A Flexible Framework for Data Exchange and Presentation between Wireless Sensor Networks and Personal Devices

Ruben Miguelez Garcia*, Paulo Carvalhal*, Manuel J. Ferreira*, Luís F. Silva†,
Heitor Almeida†, Cristina Santos* and José A. Afonso*

* University of Minho/Dept. Industrial Electronics, Guimaraes, Portugal, e-mail: *jose.afonso@dei.uminho.pt*
† University of Minho/ Dept. Mechanical Engineering, Guimaraes, Portugal, e-mail: *lffsilva@dem.uminho.pt*

Abstract—This paper presents the framework of an interface between wireless sensor networks and personal devices like PDAs, cellular phones and laptops. It possesses a flexible architecture that can be adapted easily to display any kind of data received from the network sensors using content description. The most relevant properties of this system are: portability (it can be used on any device with a standard browser), expandability (it can act as a gateway for higher systems), adaptability (the information can be presented in many different ways) and scalability (several sensor networks can be controlled at the same time). This system is currently being used display information retrieved from a Bluetooth based wireless sensor network platform which operates onboard of an unmanned aerial vehicle (UAV).

Keywords—web interfaces, wireless sensor networks, Bluetooth.

I. Introduction

Bluetooth technology provides a short range wireless communications platform that allows new designs of wireless sensor systems. Taking advantage of the standard and open nature of Bluetooth communications, a generic method for retrieving and displaying this information was developed. It allows the display, in a standard browser, of the measurements acquired from the sensors in range. The displaying device can also become an access point for that information or transmit it to a higher layer that aggregates the data from several sensor networks.

The range of useful applications for such system is vast. One example is a body sensor network communicating wirelessly with a PDA/cellular phone which stores and processes the data collected from the network and relays the information to the medical center using a wide-area cellular network. Other example is a wireless monitoring system where the readings of a measuring device (e.g. a power meter) are easily acquired and recorded when the PDA approaches the device. A third example is a context based content presentation system (e.g. in an interactive museum) where a user carrying a PDA receives relevant information as he walks through the exhibition.

Our system uses Python as programming language due to his wide platform support and extension capabilities, XML as data transport format to higher integration systems and XML together with CSS for viewing data on a standard browser.

The remaining of the paper is organized as follows. Section II presents an overview of related work. Section III describes the wireless sensor network where the framework presented in this paper is being used. Section IV provides a description of the developed framework and presents some results. Finally, section V presents the conclusions.

II. Related Work

In [1] the authors provide a study of applying wireless sensor networks to habitat monitoring, where they consider the use of PDA-sized devices to interact with the system networks. The PDA would directly communicate with the sensor nodes, providing the user with a fresh set of readings about the environment and allowing the user to interactively control the network parameters by adjusting the sampling rates, power management parameters and other network parameters. The authors expect that this device will be extremely useful during the initial deployment and during retasking of the network.

In [2] the authors describe several remote heart monitoring systems. In one of these systems, by CardioNet, the patient wears a lightweight electrocardiograph (ECG) monitor that continuously transmits the patient's data to a special PDA he carries. The PDA uploads the data to a service center through a cellular network, either periodically, when it detects a heart problem, when the physician requests new data, or when the patient feels symptoms and commands it to transmit.

The idea of using a handheld like a mobile infrastructure access point is introduced in [3]. Likewise, the device running the application could act as a computing server for those small devices that don't have enough computing resources to transform their raw data in useful information. These ideas are also explored in our application.

In [4] the authors present a scenario of use of wireless sensor networks in a medical care environment, where a PDA-like device is used to display all kind of medical information. Our application fits very well in such scenario.

III. Wireless Sensor Network Platform

The wireless sensor network platform described here is being developed in the context of AIVA [5] project. This is a broader project where an unmanned aerial vehicle

(UAV) is being designed and developed [6]. The aircraft implements onboard a Bluetooth based wireless distributed data acquisition and control system [7], which provides a flexible and modular architecture to the UAV.

Traditionally, UAVs present a processing system architecture consisting of one centralized and complex unit, with one or more CPUs, to which the instrumentation devices are connected by wires. At the same time, they have bulky mechanical connections. One of the main innovative aspects of the proposed platform concerns the flexible and distributed architecture of the onboard system, where the connections between the instrumentation and processing units are made using Bluetooth wireless communication technology. This revolutionary approach can be dubbed fly-by-wireless. At the same time, the traditional monolithic processing unit is, in this approach, replaced by several less complex units (nodes), spread out over the airplane. In that way the nodes are placed near the sensors and near the controlled surfaces, creating a network of nodes with the capacity of data acquisition, processing, actuation and communication over a wireless platform.

A. Global Electronics Architecture

The global view of architectural model of the computing and communication system is presented in Fig. 1. It is a multitasking/multiprocessor based system connected by an asynchronous local bus that allows for speed adaptation of different tasks/processors. The system supports: one processing unit for the Bluetooth (BT) piconet Master; one flight controller unit; one data logger and earth link; and one embedded vision system (EVS). In each of these nodes many critical processes are permanently running. Air-ground link communications are based on a classical producer/consumer model that will manage message queues for uplink and downlink data traffic.
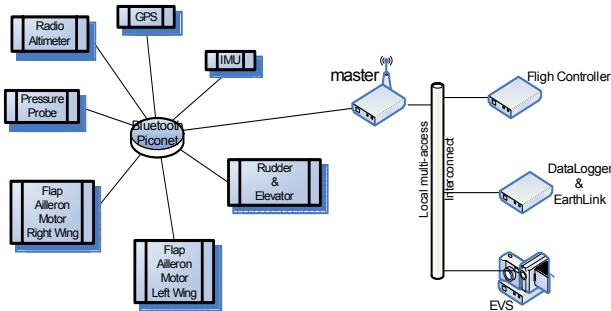

Fig. 1. Global electronics architecture of the AIVA platform.

B. Onboard Bluetooth Piconet Platform

The AIVA platform implements an onboard distributed data acquisition and control system based on Bluetooth wireless network technology (represented by the Bluetooth piconet of Fig. 1) and either on balanced or unbalanced scheduling policies. The BT technology presents some characteristics that suit the envisioned application. It possesses smaller form factor and drains less power than some alternative wireless network technologies like the IEEE 802.11; it uses a medium access control protocol that provides support for real--time traffic; and it provides an adaptive frequency hopping mechanism that allows the improvement of the performance of the communication link by avoiding

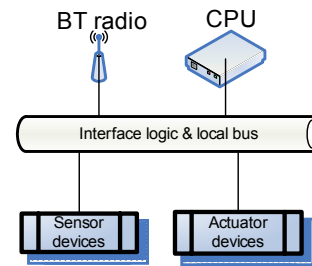interference. Fig. 2 presents the general architecture of the network nodes.

Fig. 2. Architecture of a Bluetooth node.

In order to cope with typical problems that arise in these embedded systems, namely asynchronous events, the technical design approach used to model the system was a set of state machines. They treat easily these problems and also solve other major issues like resolving simultaneous events. The state machines are organized in a two layered architecture: transport and application layers (Fig. 3). The first layer, which is common to every application, deals with network packets. The second layer implements the specific application functionalities. Both have independent design.
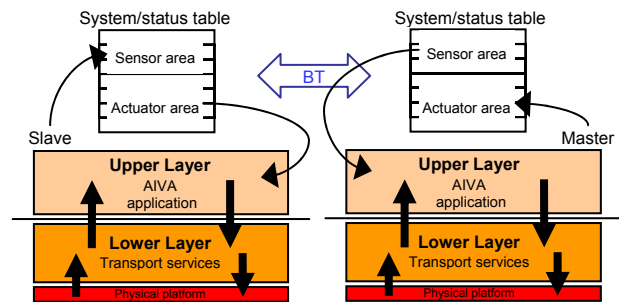

Fig. 3. Logical architecture of nodes and communication strategy.

The Bluetooth modules used are connectBlue OEM serial adapter devices [8]. The master unit is an OEMSPA33i module and the slave units are all OEMSPA13i modules, both including an integrated antenna. We plan to use external antennas in future versions of the platform, to be able to shield the modules in order to increase the reliability of the system against electromagnetic interference.

Fig. 4 shows distribution of the Bluetooth nodes on the aircraft structure. The master module (MM) is placed at the fuselage body, and as described earlier acts as the network and flight controller, onboard data logger, and communications controller for the link with the ground station. The other six nodes (Sensing & Actuation Modules -SAM) are slaves in the network. On each wing, there is a node for an electrical propulsion motor and for control surfaces like ailerons and flaps. Motor speed control, feedback loop, and operating temperature monitoring are responsibility of these wing nodes, as well as control surfaces actuation and position feedback. At the tail, there is another module for elevator and rudder control, and position feedback. In the fuselage body, we have another two additional SAM type nodes, one for GPS and the other for electronic gyroscopes, which assure some basic information assessment for navigational purposes.
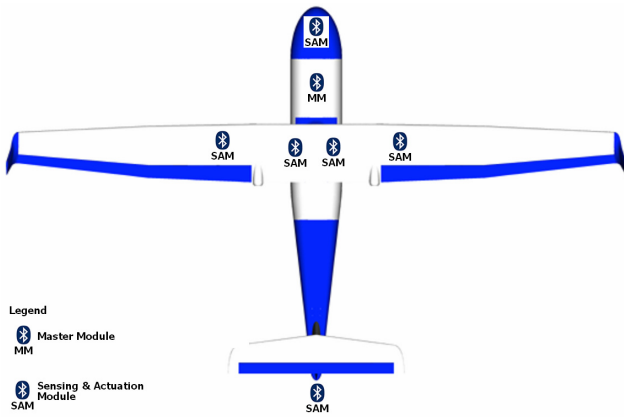
Fig. 4. Distribution of the Bluetooth nodes on the aircraft structure.

At the front node is the nose probe, which consists of a proprietary design based on six independent pressure sensors that give valuable digital information for flight control. Also in this node the information from an ultrasonic probe provides support for the automatic take-off and landing system. Fig. 5 displays the physical layout of the nose probe node. The Bluetooth module is in the lower corner, while the microcontroller is on the left hand side and the sensor hardware on the right hand side of the board.
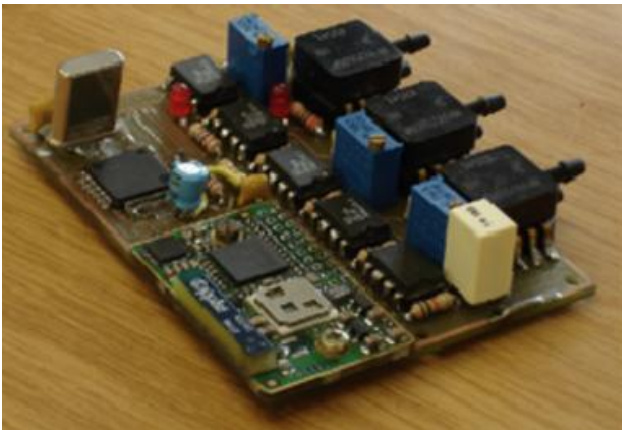

Fig. 5. Physical layout of the network node used as the nose probe.

IV. DEVELOPED FRAMEWORK

The development of the application described here was made taking in consideration the following design goals:

- The data can be show on a standard browser.
- It must be platform independent.
- Heterogeneous information should be displayed in an appropriate way according to its nature.
- It should be scalable.

On a first approach to the solution, handheld devices with screen and processing capabilities smaller than a PDA were rejected as display devices because of the complex nature and wide range of objects to display. So our application is targeted mainly for use on laptops and PDAs with Bluetooth.

Our system uses Python as programming language due to his wide platform support and extension capabilities. Python is an interpreted, interactive programming language created by Guido van Rossum in 1990. Python is fully dynamically typed and uses automatic memory management; it is thus similar to Tcl, Perl, Scheme, and Ruby.

As data transport format to higher integration systems, our system uses XML (Extensible Markup Language), a W3C (World Wide Web Consortium) standard that allows information and services to be encoded with meaningful structure and semantics that computers and humans can understand. XML presents very powerful characteristics for information exchange, and can easily be extended to include user specified and industry specified tags.

Our system uses XML together with CSS (Cascading Style Sheet) for viewing data on a standard browser. CSS is a style-sheet language used to describe the presentation of a document written in a markup language like HTML, XML or SVG (Scalable Vector Graphics). The CSS specifications are maintained by the W3C.

The use of a browser as user interface facilitates the portability of the application. The modification of the layout and appearance of the displayed data can be performed with an easy alteration of the CSS style-sheet. The core application can also be used as an access point to the information delivered by the Bluetooth sensor network through the use of his embedded web server.

Since information is transported on XML format, scalability of sensor network monitoring as well as integration with web services can be achieved easily. The structure of the XML file is a simple translation of the network structure logic: the piconet is composed of one master and up to 7 slaves and each slave transmits data from one or more sensors. The data received from the sensors can be transformed before it is written down on the XML structure and some extra information is also added in order to confer meaning to that raw data. The transformations and meaning of information must be configured into the application before operation.

The structure of the application is represented in Fig. 6. Firstly it establishes a connection with the master node of the piconet. From that moment, it is able to receive and send packets to the components on the piconet. Every time a new packet arrives, the information is extracted according with the configuration and transformed as specified. In parallel, another task takes periodically the information on the data binder and transforms it to XML format, while the embedded web server attends the requests serving the XML and CSS files together with the images if they exist.
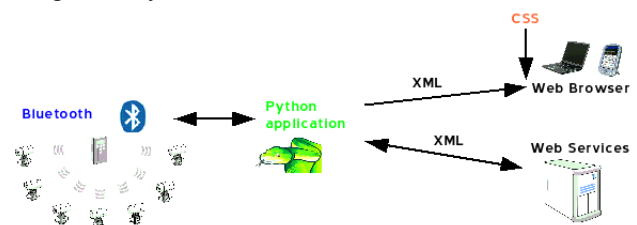

Fig. 6. General application structure.

A. Main Tasks

The application core is compound of several threads or tasks that run in parallel. Each one carries out its task with a defined frequency or on demand. In Fig. 7, a scheme of interactions between all the parts of the application is presented.
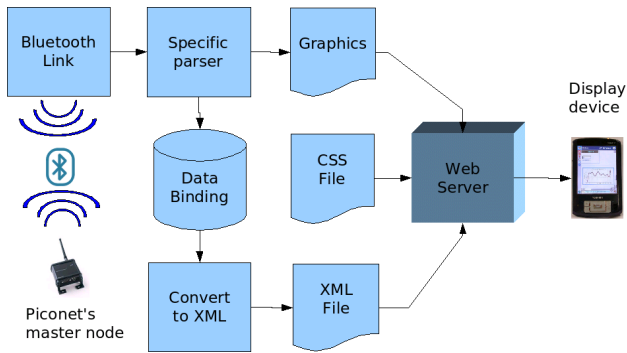
Fig. 7. Interactions between the modules.

1) Web Server: This task is a simple and tiny HTTP server embedded on the application, which serves the files that are requested to it. To increase the speed and reduce the I/O operations on disk, the XML file is kept on memory all the time during operation. The web server is implemented though the use of the package BaseHTTPServer included in the python base library.

2) XML Converter: This is a periodic task that, with a user-defined frequency, takes the data binding with the updated piconet information and converts it to XML together with some headers that will permit to display it on a standard browser. That conversion is done easily using the package generateDS, which provides an export to XML method to the classes generated by it.

3) Bluetooth Link: This task manages the communication link with the piconet, specifically with its master node. Every packet received is sent to the specific application parser which will extract the information and write it on the data binder. For the Bluetooth communications we use the BlueZ Bluetooth stack.

4) Packet Parser: This is a parser that must be configured for every new environment where the application is going to be used. It depends of how the data is transported on the packets. The parser receives the raw packets and after extraction and manipulation of the data, it writes them down on the piconet data binder.

The actual data, together with the previous ones, can be used to create graphics that will be linked from the XML file.

B. XML File Structure

The XML file with the piconet structure and data is the main output of the application and also the configuration file that is used as input.

On an empty structure like the one presented in Fig. 8, the user should write some basic information, although more information can be passed in order to have a richer output. Then, after the application establishes connection with the piconet master, the fields inside <info_slave> are updated with the acquired information. These fields contain basically the data and its type, the units and the data description, and it is also possible to add the graphics generated by the packets parser.

C. Results

Our experiments demonstrate that the use of a browser as user interface provides a convenient way to display the information. The appearance of data on the PDA browser (Fig. 9) is similar of that in the PC browser, but adapted to the smaller screen width, without requiring any modification on the XML or CSS files. For this test we used a Mozilla Minimo browser on the PDA and several free browsers on the PC platform (Opera, Mozilla, Firefox, Epiphany).

```
<piconet>
    <name></name>
    <location></location>
    <description></description>
    <master>
        <name></name>
        <address></address>
        <description></description>
        <location></location>
    </master>
    <slave>
        <name></name>
        <address></address>
        <description></description>
        <location></location>
        <port></port>
        <info_slave>

<data_description></data_description>
        <data dtype=""></data>
        <data_units></data_units>
        <a href="">
            <img src="" width=""
height="" alt="">
            </img>
        </a>
        </info_slave>
    </slave>
</piconet>
```

Fig. 8. Example of the XML file structure

The developed system can be used to create complex graphics displayed along with the last data received. The appearance of the information on the screen can be easily modified by changing the associated CSS style-sheet.

The embedded web server permits to see the results on the local device or from any other device when both devices are connected to the same network, e.g. Internet, which is appropriate to share instantly the processed information.

The configuration file and output of the application is a XML human readable file than can be edited easily and used by other applications. The link with the piconet is done knowing only the address of its master node, but if it is unknown, a discovery search can be performed.
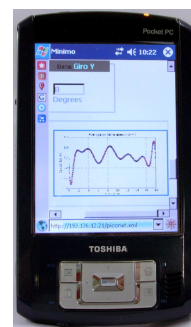

Fig. 9. Example of the application's web interface on a PDA.

V. CONCLUSION

In this paper a simple and adaptable method for information retrieval and displaying for Bluetooth wireless sensor networks has been proposed and described.

Its requirements are small enough to permit portability and its flexible design makes it adaptable to

heterogeneous data environments. The design approach keeps doors open in order to allow the introduction of new features.

Future work includes the use of forms on the web interface in order to allow the configuration of wireless sensor network parameters through the interface using the PDA.

REFERENCES

[1] A. Mainwaring et al., "Wireless Sensor Networks for Habitat Monitoring", *ACM International Workshop on Wireless Sensor Networks WSNA´02*, Atlanta, Georgia, USA, September 2002.

[2] P.E. Ross, "Managing care through the air", *IEEE Spectrum*, Vol. 41, Iss. 12, pp. 26-31, December 2004.

[3] F. Siegemund, C. Floerkemeier and H. Vogt, "The value of handhelds in smart environments", *Personal Ubiquitous Computing*, Vol. 9, Iss. 2, March 2005.

[4] V. Shnayder, B. Chen, K. Lorincz, T. R. Jones and M. Welsh, "Sensor networks for medical care." *3rd International Conference on Embedded Networked Sensor Systems*, San Diego, California, USA, November 2005.

[5] AIVA (Intelligent Aircraft with Artificial Vision), http://aiva.dei.uminho.pt/.

[6] E.T. Coelho, R. Macedo, P. Carvalhal, J.A. Afonso, L.F. Silva, H. Almeida, M.J. Ferreira and C. Santos, "A Fly-by-Wireless UAV Platform Based on a Flexible and Distributed System Architecture." *IEEE International Conference on Industrial Technology - ICIT* 2006, Mumbai, India, December 2006.

[7] E.T. Coelho, P. Carvalhal, M.J. Ferreira, L.F. Silva, H. Almeida, C. Santos and J.A. Afonso "A Bluetooth-based Wireless Distributed Data Acquisition and Control System." *IEEE International Conference on Robotics and Biomimetics - ROBIO 2006*, Kunming, China, December 2006.

[8] connectBlue, "Serial Port Adapter – 2nd Generation", User Manual, 2003, available at http://www.connectblue.com/.