1-1-2010

RecoNode: A Reconfigurable Node for Heterogeneous Multi-Robot Search and Rescue

Richard M. Voyles
University of Denver

Sam Povilus
University of Denver

Rahul Mangharam
University of Pennsylvania, rahulm@seas.upenn.edu

Kang Li
University of Denver

Follow this and additional works at: https://repository.upenn.edu/mlab_papers

Recommended Citation

Richard M. Voyles, Sam Povilus, Rahul Mangharam, and Kang Li, "RecoNode: A Reconfigurable Node for Heterogeneous Multi-Robot Search and Rescue", . January 2010.

Suggested Citation:

Voyles, R.M.; Povilus, S.; Mangharam, R.; Kang Li; , "RecoNode: A reconfigurable node for heterogeneous multi-robot search and rescue," Safety Security and Rescue Robotics (SSRR), 2010 IEEE International Workshop on , vol., no., pp.1-7, 26-30 July 2010
doi: 10.1109/SSRR.2010.5981569

©2010 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

This paper is posted at ScholarlyCommons. https://repository.upenn.edu/mlab_papers/32
For more information, please contact repository@pobox.upenn.edu.

RecoNode: A Reconfigurable Node for Heterogeneous Multi-Robot Search and Rescue

Abstract

Search and rescue robots can benefit from small size as it facilitates access to voids and movement in cramped quarters. Yet, small robots cannot be the entire answer for urban search and rescue because small size limits the size of actuators, sensor payloads, computational capacity and battery capacity. Nonetheless, we are attempting to alleviate these limitations by developing the hardware and software infrastructure for heterogeneous, wireless sensor/actuator/control networks that is well-suited to miniature search and rescue robots, as well as a host of other relevant applications. These networks of application-specific sensors, actuators and intelligence will be assembled from a backbone that includes scalable computing, a flexible I/O bus, and multi-hop data networking. But two things will ultimately give our wireless infrastructure its novelty: a dual-baseband communications layer and the embedded virtual machine. The dual-baseband communications layer augments the standard data communication layer with a secondary, sub-millisecond synchronization layer to permit high-fidelity, deterministic, distributed control across the network. The determinism of this dual-baseband communications layer, in turn, enables the creation of the embedded virtual machine, which is a programming construct that abstracts away the physical sensor/actuator/control nodes. With this infrastructure, programming will not be done at the node level, as in conventional wireless sensor networks. Instead, programming will be done at the task level with port-based objects distributed across physical resources as necessary at either compile-time or run-time. At compile-time, the system can assist in the specification of the physical network, while at run-time the system can react to changes in configuration, such as nodes exhausting their batteries or losing connectivity. This paper describes progress to-date on developing this scalable infrastructure, specifically the RecoNode high-performance, dynamically-reconfigurable computational node for the Terminator- Bot crawling robot and the FireFly mid-performance node, as well as their real-time software.

Disciplines

Engineering

Comments

Suggested Citation:

Voyles, R.M.; Povilus, S.; Mangharam, R.; Kang Li; , "RecoNode: A reconfigurable node for heterogeneous multi-robot search and rescue," Safety Security and Rescue Robotics (SSRR), 2010 IEEE International Workshop on , vol., no., pp.1-7, 26-30 July 2010

doi: 10.1109/SSRR.2010.5981569

©2010 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

RecoNode: A Reconfigurable Node for Heterogeneous Multi-Robot Search and Rescue

Richard M. Voyles

Electrical and Computer
Engineering Department,
University of Denver
Email: rvoyles@du.edu

Sam Povilus

Electrical and Computer
Engineering Department,
University of Denver
Email: spovilus@du.edu

Rahul Mangharam

Electrical and Systems
Engineering Department,
University of Pennsylvania
Email: rahulm@seas.upenn.edu

Kang Li

Electrical and Computer
Engineering Department,
University of Denver
Email: kang.li@du.edu

Abstract — Search and rescue robots can benefit from small size as it facilitates access to voids and movement in cramped quarters. Yet, small robots cannot be the entire answer for urban search and rescue because small size limits the size of actuators, sensor payloads, computational capacity and battery capacity. Nonetheless, we are attempting to alleviate these limitations by developing the hardware and software infrastructure for heterogeneous, wireless sensor/actuator/control networks that is well-suited to miniature search and rescue robots, as well as a host of other relevant applications. These networks of application-specific sensors, actuators and intelligence will be assembled from a backbone that includes scalable computing, a flexible I/O bus, and multi-hop data networking. But two things will ultimately give our wireless infrastructure its novelty: a dual-baseband communications layer and the *embedded virtual machine*. The dual-baseband communications layer augments the standard data communication layer with a secondary, sub-millisecond synchronization layer to permit high-fidelity, deterministic, distributed control across the network. The determinism of this dual-baseband communications layer, in turn, enables the creation of the embedded virtual machine, which is a programming construct that abstracts away the physical sensor/actuator/control nodes. With this infrastructure, programming will not be done at the node level, as in conventional wireless sensor networks. Instead, programming will be done at the task level with *port-based objects* distributed across physical resources as necessary at either compile-time or run-time. At compile-time, the system can assist in the specification of the physical network, while at run-time the system can react to changes in configuration, such as nodes exhausting their batteries or losing connectivity. This paper describes progress to-date on developing this scalable infrastructure, specifically the RecoNode high-performance, dynamically-reconfigurable computational node for the Terminator-Bot crawling robot and the FireFly mid-performance node, as well as their real-time software.

I. INTRODUCTION

The field of search and rescue robotics, while growing rapidly in this decade, is still relatively new. Fieldable ground robots for urban search and rescue, while growing in sophistication, are largely confined to locomotion *on* the rubble - almost exclusively categorized in the man-packable class and deployed singly with multiple operators in teleoperation mode [1], [2], [3], [4], [5], [6]. Due to the pressure of limited time to find survivors, search and rescue is an obvious domain that would benefit from multi-robot teams, forming networks of sensors and actuators delving *into* the rubble in the search for survivors where they are likely to be trapped. But small size brings limitations in actuators, sensors, computational power

Fig. 1. A tethered CRAWLER emerging from under a slab (approx. 18 cm headroom) at Disaster City.

and battery life. These limitations based on size force new ways of accomplishing complex tasks. Instead of having one robot or even one type of robot, we can have multiple robots of heterogeneous abilities that work together. This allows the requirements for actuators, sensors and computational power to be split up amongst multiple robots, if the collaboration infrastructure is suitably developed.

Many years ago, we developed the first prototype of CRAWLER (a.k.a. TerminatorBot) [7] as a small scale search and rescue robot for core-bored search [8]. Since that time, various extensions have evolved to augment capabilities [9], [10], [11], but CRAWLER has remained tethered, as shown in Figure 1.

Tethers are desirable for a variety of reasons. They provide an infinite source of power that doesn't add weight or consume volume in the robot's body. Physical conductors also provide noise-free transmission of video and other sensor signals. They also provide a fail-safe method of retrieving the robot, which is

an important characteristic to emergency responders. Finally, tethers can be used to deliver necessities such as fresh water to survivors or even provide motive force [12], [13]. However, the goal of this work is to remove the tether because tethers effectively preclude multi-robot teaming.

For small robots entering small voids, there may not be space in the entry way for multiple tethers. In the case of core-bored search, emergency responders want to bore the smallest hole possible to speed the process. In the case of CRAWLER, it is sized just a bit smaller than a 3-inch core boring tool and even a second tethered robot can not squeeze through once the first tether is in the bore hole. Therefore, the RecoNode has been designed to provide multi-hop wireless networking for communication to the base station, high computational performance to permit on-board vision processing and self-adaptation, and low power consumption to reduce battery weight. The combination of these characteristics allows us to cut the tether and eventually build large, scalable teams of collaborating robots for search.

II. PREVIOUS WORK

The current generation of wireless sensor networks (WSN) has largely focused on open-loop sensing and monitoring applications [14], [15], [16], [17]. Examples include monitoring the temperature of bodies of water in a matrix of sensing sites [17] or monitoring ground vibrations to detect passing vehicles along a network of roads or service-ways [15]. Both of these “typical” applications involve identical nodes scattered throughout the network, minimal processing at each node, and relatively low-bandwidth communication between nodes. As a result, the current WSN platforms and systems software is predominantly event-based, asynchronous and with limited network-wide reliability. Furthermore, little research has been devoted to heterogeneous wireless sensor networks that integrate nodes across a range of computational and communication capabilities. So while platforms exist for low-rate sensors and applications for which time-stamping is sufficient, the research community faces the recurring burden of reinventing the wheel when such networks are integrated with higher-rate sensors (e.g. video surveillance), actuators with timeliness and safety constraints (e.g. real-time control), and networks requiring significant distributed in-network processing (e.g. video analytics and autonomous systems).

Yet, these high-power nodes should not necessarily proliferate throughout a homogeneous network the way current wireless sensor nodes proliferate. Because cost and power concerns dominate, these advanced capabilities demand heterogeneous networks, with a variety of nodes with different capabilities applied in a power-aware fashion. We plan to develop three classes of nodes: micro, meso and macro. The macro node is represented by RecoNode and the meso node is represented by FireFly, both of which are briefly described in this paper. The micro node is currently under development.

To-date, little infrastructure exists for research in the software and hardware aspects of heterogeneous sensor/actuator networks. Furthermore, since heterogeneity often requires

agility, autonomic and reconfigurable computing paradigms must exist within these networks to improve their deployability and usefulness in the field. This is a particular strength of RecoNode.

A. Wireless Sensor-Actuator-Controller networks

Embedded Wireless Sensor-Actuator-Controller (WSAC) networks are emerging as a practical means to monitor and operate automation systems with lower setup/maintenance costs [18], [19]. While these early attempts are still based on time stamping and very low rate applications, the physical benefits of wireless control, in terms of cable routing and replacement, are apparent and automation manufacturers and plant owners have increasing interest in the logical benefits, as well. With multi-hop WSAC networks, it is possible to build modular systems that can be swapped out for off-line maintenance during faults or can even manage themselves. The benefits to multi-robot teams – particularly resource-constrained teams – is the ability to separate sensing, control algorithms, and actuators onto different nodes and still maintain real-time deadlines.

One of the unique aspects of our network infrastructure that enables these new cyber-physical capabilities is the timeliness and determinism afforded by the system. This determinism is enabled, in turn, by a specific feature of the hardware: a secondary RF network for global synchronization. Another unique aspect for wireless networks is the virtual machine concept, which is also enabled by hardware functionality. Cross-compilation across heterogeneous compute platforms requires memory and computational resources most wireless sensors networks do not possess. Finally, autonomic capabilities are enabled through dynamically re-programmable hardware for self-adaptation.

B. Embedded Virtual Machine

Several efforts have been directed at virtual machines in wireless sensor networks. Mate [20], Scylla [21] and SwissQM [22] are virtual machines while SOS [23] and Contiki [24] provide runtime programming frameworks. In all these cases, interaction is assumed to be between an end-user and a single isolated node in a network and not among the nodes themselves. We intend to provide an inter-node Embedded Virtual Machine (EVM) that standardizes the programming interfaces and synchronizes connectivity between the heterogeneous categories of hardware platforms.

The primary design goal of the EVM is to provide the runtime mechanisms for control and actuation algorithms to adapt and reorganize resource utilization in the face of planned changes (e.g. shift and mode changes, plant output targets) and unplanned changes (e.g. link and node outages, topology changes). This capability allows for reliable wireless network control and is achieved by decoupling the functionality (i.e. controller tasks) from the underlying physical substrate (i.e. physical node topology) so communication, computation and coordination may be maintained across physical node boundaries. More specifically, the EVM is the systems software

around the embedded real-time operating system (RTOS) in each node, which facilitates the mechanisms to parametrically and programmatically control the operation of the node at runtime. It enables the control logic to dynamically assign tasks to controllers at runtime through task and network slot migration, partition and reallocation.

C. Dual baseband radio

The dual baseband radio is an important and unique part of the WSAC because it allows for determinism across the whole network with much finer time granularity than has been previously achieved. This means that real, or virtually real time is achievable allowing for better control of the system by both the operators and the control nodes of the network itself. We plan to use the same dual-band concept as that developed for FireFly [25], relying on NIST broadcast of timing signals on radio station WWV. Due to its time synchronization, network lifetime is predictable as is end to end delay for packets between the gateway and any node. Because there are two radios RT-Link is more scalable and robust than other single radio, software based methods.

The WSAC infrastructure, being in the early stages of development, will be used on a number of different platforms, and use a number of software systems that already have academic implementations.

D. CRAWLER (TerminatorBot)

The CRAWLER has two limbs and six degrees of freedom, so the computation requirements are greater than the average wheeled robot. The current processor for the CRAWLER is an 8-bit ATmega128 microcontroller similar to the FireFly wireless node and Crossbow motes. Running at about 16 MIPS with a power budget of roughly 4 mW/MIPS, the CPU can handle the basic control and trajectory generation functions, but not much more. Power is provided through the tether, so no batteries are onboard.

III. CURRENT WORK - RECONODE

The macro node, which is the most capable and complex node of our heterogeneous WSAC network, is the RecoNode, shown in Figure 2. The RecoNode is a multi-processor architecture based on the Virtex 4 FPGA with multiple, hard-core PowerPCs. Capable of up to 1600 MIPS from four PowerPCs plus hundreds of additional MIPS of special-purpose coprocessing from the FPGA fabric itself, this computational node is very high performance compared to conventional wireless sensor nodes - roughly 100 times greater computational throughput. The RecoNodes we have built are dual-processor models running at 400 MIPS with a power budget of 0.9 mW/MIPS.

The name *RecoNode* stems from the fact that both hardware and software are fully reconfigurable, both statically (offline) and dynamically (online). Software reconfigurability is facilitated by our PBO/RT operating system, while hardware reconfigurability is facilitated two ways: the Xilinx FPGA fabric, which permits dynamic partial reconfiguration and our own unique Morphing Bus for I/O.

Fig. 2. A RecoNode stack with Virtex-4 FX20 CPU and two single-width camera wedges.

Fig. 3. The conventional bus forces devices to conform to the signals defined by the bus.

A. Morphing bus

The RecoNode embodies the first full-scale implementation of the Morphing Bus, proposed in [26]. The Morphing Bus is a new concept in I/O buses that minimizes interface circuitry by adapting the bus signals to the sensor or actuator rather than vice-versa. Conventional I/O buses require interface circuitry on both sides of the bus (Figure 3). The Morphing Bus eliminates this need by *morphing* the bus signals to conform to the device (Figure 4).

In fact, this is a novel type of bus architecture, as shown in Figure 5. Buses are commonly characterized as space multiplexed (serial or parallel) and time multiplexed (one data

Fig. 4. The Morphing Bus forces the bus signals to conform to the device, simplifying the interface.

		Space Multiplexing	
		Multiplexed	Not Multiplexed
Time Multiplexing	Multiplexed	Serial Buses: USB CanBus RS485	Parallel Buses: ISA PCI VMEbus
	Not Multiplexed	Multi-Serial: PCIeexpress	Multi-Parallel:

Fig. 5. Modern bus architectures only span three of the possible four combinations of space/time multiplexing. The Morphing Bus adds the fourth as it is neither time nor space multiplexed.

element at a time or multiple data elements). While hand-built interfaces sometimes use non-time-multiplexed interfaces, no generalized bus we are aware of is non-multiplexed in both space and time.

Morphing bus “wedges” stack in a helix above the board that allows for component and connector space above and below the printed circuit board. The Morphing Bus specification also allows two sizes of wedges: single and double. The single wedge is a 60-degree sector of the hexagon shown in Figure 2, while the double wedge is a 120-degree sector. The RecoNode actually provides two Morphing Buses, which interleave in a double helix. This permits cooling air to be blown up through the helix, if necessary, and a central column of free-space is designed for cable routing.

The Morphing Bus uses the FPGA fabric to create the “converter” logic indicated in Figure 4. This presents a problem when actually compiling and “floor-planning” the FPGA firmware, particularly during partial dynamic reconfiguration of the FPGA. (i.e. Re-programming only part of the FPGA during runtime.) Because the converter is tied to a specific wedge, if the wedges are re-arranged, the signals to the converters have to be re-arranged. To achieve this, other researchers have dedicated entire segments of the FPGA to an internal bus [27].

Fig. 6. CAD model of the RecoNode installed in CRAWLER, minus motors, with numerous add on modules in a single Morphing Bus stack.

Fig. 7. CRAWLER with RecoNode installed.

In most cases, it has been 100% of the “long lines” of the FPGA [28]. By creating a symmetric *internal morphing bus* inside the FPGA for routing reconfigurable signals, we have been able to reduce this overhead to about 14% of the long lines, a significant savings of resources. (See [26] for more detail.)

B. Wireless interfaces

RecoNode, like all the other parts of the WSAC, has a data communication wireless interface. Because of its large size, reconfigurability and higher capacity batteries, RecoNode can carry multiple RF data interfaces. RecoNode’s RF module contains an RF Stack and a wedge with a RF transmitter chip. The stack will be implemented in VHDL on the FPGA. Having multiple wireless protocols supported will allow RecoNode to communicate not only within the WSAC but also use other protocols to talk to other systems, networks and robots. For wireless sensor network applications in urban search and rescue scenarios, low power consumption, low cost and being highly robust are the most important characteristics. In our design, we considered two protocol options: ZigBee and the Locally Selectable Protocol (LSP) [29], [30].

ZigBee is a new global standard for wireless communication, which provides a short-range cost effective networking capability. ZigBee technology is a low data rate, low power consumption, low cost, wireless networking protocol targeted towards automation and remote control applications [31][32]. One of the most popular ZigBee chips is the CC2520 from Texas Instruments. It is a single-chip 2.4 GHz IEEE 802.15.4 compliant RF transceiver that provides extensive hardware support for packet handling, data buffering, burst transmissions, data encryption, data authentication, clear channel assessment, link quality indication and packet timing information [33]. We have developed a ZigBee wedge to interface the RecoNode with other ZigBee enabled nodes, specifically the FireFly meso node and yet-to-be-developed micro node.

In addition to ZigBee, we have designed a wedge for Bluetooth and the Locally Selectable Protocol (LSP), which is a novel routing mechanism especially designed for high-bandwidth Wireless Video Sensor Networks (WVSN) in urban search and rescue (USAR) environment. It features 1) high-bandwidth, 2) low latency of data traffic, and 3) high-

Fig. 8. A block diagram of the motor control module.

quality lossless imaging. LSP is built over Bluetooth, which guarantees higher bandwidth and data rate for image and video transition. LSP is a novel hybrid routing scheme, which applies a reactive routing scheme to a local area to recover from path failures and a proactive scheme to a global area. This shows better throughput on a small and sparse WVSN [30].

C. Motor control module

We have developed a motor amplifier wedge for the Morphing Bus that allows the RecoNode to control multiple types of motors at up to 2.8A and 24 V. The two-channel, double-wedge DU120 motor amplifier takes PWM output signals from an FPGA-based waveform generator and routes quadrature incremental encoder signals to an FPGA-based encoder counter. Between these inputs and outputs are modular PD and PID control algorithms implemented both in software for the PowerPC and in hardware for the FPGA fabric. On top of the low-level controller, is a software-based trajectory generator. The high level diagram in figure 8 shows how it works as a closed-loop control system.

The motor module consists of 5 major components, the trajectory generator in the control area(the PPC), a PD or PID controller, an encoder counter, a PWM generator and the physical motor/power circuitry. The parts of major interest to us are the three modules implemented on the FPGA; the PD controller, the encoder counter and the PWM generator. The PD controller takes in the data from the trajectory generator and the encoder counter to determine where the motor should be and where it is. It uses this information to feed the PWM generator a pulse width. The PWM module was custom coded to create a modulation at any frequency desired, which allows the motor module to control many different types of motors.

To support self-adaptation, we have implemented a number of PID control algorithms in both hardware and software and have characterized them based on power consumption, speed, and consumption of FPGA logic. An initial comparative study was completed on the Spartan II FPGA with serial, parallel, and multichannel designs that were reported in [34]. We have ported the most useful of these designs – channel-level parallel

Fig. 9. The step response of the software-based PID module with the DU120 motor wedge.

and software-based – to the RecoNode for CRAWLER. The step response on the software-based design is shown in Figure 9.

D. Vision module

Because it is the highest performance node in our network, the RecoNode is designed to support onboard vision processing through tiny cellphone camera chips. The vision module allows any RecoNode based robot to see the world around it and navigate based on what it sees.

The main control logic block is responsible for the high level operation of the digital camera. The image capture module is responsible for reading captured frame data from the image sensor. In order to detect when it needs to capture data, the image grabber module looks for certain embedded codes. These embedded codes are placed within the stream of digital data by the camera. Once it has found an escape sequence; it knows to start or stop capturing data. In addition to data capture, the RecoNode camera module has the ability to control the CCD reader chip to enable dynamic control of settings like contrast.

Image processing is probably the most memory intensive operation robots do, because of this the RecoNode has 128MB of onboard RAM, allowing image data to be stored in the control computer or on the attached ram.

IV. FIREFLY

FireFly forms the current hardware platform that supports off data band global time synchronization [25]. Currently each FireFly node features an IEEE 802.15.4 [35] transceiver for data communication and pluggable time synchronization add-on modules. The time sync modules feature either a low power Amplitude Modulation (AM) radio receiver or a passive WWVB [36] atomic clock receiver. The AM receiver is only used indoors, in this case a transceiver is attached to the buildings power grid and translates the WWVB signal to AM and rides it on the buildings power lines using a 50μS line balanced square wave. Outside, the WWVB to AM translation

Fig. 10. The FireFly and AM time receiver.

is unnecessary and a WWVB receiver is attached directly to each node.

Real-time voice communication across multiple hops with timeliness guarantees, high throughput and predictable node lifetime are achieved through this dual-radio architecture. FireFly will be implemented in our WSAC in multiple flavors but will fall into two categories, nodes with and without sensors, nodes without sensors will simply be used to forward network traffic, while nodes with sensors will both forward network traffic and generate their own data.

V. REAL TIME OPERATING SYSTEM

In RecoNode, a Kernel or real time operating system (RTOS) is needed to organize access to the CPUs, memory, network interfaces, sensors and actuators. The initial work for the WSAC on RecoNode has been to port PBO/RT to the PowerPC 405 cores on the RecoNode. Our next step is to port Nano-RK to the RecoNode and then layer the port-based automata of PBO/RT on top of the extensive features of Nano-RK to combine the features of both for the macro, meso and micro architectures [37].

A. PBO/RT

PBO/RT is a library of routines that provide task scheduling and task dispatch functions on the ATmega128, ARM7, PSX, and PC embedded platforms. The PBO/RT library is modeled after the Chimera Port-Based Object interface for subsystems servers (SBS). Port-Based Objects is a coarse object oriented model for tasks in which each task consists of several standard methods that govern its real-time operation [38].

As discussed, our embedded virtual machine is not based on an intermediate byte code and runtime interpretation as in many virtual machine architectures. This generally too slow for the limited processing power of our meso and micro nodes. Instead, our heterogeneous team can employ RecoNodes to cross-compile reusable code modules to migrate code from one node to another. In fact, it is even possible to pre-compile various code modules for different architectures and download them at runtime.

To demonstrate this, we implemented a cross-platform downloader within PBO/RT to allow code fragments to be migrated by the RecoNode to our ATmega128-based CRAWLER

Fig. 11. Nano-RK architecture figure showing user applications and RT-Link.

microcontrollers (similar to FireFly). For this experiment, a simple port-based object was pre-compiled for the UM001 and loaded into the RecoNode memory. At runtime, the RecoNode transmitted the code module to the UM001 and it successfully instantiated it as a new runtime task.

B. Nano-RK

Different from the micro-kernel scheduler, PBO/RT, Nano-RK is a fully preemptive reservation-based power-aware real-time operating system specifically designed for sensor nodes with rich APIs which includes task management, signals and semaphores, general device drivers management functions. It includes a light-weight embedded resource kernel (RK) with timing support where it supports fixed-priority preemptive multitasking for ensuring that task deadlines are met. Another of Nano-RK's advantages over other RTOS is network management: Nano-RK has an architecture that supports easy installation of various wireless networking link layer protocols including RT-Link [39], for collision free Real-Time communication, and b-mac for low power contention based communication. To facilitate network programming, Nano-RK takes the approach of socket-like abstraction and generic system support for network scheduling and routing. Currently, Nano-RK runs on the FireFly Sensor Networking Platform as well as the MicaZ motes, both of the hardware platforms are ATmega128 based.

VI. CONCLUSION

We have developed the RecoNode reconfigurable multi-processor for heterogeneous wireless sensor-actuator-control networks and for installation in the miniature CRAWLER search and rescue robot. This new computational node enables new capabilities for the robot today and provides a gateway to enhanced multi-robot capabilities as we continue its development.

The RecoNode is a high performance computational node capable of about 400 MIPS, which is 25 times faster than the ATmega128 CPU used previously and over 4 times more power efficient. The RecoNode also contains the first, full-scale implementation of the Morphing Bus, which represents

an entirely new paradigm in I/O bus architecture. The Morphing Bus is neither time multiplexed nor space multiplexed, yet provides flexibility and generality in configuration. Finally, as a reconfigurable computing platform, the RecoNode allows partial dynamic reconfiguration and we have developed and implemented an efficient method for busing internal signals – based on the internal morphing bus – that is 6 times more resource efficient than previous methods.

As we continue to develop our heterogeneous infrastructure across micro, meso, and macro nodes, we will make these tools available to the research community. We believe the common facilities of the embedded virtual machine, dual-baseband radios for determinism, and port-based object encapsulation across all three node types will dramatically enhance multi-robot USAR teams and other forms of multi-capability research.

VII. ACKNOWLEDGEMENTS

This work was supported by National Science Foundation grants CNS-1042710 and IIS-0938196 with additional support from the NSF Safety, Security and Rescue Research Center.

REFERENCES

- [1] B. Shah and H. Choset, "Survey on urban search and rescue robots," *Robotics Society of Japan*, vol. 22, July 2004.
- [2] M. J. Micire, "Evolution and field performance of a rescue robot," *Journal of Field Robotics*, vol. 25, pp. 17–30, Jan/Feb 2008.
- [3] M. Arai, Y. Tanaka, S. Hirose, H. Kuwahara, and S. Tsukui, "Development of souryu-iv and souryu-v: Serially connected crawler vehicles for in-rubble searching operations," *Journal of Field Robotics*, vol. 25, pp. 31–65, Jan/Feb 2008.
- [4] K. Ohno, S. Tadokoro, K. Nagatani, E. Koyanagi, and T. Yoshida, "3-d mapping of an underground mall using a tracked vehicle with four sub-tracks," in *SSRR*, Nov 2009.
- [5] M. Guarnieri, I. Takao, E. F. Fukushima, and S. Hirose, "Helios viii: Toward practical robots for search and rescue operations," *Journal of Robotics and Mechatronics*, vol. 20, no. 5, pp. 675–694, 2008.
- [6] A. Jacoff, E. Messina, and J. Evans, "A standard test course for urban search and rescue robots," in *Proc. of the IEEE Performance Metrics for Intelligent Systems Workshop*, (Gaithersburg, MD), Aug. 2000.
- [7] R. M. Voyles, "Terminatorbot: A robot with dual-use arms for manipulation and locomotion," in *Proc. of the IEEE Int'l Conference on Robotics and Automation*, pp. 61–66, April 2000.
- [8] R. Voyles, A. Larson, M. Lapoint, and J. Bae, "Core-bored search-and-rescue applications for an agile limbed robot," in *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, pp. 58–63, 2004.
- [9] R. Voyles and A. Larson, "Terminatorbot: A novel robot with dual-use mechanism for locomotion and manipulation," *IEEE/ASME Transactions on Mechatronics*, vol. 10, no. 1, pp. 17–25, 2005.
- [10] R. Voyles, R. Godzdzank, and T. Kim, "Auxiliary motive power for terminatorbot: An actuator toolbox," in *Proc. of the 6th IEEE Workshop on Safety, Security and Rescue Robots*, p. T4, 2007.
- [11] R. Voyles and R. Godzdzank, "Side-slipping locomotion of a miniature, reconfigurable limb/tread hybrid robot," in *Proc. IEEE Int'l Workshop on Safety, Security and Rescue Robotics*, pp. 58–64, 2008.
- [12] X. Yang, R. Voyles, K. Li, and S. Povilus, "Experimental comparison of robotics locomotion with passive tether and active tether," in *Proc. IEEE Int'l Workshop on Safety, Security and Rescue Robotics*, 2009.
- [13] R. Feller, D. Perrin, and R. Howe, "Validation and explanation of waterhammer-based locomotion," in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Orlando, FL, USA), pp. 4264–4269, 2006.
- [14] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393–422, March 2002.
- [15] P. Vicaire, T. He, T. Yan, Q. Cao, G. Zhou, L. Gu, L. Luo, R. Stoleru, J. A. Stankovic, and T. Abdelzaher, "Achieving long-term surveillance in vigilnet," *ACM Trans. on Sensor Networks*, vol. 5, no. 1, pp. 1–39, 2009.
- [16] C. Sharp, S. Shaffert, A. Woo, N. Sastry, C. Karlof, S. Sastry, and D. Culler, "Design and implementation of a sensor network system for vehicle tracking and autonomous interception," in *Proceedings of the Second European Workshop of Wireless Sensor Networks*, 2005.
- [17] R. Szewczyk, J. Polastre, A. Mainwaring, and D. Culler, "An analysis of a large scale habitat monitoring application," in *Proc. of the Second ACM Conference on Embedded Networked Sensor Systems*, 2004.
- [18] F. Xia, Y.-C. Tian, Y. Li, and Y. Sun, "Wireless sensor/actuator network design for mobile control applications," *Sensors*, vol. 7, no. 10, pp. 2157–2173, 2007.
- [19] A. Pawlowski, J. L. Guzman, F. Rodriguez, M. Berenguel, J. Sanchez, and S. Dormido, "Simulation of greenhouse climate monitoring and control with wireless sensor network and event-based control," *Sensors*, vol. 9, no. 1, pp. 232–252, 2009.
- [20] P. Levis and D. Culler, "Mate: A tiny virtual machine for sensor networks," *Computer Architecture News*, vol. 30, no. 5, pp. 85–95, 2002.
- [21] P. Marbell and L. Iftode, "Scylla: A smart virtual machine for mobile embedded systems," in *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*, 2000.
- [22] R. Miller, G. Alonso, and D. Kossmann, "A virtual machine for sensor networks," in *Proceedings of the European Conference on Computer Systems*, pp. 145–158, 2007.
- [23] C.-C. Han, R. Kumar, R. Shea, E. Kohler, and M. Srivastava, "Sos : A dynamic operating system for sensor nodes," in *Proceedings of the International Conference on Mobile Systems, Applications, and Services*, pp. 163–176, 2005.
- [24] A. Dunkels, N. Finne, J. Eriksson, and T. Voigt, "Run-time dynamic linking for reprogramming wireless sensor networks," in *Proc. of the ACM Conf. on Embedded Networked Sensor Systems*, pp. 9–17, 2006.
- [25] R. Mangharam, A. Rowe, and R. Rajkumar, "Firefly: A cross-layer platform for wireless sensor networks," *Real-Time Systems Journal*, vol. 37, no. 3, pp. 183–231, 2007.
- [26] C. D'Souza, B. Kim, and R. Voyles, "Morphing bus: A rapid deployment computing architecture for high performance, resource-constrained robots," in *Proc. of the IEEE Int'l Conference on Robotics and Automation*, pp. 311–316, 2007.
- [27] H. Ito, K. Oguri, K. Nagami, R. Konishi, and T. Shiozawa, "The plastic cell architecture for dynamic reconfigurable computing," in *Proceedings of the Ninth International Workshop on Rapid System Prototyping*, 1998.
- [28] P. Sedcole, B. Blodget, T. Becker, J. Anderson, and P. Lysaght, "Modular dynamic reconfiguration in virtex fpgas," in *IEEE Proceedings on Computers and Digital Techniques*, May 2006.
- [29] J. Bae and R. Voyles, "Wireless video sensor networks over bluetooth for a team of urban search and rescue robots," in *Proceedings of the 2006 International Conference on Wireless Networks*, 2006.
- [30] R. Voyles, J. Bae, A. Larson, and M. Ayad, "Wireless video sensor networks for sparse, resource-constrained, multi-robot teams," *Intelligent Service Robotics*, vol. 2, pp. 235–246, October 2009.
- [31] S. Safaric and K. Malaric, "Zigbee wireless standard,"
- [32] S. C. Ergen, "Zigbee/ieee 802.15.4 summary," <http://www.eecs.berkeley.edu/csinem/academic/publications/zigbee.pdf>.
- [33] *CC2520 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver specifications*.
- [34] W. Zhao, B. H. Kim, A. C. Larson, and R. M. Voyles, "Fpga implementation of closed-loop control system for small-scale robot," in *Proc. of the 12th Int'l Conf. on Advanced Robotics*, pp. 70–77, July 2005.
- [35] IEEE, "802.15.4-2003 specifications for low rate wireless personal area networks part 15.4: Wireless mac and phy," 2003.
- [36] N. I. of Standards and Technology, "60khz standard time station, ft. collins, co,"
- [37] A. Eswaran, A. Rowe, and R. Rajkumar, "Nano-rk: An energy-aware resource-centric operating system for sensor networks," in *Proceedings of the IEEE Real-Time Systems Symposium*, December 2005.
- [38] D. B. Stewart, R. A. Volpe, and P. K. Khosla, "Design of dynamically reconfigurable real-time software using port-based objects," *IEEE Transactions on Software Engineering*, vol. 23, pp. 759–776, December 1997.
- [39] A. Rowe, R. Mangharam, and R. Rajkumar, "Rt-link: A time-synchronized link protocol for energy constrained multi-hop wireless networks," in *Proc. of the IEEE International Conf. on Sensors, Mesh and Ad Hoc Communications and Networks*, Sept 2006.