The Front End Design of a Health Monitoring System

Nikos Petrellis[1], Michael Birbas[2], Fotios Gioulekas[3]

[1]Department of Computer Science and Engineering, Technology Educational Institute of Thessaly, Greece, e-mail: npetrellis@teilar.gr
[2]Analogies S.A. and Electrical Engineering and Computer Technology Dept., University of Patras, Greece, e-mail: michael.birbas@analogies.eu
[3]Sub-directorate of Informatics, University General Hospital of Larissa Larissa, Greece, e-mail: gioulekas@uhl.gr

Abstract. In this paper an efficient e-health platform based on a low-cost sensor controller system is presented, exhibiting enhanced key characteristics able to provide broad coverage of medical scenarios in a reliable and flexible way. The heart of the system is a low-cost sensor controller capable of performing both simple medical tests and more advanced ones communicating with a Gateway and a tablet or smart phone providing instructions to the patient. Equipped with a simple and flexible communication protocol for data and command exchange, the developed platform is capable of readily supporting a variety of sensors with different sampling profiles. Furthermore, first promising results of on-going work pave the way for achieving considerable enhancement of sensors' accuracy (close to high-cost commercial ones) and significant extension of platform's portability through power consumption minimization. These characteristics have been verified by experimenting with various medical scenarios one of which is demonstrated here in detail.

Keywords: health monitor, sensors, communication protocol, low power

1 Introduction

Modern mobile health monitoring systems are classified to two major categories, a) systems based on commercial sensor infrastructures, b) systems based on low-cost open source solutions. Specifically, in (Gay and Leijdekkers, 2007), the authors have proposed a monitoring system based on wireless sensors and smart phones to monitor the wellbeing of high risk cardiac patients exploiting the usage of a real-time ECG. They use a Bluetooth enabled blood pressure monitor and weight scale from A&D Medical A&D (Medical website, 2007) and an integrated Bluetooth ECG/Accelerometer sensor from Alive Technologies (Alive Technologies, 2007). These sensors communicate with the smart-phone using Bluetooth technology. The smart-phone connects to WiFi or through GSM and transmits data to the remote gateway. The sensors are based on commercial products and it requires the usage of an expensive smart-phone to serve as first hot-spot before the connection to the server. Additionally, the work in (Chan et al, 2008) proposes also a similar

Proceedings of the 7th International Conference on Information and Communication Technologies in Agriculture, Food and Environment (HAICTA 2015), Kavala, Greece, 17-20 September, 2015.

monitoring architecture like that in (Gay and Leijdekkers, 2007) based on commercial sensors.

Furthermore, the work proposed in (Mukherjee et al, 2014) introduces a perception layer, where a processing device incorporates Bluetooth, Zigbee και WiMAX protocols to connect the patient sensors to the monitoring infrastructure and collect the corresponding data. These sensors produce raw values of data which are wirelessly relayed to a central transceiver unit worn by the patient. This transceiver unit processes the raw data and converts it into meaningful metadata. The authors of this work focus on the protocol implementation rather than low-complexity and low-consumption sensors and integration platform. Authors in (Patel et al, 2012) analyze and review the available sensor technology, communication technology, and data analysis techniques towards the implementation of wearable systems and patient monitoring, denoting the importance of the development and usage of both high-accurate and low-energy sensors. It is also suggested in (Khelil et al, 2014) an experimental Body Sensor Network platform that measures important health indications, and maps the health status of the individual carrying the platform into a set of predefined classes. They employ a commercial-off-the-shelf (COTS) low-cost platform, the e-Health Sensor Shield (Cooking-Hacks, 2013) from Libelium Cooking Hacks which relies on two popular sensing and computing platforms i.e., Arduino and Raspberry. However, no information is given for the important features of such a platform, like the communication protocol employed for data/command/exchange, power consumption and sensors' accuracy. Finally, the work proposed in (Granados et al, 2014) is based on a gateway design that performs digital filtering of vital signs signals, as well as streaming to Web clients. The powering capabilities of the gateway are based on PoE (power over Ethernet) that can also serve as a power and data source for wired sensors and building automation appliances. Alternatively, there exist a number of commercial platforms like e-Shimmer Health BSN (The Shimmer Platform, 2008), and the Simband health sensor platform (SIMBAND, 2014) from Samsung which are expensive solutions though with proprietary components without providing HW open-source features. The MIThril system (MIThril, 2003) is based on monitoring using wearable systems that employ multiple sensors that are typically integrated into a sensor network either limited to bodyworn sensors or integrating body-worn sensors and ambient sensors. However, such systems by design are not suitable for long-term health monitoring.

As far as the current work is concerned, the low-cost e-Health kit of Libelium-Cooking Hacks (Cooking-Hacks, 2013) has been employed as basis (sensor controller) for the implementation of an efficient e-health platform. In comparison with similar works we further focused in significantly improving its key features, having developed a communication protocol able to serve any kind of sensors' sampling profiles and derived first promising results in sensor's accuracy enhancement and power consumption minimization, aiming towards the development of a low cost flexible e-health sensor platform able to support in a reliable as many medical cases as possible.

The description of the work performed and the developed platform are organized as follows: in Section 2 a description of the overall e-health platform is given while in Section 3 we focus on the sensor controller. In Section 4 the developed communication protocol between the Gateway and the Sensor Controller is described, in Section 5 first promising results of on- going work are presented while finally in Section 6 a representative medical scenario is described demonstrating the effectiveness of the presented techniques and methods.

2 System Description

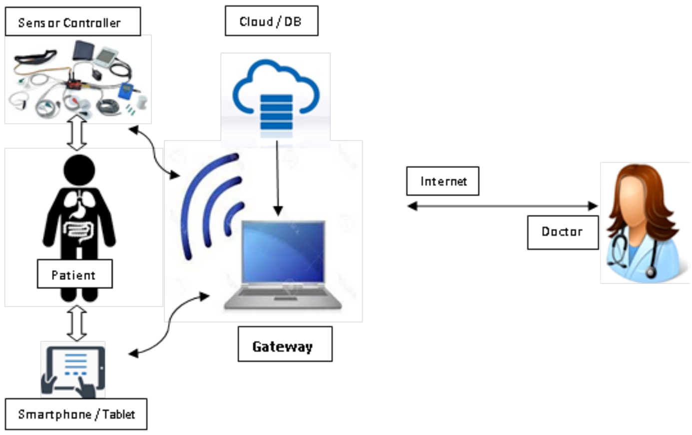

Fig. 1. The framework of the health monitoring system

The system description of the heath monitoring system is shown in Fig. 1. In such a system, the monitored patient, accepts instructions through a smart device (phone or tablet) offering a comprehensive user interface like e.g. a wide touch screen with large buttons or a mobile phone/tablet with voice recognition system

Medical sensors employed in this use include both simple tests (temperature, respiratory measurements, skin conductance, etc) and/or more advanced ones (like Electrocardiograms-ECG, Electromyograms-EMG) which are connected to the Sensor Controller usually based on a lightweight (but powerful enough to cover system specifications) portable microcontroller module. A Gateway is furthermore employed which is in most cases positioned at patient's personal place and is connected to the Sensor Controller through a wireless protocol like Wi-Fi.

A communication protocol is responsible for the transfer of the Gateway instructions to the Sensor Controller concerning the sensors that should be used and their sampling frequency. It should be stressed that the choice of an appropriate communication protocol is not a trivial issue since a variety of sensors has to be controlled with different sampling profiles.

Power consumption is another crucial issue since it is directly aligned with the degree of the platform portability that could be provided to the patient. To ensure that portability is supported, the system power supply, usually a light battery, should last sufficiently long before being recharged. It should be noted at this point that the main parameters through which power consumption can be affected are limited to the platform's microcontroller operation and the wireless communication protocol controlled, since the power required by the controlled sensors cannot be touched.

The Gateway is providing an intermediate node between the patient, the doctor/hospital and the tablet/smartphone. Its main tasks include the communication with the Sensor Controller and the tablet to forward the instructions to the patient and to gather the medical test results from the Sensor Controller. These test results can be forwarded either directly to the doctor or a hospital data base or to the cloud. Some trivial tasks and decisions can be directly taken by the Gateway like e.g. in the case where the room temperature exceeds predefined limits, while it should also activate appropriate alarms if the patient delays to respond, etc.

The ordinary sensors used in health monitoring systems are grouped in two categories: simple sensors with analog interface (like temperature, respiratory, etc) which might call for some special range adaptation circuitry to connect to corresponding ADC channels and smart sensors with digital interface (like blood pressure measurement, etc) with predetermined accuracy.

3 Sensor Controller

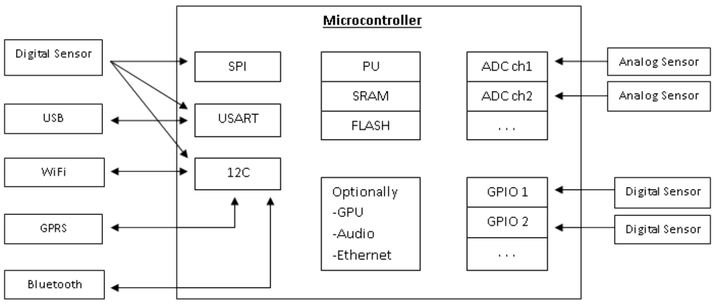

Fig. 2. Block Diagram of the Sensor Controller

As shown in Fig. 2, the Sensor Controller is a "mobile microcontroller board the rough functionality of which is described in the block diagram above. The analog sensors are connected to separate ADCs and/ or to different channels of the same ADC. So as to achieve matching of the sensor output to the ADC input, without linearity degradation, range adaptation circuits might have to be employed, including either passive components like voltage dividers or Wheatstone bridges in the simplest case or active circuits like operational amplifiers, while voltage shifting circuits may also be required. Such range adaptation circuits have to be designed carefully so as to ensure high linearity and low noise.

As far as smart sensors with digital interface are concerned (Fig2), they are connected to serial buses like Serial Peripheral Interface (SPI), I2C or USART/UART according to the interface provided by the specific sensor. However, the communication boards used to transfer information between the Sensor Controller and the Gateway may also need to be controlled by one of these interfaces. This might lead to conflicts regarding the sensors that should be connected directly to the Sensor Controller since the serial bus interface resources are limited. The sensors with the simplest communication protocol can be alternatively connected to General Purpose I/O (GPIO) pins that will be configured to operate in the same way as the serial bus supported by the sensor. However, the software drivers that will be needed to emulate the operation of such a serial bus may be a significant overhead.

It is important that the Processing Unit (PU) and the size of the internal RAM and Flash memory allow for the use of a Real Time Operating System (RTOS) so that several real-time tasks could be handled in parallel. Furthermore, if the RTOS offers a Graphical User Interface (GUI), the development and debugging process of the health monitoring system is also simplified reducing the time to market. Another important characteristic is the size of the SRAM and Flash memory in the sense that it should be adequate to store firmware with high complexity. As far as real time operation is concerned, the Sensor Controller is usually placed on a table or carried by the patient when the medical tests have to be performed. The Sensor Controller communicates with a Gateway through a wireless protocol like Wi-Fi or Bluetooth.

The current work uses as reference development platform the low cost e-Health kit of Libelium-Cooking Hacks, which exhibits most of the aforementioned features. The supported sensors with analog interface can measure: temperature, skin conductivity, position, breathing airflow, ECG and EMG (EMG cannot operate concurrently with ECG) while the digital sensors offered are: blood pressure (connected to UART), measurement of pulse and oxygen in blood (requires 8 GPIO pins) and glucometer (connected also to UART and thus cannot operate with blood pressure sensor concurrently). The e-Health kit platform can be directly plugged onto an Arduino development board or to a Raspberry PI through an Arduino bridge. The basic functionality offered by the Arduino bridge is the rearrangement of the Raspberry digital pins so as to provide Arduino-compatible headers. An additional ADC has been also placed on the Arduino bridge to increase the number of analog inputs offered by the Raspberry PI.

Due to the higher processing power (32-bit ARM core) of Raspberry PI and its increased RAM/ROM size compared to the 8-bit AVR microcontroller that forms the core of Arduino, Raspberry PI was selected to serve as Sensor Controller. Moreover, the advanced GUI offered by the Raspbian OS installed on the Raspberry PI board makes possible the use of the Sensor Controller in stand-alone mode. An SD card is required by the Raspberry PI to install the Raspbian OS which is also used to store temporary data and command files required for the communication with the Gateway. Finally, the stacking of the e-Health kit over the Arduino bridge and the Raspberry PI allows for the placement of a wireless communication module between the Arduino bridge and the eHealth kit as shown in Fig. 3. Both Bluetooth and Wi-Fi interfaces were considered as candidates for the wireless communication between the Sensor Controller and the Gateway. The Wi-Fi interface was finally selected due to its higher reliability and the incompatibilities discovered between the Bluetooth card used at the Sensor Controller and the smart phones tested. The connectors on the top of Fig. 3 are used to connect the sensors listed above.

Based on this experimental setup (Fig. 3), we focused on the definition and implementation of an appropriate communication protocol for the health monitoring system while also successful results were derived so far on both the improvement of sensors' accuracy and minimization of power consumption as part of on-going work.

4 Communication Protocol

The communication protocol is responsible for the data and command exchange between the Gateway and the Sensor Controller. The commands are used by the Sensor Controller to schedule the appropriate intervals that the various sensors should be sampled according to the supported medical case study. Various methods like HTML server or clouds could have been employed to support the wireless data exchange between the Gateway and the Sensor Controller, but in our case we preferred to have the Gateway configured as an FTP server since exchanging command and data text files is more convenient for debugging and logging purposes. This message passing communication protocol is implemented through two file types: command and data, that are exchanged between the Gateway and the Sensor Controller. The command file has the format shown in the example of Fig. 4 and is prepared by the Gateway according to instructions set by the supervisor doctor.

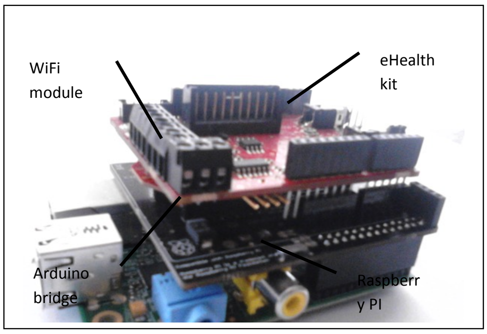

Fig. 3. Stacking of the basic modules of the Sensor Controller

The first line of such a command file defines whether the system will go to sleep mode for a predefined interval or will perform a sampling scenario (lines starting with -- are comments). In the case of Fig. 4, the first line determines that a sampling scenario is to take place. Specifically, in the first line (of Fig. 4), the first character is either (R) or (S) denoting that the sensor controller will either enter a sampling or a sleep mode respectively. Two integer numbers follow which represent parameters Q(uantum) and I(nterval). The Q parameter represents a time interval expressed in us, while the I parameter defines that the command file will be read again by the Sensor Controller after a delay of I ms. If the application software on the side of the Gateway does not change this file, the same sampling scenario will be repeated by the Sensor Controller (every I ms).

Each pair of the following lines (ignoring the comments) is assigned to a specific sensor. The first line of each pair of lines associated with a specific sensor defines only an integer number N that corresponds to a sampling interval equal to $N \times Q$ (in ms) for this sensor. If N is selected to be 0 this means that the specific sensor is not used. The second line in each pair is reserved for future options.

Based on the specification described above, the Sensor Controller software initially reads the command file and starts a loop during which, the sensors with nonzero N are sampled. A sleep interval of Q ms is inserted before the next iteration. When each sensor s is sampled it is omitted from the sampling process for the next Ns iterations in order to create the Ns×Q pause interval between successive samples.

The data samples are stored in files that are transferred back to the FTP server. The data files can contain mixed sensor indications distinguished by the sensor name and followed by the sampled value and a timestamp. Optionally, different files can be used to separate the sensor values according to the configuration set by the Gateway.

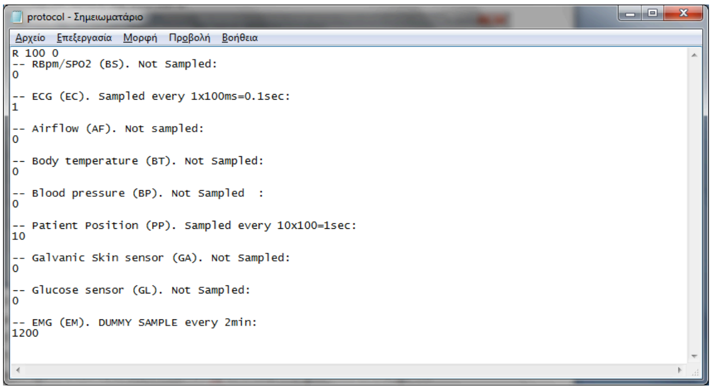

Fig 4. Sampling ECG (every 100ms) and patient position (every 1 sec) sensors.

5 On Going Work

The on-going work is towards enhancing the sensors' accuracy and lowering significantly the platform's power consumption.

Sensor Accuracy enhancement

As already described in section 3, for specific sensors (like analog ones) it is necessary that they undergo a (hardware) calibration scheme (e.g. use of Wheatstone bridge techniques etc) to update default parameters so as to ensure sufficient accuracy. These (hardware) techniques suffer however from several drawbacks, such as sensitivity to environmental, temperature and power supply variations, etc.

In contrast to this approach on going work has been focusing in the improvement of the accuracy of this type of sensors measurements through software instead of modifying the hardware of the eHealth development platform. More specifically, the calibration values are stored within configuration files that are updated by the Raspberry PI microcontroller system using information about environmental conditions. Then, a filtering procedure takes place using a simple moving average with extreme value exclusion. More specifically, first a moving average is applied on the last k sample values (v_t, v_{t-1},..., v_{t-k+1}),and an initial average A_{in} is estimated. Following this, each v_i of the k values is compared with A_{in} and if their difference exceed a predetermined threshold T_h, then a corresponding binary flag m_i is set to 0 otherwise it is set to 1. After updating all the m_i values, the final average A_{fin} is estimated and finally used. Application of this technique to a number of sensors (like airflow, skin conductance and temperature measurement sensors) has so far given very satisfactory results with the processed measurements being close enough to ones from (expensive) medically certified sensors.

Power Consumption

Another crucial factor where we are targeting to, (on going work), is the minimization of the power consumption and especially that of the Sensor Controller (as well as of the WiFi module) through the exploitation of the idle intervals of these devices between measurements by putting them in a sleep like mode. Experimental measurements show that the power consumption at full operational mode (when all the sensors are connected) exceeds 1A which means that if an external 1000mAh battery is used it would have to be recharged every hour. Sleep mode for the WiFi module (Roving Networks RN-171) can be readily enabled where only 4uA is consumed (compared to the 35 mA and the 185-210mA required at normal mode at the receive/transmit path of the device respectively).

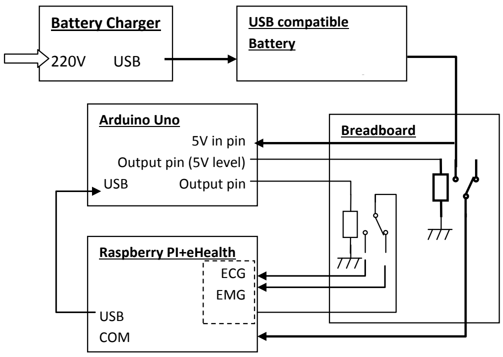

Fig. 5. Arduino-Rapsberry block diagram to implement low-power consumption

However this is not the case for the specific (Rapsberry PI) microcontroller board that does not support sleep or any other low power mode. For this reason, an additional low cost/power microcontroller that supports sleep mode is employed, (Arduino in our case) just to turn off and on the Raspberry PI module, as shown in Fig. 5 through the use of mechanical relays. Specifically, although the Arduino module is constantly powered by the external battery, the Raspberry PI is powered through a relay the normal position of which is off. When the Arduino is powered up, it immediately switches on the relay, powering the Raspberry PI, while also the Arduino can turn off the Raspberry PI for a predetermined sleep interval if it is instructed to do so by Raspberry PI. It should be noted that the 8-bit ATmega1280 microcontroller used on the Arduino board draws less than 1mA in normal operation and less than 1uA in power down mode. Important power savings have been achieved, and the battery life has been extended from just few hours to days.

6 Case Study

As a demonstration of the issues discussed in previous let's assume that the following medical scenario needs to be performed: a) initially the patient position is tested every second for 4min in order to verify that he has taken the right position for the following tests, b) an ECG test should be performed for 2min with 0.1sec intervals, c) after 10 minutes break its blood pressure should be measured just once. It should be noted here that the patient position has to be concurrently monitored in both (b) and (c) tests to verify the validity of the examinations and thus the patient position sensor is sampled every second. To implement this medical scenario, the following actions have to take place:

A command file with a sleep instruction may have been deposited at the FTP server from the last measurement and the Raspberry PI wakes up, reads this file and falls in sleep mode again until a different than sleep command is found in this file. The tablet instructs the patient to wear the position and the ECG sensors.

The patient wears the patient position sensor and the ECG electrodes, switches on the Sensor Controller module and notifies the tablet that he is ready. The Gateway is notified through the tablet to change the command file into a new one that instructs the Sensor Controller to wake up and enter the sampling mode (1st character of first line of command file is R followed by Q=1000 and I=0)) where only the patient position sensor is instructed to be sampled every 1x1000mss=1 sec (N=1 in the line of the command file corresponding to the specific sensor) as well as the EMG sensor (dummy sampling every 4 min-N=240 in the command file line corresponding to this sensor). This command file has as effect that the patient position sensor is sampled immediately (after the command file is read by the Sensor Controller) every second for N×Q=240x1000=4min when a dummy read to the EMG sensor is performedThen, immediately the command file will be read again from the FTP server and the patient position data will be sent to gateway

The Gateway during the 4 min interval described above prepares the command file shown in Fig. 4, which instructs the sensor controller to perform the ECG test for 2min. If this modified command file (Fig.4) was not prepared within the 4min interval, the patient position will be monitored for another 4 min interval before ECG starts. After the ECG test is completed, the data are sent to the FTP server and the command file will be immediately read again. This command file could have been modified by the Gateway in the last 2min interval during the 10 min sleep mode so that preparations for the blood pressure test take place. The patient in this time window removes ECG, measures his blood pressure and connects the instrument to the Sensor Controller.

After connecting the blood pressure instrument to the Sensor Controller the patient notifies the tablet that he has performed the measurement. The Gateway will alter the command file to prepare the transfer of the measurement which takes place after the Sensor Controller wakes from the 10 min interval. Following the reception of the blood pressure measurement by the Gateway, the command file can be changed to sleep mode until the patient is instructed by the tablet to turn it off.

Regarding the amount of power consumption required, it is derived that the whole energy consumed per day for this case study 111566.6mA×sec. This means that a

1000mAh battery would last 1000×3600/111566.6, i.e. more than 32 days before a recharge is needed.

7 Conclusions

A low-cost platform for remote heath monitoring of patients was presented. The system's sensor controller is based on the e-health kit from Cookie-Hacks being capable of performing a number of medical tests from temperature measurement to Electrocardiograms. A flexible communication protocol allows for the determination of any type of short or long term sampling of sensors at the desired frequency. This feature in combination with first promising results derived on sensor's accuracy enhancement and power consumption optimization opens the way for a low-cost system with its key characteristics comparable to those of high cost commercial ones.

Acknowledgments. This work was performed as part of ELTAB project (465435).

References

1. Alive Technologies (2007) [Online]. Available at: http://www.alivetec.com [Accessed 03/10/2014]
2. Chan, V., Ray, P. and Parameswaran, N., (2008), Mobile e-Health monitoring: an agent-based approach, Communications, IET , Telemedicine and E-Health Communication Systems, 2(2), p. 223-230.
3. Cooking-Hacks.(2013) [Online]. Available at: http://www.cooking-hacks.com/documentation/tutorials/ehealth-biometric-sensor-platform-arduino-raspberry-pi-medical [Accessed 03/20/2014]
4. Gay, V., and Leijdekkers, P. (2007) A Health Monitoring System Using Smart Phones and Wearable Sensors, Int. Journal of ARM, 8(2), p. 29-35.
5. Granados, J.; Rahmani, A.M.; Nikander, P., Liljeberg, P. and Tenhunen, H., (2014), Towards energy-efficient HealthCare: An Internet-of-Things architecture using intelligent gateways, 4th Int.Conf. MobiHealth, Nov14, Athens, Greece.
6. Khelil, A., Shaikh, F.K., Sheikh, A.A., Felemban, E., and Bojan, H. (2014), DigiAID: A Wearable Health Platform for Automated Self-tagging in Emergency Cases. 4th MobiHealth, Nov14, Athens, Greece.
7. Medical website (2007) [Online]. Available at: http://www.andmedical.com.au/ [Accessed: 02/12/2014]
8. MIThril (2003), [Online]. Available at: http://www. media.mit.edu/wearables/mithril/, [Accessed: 02/12/2014]
9. Mukherjee, S., Dolui, K. and Kanti Datta, S.(2014), Patient Health Management System using e-Health Monitoring Architecture, Advanced Computing Conference (IACC' 14), IEEE International, 21-22 Feb. 2014, p.400 – 405

10. Patel, S., Park, H., Bonato, P., Chan, L., and Rodgers, M. (2012) A review of wearable sensors and systems with application in rehabilitation. Journal of Neuro Engineering and Rehabilitation p. 9-21.
11. SIMBAND (2014), [Online]. Available at: http:// www. samsung.com/us/ssic/innovation_areas/#simband [Accessed: 07/02/2014]
12. The Shimmer Platform (2008), [Online]. Available at http://www.shimmersensing.com/ [Accessed 03/10/2014]