CHAPTER 3

THE EXTREME SCALE MOTE (XSM)

We present the requirements, design, and performance of the eXtreme Scale Mote (XSM). This new mote is an integrated application-specific sensor network node for investigating reliable, large-scale, and long-lived surveillance applications. The improved reliability stems from hardware and firmware support for recovering from Byzantine programs. Large-scale operation is better supported through an improved hardware user interface and remote tasking. Long-lived operation is realized through the use of adaptive low-power sensors and a hierarchical and event-driven signal processing architecture. The motivating surveillance application is the detection, classification, and tracking of civilians, soldiers, and vehicles. Performed under the aegis of the DARPA NEST Extreme Scale 2004 Minitask, a fundamental goal of this work is to demonstrate operation of a wireless sensor network at the heretofore unprecedented and extreme scale of 10,000 nodes occupying a $10km^2$ area and for a duration approaching 1000 hours. Operation at such scales make it impossible to manually adjust parameters, repeatedly replace batteries, or individually program sensor nodes. These severe constraints were selected to "kick the crutches out" and had the intended effect of elevating to first-class status several factors, such as reliability, usability, and

lifetime, which might otherwise have become afterthoughts. Another very real constraint was cost since every decision was amplified by a multiple of 10,000. Some of the more innovative ideas did not survive budgetary scrutiny and were not included in either the XSM design or in this work.

3.1 Related Work

The design of the XSM was principally influenced by three areas of related work including data collection and event detection applications, sensor network platforms, and low-power signal processing and wakeup circuit design.

3.1.1 Applications

A number of recent works have reported on the design, implementation, and fielding of wireless sensor networks for intrusion detection and response. These works identify many practical lessons which were incorporated into this work.

In, [2] and [3], Pottie, et.al. identify tradeoffs in detection and communications, ideas for network management, and scalable network architectures. The first of the two papers identifies the important question of hierarchy of signal processing functionality and advocates aggressively managing power at all levels.

In [6], West, et.al., present the challenges and tradeoffs for dense spatio-temporal monitoring of the environment and compare the characteristics of military surveillance and environmental monitoring, which are representative of event detection and data collection, respectively. The characteristics of military surveillance applications include performance-driven, mobile sensor nodes, dynamic physical topology,

distributed detection/estimation, event-driven/multi-tasking, and real-time requirements. The characteristics of environmental monitoring applications include cost-driven, fixed sensor nodes, static physical topology, spatio-temporal sampling, scheduled signel tasks, and delays acceptable/preferable. The focus of the work is on environmental monitoring so there is limited discussion of military surveillance applications. This application uses the Wisard sensor network platform.

In [8], Polastre provides a detailed case study of a habitat monitoring application and identifies several performance metrics and design considerations. For example, it is this work that demonstrates network lifetimes on the order of a year and sampling rates on the order of one sample per second. Since the work provides so many useful details, it serves as an exemplar data collection design point. This application uses the Mica, Mica2Dot, and custom sensorboards for the sensor platform.

In [10], He, et.al., describe the design and implementation of a running system for energy-efficient surveillance. This work demonstrates the effectiveness of trading off energy-awareness and surveillance performance by adaptively adjusting the sensitivity of the system. The results show that the surveillance strategy is adaptable and achieves a significant extension of network lifetime. The paper also outlines some lessons learned including false alarm reduction and software calibration of sensors. This application uses both the RadarMote and the Mica2 plus Mica SensorBoard for its sensor network platform.

In [12], Arora, et.al., report on an intrusion detection system using sensor networks and provide a set of performance metrics including latency and probabilities of detection, false alarm, classification and misclassification, as well as design considerations including reliability, energy, and complexity. Like [8], this work provides many useful

details and serves as an exemplar event detection design point. This application uses as its sensor node both the RadarMote and the Mica2 plus Mica SensorBoard.

Sharp, et.al., designed and implemented a sensor network system for vehicle tracking and autonomous interception [17]. This distributed pursuer-evader game demonstrates a networked system of distributed sensor nodes that detects an uncooperative agent and assists an autonomous robot in capturing the evader. Practical issues such as node breakage, packaging decisions, in situ debugging, network reprogramming, and system reconfiguration are addressed. This application uses the PEGSensor node which is consists of the Mica2Dot and a set of custom SensorBoards.

3.1.2 Platforms

A number of platforms/sensorboards have been developed to support wireless sensor network research and development. Research platforms tend to be optimized either for narrow and domain-specific applications or are so general that they are not able to address key application constraints. Modular platforms and sensorboards help in this regard but this problem is endemic to embedded systems in general.

- **Mica2** The Mica2 design is an improvement over the Mica design. The Mica2 uses a higher a performance microcontroller and radio. The microcontroller core is essentially backward compatible with the Mica microcontroller but the radio uses a different modulation scheme and frequency band. Some recently introduced experimental platforms that are commercially available include:

- **Mica2Dot:** The Mica2Dot is a minimalist implementation of the Mica2 in a quarter-dollar-sized footprint.

- **MicaZ:** The MicaZ uses the same microcontroller as the Mica2 but replaces the Mica2's radio with a higher performance radio that supports the IEEE 802.15.4 physical layer standard.

- **Telos:** The Telos design uses the Chipcon CC2420 radio which allows interoperability between the Telos and MicaZ platforms. However, Telos uses the Texas Instruments MSP430 microcontroller due to its lower-power operation and faster wakeup latency compared to the Atmel ATMega128L microcontroller used in the Mica2, Mica2Dot, and MicaZ.

We observe that *no widely-available experimental platform supports the passive vigilance, or very low-power continuous awareness, required for long-lived event detection.*

3.1.3 Circuits

Several circuits have been proposed for low-power signal processing, event detection, and wakeup triggering including:

- **Micropower Spectrum Analyzer:** In [18], Dong, et.al., presents a single-chip VLSI spectrum analyzer implemented in 0.8μ HPCMOS using 45,000 transistors. The system operates with a 1μA drain current at a 3V supply bias at a 200 samples/second processing rate.

- **Acoustic Wakeup:** In [19], Goldberg, et.al., describe a low-power VLSI wakeup detector for use in an acoustic surveillance sensor network.

- **Radio Wakeup:** In [11], Gu and Stankovic describe radio-triggered wakeup capability as a power management technique for prolonging the lifespan of sensor

networks. While it is not clear that the design presented in this paper will work in practice, the concept of a wakeup radio extends the event-driven hierarchy further into hardware, complementing the event-driven operating system, and allowing for node lifetimes to approach 180 days.

3.2 Application Requirements

This work was motivated by the requirements of a typical ground surveillance application. The functional requirements of such an application include detecting a breach along a perimeter or within a region, classifying the target causing the breach, and tracking the target's position as it moves through the sensor network. Our specific application scenario focuses on unattended ground sensors for military surveillance. We detail functional, usability, reliability, performance, and supportability requirements in the following sections. Other possible realizations of this application scenario include border surveillance, pipeline monitoring, and roadway right-of-way protection.

3.2.1 Functionality

Functional requirements are used to express the behavior of a system by specifying both the input conditions and output conditions that are expected to result. From an operational perspective, the XSM design was motivated by the functional requirements of detection, classification, and tracking of civilians, soldiers, and vehicles:

Detection: Detection requires that the system discriminate between a target's absence and presence. Successful detection requires a node to correctly estimate a target's presence while avoiding false detections in which no targets are present. The key performance metrics for detection include the probability of correct detection, or

P_D and the probability of false alarm, or P_{FA}. The goal, of course, is to maximize P_D and minimize P_{FA} for a given level of sensing, computation, and communication. However, P_D and P_{FA} are often positively correlated. That is, P_D and P_{FA} simultaneously increase or decrease. As a consquence, many systems specify an acceptable P_{FA} and achieve the best possible P_D given the P_{FA}.

Classification: Classification requires that the target type be identified as belonging to one of the classes of civilian, soldier, or vehicle. The key performance metrics for classification are the probability of correctly classifying (labeling) the i-th class, $P_{C_{i,i}}$, and the probability of misclassifying the i-th class as the j-th class, or $P_{C_{i,j}}$.

Tracking: Tracking involves maintaining the target's position as it evolves over time due to its motion in a region covered by the sensor network's field of view. Successful tracking requires that the system estimate a target's initial point of entry and current position with modest accuracy and within the allowable detection latency, T_D. Implicit in this requirement is the need for target localization. The tracking performance requirements dictate that tracking accuracy, or the maximum difference between a target's actual and estimated position, be both bounded and specified, within limits, by the user. The system is not required to predict the target's future position based on its past or present position.

3.2.2 Usability

Usability requirements cover human factors like aesthetics, ease of learning, and ease of use, as well as consistency in the user interface, user documentation, and training materials.

One-touch Operation: The basic theme underlying the user interface requirements is "one-touch" operation. For example, waking up a deeply-sleeping node should require no more than a single touch. Similarly, resetting a running node, verifying that a node is operating, and putting an operating node to sleep should require no more than a simple touch.

Form-factor: The XSM nodes had to be fully-self-contained to ensure they were robust and so that they could be deployed easily. The node form-factor had to support tight packaging which would leave little wasted space.

3.2.3 Reliability

Reliability requirements cover frequency and severity of failure, recoverability, predictability, and accuracy.

Retaskable: The system retasking requirement called for multi-hop wireless network reprogramming. A multi-hop wireless network reprogramming algorithm allows a sensor node (the "old node") to be reprogrammed *without* direct (i.e. one-hop) radio communications to a node which holds a new program image (the "new node") as long as there are intermediate old nodes which provide connectivity between any old node and at least one new node.

Recoverable: The system recoverability requirement meant that the nodes could be recovered and reprogrammed even if a pathologic (i.e. Byzantine) program was downloaded. Most inexpensive and low-power microcontrollers do not provide a protected mode of operation. Consequently, it becomes possible for application code to take nearly complete control over the hardware, disable timers, turn off interrupts, and leave the operating system with no mechanism to preempt a misbehaving

application. Hijacking of the operating system can occur either accidentally or intentionally. Recoverable hardware implies a protection mechanism to guarantee trusted code eventually regains control.

3.2.4 Performance

Performance requirements impose conditions on functional requirements – for example, a requirement that specifies the transaction rate, speed, availability, accuracy, response time, recovery time, or memory usage with which a given action must be performed:

Lifetime: The system lifetime requirement is 1000 hours (or over a month) of continuous operation on two series-connected AA-sized batteries which can deliver approximately 6000mWhr of energy. This translates to an average power budget of approximately 6mW and an average current consumption of 2mA. While sensor networks for data collection may allow sensor nodes to sleep most of the time, intrusion detection requires that sensors be awake, or at least passively vigilant, most of the time since targets may be present for very short durations. None of the commercially available sensorboards could detect all of our target classes and meet the lifetime requirements of our application. Consequently, this requirement implied that a new and more energy-efficient sensor node design was required. Our design adopts a signal processing hierarchy and extends the event-driven model into the sensing and signal conditioning hardware, allowing the processor to sleep most of the time.

Latency: The allowable latency or delay, T_D, between a detecting a target's presence and reporting the target's presence to an exfiltration agent.

Coverage: The sensor network needs to operate over an area of 10km^2. With 10,000 nodes and assuming 1-coverage, each sensor node would need to cover 1,000m^2. This translates to an minimum sensing range approaching 20m for all target classes. The implication of this requirement is that since none of the commercially-available experimental platforms that meet our power budget can achieve the required sensing range or even detect all of our target classes, this requirement reinforced the need for a new and more energy-efficient sensor node design that additionally supported a long sensing range.

3.2.5 Supportability

Supportability requirements cover testability, maintainability, and the other qualities required for keeping the system up-to-date after its release. Supportability requirements are unique in that they are not necessarily imposed on the system itself, but instead often refer to the process used to create the system or various artifacts of the system development process. An example is the use of a specific C++ coding standard.

Adaptive: The sheer size of the sensor field guarantees a heterogeneous environment while the number of nodes precludes manual adjustment of parameters. Therefore, the nodes require mechanisms to adjust their sensor sensitivities, detection thresholds, and other similar parameters dynamically and autonomously. The implication of this requirement is the sensing and signal conditioning hardware needs to support *electronic* adjustment of filter cutoffs, amplifier gains and offsets, and comparator thresholds, all under program control.

Backward-compatibility: A key requirement of the XSM was backward-compatibility with Mica2-platform. The motivation behind this requirement was to reduce the risk of bringing up a new platform. During [10, 12], researchers changed from the Mica to the Mica2 platform. This change involved switching from the Atmel ATmega103L to the ATmega128L microcontroller and from the RF Monolithics TR1000 radio to the Chipcon CC1000 radio. These changes required considerable TinyOS support in the form of new drivers and a completely new radio stack. While the changes were worthwhile in retrospect, they came at the cost of uncertaintly, expensive debugging, and schedule slip while the platform stabilized over the course of a few months. Consequently, we were biased against introducing more uncertainty than necessary for this new platform. We recognized substantial changes were necessary on the sensing and platform support subsystems, so based in a large part on the reports from earlier experiences with switching platforms, we opted to keep the Mica2 platform (i.e. the processor and radio) for the XSM. A more general notion of backward-compatibility required the platform to run TinyOS and also expose most of the key signals via the 51-pin connector that is standard on many motes.

3.3 Platform

The XSM, shown in Figure 3.1, integrates a platfrom with a suite of sensors. In the TinyOS community, "platform" has come to mean the microcontroller, memory, and radio subsystems, as well as supporting hardware like power management or timekeeping but *not* the sensing and signal conditioning hardware or packaging. In keeping with this tradition, this section discusses the processor, radio, and supporting subsystems.

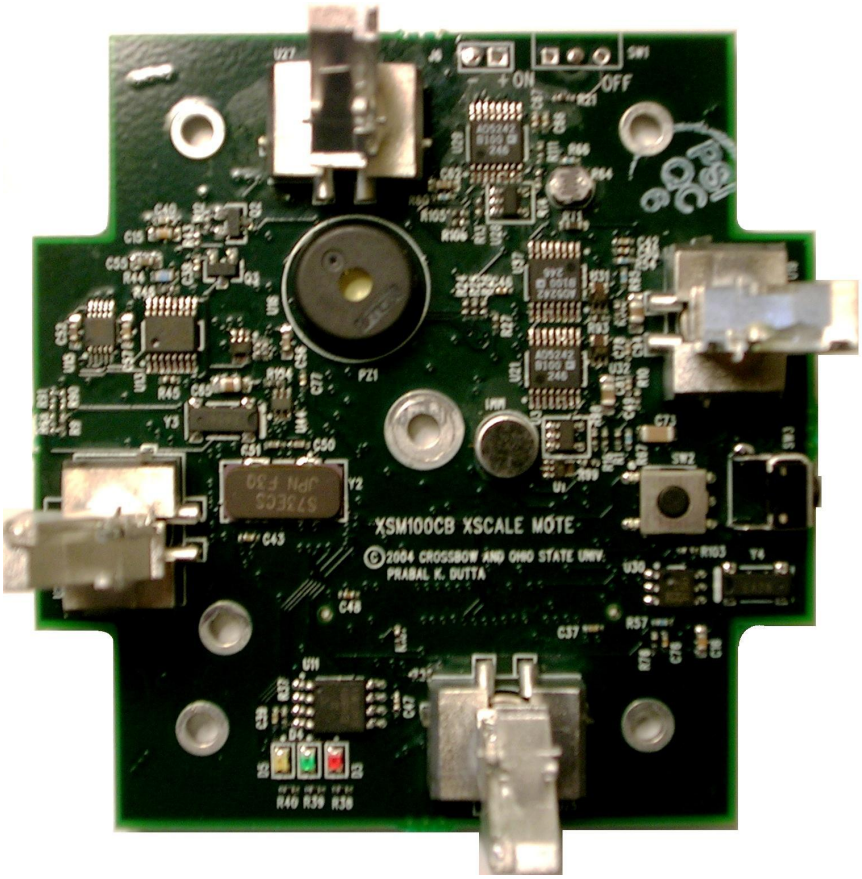

Figure 3.1: Top view of XSM (version 2).

3.3.1 Processor

Due to the Mica2-compatibility requirement, the XSM processor subsystem is almost identical to that found in the Mica2 and Mica2Dot motes. There are two minor changes which were incorporated due to supportability reasons. A versioning system now allows the software to dynamically determine the hardware revision on which the software is executing. In addition, the ADC reference voltage, AREF, is under software control in the XSM rather than being hardwired to AVCC as in the

Mica2. It makes sense to leave the ADC AREF signal unconnected from AVCC for a number of reasons. One reason is the MCU provides a facility to make this connection under software control by setting the ADMUX.REFS0 to 1 and ADMUX.REFS1 to 0. This is easily done in the hardware presentation layer. Another reason is greater flexibility: the source of the ADC reference can be selected under software control to be either AVCC or an internally generated 2.56V. A third reason is that by leaving the external AREF unconnected, the possibility of accidentally damaging the processor by shorting the external AREF to either AVCC or the internal 2.56V reference by setting ADMUX.REFS0 to 1 is eliminated. Either of these connections would cause a problem if the external AREF was not the same as AVCC or 2.56V.

3.3.2 Radio

In keeping with the requirement of Mica2-platform compatibility, the radio subsystem, like the processor, is nearly identical to the Mica2 family. However, there are two issues with the Mica2 radio design which are addressed in the XSM design.

The first issue with the Mica2 design is anisotropic radio propagation. This anisotrophy results in non-uniform radio connectivity as a function of orientation of the Mica2 mote. A second problem that was reported with the Mica2 platform was an impedance mismatch between the antenna and radio. Such a mismatch can cause RF energy to get reflected from the radio-antenna junction, rather than coupled into the antenna and radiated, resulting is diminished communications range.

In [20], Zhao, et.al, report greater than 5dBm difference in the maximum and minimum received signal strength indicator at a distance of 10ft and nearly 8dBm at 20ft, as a function of the direction in the plane, for the Mica2 radio. In other

experiments, a 40% stronger signal has been observed at a 10m range in the direction of the Mica2 power switch than in the opposite direction of the MMCA antenna connector with the motes approximately 10cm off the ground [21]. Radio anisotropy impacts media access and control, RSSI-based range estimation, localization, routing, and other algorithms.

A quarter-wave monopole constructed from a piece of wire is the predominant antenna style in use on Mica2 motes. Theoretically, a monopole antenna has uniform directionality in the plane normal to the axis of the antenna, However, a quarter-wave monopole antenna also expects an infinite ground plane. In practice, a small and approximately uniform ground plane is used when directionality is important. However, in the case of the Mica2, the antenna is located near one corner of the 1.25" by 2.25" circuit board. Hence, a ground plane is exists for only 90 degrees in the plane and principally in the direction toward the power switch. Our hypothesis is that the shape and size of the circuit board and the location of the antenna contribute the most to radio irregularity observed in the Mica2 mote.

To achieve a more isotrophic radiation pattern on the XSM, we place the antenna in the middle of the circuit board, ensuring a largely uniform ground plane in all directions. Since the XSM circuit board is larger than the Mica2, we also benefit from a larger, though still not infinite, ground plane.

The second issue with the Mica2 design is a radio-antenna impedance mismatch. While impedance mismatches are due to many underlying factors, individually addressing each factor that contributes to the mismatch is a challenge, especially when some of the factors are coupled. Consequently, we adopt a lumped-parameter approach to correcting the impedance mismatch. Such an approach does not obviate

the need for careful transmission line placement or antenna selection. Instead, we select compensation components to reduce or eliminate the mismatch after the circuit board, enclosure, antenna, and target environment is finalized. To support this ability to correct the mismatch, pads and traces are present for a compensating capacitor and inductor.

Initial experiments with the XSM radio hardware indicates that the platform has better RF characteristics than the Mica2. Figure 3.2 shows the output power of both the XSM and Mica2, programmed to transmit continuously with at a power level of 10dBm. The XSM shows a slightly cleaner signal and higher channel power. The second version of the XSM (labeled "XSM2" in Figure 3.2), incorporates several improvements including an impedance matching circuit. The output power is shown to vary for different inductor values. Unfortunately, the XSM radio outputs greater power than the XSM2. The correct values of the compensation inductor and capacitor should fix this problem, however, at the time this work was written, these values had not been identified.

In another experiment, the impedance matching characteristics of the two platforms are highlighted by comparing the transmitted (absorbed) and reflected power from the antenna into the the motes. Figure 3.3 shows reflected power from injecting a small signal into the antenna port and radio on the two motes. The injected signal is swept across a range spanning 433MHz±25MHz and with output power set as the reference at 0dB. The XSM reflects less power than the Mica2. At 433MHz, the XSM reflects about 3dB less than the Mica2. The initial XSM2 compensation circuit performed more poorly than either the Mica2 or the XSM. However, by varying the compensation inductor, we were able to reduce the amount reflection in the RF path

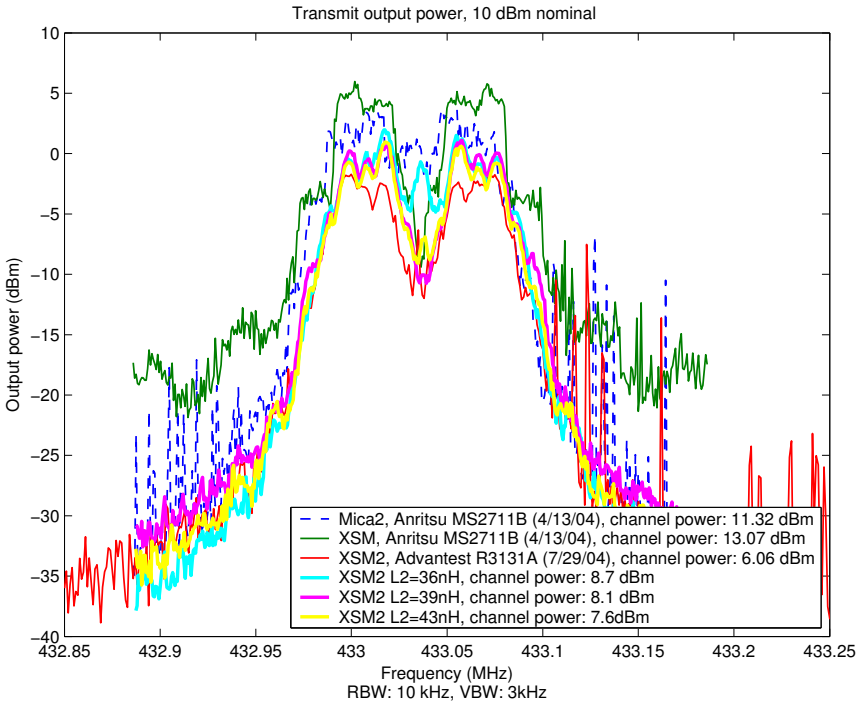

Figure 3.2: Comparison of radio transmit output power for the Mica2, XSM, and XSM2. Source: J. Polastre, C. Sharp, and R. Szewczyk, U.C. Berkeley.

by 3dB to 5dB compared with the Mica2. In normal operation, if less energy is reflected by the antenna, then more is radiated, and the RF performance is expected to improve.

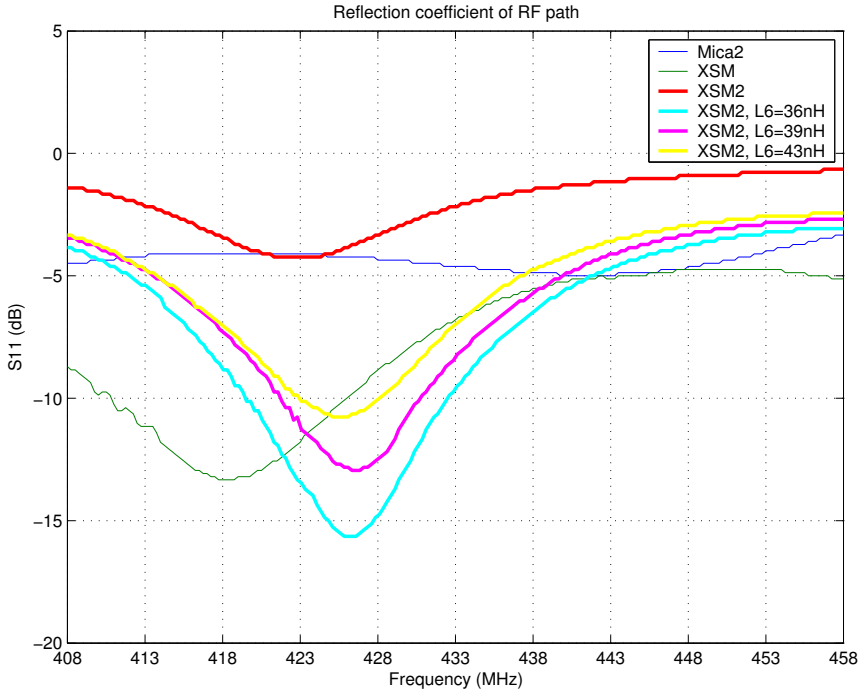

Figure 3.3: Comparison of the reflected power at the radio-antenna junction for the Mica2, XSM, and XSM2. Source: J. Polastre, C. Sharp, and R. Szewczyk, U.C. Berkeley.

3.3.3 Grenade Timer and Real-Time Clock

An important goal of the Extreme Scale project is to create robust multi-hop wireless reprogramming algorithms that allow a sensor node (the "old node") to be reprogrammed without direct (i.e. one-hop) radio or wired communications to a

node which holds a new program image (the "new node"). We argue that robustness consists of two aspects for the case of wireless reprogramming: reliability and recoverability. Reliabilty refers to the property that all old nodes within direct or indirect radio communications with a new node eventually will acquire the new image. Recoverability refers to the property that regardless of the program image executing on an old node, the old node will always (eventually) upgrade to the version that a new node is running. In other words, an incorrect or Byzantine program can never permanently disable an node from being reprogrammed with a newer image.

A number of algorithms have been proposed for providing over-the-air or multi-hop wireless reprogramming functionality including the Crossbow In-Network Programming (XNP) [22], Trickle [23], Multi-hop Over-the-Air Programming (MOAP) [24], and Deluge [25]. All of these algorithms make certain assumptions about their operation. Chief among these assumptions is that the algorithm itself is actually invoked. However, as we show in this section, such an assumption cannot be guaranteed on the existing Mica2 platform and consequently the recoverability property cannot be guaranteed either.

In a traditional preemptive operating system that runs on hardware supporting protected modes of operation, a timer is used to ensure the operating system maintains control of the processor. Before turning over control of the processor to application code running in user mode, the operating system sets a timer to interrupt the processor. When the timer interrupts, control is returned to operating system. Instructions that modify the operation of the timer are privileged, ensuring that such instructions can be executed only in protected mode by the operating system [26].

The fundamental problem in our case is that like most 8-bit microcontrollers, the Atmel ATmega128L processor used in the XSM and Mica2 platforms does not provide a true protected mode of operation. Consequently, it becomes possible for application code to take nearly complete control over the hardware, disable timers, turn off interrupts, and leave the operating system with no mechanism to preempt a misbehaving application. Hijacking of the operating system can occur either accidentally or intentionally. A description of the basic hijacking problem, and one solution to it, can be found in [27].

At the time of this work, the standard mode of compiling wirelessly reprogrammable applications under TinyOS is to create a monolithic program image consisting of the operating system, a wireless reprogramming component, and the user application. These three components are expected to co-exist in cooperative harmony. There are many ways to permanently disable wireless reprogramming under this scheme. Simply downloading an application like "Blink" that does not include the wireless reprogramming module at all is one way:

```
/opt/tinyos-1.x/apps/Blink
$ make mica2 install
```

In this case, the basic ability to wirelessly reprogram is altogether lost. Inserting an atomic block that never exits works as well:

```
atomic { while (TRUE) {} }
```

And for the more robust operating system that makes use of the watchdog timer, the following code will globally disable all interrupts, and loop endlessly, clearing the watchdog timer. This approach is a slight variation on the previous example:

```
        cli
loop:   wdr
        jmp loop
```

The problem in the last example is that when interrupts are disabled and the watchdog timer is cleared periodically, there is no mechanism for the operating system to regain control. Even if the interrupt vectors are all in a protected code segment, the `cli`, `wdr`, and `jmp` are fairly common instructions so just their presence in code is not unusual, especially in interrupt handlers. As a result, it appears non-trivial to "automatically" detect rogue code. Consider the following "seemingly" normal program that seems to loop until something happens (but R and b can be chosen such that nothing ever happens):

```
            cbi     R,b
            cli
    loop:   wdr
            ...
            sbis    R,b
            jmp     loop
            sei
```

Atmel has provided a rich set of features making the ATmega128L microcontroller in-application programmable. Atmel even provides a "quasi-protected" mode of operation that, when enabled through various combinations of fuse settings, makes it impossible for application code to modify the bootloader or interrupt vectors and handlers. However, while these features can protect the bootloader and the interrupt vectors and handlers from the application code, and even the application code from itself, the features do nothing to guarantee that control eventually returns to trusted code like the bootloader.

Unfortunately, there is no facility on the ATmega128L to guarantee that execution eventually returns to trusted code. Furthermore, since it is impractical to hand-reprogram 10,000 nodes in the event that a misbehaving image is downloaded, we choose to implement a grenade timer similar to the one described in [27]. A grenade

timer is like a watchdog timer which cannot be reset. The most succint description of the grenade timer is that "once started, the timer cannot be stopped, only sped up." The key features of our implementation include:

- **Asynchronous Trigger:** The grenade timer may be fired by any software at any time.
- **Adjustable Timeout:** The amount of time T_{fizz} that the grenade timer "fizzes" can be adjusted within bounds, typically by the bootloader, until the grenade timer's "pin is pulled." A software mechanism could exist that allows application code to request a T_{fizz} value, within certain bounds, during the next reset.
- **One-Shot Latch-out:** Once the grenade timer is started, it cannot be stopped and T_{fizz} cannot be changed.
- **Alternate Uses:** As long as the grenade timer has *not* been started, the real-time clock used to implement the grenade timer remains accessible to either the bootloader or the application and can be used freely.

Our grenade timer circuit is shown in Figure 3.4. The circuit works as follows. After a power-on-reset (POR), capacitor C_{60} begins charging through R_{69} from an initially discharged state. As long as the voltage across C_{60} is below V_{IH}, the high-level input voltage of the AND gate (U_{39}), the output of the AND gate remains low and the processor remains in reset. The processor's RD line is automatically tri-stated during a reset and hence tracks the voltage across capacitor C_{60}. The AND gate's other input is pulled high by R_{57} since the INT output of the DS2417 (U_{30}) is asserted low only during an interrupt interval and not during a POR.

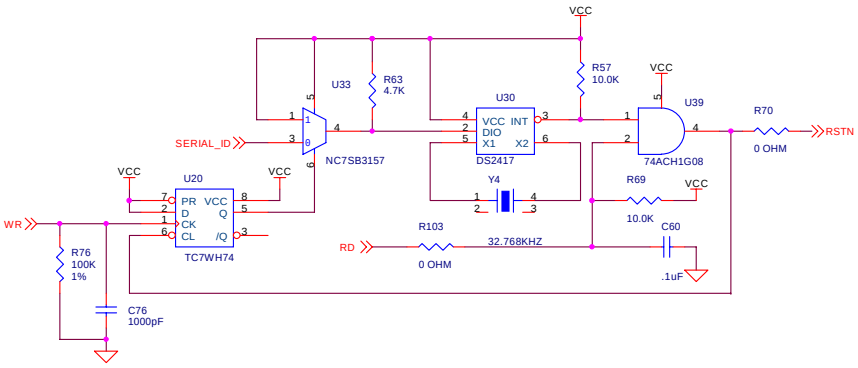

Figure 3.4: Grenade timer and real-time clock circuit.

The time constant, τ, of the $R_{69}C_{60}$-circuit is 1ms and the equation for the voltage across capacitor C_{60} is:

$$V(t) = V_{CC}(1 - e^{-t/\tau}) \tag{3.1}$$

Rearranging to solve for t, we have:

$$t = -\tau \ln\left(1 - \frac{V(t)}{V_{CC}}\right) \tag{3.2}$$

At a supply voltage of 3V, the AND gate's V_{IH} is 2.1V. Substituting 3V and 2.1V for V_{CC} and $V(t)$, respectively, gives $t = 1.2$ms. Therefore, after 1.2ms, both of the AND gate's inputs are high and the AND gate's output goes high as well, allowing the processor to exit the reset state and begin program execution.

The output of the AND gate is also connected to the asynchronous clear input of the D-type flip-flop U_{20}. By waiting 1.2ms before asserting this line, the power is given enough time to stabilize before the flip-flop's state is cleared (set low). Whenever the flip-flops's output, Q, is low, the multiplexer/analog SPDT switch (U_{33}), connects the processor's SERIAL_ID signal to the DS2417's DIO pin, allowing the processor

to communicate with the DS2417, a real-time clock with a built-in timer. To start the grenade timer, the bootloader loads the value of T_{fizz} into the DS2417 using the Dallas 1-wire bus (SERIAL_ID) and enables the device. The legal T_{fizz} values are: 1s, 4s, 32s, 64s, 2048s (34.13min), 4096s (68.27min), 65,536s (18.30hrs), and 131,072s (36.41hrs).[2]

The bootloader or application code can start the grenade timer and ensure that the no subsequent operation can alter or disable the grenade timer, by enabling the processor's WR line for output and asserting it high. Doing so creates a low-to-high transition which has the effect of clocking the positive edge-triggered flip-flop. Once clocked, the flip-flop output, Q, assumes the value of its input, D. Since D is tied to V_{CC}, the value of Q goes after the first clock and remains high until it is asynchronously cleared.

Once Q is high, the multiplexer/analog SPDT switch (U_{33}), disconnects or "latches out" the processor from communicating with the DS2417. No additional clocking can reverse this latch-out of the processor until the next DS2417 interrupt occurs or the processor asserts RD low, both of which asynchronously clears the D flip flop, resets the processor, and returns control to the bootloader. We note that if neither the bootloader nor the application code asserts WR high, the DS2417 remains accessible to the processor and can be used as a real-time clock.

We note that Atmel could have provided nearly equivalent functionality by including a fuse setting that disabled non-bootloader control of a timer and its interrupt. Alternately, Atmel could have provided a non-maskable external interrupt (i.e. an

[2]These values correspond to the U30's interval select register IS$\langle 2:0\rangle$ values of 000 to 111, respectively. The processor must also enable U30's oscillator by writing OSC$\langle 1:0\rangle = 11$ and setting the interrupt enable register, IE, by writing a 1 to this register. After this sequence of operations, the DS2417 will be configured to assert INT low for 122μs every T_{fizz} seconds.

interrupt that cannot be disabled at all). A non-maskable interrupt, when coupled with the bootloader protection mechanisms and an external source of interrupts, could have provided a more suitable quasi-protected mode. We do recognize that in certain embedded applications, all interrupts are legitimately disabled during execution of the interrupt handler for a variety of quite valid reasons. However, we argue that it is reasonable to assume that an upper bound exists on the amount of time that any interrupt handler, or atomic code block, is executing within a critical section. If this upper bound can be expressed in clock cycles, we envision that a write-once register could be used to store the maximum amount of time allowed between a non-maskable interrupt being triggered and either enabling interrupt or execution control being forcibly transferred to the non-maskable interrupt handler. Perhaps such features, which can be found in some processors today, will become more common in future microcontrollers.

The grenade timer can guarantee that the bootloader eventually regains control of the processor. By properly setting the processor's fusemap, the bootloader and the interrupt vectors and handlers can be protected from the application code. However, we have not yet discussed what the bootloader must do to *recognize* an outdated (and perhaps misbehaving) image and *recover* by either reverting to an earlier image or downloading a newer image. Those topics will be covered in a later section.

3.3.4 Network Bootloader

Our notion of a network bootloader works in conjuction with the grenade timer to implement the retaskability and recoverability requirements. The bootloader consists of a minimal network stack, a network reprogramming module, the grenade timer

drivers, and a small application. This bootloader should be factory programmed onto the XSM and the fuses should be set such that the bootloader can be erased only by manual reprogramming. The bootloader is responsible for the following operations:

- **Version Checking:** Check the version of the currently running application image. The current image's version number should be stored in an area of non-volatile memory that is protected from the application code.
- **Availability Checking:** Check with neighboring nodes for a newer version of the application image by broadcasting a request and listening for a response.
- **Download:** Initiate the downloading of a new application image, if any, from a neighboring node. Keep track of the application image version in boot flash rather than in the application image or application adjustable memory.
- **Integrity Checking:** Verify the integrity of a new application image and its version number through a message authentication code.
- **Programming:** Move or copy the newly downloaded application image to make it the default image.
- **Arm Grenade Timer:** Enable and arm the grenade timer to reset the processor in time T_{fizz}.. Disable any further operations on the grenade timer through the latch-out feature.
- **Load Application:** Jump to the application entry point and begin execution.

In addition to the responsibilities of the bootloader, the compiler/linker needs to be configured to relocate the interrupts vectors into a protected region of memory

(in particular, the RESET vector) and to generate application images which have a non-zero entry point.

3.3.5 User Interface

The XSM user interface includes two buttons, three LEDs, a sounder, and a 51-pin programming port. The RESET button is connected to the processor's reset line and depressing this button causes the processor to enter a reset state and releasing this button allows the processor to begin program execution. The USER button is connected to a processor interrupt line and requires an interrupt handler to process the button pushes. Although user applications may modify the availability or functionality of the USER button, the designer's intent is for this button to serve as a multi-function input that works in conjuction with the RESET button.

The default configuration of the XSM does not include an ON/OFF power switch. While this design choice appears to violate generally accepted design principles, we believe that in this case the choice is warranted. Since the XSM was designed for use in an experimental network of 10,000 nodes, we wanted to minimize the number of things which could go wrong including, for example, accidentally deploying nodes with the power switch set to the OFF position. Another motivation was that we never imagined that the nodes would be truly turned off in our application. Instead, the nodes could exhibit various degrees of vigilance ranging from fully awake to deeply asleep. We did recognize that for research and development purposes, a missing power switch is just plain annoying. Consequently, we included circuit board pads and traces so users could populate the power switch after market. The final motivation for eliminating the power switch was cost.

The sounder subsystem includes a transducer capable of producing 98dB in the 4kHz to 5kHz acoustic range. The transducer is driven by an op amp that is powered from a dual-output charge pump which generates $\pm 2 \times V_{CC}$. The piezo transducer's oscillation frequency is software-controllable but the transducer performs best at 4.5kHz.

3.4 Sensors

Sensor selection is a fundamental task in the design of a wireless sensor node. Choosing the right mix of sensors for the application at hand can improve discriminability, reduce density, increase lifetime, simplify deployment, reduce computational complexity, and lower probability of discovery – in short, improve performance.

Despite the plethora of available sensors, no primitive sensors exist that detect people, vehicles, or other potential objects of interest. Instead, sensors are used to detect various features of the targets like thermal or ferro-magnetic signatures. It can be inferred from the presence of these analogues that, with some probability, the target phenomenon exists. However, it should be clear that this estimation is an imperfect process in which multiple unrelated phenomena can cause indistinguishable sensor outputs. Additionally, all real-world signals are corrupted by noise which limits a system's effectiveness. For these reasons, in addition to sensor selection, the related topics of signal detection, parameter estimation, and pattern recognition are important [28, 29, 30].

Although several factors contributed to our final choice, the target phenomenologies (i.e. the perturbations to the environment that our targets are likely to cause) drove the sensor selection process:

- **Civilian:** A civilian is likely to disrupt the environment thermally, seismically, acoustically, electrically, chemically, and optically. Human body heat is emitted as infra red energy omnidirectionally from the source. Human footsteps are impulsive signals that cause ringing at the natural frequencies of the ground. The resonant oscillations are damped and propagated through the ground. Footsteps also create impulsive acoustic signals that travel through the air at a different speed than the seismic effects of footsteps travel through the ground. A person's body can be considered a dielectric that causes a change in an ambient electric field. Humans emit a complex chemical trail that dogs can easily detect and specialized sensors can detect certain chemical emissions. A person reflects and absorbs light rays and can be detected using a camera. A person also reflects and scatters optical, electromagnetic, acoustic, and ultrasonic signals.

- **Solder:** An armed soldier is likely to have a signature that is a superset of an unarmed person's signature. We expect a soldier to carry a gun and other equipment that contains ferromagnetic materials. As a result, we would expect a soldier to have a magnetic signature that most civilians would not have. A soldier's magnetic signature is due to the disturbance in the ambient (earth's) magnetic field caused by the presence of such ferromagnetic material. We might also expect that a soldier would better reflect and scatter electromagnetic signals like radar due to the metallic content on his person.

- **Vehicle:** A vehicle is likely to disrupt the environment thermally, seismically, acoustically, electrically, magnetically, chemically, and optically. Like humans, vehicles have a thermal signature consisting of "hotspots" like the engine region

and a plume of hot exhaust. Both rolling and tracked vehicles have detectable seismic and acoustic signatures. Tracked vehicles, in particular, have highly characteristic mechanical signatures due to the rhythmic clicks and oscillations of the tracks whereas wheeled vehicles tend to exhibit wideband acoustic energy. Vehicles contain a considerable metallic mass that affects ambient electric and magnetic fields more strongly and in an area much larger than a soldier. Vehicles emit chemicals like carbon monoxide and carbon dioxide as a side effect of combustion. Vehicles also reflect, scatter, and absorb optical, electromagnetic, acoustic, and ultrasonic signals.

Of course, there is a tension between the richness of a sensor's output and the resources required to process the signals it generates. Imaging and radar sensors, for example, can provide an immense amount of information but the algorithms needed to extract this information can have high space, time, or message complexity, making these sensors unsuitable for use on energy-constrained leaf nodes in a sensor network. We take that view that a collection of simple sensors, each of which requires low complexity algorithms for processing, can collaborate as an ensemble to provide a higher detection signal-to-noise ratio and a lower classification error rate than is otherwise possible on devices of this class. This view lead us to choose acoustic, magnetic, and passive infrared as the key components of the XSM sensor suite, with straightforward target detection and discriminability playing an important role in our decision.

The strengths of acoustic sensors include long sensing range, high-fidelity, no line-of-sight requirement, and passive nature. Weaknesses include poorly defined target

phenomenologies for certain target classes, high sampling rates for estimation, and high time and space complexity for signal processing.

The strengths of magnetic sensors include well defined far-field target phenomenologies, discrimination of ferrous objects, no line-of-sight requirement, and passive nature. Weaknesses include poorly defined near-field target phenomenologies, high continuous current draw, and limited sensing range.

The strengths of passive infrared sensors include excellent sensitivity, excellent selectivity, low quiesent current, and passive nature. Weaknesses include line-of-sight requirement and reduced sensitivity when ambient temperatures are the same as that of the target.

By fusing simultaneous detections from these sensors, we can discriminate our target classes using the following classification predicates:

$$civilian = pir \wedge \neg mag \tag{3.3}$$

$$soldier = pir \wedge mag \tag{3.4}$$

$$vehicle = pir \wedge mag \wedge mic \tag{3.5}$$

where *pir* refers to a passive infrared detection, *mag* refers to presence of a ferromagnetic object near a sensor, and *mic* refers to either wideband acoustic energy or the presence of harmonics in the acoustic spectra.

3.4.1 Acoustic

The JLI Electronics F6027AP microphone is at the heart of the acoustic subsystem. This sensor is an omnidirectional back electret condenser microphone cartridge. The microphone sensitivity is -46 ± 2dB (0dB is 1V/Pa at 1kHz), the frequency

response is 20Hz to 16kHz. The microphone is cylindrical shaped and 2.5mm tall by 6mm in diameter. This sensor was chosen because of its good sensitivity, small size, leaded terminals. and overall price/performance.

The output of the microphone is capacitively-coupled and amplified using using an op amp in an inverting configuration with gain $G_1 = -91$. The output of this gain stage is again AC-coupled and amplified using an inverting op amp configuration. The gain of the second stage of amplification is variable using an 8-bit, digitally-controlled potentiometer. The gain of the second amplifier stage is variable across a range of $G_2 = -1.1$ to -91, adjustable to one of 256-values along a linear scale (a logarithmical would have been better). Since these gain stages are cascaded, a total gain of $G_1 G_2 = 100$ to 8300, or approximately 40dB to 80dB is possible.

The output of the two gain stages is again capacitively-coupled to eliminate bias and then low pass filtered. The low pass filter is configured as a single-supply 2-pole Sallen-Key filter with Butterworth characteristics for minimum passband ripple [31]. The low pass filter circuit is shown in Figure 3.5. The filter transfer function is:

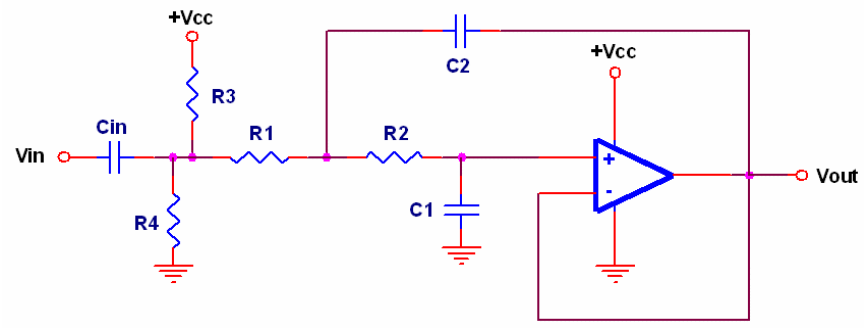

Figure 3.5: Single supply Sallen-Key low pass filter with Butterworth characteristics.

$$H(s) = \frac{V_o}{V_i} = \frac{k}{As^2 + Bs + 1} \tag{3.6}$$

where $k = 1$ since we use a unity gain op amp configuration, $A = R_1R_2C_1C_2$ and $B = R_1C_1 + R_2C_1 + R_1C_2(1 - k)$. The filter's cutoff frequency is:

$$f_c = \frac{1}{2\pi\sqrt{R_1R_2C_1C_2}} \tag{3.7}$$

In our implementation R_1 and R_2 are each composed of a fixed resistor (1.1kΩ) in series with a digitally-controlled potentiometer (0 to 100kΩ). The values of R_1 and R_2 should be set equal for Butterworth characteristics. In our implementation, $C_1 = 0.01\mu$F is and $C_2 = 0.022\mu$F. The ratio of $C_2/C_1 = 2$ is used for Butterworth filter characteristics. The range of cutoff frequencies possible vary from approximately 100Hz when $R_1 = R_2 = 101.1$kΩ to 10kHz when $R_1 = R_2 = 1.1$kΩ. Resistors R_3 and R_4 are 100kΩ resistors which set the signal bias to $V_{CC}/2$. This biasing is necessary since the circuit operates from a single supply.

The output of the low pass filter is again capacitively-coupled to eliminate bias and then high pass filtered. Like the low pass filter, the high pass filter is configured in a 2-pole Sallen-Key configuration with Butterworth characteristics. The high pass filter circuit is shown in Figure 3.6. The filter transfer function is:

$$H(s) = \frac{V_o}{V_i} = \frac{s^2kA}{As^2 + Cs + 1} \tag{3.8}$$

where $C = R_2C_2 + R_2C_1 + R_1C_2(1 - k)$. The coefficients A and k are the same as in the low pass filter case, and the cutoff frequency is computed in an identical manner as well. The roles of the resistors and capacitors are reversed in the high pass filter case so the values of C_1 and C_2 should be set equal and the ratio $R_1/R_2 = 2$ provides Butterworth characteristics. In our implementation R_1 and R_2 are each

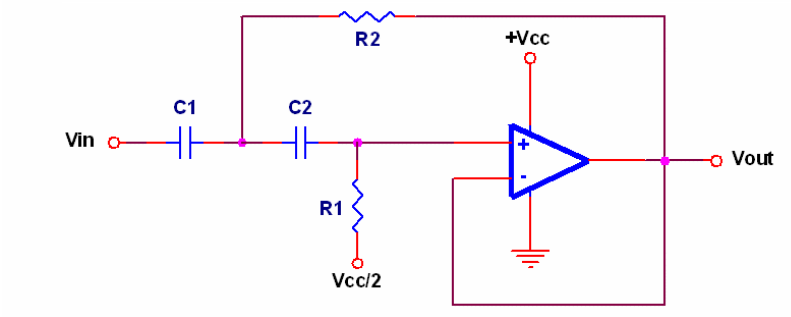

Figure 3.6: Single supply Sallen-Key high pass filter with Butterworth characteristics.

composed of a fixed resistor (470Ω and 240Ω, respectively) in series with a digitally-controlled potentiometer (0 to 100kΩ). The range of cutoff frequencies possible vary from approximately 20Hz when $R_1 = 100.47\text{k}\Omega$ and $R_2 = 50.24\text{k}\Omega$ to 4.7kHz when $R_1 = 470\Omega$ and $R_2 = 240\Omega$. Although not shown, the signal bias is set to $V_{CC}/2$ using a similar approach as in the low pass filter case. The cascaded filter pair implements a tunable bandpass filter with independently adjustable cutoffs. Figure 3.7 shows the frequency response of the low pass filter, high pass filter, and the pair cascaded together to realize a bandpass filter.

The output of the high pass filter is connected to an ADC channel as well as the negative input of a comparator. The positive input of the comparator is connected to the wiper terminal of a digital potentiometer configured as a voltage divider. The output of the comparator is connected to an interrupt line on the processor. This circuit allows us to set a detection threshold which, when exceeded, interrupts the processor, eliminating the need for continuous sampling and signal processing. It it this *wakeup circuit* that supports energy-efficient passive vigilance.

3.4.2 Magnetic

The Honeywell HMC1052 magnetoresistive sensor is at the core of the magnetic sensing subsystem. This sensor was chosen because of its two-axis orthogonal sensing, small size, low-voltage operation, low-power consumption, high bandwidth, low latency, and miniature surface mount package [32]. Internally, the magnetometer is configured as a Wheatstone bridge whose output is differentially amplified by an instrumentation amplifier with a gain $G_1 = 247$. The output of this instrumentation amplifier is low pass filtered using an RC-circuit with cutoff frequency $f_c = 19$Hz. The output of the low pass filter is fed into the non-inverting input of a second instrumentation amplifier with gain $G_2 = 78$, for a combined gain approaching 20,000 or 86dB. The inverting input of the second instrumentation amplifier is connected to the wiper terminal of a digital potentiometer configured as a voltage divider. The instrumentation amplifier's inverting input is the only user-adjustable parameter in the magnetometer subsystem and varying it adjusts the bias point of the amplifier.

3.4.3 Passive Infrared

The Kube Electronics C172 pyroelectric sensor is at the core of the passive infrared subsystem. This sensor consists of two physically separated pyroelectric sensing elements and a JFET amplifier sealed into a standard hermetic metal TO-5 housing with an integrated optical filter. The sensor is fitted with a compact "cone optics reflector" that obviates the need for additional lenses. Passive infrared (PIR) sensors are very popular for detecting human and vehicle presence. These devices are the central component in many motion sensors for automatic lighting, security systems, and electric doors. PIR sensors are a good choice for presence and motion detection

owing to their low power, small size, high sensitivity, low cost, and broad availability. The sensors themselves draw just a few μWatts of power and sensing circuits can be designed with multi-year lifetimes.

The passive infrared subsystem is composed of several blocks including power control, power supply filtering, quad sensors, active band pass filters, a summing op amp, and a window comparator. Each PIR sensor has a 100° field-of-view. The four PIR sensors are mounted on 90° intervals so their fields-of-view overlap slightly. The sensing and active filter blocks operate in parallel, to a degree, until they are combined into a single analog signal at the summing op amp whose output is connected to an ADC input channel on the processor. The frequency response of the filtering electronics is shown in Figure B.2

This analog signal that is fed to the processor's ADC is also fed to the negative input of a window comparator. The comparator's positive input is the only user-adjustable parameter in the PIR subsystem.

3.4.4 Photo and Temperature

The XSM includes a CdS photocell and a thermistor which share an ADC channel. The photocell allows the XSM to measure the ambient light level. A historical profile of light levels can be used to predict the expected number of daylight hours or discriminate variations in cloud cover from nightfall. The photocell forms the top-half of a voltage divider and is connected in series with a 10kΩ resistor. Higher light levels correspond to lower photocell resistances which in turn correspond to higher ADC values.

The thermistor allows the XSM to measure ambient temperature. Such a capability has a number of uses. For example, some of the other sensors have temperature coefficients for parameters like sensitivity and bias point. If the ambient temperature is known, we can compensate for variations in these parameters. In other cases, a node may power down in the event that the ambient temperature exceeds the node's operating range. Like the photocell, the thermistor forms the top-half of a voltage divider and is connected in series with the same 10kΩ resistor as the photocell. Care must be taken to ensure that the operation of the photecell and thermistor is mutually exclusive. In contrast with the photocell behavior, higher temperatures result in a higher thermistor resistance, which in turn corresponds to a lower ADC value.

3.4.5 Acceleration

The XSM includes pads and traces for an Analog Devices ADXL202AE accelerometer, and supporting electronics, as shown in Figure 3.9. The ADXL202AE is a small, low-cost, solid-state, $\pm 2g$, dual-axis sensor. We were unable to include this sensor in the production units due to cost considerations. However, the pads were provided to gain an additional degree of freedom for our own research and also in the hopes that future researchers might be able to use this platform to measure acceleration with some very minor after market modifications.

3.5 Power

The estimated power consumption of the various XSM subsystem is shown in Table 3.1. The key point to note is that the acoustic and PIR subsystems together draw 650μA, or approximately 2mW, during *continuous operation*. This falls within our acceptable average power budget of 6mW. We note that none of the remaining

Subsystem	State	Current (at 3V)	Units
Acoustic	off	1	μA
Acoustic	on	350	μA
Magnetometer	off	1	μA
Magnetometer	on	3	mA
PIR	off	1	μA
PIR	on	300	μA
Sounder	off	1	μA
Sounder	on	16	mA
Radio	off	1	μA
Radio	receive	8	mA
Radio	transmit	16	mA
Processor	sleep	10	μA
Processor	active	8	mA

Table 3.1: Estimated current draw of XSM subsystems.

subsystems can be powered continuously without exceeding our power budget. Consequently, we are forced to use a low-power listen mode of communications or perhaps something more radical like the acoustic sensing channel as a wakeup "radio" since the XSM includes a sounder capable of producing 98dB of output in the 4kHz to 5kHz frequency range.[3] It is also possible that scheduled communications might suffice but it is not obvious that the system latency requirements can be met with such an approach.

3.6 Packaging

Sensor nodes for intrusion detection may experience diverse and hostile environments with wind, rain, snow, flood, heat, cold, terrain, and canopy. The sensor

[3]The wisdom of using such high probability of detect channel in an application for intrusion detection notwithstanding.

packaging is responsible for protecting the delicate electronics from these elements. In addition, the packaging can affect the sensing and communications processes either positively or negatively. Figure 3.10 shows the XSM enclosure and how the electronics and batteries are mounted. The XSM enclosure is a commercial-off-the-shelf plastic product that has been modified to suit our needs. Since the enclosure plastic is constructed from a material that is opaque to infrared, each side has a cutout for mounting a PIR-transparent window. Similarly, a number of holes on each side allow acoustic signals to pass through. A water-resistant windscreen mounted inside the enclosure sensor reduces wind noise and protects the electronics from light rain. A telescoping antenna is mounted to the circuit board and protrudes through the top of the enclosure. A rubber plunger makes the RESET and USER buttons easily accessible yet unexposed.

3.7 Summary

We have presented the requirements, philosophy, and design of the eXtreme Scale Mote. This is the first highly-integrated mote-class sensor node that directly supports recoverability and passive vigilance – two essential features for large-scale and long-lived operation. Recoverability is achieved through the use of a grenade timer. We are unaware of any other device which implements a hardware grenade timer explicitly for the purposes of recoverability. Passive vigilance is achieved using wakeup sensing circuits. These circuits, through the use of low-power sensing and signal conditioning electronics, combined with an event-driven sensor interface, allow the processor to sleep a large fraction of the time, extending the system's lifetime.

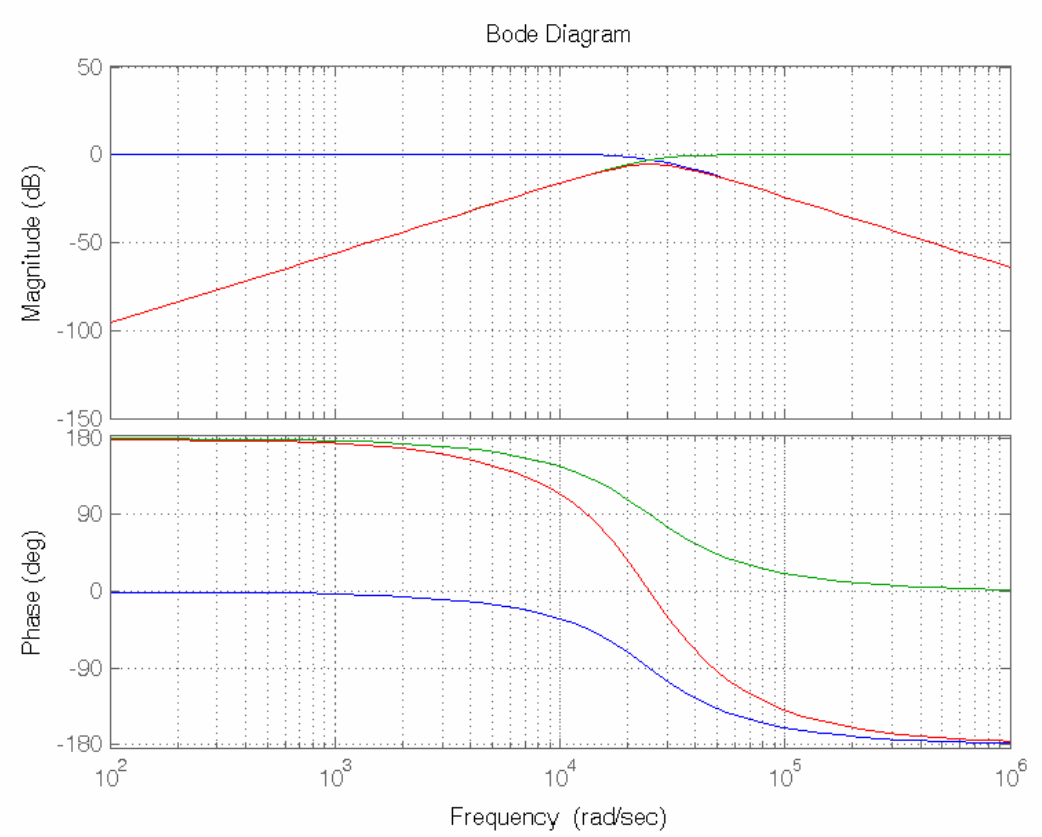

Figure 3.7: Bode diagram of cascaded Sallen-Key low pass and high pass filters with both cutoff frequencies set to 4kHz (25krad/sec). The low pass filter magnitude and phase (blue) at 10^2 rad/sec is 0dB and 0°, respectively, and at 10^6 rad/sec approaches -60dB and -180°. The high pass filter magnitude and phase (green) at 10^2 rad/sec is approximately -90dB and 180°, respectively, and at 10^6 rad/sec approaches 0dB and 0°. The cascaded filter response implements a band pass filter that is the sum of the low pass and high pass filters. The band pass filter magnitude and phase (red) at 10^2 rad/sec is approximately -90dB and 180°, respectively, and at 10^6 rad/sec approaches -60dB and -180°.

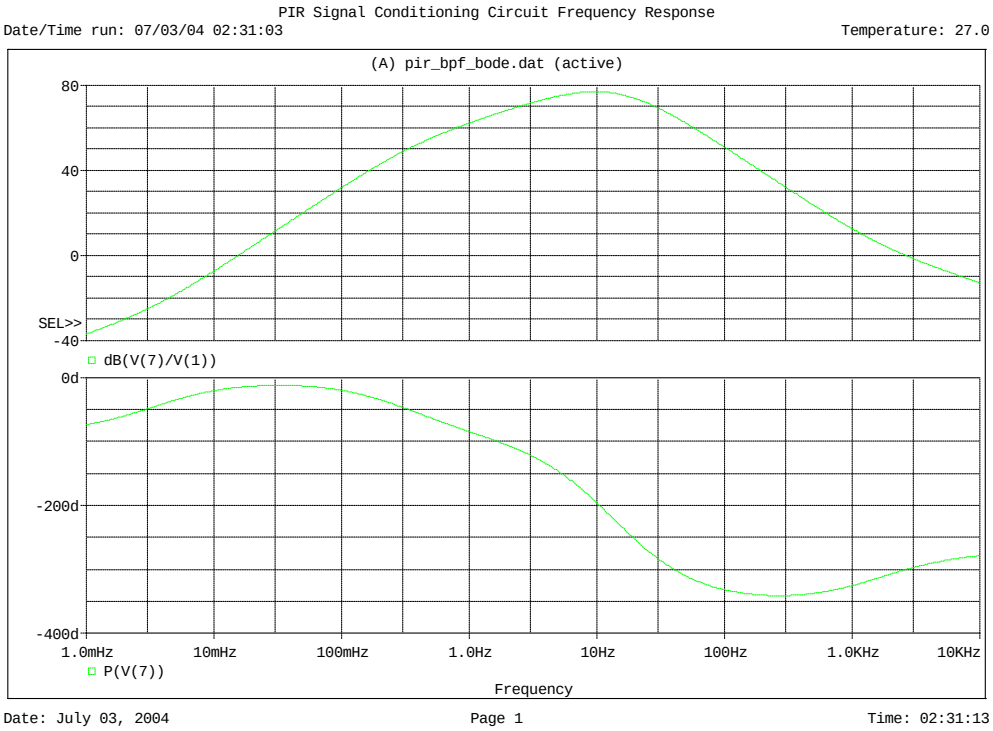

Figure 3.8: Frequency response of PIR signal conditioning circuit for XSM.

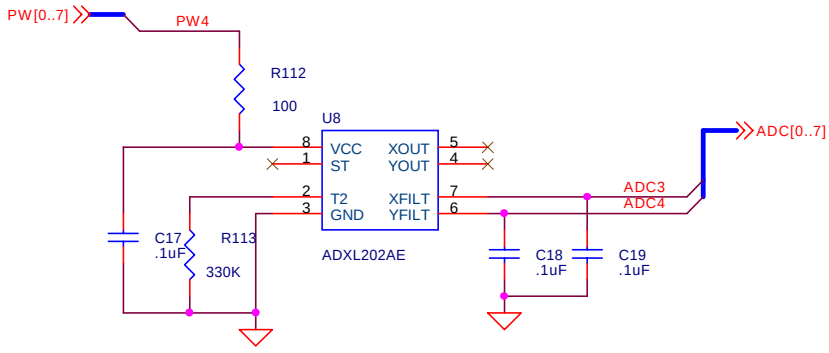

Figure 3.9: Accelerometer circuit available on the XSM (unpopulated).

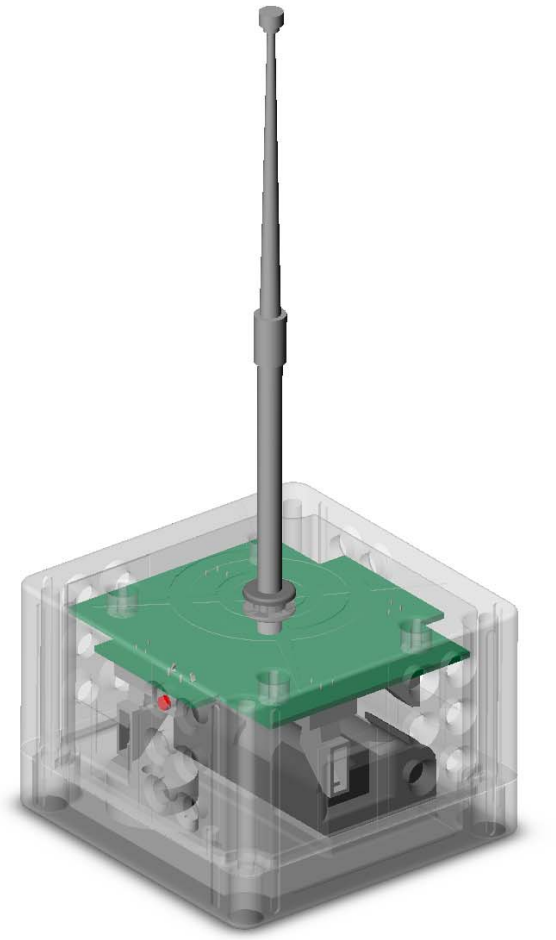

Figure 3.10: This model shows how the XSM electronics and batteries are mounted in the enclosure. Source: Crossbow Technology.