NODES: A NOVEL SYSTEM DESIGN FOR EMBEDDED SENSOR NETWORKS

S. Neema, A. Mitra, A. Banerjee, W. Najjar, D. Zeinalipour-Yazti, D. Gunopulos, V. Kalogeraki
University of California - Riverside
{sneema, amitra, anirban, najjar, csyiazti, dg, vana}@cs.ucr.edu

ABSTRACT

In this paper we present the RISE (RIverside SEnsor) platform, a novel system design for embedded sensors built around a System-on-Chip device interfaced with a large external storage memory in the form of off-the-shelf SD (Secure Digital) Card. We describe the hardware and software structure of RISE, which supports the standard TinyOS and NesC environment. We demonstrate that significant energy savings together with the additional benefits of reduced complexity and increased ease of use are achieved by adopting the sense and store methodology in which we transmit only the data of interest.

1. INTRODUCTION

Flash memory allows for a very low energy storage budget. Also, higher levels of device integration at low cost and size now provide us with powerful, compact and economic single chip solutions for our sensory and communication needs. The RISE (RIverside SEnsor) platform leverages these trends essentially by employing the Chipcon^{TM} CC1010 device. The CC1010 is a true single-chip RF microprocessor / transceiver with an integrated high performance 8051 microcontroller that benefits from a very wide array of publicly available software including compilers and debuggers.

Our contributions include porting of the most prevalent design environment, the TinyOS (version1.1) and NesC (version 1.2alpha1), facilitating easier and modular programming, interfacing of an SD-Card [4], and a carbon dioxide sensor and developing the reactive methodology of query based response on large datasets stored locally on the nodes. We are currently developing a simple file system to support efficient access to the memory. The open source SDCC (small device C compiler) is used to generate the 8051 compatible C code. The platform lowers communication costs significantly by reducing the overall traffic to include only the subset of the data that is of interest to an application querying the node. We calculated that the energy cost of writing to flash memory is less than 10% of the RF transmission cost.

2. THE RISE SENSOR PLATFORM

The RISE node is based on a modular design using widely available commercial off-the-shelf technology. RISE was designed from the ground up, to serve a wide spectrum of applications demanding flexibility, and versatility as opposed to power thriftiness rather than power thriftiness. Figure 1 shows the block diagram of the RISE sensor platform, hashed boundaries indicate the components that are currently under development.

2.1. The Chipcon CC1010 SoC

The CC1010 is a true single-chip UHF transceiver with an integrated high performance 8051 microcontroller with 32 KB of Internal flash memory. The CC1010 together with a few external passive components constitutes a powerful embedded system with wireless communication capabilities. The features of the Chipcon SoC [1] are as follows:

- High-Performance and low-power optimized 8051-core microcontroller with idle and sleep modes for reduced power consumption. The system can wake up on interrupt or when ADC input exceeds a set threshold.
- Fully integrated UHF RF Transceiver with programmable frequency and output power and low current consumption. It also has fast PLL setting for frequency hopping protocols, Manchester encoding and decoding in hardware and also RSSI output which can be sampled by the on chip ADC.
- 32 KB of nonvolatile Flash memory with programmable read and write locks for software security, 2KB + 128 Byte of internal SRAM.
- Peripheral features include three channel, 10 bit ADC, programmable watchdog timer, 32 KHz real time clock, two programmable UARTs, SPI interface, two counters and pulse width modulators, 26 configurable general-purpose I/O-pins, random bit generator and H/W DES encryption.

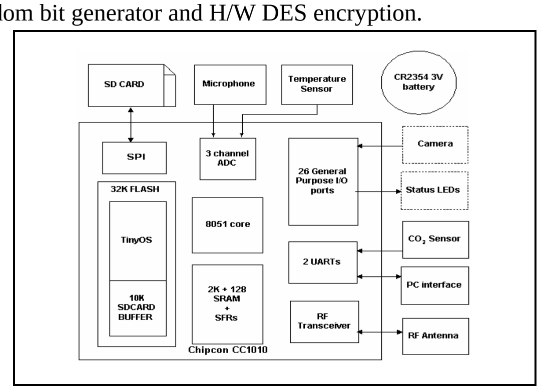

Fig1. The RISE platform block diagram.

2.2. TinyOS on RISE

TinyOS provides an event based execution and a constrained runtime environment ideally suited to, and the present de-facto standard operating system for sensor network systems. The modular design of TinyOS is amenable to simple applications not requiring complex hardware resources. The latest stable versions of TinyOS, tinyos-1.1, as also the NesC compiler, NesC v1.2alpha1, were ported on to the RISE platform. We have used the *SDCC* (small device C compiler), and integrated it such that the single make of the application creates a hex file ready to be burned on the on-chip flash.

2.3. Interfacing the SD-Card

SD-Cards are commercial low power flash memory storage devices. The connection from the platform to the SD-Card involves dedicating four I/O pins (Clock, Data In, Data Out, Chip Select) from the microcontroller. The micro-controller transfers data using the SPI protocol by linking to a wrapper component that provides read/write/erase macros to facilitate data transfer to and from the SD-Card. Each write transaction to the card involves writing a 512 byte block of data, while reads may be arbitrarily sized up to a maximum of 512 bytes, at a maximum rate of 3Mbps. Write efficiency is enhanced by first buffering the sensor data to the on-chip buffer. A full buffer initiates a data flush from the on-chip flash to the SD-Card. We are currently working on developing tiny access method structures, which allow for efficient sorted and random access to local data. NAND Flash memory, which is the most prevalent type of flash memory used in most flash cards, has some distinct characteristics which can be summarized as following: i) Every block can only be written a limited number of times ii) Writing to a block requires that the block is already deleted and reading can be performed at any granularity.

3. ENERGY EVALUATION

We calculate the energy gains that can be achieved by use of the sense and store methodology and contrast and compare storing a byte of data, as opposed to blindly sending it over the wireless network. For a single hop, energy consumed by the radio to transmit a single byte is .164mJ, and to receive is .08mJ. On the other hand energy consumed by the SD Card to write a byte is 0.0015 mJ and to read is a mere 0.00005 mJ.

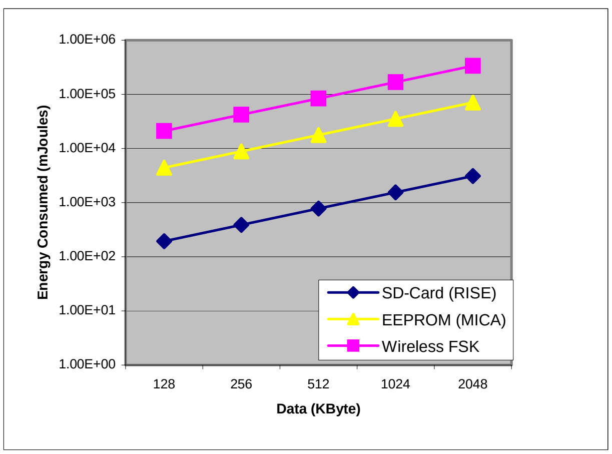

Fig2. A comparison of the amount of energy expended to transfer data via the wireless (pink trace) interface Vs Storing it on the on-chip EEPROM (yellow trace) and the on-board SD-Card (Blue trace).

Note the considerably higher data transfer rate (800Kbits/s) of the SD Card as compared to the radio (9.6 Kbits/s), implies that the read or write operation on the flash occurs for a much shorter time and hence the energy expense for flash operations is less than 10% of the radio operations.

4. THE PARADIGM OF SENSE AND STORE.

4.1 Rationale behind Sense and Store

Typically the sensor network applications generate vast amounts of temporal data over very long time intervals that need to be collected and processed in a power efficient manner. Of the vast amounts of data generated, applications are typically interested in only aggregates; hence database approaches have been advocated [3]. We discuss how employing large memories on the sensor nodes can fulfill the requirements of these applications; in an overall methodology we call "sense and store". During normal conditions, the sensory data remains predictable with gradual changes, and hence is not of particular importance. But often the question that we are interested in answering is of the form *"When did we have the last 5 highest temperature readings?"* or *"What was the average temperature last week?"*, or consider our present scenario wherein sensors are deployed in a forest to track climate conditions, and there is a forest fire. In this case the data set of interest is the one, say up to a month leading to the event. Hence, in most cases the queries request only a part of the sensed data, sorted by either the index (the timestamp in our case) or the value and the data of interest in just a fraction of the sensed data. Typically these sets involve temporal and top-k class of queries.

4.2 Providing Local Access Methods

Efficiently evaluating temporal and top-k queries in our framework requires efficient access to the data that is stored on the "external" flash memory. Therefore we plan to deploy certain access methods (indexes) directly at the sensor nodes. These access methods will serve as primitive operations for the efficient execution of a wide spectrum of queries. Since the flash card on each node v_i can only hold m pages ($o_{i0}\ldots o_{im}$) the available memory is organized as a circular array, in which the newest o_{ij} pair always replaces the oldest o_{ij} pair. Note that this *sorted page organization* allows each sensor to have random or sorted access by timestamp in $O(1)$, without the requirement of any index. Additionally, an index will enable the local execution of queries by value e.g. "Find when was the temperature equivalent to 95F". This will allow sensors to locally filter results before these are transmitted back to the sink.

5. CONCLUSIONS

Overall the benefits of sense and store are perceptible in sensor networks deployed for monitoring time series data with predictable queries and where post mortem analysis and collection of data would be undertaken. Keeping these requirements in mind, we are currently calculating and storing the running average, daily min-max, daily average, in the on-chip flash, which are then stored into the SD-Card past elapse of twenty-four hours (RISE incorporates timekeeping, thanks to the real time clock on the platform [1]). Currently under implementation are the top-k and range queries, which would be completed after due field runs to determine a practicable value of the query window, along with generating diagnostics information (battery voltage) from the ADC.

6. REFERENCES

[1] http://www.chipcon.com/

[2] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister, "System architecture directions for networked sensors", In *Proceedings of ASPLOS*, pages 93–104, Boston, MA, USA, Nov. 2000.

[3] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TAG: A Tiny AGgregation service for ad-hoc sensor networks", *Fifth Symposium on OSDI '02*, Boston, Dec. 2002.

[4] www.sandisk.com/pdf/oem/ProdManualSDCardv1.9.pdf, Sandisk SD-Card Product Manual v1.9.