ScoutNode: A Multimodal Sensor Node for Wide Area Sensor Networks

Abhishek Sharma, Kevin Yoon, Dave Vanhorn, Matheiu Dube, Volus McKenna and Michael S. Bauer
MobileFusion Inc.
2715 Sarah St. Pittsburgh, PA, USA
Email: abhi@mobilefusioninc.com

***Abstract*—In this paper, we present a commercial sensor node platform called ScoutNode. It is based on a novel modular architecture that supports three very powerful functionalities - multimodal support with a maximum of twenty five sensors supported on a single node that includes multiple video cameras; a user-friendly API for application development, and interoperability with a variety of 802.x hardware and protocols. ScoutNode architecture has been developed to support the operational concept and requirements of unattended ground sensor (UGS) nodes for net-centric information collection for the military. It offers a very comprehensive sensing, processing, and communication platform for building a variety of military and commercial applications. The ScoutNode platform described in this paper is fully functional and a node based on this platform designed specially for the military was tested successfully by the US Army.**

I. Introduction

Next-generation sensor network applications, both for the military and the commercial world, will require more powerful sensor nodes, also known as motes [1], [2]. These sensor nodes are expected to function in several roles, including surveillance, target acquisition, situational awareness, and Chemical, Biological, Radiological and Nuclear (CBRN) early warning. To support these capabilities, it is becoming necessary to develop new architectures and design concepts that offer multimodal sensing without sacrificing the attractive low size, weight, and power (SWAP) capability offered by the conventional motes.

Commercial sensor nodes that are available in today's market such as, iMote2 from Crossbow [3], IPsensor Node from Archrock [4], and a family of sensor nodes from MicroStrain [5], have significant shortcomings that limit their potential to serve as a foundation for next-generation sensor network applications. Chief among these limitations are the lack of multi-sensor support, low processing power, and the lack of communication interoperability [6]. These shortcomings render the current commercial solutions inadequate for the challenge of developing next-generation sensing applications.

We introduce a novel sensor platform called ScoutNode to address this challenge. ScoutNode has been designed from the bottom up to provide capabilities that make it ideal to serve as the foundational platform for the next-generation sensor network applications.

ScoutNode introduces a novel architecture that offers several powerful features:

- *Multimodal* - Each node is embedded with a novel sensor interface processor called the *Universal Sensor Port (USP)*. This interface supports multiple sensors (maximum of 25 per node), such as video, audio, acoustic, passive infra-red (PIR), accelerometer, CBRN, seismic, magnetometer, temperature, pressure, and RF sensors.
- *Programmable* - The controller module of the node is a low-power processor with a small footprint real-time embedded Linux (kernel 2.6.22) distribution. This module contains several C/C++ APIs for data I/O, sensor control, radio control, data processing and power management (can manage overall budget ranging from 2W all the way down to 40mW).
- *Interoperable* - The controller software supports a variety of IP based 802.x radio modules along with the necessary drivers and control API

ScoutNode is well suited for a diverse range of applications that require the flexibility of multi-modal sensing, programmable customization, and ad-hoc mesh networks based on interoperability of sensor nodes. Therefore, ScoutNode offers a comprehensive solution for next-generation sensor network applications.

ScoutNode has been successfully tested by the United States Army through rigorous field trials. These trials subjected ScoutNode to a variety of tests that evaluated different operational and deployment characteristics, chief among them is to determine the utility of such a sensor system to the soldiers.

This paper details the design of the ScoutNode platform to provide researchers a description of its functionalities and performance. Through this introduction, we expect to foster an open-source community that will collaborate to develop powerful next-generation applications on the ScoutNode. It is our belief that ScoutNode will be the next-generation multimodal sensor platform.

II. Related Work

A sensor node, or mote, in a wide area sensor network is a node capable of performing autonomous sensing, and communicating the information over a self-organizing and fault-tolerant wireless network. Since these nodes possess low SWAP (size, weight, and power) capabilities at a low price they have become very attractive for military intelligence surveillance and reconnaissance (ISR) infrastructure. Due to this very reason the US Army Future Combat System (FCS)

Fig. 1: In this figure we show wide a variety of sensor nodes that have been developed on the ScoutNode platform. From right to left; the laptop shows our real-time analytics of the environment being monitoring. The ball adjacent to the laptop shows a basic sensor node based on the ScoutNode platform it consists of four cameras, along with acoustic, audio, PIR sensors. The middle sensor node is ruggadized version of the basic sensor node, according to the specification of the United States Army. The rightmost sensor node consists of an extensive array of sensors (listed in Table 1). Notice the small form factor of the node when compared to a whiteboard market making it ideal for remote deployment and concealment

UGS program [7] is, at the present time, actively focusing on developing the next-generation sensor nodes for the military. These nodes, while being more capable at sensing (by supporting multiple sensors on the same node), would also be smarter, more power efficient, interoperable with a variety of communication systems/gateways and highly configurable for various missions. Depending on the mission, the nodes have been divided into two categories: Tactical-UGS (T-UGS), which includes Intelligence, Surveillance and Reconnaissance (ISR)-UGS and Chemical, Biological, Radiological and Nuclear (CBRN)-UGS; and Urban-UGS (U-UGS), also known as Urban Military Operations in Urban Terrain (MOUT) Advanced Sensor System. In 2007 when we started to prepare ourselves to participate in the UGS program we carried out in-depth analysis of UGS requirements and performed subsequent research on the architecture, design and implementation of available sensor nodes - in both the academic and commercial worlds (for a comprehensive survey please see [8]; commercial products from Crossbow [3], Archrock [4], MicroStrain [5]). Our analysis concluded that all these available platforms have several shortcomings, namely;

1) *Limited sensor support* - Most available sensor nodes are not sensor platforms but rather embedded devices with one or two low-power, short-range sensors, such as temperature, humidity, pressure, and seismic sensors. Moreover, these nodes are limited to performing environmental monitoring or motion detection (vehicles etc.) and couldn't be expanded further to include high-resolution videos (even at a low bit rate) deemed *very important* for UGS.
2) *Limited processing* - Local processing is paramount for power scavenging [9] but the existing sensor nodes have low-power processors(mostly microcontrollers like ATMEL or low-power ARM processors) that lack computing power to support high-speed, high-bandwidth sensors like video cameras, their processing and related communication overhead. Lack of computing also limits development of advanced signal processing applications as needed for advanced surveillance.
3) *Low-bandwidth and interoperability* - Motes generally rely on off-the-shelf radio units (like 802.15.4). These radios are very low-bandwidth, are unable to support high throughput video data and are incompatible with a variety of other military radios because of security concerns [10].

These limitations, along with the inflexibility of existing designs to be *easily* adapted for our requirements of advanced ground based surveillance, necessitate the creation of a new sensor node platform. We assessed that our sensor node platform should be truly modular, computationally powerful, highly interoperable, and possess low SWAP features. Furthermore, we assessed that our sensor node platform should be a *product platform* that could easily be customized to develop products for other applications, including agricultural monitoring, infrastructure monitoring, and healthcare.

III. Design of ScoutNode Hardware and Software

The ScoutNode platform and accompanying software infrastructure for sensing, processing, communicating, control and visualization has been implemented using a combination of proprietary and off-the-shelf hardware. Since, this platform is highly modular it can be housed in a variety of form factors (spheres, cylinders, box etc.). Few of the nodes that we are developing exclusively for military applications are as shown in Figure 1. These ruggadized packages vary from 5.5 inches (rightmost) to 3.5 inches in diameter(leftmost), and weigh around 2.1 lb. They are all embedded with multiple sensors (listed in Table 1), 802.11b/g (2.4GHz) radio and provide information to a remote laptop or a hand-held running the application software called *SmartObserver*. Each node is powered by rechargable batteries that can power these nodes for 4 hours (continuous operation with full video support). The node has several power modes (full, idle, and sleep) providing it a lifetime of over 5 days on a single charge.

In the subsequent subsections we will discuss the design details of the various components of this platform.

A. Hardware Architecture

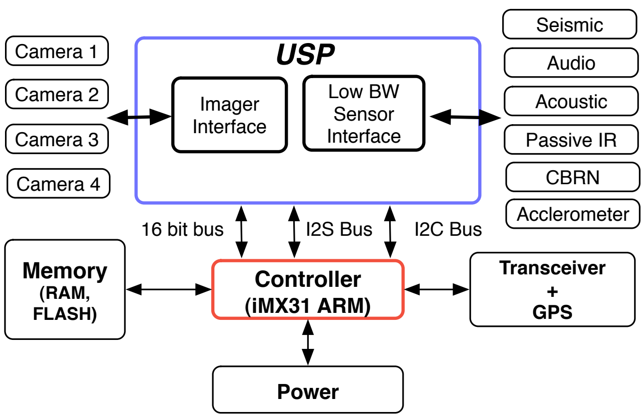


Fig. 2: The hardware architecture of ScoutNode is modular. Sensor module is decoupled from communication module and both modules can be customized for particular applications

The hardware architecture of the ScoutNode is divided into three physically separated modules powered by a single power source; **sensor module** or the *Universal Sensor Port (USP)* , **controller module**, and the **communication module**. To facilitate efficient packaging all these three modules are fabricated on a single board measuring about 3× 3 in. This board supports two different data input interfaces - LVDS (low voltage differential signaling) for cameras and the I2S, I2C interface for waveform and discrete sensors. Output interface is USB 2.0, Ethernet, both can be used to support radio modules. The board has an on-board RAM (1024 MB), and static memory (8 GB flash).

The sensor module or the *Universal Sensor Port (USP)* of the ScoutNode vastly differentiates it from other existing sensor nodes. Designed using a novel architecture it provide a modular interface and control infrastructure to integrate and manage multiple sensors (for supported sensor see Table 1) of varying speed and data bandwidth all through a single data bus. Internally, this module is divided into two individual interfaces

- *HBW* - High speed, and high data bandwidth like cameras are connected through LVDS with a data throughput of 27 Mbps. The interface I/O (data transfer from the sensors in to the RAM and flash), and control software is implemented in a low-power FPGA (Spartan 3 from Xilinx [11]). This FPGA currently contains application software needed for high-performance video processing, such as, compression, low-level computer vision and image processing. This FPGA module is interfaced with the control module via 16 bit bus and an I2C control bus. This I2C control bus (shown in Figure 2) provides the developer capability to control various settings on the I/O interface, such as, frame rate, image size, compression quality, output from low-level image processing etc.
- *LBW* - Low bandwidth sensors (few KHz) such as audio, acoustic, magnetometers as well as discrete sensors such as CBRN are supported through this interface. It contains two separate interfaces- I2S for audio and I2C for other waveform and discrete sensors. The I/O and control software is implemented using its own dedicated low-power processor and it contains low-level software to perform local noise filtering. The data received from this interface is processed directly by the control module

The control module of the ScoutNode platform is a low-power ARM(iMX31, 532MHz) processor that runs an embedded Linux distribution (2.6.22 kernel). The ARM has a JTAG interface for debugging and application development. The control module currently has three functions; data I/O from the USP, data processing, control communication and power modules. This module is very similar to the control module of a traditional sensor node. However, the software architecture of the controller is such that it provides the developer a rich set of APIs (C++ classes) that makes the sensor node completely programmable ranging from allowing individual sensor control, choice of communication module, protocol to control the overall power budget of the node (via different power modes).

The communication module of the platform is very flexible as the network abstraction layer in the control module supports various USB and IP based radios. We have developed our own drivers for some of the radio modules enabling us to integrate a variety of radios from several vendors (802.11b/g/n, 802.16 WiMax[12], 802.15.4). The communication module currently supports the open-source 802.11s [13] mesh networking functionality, which we continue to develop further to support advanced applications.

B. Power

ScoutNode's peak power requirement is a little over 2W. This is seen while continuous transmission of data (minimum of 3 Mbps) using peak radio power output of 30mW. This power budget renders the lifetime of the node to be 4 hours (using 2600mAH li-poly battery pack). While designing the node we realized short lifetime will be a limitation for a sensor node that is expected to survive several days of unattended operation. To overcome this shortcoming we developed a power management module that allows the application developer to program the node in three different modes, while maintaining useful autonomous sensing operation.

1) *full* - All the modules (sensor, processing and communication) are powered and operating at peak performance.

TABLE I: Various components of ScoutNode

Components	Specifications
Cameras	752x480 resolution, color, FOV(120^0 H,85^0V)
IR	940 or 850 nm LEDs (variable power upto 1W output)
Audio	300-3KHz, 12 dB compression
Acoustic	ultrasonic sensors with 15ft range (47^0 cone)
PIR	3000-1000 nm IR LEDs with 30ft range (90^0 cone)
RF sensor	900MHz-4.8 GHz power detector
GPS	-159dB, 20 channel receiver
Acclerometer	3-axis (2G)
Radiation	Gamma radiation sensor (triggers at 20 mRad)
Chemical	VOX vapor detector
Magnetometer	2-axis (± 50 μTesla)
Seismic	3-axis (100 mG)
Camera Processor	Proprietary (FPGA based designed by MobileFusion)
Controller	i.MX31 ARM 532MHz
Radio	802.11b/g (30 mW)
Batteries	Li-poly (2600 mAH)

TABLE II: Power Budget

Processes	Power
Radio send	2W
Radio receive	1.2 W
Radio sleep	700mW
Processing (full mode)	2 W
Processing (idle mode)	100mW
Processing (sleep mode)	40mW

2) *idle* - Low-bandwidth sensors are fully powered, while the processor is working at a low rate to process these sensors. The communication module is also working in low-power mode in order to broadcast the alerts to the clients at a low bit rate (8Kbps)
3) *sleep* - Sensors such as PIR and audio are being processed to generate alerts for broadcast and the radio module is completely powered down. This mode has control signal to wakeup the radio to go to idle mode and then power up to full power mode to transmit video and other recorded information.

In our testing we have validated the efficiency of these modes. In idle mode we can conserve power upto 50 hours, with 45 % of charge still remaining for further operation. In complete sleep mode the node can last 5 days.

Table 2 highlight these various modes for various modules on the node. As one would expect the major power consumption in our sensor node is the camera processing and the radio. While, one can power down the various video computing modules within the FPGA, bringing down the transmitted RF power below 30mW is not practical because it severely impacts the transmission range of the node.

C. Software Architecture

Each ScoutNodes node is capable of processing the multiple data streams that it acquires and transforming them into key pieces of metadata that can then be shared between nodes

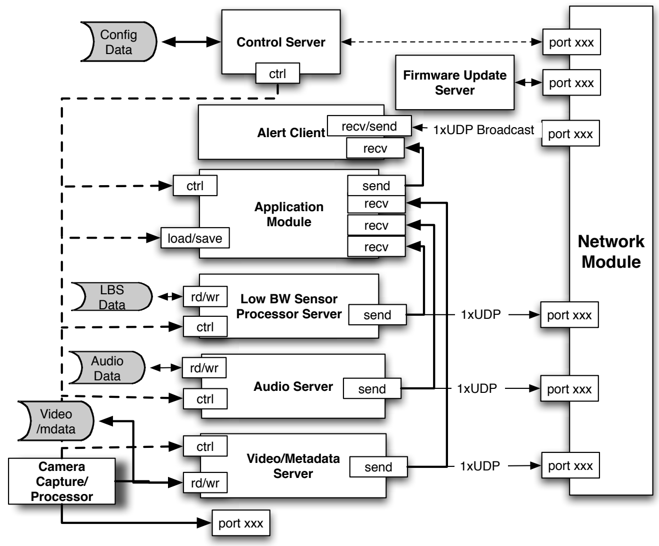


Fig. 3: The ScoutNode software manages data I/O, processing and communication using a client-server architecture. Each of the server communicates with the network module via UDP. The data from each server can be transmitted as separate packet to the client/receiver

without overloading communication bandwidth or computational resources at any one node. The approach also affords the additional benefit of decentralized awareness of a scene. Because there is no central client doing all the processing, a network of nodes can monitor a scene even when no client is present. When a client does eventually connect to the network, it can download a summary of what happened in the metadata that was generated from each device.

To enhance performance and preserve flexibility the ScoutNode processing module software architecture has been developed using a client-server philosophy. The architecture has three key modules; **Control Server**, **Firmware Update Server**, and the **Alert Server**.

The Control Server controls the data I/O through three different data servers- Video Server (from the FPGA), Audio Server (I2S), and the Low Bandwidth Sensor Server (I2C). It controls the data processing via application module. The Application Module is where the raw data from all sensors is processed by signal processing and machine learning based classifiers (that have been trained to look for various objects of interest (OOI), such as humans or vehicles, on each data stream). This module converts data in to alerts, metadata and other relevant information. These alerts in turn are used to trigger the recording of sensor data that is recorded on-board the node via data servers directly onto the flash memory.The Application Module has several API that can be used by developers to design their own classifiers and signal processing applications for autonomous sensing.

The Control Server also serves as dedicated communications channel between a ScoutNode and the client that is connected to it. Via this module, the client can control sensor settings such as video framerate, camera exposure settings, audio enable/disable, etc. The Control Server also controls the power management unit of the node and provides the developer the flexibility to manage the power budget.

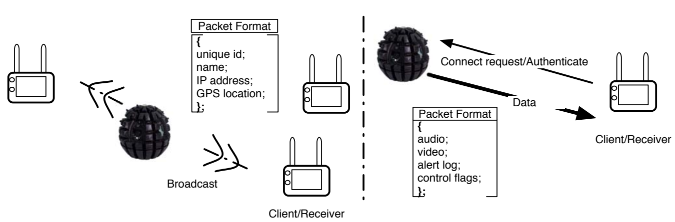
The Alert/Metadata Server is the conduit by which a ScoutNode can send messages to a shared broadcast channel. Both alerts and metadata computed from processed sensor data may be of interest to not only multiple clients, but possibly other ScoutNodes as well.

This whole architecture has been implemented using C/C++ and runs on the Linux 2.6.22 kernel. Each module has its own API (C++ classes) that can be used for further application development.

D. Communication Protocol

The network abstraction layer of the controller allows several radio modules to be interfaced to the node. However, to facilitate efficient power management and optimize use of channel bandwidth each ScoutNode follows a two step communication protocol (shown in figure 4). At every time instance when it generates an alert by on-board sensor processing (PIR, audio, CRBN sensor) it broadcasts a UDP packet of a size(10 Bytes) containing the id, name of the device, IP address and the associated GPS location. Nearby clients/receiver (or another ScoutNode) can receive this broadcast.

In the second step the nearby client or another ScoutNode sends a connect request and also an authentication request (with a unique encrypted key). This request also contains the necessary control flags to the ScoutNode to only obtain specific data from the node. Depending on the control flags the ScoutNode sends data and alert information using a combination of UDP and TCP packets. The UDP packet contains video, audio, and alert log associated with the past event, while TCP packet carries only the alert log.



(a) (b)

Fig. 4: Two step communication protocol is as shown. Figure (a) shows how the broadcasting nodes communicate with other nodes as well as the receivers. Figure (b) shows the send/request between the node and the client receiver. Multiple clients (upto 4) can connect to the same node.

Currently, the radio module on-board ScoutNode has been tested for range and expected QOS, especially for video data. Our field experiments performed in urban as well as denied areas with LOS maintained between the nodes and the receiver/client equipped with a 12dBi transceiver show significant data throughput (see Table 3) over traditional 802.11g wifi radios used for wireless cameras.

TABLE III: QOS with range

Video Rate (Hz)	Distance(m)
10	310
5	405
3	507
1	615

We expect these results to improve further as we move towards introducing gateway nodes for repeaters and also using powerful radios (especially for our military customers).

IV. Application Software

The ScoutNode communicates with the base location using a stand-alone application software called SmartObserver. This software offers several functionalities; monitoring (live, alert mode); sensor data display, sensor/node control, event data download, playback, map registration, node configuration, and alert configuration.

On a higher level SmartObserver software has two modes:

1) **Network Map View** - This view shows the geo registered deployment map of each ScoutNode(shown in Figure 5). This is the primary view of the client software and allows the user to get a bird-eye-view of the overall wide area network. On this deployment map each node is color coded to show the status (red indicates node is active and has a threat in its field-of-view), along with the network id. The alert log from each node is also presented to the user alongside the map view.
2) **Device View** - Individual sensor data, alerts, and metadata are made available in a form that is easy to view and navigate. For example, the user can view the video feed from any camera, or the user can scroll through a list of generated alerts. The individual settings for each node, including such parameters as the name and various sensor settings, can also be configured. The settings for one device, in fact, can be batch uploaded to multiple ScoutNodes. Similarly, settings for alerts which include timing and threshold parameters can also be configured at the client and uploaded to ScoutNodes. This view also allows to retrieve the recorded and stored data from the node and be downloaded to the client for later offline playback.

The application software is platform independent, touchscreen friendly and has been developed in C++ and uses Qt for the GUI front end. Behind this GUI front end it has a rich set of API for implementing a variety of network management, data management, visualization, sensor control features.

During our field trials with the US Army we performed several experiments using the application software, where sensor nodes were setup to perform autonomous wide area sensing in denied areas. In a subsequent research article we will report these experimental results.

Fig. 5: Single snapshot of the SmartObserver. Deployed nodes are shown as dots on a GIS map and can be accessed by the user by clicking on them. Each node changes color (red) when there is movement in its FOV. The user can click on the active node to watch the recorded as well as live video feed from that node.

V. Future Work

ScoutNode hardware platform discussed in this paper is stable and we expect it to support our product roadmap that includes applications in ground surveillance, remote monitoring, agriculture, and healthcare. The software architecture of ScoutNode platform continues to evolve to support functionalities both for application development as well as communication interoperability. In particular, our current focus is to enhance these two specific capabilities:

- *Communication* - ScoutNode is an open platform that can support several radios. However, to effectively communicate in truly interoperable radio networks we will need to further develop robust, power efficient mesh networking capabilities. We have been evaluating several proprietary Software Defined Radios [14] that comply with military JTRS [6]. These radios possess RF end that can operate in the band of 900MHz -5.6 GHz and have onboard NSA certified security features. The control software in these radios provides features to control bandwidth providing and optimal trade-off between QOS, while maintaining optimal long range (minimum of 5 miles), necessary for surveillance applications.
- *Sensor Processing* - We are currently focused on developing a embedded software toolkit that will include API for a vast number of computer vision and machine learning algorithms. We see this as a big step towards making future sensor nodes smart by providing expandable computing infrastructure for developers.

Apart from developing software we continue to work closely with our military and commercial users to develop rugged, all weather packages that provides functionalities of ScoutNode platform in user friendly form factors.

VI. Summary

We introduced a novel sensor node platform that provides powerful multimodal sensing, processing, and application development capabilities to application developers and researchers alike. We described the hardware, software design details along with necessary performance parameters related to power, and range. We also discussed future work that we continue in developing this platform further.

Acknowledgment

We will like to thank our sponsors, and other members of MobileFusion team for support. We will also like to thank our evaluation sponsors - US Army Electronics Proving Ground for supporting operation, performance, environmental testing of our devices and also for providing invaluable feedback on usability, and operational utility. We also thank the support from members from our corporate partner Lockheed Martin -Marine Sensors and Systems (MS2) for providing equipment and invaluable technical insights.

References

[1] B. Warneke, M. Last, B. Liebowitz, and K. S. J. Pister, "Smart dust: Communicating with a cubic-millimeter computer," *Computer*, vol. 34, no. 1, pp. 44–51, 2001.
[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks*, vol. 38, pp. 393–422, 2002.
[3] xbow, "Crossbow wireless sensor network product family," *http://www.xbow.com/Products/wproductsoverview.aspx/*.
[4] Archrock, "Archrock wireless sensor network product family," *http://www.archrock.com/products/*.
[5] Microstain, "Microstrain wireless sensor network product family," *http://www.microstrain.com/wireless-sensors.aspx*.
[6] DoD, "Joint tactical radio system," *http://jpeojtrs.mil/*.
[7] ——, "Army future combat systems, unattended ground sensor systems," *https://www.fcs.army.mil/systems/ugs/*.
[8] C. Basaran, "Research integration: Platform survey; critical evaluation of platforms commonly used in embedded wisents research," Technical University of Berlin, Tech. Rep., 2006.
[9] W. K. G.J Pottie, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 551–558, 2000.
[10] N. Sastry and D. Wagner, "Security Considerations for IEEE 802.15.4 Networks," in *ACM WiSe*, 2004.
[11] Xilinx, "Xilinx products," *www.xilinx.com/products/spartan-3*.
[12] Wimax, "Wimax forum," *http://www.wimaxforum.org/*.
[13] O. Workgroup, "802.11s," *http://www.open80211s.org/*.
[14] Harris, "Software defined radios," *http://www.govcomm.harris.com/SDR/*.