WARPWING: A complete open source control platform for miniature robots

Ankur M. Mehta, Kristofer S. J. Pister
Department of Electrical Engineering and Computer Sciences
UC Berkeley, Berkeley, CA, USA
{mehtank,pister}@eecs.berkeley.edu

***Abstract*—The electronics packages for many robot control systems have very similar requirements, yet are often redesigned for each custom application. To reduce wasted time and effort, the project presented in this paper (the Wireless Autonomous Robot Platform with Inertial Navigation and Guidance, WARPWING) is intended to create a complete and easily customizable general purpose control system for miniature robotic systems, in particular micro air vehicles. In its default configuration, hardware designs, firmware, and software are all available to deliver an out-of-the-box robot control solution comprising 6 degree-of-freedom inertial sensors, a microprocessor, and wireless communication, along with general purpose input/output pins, serial ports, and control outputs for interfacing to additional sensors and actuators. The entire project is open source and a process is in place to enable modification of any component, allowing for easy adaptation to any need. WARPWING is already in use in a number of labs, with each research group contributing its expertise to enhance the platform and make such modifications available to others as well.**

***Index Terms*—Mobile robots, Control systems, Wireless sensor networks, Inertial navigation**

I. Introduction

There is much recent work on miniature robotic systems and subsystems, and in the course of that research a considerable amount of effort is often duplicated. From sensors and devices through autonomous robots to swarm behaviors, there are a number of base requirements necessary to test, evaluate, and refine all levels of robotics research. In particular, processing and communication, specifically wireless communication, are necessary for any autonomous robotic endeavors. Typically, an inertial measurement unit (IMU) is also used to generate sensor data for feedback control.

Many research groups independently design their own hardware solutions, or adapt a handful of fixed commercial subsystems to their custom application. However, these approaches are often time-consuming and inefficient, causing valuable research effort to be spent recreating solutions, often by those inexperienced in hardware design.

The Wireless Autonomous Robot Platform with Inertial Navigation and Guidance, WARPWING, is designed to streamline robot development by collecting all the electronics requirements for robot control in one open source project, from hardware selection to board design to firmware and software programming. By providing a full solution in an open source format, researchers can focus solely on their areas of research, while still maintaining a cohesive environment in which to develop and test a complete system. As users adapt parts of WARPWING to their varied custom applications, their changes are also shared with the community, obviating the need for redundant independent development.

WARPWING is centered around the Guidance and Inertial Navigation Assistant, GINA, shown in figure 1. This is a low mass wireless inertial measurement unit (IMU) designed for use as a general purpose micro air vehicle (MAV) or robot controller. It comprises inertial sensors for angular rate and linear acceleration along with a general purpose feature-laden microprocessor. It can interface to additional sensors or drive a number of actuators (some of which are described below) via an expansion header, and provides built-in communication over a 2.4GHz wireless link.

The WARPWING project consists of hardware components – a device database, schematic design, and board layout – and software components – firmware for the onboard microprocessor and applications and modules for a basestation computer – along with relevant documentation. This entire project is made available on Sourceforge, a popular open source community site, at http://warpwing.sourceforge.net.

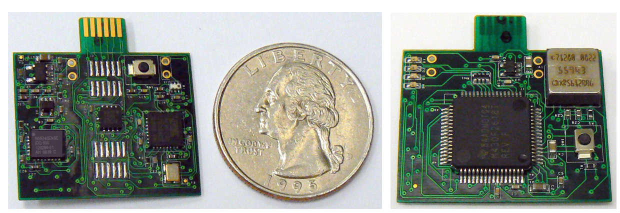

Fig. 1. GINA provides inertial sensing, processing, control, and wireless communication on a board smaller and lighter than a US quarter.

Section II describes some of the considerations driving the WARPWING project decisions and GINA design. The specific designs are presented in section III, detailing the current GINA hardware along with the firmware and software needed to use the GINA board. Section IV expounds on the open source nature of the WARPWING project, giving examples of modifications made to the default project by other research groups in their robotic applications. Finally, section V examines the case studies from the previous section to draw some conclusions about open source hardware for robot development, and presents a vision for how this project can evolve in the future.

II. Background

Initially created as a flight control system for a small inertially guided rocket (see figure 2), GINA was designed to be the smallest, lightest, and lowest power stability and guidance control solution with off-the-shelf components. It was then also used on a miniature coaxial helicopter for use as an autonomous MAV, as seen in figure 3. As development on both those projects continued in parallel, it became clear that the requirements for both, and indeed a large portion of autonomous air vehicle and small robot research in general, were highly similar.

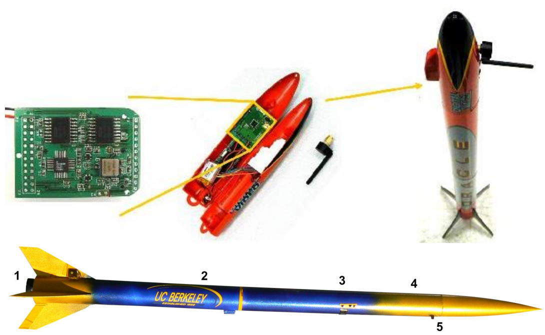

Fig. 2. The original rocket containing GINA sensors (above) has evolved to the rocket shown on the bottom, using the WARPWING platform for 3 axis inertial attitude feedback control.

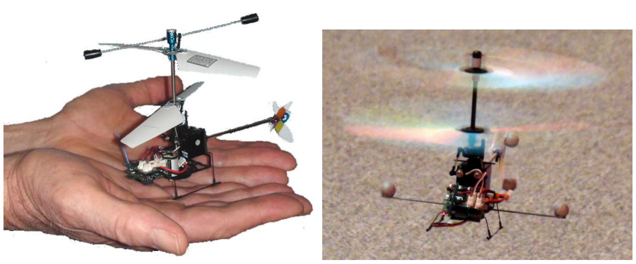

Fig. 3. GINA is mounted onto an off the shelf miniature coaxial helicopter, using a daughter board to connect to the actuators. On the right, this helicopter is shown using the WARPWING system to autonomously hover.

In practice, the overall size of a robotic system is often dictated by the weight it needs to carry, so minimizing overall size requires minimizing the total mass of the system. Lowering power draw is also beneficial, as it reduces the size of the battery necessary. The specifics of the GINA hardware are described in the following section; the primary design consideration in the development of the GINA electronics was to minimize weight while still providing the required functionality for a general autonomous robot.

The 6 cm^2 GINA board weighs 1.6 grams and runs off of a single 3.7V lithium polymer cell. It consumes a maximum of 30mA at 3V when fully powered; intelligent duty cycling can make use of low power modes to lower average power and increase lifetime. The system uses entirely commercial off-the-shelf components aside from a custom printed circuit board (PCB), and can be made in its entirety in one-off quantities for less than $200.

TABLE I
Noise in the inertial sense axes

Axis	Bandwidth	Measurement noise σ	Zero-bias error σ_{bias}
a_x	500 Hz	$.57m/s^2$	N/A
a_y	500 Hz	$.46m/s^2$	N/A
a_z	500 Hz	$.61m/s^2$	N/A
ω_p	140 Hz	$0.42°/s$	$0.07°/s$
ω_q	140 Hz	$0.26°/s$	$0.06°/s$
ω_r	140 Hz	$1.04°/s$	$0.04°/s$

III. WARPWING architecture

A. Hardware design

1) Inertial sensors: GINA has full 6-axis inertial sensing, composed of 3 MEMS sensors.

Angular yaw rate is measured by an Analog Devices ADXRS610 single axis analog gyro. The pitch and roll angular rates are measured by an Invensense IDG650 dual axis analog gyro. These gyros measure rates up to ±300°/s, and are digitized by an analog to digital converter (ADC) on the processor. Both of these devices have on-chip temperature sensors for use in calibrating out temperature coefficients.

There are two components to the noise in the gyro outputs. First, after subtracting a linear temperature-dependent offset, the zero-bias of the gyros varies across runs. Second, in a single run, the output of the gyro displays additive white gaussian noise in each measurement.

These noise terms are shown in figure 4 and summarized in table I.

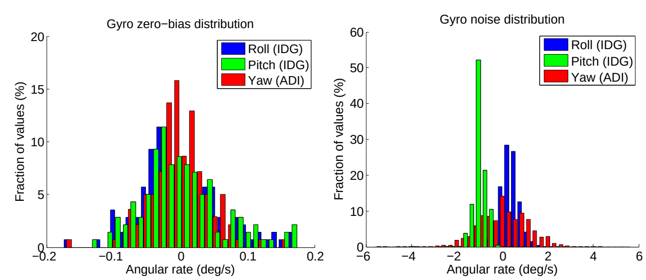

Fig. 4. The variation of the gyro zero-bias for a single board across 150 runs over 5 days is shown in the graph on the left for each of the three axes. The graph on the right shows the zero-rate gyro outputs over 2000 measurements during a single run. The bias offsets are clearly visible in the right graph, as well as a notable difference in performance between the two gyros.

Linear motion is detected using a Kionix KXSD9-1026 3 axis digital accelerometer, measuring a configurable full-scale range of ±2g, ±4g, ±6g, or ±8g with a 12 bit ADC at 400 Hz. It has a much lower temperature sensitivity and zero-bias noise, so can be assumed to have a single zero-mean additive white gaussian noise term added to each measurement across all runs. These values are summarized in table I.

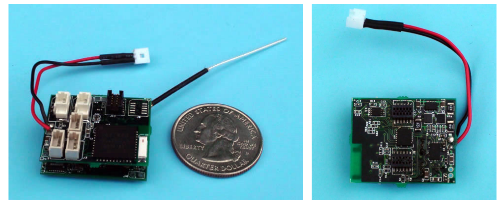

Fig. 5. A daughter board for the GINA mote brings PWMs out to connectors for servos, uses other PWMs for brushed motor controllers, and includes additional sensors such as a battery fuel gauge, GPS receiver, and vision processors.

It is important to note that though this sensor suite is comparable to or better than commercially available IMU options, these are merely the ones currently used on this board; there are now more advanced (e.g. lighter, lower noise, more tightly integrated) sensors on the market. Because the schematics are open source, it is a fairly straightforward task to replace the sensors with their newest counterparts; once done, it becomes available to anyone else who would like to use it. In fact, as will be described below, one such revision includes a more sensitive accelerometer to be used for vibration sensing instead of motion detection.

2) Wireless communication: Communication to the board is handled by an Atmel AT86RF231 wireless radio. This is an IEEE 802.15.4 compliant 2.4GHz RF transceiver capable of half-duplex communication at 250 kbps. It is fully interoperable with off the shelf 802.15.4 devices, so no special hardware is necessary to communicate with a GINA mote. It also has enhanced non-standard modes capable of communicating at higher data rates up to 2 Mbps.

GINA allows for a monopole wire antenna, though an on-board chip antenna can easily be substituted in. This allows for generally reliable communication within a room-sized area, although care may need to be taken to mitigate interference and multipath effects (see [1] for a possible solution). For long range communication, directional high-gain antennas can be used at the basestation and/or on the GINA mote itself; a +14dBi antenna at the basestation and a +8dBi antenna on the mote allowed reliable communication over a 1km long line-of-sight link. A collection of motes can be connected in a mesh network.

3) Processing: The on-board processing is handled by a Texas Instruments MSP430F2618. This is a highly capable 16 MHz 16 bit microprocessor. It has onboard analog to digital converters (ADCs) and an I2C serial interface to read the sensors, and an SPI module to interface to the radio. A hardware multiplier allows for 16 bit integer or fixed point single cycle multiplies. Additional on-chip functionality includes direct memory access, timers, and a well developed interrupt stack.

The processor also has a number of peripherals, accessible via expansion headers on the GINA board. In particular, in addition to general purpose input-output (GPIO) pins, there are two additional serial ports to read from and write to digital sensors and actuators, as well as a handful of PWM outputs to drive conventional robot actuators such as servos and motors.

4) Daughter boards: The expansion headers are intended to connect to an application-specific daughter board. A general daughter board was designed to break out commonly used signals to appropriate connectors, as shown in figure 5. Two independent sets of PWM outputs from the processor can be used to control up to 3 servos (or brushless motor controllers) and 2 motor drivers consisting of a mosfet and flyback diode.

Additional sensors were also included for evaluation purposes on this daughter board. A GPS receiver with an on-board chip antenna is connected to one of the serial ports, while a battery monitor / fuel gauge is connected to another. Connectors were also included to interface to Wiimote cameras, which provide IR blob detection for an off-the-shelf vision-based localization system.

Magnetometers, a common addition to IMUs, were intentionally left off of the main GINA board. Since the primary application of WARPWING is for small robotics, the board will most likely be placed next to DC motors, which generate a magnetic field far stronger than that of the earth, rendering magnetometers useless. However, if magnetometers were to be desired for a particular application, they can be added to a daughter board in much the same way as the other additional sensors.

5) Recent updates: The WARPWING platform is constantly evolving to take advantage of newer and better hardware, and since the original submission of this paper, a number of improvements have been made. The inertial sensors have been updated with a new three axis digital gyro (Invensense ITG-3200) replacing the previous gyros. Based on user requests, a three axis digital magnetometer (Honeywell HMC5843) has also been included, along with an additional high-sensitivity accelerometer (STMicro LIS344). The antenna solution now comprises a coax connector and an onboard chip antenna, together with automatic antenna selection for communication robustness.

B. GINA microprocessor firmware

The MSP430 microprocessor on the GINA board was programmed using embedded C and compiled in the IAR integrated design environment (IDE). The basic firmware was designed to be broadly applicable for most requirements. The processor starts up in a holding loop, listening for commands over the 15.4 radio. A number of commands and operating modes are available, and described in the API documentation available with the source code on Sourceforge.

The most relevant mode is the control loop. A hardware timer is used to run this loop every 3 ms. Every iteration, the inertial sensors are polled and the data is placed into a packet sent out over the 15.4 radio. Then the processor waits for commands, in particular setting the PWM duty cycles for actuators. This enables an off-board feedback controller operating on a basestation computer to receive and process data to generate actuator control signals, and pass that back to be implemented on the GINA with a short latency.

This loop provides a sensor update rate of 333 Hz, with about 48k instruction cycles available per loop. With the processor only handling interfacing with the sensors and radio, most of these cycles are spent idle; an autonomous system can use these to implement an application specific feedback control directly on-board.

C. Basestation software

The default behavior of the GINA firmware offloads the responsibility of actual robot control to a basestation computer, allowing for much easier design and development of robot control. A hardware device is required to allow the computer to speak to the GINA using the 802.15.4 wireless standard; any such transceiver will work, but slightly specialized firmware has been written for the Atmel RZ USBStick to simplify the basestation software.

The software must then interface to the RZ USBStick (appearing as a serial port to the computer); again, the full API is documented with the source code available on Sourceforge. A collection of utilities has been written in Python, forming an easy to read and modify cross platform skeleton upon which to build application-specific software. Default programs that are included in the WARPWING package allow real-time graphing of IMU outputs, data logging, and manned control using a 4 axis joystick to control two motor and two servo outputs on the GINA mote. More advanced routines, such as an extended Kalman filter (EKF) to generate an attitude estimate from the inertial rates (adapted from [2]) have also been developed.

As is common in the open source world, users sometimes write their own software and make it available to the community. Matlab and Labview routines have been implemented by groups using WARPWING, and can also used to interface to GINA motes.

D. Performance

The Microstrain 3DM-GX2 is one of the smallest commercially available IMUs. It weighs 16g without its enclosure, and consumes 90mA at 4.5V. It is very widely used in autopilots for unmanned aerial vehicles, and has an optional wireless-enabled model. The 3DM-GX2 costs \$1695, or \$3095 for the wireless model. From its website, the "3DM-GX2 is a high-performance gyro enhanced orientation sensor which utilizes miniature MEMS sensor technology. It combines a triaxial accelerometer, triaxial gyro, triaxial magnetometer, temperature sensors, and an on-board processor running a sophisticated sensor fusion algorithm." [3]

A GINA board was mounted to the 3DM-GX2 unit, and data was collected simultaneously from both devices. The scaled data is shown in figures 6 and 7 for a sample angular rate and linear acceleration trace. It is seen that GINA provides sensor outputs that are slightly noisier than the commercial device, but otherwise matches the output very closely. A 35 ms latency in the 3DM-GX2 is also visible from this data. Taking into account this lag, the sensor data from GINA has a $> 99.5\%$ correlation to the commercial IMU output.

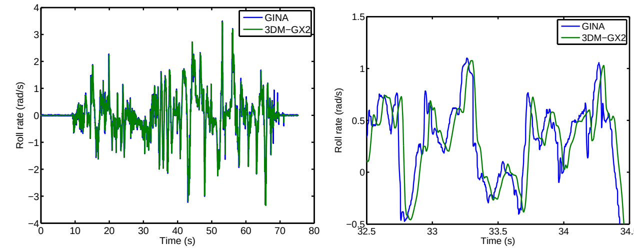

Fig. 6. Comparison of gyro output between GINA and 3DM-GX2

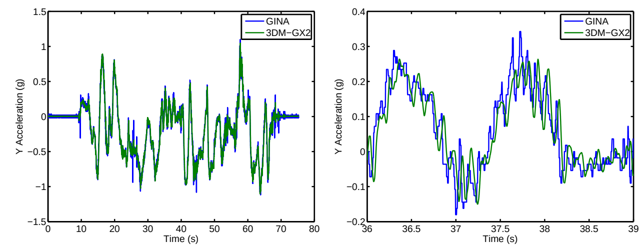

Fig. 7. Comparison of acceleration output between GINA and 3DM-GX2

The 3DM-GX2 also provides an attitude estimate which can be compared with the output of an EKF running in real time on GINA data. The two attitude angle estimates are shown in figures 8 and 9. It is again seen that the GINA estimates are a bit noisier, but this comes with a quicker response time to motion. This time constant is most visible at the end of the pitch trace. The attitude estimates between GINA and 3DM-GX2 have a $> 98\%$ correlation.

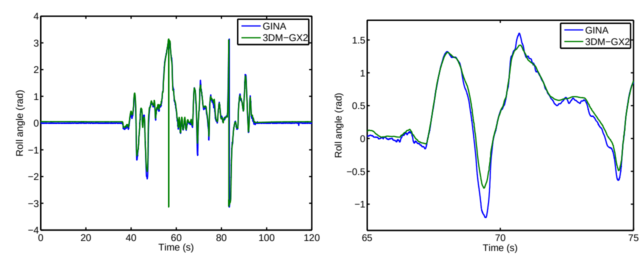

Fig. 8. Roll angle estimate comparison. GINA and 3DM-GX2 agree closely in their state estimates.

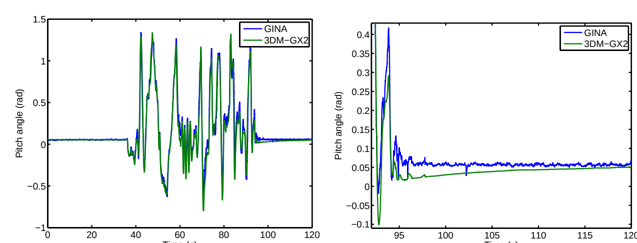

Fig. 9. Pitch angle estimate comparison. The noisier GINA estimate also allows a much shorter settling time than the 3DM-GX2.

IV. Modification and Sharing

The goal of the WARPWING project is not just to present an ultra-small full-featured wireless IMU and robot controller, but to use that to advance other robotic research as well. Any aspect of WARPWING can be modified to fit the specific needs of a design. Thus, even with little to no electronics ability, researchers can have a platform on which to base an inertial robot controller.

The default WARPWING distribution as described above allows researchers to mount the wireless GINA mote to a dynamic system and log its inertial rates on a basestation computer, all out of the box. The system was distributed to over a dozen research groups around the world, and all were able to collect data within days. Some examples are shown in figures 10, 11.

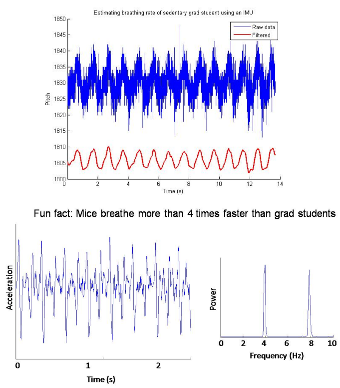

Fig. 10. Inertial data collected from GINA motes mounted on mammals is used to calculate sedentary breathing rates. Data courtesy Subramaniam Venkatraman.

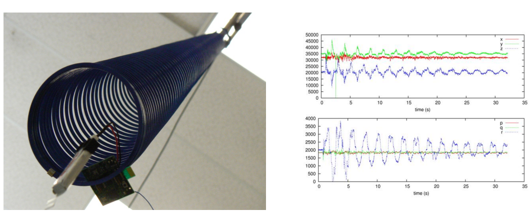

Fig. 11. Inertial data collected from a GINA mote mounted on a coiled spring, demonstrating both oscillatory rotation and translation. Data courtesy Thomas Watteyne.

When used for more than simply measuring inertial rates, however, the default WARPWING configuration is likely insufficient for controlling an arbitrary robot. But because the system is very general, typically only a few specifics need to be modified to build the desired controller. Because of the nature of the open source community, these changes are shared, simplifying similar changes in the future.

There are already projects using WARPWING that have modified some or all of the components of the entire system. A select few are summarized below as case studies for the ways WARPWING can be adapted to simplify robot development.

A. Software

In order to use WARPWING to control a specific robot, the software is almost certainly going to need to be rewritten. For a large portion of robots, however, that will be the only component that needs to be modified.

The Costello Research Group at the Georgia Institute of Technology have developed a hybrid air/ground vehicle called the hopping rotochute that uses rotors to generate the lift required to cause the robot to hop over obstacles and along trajectories [4], [5].

The default hardware and firmware configuration for the GINA mote is sufficient to control the actuators, and so with the control hardware taken care of by WARPWING, research effort can be dedicated to mechanical vehicle design and control algorithms for newer versions of the hopping rotochute, as shown in figure 12. The software only needed to be modified insofar as to interface the existing controller written in MATLAB to the WARPWING basestation transceiver.

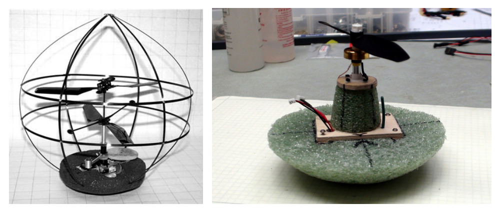

Fig. 12. The hopping rotochute developed at GATech uses WARPWING as its flight controller. The software interface was rewritten to fit in to the existing MATLAB framework, while the rest of the WARPWING system remained unchanged. This allowed for quick development of the mini rotochute, shown at right.

B. Firmware

For more complicated robots, with more advanced onboard behavior, the firmware on the GINA mote may also need to be modified. A mini quadrotor, shown in figure 13 has been developed by Daedalus Flight Systems and the University of Maryland. It uses four brushless DC motors for its actuation and requires an onboard attitude stability loop.

The WARPWING platform took care of the majority of the flight control system requirements, and the only elements that needed to be added in firmware were specific to the feedback control loop. Routines were written to derive setpoint values for the inertial rates, mix them with joystick inputs sent via radio, and wrap a loop around the measured rates to drive the PWM outputs controlling the motors. For this project, the software was also ported to Labview for convenience of the researchers.

Because the hardware and the bulk of the firmware was already implemented in WARPWING, the group was able to focus on mechanical system design and development and control system tuning.

C. Layout

The Biomimetic Millisystems Lab at the University of California, Berkeley have developed a hexapedal crawling robot

Fig. 13. The mini quadrotor developed by Daedalus Flight Systems and UMD uses the GINA hardware, but modifies the firmware to include an onboard attitude stability loop to control its four brushless DC motors.

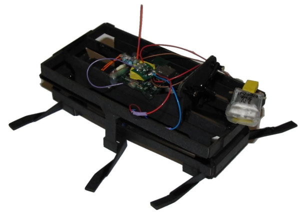

Fig. 14. DASH, a small, lightweight crawling robot developed at UC Berkeley, uses the WARPWING system to measure inertial rates and motor back EMF for dynamic analysis and control.

Fig. 15. This small lightweight footstep detector is designed from the schematic level on up using the WARPWING system as a base, but replacing the accelerometer with one far more sensitive.

called DASH [6]. In order to study the dynamics of DASH towards a future adaptive controller, the researchers needed to incorporate an IMU onto the robot. The WARPWING platform provided a lightweight, low power, and easy to use single board solution that interfaced to the existing motor driver to control the robot.

One additional requirement for the system was to measure and report an analog signal from the robot: the back EMF of the motor. In the default GINA layout, there was an unused ADC input to the processor, but it wasn't exposed in the expansion header. It is possible then to adjust the layout of the GINA board to rewire an ADC input to an expansion pin rather than a GPIO, allowing it to be used for the additional analog sensor. In this case, the firmware was also modified slightly to read that ADC channel and send it back over the radio, and the software was amended to handle that data appropriately.

D. Schematic

Another project in our group at UC Berkeley considered the use of networked sensors (shown in figure 15) distributed across a floor in a room to determine the location of footsteps using the time difference of arrival of the vibrations at each of the sensors. These sensors could potentially be dispersed by slightly larger autonomous robots, and so minimizing size and weight is an important consideration.

The WARPWING system can provide most of the needed functionality from measuring sensor readings to forming a time-synchronized mesh network, but the accelerometer is designed for robotic systems requiring higher dynamic range at the expense of sensitivity. The system was modified to replace the accelerometer with a much more sensitive analog sensor. This required the change to propagate through the layout, firmware and software, but still much of the system remained unchanged, greatly speeding development time.

V. Conclusions

The open source model for software development has begun to make inroads into the hardware community, and the WARPWING project fully embraces that approach. Starting from GINA, an extremely small and lightweight but powerful wireless IMU and processor, the WARPWING project provides a very broad base from which to design robot controllers and more generally sensor motes for which size and weight are at a premium. The examples presented above show that any of the system can be changed, whether it be a sensor composing GINA, the layout of the board, the firmware running on the GINA processor, or the software running at the base station. The rest of the system can still be used as is, thus obviating the need for redundant development effort, and greatly increasing the efficiency of robotic research and design.

As is the nature with open source projects, developments made by users in the community are fed back into the project to be spread throughout the system. As more users contribute to WARPWING, it becomes more versatile and less new development is needed for new applications. At that point, it is conceivable to take a step back and envision a hardware compiler – software that takes the various modules developed by users and intelligently resolves dependencies to generate a complete controller from a block level description of desired components. While that system is still a distant vision, the current WARPWING platform does greatly simplify robot controller development, and increases research efficiency.

Acknowledgments

The author would like to thank the users of the WARPWING system, in particular the researchers at GA Tech, Daedalus Flight Systems, UMD, and UC Berkeley, for their contributions and feedback. The author would especially like to thank Dr. Anita Flynn for her tireless help with design and documentation. Portions of this research were sponsored by the Army Research Laboratory under Cooperative Agreement W911NF-08-2-0004.

References

[1] T. Watteyne, A. Mehta, and K. Pister, "Reliability through frequency diversity: why channel hopping makes sense," in *PE-WASUN '09*. New York, NY, USA: ACM, 2009, pp. 116–123.

[2] A. M. Eldredge, "Improved state estimation for miniature air vehicles," Master's thesis, Brigham Young University, 2006.

[3] "Microstrain 3DM-GX2," http://microstrain.com/3dm-gx2.aspx.

[4] E. Beyer and M. Costello, "Measured and Simulated Motion of a Hopping Rotochute," *J. of Guidance, Control, and Dynamics*, vol. 32, no. 5, 2009.

[5] ——, "Performance of a Hopping Rotochute," *International J. of Micro Air Vehicles*, vol. 1, no. 2, pp. 121–137, 2009.

[6] P. Birkmeyer, K. Peterson, and R. Fearing, "Dash: A dynamic 16g hexapedal robot," *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 2683 –2689, Oct. 2009.