Wireless sensor devices for animal tracking and control

Pavan Sikka, Peter Corke and Leslie Overs
CSIRO ICT Centre
P. O. Box 883, Kenmore QLD 4069, AUSTRALIA
E-mail: firstname.lastname@csiro.au

Abstract

This paper describes some new wireless sensor hardware developed for pastoral and environmental applications. From our early experiments with Mote hardware we were inspired to develop our devices with improved radio range, solar power capability, mechanical and electrical robustness, and with unique combinations of sensors. Here we describe the design and evolution of a small family of devices: radio/processor board, a soil moisture sensor interface, and a single board multi-sensor unit for animal tracking experiments.

1. Introduction

We began experiments with Mote hardware[3] in early 2003 deploying a large (54 motes) network of Mica 1s outdoors and we investigated communications between ground nodes and between a node mounted on a flying robot[1]. We found that it was difficult to establish communications between nodes 4m apart even though we could achieve many times this range in the lab. We learnt that radios sitting on damp soil lost most of their emitted signal to the ground, and that by placing them on inverted plastic flower pots we could increase the range. We also learnt that packet transmission was probabilistic rather than simply working or failing, and that often a received packet would be fine except for a single flipped bit. We also learnt that communications was not symmetric, that is A can hear B, but B cannot hear A, and further, that the reception range was not circular and depended on the relative orientation of the antennas (we used 916MHz radios with the small external spiral antennas). Similar experiences are reported by Kotz et al.[4].

From our early experiments with Mote hardware we were inspired to develop devices with improved radio range, solar power capability, mechanical and electrical robustness, and with unique combinations of sensors. Here we describe the evolution of the processor board, a soil moisture sensor interface, and a single board multi-sensor unit for animal tracking experiments. We call our family of hardware devices “Flecks”, perhaps the last unclaimed synonym for mote, speck, dust etc.

Our concerns with the Mica 1 motes included:

1. poor radio range
2. unreliability of the connector from the processor board to the sensor board
3. need for unique sensor combinations which were not available commercially
4. no support for solar cells
5. no useful functionality without a sensor board

Of course, in parallel with our development of the Flecks, Crossbow developed the Mica 2 which overcame the radio problem, but in a way that is different to, and incompatible with, the path that we chose.

2. The Fleck family

Engineering is all about trade-offs, and we have emphasized different criteria to those of the Mica 2 designers. The biggest change, and the one that caused us the most angst, was the radio, which is unfortunately incompatible with the Mica 1 and Mica 2. Specific details of the differences are described in the following sections.

Our first board, developed in late 2003, comprised just an Atmel (http://www.atmel.com) Atmega 128 processor and the Nordic (http://www.nvlsi.no) NRF903 radio chip, no sensors at all, and has a small footprint (50 × 50 mm). We used this to prototype and debug the electronics and to port the TinyOS [3, 2] radio stack to the Nordic chip, described in more detail in Section 3.1. This was followed by parallel developments of the Fleck 1b/c and the Fleck 2.

The Fleck 2, shown in Figure 2 was designed for a specific application in animal tracking and control. We needed a compact and low-cost solution with a diverse number of sensors including: GPS, 3-axis acceleration, 3-axis magnetic field and temperature, as well as the ability to hold considerable amounts of data. The problem of large

Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN’04)
0742-1303/04 $ 20.00 IEEE

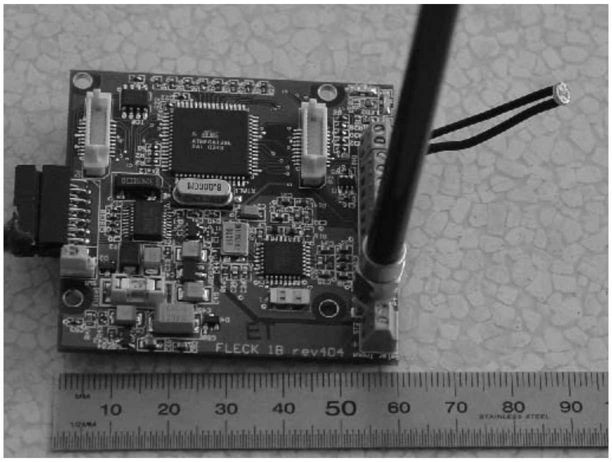

Figure 1. The Fleck1b board (Note the light sensor connected to the screw terminals at the edge of the board).

Figure 2. The Fleck2 board.

amounts of logged data and the time required to download it over the radio link (at least in early testing) led us to add provision for an MMC flash memory card.

The Fleck 1b (Figure 1) is based on the original Fleck 1a. It is a little bigger (60 × 60mm); however, it has support for various power sources, including solar cells, and also has on-board temperature and light sensors. It is designed so that the light sensor can be mounted off the board. We have used the Fleck 1b to interface to an industry-standard load-cell transmitter (via the UART) and also to soil moisture sensors (via a custom extension board). The Fleck 1c has a slightly redesigned power supply and is in mass production.

Both devices have an expansion interface that is designed to be more rugged than that on the Mica series. The Fleck 2 "all in one" board could have been achieved by using daughter boards on the Fleck 1b, but compactness and cost drove the single board solution.

As mentioned above, the differences between Fleck and Mica 2 are in: the radio transceiver, the power supply architecture, solar charging, and connectors for interfacing and expansion.

2.1. Connectors

The Micas have a single good-quality connector used to interface daughter boards, or the programming board in order to gain an RS232 connection to the Mote. In some earlier work[1] we deployed a large number of Mica 1s in the field and had some issues with proper seating of the sensor boards. One of the Motes was attached to a robotic helicopter, and there we found that the connection of Mote to the programming board was very unreliable — it was difficult to keep the sensor board parallel to the processor board and the connector properly seated. We had to place slabs of foam between the two boards and wrap the whole assembly in duct tape.

By contrast the rest of the computer systems on-board the helicopter, based on PC/104 hardware, proved mechanically solid and very reliable. The key difference is the provision for spacers to provide mechanical support, bolts to hold the assembly together, and for PC/104+ two connectors at opposite ends of the boards being mated.

For the Flecks we have three connectors:

1. Datamate 14-way connector for programming and RS232 access.
2. Two M60 20-way connectors for the expansion interface. Two connectors allow the daughter board to sit securely.
3. Miniature screw terminals (Fleck 1b/c only) that provide access to:
    - external light sensor, replaces unit on the board itself
    - two ADC channels
    - four Atmega128 port pins which can be configured as inputs, outputs, interrupts, counters or PWM generators.

Thus a Fleck 1b/c baseboard can meet many common interface applications without needing an expansion board.

2.2. Radio

In our design we emphasized maximum range over other factors. We chose the Nordic NRF903 which has an order of magnitude greater range (500m) and much higher data rate

Antenna length	TX power (dBm)	Range (m)
$\lambda/4$	10	700
$\lambda/4$	-8	20
1cm	10	50
1cm	-8	1

Table 1. Fleck radio range as a function of TX power and antenna length.

(76kbaud). Its disadvantages are that it is incompatible with the Mica 1 and 2, has no Manchester encoding, greater receive current and no RSSI output.

A new radio announced early this year by Nordic, the NRF905, provides an even simpler interface along with higher data rates and thus provides a promising path forward. Nordic have also released a radio which includes an 8051 core, potentially capable of running TinyOS, though very limited in terms of program and data memory.

Range The range is determined by transmitter power, antenna gains and receiver sensitivity. Range can be increased by increasing transmitter power and/or by increasing receiver sensitivity; however, this increased range is at the expense of increased power consumption.

Side by side comparison of the Fleck and Mica 2, with identical antennas and operating at 433MHz, show that the Fleck works up to 700m, whereas the Mica 2 works up to 50m only.

While long range is useful, it is also useful to have low range and this can be achieved by reducing the transmit power. For one of our applications we wish to determine the proximity of two devices. Since the Nordic does not provide a useful RSSI signal we use a binary connectivity measure and Table 1 shows some results. Even at lowest transmit power the range is still 20m. We investigated the effect of reducing the antenna length and find that with a 1cm antenna we can achieve a maximum range of 50m (as good as the Mica 2), and a minimum range of 1m which provides good connectivity. Ultimately the choice of antenna length will be a function of the application.

CPU interface Another important consideration is the CPU resources required by the radio transceiver. The radio on the Mica 2 requires exclusive use of the SPI channel on the CPU. This makes it difficult to interface with other SPI slave devices. The radio on the Fleck interfaces with one of the two UART channels on the CPU, thus leaving the CPU SPI channel free for interfacing with other SPI slave devices (sensors, flash etc).

Data rate and format The Mica 2 CPU provides an effective data transfer rate of 2400 bytes/s (19.2K bits/s, with Manchester encoding for over-the-air baud-rate of 38.4K). At this setting, the receiver sensitivity is -103dBm, which is a little less than the Nordic radio (-104dBm). However, the Nordic radio achieves a significantly higher effective data transfer rate of 6980 bytes/s (76.8K bits/s, with a start bit, a stop bit, and dynamic scrambling).

The advantage of the Nordic is that it uses GFSK compared to the Chipcon's FSK. According to data-sheets, this advantage translates into better range and more channels. The other advantage is data transmission speeds. The Chipcon radio can provide a maximum effective data transfer rate of 38.4Kbps (with Manchester encoding). At this setting, the receiver sensitivity is -98dBm (with freq separation of 20kHz) and -101dBm (with freq separation of 64kHz). The Nordic radio has a sensitivity of -104 (with freq separation of 46kHz) at a effective data transfer rate of 76.8Kbps, a very clear advantage.

Current draw The Chipcon allows transmit power to be varied from -20dBm to 10dBm in steps of 1dBm. At 10dBm, the chip uses 26.7mA. The Nordic provides only 4 power levels -8, -2, 4 and 10dBm. At 10dBm, it uses 24mA, which is slightly better than the Chipcon. The Mica 2 stack initializes the transmit power to 6dBm (15.8mA), although a comment in the code suggests it is using 0dBm.

The Mica 2's Chipcon operates at 19.2K, which gives a baud-rate of 38.4K with Manchester encoding. This provides a receiver sensitivity of -100dBm with a frequency separation of 20KHz, or -103dBm with a frequency separation of 64KHz. The current consumption is 9.3mA. This compares with a receiver sensitivity of -104dBm at 18.5mA for the Nordic. Thus, the Nordic provides enhanced receiver sensitivity at higher data rates but uses more current.

RSSI The Nordic provides essentially a carrier detect signal rather than a proportional signal strength signal. We connect this to an interrupt pin, allowing the CPU to wakeup when a message is "on the air". Sleeping the processor in this fashion pretty much cancels the Chipcon radio's 10mA advantage mentioned above.

The loss of RSSI is not as bad as it might at first seem. RSSI can be used as an analog for range, but the relationship is complex and dependent on the path between the transmitter and receiver, so while somewhat useful, we considered it an acceptable sacrifice.

2.3. Power supply

The wireless sensor network literature takes as a given that energy efficiency is critical, both in computation and network protocols. Battery lives of 6–12 months are mentioned, although in practice we observe much less in applications where the receiver is always on. For our goal

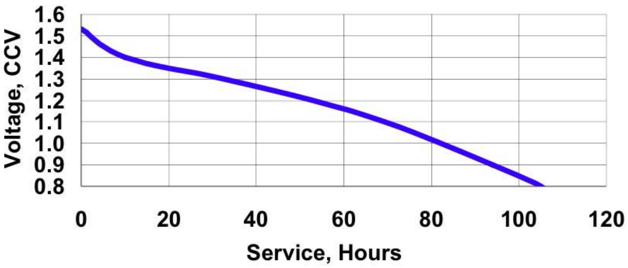

Figure 3. Battery terminal voltage versus time for light duty (Energizer 3-315). Courtesy Eveready Battery Co.

of much longer term unattended operation changing batteries, even every 6 months, is an unwanted complication. Our power supply differs from that of the Mica motes in two significant ways: a DC-DC converter allows operation over a wide range with a low minimum voltage of 1.3V, and an inbuilt solar battery charger.

Power supply The Mica 1 employed a DC-DC converter but the Mica 2 does not, directly connecting the 3V battery to the radio and CPU chips. The minimum voltage for the radio is 2.1V and the CPU is 2.7V making it the first to fail as the battery discharges. For the Fleck 1c we use a DC-DC converter which has a minimum input voltage of 1.3V.

Figure 3 shows terminal voltage as a function of time for constant light usage. The Mica 2's minimum voltage of 2.7V (1.35V per cell) corresponds to a life of 15 hours compared to 105 hours for the Fleck's minimum voltage of 0.7V per cell.

The Fleck input voltage can range from 1.3 to 5.3V and it can supply up to 500mA of current. The input is protected by over-voltage and reverse voltage diodes and a fuse. The National LM2621 switching regulator is a high efficiency step up converter that operates at frequencies up to 2MHz. Efficiencies up to 90% are achievable. The switching regulator can step up voltages from as low as 1.3V. Once the regulator has started, however, input voltages as low as 0.7V can be tolerated. For the Fleck 1c the highest input voltage is 5.3 volts and at this point the regulator switching FET is on all the time. The quiescent current used by the switching regulator is typically 80uA.

Following the DC-DC step up converter are two small linear regulators of type Torex XC6203 3.3V. The dropout voltage for the regulator is 0.15V and the typical quiescent current is 8uA. The output voltage from the switching regulator is set to 3.6 V and this gives a 0.3V noise and dropout margin for the linear regulators. This is important as the type of control scheme used by the switching regulator to give low quiescent current also gives a high level of switching noise (0.1V). There are some inefficiencies in this process: loss in the DC-DC converter, and loss in the voltage drop across the linear regulator. However we believe that the ability to operate from a lower input voltage more than makes up for this.

The Fleck 1c uses one of the regulators for the radio chip and the other for the rest of the circuit. It is important to provide the radio chip with a clean supply voltage as it is essentially an analog device. Noisy supply voltage reduces the radio range.

The regulated 3.3V has a high-side FET power switch connected. The Fleck can therefore switch off power to sensors and other devices when not in use. The battery voltage, solar cell voltage and charging current are all monitored via the CPU's ADC channels.

Solar battery charger The Fleck 1b/c includes a simple solar battery charger on the main board. The cells we used are supplied by Plastecs and are nominally 4V output. The large cell is 8.5 $\times$ 11.5 cm and is rated at 300mA, the smaller cell is 6 cm square and rated at 100mA. The rechargeable battery was rated at 2000 mAh.

Figure 4 shows some results over 2 days obtained with a Fleck 1b. This unit had a power supply configuration as shown in Figure 5 which cuts out with a battery terminal voltage of 3.8V. The test setup comprised two solar-powered Flecks that measured their sensors once every 5s and radioed them to a battery-powered Fleck which logged the results on a removable flash memory device. The solar-powered Fleck applications were not energy efficient, the radio was mostly in RX mode, and the current draw was approximately 30 mA. The tests were run in Brisbane in the winter time when days are short. Figure 4 shows the solar cell output voltage and charging current, as well as the battery terminal voltage, light sensor and temperature.

The Fleck with the large solar cell stopped charging around 4pm and stopped running just before 6am next morning. By 9am the battery had recovered sufficiently to enable the Fleck to resume operation. The Fleck with the small solar cell stopped charging at around the same time and failed almost immediately, indicating that the amount of charge gathered during the day was insufficient to hold up the battery voltage. Per day the large cell generated 1.6Ah of energy, the small cell 0.5Ah. The application, drawing 30mA, consumes 0.7Ah per day which explains why the small solar cell is insufficient. However the consumption is much less than the generation from the large solar cell and it too was insufficient to run continuously. The charge is available in the battery but at a voltage less than the minimum required for this version of the Fleck. On this basis we redesigned our power supply so that the DC-DC converter is as shown in Figure 5 so that we can work with a much lower battery terminal voltage, but the solar cell tests

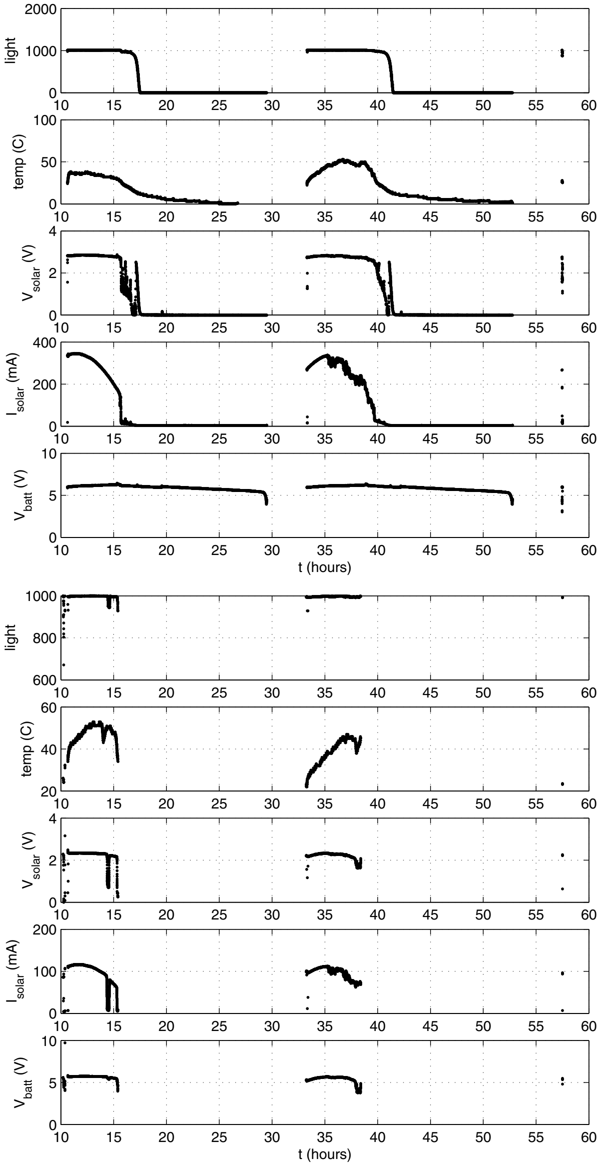

Figure 4. Small solar cell results. (top) Large cell, (bottom) small cell.

have not been repeated. The clear conclusion from this is that a very small solar cell, perhaps as small as your thumb, could sustain an energy-aware application indefinitely.

Current draw Table 2 compares the current consumption of the Fleck and Mica 2 for various operating modes. In full sleep mode the Fleck draws more current due to the quiescent load of the DC-DC converter. In RX mode the difference is due primarily to the radio chip.

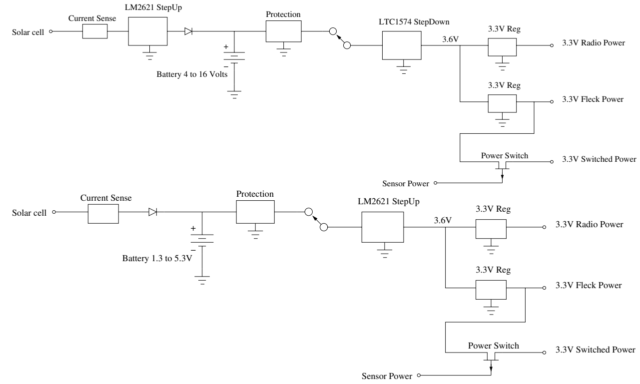

Figure 5. Power supply block diagram. (top) Fleck 1b, (bottom) Fleck 1c.

State	Fleck	Mica 2
All off All off and micro in sleep mode	120uA	16uA
Micro on, radio and other sensors off	8mA	8mA
Micro on, radio in RX mode	26mA	17mA
Micro on, radio in TX mode (max power)	32mA	33mA

Table 2. Measured comparison of Fleck and Mica 2 current draw in different modes.

3. Software

The Flecks use the Atmel Atmega128 processor (same as Mica 2) and have a similar programming interface. Therefore, the core of TinyOS presents no porting problems. In fact, the "Blink" application for the Mica 2 runs on the Fleck without modification since the Fleck's 3 LEDs are connected to the same pins on the CPU as the Mica 2. However, there are significant differences in the peripheral chips on the board. Therefore, the first step in porting TinyOS to the Fleck is to create a new platform called "fleck". The second, most significant, step is to port the Mica 2 radio stack based on the Chipcon CC1000 radio chip (http://www.chipcon.com) to the NRF903 radio chip from Nordic.

3.1. The radio stack

The Mica 2 uses the Chipcon CC1000 radio chip while the Fleck uses the Nordic NRF903 radio chip. While these two radio chips are similar in their electrical specifications,

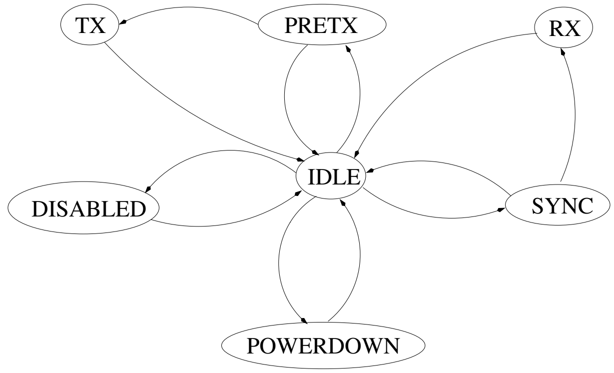

Figure 6. The state machine implemented by the Mica 2 radio stack.

the methods used to interface them to the Atmega128 CPU are very different. The Mica 2 uses the SPI channel in slave mode to connect to the data pins on the radio chip. It uses a couple of digital I/O pins and a software SPI interface to configure the radio chip. This method has a big disadvantage in that it ties up the SPI channel on the Atmega128. In contrast, the Fleck uses the SPI channel as master to configure the radio chip. It uses one of the two Atmega128 UART channels to connect to the data pin on the radio chip. This allows us to utilize the SPI channel to communicate with a host of other peripheral devices that behave as SPI slaves.

The main porting effort was to replace the Chipcon CC1000 related TinyOS components in the Mica 2 radio stack with equivalent Nordic NRF903 TinyOS components.

The Chipcon CC1000 radio connects to the Atmega128 as SPI master. This implies that the radio chip is in charge of all communication. In transmit mode, the radio chip reads data from the SPI channel. Each byte transferred over the SPI channel provides an interrupt to the Atmega128, which loads a new byte to be transferred. Similarly, in receive mode, the radio chip provides a continuous bit-stream over the SPI interface. Every time a byte is transferred, an interrupt is generated on the Atmega128, which causes it to handle the received data. Since the radio chip acts as a constant source of interrupts in both receive and transmit modes, the TinyOS radio stack uses this source of interrupts to drive a simple state machine shown in Figure 6.

The Nordic radio is interfaced to the Atmega128 CPU using a UART channel. Since the TinyOS radio stack makes assumptions about how the data interface to the radio works, we could not use the HPLUART interface normally used for the UART channels. We had to implement the serial interface to mimic the serial interface used by the Mica 2 radio. The serial interface also took care of dynamic scrambling as the Nordic radio does not provide any hardware support for encoding the serial stream over the radio. Figure 7 shows the design of the Fleck radio stack in terms of the TinyOS components and how they are wired together.

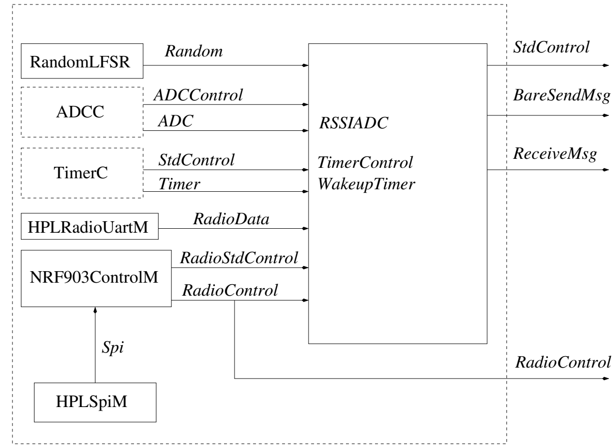

Figure 7. The Fleck TinyOS radio stack. The solid boxes refer to modules, the dashed boxes refer to configurations, and the labels in italics refer to TinyOS interfaces.

3.2. The FAT file system

The Fleck 2 provides a socket for an SD/MMC flash memory card. These cards support two interfaces: a fast, proprietary interface and a comparatively slow SPI interface. Therefore, these cards can be easily interfaced with the Atmega128 SPI channel. The MMC (SD) standard, available from the Multimedia Card Association (http://www.mmca.org), provides a simple protocol to read and write the card and this protocol can be used to support any desired file-system. Typically, these cards come pre-formatted with the DOS file-system, and that makes it easy to read them from any normal PC.

We purchased a library (http://www.prllc.com) that provides the ability to read and write DOS files on these cards. The porting effort involved customizing the library to suit our hardware configuration and then wrapping it up as a TinyOS component providing a simple interface.

3.3. The Fleck2 sensor suite

Corke et al. [5] describe a system to monitor and control the location of cattle using a Zaurus PDA and a Garmin E-Trex GPS unit. It could log animal position to flash memory or transmit it via an ad-hoc network layered over 802.11b. The system was built using off-the-shelf component and tested on a small herd of dairy cattle over the Northern summer of 2003. Commercial animal tracking devices are available from several manufacturers and are widely used by natural scientists studying various wildlife. Position data can be logged periodically to internal storage or radioed out.

To address a need by animal scientists within CSIRO to monitor animal activity we developed the Fleck 2. It provides:

1. GPS,
2. 3-dof digital compass,
3. 3-dof accelerometer, and
4. temperature.

These allow us to determine the attitude of the Fleck device on the animal from which we can determine whether the animal's head is up or down. It may also be possible to determine whether the animal is walking and possibly even its gait (is it walking or running?). The magnetometer, in conjunction with the accelerometers, allows us to determine which way the animal is oriented with respect to magnetic north. The Fleck 2 has sufficient spare I/O capability that it could also drive a stimulus device as would be required for animal position control experiments. Power to the GPS and all other sensors can be controlled directly by the CPU.

GPS The GPS modules (TIM-LC) on Fleck 2 are made by U-Blox (http://www.u-blox.com). These are based on a chip-set jointly developed by U-Blox and Atmel and manufactured by Atmel. In their default state, these modules transmit NMEA (http://www.nmea.org) sentences at 1 Hz when they are powered up, though capable of providing fixes at 4Hz. The simplest TinyOS interface processes these NMEA messages and provides them to other modules as an event. The serial connection to the GPS chip is bidirectional allowing the device to be controlled, and also to provide it with differential corrections. We have not yet tested this, but propose to propagate differential corrections from a base station over the ad-hoc network.

These GPS modules also support a proprietary binary interface which is more interesting and powerful. This interface allows the module to be configured in different modes (power saving, device models, etc) and also provides raw data which can be used to generate a RINEX observation file which can then be post-processed to get a very accurate trajectory. We place the GPS chip in sleep mode and let the processor wake it up and poll it for fixes. The chip also provides an option to increase receiver sensitivity and also to select receiver models (stationary, pedestrian, automotive (default), etc). A TinyOS component to expose this extended functionality has been written.

The GPS receiver consumes about 80mA @3.8V - in sleep mode, this can go down to less than 30mA. However, it needs anywhere from 1–35s to get a first fix when it is turned on again. Furthermore, in this mode, fix quality degrades over time. The chip requires ephemeris data every 2 hours (which happens only if it is turned on for more than 36s). These constraints mean that modest duty cycling does not necessarily result in much current saving.

Digital compass The digital compass consists of a 2-axis (HMC1002) and a 1-axis (HMC1001) magnetic sensor from Honeywell Sensor Products (http://www.ssec.honeywell.com). These sensors are mounted to provide a 3-dof magnetic sensor that can be used to determine a unique orientation with respect to the Earth's magnetic field. These sensors are interfaced to the Atmega128 through 3 ADC channels. The sensors also provide a facility to calibrate offsets on-the-fly via some dedicated circuitry triggered through two digital I/O pins.

The TinyOS component provides commands to calibrate the sensors and to acquire data from the sensors. It also provides an event that signals valid data.

Accelerometer The 3-dof accelerometer consists of two dual-axis accelerometers (ADXL202) from Analog Devices (http://www.analog.com). These sensors are mounted to provide a unique 3-d acceleration vector with respect to the Earth's gravity field. These sensors provide both digital PWM output as well as analog output. The digital output is preferred since it provides invariance against temperature drift; however, the Atmega128 provides only 2 input capture channels. Therefore, one sensor is wired up to the two input capture channels, while the other sensor is wired up to two ADC channels on the Atmega128. These sensors need to be calibrated by placing them in known positions with respect to the Earth's gravity field. The TinyOS component provides a command to acquire a reading from the sensor and an event that signals valid data.

Temperature We use the Digital Thermometer with SPI interface (DS1722) made by Dallas Semiconductor (http://www.maxim-ic.com). This sensor is a SPI slave and provides the temperature in degrees Celsius as a signed number over the SPI interface. The sensor can be configured to sample the temperature continuously, or on demand (for power-save modes). It also allows the resolution to be controlled in steps from 1.0 degrees, 0.5 degrees, 0.25 degrees, 0.125 degrees, and 0.0625 degrees.

Again, the TinyOS component provides a command/event based interface to read the temperature.

4. The Application

Our application domain is a paddock in which we wish to monitor the state of the animals and the landscape, and our test site is at Belmont near Rockhampton. Several Fleck 1cs are interfaced to digital weigh-bridges (via an RS232

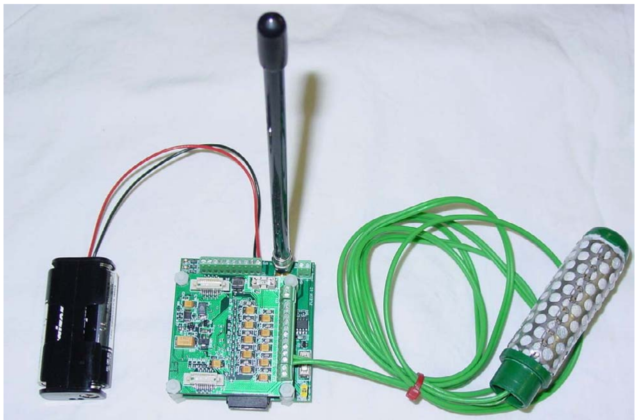

Figure 8. Fleck 1c with moisture probe interface board and one (of five) soil moisture probes.

serial link), as well as water trough flow meters (via an analog input). Some Fleck 1cs have an extension board that interfaces with up to 5 soil moisture sensors (see Figure 8) allowing measurement of the vertical moisture profile in the ground. The mobile component of this network is 20 Fleck 2s which are worn by the animals (Figure 9).

All these Flecks are connected to a central facility via a pair of Fleck 1cs acting as gateways to a PC which provides a route to the Internet via an ISDN link. The long hop from the paddock to the central farm building is achieved using high-gain antennas.

The initial testing occurred before the network link was established. The Fleck 2s were programmed to write all sensor data onto the flash memory card in plain text format. However, the Flecks were also programmed to broadcast their identity at regular intervals and to record all received broadcast messages. This allows us to build up and analyze connectivity information over time.

Figure 10 shows the TinyOS component diagram of the application that has been deployed in the first instance. The aim is to shortly deploy an application that relies heavily on the radio and does not require the flash memory card to be physically removed as often as it needs to in the current setup.

5. Some results

Figure 12 shows a segment of data from a single collar which represents the position of an animal at 1 second intervals. Figure 12 plots orientation, derived from the onboard accelerometers as a function of time. From information like this we hope to be able to determine whether the animal has its head down (eating) or head up (walking). Further work is needed to accommodate slippage of the collar which tends to rotate around the animal's neck.

Figure 9. The packaged Fleck2 board on the cow.

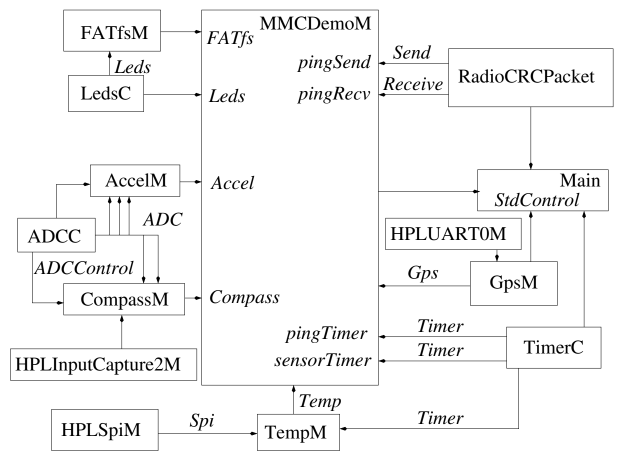

Figure 10. The TinyOS application deployed in the field.

6. Conclusion

In this paper we have described a new family of wireless sensor devices. They were developed with different engineering trade-offs to existing devices and to meet the needs of particular applications.

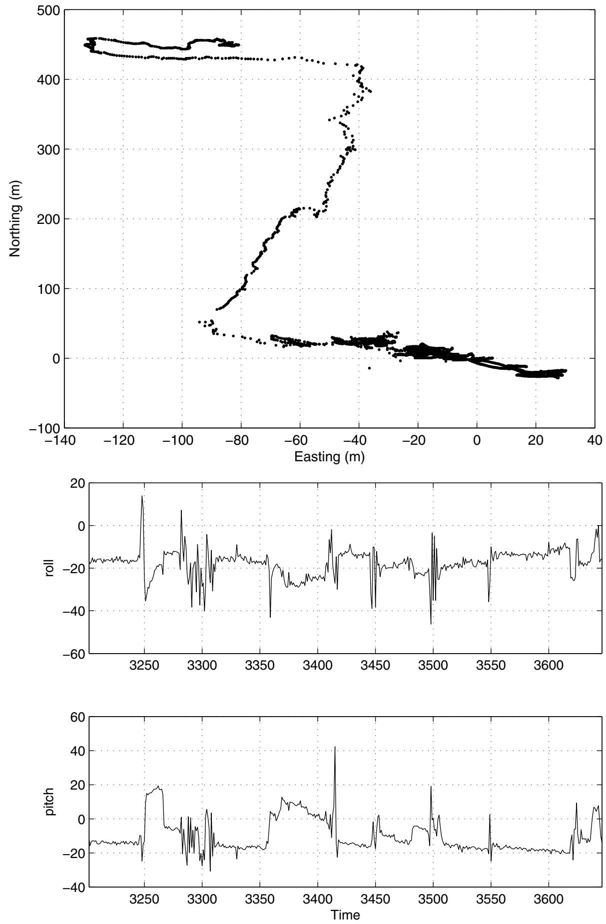

Figure 12. Roll and pitch angle of Fleck2 as a function of time.

Acknowledgments

The authors would like to thank the rest of the CSIRO Robotics team in particular Graeme Winstanley and Stephen Brosnan for making the Fleck concept become a reality. The CSIRO SmartSpaces project funded the design of the Fleck hardware family and the CSIRO Food Futures Flagship has funded the manufacturer and deployment of the Fleck 2s. We would also like to thank Daniela Rus (Dartmouth and MIT) for starting our interest in wireless sensor networks and with Ron Peterson for collaboration on the problems of animal tracking and control.

References

[1] P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, and G. Sukhatme. Deployment and connectivity repair for a sensor network with a flying robot. June 2004.

[2] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler. The nesc language: A holistic approach to networked embedded systems. In *Programming Language Design and Implementation*, 2003.

[3] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for network sensors. In *Architectural Support for Programming Languages and Operating Systems*, 2000.

[4] D. Kotz, C. Newport, and C. Elliott. The mistaken axioms of wireless-network research. Computer Science TR2003-467, Dartmouth College, July 2003.

[5] Z.Butler, P. Corke, R. Peterson, and D. Rus. Virtual fences for controlling cows. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 4429–4436, Apr. 2004.