Hermes: A versatile platform for wireless embedded systems

André Rodrigues, Tiago Camilo, Jorge Sá Silva, Fernando Boavida
Centre for Informatics and Systems
University of Coimbra
Coimbra, Portugal
{arod, tandre, sasilva, boavida}@dei.uc.pt

Miguel Silva, Nelson Blanco, José Pedro, João Martins
PDMFC
Lisboa, Portugal
{miguel.silva, nelson.blanco, jose.pedro, joao.martins}@pdmfc.pt

***Abstract*—This paper presents Hermes, a new modular platform that can be used in a variety of scenarios, including the role of body area network coordinator, enabling temperature, movement and heartbeat sensing, local and remote communication, and indoor and outdoor localization. Hermes is based on two main modules that can operate as a single system in a coordinated fashion or individually, depending on the application scenario. One of the modules uses a Telos-inspired architecture, with new processing, communication, sensing, storage and energy subsystems, and executing TinyOS. The other module is designed around a PIC 24F MCU, also supporting communication, sensing, storage, and executing a custom operating system. Both modules have expansion capabilities. The main innovative aspects of the implemented platform are its modularity and its capability to provide device abstraction, which considerably eases application development. The paper addresses the platform motivation, requirements, hardware architecture, implementation details, and performance evaluation.**

Keywords-sensor networks; body area networks; mote; embedded system

I. INTRODUCTION

In this work we present a new hardware platform that was developed not only as a response to real market requirements, but also to support research and development of wireless network embedded systems such as wireless sensor networks (WSN) or body sensor networks (BSN). The initial objective was to develop a hardware platform that could be used in a system designed to help Alzheimer patients, by making it possible to localize them (indoor and outdoor) as well as detect events that could put them in danger. Nevertheless, the targets list grew up, and now it can be described as a network sensing and coordinating node.

The design of this platform built on top of the WSN knowledge accumulated by the team over the last years, in order to reduce development time and risk. This translated into following some of the design solutions pursued by Telos [1] (and by some of its derivatives), as well as into supporting TinyOS, the most common operating system for WSNs. We also took advantage of recent hardware developments to develop a state-of-the-art node.

This paper presents the motivation, requirements, architecture, implementation, and evaluation of the developed Hermes platform, whose innovative aspects are the following:

- compact and modular approach that targets a variety of application domains;
- possibility of using each module as a standalone platform or of combining the functionality of both modules in a flexible way, to better support application requirements;
- communication and device abstractions that ease application development work, by allowing the developer to access the various sensing devices in the same way, irrespectively of the module they actually reside in;
- rich set of hardware components that, taken individually, are not uncommon in other platforms but, as a set, are not available in compact platforms; these include a very generous selection of sensors, integrated power management system, SD card storage, cellular and short distance radios, GPS and RFID receivers.

The paper is organized as follows. Section 2 presents the requirements underlying the platform, which led to the logical and hardware architectures presented in section 3. The justification for the component choices and the most relevant implementation details are presented in section 4. The Hermes platform was subject to intense testing and evaluation, whose main results are presented and discussed in section 5. Section 6 provides the conclusions and guidelines for further work.

II. REQUIREMENTS

The first motivation for this work arrived from a psychiatric institution, in the form of a request to develop a system that could ease the monitoring of hospitalized patients. The initial requirement was that in case a patient entered reserved areas or left the institution, an alert message would be sent, followed by the initialization of a localization procedure. The list soon grew up to support requirements such as monitoring the patients' physical condition, detect accidents, or enabling a person to request help.

As we had already developed platforms for other sensing application scenarios and we were also involved in research activities, we transformed this new development request into an opportunity to analyze what other solutions and/or research activities could benefit from the new platform. The idea was to widen the applicability of the platform without compromising the requirements of the original driver application scenario. As a result, we decided to also incorporate the requirements of two other systems previously developed by the team: the first is an equine monitoring system, named iHorse [2], initially using TMoteSky, and currently using IRIS, requiring specifically developed add-on boards; the second one is an outdoor localization and logging device used in golf greens.

Table 1 presents the requirements that the Hermes platform should meet, in order to accommodate the partial requirements of the various scenarios, namely Hospital (i.e., the driving application scenario), iHorse, Golf, and Research (i.e., platform use for research purposes).

Concerning the **Hospital** application scenario, the client concerns, which led to the requirements in Table 1, included the ability to detect patients' potential danger situations such as falls, support for an easy to use alert mechanism, indoor and outdoor localization and communication, comfortable use because the device is to be used during long periods of time (even during bath) either attached to a belt or to an ankle, and one-week energy autonomy.

The **iHorse** application scenario posed the following overall requirement: support for online continuous monitoring, allowing users to access and evaluate the subject's vital signs, as well as behavioural and environment parameters, anytime and from anywhere, as long as there is an Internet connection. This translated into several requirements that are met by the existing iHorse platform and, thus, should also be met by Hermes, namely, sensing movement, posture, and heartbeat, capability to communicate with the local management unit having as much range as possible to reduce the need for router devices, flash based storage of sensor raw data to cope with communication problems, energy autonomy of at least one month, registering and alerting functionality, horse comfort and equipment robustness to shock.

TABLE I. PLATFORM REQUIREMENTS ACCORDING TO USE CASE

Requirements on	**Hospital**	**iHorse**	**Golf**	**Research**
Sensing				
- movement	●	●	●	●
- position	●	●		●
- heartbeat	●	●		●
- ambient temperature				●
Actuation (alert button)	●		●	●
Comunication				
- WLAN	●	●		●
- WWAN	●		●	●
Localization				
- indoor	●			●
- outdoor	●		●	●
Storage		●	●	●
Energy				
- lifetime (days)	● (7)	● (30)		●
- remaing energy level	●	●		●
- energy consumption				●
Physical caracteristics				
- compact and light	●	●		●
- water resistant	●	●	●	
- shock resistance	●	●		

The main requirements for the **Golf** application were team localization (to optimize the admittance of new teams into the field), movement detection, help and assistance button, and enough storage to register car trajectories and field information.

The **Research** use case global requirement was to have as much flexibility as possible, which translated into the platform having all the capabilities that were already required by the other applications plus ambient temperature sensing, real-time energy consumption measurement, being as modular as possible, and supporting development approaches already known to the community.

Before deciding that there was a need to develop a new platform to support the requirements of the Hospital application (our primary target) other possibilities were evaluated. One of them was the use of a Smartphone hardware platform that has native support for GSM/GPRS, WIFI, GPS, accelerometer, and energy management. The decision not to go that way was based of the following set of reasons:

- the device should be energy savvy, support an indoor localization mechanism, be comfortable to use (i.e. light and small), and be water resistant. We did not find Smartphone offers that satisfied these requirements. Even if we adapted some of the offerings, the following two reasons still applied;
- it was difficult to have access to suppliers of Smartphone platforms for small quantities;
- basing a solution on a Smartphone platform would make future solution support more difficult because of the constant hardware innovation in this fast moving area.

Another possibility would be Shimmer [3], a well-known platform for research in connected health and assisted living solutions that was designed with wearable health sensing in mind. Shimmer is a modular platform with the following characteristics: MSP430F1611 microcontroller, Bluetooth and 802.15.4, micro SD card, 3-axis accelerometer, tilt / vibration sensor, Li-ion battery management, and internal and external connectors. In addition it has a set of add-on expansion boards that can be grouped according to their functionality into: kinematic sensing (gyroscope, magnetometer and 9DoF boards), biophysical sensing (ECG, EMG and GSR boards), and ambient sensing (PIR, GPS and temperature/pressure boards). The hardware is protected inside a light and small sized box that can be attached to the body via straps. In what concerns software, the system uses TinyOS at its core and can interface to data analysis software. Considering all its offers, Shimmer is a very interesting product. Nevertheless it was not used because:

- it does not support WWAN communications (relying on the use Bluetooth communications to connect to an external gateway) which would require using an extra device or to develop an expansion board;
- it does not support the necessary localization mechanisms;
- the offered case is not water resistant;

- there is no guarantee that the device will have support / exist for the solution projected lifetime.

III. HERMES ARCHITECTURE

A wearable device should be comfortable to use. This implies to be small sized and light. The usual approach, in mobile devices, is to design the hardware in order for it to fit a single board. The decision not to go that way was due to the facts that this would restrict the platform's flexibility (benefiting the Hospital use case at the expense of others), complicate debugging, increase the placement area, and make the RF design more difficult. A two-board solution is an acceptable compromise between flexibility and size.

Further on, it was necessary to decide which functionality to deploy in each board. The requirements of each application (presented in Table 1) and the practice followed by others (e.g. MicaZ, Telos, and Shimmer) were taken in consideration in this decision. The final architecture requires the use of both boards (named Pegasus and Fenix) for the hospital application case and just one board for the other application scenarios. When both boards are used the platform is called Hermes and it provides the capabilities presented in Fig. 1 (NOTE: several functions are omitted for clarity - buttons, leds, energy metering, SD card; those are clearly identified in each board's description, later on).

The followed approach was to have the typical WSN node functionality (processing, storage, sensing, and short range communication), a power management system based on LiPoly batteries, debugging and programming hardware, and expansibility connectors in the Pegasus board. Pegasus satisfies all the functional requirements of iHorse as can be verified by comparing their description (see Table 1) with the hardware modules presented in the Pegasus functional diagram in Fig. 2.

The second board, named Fenix, can be described as a localization system supporting indoor and outdoor localization, WWAN communication, processing, storage, movement detection, debugging and programming hardware, and expansibility connectors. When used in the context of the Hospital application case, it provides localization and WWAN capabilities; when used as standalone module, as is the case of the Golf application, is provides localization, communication, storage, and processing services, being an autonomous system that just requires a power subsystem. The Fenix functional block diagram is presented in Fig. 3.

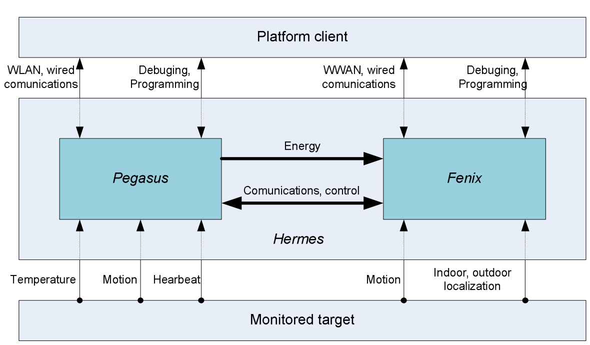

Figure 1. Hermes simplified architecture.

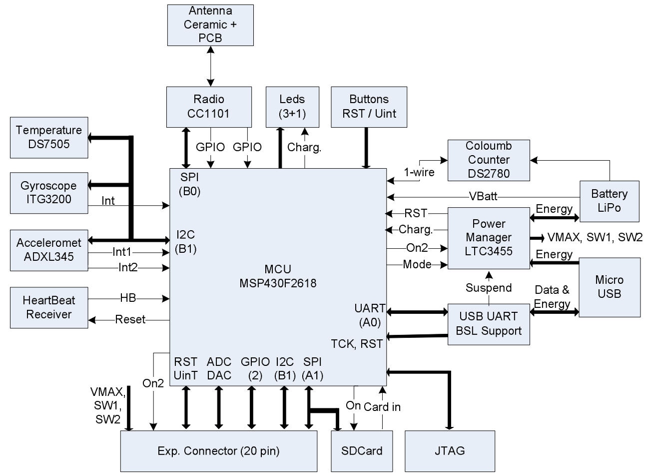

Figure 2. Pegasus functional block diagram.

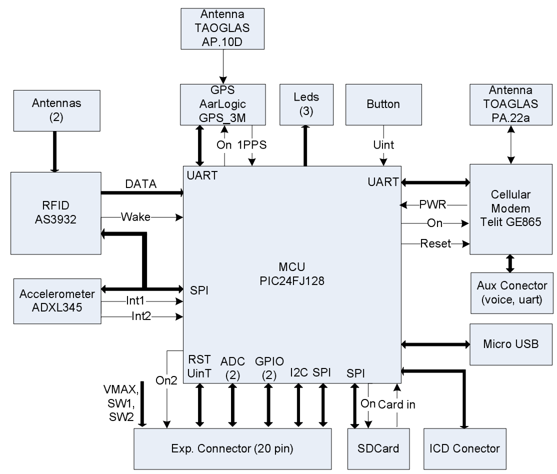

Figure 3. Fenix functional block diagram.

IV. IMPLEMENTATION DETAILS

This section addresses the component choices made for the Hermes platform (with each board being described in a subsection) and an explanation on how the functionality created to ease application development using both boards were implemented. Fig. 4 presents the Hermes platform (Pegasus on the bottom and Fenix on the top).

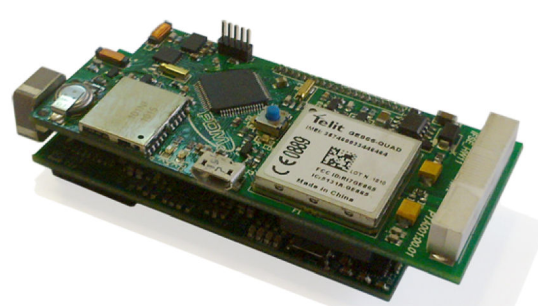

Figure 4. Hermes platform.

The total Hermes platform dimensions are 83 x 36 x 22 mm, with a weight of 36 g (battery adds 22 g).

A. Pegasus

The design of this board was inspired by the architecture of Telos, which offers sensing, processing, communication, programming / debugging, and expandability options in a single board design. The main idea was to take advantage of the available TinyOS 2 (T2) codebase and team knowledge in this operating system on several platforms. The initial goals for Pegasus design were to: ease portability to T2, have a more recent processor and an increased communication range compared to Telos, support an SD card, support all the applications' sensing requirements, support a power management and accounting system for LiPoly batteries, have expandability capabilities, offer debugging and programming functionalities, be compact and light. The functional block diagram identifying the chosen components is presented in Fig. 2. Below, a succinct explanation of the reasons for choosing each component is given.

1) Processing

The selected processor was TI MSP430F2618, a 16 bit RISC architecture that runs at up to 16MHhz and has 128KB Flash, 8KB RAM, 0.5mA/MHz active current, 1.1uA sleep, 1uS wakeup, as well as an array of very energy-efficient hardware such as DMA, ADCs, DACs, SPI, I2C, UART, HW multiplier. The TI MSP430 series offers several advantages when compared with other options, as stated in [4]. Compared to MSP430F1611 (used in Telos and Shimmer), MSP430F2618 doubles the maximum operating frequency, more then doubles the Flash memory, reduces the sleep current (from 2.6uA to 1.1uA) and the wakeup time (from 6uS to 1uS), provides more precise oscillators, and offers two universal serial communications interface (USCI) modules with a total of four independent serial communication channels. The MSP430 series 5 runs at up to 20MHz, reduces the active current (from 0.5 to 0.28 mA/MHz), but has higher sleep current (1.7uA) and wakeup time (6uS) than series 2 microcontrollers, offers more peripherals like USB, real time clock (RTC), and up to 4 USCI modules. Other recent proposals like the PIC24 XLP series from Microchip, which offers very low power sleep modes [5], or the EFM32 [6] from Energy Micro, an ARM Cortex-M3 architecture with active power consumption of 160uA/MHz, 2uS wakeup time, and 1.2uA sleep mode with RAM retention, are also very efficient. We decided to use the MSP430F2618 because at the time we began designing the system (late 2009) there was no support for the MSP430 series 5 processors in T2 (or contributed code), the PIC24 was not supported by the T2 tool chain compiler, and the EFM was not available.

2) Communications

The communication range is an important factor, since in our application scenarios it is necessary to cover an area as large as possible without deploying extra router nodes. Modern IEEE 802.15.4 based radios (e.g. TI CC2520, Atmel RF231) improve the link budget (~9dB) when compared with the well-known CC2420 (used in Telos and Shimmer), enabling the development of nodes with higher range and supporting interoperability between different hardware (e.g. interoperability between IRIS and Telos). Newer radios like Energy Micro Draco [7], introduced in 2011, go even further, with an improved link budget of 14dB and very low energy consumption. Nevertheless, they were not available. To further increase the communication range two options were available: the use of high-gain antennas and/or the use of signal amplifiers. The first solution did not apply to the Hospital / iHorse use cases. The second implied extra energy consumption. We selected the TI CC1101 transceiver that offers a 13dB increase (compared to CC2420) and because the nodes operate in the 868MHz ISM frequency we can expect another increase in the range due to the lower communication frequency factor. This comes at the cost of a reduction in interoperability and world wide usability. This transceiver, integrated in the TI CC430 SoC, is used in the OSIAN [8] WSN technology to improve range, in the TelosW [9] because of its wake-on radio functionality that enables to save energy by supporting hardware-based low power listening, and in GNode [10], a low-cost node based on MSP430F2418.

3) Movement Sensing

The 3-axis accelerometer (Analog Devices ADXL345) and the 3-axis gyroscope (Invensense ITG3200) are very capable devices used in several projects. Both support I2C bus interfaces, have sleep modes, support high frequency sampling, and configurable measuring ranges. In the case of ADXL345, it also generates interrupts for detecting specific movement patterns and can balance between resolution and consumption. In WSN hardware, the ADXL345 is used in the Zolertia Z1mote [11], and the ITG3200 in the GINA OpenWSN mote [12]. The reason to select digital devices was that we wanted to move the processing closer to sensing. This is already supported with the advanced features of ADXL345, but we wanted to go further and support sensor fusion using an integrated accelerometer / gyroscope device like Invensense MPU-60X0. The device was not available at production time, but being pin-compatible with the ITG3200, as soon as it is available it will be deployed, substituting the ITG3200 and the ADXL345.

4) Heartbeat detection

The decision to include a heartbeat receiver in the platform instead of developing the ECG detection circuitry was based on three main reasons: the iHorse application already uses a heartbeat belt that perfectly meets the application requirements in what concerns the subject's wellbeing; there are market available heartbeat emitters that are compatible with the CC1101 radio [13] that can potentially be used for other scenarios; one can always substitute the OEM heartbeat receiver module by a custom-made, single-side board that implements a heartbeat detection circuit.

5) Storage

To support behaviour analysis, a requirement of the iHorse application, the movement sensors should acquire data at a minimum of 40Hz to enable capturing movement components in the 0.1Hz to 20Hz range (according to [14], recent wearable motion detectors for human physical activity monitoring sample the accelerometers at ~30Hz, with an exception sampling at 128Hz). The behaviour parameters are calculated online by the node; nevertheless, access to raw data for offline analysis is a requirement to enable algorithm optimizations and future functionality. When calculating the required storage capacity we assumed sampling at 100Hz both for the gyroscope and the accelerometer, which accounts for 12 bytes per sample period. Not saving the raw data and just transmitting it is not safe because there may be relatively long periods of time during which transmission may not be possible or being energy costly. SD memory cards, up to 2GB, are supported providing space for large amounts of raw accelerometer / gyroscope data without the risk of missing samples. Pegasus supports the FAT filesystem using a modified version of the of the T2 port of FatFs (created by ChaN [15]) done by Shimmer Research.

6) Energy

The system is powered by a LiPoly battery (usually a 1000mA capacity one, due to size considerations) via a Power Management System (PMS) that also charges the battery whenever the micro USB connector is being used. The PMS provides two step down switch converters that can operate in pulse wide modulation (PWM) mode or in higher efficiency burst mode (with output voltage ripple slightly higher), offering up to 96% efficiency. The system is configured to use the first switch (SW1) to power the Pegasus board, leaving SW2 free to be controlled and used by the Fenix board. Typical current limit for SW1 is 600mA and for SW2 is 900mA. The quiescent current is ~90uA. The LTC3554, if it were available at design time, would be a good alternative to this PMS, allowing higher efficiency at small current values (i.e. between 0.01mA and 1mA) and a quiescent current of ~8uA. We intend to use this PMS in the next version of Pegasus.

To support available energy estimation, Pegasus includes a coulomb counter that measures voltage, current and temperature, and combines this information with cell-specific data to estimate battery energy level using on chip algorithms. We also use this device to measure energy consumption, by accessing the accumulated current registers. Using this approach to measure energy consumption and to estimate battery energy level consumes more energy than methodologies based on iCount [16]. We did not use that approach because the PMS switches are PWM based (and not pulse frequency modulation based), not allowing to use the iCount approach (i.e. mapping the switching frequency to current consumption). The selected device was DS2780 from Maxim, which enables the measurement of currents with a 78.13uA resolution with a 20mOhm sensing resistor (when Pegasus is used alone we can choose a higher value resistor to increase the resolution) sampled at 18.6kHz with an active mode consumption of 65uA. Being a 1-wire device, it has a universal identifier that can be used for node addressing.

7) Buses

The expansion connector provides VMAX (a PMS output connected to the active energy supply), SW1 and SW2 outputs. Pegasus is designed to supply add-on board devices via SW2. In case the add-on has an MCU it can be supplied via SW1 and should further control SW2. VMAX is used for special cases were there is a need for more power. The I2C bus enables access to Pegasus sensors. SPI is shared between the expansion connector and the SD card reader, requiring the add-on board to do bus reservation. When Pegasus is used in conjunction with Fenix, the SPI bus is used to support MCU-to-MCU communication. Pegasus used individually does not require bus reservation because each USCI port is assigned to a specific use as can be seen in Fig. 2. Other available lines include: reset, user interrupt, two ADC/DAC, and two GPIOs with interrupt support. Node reprogramming is supported via the onboard JTAG connector or via the bootstrap loader.

B. Fenix

As mentioned before, Fenix was designed to be used in standalone mode, as in case of the Golf application, or as a localization and WWAN communication add-on to Pegasus to support the requirements of the Hospital application. Fenix design was highly influenced by the team previous experience in Microchip PIC microcontrollers and with Telit cellular radios. In addition, it was decided to take advantage of in-house available code libraries that offer operating system functionality and driver support. The development tools came from Microchip, which offers a fully integrated IDE that enables simulation and real-time debugging, supports an optimized and ANSI-compliant C compiler, and offers access to libraries for USB and SD card storage that provide USB mass storage profile and FAT filesystem functionality.

1) Processing

The board uses a recent PIC24FJ128GB206 16-bit MCU toping 32MHz (16 MIPS), with 128KB flash and 96KB RAM, offering an array of functionality including 4 UART, 3 SPI, 3 I2C, USB, RTC, 10-bit ADC with 16 channels, five 16-bit timers, hardware multiplier and divider, brown out reset, watchdog, compare/capture/PWM modules, 800uA/MIPS current consumption, and standby current of 22uA. The main reasons to select this MCU were its functionality, software availability, and team's PIC expertise. The price to pay is a small increase in energy consumption that does not impact the node usability.

2) Communications

For GSM/GPRS communications, the decision was to use Telit GE865 Quad since it is ultra compact (uses a BGA package), supports quad-band operation, GPRS class 10, and a Python script interpreter that enables to develop applications that run inside the modem and have Internet access via the integrated TCP/IP stack. This module also offers the possibility to remotely run AT commands and includes support for over-the-air firmware upgrade.

Very short range low frequency data reception is supported by the austriamycro-systems AS3932 3D wake up receiver, which operates in the 110-150 KHz carrier frequency range (supporting a maximum receive data rate of 4kbps). This receiver consumes less than 10uA while listening and, on carrier detection, can be switched to data receive mode, directly wakeup the MCU, or correlate the message preamble with a previously defined 16-bit pattern to decide whether to wake up the MCU. This device can be useful in scenarios were Pegasus (CC1101) or Fenix

(GE865) radios are turned off and a low frequency signal wakes the boards to enable communication. This reduces listen times for the other radios, improving overall energy consumption, but requires the existence of carefully located transmitters.

3) Localization

The hardware selected for the Fenix platform enables several localization mechanisms, for both indoor and outdoor operation, making it always possible for Fenix to be located. For outdoor localization, an Arlogic 3M Sirf 3 GPS receiver is used, connected to an active antenna with a 25 dB high-performance low noise amplifier (LNA). It offers an update rate of 1Hz, being suitable for any type of outdoor localization. If GPS is not active, GSM can extract the cell id and perform the multilateration of the signal from the cells serving the module in order to determine the geographical position of the device. Indoor localization can be achieved using the wake-up receiver described earlier, if appropriate transmitter antennas are installed at certain known points. This method is useful to detect passages from predefined barriers, like doors, and was the reason AS3932 was used.

4) Storage

Storage is supported by means of an SD card reader connected to a SPI bus. The system supports FAT16 and FAT32. Data can be retrieved either by reading the SD card at an external device or by connecting Fenix to a PC USB port, enabling the SD card to be recognized as a USB mass storage device.

5) Buses

The Fenix platform is programmed and debugged using Microchip default programming connector. Future installation of an appropriate bootloader for the Fenix microcontroller will make firmware upgrade easier. Although Fenix was not designed to support voice communications, this was not compromised because a 16-pin expansion slot to transport voice signals to GE865 was provided (this allows extending Fenix capabilities to handle, for instance, voice calls in emergency applications). In order to be connected to Pegasus (or other boards) a 20-pin connector enables access to power, SPI, I2C, ADC channels, GPIOs, user button, and reset signals.

C. Developing Hermes applications

In order to simplify the development of applications that benefit both from Pegasus and Fenix devices (e.g. sensors, radios), we decided to develop the ActiveMessage abstraction to support inter-board communication, thus extending the paradigm already used to develop TinyOS-based WSN applications (i.e. ActiveMessage provides link layer communications functionality). The abstraction is supported by a 5-layers architecture (Fig. 5) that is implemented by Pegasus and Fenix. In Fig 5 most of the interfaces are bidirectional but the arrows are used to indicate the initial direction (when applicable). For illustration purposes, some interfaces (highlighted in bold in the figure) have their commands and events presented.

In the first layer (Physical SPI InterBoard) the communication between Pegasus and Fenix is supported by the use of an SPI bus on USCI port A1. The USCI abstraction for the msp430X chips (Dexma contributed code) was extended to support SPI communication in either slave or master mode (online configurable). The communication primitives' implementation benefits from SPI being a full duplex bus.

The Logical InterBoard is the second layer and controls the SPI bus communication. It performs error recovery, maintains and can export statistics on valid and invalid packets - either transmitted or received (interface omitted for picture simplicity), and maintains information on physical state of the hardware (i.e. transmitting, receiving, idle, and off).

The third layer (Adaption InterBoard) supports the packet level abstractions. The InterBoardPacket (IBPacket) interface supports navigation inside the packet (i.e. extracting header, footer or payload). The Packet and AMPacket interfaces enable the execution of several operations on a packet (e.g., get/change the type or group, get/change the destination and origin, and change the payload length). These interfaces are omitted for simplicity.

The fourth layer is the ActiveMessage InterBoard. This layer provides interfaces that enable to Send, Receive, and interact with a packet (the latter not shown for simplicity).

The higher layer is the Application Layer, which can have multiple instances of the Send and Receive interfaces, each one with a specific *id*, thus enabling packets to be forwarded to the correct listener.

Using the presented communications architecture, we developed a sensor abstraction mechanism that enables transparent access to local sensors (i.e., sensors residing in the same board) or remote sensors (i.e., sensor residing in the other board). This approach: i) is easily extensible, because adding support for other remote sensors just requires to modify a template component that maps the local queries to remote ones, with the ActiveMessage *id* being used as sensor identifier; ii) is symmetric, i.e., applications residing in Pegasus can query a sensor in Fenix, and vice-versa; iii) opens the way to more advanced sensing, as each board has an MCU (e.g. processed data can also be sent, by implementing feature extraction algorithms).

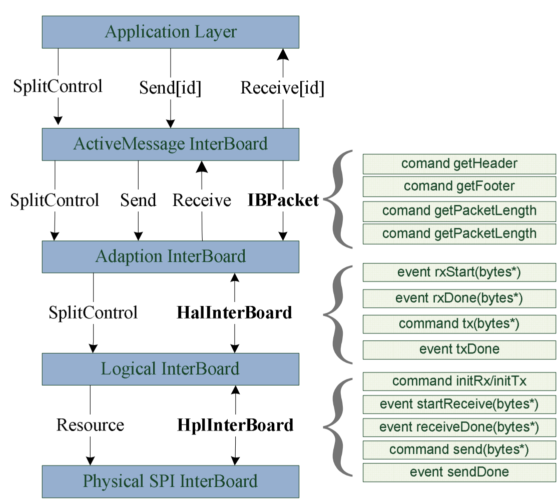

Figure 5. Inter-board communications architecture .

V. Evaluation and Lessons Learned

A. Sensors sampling rate

This section presents movement sensors (i.e. accelerometer and gyroscope) sampling rate evaluation, from the Pegasus perspective. First, the on-board sensors are measured, and then the remotely-accessed accelerometer (in Fenix) is analyzed.

Fig. 6 shows the maximum number of samples per second achieved in three scenarios that differ in processor clock, I2C bit rate, and number of axes sampled simultaneously. The presented processor clock / I2C bit rate are usual combinations. The maximum number of samples when reading one axe, three axes, or six axes (when supporting the MPU-60X0 that enables to simultaneously access gyroscope and accelerometer data) are presented.

The results clearly show that Pegasus can be used to measure movement, independently of the number of axes or selected clocks combination, as it always supports more than 100Hz sampling.

We also assessed the impact of inter-board sensing. Fig.7 shows the maximum sampling rates achieved when directly querying a local 3-axis accelerometer (**Direct**), querying the local sensor but using the inter-board communications mechanism by configuring a single Pegasus board in SPI loopback mode (**P-P**), and accessing the remote accelerometer (**P-F**) (note that both boards use the same accelerometer model).

When analyzing the graphs one should note that the processors clock values are comparable on both nodes, the bit rate values shown apply to the SPI interface between the boards (and were the maximum achieved in the current software version), and the bitrates for the accelerometers (not shown) are according to Fig. 6. From these results we conclude that calling a remote device has a significant impact. Nevertheless, this does not compromise usability because in the worst case (for **P-F**) sampling can be done at 167Hz. We also concluded that the 250Kbps SPI bandwidth is the bottleneck (this can be realized by comparing the performance of **P-P** and **P-F** cases for 8MHz and 16MHz clocks). To further understand what would be a reasonable bandwidth to the SPI bus, we also measured the **P-P** scenario at 16MHz / 1Mbps, achieving 542 reads per second. This shows that for the **P-F** scenario at 16MHz a 1Mbps SPI would be necessary. Based on this observation, the next improvement will be to support an inter-board SPI bandwidth of 1Mbps.

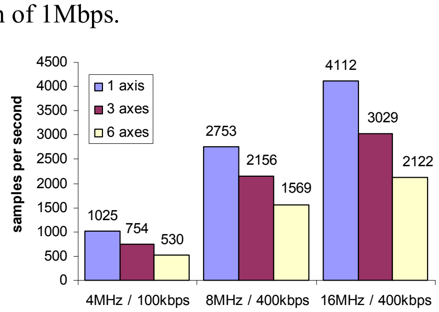

Figure 6. Pegasus I^2C based movement sampling capabilities.

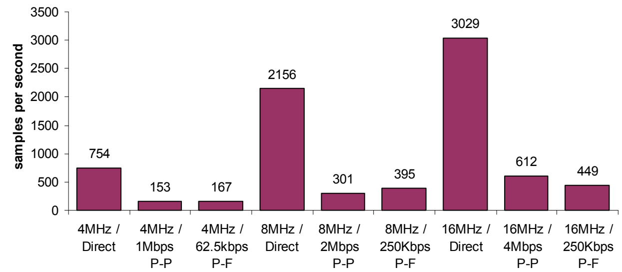

Figure 7. Pegasus sampling capabilities considering sensor location.

B. Communication range

Communication range is impacted by several factors, such as propagation and interference problems that are common in indoor environments. Outdoor environments are simpler, but terrain relief, vegetation, node height and antenna orientation also impact range. Hermes communication range was evaluated on a per-application scenario basis. In the iHorse application scenario, the communication range was measured for the three supported hardware platforms (TMoteSky, IRIS, and Pegasus) and the results are presented in Fig. 8.

The measurements were done outdoor, on an open field with vegetation less than 0.30m high, and inside a stable with 55m diagonal and 18 boxes. The 1.50m node height accounts for a standing horse, and the 0.25m applies when the horse is lying down. The base station node was deployed at a height of 2m. The communication ranges were measured with an application that queries a remote node one time per second. The shown distances are the ones achieved for links in which the percentage of answered queries was more than 95%. The distances inside the stable are for the worst locations (i.e., the ones that had the highest number of concrete box walls between the node at the horse and the base station).

Only the Pegasus node was able to assure one-hop communication irrespectively of the node orientation. In fact, the 55m limitation comes from the stable dimension and not from propagation or communication path issues. Further details and analysis are presented in [2]. The evaluation of Hermes communication capabilities in the context of the Hospital application will be done as future work.

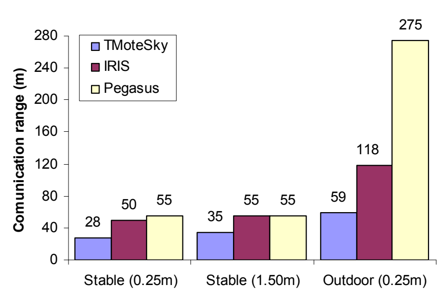

Figure 8. iHorse platforms communication ranges.

C. Filesystem performance

The test goals were to measure the performance of the FatFs filesystem when used in the Pegasus node, in order to understand the platform limitations in what concerns storage, using the available SD cards.

All the tests were done with the MCU running at 16MHz and the SPI bus clock at 4MHz. The sector size was set to 512 bytes, and the filesystem was FAT16 without long file name support.

Fig. 9 shows the write/sink execution times and their distributions, an important aspect not shown in averaged write/sink times. Note that write operations do not necessarily write data to the SD card, as data may simply be cached, as opposed to sink operations, which always cause the data to be written to the card. For each card / operation the minimum, 5% percentile, 95% percentile, and maximum operation times are presented. One can realize that the distribution of operation execution times is very diverse. The reason we decided to analyze these distributions was because the filesystem implements writing as a synchronous operation, having impact on the platform sampling rates.

Fig. 10 shows the read / write throughput for each SD card when used in Pegasus (with an error less than 4%). The procedure was to read/write 10MB of data to a card as fast as possible, in different blocks sizes. The bars marked with a "B" mean that the write was at the end of a file that was already filled with 10MB of data. The results show better reading times when compared with writing times, suggest that if multi-sector operations are supported (for 2048 block operation) results can be improved because there is available bandwidth in the SPI bus, show that writing 100bytes blocks does not impact significantly the throughput, and do not lead to the conclusion that writing to big files degrades the throughput.

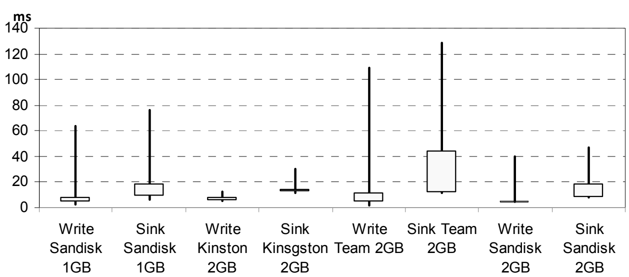

Figure 9. Write and Sink times for a 512 byte block.

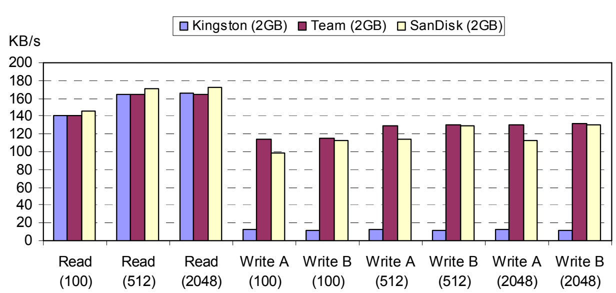

Figure 10. Read and write throughput.

D. Energy consumption

Table 2 presents Pegasus' current consumption measured at the battery (4V at measurements). The mote clock was at 4MHz. When comparing it with Telos consumption, it becomes clear that current consumption during standby and during MCU idle is higher. This was the result of several options that will be changed in the next version, namely the already mentioned PMS low efficiency at very low currents and the lack of protection against USB circuitry consumption when the node is detached. Individual devices also add to current consumption in low power modes (see Table 3 and Table 4).

Telos was designed to run for years with a low duty-cycle. This is not the goal of the Hermes platform, as the requirement was 30 days for the iHorse application and 7 days for the Hospital application. From this point of view, Pegasus' standby consumption during a 30-days period represents 10% of the battery capacity, something we can cope with considering that consumption in other modes are not so different from Telos.

TABLE II. TELOS AND PEGASUS CURRENT CONSUMPTIONS

Hardware state	Telos	Pegasus
Mote Standby	5.1 uA	140 uA
MCU idle, Radio off	54.5 uA	224 uA
MCU active	1.8 mA	2.76 mA
MCU + Radio RX	21.8 mA	19.2 mA
MCU + Radio TX (0 / 12dbm)	19.5 mA / NA	17.3 / 28.2 mA
MCU + Flash/SD Write	15.1 mA	~30 mA

TABLE III. PEGASUS PARTS CURRENT CONSUMPTION

Part	Active (mA)	Sleep (uA)	Notes
MCU	0.5/MHz	1.1	
Radio	34.2 / 16.9	0.2	transmission @12dbm / receiving
Temperature	<0.75	2	
Gyroscope	6.5	5	
Accelerometer	0.14	0.1	active value for max sampling
HeartBeat	0.06	-	can be turned off
Fuel Gauge	0.065	-	
SD card	20 to 100	~100	depend on model, can be off
PMS	-	90	SW1, burst mode, not switching

TABLE IV. FENIX PARTS CURRENT CONSUMPTION

Part	Active (mA)	Sleep (uA)	Notes
MCU	0.4/MHz	22	0.8mA/MIPS
Accelerometer	0.14	0.1	active value for max sampling
RFID	0.0083	0.8	active value for all antennas
GSM	1.6 / 475	62	registered / GPRS transm.
GPS	28 / 32	1.5	active values track. / acquisition

E. Requirements and inovation aspects summary

The evaluation results confirmed that the platform as a whole successfully meets the requirements of the targeted sensing scenarios, independently of sensor location, as presented in Fig. 6 and Fig. 7, with the maximum sampling rates for 3-axis movement sensors well above the values at which other devices operate [14]. The communication capabilities include Wireless Wide Area Networks (WWAN) and Wireless Local Area Networks (WLAN) technologies to support an always connected approach. For the WLAN case, the results presented in Fig. 8 show that the platform enables longer range compared to other offerings like Telos, IRIS and Shimmer. The platform supports several localization mechanisms, including GPS, RFID, WLAN and WWAN, allowing a wide range of choices to fit every use case. The storage subsystem enables to save more than 2 weeks of data (for both movement sensors sampled at 100Hz) using ~1% of the writing throughput, for 512 byte sectors on a SanDisk, as seen in Fig. 10. With the exception of low-power modes, energy consumption is on a par with the very energy-efficient Telos, as can be seen in Table 2. Note that for the intended application scenarios the maximum required node battery lifetime is 30 days, being mostly driven by communications and processing. The node is compact, light and will be provided in a shock resistant, dust and water protected box.

Compared with Shimmer (a well recognized platform used on body sensor networks research), Hermes innovates in the supported sensor set capabilities, in the communications capabilities by not requiring a gateway to be accessible from the Internet, in the localization capabilities by supporting GPS and RFID, and in the processing capabilities by supporting two MCUs (each one with more capabilities than the one in Shimmer). On the other hand, the device is bigger and not as light, mostly because of using a bigger capacity battery and protective casing. This comparison does not account for the fact that Hermes is composed of two modular platforms (Pegasus and Fenix) that can be used separately in a diversity of scenarios, another of its innovative aspects.

Because Hermes is composed of Pegasus and Fenix, which are motes on their own, it was necessary to create software components that enable an easy development of applications that benefit from all the offered functionality. First, the ActiveMessage communication functionality was provided as the communication mechanism between Pegasus and Fenix. Based on this, a sensor device location abstraction mechanism was developed that enables an application developer to use a sensor irrespectively of the platform on which it resides. This functionality allows the development of an application in Pegasus that uses Fenix sensors (as was the case in Fig. 7) or the opposite. But the novelty here is not on using sensors on add-on boards. The novelty is that this mechanism works both ways (i.e. any node can use it) and it is general, meaning that it can be used by other devices like communications, storage, or even processing, opening the way to highly distributed applications.

VI. Conclusions and Future Work

In this paper, the Hermes multi-scenario platform was presented. The distinguishing features of this compact, lightweight platform are its modularity, flexibility, sensor device location abstraction capabilities, multi-technology communication capabilities and extensive sensing functionality. The paper presented Hermes' motivations, requirements, architecture, implementation details and performance data. The latter have shown that the platform perfectly meets the performance requirements of the intended application scenarios, in terms of sampling rate, communication range, storage, and energy consumption.

Despite its very good performance, optimising Hermes' performance will broaden its field of application. With this in mind, further work will evaluate the Hermes performance in the hospital application scenario, improve the performance of the inter-board communication stack, provide the support for more types of remote devices, and minimise energy expenditure.

References

[1] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," In Proceedings of IPSN 2005, IEEE, Piscataway, NJ, USA, pp. 364–369.

[2] A. Rodrigues, J. S. Silva, F. Boavida, and T. Camilo, "iHorse – A WSN-based Equine Monitoring System," SenseAPP 2011 Workshoop, In Proceedings of LCN 2011, IEEE, Bonn, Germany, pp. 1036-1043.

[3] M. McGrath, T. J. Dishongh, "A Common Personal Health Research Platform—SHIMMER and BioMOBIUS," In Intel Technology Journal, 2009, vol. 13(3), pp. 122-147.

[4] P. Dutta, J. Taneja, J. Jeong, X. Jiang, and D. Culler, "A building block approach to sensornet systems," In Proceedings of SenSys 2008, ACM, New York, NY, USA, pp. 267-280.

[5] "XLP PIC," 2011. [Online]. Available: http://www.microchip.com/en_us/technology/xlp/technology.html

[6] "EFM32," 2011. [Online]. Available: http://www.energymicro.com/technology/efm32-microcontrollers-energy-mode-3

[7] "EFR4D Draco," 2011. [Online]. Available: http://cdn.energymicro.com/dl/pdf/EFR_brochure.pdf

[8] "OSIAN," 2011. [Online]. Available: http://www.peoplepowerco.com/OSIAN/

[9] G. Lu, D. De, M. Xu, J. Cao, and W. Song, "TelosW: Enabling Ultra-Low Power Wake-On Sensor Network," In Proceedings of INSS 2010, pp. 211-218.

[10] "GNode 301," 2011. [Online]. Available: http://www.sownet.nl/download/G301Web.pdf

[11] "Zolertia Z1," 2011. [Online]. Available: http://zolertia.com/sites/default/files/Zolertia-Z1-Datasheet.pdf

[12] "GINA," 2011. [Online]. Available: http://openwsn.berkeley.edu/wiki/OpenGinaKit#GINA

[13] "BM innovations," 2011. [Online]. Available: http://www.bm-innovations.com/index.php/ez430-chronos

[14] C.-C Yang, and Y.-L Hsu, "A Review of Accelerometry-Based Wearable Motion Detectors for Physical Activity Monitoring," In Sensors 2010, vol. 10, pp. 7772-7788.

[15] "FatFs," 2011. [Online]. Available: http://elm-chan.org/fsw/ff/00index_e.html

[16] P. Dutta, M. Feldmeier, J. Paradiso, and D. Culler, "Energy Metering for Free: Augmenting Switching Regulators for Real-Time Monitoring," In Proceedings of IPSN 2008, IEEE, pp. 283-294.