INGA - Architektur eines universell einsetzbaren Sensorknotens

Ulf Kulau
Institut für Betriebssysteme und Rechnerverbund
TU Braunschweig
Email: kulau@ibr.cs.tu-bs.de

Felix Büsching
Institut für Betriebssysteme und Rechnerverbund
TU Braunschweig
Email: buesching@ibr.cs.tu-bs.de

Wolf-Bastian Pöttner
Institut für Betriebssysteme und Rechnerverbund
TU Braunschweig
Email: poettner@ibr.cs.tu-bs.de

Lars Wolf
Institut für Betriebssysteme und Rechnerverbund
TU Braunschweig
Email: wolf@ibr.cs.tu-bs.de

***Abstract*—Nachdem wir einige Erfahrung auf dem Gebiet der drahtlosen Sensornetze gesammelt haben, haben wir uns entschieden, einen weiteren – eigenen – Sensorknoten zu entwickeln. Ziel war es dabei, die Stärken vorhandener Sensorknoten möglichst zu kombinieren und die Schwächen zu minimieren. Außerdem war es uns wichtig, ein Set an aktuellen Sensoren anzubinden, welche eine Aktivitäts- und Bewegungsüberwachung erlauben. Heraus kam dabei INGA - "Inexpensive Sensor Node for General Applications".**

I. Einleitung

Ausgangspunkt für unsere Arbeit war eine Analyse vorhandener und frei erhältlicher Sensorknoten. Da wir in mehreren Projekten "Contiki" als Betriebssystem einsetzen, fiel ein besonderes Augenmerk bei der Analyse auf den weit verbreiteten TMote Sky und auf den Atmel AVR Raven. Im FP7-Projekt GINSENG [1] sind wir des Öfteren an die Grenzen des adressierbaren Speichers des MSP430 Microcontrollers im TMote Sky gestoßen und auch der Energiebedarf dieses Knotens ist verhältnismäßig hoch. Beim GAL-Projekt [2] und den dort verwendeten AVR Raven Sensorknoten störten die Größe, die schlechte Handhabbarkeit und das ungeeignete Sensorset. Letztendlich haben wir uns für eine Basis-Architektur bestehend aus einem AVR Amega1284p und einem AT86RF231 entschieden. In Hinblick auf den Einsatz des Sensorknotens in Forschung und Lehre, ist die große AVR Community, welche den Einstieg in die Programmierung erleichtert, ein weiteres Argument für unsere Entscheidung.
Die Auswahl der Sensorik richtet sich primär an Anforderungen aus dem medizinischen Umfeld: Zum Messen von Beschleunigungen wird ein Accelerometer eingesetzt, welches die Kraft, die auf eine seismische Masse wirkt, in drei Dimensionen quantifiziert. Damit lassen sich Bewegungen und Bewegungsabläufe erkennen, die ein Körper vollzieht, an welchem der Sensor befestigt ist. Auf diese Weise lassen sich Stürze erkennen oder auch Ganganalysen durchführen. Da jeweils auch die Erdbeschleunigung gemessen wird, lässt es sich auch für eine Lagebestimmung nutzen. Zur genaueren Lagebestimmung und Erkennung von Richtungsänderungen kommt ein micromechanisches Kreiselinstrument zum Einsatz. Das verwendete Gyroskop kann Änderungen von bis zu 2000 Grad pro Sekunde in drei Dimensionen detektieren. Ebenfalls zur Ganganalyse, bzw. zur Aktivitätserkennung kann der Luftdrucksensor verwendet werden; er ist in der Lage, Luftdruckänderungen im Bereich von bis zu 0,01 hPa zu registrieren, was einer Höhenänderung im Bereich von wenigen Zentimetern entspricht. Zur Temperaturkompensation ist ein Temperatursensor integriert, was auch die Bestimmung des absoluten Luftdrucks erlaubt.

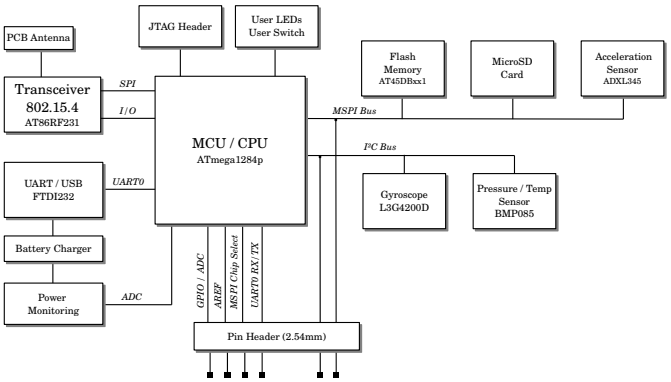

Fig. 1. Architektur von INGA.

II. Architektur

In Abbildung 1 ist ein Blockschaltbild des INGA Sensorknotens zu sehen: Herzstück ist die ATmega1284p MCU, die auch beim Atmel AVR Raven zum Einsatz kommt. Als Funktransceiver kommt hier aber der neuere und leistungsfähigere AT86RF231 zum Einsatz, der unter anderem auch eine Hardwareunterstützung für AES bietet.

A. PCB-Antenne

Die Sendereichweite der AVR-Raven hat sich in unseren Untersuchungen als sehr gut herausgestellt. Da eine

'gedruckte' PCB-Antenne noch dazu billig ist, haben wir uns entschieden, dieses Antennenlayout einfach aus den Designrichtlinien von Atmel zu übernehmen. Als Nachteil kann hier jedoch die Größe gesehen werden: knapp die Hälfte des Sensorknotens ist "Antenne".

B. UART/USB Interface

Der UART kann zur Kommunikation mit einem Host-PC verwendet werden, wovon auch beim INGA Gebrauch gemacht werden soll. Die Designanforderungen des INGA sehen deshalb eine Unterstützung der USB Schnittstelle vor, welche durch ein FTDI232, einem UART-USB-Konverter, realisiert wird. Diese Lösung findet sich u.A. auch im Tmote Sky wieder und bietet mehrere Vorteile:Mit der Fähigkeit des ATmega1284p, den UART mittels eines Bootloaders als Schnittstelle zur Programmierung des Programmspeichers zu nutzen, kann somit die USB-Schnittstelle des INGA sowohl zur Kommunikation (z.B. Debug-Ausgaben printf) wie auch als Programmierschnittstelle genutzt werden. Die vom USB gelieferte Leistung wird außerdem genutzt, um über einen Laderegler den LiPo-Akku des INGA aufzuladen.

C. Power Management und Monitoring

Sobald INGA von der USB - Schnittstelle getrennt wird, besteht keine Notwendigkeit mehr, den FTDI232 weiterhin mit Spannung zu versorgen. Die kontrollierte Ladung des Lithium-Ionen-, bzw. Lithium-Polymer-Akkus erfolgt durch einen Maxim Max1555, welcher maximal 100mA Ladestrom aus dem USB abzieht, um einen 1 Zellen Akku aufzuladen. Das Design sieht alternativ eine Versorgung über 'normale' Batterien bzw. andere externe lineare Spannungsquellen vor, wobei Batterien durch eine Diode vor parasitären Ladeströmen geschützt werden. Eine interne Temperaturkontrolle reduziert bei zu hoher Chiptemperatur den Ladestrom, sodass eine Beschädigung des Akku vermieden werden kann. Für das Power-Monitoring werden zwei Kanäle des internen ADC des ATmega1284p genutzt, um die Akku- / Batteriespannung sowie den Versorgungsstrom des Sensorknotens über die Zeit zu ermitteln. Hierfür wurde ein Impedanzwandler bestehend aus einem Operationsverstärker durchaus Sinn ergeben, doch in Hinblick auf ein energiesparendes Design würde eine zusätzliche Belastung des Akkus die Folge sein. Die minimalistische Lösung eines hochohmigen einfachen Spannungsteilers ist durch die hohe Eingangsimpedanz des ADC möglich und daher vorzuziehen.

D. MSPI-Bus

Der USART des ATmega1284p kann alternativ als MSPI (Master SPI) genutzt werden. Die MCU (ATmega1284p) dient dabei als Master und hat die Aufgabe den jeweiligen Kommunikationspartner (Slave) über eine Chip-Select Leitung (CS) explizit auszuwählen, sowie den Takt (SCK) zur synchronen Datenübertragung zu generieren. Für jeden Slave wird demnach eine eigene Chip-Select-Leitung benötigt, sodass der Verdrahtungsaufwand bei 3+n liegt. Um den Aufwand an GPIOPins der MCU zu reduzieren, wird im INGA ein

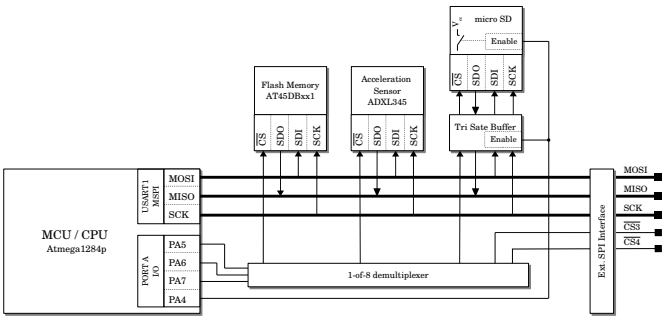

Fig. 2. Der MSPI-Bus.

Demultiplexer eingesetzt, sodass für n Slaves lediglich i = log2(n) I/O-Pins verwendet werden müssen. Das Logik-IC 74LCX138 leistet das gewünschte Verhalten eines 1-of-8 Demultiplexers mit lowaktiven Ausgangspins zur Selektion der einzelnen Slaves.

Nach Auswahl des Slave über die Chip-Select Leitung, wird auf der MOSI Leitung (Master Out Slave In) die Adresse des Registers übertragen, welches man lesen oder schreiben möchte. Je nachdem, ob es sich dabei um einen Lese- oder Schreibzugriff handelt, werden anschließend taktsynchron Daten vom Master an das Slaveregister gesendet (MOSI), oder der Slave sendet ebenfalls taktsynchron Daten über die MISO Leitung (Master In Slave Out) an den Master. Die Übertragung wird durch das lösen der Chip-Select Leitung beendet. Die auf dem INGA eingesetzten Slaves sind ein Flash-Speicher (AT45DBxx1 series), ein Beschleunigungssensor ADXL345 und ein Slot zur Verwendung einer microSD Karte. Zusätzlich sind das MSPI und zwei weitere Chip-Select Leitungen auf das User-Interface geführt und ermöglichen bei Bedarf den Anschluss weiterer Slaves (vgl. Abbildung 2) in Form von Sensoren, Speichern oder anderer SPI kompatibler Komponenten.

E. Micro-SD-Card

Zusätzlich zum internen Speicher der MCU und des onboard-Flash-Speichers ist als flexibles Speichermedium die Möglichkeit zur Verwendung von microSD Karten vorgesehen. Neben dem Vorteil eines austauschbaren Datenträgers, ist eine microSD Karte als Consumer-Produkt sehr gut verfügbar und lässt sich über das integrierte SPI Interface komfortable in ein Mikrocontrollersystem integrieren. Dennoch ist die Verwendung einer microSD Karte nicht vergleichbar mit anderen SPI-fähigen Komponenten und gerade im mobilen Einsatz stellt sich die Frage, wie man die hohe Leistungsaufnahme (I_{sd} ¡ 45mA) der microSD Karte begrenzen kann. Ein Problem mit dem SPI der microSD Karte entsteht dann, wenn sich mehrere Slaves den Bus teilen. Zur Initialisierung müssen an die microSD Karte mindestens 74 Takte auf der SPI Taktleitung (SCK) gesendet werden, allerdings bei Nichtaktivierung des Chip-Select[42]. Nun ist es aber so, dass sich klassischerweise alle am Bus befindlichen Komponenten die Signalleitungen MOSI, MISO und SCK teilen, sodass der Fall von Takten auf der SCK-Leitung bei nicht gewählter microSD Karte durchaus

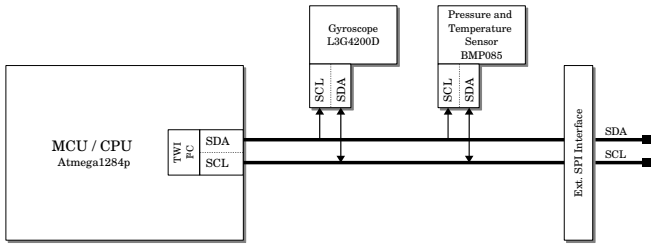

Fig. 3. Der I²C-Bus.

eintritt. Eine einfache Möglichkeit, welche auch das Problem der hohen Leistungsaufnahme löst, ist das Abschalten der Versorgungsspannung bei Inaktivität. Leider zeigt sich, dass microSD Karte aufgrund des internen Aufbaus ihres Interfaces auch über ihre I/O-Leitungen mit Strom versorgt werden kann. Mit einem Tri-State Buffer ist es möglich, die microSD Karte komplett vom Bus zu isolieren, da dieser wahlweise einen hochohmigen Zustand annehmen kann. In Abbildung 2 ist diese Lösung schematisch illustriert.

F. I²C-Bus

Der zweite im INGA eingesetzte Peripherie-Bus ist der I²C. Im Gegensatz zum MSPI ist der Leitungsaufwand deutlich geringer, da unabhängig von der Anzahl eingesetzter Slaves, lediglich zwei Leitungen (Serial Data (SDA), Serial Clock (SCL)) zur Auswahl eines Slave und zur bidirektionalen Kommunikation benötigt werden. Ein Slave wird dabei mittels einer 7 Bit breiten Adresse ausgewählt. Die auf dem INGA eingesetzten Slaves sind ein 3-Achsen Gyroskop mit integriertem Temperatursensor (L3G4200D) und ein Luftdrucksensor (BMP085), welcher ebenfalls über einen internen Sensor zur Temperaturmessung verfügt. Zusätzlich sind SCL und SDA auf das User-Interface geführt und ermöglichen bei Bedarf den Anschluss weiterer Slaves (vgl. Abbildung 3) in Form von Sensoren, Speichern oder anderer kompatibler Komponenten.

III. AVRDUDE KOMPATIBLER BOOTLOADER

Die klassische Variante, ein kompiliertes Programm in den On-Chip Flash - Programmspeicher eines AVR Mikrocontrollers zu schreiben oder auszulesen (Upload / Download), ist die Benutzung des ISP (in System Programmer) oder der JTAG (Joint Test Action Group) Schnittstelle. Beide Methoden setzen jedoch den Einsatz einer separaten Hardware voraus,was einen zusätzlichen Kostenaufwand und Umstand bedeutet. Des Weiteren ist ein Update von im Feldversuch eingesetzten Sensorknoten immer mit einem relativen Aufwand verbunden, da der Sensorknoten und ein entsprechender Programmieradapter lokal verfügbar sein müssen. Die Fähigkeit von AVR Mikroprozessoren zur Nutzung eines bestimmten Bereiches ihres Programmspeichers als Bootloader-Section, bietet eine sehr flexible Basis, um auch zukünftige Anforderungen erfüllen zu können. Auch das Problem der lokalen Verfügbarkeit eines Sensorknotens für ein Softwareupdate, lässt sich somit erschlagen, indem zukünftig ein Wireless-Softwareupdates realisiert werden kann. Ein Anforderung an den Bootloader ist die Kompatibilität zum Host Programm AVRDUDE. Dies ist ein plattformunabhängiges Open Source Programm, welches die Übertragung eines kompilierten Programmes auf dem Mikrocontroller koordiniert und verifiziert. Ein weiterer Vorteil ist die direkte Integration in eine Entwicklungsumgebung (u.A. Eclipse), sodass durch den Bootloader ein Programmieren des INGA via USB direkt aus der Entwicklungsumgebung möglich wird.

Als eigentlicher Flaschenhals kristallisierte sich die standardmäßig geringe Übertragungsrate des UART heraus. Aufgrund der asynchronen Übertragung ist das Zeitverhalten des UART wichtig für eine fehlerfreien Übertragung, da lediglich ein Startsymbol den Beginn einer Übertragung signalisiert und die Abtastung aufgrund der vereinbarten Baudrate stattfindet. Nicht jede Baudrate wird von einem Host unterstützt, sodass man sich an den standardisierten Übertragungsraten orientieren muss, wenn man die Übertragungsgeschwindigkeit seitens des Mikrocontrollers erhöhen möchte. Genau dabei entsteht allerdings ein Problem, den die Baudrate des UART ist abhängig von der Taktfrequenz des Mikrocontrollers.

Unsere Implementierung einer Optimierung im Bootloader bietet eine um den Faktor 6 schnellere UART-Übertragungsrate, sodass z.B. die Übertragung von Contiki weniger als 5 Sekunden benötigt.

IV. ERGEBNIS

Eine komplette und ausführliche Darstellung aller Eigenschaften und Besonderheiten von INGA bedarf eines größeren Raumes und wird in naher Zukunft und auch mit ersten Messergebnissen folgen. Nach unserem Ermessen vereint INGA die Vorteile früherer Sensorknoten und ist dabei klein, kostengünstig und stromsparend. Die umfangreiche Sensorik eignet sich besonders zur Aktivitätsüberwachung im medizinischen Umfeld, ist aber genausogut auch universell einsetzbar. ”Contiki” wird als OpenSource-Betriebssystem unterstützt und der Sensorknoten ist als OpenHardware konzipiert, sodass er frei nachgebaut und verändert werden darf. Es kommen nur bleifreie Materialien zum Einsatz und auch auf Produkte aus ”seltenen Erden” (Coltan/Tantal) wurde bewusst verzichtet. Auch wenn wir die Prototypen bestücken lassen haben ist nicht unmöglich, die einzelnen Bauteile selbst zu löten, bzw. die 2-Layer-Platine selber zu ätzen. Alle Pläne und Dateien werden auf den Webseiten des Projekts zur Verfügung gestellt, sodass ein Nachbau bzw. eine Anpassung an eigene Projekte möglich ist.

REFERENCES

[1] B. Srean, J. Silva, L. Wolf, R. Eiras, T. Voigt, U. Roedig, V. Vassiliou, and G. Hackenbroich, “Performance control in wireless sensor networks: the ginseng project - [global communications news letter],” *Communications Magazine, IEEE*, vol. 47, no. 8, pp. 1 –4, august 2009.

[2] M. Eichelberg, A. Hein, F. Büsching, and L. Wolf, “The gal middleware platform for aal,” in *e-Health Networking Applications and Services (Healthcom), 2010 12th IEEE International Conference on*, 2010, pp. 1 –6.