I3ASensorBed: a testbed for wireless sensor networks

Antonio M. Ortiz, Fernando Royo, Raúl Galindo
and Teresa Olivares

Albacete Research Institute of Informatics
Campus Universitario, s/n
02071 Albacete, Spain
email: {*amanuel, froyo, raul, teresa*}*@dsi.uclm.es*

1 Introduction

After several years researching on WSNs, our group decided to implement its own testbed with the objective of analyse, monitor and test the ongoing hardware and software developments.

In the past, we focused on the use of simulators such as Omnet++ [1], the Avrora [2] emulator and real nodes like MicaZ [3], TelosB, TmoteSky and Iris, in order to test and evaluate the protocols and applications that we had developed. Normally, after simulated and emulated experiments, the trend was to implement the proposals in real hardware to compare the results obtained previously. Some examples of this way of work are the IntellBuilding network [4] and the Wisevine project [5]. These networks had several problems such as battery replacement and node reprogramming, that were tedious tasks. It was necessary for us to have a platform composed by real nodes, wired and able to be configured and updated in order to really test protocols and applications without the problems caused by battery exhaustion and one-by-one node reconfiguration.

The I3ASensorBed arises with the objective of providing a reliable platform were to test new protocols and applications in a controlled fashion.

Next, the testbed architecture as well as the main components of the system and the user interface are detailed.

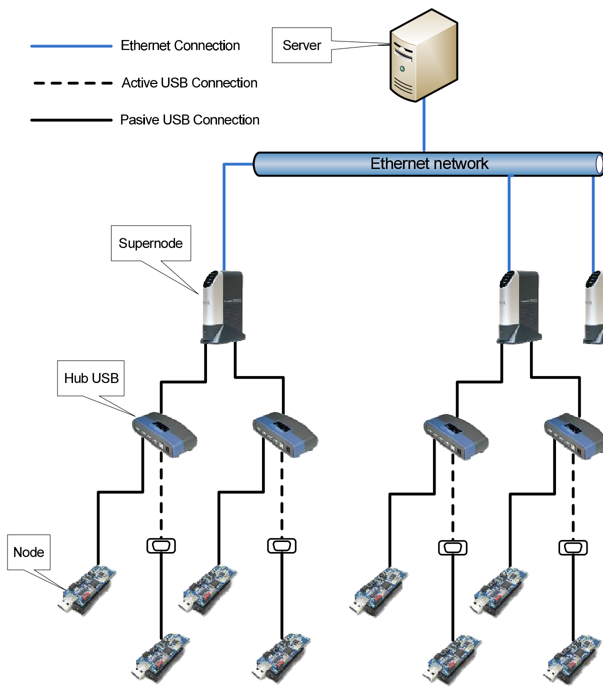

Figure 1: I3ASensorBed Architecture

2 System Architecture

The components of the system are organized in a hierarchy forming a three-level tree, where the interconnections between level use different communication technologies. Sensor nodes are the leaves of the tree; they are connected to a certain point named socket, by using USB connections, to a special nodes named *supernodes*. Finally, the central server acts as tree root where supernodes are connected via Ethernet. This server also houses the database where all the testbed data is stored.

Between supernodes and nodes, USB hubs are installed in order to allow the connection of several nodes to each supernode, taking advantage of the multiple connection availability of the USB technology, and also reducing the number of necessary supernodes. Similarly, between supernodes and Ethernet wiring, we have located an Ethernet switch to be able to connect several supernodes to the central server. The full system architecture is detailed in Fig. 1

3 Components

Next, all the components that form the I3aSensorBed architecture are detailed.

3.1 Sensor nodes

Sensor nodes are the final elements of the system. They are located according with the data acquisition needs. As commented above, there exists different types of sensor nodes, with different technologies and with special methods to program them. Old models such as Mica2 and MicaZ require some external hardware to allow data access and reprogramming, using a gateway through RS-232 connection. The deployment of such technology would require the acquisition of proprietary hardware, specifically, one gateway per node. With the recent technological advances, new node models incorporates USB connection that provides easy connectivity.

At present, there exists several node models that incorporate USB ports, allowing so the direct connection to a common PC. This kind of nodes was selected to be included in the I3ASensorBed, since it is also very easy to connect several nodes to the same USB port by using USB hubs. Other remarkable feature of these nodes is that they are able to take energy from the USB, thus solving the problem of energy management.

The most popular USB-based nodes are TelosB from Crossbow and Tmote-Sky from Sentilla. There also exists a new model manufactured by Maxfor, named MTM-CM5000-MSP (see Fig. 2), that is compatible with TelosB and TmoteSky, since it incorporates the same microcontroller (MSP430) and radio chip (CC2420). This Maxfor node can also be programmed by using the well known operating system TinyOS.

3.2 Sockets and USB wiring

In the I3ASensorBed, a socket is a point where the USB node interface is connected to the USB infrastructure. Each one of the USB connections in the testbed has a unique identifier and it is located in a known geographic position. Thus, in the testbed, a node will be associated to a certain socket, so knowing the geographical position of every node in the network.

Each socket is connected to a USB hub using active or pasive USB cable, depending on the distance from the socket to the USB hub.

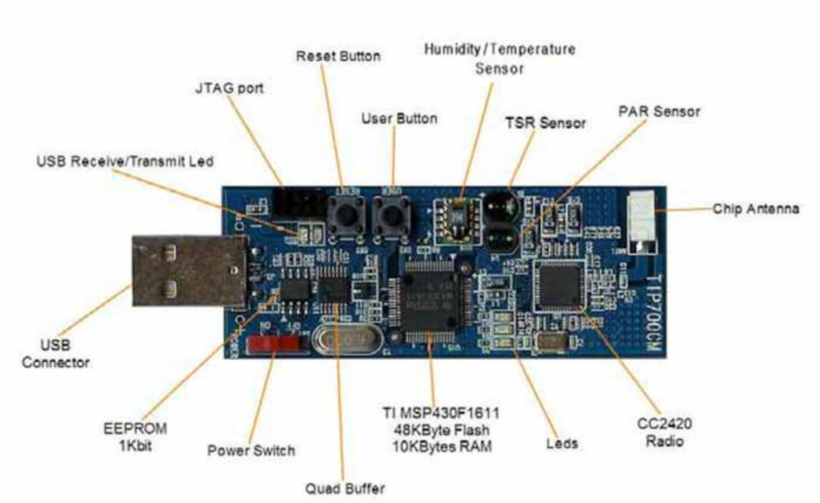

Figure 2: Maxfor node based on TelosB

3.3 USB hubs

USB hubs are multiplexing devices that allows multiple connections to each USB supernode port. The possibility of controlling the on/off of each port in the USB hub allows individual node management, enabling or disabling nodes as needed.

3.4 Supernodes

With the aim of extending the possibilities of USB connections (127 devices and maximum distance of 30m), it is necessary to use some other connection technology. The most common network connection is Ethernet with UTP wiring and RJ45 connectors. The major problem is finding a device able to switch between USB and Ethernet.

For that interconnection, we have used Linksys Network Storage Link for USB 2.0 (NSLU2), that is shown in Fig. 3. NSLU2 is equipped with an Ethernet port and two USB connectors. It uses an Intel IXP420 processor at 133 MHz, it has 32MB SDRAm and 8MB of flash memory. The firmware of these devices have been modified in order to obtain full functional devices that allow the medium change between USB and Ethernet.

3.5 Server

The server is the element that controls the full system, directly interacting with the supernodes by using the Ethernet connection, and offering external

Figure 3: Supernode Linksys Network Storage Link for USB 2.0 (NSLU2)

access through Internet connection. It offers networks services such as DHCP, DNS, NTP and NFS, that are necessary for the correct operation of the supernodes.

On the other hand, the server also houses a database to store all the information of the testbed, it is, sensor nodes, hubs, supernodes, node placement and node data. This database is managed by PostgreSQL [6], a open source, robust and stable database system.

The last function of the server is to execute (by using SSH connections), the Python scripts that allow the interaction between nodes and supernodes such as node programming or data collection.

3.6 Web Interface

Users access the testbed by using a web interface (see Fig. 4). It allows an easy access while it also provides remote connection, in order to be able to use the testbed from everywhere. Using the web interface, users can program, reset, switch on/off nodes, etc.

The web interface allows user task management, and data coming from testbed nodes are stored in the database where users can access once they are ready.

The main features that offers the web interface are:

- **Node selection**: users can select those nodes in which they can execute their applications.

- **Node programming**: nodes can be programmed individually, or as a whole, providing to the user the possibility of different nodes executing different applications. This is very useful when nodes are equipped with several sensors and the applications only use some of them.

- **Task schedule**: the system allows the selection of the time schedule for application execution. For example, it can be necessary to take data only in the morning or only working days and not at the weekend, etc.

4 Node deployment

The I3ASensorBed is composed by 43 nodes deployed in the first floor of the Albacete Research Institute of Informatics [7]. Those nodes incorporate temperature, humidity, CO_2, presence, door and window state (open/close), smoke and energy consumption sensors. In order to accommodate all these nodes it have been necessary the use of 12 USB hubs and 6 supernodes. Figure 5 shows node and sensor location in the building map.

Acknowledgements

This work was supported by the Spanish MEC and MICINN, as well as European Commission FEDER funds, under Grants CSD2006-00046 and TIN2009-14475-C04, and the Regional Council of Castilla-La Mancha under Ecosense project POII11-0118-3014.

References

[1] The Omnet++ Simulator. http://www.omnetpp.org/, 2011.

[2] Avrora, the AVR Simulator and Analysis Framework. http://compilers.cs.ucla.edu/avrora/, 2011.

[3] MicaZ, Wireless Measurement System Datasheet. http://www.openautomation.net/uploadsproductos/micaz-datasheet.pdf, 2011.

[4] F. Royo L. Orozco-Barbosa V. Lpez-Camacho P. Pedron T. Olivares, J. C. Castillo. La red sensor intellbuilding. In *Proceedings of the XVII Jornadas de Paralelismo*, 2006.

[5] V. Lpez-Camacho P. Pedron T. Olivares, L. Orozco-Barbosa. Wisevine: Wireless sensor network applied to vineyards. In *Proceedings of the RealWSN*, 2006.

[6] PostgreSQL project website. http://www.postgresql.org, 2011.

[7] Albacete Research Institute of Informatics. http://www.i3a.uclm.es.

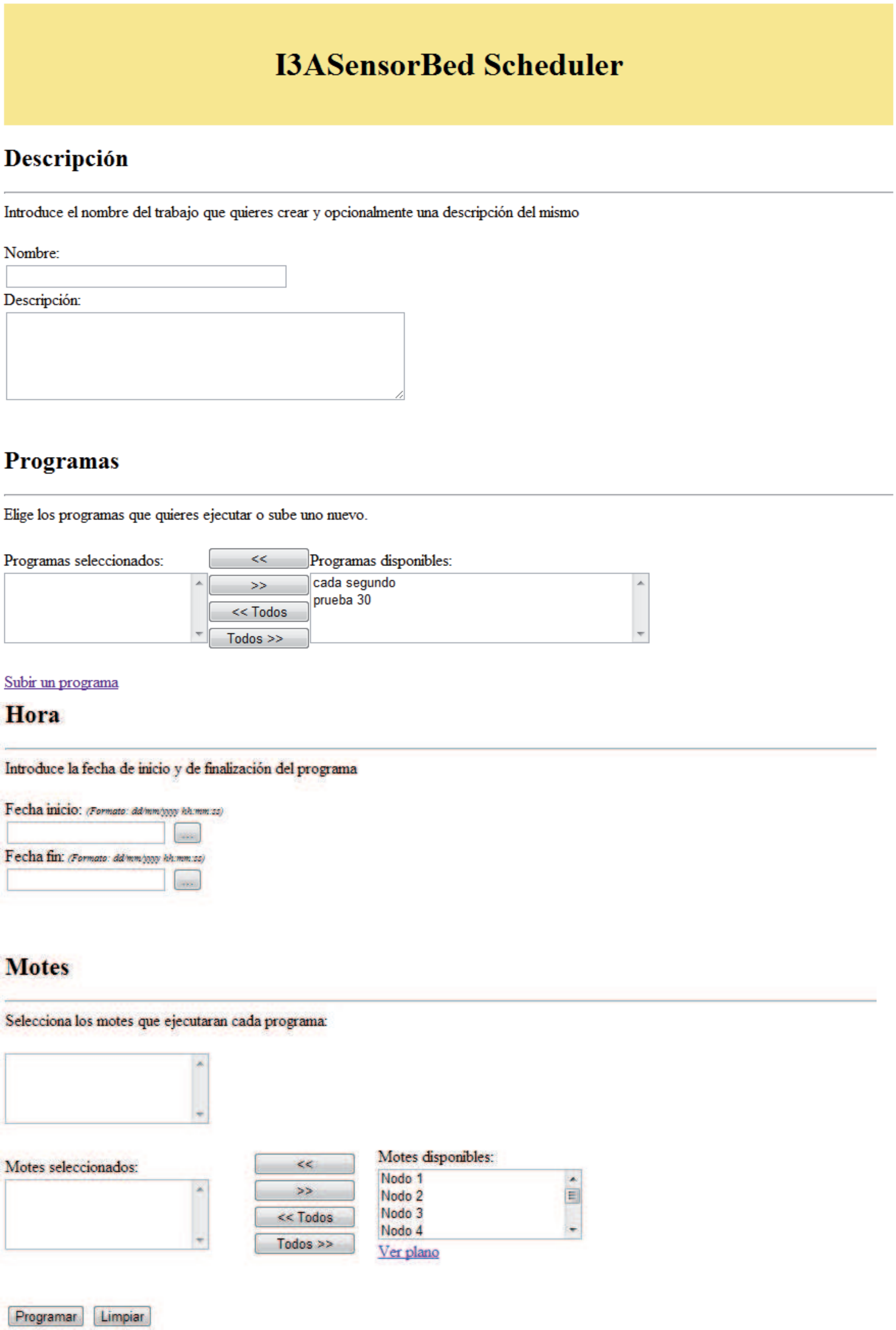

Figure 4: Task programming interface

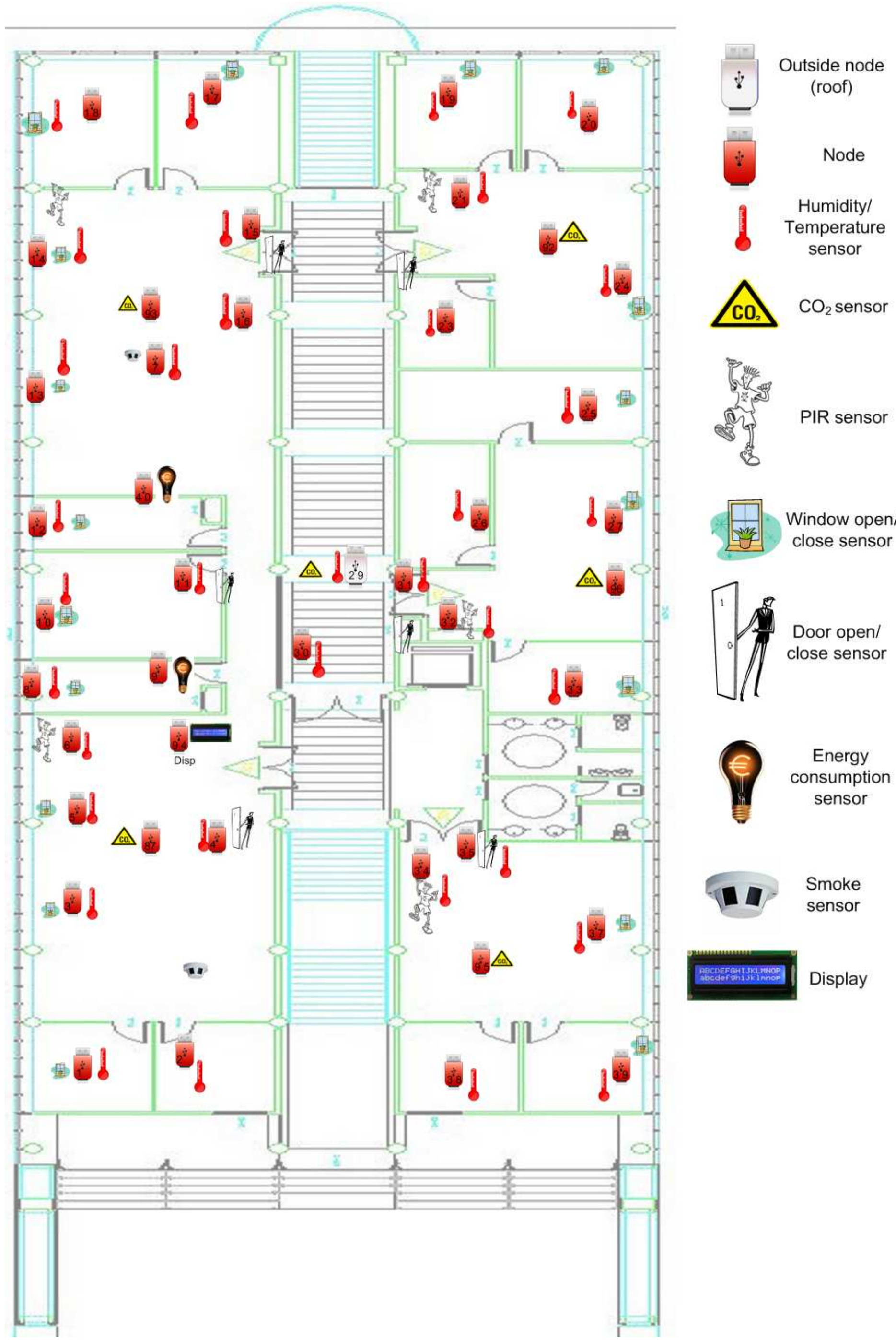

Figure 5: Node location and capabilities in the I3ASensorBed