SADmote: A Robust and Cost-Effective Device for Environmental Monitoring

Atis Elsts[1][2], Rihards Balass[1], Janis Judvaitis[1],
Reinholds Zviedris[1][2], Girts Strazdins[1][2], Artis Mednis[1][2], and Leo Selavo[1][2]

[1] Faculty of Computing, University of Latvia,
19 Raina Blvd., Riga, LV 1586, Latvia
[2] Institute of Electronics and Computer Science,
14 Dzerbenes Str, Riga, LV 1006, Latvia
`atis.elsts@edi.lv,{rihards.balass,janis.judvaitis}@lais.lv`
`{reinholds.zviedris,girts.strazdins,artis.mednis,leo.selavo}@edi.lv`

Abstract. Time to deployment for wireless sensor networks could be reduced by using commercial sensor nodes. However, this may lead to suboptimal flexibility, power consumption and cost of the system. Our pilot deployment for precision agriculture and fruit growing research showed similar conclusions and outlined the design decisions leading to SADmote: a new sensor node for environmental monitoring. It was evaluated both in the lab and field, showing improved energy consumption over commercial solutions such as Tmote Sky and Waspmote.[*]

Keywords: Hardware design, Environmental monitoring, Precision agriculture, Sensor networks

1 Introduction

Wireless sensor networks (WSN) – systems of numerous small and resource constrained devices – enable a new scope for applications and research in environmental monitoring and agriculture, due to the increased spatial and dynamic resolution, remote accessibility, reduced deployment and maintenance costs. Several WSN systems with these features have been demonstrated, including microclimate monitoring of Redwood trees [11] and light environment under a shrub thicket [9]. Agricultural application examples include plant monitoring [5] and farm animal tracking [12]. However, novel technologies and cost-efficient components may improve these WSN applications by providing additional flexibility, larger scope and longer system lifetimes.

SADmote is a sensor node designed with the above-mentioned features in mind. It is a miniature device developed for micro-climate monitoring in agriculture and environmental applications. SADmote is designed around MSP430-F1611 controller by Texas Instruments due to its low power features. For wireless communication it uses MRF24J40 radio transceiver by Microchip, because of its

---

[*] The original publication is available at www.springerlink.com

low cost and IEEE 802.15.4 compatibility. We wanted the adaptability, simplicity and efficiency offered by a custom, application-specific hardware design. In this way we were able to make SADmote budget-efficient and robust, minimizing the number of points of failure. On the other hand, the design of SADmote is generic enough to make the mote lucrative for other uses in research, development and teaching.

The paper is structured as follows. In Section 2 we describe our pilot deployment in 2009 and the motivation for SADmote. Section 3 briefly touches on related work, Sections 4 and 5 describe the hardware and software aspects of our solution. We present the experimental results in Section 6, and conclude the paper with Section 7.

2 Motivation and pilot deployment

The researchers at Latvia State Institute of Fruit-Growing (LSIFG) [2] were looking for a way to record and store micro-climate data at various points of the institute's orchard. Solar radiation and humidity measurements in fruit tree canopies, strawberry and raspberry plantations during the vegetation season were of particular interest, because shady and wet environment can lead to development of diseases such as apple scab (*Venturia inaequalis*). In addition, fruits (in particular, apples) that have received more sunlight during their development are more colored. Since fruits with brighter colors (red rather than green) are more visually appealing, they are also more marketable. Therefore, finding the exact correlations between precise, cumulative and localized micro-climate data from the one side, and development of diseases and fruit coloring from the other side would provide valuable recommendations regarding protective plant covers and tree canopy forming techniques.

Typically the micro-climate data from the orchard was collected manually, either with data-loggers or a portable device. These methods are labor intensive and offer limited resolution. This could be improved significantly by deploying a sensor network. Our pilot research experiment in 2009 included eight Tmote Sky sensor devices with built-in sensors for humidity, total solar radiation, and photo-synthetically active radiation. The close proximity of a weather station provided a constant power source for the sink. The data was collected by a single-hop sensor network. This network was active from August 17th until September 15th, 2009. A total of 661 hours of real time data included 405061 measurements collected at 20 second intervals.

From this initial experiment several lessons were learned. Firstly, we become convinced that custom sensors are necessary for this application. The default light sensors located on Tmote Sky are not suitable for agricultural monitoring due to their limited dynamic range. During sunlight the sensors become saturated, yielding inaccurate data about the amount of radiation received by the plants. Also, the agroscientists desired higher resolution than Tmote Sky could provide with the 12-bit ADC. In addition, the light wavelength sensitivity curve of the PAR sensors was different from the actual plant response. Although the SHT11 sensor had sufficient precision, it was located on-board, inside the weatherproof box, where humidity and temperature values are significantly different from the conditions outside, in the orchard.

Secondly, a significant proportion of the pilot measurements were lost due to erratic radio links. Packet delivery failures, being frequently encountered in WSN, ought to be taken in account as one of design considerations. If the data was stored on the sensor nodes, using a flash storage chip or card, it could be collected manually at a later time.

Thirdly, a network with multiple hops is necessary. The ability to attach an external antenna and use radios with higher transmission power could be useful as well, at least for a second-tier, long-haul network.

Regarding the runtime assurance – visual means such as a blinking LED are necessary to confirm the sensor network health in-field. This may sound as a bad design decision due to the extra energy consumption, but discussions with domain scientists convinced us that the extra assurance is worth the price.

Last but not least, the sensor network should be energy efficient. The domain scientists cannot worry about changing batteries every few weeks. The devices should be able to function for months without interruptions. In the ideal case, they should function during the whole vegetation season, which in Latvia lasts from April to October.

3 Related work and design considerations

To face the challenges outlined by our pilot study one could purchase an existing sensor platform and extend it with appropriate sensors. This has the benefit of simplicity, however, this may also result in suboptimal extendibility, energy consumption and cost.

A TelosB-compatible [8] sensor device, such as Tmote Sky, is a popular first choice due to its ultra-low energy consumption, versatility, and reasonable 100$ price range. Tmote Sky has TI MSP430F1611 MCU, IEEE 802.15.4 compatible radio transceiver, 1MB external flash memory chip, and 16 pin expansion port with ADC and digital inputs. However, the size of the flash may be insufficient for long-term micro-climate data. The storage capacity could be increased by attaching a larger data storage entity, such as an extension shield with SD card slot. Unfortunately, the MSP430 SPI interface is not fully exported. Another drawback is the 2.7V power voltage limit for successful flash data storage, limiting the useful battery lifetime.

Another option was to use a commercially available off-the-shelf (COTS) embedded device such as Arduino [1], with accompanying extension shields for sensors and communication. The cost would be reasonable. However, the target application would require many extension shields increasing the cost and complexity. Arduino has significant drawbacks in comparison with Tmote Sky – higher energy consumption and less resources (RAM, flash memory, IO ports). Therefore, Arduino is as platform more suited for hobbyist projects and rapid prototyping rather than for long-living research sensor networks.

A reasonable alternative is a COTS sensor mote outfitted for agricultural monitoring such as Waspmote [3] platform by Libelium due to a reduced time to deployment. However, the cost is much higher: 135 Euro for the mote and 250 Euro for Agriculture PRO extension board, not counting additional equipment, such as custom rechargeable batteries and enclosures. Also, Waspmote has higher energy consumption. It is based on Atmega1281 MCU – one of the largest and most energy hungry MCUs in this family. However, these drawbacks are partially countered by the expected simplicity and ease-to-use. Libelium provides simple API and software library for radio communication and for accessing various sensors. In the end, we purchased eight Waspmotes with corresponding extension boards and sensors, and five more Waspmotes without sensors.

The final option is to design a custom hardware. In our case, by leaving out the components nonessential to our application (such as USB interface, LDO, on-board sensors), the cost and energy efficiency can be optimized. Also, by limiting the number of components the design becomes more robust.

4 SADmote hardware design

SADmote components. Taking in account the issues outlined in previous section, we decided to build our own hardware solution from scratch – SADmote (Figs. 1 and 2).

Two other sensor devices were selected for comparison with SADmote (Table 1): Tmote Sky as TelosB-compatible design that has stood the test of time, and Waspmote that is marketed as suitable for agricultural monitoring.

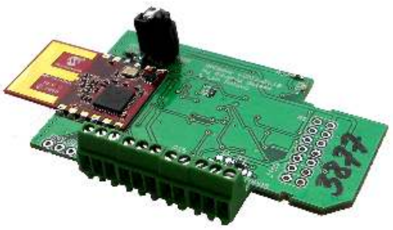

Fig. 1. Top of SADmote v2

Fig. 2. Bottom of SADmote v2

We chose MSP430F1611 as the MCU for SADmote. It has ultra-low energy consumption, is supported by most of the WSN operating systems, and is sufficient for our current needs. Should the resource requirements increase, MSP430x2xxx series and MSP430x5xxx series provide powerful alternatives, and might be considered for the next version. When compared to Atmega microcontroller based devices such as Waspmote, more efficient duty cycling can implemented, because sleep timer has 16-bit range, as opposed to the smaller range provided by 8-bit Atmega MCUs. Because of these reasons and the support for DMA memory access, MSP430 was also selected for other recent sensor platforms such as Epic Mote [6].

As for the radio chip, 802.15.4 PHY compatible hardware was our choice to ensure interoperability with other devices, and as a well-tested and developed standard. DSSS frequency hopping used by this standard helps to achieve the robustness we are looking for due to the additional noise-tolerance. 802.15.4

	SADmote	**Tmote Sky**	**Waspmote**
MCU	MSP430F1611	MSP430F1611	Atmega 1281
RAM size	10KB	10KB	8KB
Flash size	48KB	48KB	128KB
Op. voltage	2.4–3.6V	2.7–3.6V	3.3–4.2V
802.15.4 radio	MRF24J40	CC2420	Xbee
External memory	AT25DF161	M25P80	miniSD card
Ext. memory size	2MB	1MB	2GB max
LED	One	Three	Two
Onboard sensors	None	Light, Temp., Humidity	Accelerometer
Price	< 100$	< 100$	200$

Table 1. Comparison between sensor devices

MAC support (CCA, LQI, CRC, hardware security) is good to have as well. Even though 2.4 GHz frequency band has the drawback of being heavily absorbed by tree foliage and other obstacles, these additional benefits and faster prototyping time outweight using a lower frequency band for the first versions of SADmote. We did not plan to use these versions for any long-range radio links, and for short-range links 2.4 GHz frequency band is usable even for agriculture applications.

We considered CC2420 transceiver chip (2.1–3.6V operational voltage, 18.8 / 17.4mA Tx/Rx current consumption, -95dBm sensitivity) and MRF24J40 transceiver based module (2.4–3.6V, 18 / 22mA, -91dBm). Despite CC2420 having slightly better characteristics, we chose Microchip's radio for the first versions of SADmote due to its availability, and the fact Microchip provided a ready-to-use radio module (MRF24J40MA, -94dBm sensitivity). If the need arises, the module can be replaced with other by the same manufacturer: MRF24J40MB-I (higher Tx power) or MRF24WB0MB (external antenna connector). Ready-to-use radio modules with Texas Instruments chips also exist, for example, Amber Wireless AMB2720, but it has approximately two times higher price.

SADmote peripheral components include 2MB external flash memory chip AT25DF161. This chip not only supports erasing data in smaller units (4KB compared to 64KB minimum on Tmote Sky), but also has a version that supports data writing with operation voltage as low as 2.3V. The future versions of SADmote will feature an optional mini-SD card slot due to easier data access.

SADmote has 16-bit Analog-to-Digital converter for attaching external analog sensors that require higher resolution than the MCU built-in 12-bit ADC. SADmote provide one to two such channels using Texas Instruments ADS1114 or ADS1115, depending on the mote revision.

SADmote features DS2401P+ serial number chip, which holds an unique 64 bit registration number, useful for automatic generation of a network address and unique identification of the mote.

Other on-board components include: JTAG expansion ports, 32 768Hz oscillator, resistors and capacitors. To connect the mote to a programmer, either JTAG interface or SD-card compatible interface can be used. On mote's side, the SD card connector pads are simply plated on PCB. We chose it as the simplest and most cost effective solution, not requiring any extra components. To the best of our knowledge, no other WSN motes use an interface like this.

One of our design objectives for longer lifetime was to create a device that would work even if the batteries are already running low. We note that SADmote is fully operational at voltages as low as 2.4V. This means SADmote has a larger energy budget to work with, when compared to Waspmote or even Tmote Sky. The current limiting factor for power voltage reduction on SADmote is radio, while the limiting factor on Tmote Sky is flash memory.

All of the components are COTS and are soldered on a two-layer PCB.

SADmote sensors. Separate sensor board was developed for two different digitally controllable light sensors, Intersil ISL29003 and Avago Technologies APDS9300. Both sensors can be attached to the SADmote by using I2C expansion ports. The design of the extension boards was keep simple to increase robustness of the system. Being out of the protective enclosure, the boards are not as well in insulated and protected as the rest of the systems. Any additional electronic components would increase risk of malfunctioning.

SHT75 humidity and temperature sensor and a high-precision solar radiation sensor such as SQ-110 can be attached to SADmote as well. The sensor modules are attached to the board using terminal block headers with screws. This allows to replace the sensors more easily in field conditions, compared to soldering them directly to the board.

Dimensions of the SADmote. Another of our design objectives was to make SADmote small and light enough, so that it could be fitted in a compact enclosure and attached to small branches or bushes. The current dimensions of SADmote are 72 x 37 x 29 mm (length x width x height), compared to 81 x 32 x 20 mm for Tmote Sky and 73 x 50 x 22 for Waspmote (including attached batteries). Again, the radio module was the limiting factor. SADmote PCB cannot be made narrower without violating the design constrains for the antenna of the MRF24J40MA. The limiting factors for height are the battery holder and the headers. In summary, SADmote is about the size of Tmote Sky, and notably smaller than Waspmote, though with increased height.

5 SADmote software design

MansOS operating system [10] was used to program the motes. This WSN OS is developed at the University of Latvia and Institute of Electronics and Computer Science (IECS). Driver support for SADmote chips is included in latest revisions of MansOS. To the best of our knowledge, there are no other open source drivers of MRF24J40 radio chip for MSP430 architecture. All other chips, except the MCU itself, have no support in TinyOS either.

Simple and easy-to-learn API. MansOS was designed to be easy-to-use for beginners MansOS uses plain C and UNIX-like abstractions, which should be familiar to many non-WSN programmers. This usability allows use to hope the number of programming errors is going to reduced.

Over-the-air reprogramming. Through a network management protocol that is included in MansOS the user can reprogram a mote wirelessly. Partial reprogramming is also possible. This feature allows us to fix bugs and upgrade software versions without requiring physical access to the motes.

Networking stack. MansOS has built-in networking stack, which features a simple CSMA MAC protocol with optional acknowledgments. They can be turned on to make data delivery more reliable. It also features a distance-vector routing protocol with built-in time synchronization. On top of these mechanisms, UNIX-like sockets are implemented.

Simple and robust multithreading. MansOS supports preemptive multithreading. As opposed to other WSN operating systems like TinysOS or Contiki, MansOS can continue to function even when the user has included an infinite loop in his code, because the kernel is able to preempt this defective code. The price one has to pay is additional code size (1118 bytes) and additional RAM usage (32 bytes + the size of memory required for at least one thread's stack, e.g. 80 bytes). However, even with this additional overhead, MansOS applications can have smaller code size than TinyOS applications. For example, Blink on MansOS with threads uses 2376 bytes code and 52 bytes in RAM + stack overhead, compared to 2586 and 52 bytes respectively for TinyOS Blink.

6 Evaluation

6.1 Energy consumption

As part of the experimental evaluation we measured current consumption on a SADmote and other sensor devices (Table 2). The results clearly show that SADmote has extremely efficient sleep mode when compared to other devices. In active mode it has current consumption comparable with Tmote Sky, while radio communication (especially for Rx) is slightly less energy-efficient.

Mode		**SADmote @ 3.0V**	**Tmote Sky @ 3.0V**	**Waspmote @ 4.1V**
Active	measured	7.5mW	6.9–8.4mW	38.5mW
	by datasheet	–	7.2 mW	>29.7mW[1]
Sleep	measured	90.0μW	210μW	3066.8μW
	by datasheet	–	163.5μW	>204.59μW[1]
Radio Rx	measured	76.5mW	63.9mW	214.5mW
	by datasheet	–	65.4mW	>194.7mW[1]

Table 2. Energy consumption of sensor devices, without sensors

Waspmote is clearly the least efficient of all three. Regarding sleep mode, we were surprised by the inconsistency between current consumption values declared in datasheet and the values we measured. The software we used was the low power demo example provided by Libelium. We note that Waspmote has an even lower power consumption mode ("Hibernate mode"), however, it requires use of an auxiliary battery.

We also measured energy consumption on SADmote and Tmote programmed with our agriculture application during one work cycle (sensor measurement, write to flash, send to radio) and estimated the consumption for longer intervals (Fig. 3). The technique used was to measure voltage drop across a 4.7Ω resistor,

---

[1] At unknown voltage, assuming 3.3V minimum

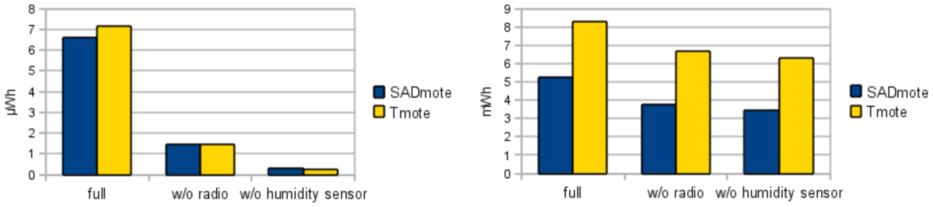

Fig. 3. Measured energy consumption per sensor reading, μWh (left), estimated energy consumption per day, mWh (right)

using two op-amps and a separate mote with 16-bit ADC for sampling the voltage values with the average frequency approximately 600 Hz.

At the moment we have not managed to put the devices in the most efficient sleep mode when sensors are attached, so the values in the right figure are theoretical. The energy consumption in sleep mode is estimated as the sum of mote's consumption without sensors attached (Table 2) and APDS9300 light and SHT75 humidity sensor standby mode consumptions from their respective datasheets (45μW and 5μW). Assuming that SADmote indeed has more efficient sleep mode even with sensors attached, it as well has higher long-term energy efficiency for this application than Tmote, and both SADmote and Tmote could function for several years from two ordinary 2800 mAh AA batteries.

6.2 Radio communication

We conducted several radio communication tests. To find out the SADmote radiation pattern, we performed measurements using a spectrum analyzer. Radiation from ten SADmotes was measured, each in eight positions (motes horizontally, with battery holders down). As the results show, the most efficient direction on average is with antenna positioned directly towards the receiver (corresponding to 0 degrees in Fig. 4).

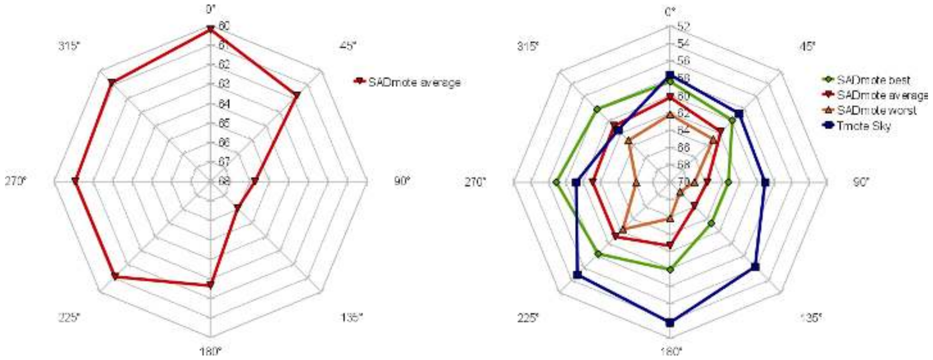

Fig. 4. Measured radiation pattern (left), in comparison with Tmote Sky (right)

For comparison we also measured a Tmote Sky mote using the same setup. On the average, Tmote had 4.83 dB higher radiation, and SADmote performed better only in a single direction. We expected this result, but were surprised by the magnitude of the difference. It can be partially explained by the antenna on MRF24J40MA module being less efficient than the one Tmote has. Texas

Instruments application note [4] lists 3.3 dB maximum gain of the antenna Tmote uses, while MRF24J40MA datasheet [7] mentions only 2.09 dB gain.

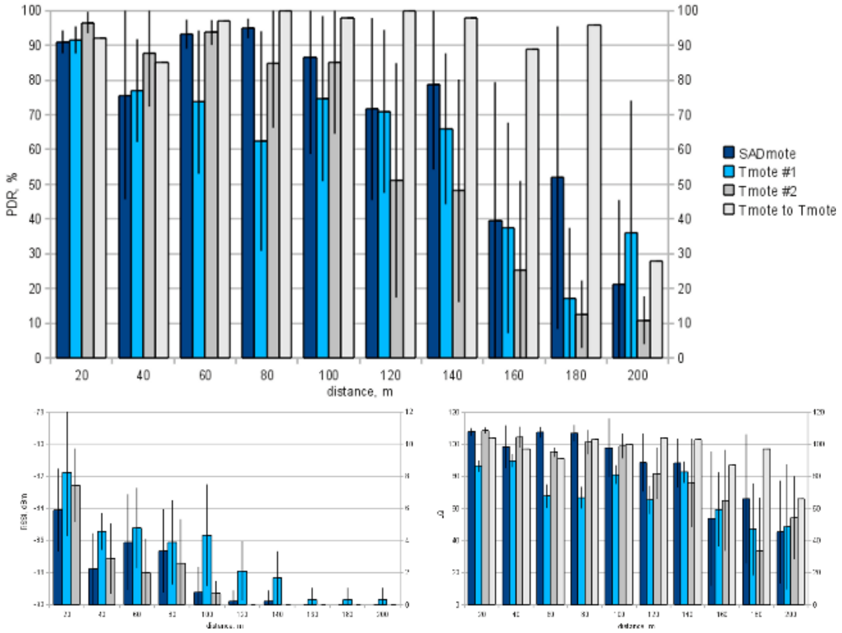

Fig. 5. SADmote packet delivery ratio (PDR) (top), RSSI (bottom left) and LQI (bottom right) in comparison with Tmote Sky

Afterwards we measured communication range outdoors (Fig. 5), using the most efficient antenna position for SADmote. Nine co-located SADmotes were used as receivers (seven for the third test), and a single SADmote (having average results in radiation pattern test) as a transmitter. The results were compared with two Tmotes as transmitters. In the third test, a single Tmote was also used as a receiver.

An interesting observation is that despite higher signal strength (also higher RSSI), using a Tmote did not result in higher packet reception rates. The results indicate that LQI on SADmotes, unlike RSSI, correlates highly with PDR.

The results demonstrate that SADmote can be used reliably for short distance radio communication, and validate Microchip's claimed 400 feet (approximately 120m) communication range for MRF24J40MA radio module. On the other hand, Tmote-to-Tmote radio communication is clearly superior. Taking these results in account, we have decided to consider other radio communication alternatives for future versions of SADmote.

6.3 The initial field test of SADmote

A single SADmote was left in the testing site (LSIFG orchard, Dobele, Latvia), where it gathered data from SHT75, ISL sensors, and internal voltage data with five minute interval. The mote functioned from August 12th until September 22th, for 979.5 hours, during which 11734 measurements were recorded.

We can estimate the energy used by observing that during this time the operating voltage of the mote declined from 3V to a little more than 2.2V. Surprisingly, even at this point the mote was still able to measure and record data in external flash! We note that the mote spent majority of its energy in blinking the on-board LED, and that its lifetime could greatly extended by turning the LED off.

Another pleasant find was the precision of the time accounting system, both hardware and software. According to the internal time accounting, mote was active for 979.334 hours. Compared to the real, externally observed lifetime of 979.5 hours, we can see that relative error is only 0.0017%.

6.4 SADmote sensor network deployment

Test description. A larger-scale field test took place in LSIFG orchard from September 19th until November 2nd. The main objectives of this test were to determine the feasibility of a medium size, medium lifetime sensor network formed by SADmotes, test interoperability between SADmote with Waspmote, and compare robustness and lifetime of SADmotes with Waspmotes in field conditions.

The sensor network deployed (Fig. 6) consisted of twelve SADmotes with sensors, as well as five Waspmotes which formed the backbone of the WSN and were used for data forwarding. Each SADmote measured light, humidity and temperature with five minute interval sent the measurements to the nearest Waspmote.

SADmotes were placed either on ground or approximately a meter above it (Fig. 7). All motes were put in waterproof boxes. SHT75 sensors, as well as light sensors were fastened outside of the boxes, separated from the environment only by a protective lacquer covering.

Fig. 6. SADmotes for deployment

Fig. 7. A mote in raspberries

Network architecture. A hierarchical two tier sensor network was used[1]. Tier one consisted of SADmotes, which functioned as data sources. Tier two

---

[1] The map of the network is available at `http://tinyurl.com/5szy3jd`

was formed by Waspmotes, which acted as network coordinators for the SADmotes, and forwarded data towards the base station. Static routing was used (preconfigured next-hops), as only a single path to sink was possible.

Waspmotes were used for data forwarding as they had more powerful batteries (6600 mAh rechargeable versus 2800 mAh of AA batteries used by SADmotes), and more powerful radio modules with external antennas. The Waspmotes were placed approximately four meters above ground. In this way, a network with four intermediate hops were formed. Total distance covered by this sensor network was 600m. Waspmotes were located in 150–200m intervals from each other. Between some Waspmotes there was a clear line of sight, while other links were partially blocked by some trees. The last of the Waspmotes was attached to a wireless router through USB port, and functioned as a data sink for the network. The router read data from serial port, stored the data locally and forwarded them to a server in LSIFG building via a Wi-Fi link.

Results. The first thing we noticed about our deployed sensor network was the poor performance of the forwarding tier. Due to constant rebooting and freezing of Waspmotes, only about a hundred of measurements were collected in the server. It is possible that our Waspmote software contained some errors; however, we designed our software by using Libelium's examples as a base.

Even ignoring the troubles with Waspmotes, the initial deployment was not very successful. We performed network maintenace operations in 7th October after the first 18 days of the experiment, and discovered that only two of the motes have been functioning correctly for whole duration of the test. These motes had collected a few thousands of readings. The rest of SADmotes apparently had frozen soon after deployment and had stored only a few hundred readings total. We suspect this is due to instability of MRF24J40 radio driver.

The second part of the experiment was more successful, even though we did not try to revive the forwarding network because of lack of replacement batteries for Waspmotes. In 7th October eight SADmotes had their batteries changed and software updated. From these eight motes five were still running at the end of the experiment (2nd November, close to 26 days total for this period). One had stopped worked recently before the end, and one had stopped working half-way the experiment; their SHT75 sensors were damaged due to corrosion. The remaining one was displaced and probably mishandled by field workers. We conclude that in future we need to make sure the external sensors are protected better, and inform people working in the orchard about our devices.

7 Conclusions and future work

SADmote is a sensor hardware device designed for the specific project of collecting micro-climate data in an orchard. The design considerations came from our earlier pilot deployment using Tmote Sky motes. SADmote is made cost-effective, robust, and energy-efficient by trading simplicity for versatility and minimizing the number of components. Nevertheless, the mote features all the sensors currently needed by domain scientists and is extensible for future needs.

SADmote was evaluated using custom application and systems software that includes radio communication stack, over-the-air reprogramming, and preemptive multi-threading. We have supported our claims about SADmote with at least one successful mid-term deployment, as well as with numerous other tests performed in the lab and in field. In particular, we have showed that SADmote has higher energy efficiency in sleep modes than Tmote Sky and is fully functional at lower battery voltage, thus increasing the energy budget available.

We believe SADmote or its modifications will find other uses in research, development and education, including IECS in-house projects. SADmotes are also targeted as Tmote Sky replacement in a WSN course at the University of Latvia. Our future plans include usage of SADmote in field tests in the spring and summer of 2012. With help of domain scientists we plan to improve on the light sensor choices, parameters and signal processing.

The third version of SADmote is under development and features an optional, more powerful (20dBm) radio transceiver and external antenna connector. A sub-1GHz frequency band is being considered for long-distance links in order to minimize the radio signal losses due to tree foliage.

Acknowledgements

SADmote was developed in collaboration with LSIFG as part of precision agriculture research project supported by European Regional Development Fund, Project No. 2010/0317/2DP/2.1.1.1.0/10/APIA/VIAA/142.

References

1. Arduino. `http://arduino.cc/`
2. Latvia State Institute of Fruit-Growing. `http://www.lvai.lv`
3. Libelium WaspMote. `http://www.libelium.com/products/waspmote`
4. Andersen, A.: 2.4 GHz Inverted F Antenna, `http://www.ti.com/lit/an/swru120b/swru120b.pdf`
5. Baggio, A.: Wireless sensor networks in precision agriculture. In: ACM Workshop on Real-World Wireless Sensor Networks (REALWSN 2005), Stockholm, Sweden. Citeseer (2005)
6. Dutta, P., Taneja, J., Jeong, J., Jiang, X., Culler, D.: A building block approach to sensornet systems. In: Proceedings of the 6th ACM conference on Embedded network sensor systems. pp. 267–280. ACM (2008)
7. Microchip: MRF24J40MA Data Sheet, `http://ww1.microchip.com/downloads/en/devicedoc/70329b.pdf`
8. Polastre, J., Szewczyk, R., Culler, D.: Telos: enabling ultra-low power wireless research. In: Proceedings of the 4th international symposium on Information processing in sensor networks. pp. 48–es. IEEE Press (2005)
9. Selavo, L., Wood, A., Cao, Q., Sookoor, T., Liu, H., Srinivasan, A., Wu, Y., Kang, W., Stankovic, J., Young, D., et al.: Luster: wireless sensor network for environmental research. In: Proceedings of the 5th international conference on Embedded networked sensor systems. pp. 103–116. ACM (2007)
10. Strazdins, G., Elsts, A., Selavo, L.: MansOS: Easy to Use, Portable and Resource Efficient Operating System for Networked Embedded Devices. In: Proc. SenSys'10 (2010)

11. Tolle, G., Polastre, J., Szewczyk, R., Culler, D., Turner, N., Tu, K., Burgess, S., Dawson, T., Buonadonna, P., Gay, D., et al.: A macroscope in the redwoods. In: Proceedings of the 3rd international conference on Embedded networked sensor systems. pp. 51–63. ACM (2005)
12. Wark, T., Corke, P., Sikka, P., Klingbeil, L., Guo, Y., Crossman, C., Valencia, P., Swain, D., Bishop-Hurley, G.: Transforming agriculture through pervasive wireless sensor networks. IEEE Pervasive Computing pp. 50–57 (2007)