GENERIC ARCHITECTURE DESIGNED FOR BIOMEDICAL EMBEDDED SYSTEMS

L. Sousa, [1] M. Piedade, [1] J. Germano, [1] T. Almeida, [1] P. Lopes[1],
F. Cardoso, [2] and P. Freitas [2]

[1]*INESC-ID/IST, TULisbon*
1000-029 Lisboa
Portugal
{las,msp,jahg,tmma,paulo.c.lopes}@inesc-id.pt

[2]*INESC MN/IST, TULisbon*
1000-029 Lisboa
Portugal
{fcardoso,pfreitas}@inesc-mn.pt

Abstract Embedded Systems assume an increasing importance in biomedical applications such as clinical analysis and patient monitoring. The lack of generic architectures make the design of this type of autonomous embedded systems a cumbersome and expensive task. This paper proposes a generic architecture for developing biomedical embedded systems, considering both the hardware and the software layers. A prototype of one of these systems for biomolecular recognition, based on magnetoresistive sensors to detect magnetic markers, was already implemented by off-the-shelf components usage. Experimental results show the effectiveness of the proposed architecture and the advantage of its application to develop distributed biomedical embedded systems.

Keywords: Embedded systems, biomedical applications, computing architectures, autonomous communication systems

1. Introduction

In the last few years there has been a crescent interest on embedded systems for biomedical applications, increasing the demand on computing and communication while, at the same time, maintaining the necessity of a portable and autonomous system. Applications such as biochemical operations for clinical analysis (e.g, glucose/lactate analysis), DNA analysis and proteomics analysis for clinical diagnostics [Piedade et al., 2006] and real-time pervasive patient monitoring [Jovanov et al., 2001] [Halteren et al., 2004] are typical examples where high computing and communication requirements must be effective.

Portability and computing power are requirements that lead to integration of wireless devices on embedded systems in order to communicate with general purpose computing systems and also with the setup of distributed computing platforms based on all these computing engines.

Actual embedded systems for biomedical applications have to be equipped with low power communication systems and, on the other hand, have to be easily integrated with more general distributed computing platforms where reliability and security issues have to be considered, both at the computation and the communication levels. The enormous diversity of sensors and medical apparatus demand the development of general computation/communication architectures for deploying distributed embedded systems that may cover a wide range of applications and environments.

This paper proposes a communication architecture for implementing a distributed platform that supports autonomous embedded systems for medical applications. The considered architecture includes both the hardware and software components and allows the development of autonomous but collaborative embedded systems through actual technologies usage. Moreover, the paper presents the application of the proposed architecture for developing a handheld microsystem for biomolecular recognition, based on an integrated magnetoresistive biochip. Experimental results obtained with a system prototype show that the proposed architecture is flexible and may be directly applied in designing embedded systems for several different applications and that it can be easily adjusted to fulfil all particular requirements of each biomedical experience.

2. Proposed Architecture

Figure 1 presents the block diagram of the hardware component of the proposed architecture. In the core of the Autonomous Communication Platform (ACP) there is a communication manager, which is responsible for communicating data and commands from and to a local Acquisition/Processing Platform (APP). Each pair of these two platforms are tightly coupled through a Serial Peripheral Interface (SPI) with a simple protocol but with large a bandwidth. Together they compose an embedded system which communicates with more general computing devices, here designated as master devices, such as a laptop or a Personal Digital Assistant (PDA), through communication modules. These communication modules implement all the necessary protocols and provide electrical interface for serial yet standard protocols, wire-based or wireless, such as the Universal Serial Bus (USB), Bluetooth or Wi-Fi.

As depicted in fig. 1, multiple portable communication/processing platforms may be connected to a single master by using the capacity of the communication standards to set up small networks, e.g., the Bluetooth that uses a radio-

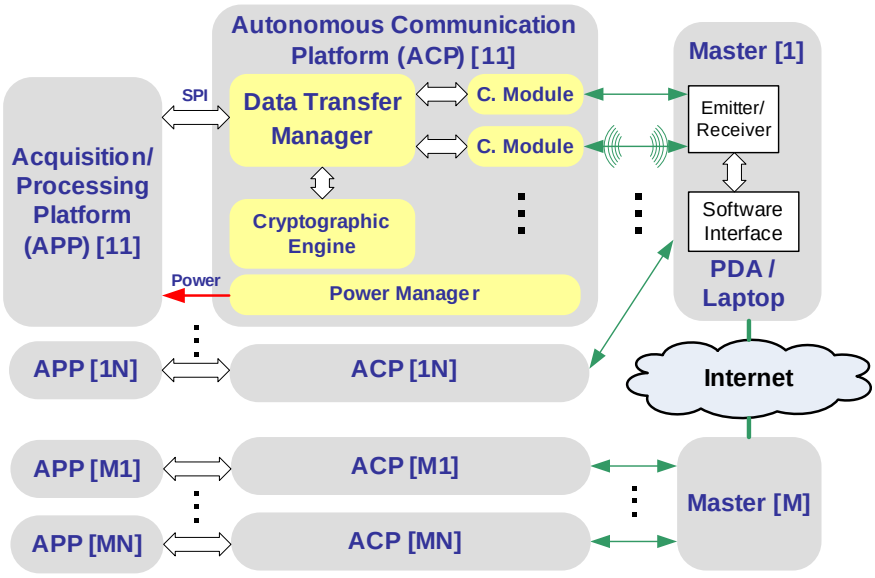

Figure 1. Block diagram of an implementation of the proposed architecture.

based link to establish a connection whenever devices come within 10 meters of each other. Furthermore, masters act themselves as a second communication layer by directly using the IEEE 802.11 standard for wireless Local Area Networks (LANs). We propose the usage of the HyperText Transfer Protocol (HTTP) and WebServices in order to develop a distributed environment in which different masters can be relatively far away and connected by a Wide Area Network (WAN). For security reasons it is advisable to adopt an implementation of the HTTP on the top of Secure Sockets Layer (SSL) or Transport Layer Security (TLS) leading to HTTPS.

Two additional important blocks are present in the ACP: the power manager and the cryptographic engine. Since the platform is autonomous it has to be equipped with a battery. The power manager is responsible for monitoring the state of this battery and for providing its recharge whenever it is necessary. When using some buses, such as the USB, the controller is also able to provide electrical power to other devices. If such is the case, this electric energy can also be supplied to the acquisition/processing platform, which usually also has to be autonomous.

The cryptographic engine present in the ACP assures privacy and integrity during data transfer to the master. In the particular case of this application, a public-key or a symmetric cryptosystem can be applied. Examples of cryptographic algorithms that may be applied are the Elliptic Curve Cryptography (ECC) and the Advanced Encryption Standard (AES) [Schneier, 1995], which

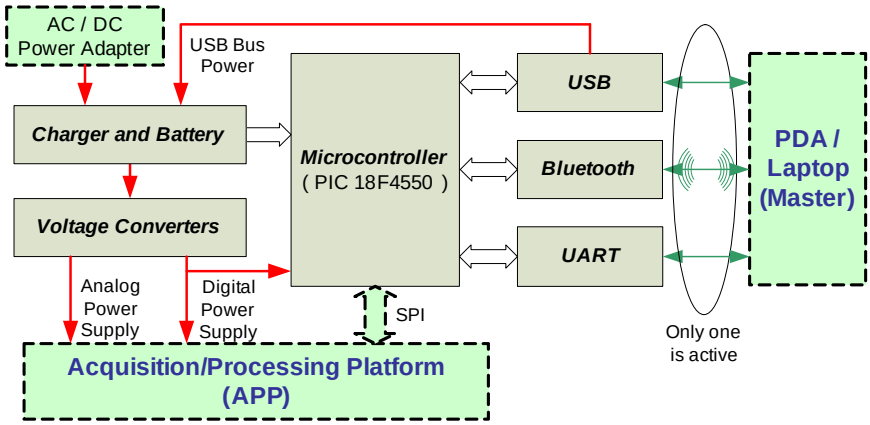

Figure 2. Block diagram of the hardware component of the proposed architecture.

require small key sizes and allow high strength per bit. For example for the ECC, there are also available some optimized implementations for portable and autonomous devices with modest computational capacity [Weimerskirch et al., 2001]. Other encryption algorithms that require less computation resources can also be considered for the communication platform if extremely low power operation is required.

Figure 2 presents a block diagram of an actual implementation of the autonomous and portable communication platform. At the core of the platform there is an off-the-shelf microcontroller (PIC) that integrates a USB stack and transceiver [Microchip, 2004]. This PIC manages all communications with the ACP through a SPI simple protocol with a transfer rate up to 30 Mbps. This interface is used to transfer data and commands between the acquisition and processing platforms. The PIC also provides the universal asynchronous receiver/transmitter protocol which can be directly used to connect the platform or may be used as an interface to feed data to a Bluetooth communication module (e.g. [Bluegiga Tech., 2006]). These different communication modules allow communicating at different rates and distances, for example 12 Mbps for USB while 2 Mbps are achieved to communicate up to 10 m with the Bluetooth. As it is depicted in the diagram, the USB can also supply electric energy to both the communication and the acquisition/processing platforms.

Communication with the APP is performed using a simple protocol that only uses two types of data packets. First, using a fixed length packet, a command is send followed by the number of data values to be transmitted. Then the receiver interprets the command and send an acknowledge signal. Finally, using a variable length packet, the sender transmits the specified number of data values. Commands can also be divided into two classes: configuration and ac-

quisition. Although, the packet structure is the same, the acquisition command only generates an additional variable length packet sent by APP. The communication between the master and the ACP also uses a similar protocol, but the acknowledge signal is implemented in a different manner. The PIC is also responsible to implement the encryption algorithm when secure data transfer to the master is required. Whenever the master enables this feature data packets sent by the APP are encrypted before being sent to the master.

The power manager block is composed by battery charge circuits with voltage sensing that can draw energy provided from the USB master or, alternatively, it gets the energy from an external power supplier. The sensing circuit is used for applying low power consumption techniques at the software level and also by dynamically adjusting the microcontroller operating frequency. Supply voltages required by the platform components are obtained through high efficient switched voltage converters usage.

The software of the communication platform was mostly written in C, but some of the critical parts were coded in assembly language. The encryption engine is an optional software module that encrypts messages before sending them to the master. This is one of the modules that can be switched off by the power manager. To develop the software needed by the masters, the Microsoft operating system and tools were adopted. Classes were programmed in C++ for decryption, communication and general user interfacing, by using the Visual Studio environment. Communication between masters is made by exchanging requests or replies based on the Simple Object Access Protocol (SOAP) via WebServices. WebServices provide a request acceptance and a reply service by using Extensible Markup Language (XML) based files. When a master needs to send a request, request acceptance services only have to write the request XML file to a given directory in the server machine and to return a request ID. Reply services simply return the content of a XML file related to the request ID, if it exists on a reply directory of the server machine. Therefore, masters just have to exchange request/reply data.

3. Architecture Application to an Embedded Microsystem for Biological Analysis

The architecture described in the previous section was applied to develop an embedded hand-held microsystem for biomolecular recognition assays, using target biomolecules marked with magnetic particles.

Labelled targets are recognized by biomolecular probes immobilized on the surface of the chip over sensing sites and the markers fringe magnetic fields are then detected by magnetic sensors [Piedade et al., 2006]. The main advantage of this type of microsystem is the possibility to directly detect biomolecular recognition (e.g. DNA hybridization) by reading the magnetic field created by

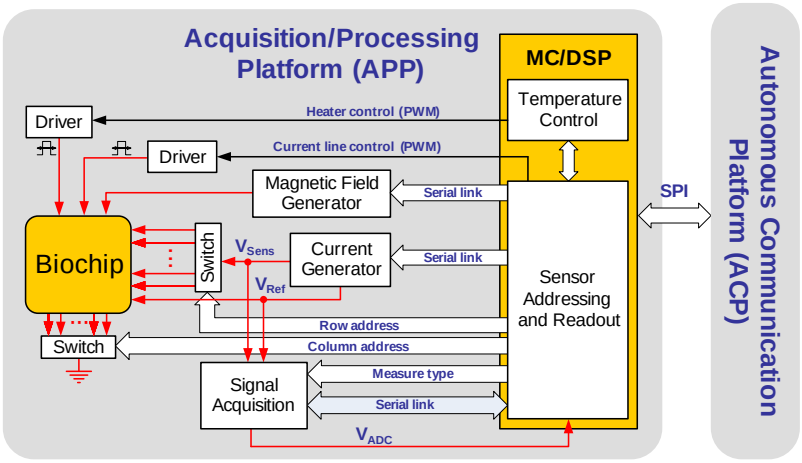

Figure 3. Block diagram of the acquisition/processing platform of the microsystem for biological analysis.

the markers using a sensor located below each probe site. The action of taking an electrical measurement, instead of an optical one, reduces considerably the readout system complexity and increases sensitivity.

Figure 3 presents the Acquisition/Processing Platform of the microsystem for biological analysis, which is connected to the ACP through a SPI. It generates all electric signals to drive the sensor array and to individually address and readout signals provided by each of the sensors. Moreover, it is also in charge of individually measuring and controlling the temperature in subsections of the biochip, by using both the biochip Current Lines (CL)s and by taking advantage of the Thin Film Diode (TFD) voltage-temperature characteristic [Piedade et al., 2006].

Sensor addressing is based on a commutating matrix of integrated TFDs, each one acting as a temperature sensor and as a switch in series with a Magnetoresistive Tunnel Junction (MTJ) which is the magnetoresistive sensor. The microcontroller provides row/column addresses to sensor reading and define the drive current needed through a digital-to-analog converter (DAC). This allows the usage of a single DAC and a single instrumentation amplifier to drive and to read all the sensors in the matricial array. These are the only analog circuits, since control and signal processing are performed by digital processors associated to a 1 Mbit memory for storing all acquired and processed data.

The TFDs, acting as temperature sensors, are calibrated by programming the digital processor to generate current pulses modulated in width (PWM). The calibration is performed in DC, by amplifying a voltage at the terminals of the serial circuit in each sensing site. This calibration phase is performed at setup

time in order to experimentally extract TFDs parameters that allows voltage-temperature characterization. Calibration tables are filled for each sensor with absolute and differential voltages measured using reference sensors available on the biochip. To measure MTJs resistance variation due to magnetic field variation, an AC excitation is performed, using an external magnetic field generated by a coil placed below the biochip. The generation of this magnetic field is digitally controlled. This AC analysis allows the measurement of small relative resistance variations (less than 1%) by using differential mode of amplification. The necessary reference signal can be generated by a microcontroller or registered from the sensors themselves in specific operating conditions.

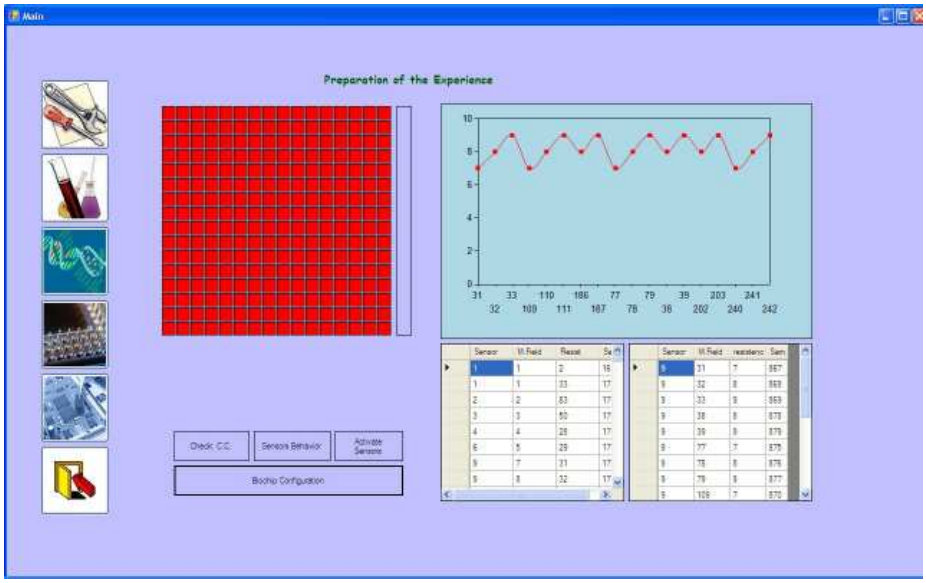

Figure 4. Snapshot of the menu for preparing the experience taken directly from the laptop display.

For this particular case, a Pocket Loox 720 with an Intel XScale PXA 272 520 MHz processor, 128 MB RAM memory, Bluetooth 1.2 and USB 1.1 host capabilities was used. The master may also be hosted in a laptop computer with minimal changes on the software. This master device can perform all the necessary data analysis or it can send the data to be further processed in other master devices. One of the menus of the graphical user interface provided by the master is presented in fig. 4. This particular menu allows the configuration of the experience to be performed, the definition of the biochip geometry, in this particular case a 16×16 matrix, the activation/deactivation of some of the sensors and their main characteristics extraction. Other more detailed sub-

menus are available in order to define the type of excitation to be applied to the sensors and to choose which measures have to be collected and registered.

4. Implemented Prototype and Experimental Results

The described architecture of the communication platform was implemented on a conventional two layer board PCB with room for the Bluetooth module. Because the biological analysis platform measures signals with a slow variation, the use of a Bluetooth module which provides a lower bit rate but consumes less power was considered. The achieved board has a square shape with an area of about 32 cm^2, which is smaller than a credit card. The size of the communication board is the same of the acquisition board, allowing the boards to be stacked and thereby making the system more compact. The implemented prototype of the communication platform and the acquisition and processing board are depicted in 5.

(a) Biological analysis platform. **(b)** Communication platform.

Figure 5. Prototypes of the embedded microsytem.

According to the experimental results obtained for ACP power consumption it exhibits an autonomy of about 30 hours, when the platform is continuously operating at full speed and a battery of about 700 mAh is used. This autonomy can be significantly increased by the power manager if the operating frequency is properly adjusted. For the envisaged application the acquisition sampling rate is low and, in order to save power, the Bluetooth transfer rate is set to 9600 kbit/s. Due to communication overheads the effective data transfer rate drops to 7000 kbit/s, which still fulfils biological analysis system requirements. Some results were also achieved for the cryptographic engine. In this prototype it is used the AES, with 16-byte key size applied to 16-byte block of data, as there are some optimized implementations available for the target microcontroller [Microchip, 2005]. Since this encoding method only requires 367

instruction cycles per byte, it still can be used to secure data in real-time data acquisition.

The embedded system prototype was tested using a solution of 2.3×10^9 particles/ml with 1.5 μm diameter magnetic nanoparticles. A 5 μA DC current was driven by the DAC through a 10 kΩ MTJ. The voltage signal was measured by an ADC at a sample rate of 6 Hz after passing through a suitable anti-aliasing filter and the measurement time was about 80 seconds. The solution was dropped on the sensor after about 20 seconds and after more 30 seconds the sensor was washed with distilled water. The acquired data was registered on the PDA and sent to a desktop computer using the SOAP. This data is graphically represented as a web page in fig. 6, after the removal of a 47 mV DC signal. This web page was generated using php 5.0 hosted in an Apache 2.0 web server, in order to visualize the received XML data file. The graphic is drawn through the use of the JpGraph object-oriented graph creating library in order to generate a png file that can be interpreted by a web browser.

These experimental results clearly shows a 190 μV signal due to the presence of nanoparticles, demonstrating that que the developed embedded system can be used for particle detection.

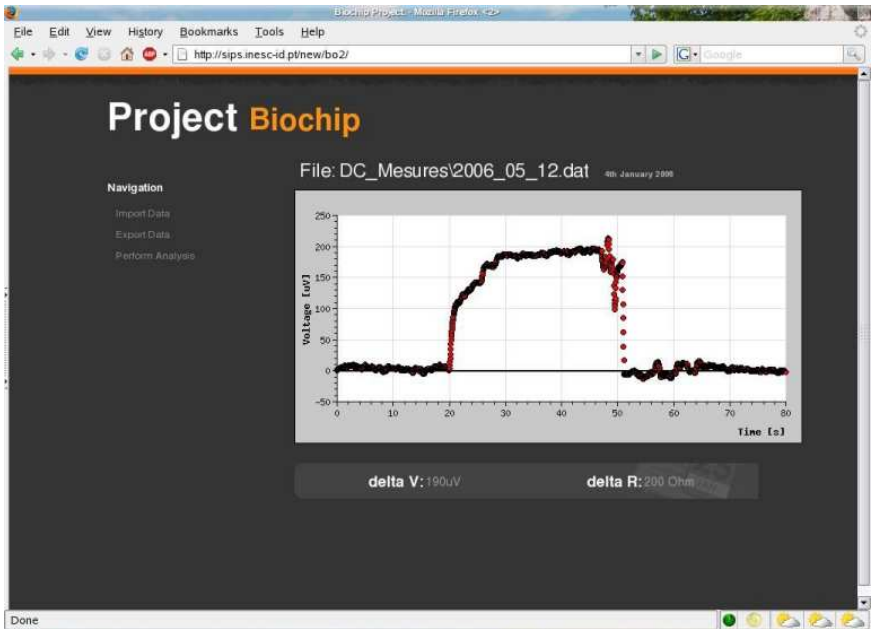

Figure 6. Snapshot of the web-based XML data visualization.

5. Conclusions

This paper presented a new generic and modular architecture for designing biomedical embedded systems. Since biomedical programmable systems are considered, both hardware and software components are important. The developed platform is adaptable to the specific requirements of a given acquisition platform. The effectiveness of the proposed architecture was shown by implementing a prototype of a biomolecular recognition system. This embedded system is based on magnetoresistive sensors to detect magnetic markers and it was implemented with off-the-shelf components. The prototype includes all the devices required to communicate with a master system and provides security mechanisms based on data encryption. The achieved data transfer rate of 7000 kbit/s is suitable for the acquisition platform but can be increased if required. Finally, the low power consumption of this prototype allows its autonomous full speed operation for more than 30 hours. These experimental results show the interest of the proposed architecture to develop distributed biomedical embedded systems.

References

[Bluegiga Tech., 2006] Bluegiga Tech. (2006). WT12 Bluetooth Module. Version 2.3.

[Halteren et al., 2004] Halteren, A., Bults, R., Wac, K., Konstantas, D., Widya, I., Dokovski, N., Koprinkov, G., Jones, V., and Herzog, R. (2004). Mobile patient monitoring: The mobihealth system. *The Journal on Information Technology in Healthcare*, 2(5):365–373.

[Jovanov et al., 2001] Jovanov, E., Raskovic, D., Price, J, Chapman, J, Moore, A, and Krishnamurthy, A (2001). Patient monitoring using personal area networks of wireless intelligent sensors. *Biomed Sci Instrum*, 37:373–378.

[Microchip, 2004] Microchip (2004). PIC18F2455/2550/4455/4550 Data Sheet. ref: DS39632C.

[Microchip, 2005] Microchip (2005). Data Encryption Routines for the PIC18. ref: DS00953A.

[Piedade et al., 2006] Piedade, M., Sousa, L., Almeida, T., Germano, J., Costa, B., Lemos, J., Freitas, P., Ferreira, H., and Cardoso, F. (2006). A new hand-held microsystem architecture for biological analysis. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 53(11):2384–2395.

[Schneier, 1995] Schneier, Bruce (1995). *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, 2nd edition.

[Weimerskirch et al., 2001] Weimerskirch, A., Paar, C., and Shantz, S. (2001). Elliptic curve cryptography on a palm os device. In *Proceedings of the 6th Australasian Conference on Information Security and Privacy*, volume LNCS, 2119, pages 502 – 513.