

# Low Power Operating System and Wireless Networking for a Real Time Sensor Network

Andrew J. Wixted, Daniel A. James *Member IEEE* and David V. Thiel, *Senior Member IEEE*

**Abstract**—Wireless networking of sensor devices is becoming more pervasive as technology, particularly miniaturization makes more sensing applications possible. Power efficient operating systems and network protocols are key components for wireless sensing applications. This paper presents a low-power operating system and associated wireless protocols that have been developed to operate in a centrally synchronized real-time sensor network. The developed operating system is generic and loosely coupled to a commercially available low-power micro-controller, thus providing an upgrade path as more efficient hardware becomes available. The inherent flexibility of the operating system allows the simple implementation of wireless protocols to suit specific applications. A number of example prototype systems have been developed and the results presented.

**Index Terms**- Wireless Networks, Low Power, Operating System

## I. INTRODUCTION

WIRELESS monitoring imposes a variety of requirements and restrictions on system topologies, system and node data rates, power consumption, system weight and sensor interfacing. The system currently being developed is a general-purpose wireless sensor platform designed to be semi or totally encapsulated. System power will be supplied by inductively recharged batteries and for some applications may include active or passive energy harvesting. Weight and size for the total package is restricted and the wireless range is non-trivial therefore a power effective operating system and network protocol is required.

## II. CURRENT AND FUTURE SYSTEMS

Considerable funding worldwide has been devoted to research in miniature wireless sensor platforms. A number of major universities have research centers devoted to the various aspects involved. Notable instances include the Massachusetts Institute of Technology  $\mu$ -AMPS Project (micro-Adaptive Multi-domain Power aware Sensors) [1] and the PicoRadio [2] and associated projects from the Wireless Research Center at the University of California at Berkley. Many other

This work was supported in part by the Co-operative Research Centre for Microtechnology. This overview paper prepared for the ICITA 2002 Conference, Track 5, IT in Engineering.

A. J. Wixted, (email: [a.wixted@mailbox.gu.edu.au](mailto:a.wixted@mailbox.gu.edu.au)), D. V. Thiel, (email: [d.thiel@me.gu.edu.au](mailto:d.thiel@me.gu.edu.au)), and D. A. James, (email: [d.james@me.gu.edu.au](mailto:d.james@me.gu.edu.au)) are with the School of Microelectronic Engineering, Griffith University, Brisbane, QLD 4111 Australia (fax: 61 7 3875 5384)

institutions are also researching aspects including miniature antenna arrays [3], spread spectrum technologies, variable power technologies and energy harvesting systems [4]. Commercialized technologies such as Bluetooth [5] can provide solutions for some but not for all proposed applications. As availability of integrated Bluetooth controller and RF components in chip size packaging improves these devices may need to be reconsidered. The advantages of high data rates and refined network protocols are attractive for a number of implementations. Some commercial integrated circuits exist that incorporate microprocessor and radio frequency components but those surveyed did not offer the required functionality.

Since hardware is continually evolving and devices with higher efficiency and functionality are being developed, the general-purpose wireless sensor platform presented in this paper is developed using devices that best meet the requirements *at this time*. The developed system consisting of a loosely coupled operating system and wireless protocol is such that it can easily incorporate devices of higher flexibility or efficiency as they become available.

## III. DESIGN REQUIREMENTS

The general-purpose wireless platform is designed for use across a range of generic real time monitoring applications. In most instances the platform is in an environmentally sealed package, of low weight and size. While specific physical restrictions are dependent on the actual application sizes are intended to include applications such as ear tags on cattle. Each of the design requirements are dealt with in this section.

### A. Network Topologies

The general-purpose wireless platform is required to operate in diverse network topologies, these topologies being partially imposed by the specific problem to be solved. Fig. 1 depicts some example topologies. Single and multi-channel networking and the synchronization technique further complicate the topologies depicted. The general-purpose wireless platform can be deployed in any of the roles depicted in Fig. 1. It is also used as the receiver at the client. For multi-channel operation a number of general-purpose wireless platforms can be combined in the repeaters and at the client. Sensor only nodes may be mobile or stationary.

### B. Data Rates

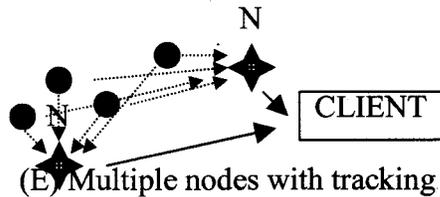
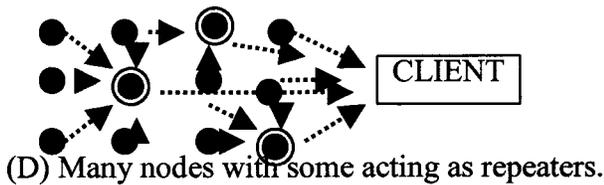
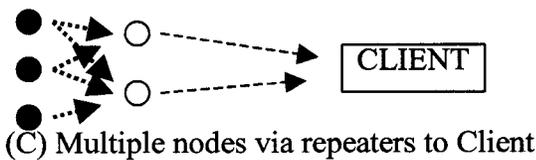
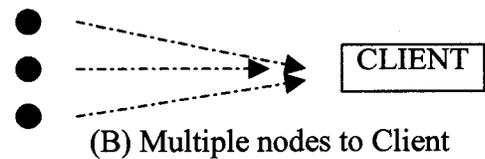
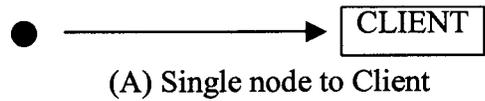
The maximum required system data rate is usually open ended, often being more than can realistically be provided. The general-purpose wireless platform is being developed for

some generic and novel applications. Currently data is simply being collected at the maximum possible rate to assist in determining the data requirements. Examples of the required sensor channels, sample rate, sample size and nodes in the network are given in Table I, these provide some guidelines to the required data rates. As additional wireless channel capacity and system capacity becomes available it will be consumed by additional nodes, higher sampling rates and additional sensor channels.

investigated including low power network design and energy harvesting.

Network Data Requirements				
Sensor Channel Count	Sample Rate (Hz)	Bits Per Sample	Sensor Nodes	Raw Data Rate (bits per second)
8	1000	10	2	160,000
3	100	10	15	45,000
5	1 per hour	8	1000	12

TABLE I. INITIAL REQUIRED DATA RATES – VARIOUS APPLICATIONS



General-purpose wireless platform operating as:

- Sensor only network node
- Repeater only network node
- ⊙ Repeater/sensor network node
- N Direction sensing repeater node.
- ★ **Note: repeater nodes are not power limited.**

Fig. 1. Required network topologies.

### C. Power Requirements

In many cases network nodes are mobile and can have no directly coupled external power source. Fortunately in many applications where there is a high data rate requirement there is a corresponding low required battery life. Required battery life generally is from one hour to six months. To reduce battery size and weight a number of strategies are being

### D. Sensor Interfacing

Commercial and propriety in-house sensors are interfaced to the general-purpose wireless platform. Sensors include MicroElectro-Mechanical Systems (MEMS) such as accelerometers [6], magnetic sensors [7], gas and chemical sensors. Standard interfacing techniques such as Analog to Digital Conversion (ADC), Pulse Width Modulation (PWM) and pulse counting are implemented. These sensors monitor movement, orientation and some environmental conditions.

Fixed Active Switched Parasitic Array (FASPA) tracking antenna also interface with the wireless platform [8].

### E. Operating System

The operating system of the general-purpose wireless platform is to minimize power consumption and be loosely coupled to the hardware. The operating system needs to operate the generally tightly coupled wireless network.

### F. Network Protocols

The primary aim of the sensor platforms is to deliver sensor data wirelessly using a network protocol, hence there is only minimal requirements for operating protocols. Protocol development is focused on two key areas, the configure/start-up protocols and the network synchronization protocol. The underlying philosophy of all protocols is for power minimization. Generally the network will be tightly coupled to maximize data throughput. For some low data rate implementations a looser arrangement can exist where nodes can poll the client in an ad-hoc manner.

Protocol development allows single channel and multi-channel network operation. Due to the commercial RF devices, nodes only operate in half duplex mode.

### G. Signal Processing

Signal processing will need to be applied at two levels, on board node processing and client processing. Initial on-board loss-less processing is required to reduce network data traffic. Client processing is required to perform application specific signal processing. Typical processing includes integration of sensor data and graphical display of derived data for one or more network nodes. For later more sophisticated analysis the client will log raw data to file.

## H. Client Software

Client software configures, monitors and maintains the network in addition to performing logging and application specific processing. As such the network does not exist without the client software, it provides the necessary network start-up processes and captures network data. Multiple clients can exist but only one operates as the network controller.

### I. Security

Many applications require data security. Due to the operating system design, encoding of data at each network is relatively simple to implement.

## IV. PROTOTYPE SYSTEMS

### A. Hardware

Initial hardware prototypes were developed using commercially available micro-controllers and Radio Frequency (RF) devices coupled to directional antenna. These provided proof of concept and a base of raw sensor data for investigation of signal processing techniques. A second series of prototypes has been developed incorporating power controllable transceivers and low power high performance RISC based micro-controllers (see Fig. 2). Operating at maximum frequency the micro-controller draws approximately 10mA. For low data rate sensors the micro-controller can switch to lower powered modes drawing as low as 25 $\mu$ A while processing and 5 $\mu$ A while on standby. The processor clock frequency and hence processor power consumption can be scaled to match the data requirements. All micro-controller subsystems, including the Arithmetic & Logic Unit (ALU) can be placed in standby mode further reducing power consumption. Powered sensors, when operating at low sampling rates are switched on and off as needed.

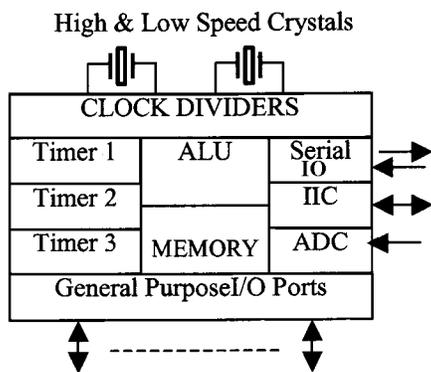


Fig. 2. Micro-controller Subsystems

Current prototypes interface to either a single 64 kbps channel device or dual 20 kbps channel device.

Battery charging is via an inductively coupled charging circuit. During charging and while the battery is on float the sensor hardware platform are electrically disconnected.

### B. Wireless Protocol

The predominant requirement is to transfer data in such a way as to maximize the efficiency of the available data

channel. In key applications this will continue to be the case even with the implementation of higher rate multi-channel RF devices. In those applications where the channel capacity is sufficient, the power efficiency of the protocol becomes critical. Asynchronous networking schemes are inherently inefficient in terms of channel utilization and power use. Synchronous networking schemes require tight synchronization between all network nodes and the synchronizing packets consume system bandwidth. To meet the requirements it is necessary to implement a synchronous network scheme and attempt to minimize the bandwidth loss due to both synchronization signaling and inter-packet separation.

Sensor data is transferred via a heterogeneous Time Division Multiple Access (TDMA) scheme. Timeslot width and repetition rate for any individual node are determined by the client configuration software. This allows nodes with different data requirements to co-exist in the one system. The wireless protocols are created by the triggering of specific transmission or reception sequences at specific instances. This triggering is performed by a synchronized scheduler within the operating system.

Scheduler synchronization is performed by the transmission of clock information at a rate is based on the characterized network drift, environmental noise and the desired inter-timeslot separation. Network drift can be determined during the network configuration phase or by prior experience. For noisy environments the nodes can be instructed to apply synchronization adjustments in the event of missed or corrupted synchronization signals. This technique allows synchronization signaling to occur in a dedicated slot (expensive in both the power and network budgets) or in a slot stolen from a sensor node (at the cost of a lost packet). In some cases where power is not an issue the dedicated slot may be implemented with no loss of network efficiency. This is due to the tighter inter-slot tolerance associated with tighter synchronization. The actual technique used in any specific situation can be programmed into the sensor node's schedule. In the case of stolen slots the node currently scheduled to transmit data is also scheduled to receive the synchronization signals, by correct prioritization the node will go into receive mode.

Techniques such as data inter-leaving, error correction and compression can be applied as necessary by appropriate processing prior to transmission.

Since the scheduler is initially loaded by the client software, the sequencing of transmission and reception tasks on all nodes is entirely flexible and any specific network topology can be created by the client software. This technique is highly flexible without excessive complexity. By its nature it allows flexible protocol generation providing a platform for further investigation of networking and power efficiency techniques. By appropriate scheduling the network can be pre-programmed to go into standby mode until a specific time, wake-up, re-synchronize and start work, a technique that is both convenient and power efficient.

### C. Operating System

The operating system provides services for the collection of sensor data, synchronizing a scheduler for wireless protocol management and minimizing power consumption. Power consumption factors that can be managed on the chosen micro-controller include frequency of operation, subsystem utilization and the clock cycles consumed by the ALU.

#### 1) Operating System Schedulers

The operating system is implemented as a dual scheduler system (see Fig. 3). A time critical scheduler controls the wireless protocol timing and, if necessary the sensor sampling. Tasks run by this scheduler are generally atomic or very short in nature. Longer processing tasks run in the background via a background task scheduler. On an interrupt from the primary interval timer the current interval value is updated and checked to confirm if tasks are scheduled on this interval. If so, the time critical schedule is run, if not a check is performed to ascertain if the background scheduler requires running. The background scheduler will only run if tasks have been added to it. If no tasks are scheduled the processor goes to sleep. Where background tasks are running when an interrupt occurs these tasks resume at the completion of the interrupt service routine. For very low sensor update rates the processor drops into standby mode, only waking for the scheduler interrupt. The processor moves to a higher speed mode when transmission is required. This technique helps to minimize power consumption in the processor. The clock controlling this scheduler is kept synchronized via a wireless protocol synchronization message. Wireless protocols are managed by inserting the appropriate protocol calls into the scheduler.

The primary interval depends on the particular network implementation and may range from milliseconds to seconds. Tasks on either schedule can add other tasks to either schedule. Because the wireless networking is time based most tasks on the time critical schedule are repetitious and once run will reschedule. Rescheduling can occur continuously or for a set repetition count.

#### 2) Application Programming

As the applications are based on the collection and transmission of sensor data, applications typically require the use of micro-controller subsystems. Application development is a simplified process involving short routines for subsystem initialization, interrupt servicing and shutdown. If data processing prior to transmission is required this is performed by a background task. The subsystem initialization identifies the appropriate interrupt service routine. This is reset by the subsystem shutdown routine. In Fig. 4 the operating system provided functions are represented by solid-line boxes while the dashed boxes represent the application specific routines.

#### 3) Network Synchronization

In the current implementation, network synchronization is obtained by the controlling station transmitting scheduler-clock synchronization messages. These messages use a forward clock adjustment to allow for the transmission delay. Multiple messages are sent within one time slot. To minimize

the processing load of the receiver these messages are sent as asynchronous data. The clock data is protected at the character, block and data levels. Approved backward clock adjustments can be immediately applied. Forward clock adjustments are checked to ensure that items in the schedule will not be skipped.

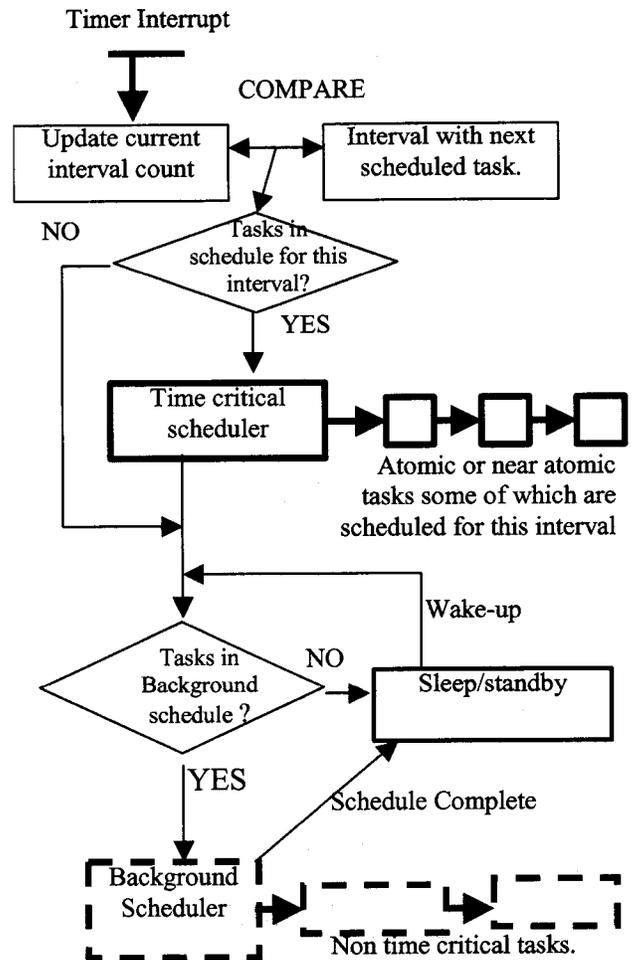


Fig. 3. Operating System Task Scheduling.

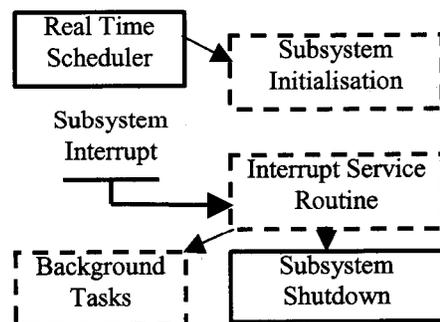


Fig. 4. Application Implementation

To enable large primary intervals the clock synchronization messages include clock interval and sub-interval values. This allows a primary interval of (say) 20ms to be accurate to within 1ms of the controlling node.

#### 4) Code Synchronization and Deadlocks.

This operating system only allows background processes to

be pre-empted. Interrupt processes such as the operation of the time critical scheduler cannot be pre-empted. A background task must run to completion before the scheduler will run the next background task. Although there may be many concurrent threads the only critical code that requires protection from concurrent access is where data structures are shared between background tasks and any other task. All critical code is protected from interrupts. Deadlocks are not possible in this implementation.

#### 5) Signal Processing, Compression and Encryption

Functionality such as signal processing or encryption can be added as a background process specific to the application. When sufficient sensor data has been gathered the appropriate background process can be triggered.

#### 6) Client Implementation

Client Implementation is currently limited to programming via a wired link with wireless synchronization and monitoring of the network traffic. Received data can be both logged and graphically displayed.

### V. RESULTS

To test power efficiency, four combinations of micro-controller clock and scheduler interval size were implemented. To test network synchronization a TMDA network of ten sensor nodes transmitting in 10ms timeslots with a targeted inter-timeslot separation of 3ms was implemented. The theoretical clock drift based on crystal specifications could result in the loss of timeslot separation in approximately 35 seconds if not corrected. The characterized drift based on the sample units was 3.72ms per minute. The data payload, after allowance for RF, serial port and data synchronization was 20 bytes. The data transmission was at 38,400bps (asynchronous characters) carried on a single 64,000bps RF channel. Sensor sample rate was set to 50Hz on three 8-bit channels with sensor data packed into the transmit data buffer by the sampling application interrupt service routine. Synchronization messages were sent every 10.01 seconds and synchronization errors were obtained by nodes transmitting the ‘before’ and ‘after’ values of the scheduler clock. A low interference environment was used. The following results were obtained.

#### 1) Program Code Size

The combined operating system and application code sizes appear in Table II. The total operating system has been implemented in less than 2k bytes of ROM and 1k byte of RAM. The maximum stack depth is calculated as the combined worst case for both a background task and an interrupt driven task.

System Memory Requirements	
Component	Memory Requirement
Program	1748 bytes (ROM)
Global Variables	99 bytes (RAM)
Constants	40 bytes (ROM)
Stack Depth	128 bytes (max)
Stack Size	256 bytes
Heap Size	512 bytes
<b>Total ROM</b>	1788 bytes
<b>Total RAM</b>	995 bytes

TABLE II. BREAKDOWN OF MEMORY REQUIREMENTS

A heap size of 512 bytes allowed a maximum of 32 schedule items to be created, less if applications utilize heap space

#### 2) Operating System Clock Cycles

The number of clock cycles consumed by the operating system is dependent on the duration of the primary interval and the depth of the time critical schedule. The number of clock cycles available is dependent on the crystal frequency and the clock divider value. Fig. 5 shows the processor load by layer, for the four combinations of processor clock speed and scheduler primary interval size. The large sleep period maximizes power saving in the ALU, alternatively this unused processor capacity is available for application processing. With a 1.2288MHz clock speed and a 1ms primary interval it appears, from Fig. 5 that the processor has 80% spare capacity. In this configuration the processor is operating at close to peak capacity during data transmission, changing the primary interval to 10ms relieves this problem. For these tests the combination of an operating frequency of 1.2288MHz and a primary interval of 10ms maximizes the power savings, (see Table III), and minimizes the processing load, (see Table IV). At this setting sufficient processor cycles are available to increase the data transmit rate to 60kbps (synchronous characters) if necessary. The prototype implementation found the double-buffered serial interface to be a processing bottleneck, hence a deeper buffer would be more effective and allow a lower operating frequency and is recommended for future implementations.

Microprocessor Current (mA)			
@ 9.8304MHz		@ 1.2288MHz	
1ms interval	10ms interval	1ms interval	10ms interval
7.5mA	7.5mA	3.2mA	3.0mA

TABLE III. MICROPROCESSOR CURRENT AT DIFFERENT CONFIGURATIONS.

Micro-controller current for all tested techniques is substantially less than the 10.4mA drain incurred at 100% ALU, ADC and timer utilization (@9.8304MHz).

Clock Cycle Breakdown by Layer @10ms Interval Averaged over 100ms. Available clock cycles 122880 @ 1.2288MHz		
Layer	Clock Cycles	Percent
Operating System	4232	3.44%
Application	600	0.49%
Communications	3600	2.9%
<b>Total</b>	<b>8432</b>	<b>6.86%</b>

TABLE IV. PROCESSOR LOAD FOR OPERATION AT 1.2288MHZ AND 10 MILLI-SECOND INTERVAL

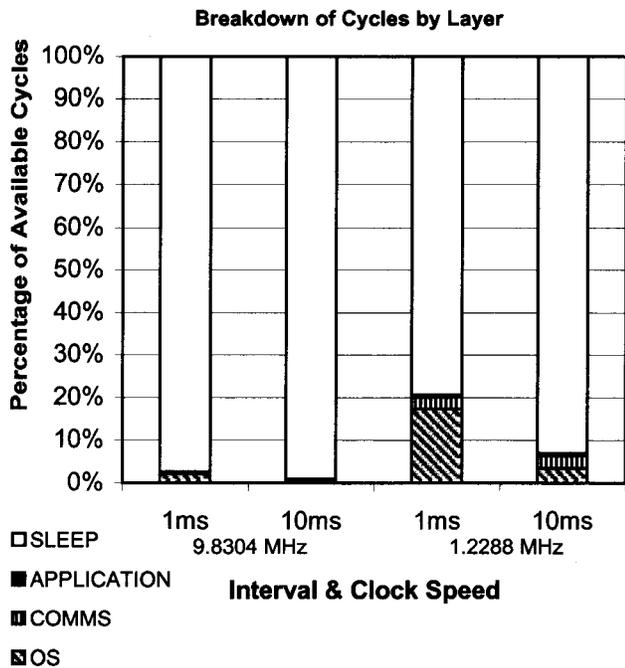


Fig. 5. Processor Load Breakdown for Different Configurations

### 3) Scheduler Clock Errors

The clock errors were limited to +/- 1ms of the control station, potentially reducing inter-timeslot separation to 1ms. Note that this is a low noise environment and the scheme used to demonstrate the operation may not be viable in practice.

## VI. CONCLUSIONS

This paper presents a viable low power, dual scheduler system for use on low power wireless devices to form a comprehensive data-collecting network ready for high and low data rate applications alike. The techniques developed are scalable and not dependant on any particular hardware configuration. The flexibility of the system simplifies the adaptation of the platform to a variety of applications. A number of prototypes have been implemented (hardware and software) and have demonstrated the acquisition of data in a real time environment. The developed operating system provides facilities for further development and investigation of a number of network topologies and applications.

## VII. FURTHER WORK

At this time further work is divided across key aspects of the operating system and wireless networking although some areas are of higher importance than others. The immediate ongoing development includes the following.

### A. Operating System and Software

- Update Operating System to enable the uploading of new and improved function blocks over wireless link.
- Quantify effectiveness of power saving techniques.
- Develop more effective signal processing techniques.
- Fill out the operating system functionality

### B. Wireless Network Software

- Develop low-power network synchronization techniques for each topology and quantify the performance of each technique.
- Develop network and device start-up and configuration protocols.

### C. Client Software

- Provide all the necessary features to configure and manage a full network wirelessly.

### D. Prototype Implementation

Concurrent with the above platform development a number of networks are being implemented to collect sensor data. These include: -

- a high speed data acquisition network,
- a low power low data rate network.

This is an important precursor to the development of robust algorithms to encode and successfully transfer key information in lossy medium for a variety of applications.

## REFERENCES

- [1] MIT u-AMPS Home Page, <http://www-mtl.mit.edu/research/icsystems/uamps/>
- [2] Jan M. Rabaey, Josie Ammer, Tufan Karalar, Suet-Fei Li, Brian Otis, Mike Sheets, Tim Tuan, "12.3 PicoRadios for Wireless Sensor Networks - The Next Challenge in Ultra-Low Power Design", 2001, [http://bwrc.eecs.berkeley.edu/Publications/2002/presentations/isscc2002/12\\_3\\_text.pdf](http://bwrc.eecs.berkeley.edu/Publications/2002/presentations/isscc2002/12_3_text.pdf)
- [3] David Gesbert, "Digital Wireless Communications using MIMOlinks", accessed 2002, [http://www.ifi.uio.no/~gesbert/mimo\\_research.html](http://www.ifi.uio.no/~gesbert/mimo_research.html)
- [4] Wearable Computing Laboratory, Swiss Federal Institute of Technology, Zurich, <http://www.wearable.ethz.ch/>
- [5] The Official Bluetooth Website, <http://www.bluetooth.com/>
- [6] <http://www.analog.com/iMEMS/products/ADXL202.html>
- [7] [http://www.magneticsensors.com/spec\\_sheets/specs\\_1023.html](http://www.magneticsensors.com/spec_sheets/specs_1023.html)
- [8] J.W. Lu, D.V. Thiel, B. Hanna and S. Saario, Multi-beam Switched Parasitic Antenna Embedded in Dielectric for Wireless Communications Systems. In Electronics Letters, Vol.37, No.4, July 2000, pages 871-872.