jNode: a Sensor Network Platform that Supports Distributed Inertial Kinematic Monitoring

Philipp M. Scholl and Kristof Van Laerhoven
Embedded Sensing Systems, ESS
Technische Universität Darmstadt
Email: {scholl,kristof}@ess.tu-darmstadt.de

Dawud Gordon, Markus Scholz and Matthias Berning
Pervasive Computing Systems, TecO
Karlsruhe Institute of Technology
Email: {gordon,scholz,berning}@teco.edu

Abstract—**Because of the intrinsic advantages of wireless inertial motion tracking, standalone devices that integrate inertial motion units with wireless networking capabilities have gained much interest in recent years. Several platforms, both commercially available and academic, have been proposed to balance the challenges of a small form-factor, power consumption, accuracy and processing speed. Applications include ambulatory monitoring to support healthcare, sport activity analysis, recognizing human group behaviour, navigation support for humans, robots and unmanned vehicles, but also in structural monitoring of large buildings. This paper provides an analysis of the current state-of-the-art platforms in wireless inertial motion tracking and presents a novel open-source and open-hardware hybrid tracking platform that is extensible, low-power, flexible enough to be used for both short- and long-term monitoring and based on a firmware that allows it to be easily adapted after being deployed.**

I. Introduction

In this paper we introduce a novel research platform for supporting distributed kinematic analysis. Physical systems of interest for monitoring are either inanimate objects such as machines or buildings, living beings such as humans or animals, or combinations of the two. Inanimate objects are generally reactive, and motion results from forces in their environment being exerted on them, for example structural monitoring of buildings during earthquakes [1]. Human beings, while also subject to external forces, are not only reactive but also proactive. Still, both adhere to the same kinematic laws of physics, but predicting the motion of reactive systems can be done with a much higher confidence. This leads to different monitoring applications [2], [3], making it important to distinguish between them in terms of requirements.

Placing miniature inertial sensors on monitored systems has proven to be an effective method for measuring their kinetic properties [4]. As opposed to video and audio-based monitoring solutions for example, this method allows sampling of parameters which are very difficult to sense remotely, such as acceleration and vibration [4]. One drawback of attaching miniature sensors to complex kinetic systems is that each sensor can only sample parameters local to a specific area. This information must then be aggregated in order to monitor the system as a whole. In the next section we analyze three different use-cases and derive requirements for a monitoring tool. After that the system design of the jNode platform is presented and evaluated in the final section.

978-1-4673-1786-3/12/$31.00 © 2012 IEEE

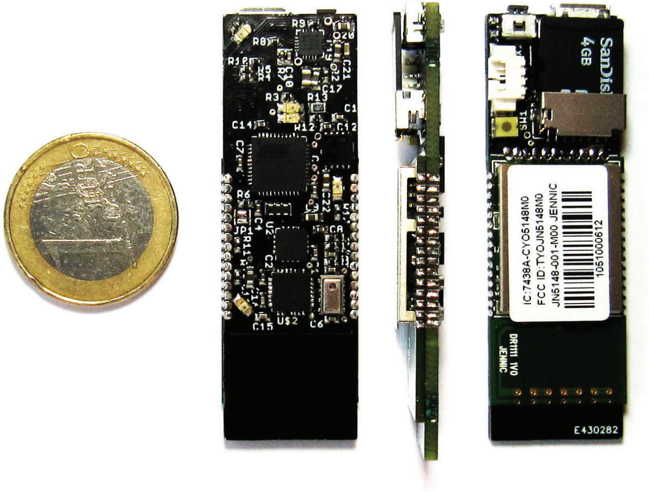

Fig. 1: The jNode platform ($51x17x7$mm) with sensors on the top (left) and the Jennic JN5148 wireless μcontroller, μSD-card and USB adapter underneath (right).

II. Analysis

In this section we derive the requirements for a sensor platform for performing different kinds of kinematic investigations directed towards machines, humans and hybrid systems (i.e. systems which incorporate aspects of the former systems). These are based on extensive practical experience and motivated by the following three use-cases.

A. Machine Monitoring

In today's industrial environments, permanent sensors and asset supervision systems deliver the data for plant automation and maintenance. These sensors are typically installed based on best practice knowledge and may not deliver sufficient diagnostic information for investigating operational problems occurring after decades of machinery run-time. In [2], [5] functional requirements for a flexible, ad-hoc deployable industrial servicing platform were identified based on a service worker use-case: for linking data to machine parts the *capability to identify machinery and machine parts* is required. *time synchronization, high frequency sampling and an optimzed*

TABLE I: Technical properties of wireless inertial measurement units from publicly available data. Energy consumption measurements are taken at typical usage while sampling the full-set of inertial sensors at the specified frequency and transmitting samples through a wireless channel. For clarity, the proposed jNode platform is highlighted.

	current	maximum range	size	customizability	maturity	processing power
XSens MTw	65mA @ 120Hz	$\pm 1200°/\mathrm{s}, \pm 120\mathrm{m/s}^2$ $\pm\, 1.5 gauss$	$58x35x14$mm (packaged)	black box	> 5 years commercial	-
Shimmer	39mA @ 500Hz	$\pm 500°/\mathrm{s}, \pm 60\mathrm{m/s}^2$ $\pm\, 4 gauss$	$53x32x25$mm (packaged)	modular design, tinyos	< 3 years commercial	msp430 @8MHz
jNode	36mA @ 500Hz	$\pm 2000°/\mathrm{s}, \pm 80\mathrm{m/s}^2$ $\pm\, 8.1 gauss$	$51x17x7$mm	open hardware, contiki	academic	JN5148 @16MHz atmega @16MHz
Orient-2	49mA @ 64Hz	$\pm 1200°/\mathrm{s}, \pm 60\mathrm{m/s}^2$ $\pm\, 6 gauss$	-	specific firmware	academic	DSP @12MHz
SportSemble	50mA @ 1kHz	$\pm 11000°/\mathrm{s}$ $\pm\, 1200\mathrm{m/s}^2$	$50x56$mm	specific firmware	academic	avr32 @16MHz

sensor set are required for Accurate and distributed acquisition of relevant machine data, as well as *ad-hoc enabled, robust wireless connectivity is mandatory* to distribute sensor nodes across the machinery to be monitored. *Device-discovery and security* and *compatibility with current and future devices* is necessary to connect the system to hand-held/existing diagnostic equipment. For in-situ data analysis, *sufficient local memory* and *real-time pre-processing of data using typical signal analysis algorithms* must be possible. The platform should further be *highly configurable and flexible regarding the deployment of software* so that a service engineer may react to specific diagnostic tasks on site.

B. Group Activity Recognition

Nowadays, many of us constantly carry one or more intelligent devices, and the number of intelligent systems in our environment is steadily increasing. A novel challenge in the field of ubiquitous computing is the recognition of not only the context of the single user who is interacting with the device but now attempting to recognize the activity of a group of individuals who are in a specific environment or interacting with the system [6]. The group activity is not necessarily the same as the sum of the activities of the individuals in it, but rather is a (complex) function of the activity or context of all individuals in the group. Therefore, data from wearable sensors on multiple users must be aggregated in order to infer the group context [3]. From [3], [6] the requirements for a platform facilitating this type of research become clear: *peer-to-peer communication* is required since devices require information sampled at remote locations. Due to the different amounts of data to be communicated a *high bitrate* is advantageous. *Significant processing power and memory* are required locally in order to save and execute activity models as well as save sensor and model data. Accurate *time synchronization* is necessary for the temporal correlation of distributed sampled sensor data. *Large array of sensor modalities* is useful in order to fully capture the context of environments and activities. Lastly, *long battery lifetimes* are an important requirement to enable several days of sensor node run-time.

C. Embedded Motion Tracking

This use-case focuses on monitoring motion of sensor node-enriched tools or objects which are used in order to accomplish a task. In particular, we want to apply this approach to the domain of firefighter indoor navigation support. In contrast to previous work in this field [7], [8], we want to investigate if navigation information can be derived from sensors attached to the fire hose or lifeline, which both are crucial firefighting tools [7], [8]. Various requirements must be met by such embedded sensors: a *small form factor* is required to allow attachment onto rope-based lifelines with a diameter of 18mm. The possibility to *reprogram the software in-situ* is necessary to allow firmware modification after embedding the sensor nodes. A *flexible hardware platform* supports testing various sensors. While *reliable on-node storage of sensor data* and *wireless real-time monitoring* is needed for on-the-fly labeling of data. On-node storage allows gathering of data at high speed in an energy-efficient and reliable way and real-time monitoring allows the sensor data to be labeled and synchronized with other monitoring systems, like cameras.

D. Related Wireless Sensor Platforms

In the past, sensor nodes providing these features have emerged. Among the most popular is Shimmer [9] targeted at short- and long-term biomedical studies, Orient-2 [10] which targets body posture estimation via inertial measurements, the well-known XSens MTw platform [11] and SportSemple [12], a device to help baseball players monitor their performance. All of these sensors generally provide the means to monitor human motion, however they fail to fulfill at least one of the above requirements. Especially the demand for high flexibility in software and hardware e.g. through an open-hardware and software approach and the optimization for machine monitoring are rarely met in commercially available kinematic monitoring tools. An overview of the features of these devices compared to the jNode platform can be found in Table I with Shimmer being the most comparable platform.

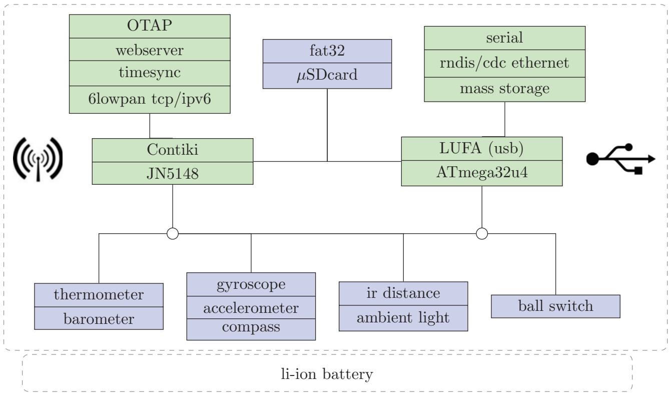

Fig. 2: System Architecture: Two micro-processors handling USB and IEEE802.15.4 traffic, connected via a serial line and mutually exclusive access to the µSD-card and sensors.

III. System Design

Three requirements introduced in the last section have primarily driven the design of the jNode: *reliable monitoring* using a *large number* of nodes for *long periods of time* in an *ad-hoc setting*, ensuring the simple *modification of hardware and software*, and the integration of a *large number of sensing capabilities* on the node. Fig. 2 shows the general HW/SW system architecture.

All of the sensors used are interfaced via I^{2}C. For kinematic observations sensors were added to sense acceleration, magnetic heading and angular acceleration. These exhibit comparable specifications to sensors used in systems to estimate human motion like the XSens system. Furthermore a barometer allows measuring the relative altitude of the node, and an integrated infrared distance sensor has been added for the purpose of detecting objects in the vicinity of the node. An array of additional infrared-LEDs allows camera-based remote identification in augmented reality applications [5]. Additionally a micro-vibration switch can wake up the device from sleep when physical movement is detected. A µSD-card is used for long-term storage of sampled data. The core of the system is a Jennic JN5148 System-on-Chip, which consists of a 32-bit microcontroller and an IEEE802.15.4-compliant radio transceiver, an ATMega32 provides USB-capabilities when connected to a PC. We ported the open source Contiki operating system [13] to the Jennic JN5148 microcontroller. This allows users to benefit from a large number of existing functionalities and libraries. These include the 6lowpan network stack and functions like FFT calculations, and over-the-air run-time dynamic linking.

Wired communication via USB is implemented with the Lightweight-USB-Framework-for-AVRs (LUFA) through three USB devices: a serial-line device, an RNDIS-network device and a mass-storage device. The mass-storage emulation is used to access data stored on the Fat32-filesystem of the µSD-card. Accessing the 6lowpan-network is accomplished via an RNDIS-network device which enables IP-based access without any special program on the PC. However, network access can also be accomplished via the serial line which allows network access on mac-level. Furthermore the microcontroller can be reprogrammed (hands-free) and applications can communicate with a PC via the serial line.

As pointed out in section II several applications require long-term operation of the device, i.e. low-energy consumption. Table II contains the energy consumed by each component on the jNode. The majority of energy is consumed by the RF-transceiver, making it a prime candidate for energy optimization, which we do by duty-cycling the network. Here, energy can also be conserved by compressing the data allowing for longer "silent" periods. But compression is also needed to transmit data over the wireless link - the worst case data-rate resulting from sampling the complete sensor set at full rate can be calculated as $12.29\mathrm{KiB/s}$. However the total bandwidth of a 6lowpan UDP-network with n nodes is limited to $\sim \frac{20}{n}\mathrm{KiB/s}$ per node. So, in order to conserve energy and transmit samples over the wireless link, data needs to be compressed. However, the driving factor is not a high compression ratio but rather a *guaranteed* ratio. Which is the reason for decreasing the sample rate to an application-specific value, and to run a mean-variance compression scheme on top of the sampling process.

The second candidate for saving energy is the JN5148, where using special features the main processor can be turned off while peripheral controllers continue to work (including the RF transceiver); processing will continue as soon as a hardware interrupt occurs. So, for example, whenever new data is ready on the I^{2}C-bus or the RF link, the CPU wakes up. This is hard to manage in software, but the event-based nature of the Contiki operating system allows for a very elegant solution. During the main scheduler loop, we check if there are any pending events to be processed, if there are none we can check when the next timer event shall be fired and put the device to sleep until that time.

The wireless network link provides time-synchronization, control over the nodes' operation, real-time monitoring of sensor data and reprogramming capabilities. Protocol-wise we exploited the Guaranteed Time Slot mechanism of the IEEE802.15.4-protocol for transmitting sensor data. And the 6lowpan TCP/IP implementation of Contiki for time-synchronization, reprogramming and controlling the nodes. This has the advantage that standard protocol stacks can be used on a PC to communicate with the jNode and allows the construction of heterogenous networks with different implementations of wireless sensor network nodes.

IV. Evaluation

As a first measure of comparison with other platforms we measured the energy consumption of the jNode platform. We did so with standard lab equipment, i.e. an oscilloscope, a lab power supply (3.3V) and a small resistor (10Ω) put in series in front of the main circuitry. The voltage drop across that resistor is then linear related to the current drawn by the jNode board. Table II contains the measured figures for each component. To obtain these values we first measured the

TABLE II: A listing of the main components on the jNode, their respective measured average power consumption at $3.3\mathrm{V}$, and achievable sampling frequencies. These figures contain the worst-case power consumptions without using any energy optimizations, and at full achievable sensor sampling rates.

	component	current	rate	resolution
$^\circ\mathrm{C}$	MPL115	$0.6\mathrm{mA}$	$321\mathrm{Hz}$	$10\mathrm{bit}$
kPa	MPL115	$0.7\mathrm{mA}$	$321\mathrm{Hz}$	$10\mathrm{bit}$
lx	VCNL4000	$1.3\mathrm{mA}$	$10\mathrm{Hz}$	$16\mathrm{bit}$
$\mathrm{cm}(ir)$	VCNL4000	$1.0\mathrm{mA}$	$860\mathrm{Hz}$	$16\mathrm{bit}$
$\mathrm{m/s}^2$	LSM303	$0.6\mathrm{mA}$	$708\mathrm{Hz}$	$3x12\mathrm{bit}$
$gauss$	LSM303	$0.5\mathrm{mA}$	$500\mathrm{Hz}$	$3x16\mathrm{bit}$
$^\circ/\mathrm{s}$	L3G4200D	$6.2\mathrm{mA}$	$644\mathrm{Hz}$	$3x16\mathrm{bit}$
RF RSSI	JN5148	$17.5\mathrm{mA}$	$250\mathrm{Hz}$	$8\mathrm{bit}$
CPU	JN5148	$5.5\mathrm{mA}$	$16\mathrm{MHz}$	
	ATmega32	$10\mathrm{mA}$	$16\mathrm{MHz}$	
RF TX/RX	JN5148	$16.3\mathrm{mA}$	$250\mathrm{Hz}$	
µSD	-	~3.4mA		
total	RF+µSD	$36.05\mathrm{mA}$		
total	RF	$32.65\mathrm{mA}$		
total	µSD	$20\mathrm{mA}$		

baseline consumption of the JN5148 CPU core by turning off all other components on the board. Then we measured RF consumption by putting the system into constant receive and transmission mode. After that we obtained the current consumption and maximum sampling frequency for each sensor individually through the standard Contiki application programming interface (API), i.e., in the same way an application would obtain the sensor readings. These figures present the worst-case current consumption when the system is under maximum load. Compared to other systems (refer to Table I), the jNode shows a lower consumption which is due to the specific choice of components.

Furthermore we tested the maximum achievable individual sensor sampling rates, which can be found in Table II. For testing, each sensor was individually activated and newly sampled values transmitted through an irq-driven $1\mathrm{M}Baud$ serial line, which allowed us to determine its maximum sampling rate. We found that sampling the complete sensor set, excluding the light sensor, can be done at $110\mathrm{Hz}$ and only reading the accelerometer, gyroscope and compass at $477\mathrm{Hz}$. Because some sensors synchronize only via clock stretching, the sampling rate for the complete sensor set is rather low. This situation can be remedied however by either using interrupt lines or clock-based synchronization in the driver's code. As it is a prototype platform intended for academic exploration, the jNode's hardware is open source, with its schematics and design freely available[1], allowing others to build the same or similar platforms based on this work and encouraging reproducible experiments.

V. Conclusion and Future Work

We introduced the jNode platform that is energy-efficient, allows both short- and long-term monitoring and is highly customizable (both in software and hardware). Especially the provision of two powerful and energy-efficient micro-controllers allows it to be adapted to the processing requirements of different applications. While it does not support that many sensor modalities yet, sensors for inertial observations, relative distance based on infrared light, and identification methods for camera-based augmented reality settings are supported. Providing a self-contained development and data-distribution environment accessible via each node in the network is the next step in simplifying the jNode system. For this, facilities provided by a web-server on each node, to edit application code on single and multiple nodes of the network, and to distribute data via web-services will be included in future work. Due to jNode's ability to be modified after being deployed, its small form factor, reliable data storage, and its processing power it can be used to quickly test new ideas without major efforts, making it an adequate tool for a large number of kinematic investigations.

Acknowledgments

The authors would like to acknowledge partial funding by the German Federal Ministry of Education and Research (BMBF) through the Koordinator project.

References

[1] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks," *2007 6th International Symposium on Information Processing in Sensor Networks*, vol. 9, no. 3, pp. 254–263, 2007.

[2] N. L. Fantana and T. Riedel, "Servicing industrial machinery: Challenges using wireless identification and networked sensing systems," in *6th International Conference on Networked Sensing Systems*, 2009.

[3] D. Gordon, M. Scholz, Y. Ding, and M. Beigl, "Global Peer-to-Peer Classification in Mobile Ad-Hoc Networks: A Requirements Analysis." in *Proceedings of the Seventh International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT'11)*, 2011.

[4] G. Welch and E. Foxlin, "Motion Tracking: No Silver Bullet, but a Respectable Arsenal," *Ieee Computer Graphics And Applications*, 2002.

[5] M. Berning, T. Riedel, Y. Ding, T. Miyaki, N. Fantana, and M. Beigl, "Augmented Service in the Factory of the Future," in *Adjunct Proc. of the International Conference on Networked Sensing Systems*, 2012.

[6] D. Gordon, J.-H. Hanne, M. Berchtold, T. Miyaki, and M. Beigl, "Recognizing Group Activities using Wearable Sensors," in *8th International ICST Conference on Mobile and Ubiquitous Systems*, 2011.

[7] H. G. C. Fischer, "Location and Navigation Support for Emergency Responders: A Survey," *Pervasive Computing, IEEE*, 2010.

[8] L. Ramirez, M. Betz, T. Dyrks, M. Scholz, and J. Gerwinski, "Landmarke: an ad hoc deployable ubicomp infrastructure to support indoor navigation of firefighters," *Personal and Ubiquitous Computing*, 2011.

[9] A. Burns, B. R. Greene, M. J. Mcgrath, B. Kuris, S. M. Ayer, F. Stroiescu, and V. Cionca, "SHIMMER™ – A Wireless Sensor Platform for Noninvasive Biomedical Research," *Sensors Journal, IEEE*, 2010.

[10] A. D. Young, "From Posture to Motion: The Challenge for Real Time Wireless Inertial Motion Capture." *Proceeding of the 5th International Conference on Body Area Networks*, 2010.

[11] D. Roetenberg, H. Luinge, and P. Slycke, "Xsens MVN: Full 6DOF Human Motion Tracking Using Miniature Inertial Sensors," 2009.

[12] M. Lapinski *et al.*, "A Distributed Wearable, Wireless Sensor System for Evaluating Professional Baseball Pitchers and Batters," *International Symposium on Wearable Computers*, 2009.

[13] A. Dunkels and T. Voigt, "Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors," *Computer Networks*.

[1] JenniSense Open-Hardware Project: https://github.com/teco-kit/Jennisense