GENERIC ARCHITECTURE DESIGNED FOR BIOMEDICAL EMBEDDED SYSTEMS

Abstract Embedded Systems assume an increasing importance in biomedical applications such as for clinical analysis and for 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 software layers. A prototype of one of these systems for biomolecular recognition, based on magnetoresistive sensors to detect magnetic markers, was implemented by using off-the-shelf components. Experimental results show the effectiveness of the proposed architecture and the interest in 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 was been a crescent interest on embedded systems for biomedical applications, increasing the demand on computing and communication while, at the same time, maintaining a portable and autonomous system. Applications such as biochemical operations for clinical analysis (e.g, glucose/lactate analysis), DNA analysis and proteomics analysis Me, 2006 for clinical diagnostics and real-time pervasive patient monitoring Jovanov et al., 2001 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 computation and

communication levels. The major diversity of sensors and medical apparatus demand the development of general computation/communication architectures for deploying distributed embedded systems that may cover a wide band 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 show that the proposed architecture is flexible and may be directly applied in designing embedded systems for several different applications and it can be easily adjusted to fulfil all the particular requirements of each biomedical experience.

2. Proposed Architecture

Fig. 1 presents the block diagram of the hardware component of the proposed architecture. In the core of the Autonomous Communication Platform (ACP) a communication manager, which is responsible for communicating data and commands from and to a local acquisition/processing platform, must be present. Each pair of these two platforms are tightly coupled through a Serial Peripheral Interface (SPI) with a simple protocol but with large 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 the necessary protocols and provide electrical interfaces 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-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.

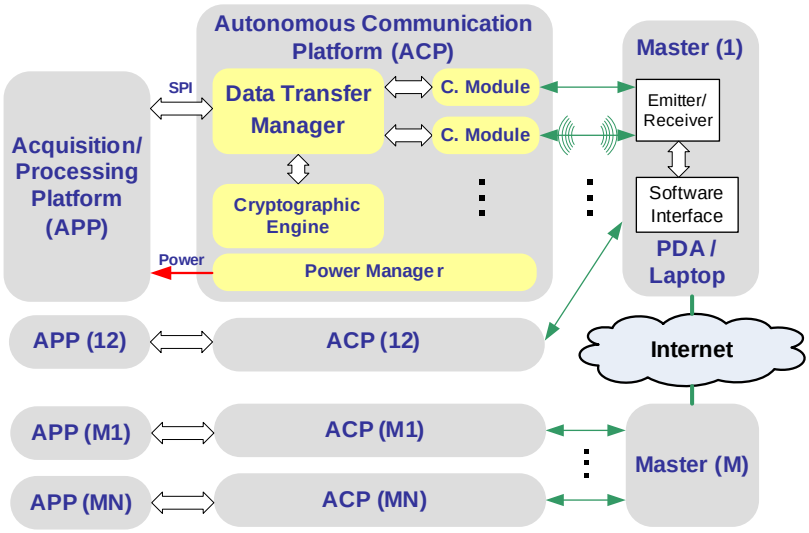

Figure 1. Block diagram of an implementation of the proposed architecture.

Two additional important blocks are present in the autonomous communication platform (1): 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 providing the recharging procedure. In some buses, such as the USB, the controller is also able to provide electrical power to the other devices. If such is the case, this electrical energy can also be supplied to the acquisition/processing platform, which usually has also to be autonomous.

To assure privacy and integrity during data transfer to the master, the communication platform may include a cryptographic engine. In the particular case of this application, it can be applied a public-key cryptosystem, with distinct private and public keys. In particular the Elliptic Curve Cryptography (ECC), which requires the smallest key size and the highest strength per bit, may be considered. There are also available some optimized implementations for portable and autonomous devices with modest computational capacity Weimerskirch et al., 2001.

Fig. 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, implementing a SPI simple protocol with a transfer rate up to 30 Mbps. It 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

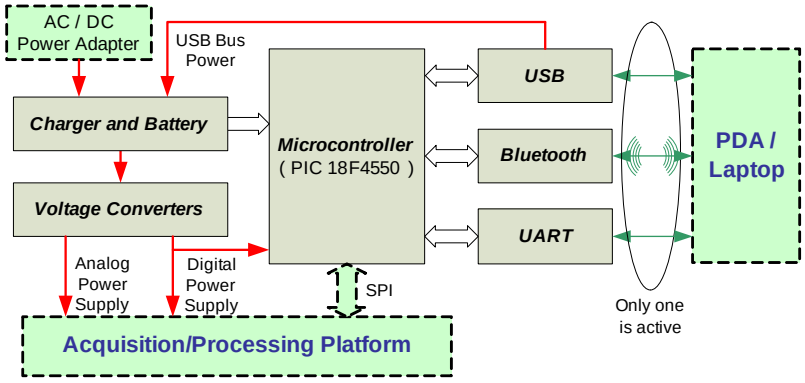

Figure 2. Block diagram of the hardware component of the proposed architecture.

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. The microcontroller is also responsible to implement the elliptic curves over GF(2^{163}), which is often considered security-equivalent to RSA with 1024-bit key length Blake et al., 1999.

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

The software of the communication platform is 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 with the public key 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 have been 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, the request acceptance services only have to write the request XML file to a given directory in the server

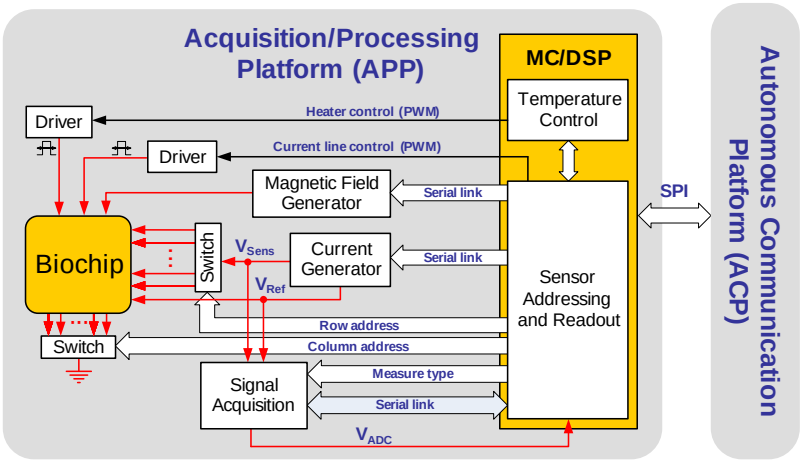

Figure 3. Block diagram of the acquisition/processing platform of the microsystem for biological analysis.

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 the request/reply data.

3. Application to Implement an Embedded Microsystem for Biological Analysis

The architecture proposed 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 Me, 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 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.

Fig. 3 presents the Acquisition/Processing Platform (APP) of the microsystem for biological analysis, which is connected to the ACP through a SPI. It generates all the 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.

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 the 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 the 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 the MTJs resistance 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 5%) 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.

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. This master device can perform all the necessary data analysis or it can send the data to be further processed in another master devices. One of the menus of the graphical user interface provided by the PDA is presented in fig. 4. It as been developed using Microsoft embedded Visual C++ environment. This particular menu allows the configuration of the experience to be performed, defining the geometry of the biochip, in this particular case a 16×16 matrix, activating/deactivating some of the sensors and checking their main characteristics. 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 collect and register.

4. Experimental Results

According to the experimental results for the ACP power consumption it exhibits an autonomy of about two days, when the platform is continuously

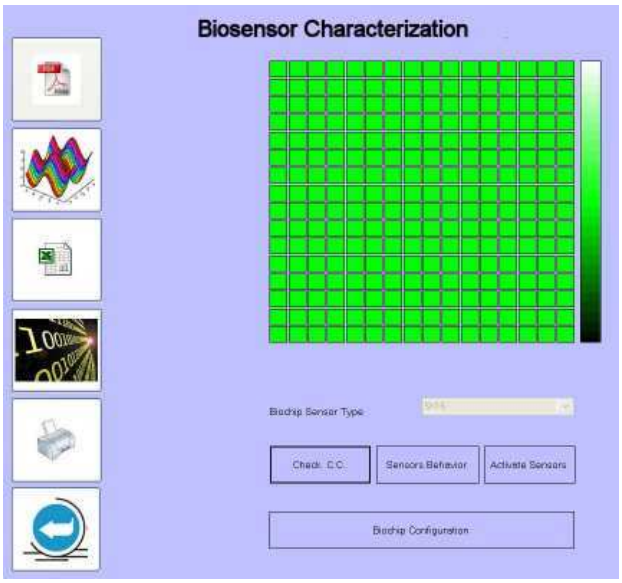

Figure 4. Snapshot of the menu for preparing the experience taken directly from the PDA display.

operating at full speed and a battery of about 2000 mAh is used. This autonomy can be significantly increased by the power manager if the operating frequency is properly adjusted.

The embedded system developed for biological analysis 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 mea- sured by an ADC at a sample rate of 6 Hz after passing through a suitable anti-aliasing filter. 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 PDA and sent to a desktop computer using the SOAP. This data is graphically represented as web page in fig. 5, 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.

Fig. 5 clearly shows a 150 μV signal due to the presence of nanoparticles, demonstrating that que the developed embedded system can be used for particle detection.

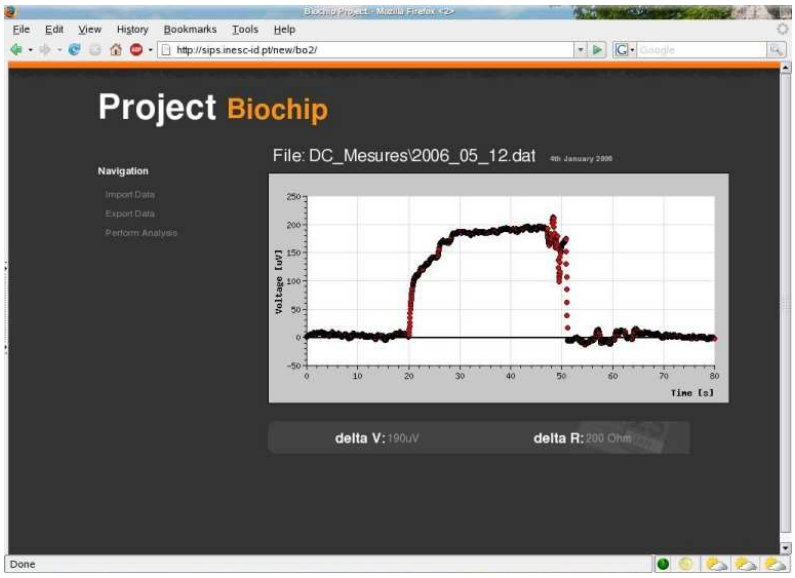

Figure 5. Snapshot of the web-based XML data visualization.

5. Conclusions

This paper presented a new generic architecture for designing biomedical embedded systems. Since biomedical programmable systems are considered, both hardware and software components are important. 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. Experimental results show the interest of the proposed architecture to develop distributed biomedical embedded systems.

References

Blake, I., Seroussi, G., and Smart, N. (1999). *Elliptic Curves in Cryptography*. Cambridge University Press, Cambridge.

Bluegiga Tech. (2006). WT12 Bluetooth Module. Version 2.3.

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.

Me (2006). Blinded for review.

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

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.