den Nahbereich. Eingebettete Funkssysteme: Vergleich von standardisierten und proprietären Verfahren: Grundlagen, Verfahren, Vergleich, Entwicklung*. Vieweg + Teubner.
- Pollin-Electronic (2011):** *AVR Net-IO Board*. <http://www.pollin.de/shop/downloads/D810049S.ZIP>.
- Produktübersicht (2011):** *RF-Funkmodul: Rfm12 Produktübersicht*. http://www.hoperf.de/rf_fsk/fsk/21.htm.
- Punk O & Heuert U (2011):** *Universal Protocol: Cross Code Generation*. 12. Nachwuchswissenschaftlerkonferenz, Tagungsband, 103–108, Wernigerode, April 2011.

Zu zitieren als: **Schwarz H, Heimbach H & Heuert U (2012):**
Anwendung drahtloser Kommunikationstechnologien zum
Aufbau eines Sensornetzwerkes für verteilte Messaufgaben.
Zeitschrift für Nachwuchswissenschaftler 2012/1

Please cite as: **Schwarz H, Heimbach H & Heuert U (2012):**
Application of wireless communication technologies to build

a sensor network for distributed measurement. *German Journal
for Young Researchers* 2012/1

URL: <http://www.nachwuchswissenschaftler.org/2012/1/80/>

URN: [urn:nbn:de:0253-2012-1-805](https://nbn-resolving.org/urn:nbn:de:0253-2012-1-805)