ABSTRACT

DESIGN AND IMPLEMENTATION OF BIOMESENSI SOFTWARE PLATFORM

By

Keerthi Parameshwaran

August 2015

The objective of this thesis is to implement the software stack in kernel level for a wearable body sensor device named BIOMESENSI. The developed software stack enables necessary protocols to allow the users to obtain their vital physiological data when they wear the device. This vital physiological data aggregated and processed by BIOMESENSI, consists of skin and core body temperatures and heart rate. Furthermore, the data can be used to plot electrocardiography (ECG) using an application. Moreover BIOMESENSI is supported with other state of the art sensors including accelerometer and gyroscope to monitor physical activities. Above all, it also has proximity and ambient light sensors for power saving purposes in future applications. Finally, BIOMESENSI provides Bluetooth communication to transfer the processed vital data to the base station (such as android based tablet or mobile) from the sensor nodes. BIOMESENSI uses ARM Cortex M3, which is a 32 bit microcontroller known for its low power consumptions and high performance.

DESIGN AND IMPLEMENTATION OF BIOMESENSI SOFTWARE PLATFORM

A THESIS

Presented to the Department of Electrical Engineering

California State University, Long Beach

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Electrical Engineering

Committee Members:

Mohammad Mozumdar, Ph.D. (Chair)
Anastasios Chassiakos, Ph.D.
I-Hung Khoo, Ph.D.

College Designee:
Antonella Sciortino, Ph.D.

By Keerthi Parameshwaran

B.E., 2009, Visveswaraya Technological University, Belgaum, India.

August 2015

TABLE OF CONTENTS

LIST OF TABLES.

LIST OF FIGURES

FIGURE Page

FIGURE Page

CHAPTER 1

INTRODUCTION

The humans in the world have a different life style according to their culture and geographic location. As a result, they may face with various diseases. Most of these diseases are curable, if detected at an early stage. Detection of these diseases is mostly provided in hospitals which require travelling and affordability [6, 7].

With the present day advancements in wireless sensor networks (WSN) and embedded computing technologies, portable monitoring devices have become practically viable. In addition, the lately proposed Body Sensor Networks (BSN), a collection of body sensors, provides continuous monitoring and analysis of vital parameters of the body.

As discussed above, monitoring this crucial information helps us get a better understanding on the functionality of our body. This progress of understanding the functionality based on the body and environment gives an edge to the health care system [8]. Combining the information helps us to comprehend it in a better way. If there is an unusual activity within our system, it gets detected by our device to gather the useful information. Earlier research in the medical field tells us that humans can recover from illness and even from many serious diseases. This is one of the main reason why we need to detect health issues at an early stage so we can recoil back easily and live a healthy life. A development platform, which is compatible with BSN hardware, is presented to simplify study, design and implement from multiple sensor data.

A BSN Development platform has to be chosen with constrains such as, low power consuming, supple and firm design. The BSN nodes deliver a supple environment, which is good for doing research, design and development. Then the data could be sent to any storage through the wireless protocols like Bluetooth, RFID, Wi-Fi et cetera [9].

The portable hardware platforms which was implemented, consists of different blocks such as ambient light and proximity sensor, motion processor, temperature sensor and ECG based on Galvanic skin response (GSR) [11].

The ambient light and proximity sensor is used for power saving application, motion processor supports accelerometer, magnetometer and gyroscope data. It consists of power module for power management and other electro static device (ESD) protection on-board chips. Bluetooth is used for wireless data transmission between the hardware and the mobile application. One single Arm Cortex M3 MCU (LPC1549) is used to connect all of the above components for computational processing bus management [10].

This thesis demonstrates a complete implementation of a software stack, for BIOMESENSI device. This software architecture is responsible for building the kernel infrastructure of the main microcontroller unit and enables communication to the peripheral devices which uses industry standard protocols I2C and USART to transmit and receive data wirelessly over Bluetooth.

Also, maintaining the primacy of the device is more important. We use the heart sensor which could be given higher precedence, because we humans depend more on heart. Similarly, all the peripheral has to be given some importance, so that we get vital data from all the attached sensors with small intervals, which in turn helps to check vitals related to the sensors attached to the device.

CHAPTER 2

RELATED WORKS

By comparing with many existing Body Sensor Network (BSN) or Body Area Networks (BAN) the major issue with BSN is power management. The hardware consists of battery connection externally but recharging the battery often is troublesome and hence the hardware should be energy efficient and power management is also necessary for long battery life and the hardware which we has Energy efficient MAC [1]. Also, sensors we use should consume less power. There should be battery management in software to indicate the battery level.

Some of the existing software stacks for BAN (SPINE, SHIMMER and Mercury). I will discuss about them in below paragraphs.

SPINE (Signal Processing In Node Environment)

To program a BSN, is a complex task, because of the resource constrains for wearable device and non-availability of an easy and flexible software stack. To program a BSN, three main methods were developed. The Monolithic method is the first method or general approach, in which the software is developed by accumulating reusable codes, application specific logic and low level services. This method is difficult to extend because it is poor in reusability. The second way of programming is a middleware approach. A software layer consisting of routines which is already implemented and it is called middleware and its working is in the high level. Thus developer need not concentrate on the low services and programming is easier (e.g., Agilla [2]).

The third method is based on the framework and is a mix of good features from both first method and second method. This approach has a framework that has libraries for specific domains and has options to reuse the exiting application and this approach provides target specific application, which reduces the programming time through reuse and modularity.

Spine programming is based on third approach. It has a framework which is domain specific and it is an open source framework. Many libraries and service architecture specific to different BSNs is offered by the SPINE. Spine is based on the principles which are as follows. It is open source project, the open source could attract more developer and users, it is high level concepts and it provides high level libraries[3] for programming, including the protocols, functions processing the data, utilities and provides support to specify easy features and services.

Spine libraries allow the developer to design a code in high level abstraction, than the operating environment currently available like TinyOS. It allows interoperability through APIs. The APIs that are provided by SPINE can be operated with various devices, such as mobile phones or PCs. The light-weight JAVA APIs is provided for local applications and remote applications, which is used to issue service routine or manage sensor nodes. Also, SPINE helps the designer to efficiently prototype with respect to the channel bandwidth and energy used, distributed BSNs data classification algorithm.

Spine Network Architecture is shown in the figure 1, in this architecture, there is one coordinator and many sensors attached to it, which forms a network. The coordinator manages all sensor connected to it, and collects the data and analysis it. Coordinator acts as a gateway between BSN and Wide Area Network. The local sensor node measures the raw physical data and send the data to the coordinator.

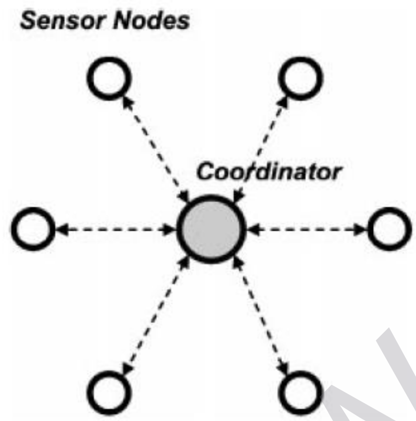

FIGURE 1. SPINE Network architecture.

Spine network currently support Star network, whereas it can be extended to multi-hop network.

The common tasks supported by node side BSN are as follows, Sampling is the first step to develop BSN application; feature extracting on a node side reduces the radio usage, and queries are supported for available sensors. Node synchronizing is a must because data acquired from different sensors should represent the same time intervals and duty cycling is used to check radio status whether it is on, off or idle, to cut down battery consumption and increase its battery life. An interface to body sensor network is interfaced to the user application by the coordinator. The SPINE based network is managed by the user application through the light weight SPINE APIs [3].

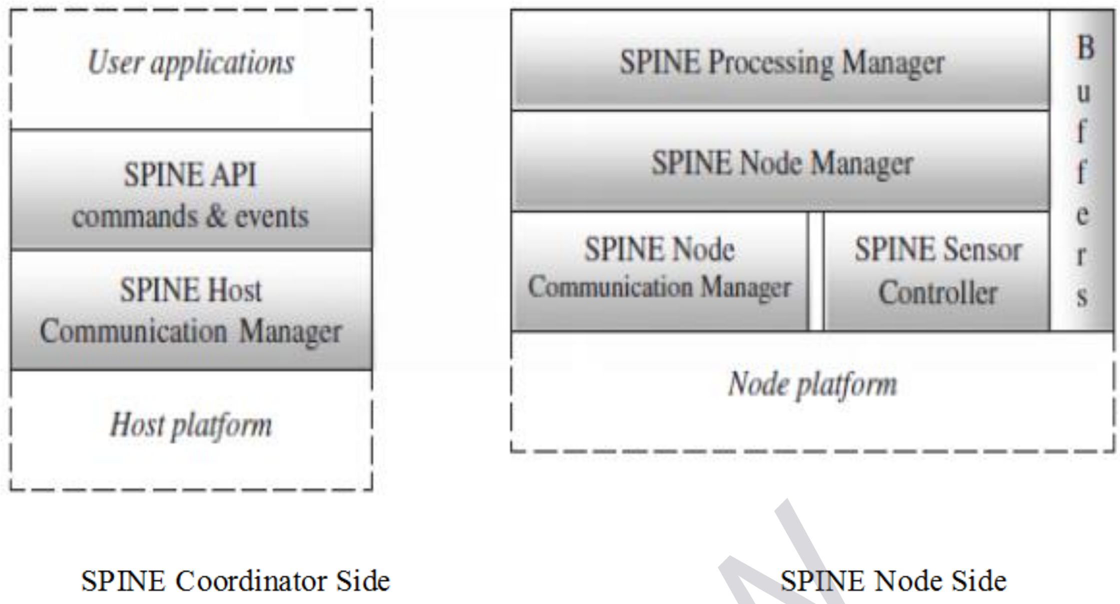

FIGURE 2. SPINE Architecture.

Generated high level events such as node alarms, new node discovery, sensor node communication and messages which is generated by system, such as warnings for low battery by the remote BSNs gets notified to the registered user through the surface level of SPINE. The event generated by the network and the user application commands are respectively decoded by SPINE host communication manager in the high level information host and coded in low level SPINE messages. This unit is responsible for packet generation and retrieving the packet, this is interfaced to host platform and physical radio is accessed by doing so.

The SPINE framework on the node side takes care of providing the hardware resource abstraction such as sensors and the radio to the developer. It has a flexible and a modular architecture, a ready to use default common signals and the architecture is flexible to extend the framework, when a new signal processing service is introduced and the new physical layer is supported. In specific the communication manager of SPINE node acts equivalent to the communication manager of SPINE host [3].

To optimize energy consumption, management policies can be taken into consideration such as a proper use of the radio module. The sensor on the node manager is managed and abstracted by the SPINE sensor controller and the sensed data is stored in a buffer. The central component is the SPINE node manager which manages the requests from a remote device and send to appropriate components. The SPINE processing manager does the actual processing services by the dispatcher within the processing manager and user-defined service integration is interfaced using this.

During the compile time, the developers are allowed to tune the sensor nodes with various parameters. We can tell the compiler which functions and sensor drivers have to be included in the compilation process, and also the number of buffers to be allocated to sensors and the size of the buffer. Other parameters are tunable as in TinyOS.

Also, some parameters are tunable during the runtime. All the remote sensors which are available are in an idle state, and can be initiated sampling any time, the sampling time and time scale can be tuned at runtime. Alternatively, the processing function which is computable can be initiated or stopped remotely [3].

<u>BioMOBIUS SHIMMER</u>

The SHIMMER is a platform designed flexibly for sensing and monitoring the patient. It consists of diagnostic tools with hardware power management, an open software which can be used for any research work. Any modification in this platform is done through configuring the software device and configuring physical, which requires simple software and hardware support. To improve sensing structure, a single daughter board is placed, which consists of transducers, signal conditioning and other subsystems, which is connected to the baseboard and provides boosted communication and computational proficiencies. Also, connectors are deployed if expansion is required, which supports the secondary daughter board [4].

There can be two types of add-on board in TinyOS, one extension is motherboard and other is standard. The standard board consists of two optional sensors temperature and humidity of the sensor. The mother board consists of option for SD card data storage, General Packet Radio Service (GPRS), the local area network and wireless local area network. The shimmer is approached in a way it helps in both engineering and medical field by improving sensing in healthcare. BioMOBIUS is widely used in the protocols, such as gait analysis, in-home gait velocity measurement and alertness training platform.

The baseboard which exists in SHIMMER helps in communication, connecting to the daughter board, sensor computations, and data storage. The baseboard allows to extend the feature through the daughter board which gives access to ambient sensing, kinematics and physiological [4].

<u>Primary Computation</u>

As it is illustrated in Fig 3. The Baseboard of the SHIMMER has a core element of Texas Instruments MSP430, it is mostly used in wireless sensing. This microcontroller unit (MCU) main feature is that during the inactive state it has extreme low power consumption.

In shimmer, using synchronous SPI interface the micro SD card socket and radio IEEE 802.15.4 is connected. The external system can be connected through three or four pins in synchronous mode via USART. The core MCU has Eight Bit ADC channel conversion. Data input from XYZ accelerometer, external and internal expansion connector uses the SHIMMER's external ports. The ADC is enabled when it is in use and disabled, if not used to support low power capabilities.

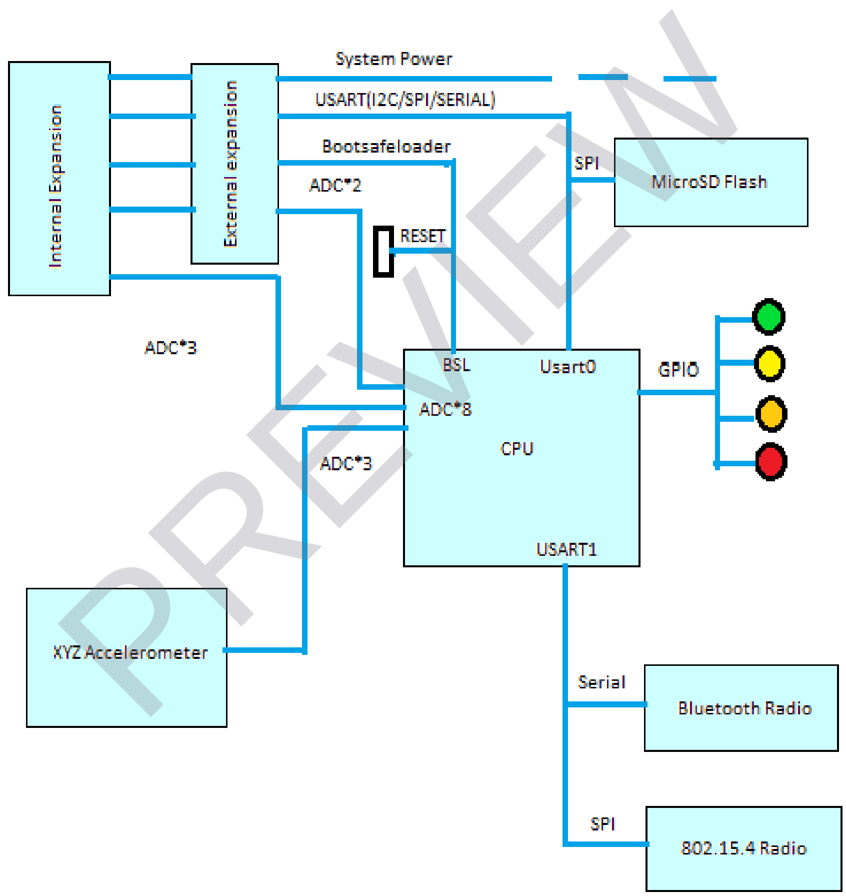

FIGURE 3. Shimmer Base Board interconnection.

Communications:

The SHIMMER uses CC2420 radio transceiver which manufactured by Chipcon and 2.4 GHz antenna is used for wireless communication and it is controlled by Serial Peripheral Interface over the USART1, and through USART1 the Bluetooth module is connected and controlled.

Micro SD storage:

The storage supported by the Micro SD card which is part of SHMMER baseboard is up to two gigabytes. This simplifies non-data streaming application storage and makes sure there is no loss in data when it is mobile, when changing battery and during outages of the network.

Power management:

Making sure to disable the ADC when not in use by the hardware and ensuring to power off the accelerometer, Micro SD card, radio and Bluetooth, when not in use through software also accounts for battery life. Reducing the use of radio for communication to the micro SD card can extend the battery life, but to capture high sampling rate data this is not an optimal solution.

Shimmer expansion:

Both internal connector and external connectors are in-built in the SHIMMER baseboard to expand platform. The daughterboard such as electrocardio gram (ECG), galvanic skin response and kinematics are connected through the internal expansion.

At external expansion, the external programming port can be connected to the board by the user.

SHIMMER Sensing Capabilities

Sensing Motions:

The accelerometer is interfaced to the MSP430 micro-controller through the ADC and uses the three channels of ADC. The gyroscope is connected as a daughterboard and consists a pair of (Miniature Electro Mechanical systems) MEMs gyroscopes. There is a flex segment over the module, used to orient the sensor properly for X, Y, and Z measurements. The daughterboard of the magnetometer uses a magneto-resistive sensor designed for low magnetic field detection and uses a tri-axial gyroscope.

Ambient sensing:

The daughterboard of motion sensor, uses a Panasonic (NaPiOn) passive infrared motion sensor, which is exceptionally consolidated and consumes less power. The daughterboard of ADL (activities and daily living) have two sensors, light and temperature sensor.

Physiological sensing:

The ECG daughterboard consists of lead inputs, which comprises weak pull ups for the detection of floating electrodes. The frequency response of the ECG daughterboard is 0.05 to 150 and has an amplifier gain of 175. The board current consumption is 0.18 mA.

External hardware integration:

SHIMMER supports two expansion modules, General Purpose and PRIMMER. In general purpose expansion, two analog signals are sent to SHIMMER, and in the PRIMMER daughterboard, the SHIMMERS internal connector is extended. The