Poster abstract: Gumsense - a high power low power sensor node

Kirk Martinez, Philip Basford, Joshua Ellul, Robert Spanton

Electronics and Computer Science, {km, pjb08r, je07r, rds}@ecs.soton.ac.uk
University of Southampton, UK.

I. Introduction

The development of increasingly complex algorithms for sensor networks has made it difficult for researchers to implement their design on typical sensor network hardware with limited computing resources. The demands on hardware can also mean that small microcontrollers are not the ideal platform for testing computationally and/or memory intensive algorithms. Researchers would also like access to high level programming languages and a wider range of open source libraries.

To address this problem we have designed and implemented an architecture, Gumsense [2] which combines a low power micro-controller (8MHz MSP430) with a powerful processor (100-600MHz ARM) on a Gumstix board [1] running Linux. This Open Embedded OS supports a wide variety of programming languages, package management and development tools. A similar hybrid approach was also used in the LEAP platform [3]. The microcontroller wakes up frequently to manage tasks such as activating sensors and gathering data. The intended use-case is to power-up the ARM board and storage only during the brief periods it is needed, for example performing computation or communication.

Fig. 1 shows the basic architecture of the platform with I^2C being used to communicate between the MSP430 and ARM platforms. The Gumsense board provides adjustable instrumentation amplifiers to allow the connection of a wide range of sensors as well as a USB port for additional connectivity.

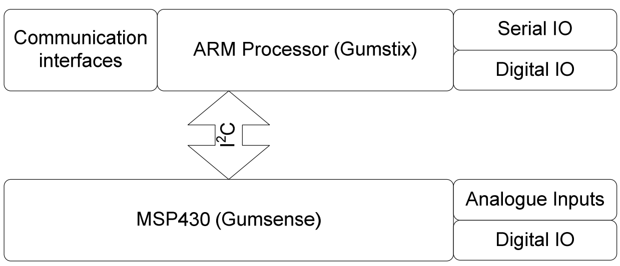

Fig. 1. Gumsense architecture

II. Hardware Design

The Gumstix does not have a suitably low power sleep mode so when it is not in use it is completely powered down, unlike the use of a sleep state used by LEAP[3] meaning that energy cost of sleeping is down to the MSP430 and some support circuitry. The Gumsense board comprises power control, I/O pins and the microcontroller with its analogue and digital I/O. The MSP430 is responsible for sensor reading, setting the next wakeup and controlling power rails according to the schedule set by the Gumstix. The MSP has 60k of flash memory which is used as a ring buffer for sensor values, which are then fetched by the Gumstix when it next wakes.
The design is shown in Fig. 2. In the top view of the 104mm wide PCB the Gumstix is visible in the upper half, surrounded by headers which make I/O and controlled power available for sensors and a radio. The underneath shows the potentiometers for gain settings, MSP430 and backup battery.

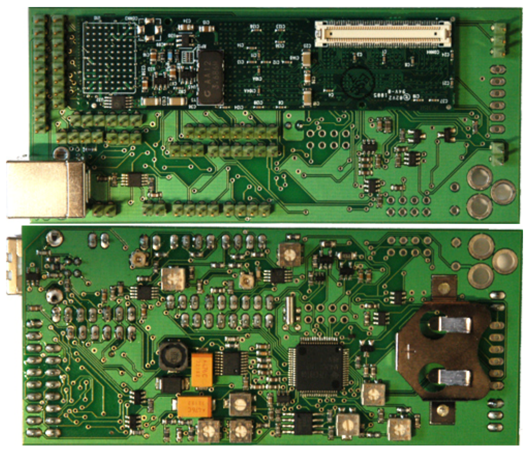

Fig. 2. PCB showing Gumstix (darker pcb on top picture) and underside view (lower).

III. Power Consumption

Typical sensor nodes will employ a periodic sleep-resume cycle in order to prolong node lifetime. These nodes have three different power levels as shown in Table 1, the majority of the time the node will be in state 0, with sensor readings being taken in state 1. State 2 will only be used for short periods in which communication or high powered computation is being performed.

Table 1 Power draw of a node in different states

ID	State	Power (mW)
0	Sleep	0.00720
1	MSP Active (Gumstix off)	52.4
2	MSP & Gumstix powered	900

As an example if the board is used to take 4 measurements (each taking 10 seconds) an hour, and then wakeup the ARM board for 2 minutes a day for processing as well as performing 15s of communication a day then with a 12volt 2AH battery the predicted lifetime is in excess of a year. This estimate shows that although the Gumstix energy use is a large part of the overall budget, real system deployments are possible.

IV. Programming

In order to make the system easy to control by Linux programmers the interchange of schedule and data has been made as simple as possible. The MSP430 maintains a list of recurring jobs with a set of analogue inputs, set of power rails to manage and sampling interval. The Gumsense also provides an adjustable delay time between the powering on of sensors and taking the readings to allow voltages to settle.

Communication between the Gumstix and Gumsense modules is achieved using nine registers on the I^2C bus with sensor readings being provided in 32byte blocks. Functions to manage this communication are provided as a kernel module for Gumstix with additional bindings to allow access from scripting languages.

V. Possible deployment scenarios

There are three different deployment scenarios for which this platform is suited. The first as a gateway node for other sensors; secondly as a mesh of cluster heads to form a two-tiered network; finally as a complete network of homogeneous sensors. In the first scenario the gumsense platform can be equipped with an interface to allow communication with the sensors, the node could then perform any calculations required and forward the data on. In the second scenario the sensors can provide in network processing before transmitting the data to researchers, the amount of storage available on the platform would also allow data to be backed up over multiple nodes. Finally if the entire network consists of Gumsense boards then it can provide a useful test bed for algorithms, as powerful debugging environments can be run on the individual nodes if required.

A complete test of the system was carried out in Iceland in the summer of 2008 (see Fig. 3) when a system was used as the gateway for subglacial probes [5]. This included control of a dGPS and used a GSM/GPRS unit for communications home. The improved sleep power meant the system could run through the winter easily and batteries were seen to remain full thanks for solar/wind charging. It has been running since August 2008.

VI. Conclusion and future work

The design provides a powerful processor (100-600MHz) with 64-128MB of memory (additional storage can be added in the form of SD or compact flash cards if required) while allowing for real deployments where the data sampling is generally handled by a low power microcontroller. A test deployment within the department will be used to develop this concept as a prototyping platform for researchers. These can use WiFi cards for convenient configuration and easy updating as well as a low power radio for research.

At the moment the MSP430 runs a custom operating system, however this could be changed to tinyOS2[6] or Contiki[7]. If done in a manner adhering to the I^2C protocol currently in use then this change would be transparent to the higher levels.

This platform overcomes a number of limitations of existing low power sensor networks by combing very low sleep power consumption (approx 72µW) with sufficient computational capability to support, for example, complex sensor network behaviour algorithms, asymmetric cryptography and autonomous experimentation algorithms.

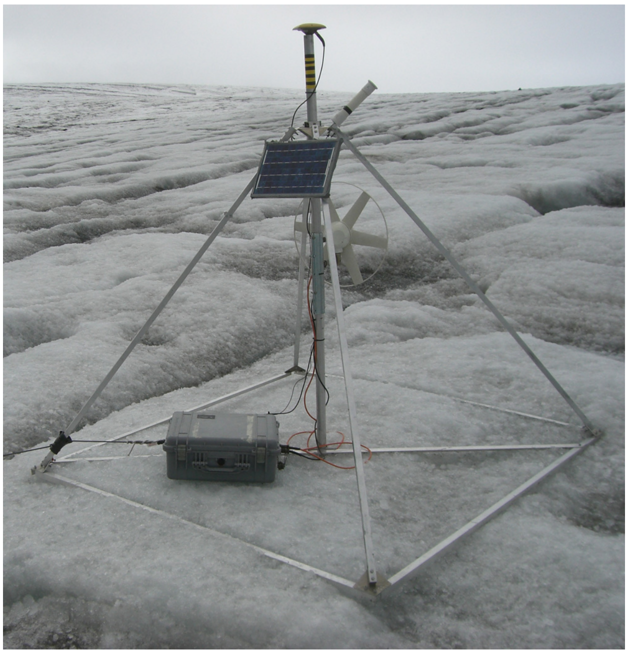

Fig. 3 deployment on the Skalafellsjokul glacier in Iceland

Acknowledgments

Thanks to Royan Ong (Monash University Malaysia) for the version 1 design, Klaus Peter Zauner for many discussions, Mareike Bauer for prototype assembly and Jeff Gough and Tom Bennellick for work on the Iceland deployment. This research was funded in part by the EPSRC project Glacsweb, The Nuffield Foundation (URB/34245), and the ECS dept.

References

[1] Gumstix: www.gumstix.org
[2] Gumsense: gumstix.ecs.soton.ac.uk/wiki/index.php?title=Gumsense
[3] D. McIntire, K. Ho, B. Yip, A. Singh, W. Wu, and W.J. Kaiser. "The low power energy aware processing (LEAP) embedded networked sensor system", *Proceedings of the fifth international conference on Information processing in sensor networks*, pp. 449–457, 2006.
[4] Martinez, K., Hart, J. and Ong, R. "Environmental Sensor Networks", *IEEE Computer*, 37 (8). pp. 50-56. 2004.
[5] Martinez, K., Padhy, P., Elsaify, A., Zou, G., Riddoch, A., Hart, J. K. and Ong, H. L. R. (2006) Deploying a Sensor Network in an Extreme Environment. In *Proceedings of Sensor Networks, Ubiquitous and Trustworthy Computing*, pp. 186-193, Taiwan.
[6] TinyOS2 www.tinyos.net/
[7] Contiki www.sics.se/contiki/