Community Health Device Manager with Active Calibration

I S Naveen[1] isnaveen98@gmail.com
Subhadeep Purkayastha[1] subhadeep422@gmail.com
Vimal Mishra[1] vimal11592@gmail.com
Vishakha Bargotra[1] vish2698319@gmail.com
Swastik Gupta[1] swastika.gupta@smvdu.ac.in

[1]School Of Electronics & Communication Engineering
Shri Mata Vaishno Devi University
Jammu &Kashmir, INDIA

***Abstract*—This paper presents a noble design and algorithm to boost interoperability of medical devices. The local processors of the medical devices and the standards are replaced by a common plugging device where only sensors, camera and the output devices like automatic injection system etc can be plugged and configured through an android application to trigger upon checking of a parameter with a threshold value. Here data from the sensors or camera are not processed locally instead send to a remote server where appropriate function will be called and the value of the parameter will be calculated. At the server end, a labview based application is developed with different object specific matlab codes embedded in it. The whole model is implemented and for communication between user and the server end, zigbee is used. To nullify the effect of temperature and humidity on the sensor reading, an adaptive algorithm is also implemented at the server end. The device specific terminologies and operation models and standard are implemented on a single platform which not only reduces the cost of remote health monitoring but also helps to maintain uniform standards as all the interaction with all the data frames are occurring at only one end and thus implementation of standards is easy and cost-effective. It will also help to introduce new proven algorithm to be implemented in a very short time and with lesser effort as compared to today's system where the end user needs to buy the entire new device for getting the benefits of the new technological change.**

Keywords— interoperability; threshold value; server; zigbee; adaptive algorithm

I. INTRODUCTION

Cost of each personal medical device like glucometer, blood- pressure meter etc. is very high and for personal health-care system each of those parameters are essential to predict the patient's health status. Thus each device individually can provide only a part of the information and for complete analysis we need majority of the devices. Now each of these devices differs from one another only in terms of sensors used and the codes embedded in processors. So, there can be a system where same processor can shared by multiple sensors for example we can have a plug and play device where only sensors can be plugged in a port assigned for that particular sensor and data can be passed to the processor where the appropriate function will be called and the parameter can be measured. This will save the cost of having individual processor for each device.

Today technology is changing at a very fast pace and every day we have a better algorithm available for detecting diseases or for measuring body parameters, but todays approach of having separate embedded device denies up gradation.

Instead we have developed a system where user only needs to buy the sensors he/she is willing to use and the plug and play device that we designed. All the codes are hosted on a common server which processes the sensor data. This approach will enable us to upgrade/modify the code whenever we want and only one high end processor at the server end can be used to measure multiple parameters.

This system will also help researchers to implement their algorithm and reach their target audience at a much faster rate. Till now these algorithms remained confined only within laboratories as a complete new device must be designed where the algorithm must be implemented to reach the customers along with appropriate marketing stetegies. This process needs time and by the time this product reaches market, a better algorithm emerges for the same problem.

Moreover it is very cumbersome to maintain a large number of medical devices. With advancement of Internet of things (IOT), almost everyone has a mobile and internet connection. So, along with maintaining medical records and tele-medicare we can use the same resources to reduce the number of independent medical devices to one and instead can only plug the required sensors. This will also reduce the cost.

Today we have Matlab based algorithms which can detect diseases like tuberculosis, cancer etc through image processing. But those devices need very high end processors along with camera interfacing to work. Thus cost is very high and for above mentioned reasons they are still not in practical use. In our system user can interface a camera with the device and image will be send to the server where it can be processed. Moreover some applications require such processing capability and software for precise results that it cannot be accessed at individual level but it becomes affordable at a community level.

Any new device like automatic injection system etc which comes to market can be easily plugged just by changing the configuration of output ports of the plug and play device. Doctors can alter the thresholds for triggering the outputs and then input sensor data will be processed and output device can be actuated as per the setting. It will foster remote patient care.

IEEE 1073 and continua health alliance stresses on interoperability of medical device information[1], but as per this approach now we don't have separate medical devices instead we have medical sensors and actuators which are plugged to a single device communicating with a single processor/server. Thus interoperability will be fostered.

II. SYSTEM OVERVIEW

The proposed system consists of three parts namely User end, communication channel/protocol, and Server end.

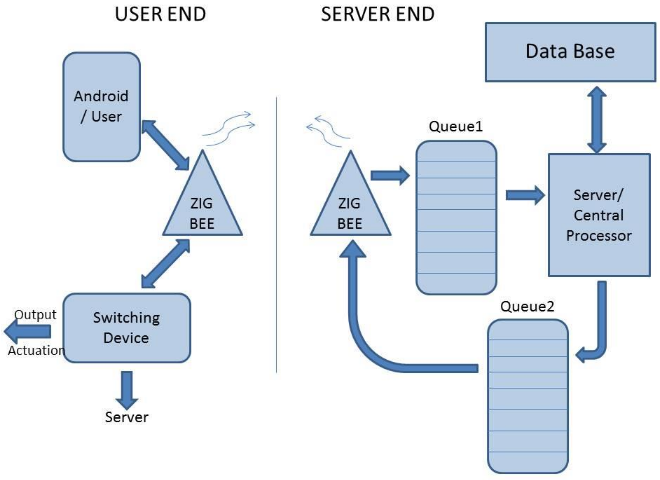

Figure 1. Block Diagram

A. User end

At the User end there are five main sub-parts
a) switching/plugging device b) Sensors c) Camera
d) Temperature and humidity sensors e) user interface

The switching device is a very basic microcontroller which has switching capability unlike the medical devices used today which require high end microcontroller for real-time processing and large memory and other peripherals like multiple timers etc. Here microcontroller is just used as a switching device and its cost is very low compared to the high end RTOS based controllers.

The basic purpose of this switching device is to provide input and output ports which can be driven as per need. The embedded code will not perform calculations here but will have functions which will receive the pin numbers and the time for switching the output ports. The code will be generic in nature.

Some ports will be predefined for specific sensor input and those sensors can be plugged here. For unknown sensors a specific port is defined and when any sensor is plugged into that port, user interface will prompt to enter the conversion-graph/ datasheet for that sensor. Now at the user interface a function is called to trace the points from the graph and these values will be sending over the communicating channel to the server end to generate conversion formula.

For detecting diseases now a day, image processing is used extensively, so we have interfaced a camera to the switching device which can help to capture image and image data can be transmitted to the server. If this system is to be implemented in a hospital, communicating protocol can be zigbee but for individual user communicating protocol can be internet. For zigbee based system image data needs to be reconstructed at the server end through matlab.

Temperature and humidity can play a vital role and alter the readings of the other sensors when uncovered sensors are used directly, so in this system we have a humidity and temperature sensor attached which sends the difference of temperature and humidity between the present and the past reading to the server and accordingly data received can be recalibrated for accurate reading. For the first time when device is turned on the exact temperature and humidity value is send as reference and afterwards this data is transferred only when differential output is large.

User interface is designed on android platform through which a user can configure the settings needed. As this system is generic, so at the output end any device can be interfaced and to acuate them, the user need to specify the sensor port, threshold level and the actuating frequency. For example if a user wants that whenever blood sugar level goes above 120 mark automatically injection needs to be given, now through this system user can select the port in which blood sugar monitoring sensor is plugged, level of blood sugar above which injection is to be administered, and the switching pattern of the motors involved in the injection system. This setting should be done only once for a specific device and device information can be retrieved from the datasheet or when the driver file is available it can be uploaded in the server instead of setting.

B. Zigbee

For communication between the user end and the server end, zigbee protocol has been used. Users end is configured as the end device and the server is configured as the coordinator. Both of them are programmed in the trans-receiving mode. The data packets and the signals are transmitted serially through this communication channel. Image captured by the camera at the user end is also transmitted to the server via zigbee.

C. Server end

At the server end, there are two sub-parts namely processor and the queue[3].

The data received from different users are stacked in a queue and one after the other is processed. Here the data and its context is also taken into consideration and if there is an scope for parallel processing then it is also performed. For example if a complex image processing is being performed in Matlab then simultaneously next data can be pre-processed and database operations can be handled.

Various functions are hosted here in matlab and each data coming from the user end (depending upon from which port/sensor data is coming) can call the particular matlab function. The whole control and dynamically calling of matlab is controlled through LABVIEW software. If at the server end a new code is to be implemented or up gradation of the code needs to be done then this can be easily done by changing the matlab code running inside labview.

Various preference orders have also been designed to facilitate users in case of emergency. A database is also implemented in MYSQL which is also interfaced with labview software. Here the user-id, data of the patient, configuration setting of the user end etc are stored.

Cost Analysis	
Cost of Typical Health Monitoring System	₹25000 (approx.)
Cost of Our Proposed System	₹20000(One time setup cost) ₹2500(per User)

Table 1. Cost Analysis

Performance Analysis		
Parameters	**Other Available Systems**	**Our Proposed System**
Time to measure Blood Pressure	25 Seconds	30 Seconds
Time to measure Glucose Content	20 Seconds	24 Seconds
Time to measure Body Temperature	10 Seconds	12 Seconds

Table 2. Performance Analysis

Error Analysis	
Devices	**Error Percentage**
Typical Devices (having integrated sensor and processor)	±2.5%[4]
Our Proposed System (having sensor at user's end and processor at server's end with temperature, humidity etc. recalibration)	±1%

Table 3. Error Analysis

No. of simultaneous users (possible load capacity):- 10 at prototyping stage.

III. SIGNALLING

There are mainly three types of data and control signals
1) Control signal 2) configuration data
3) Conditional data

Control signals are also of different types
1) Pin data: The sensors are plugged into the system but we sometime wish to monitor only some of the parameters while at other times we may prefer to monitor each of the parameters for which sensors are plugged. So, first of all the user needs to select the sensors desired to be monitored through their pin number. This selected data from the user interface forms a packet in which start bit contains the user id (which is a header) followed by the pin numbers selected and then the stop bit which is an indicator.
2) Acknowledgement: Some sensors need certain actuators to be switched on for capturing accurate data. For example blood pressure meter needs a motor actuation to pump air into the band before capturing the data.

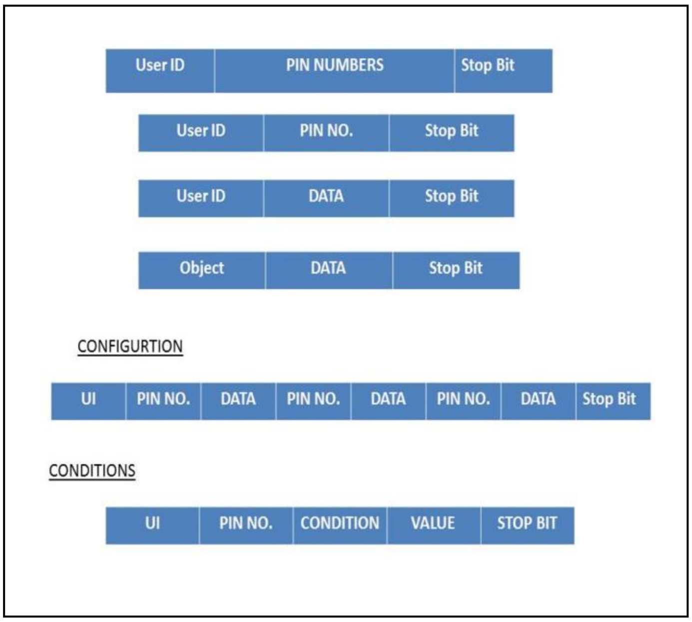

Figure 2. Signals Configurations

Once pin data reaches the server the pre-processor analyses the pin data and if it detects one of those pin where sensors need pre-actuation, then it sends back an acknowledgement which also contains the signal to make the output port where motor is attached to be high. Otherwise it does not send any acknowledgement. The user-end transmitter waits for thirty seconds for the acknowledgement. If it arrives, the transmitter passes the pin number to the function which makes that pin high else it starts to transmit the data from the sensor.
3) Data packet: The data packet contains a header bearing user-id followed by actual data.

Figure 3. Graphical User Interface

Configuration Data: This data can be configured by the user through the user interface. User selects the output ports in which the external device are interfaced which one desires to actuate after comparing the data of sensor with that of the threshold. The pin should be selected in the sequence in which we want to actuate them, if number of pins needed by the

device is more than one or we want to drive more than one devices.
Along with the pin numbers we need to specify the time for which we want the pin to be on. If the time for which output port needs to be high depends on the the data from sensor or the processed data, then we can simply select the '#' value.
Header containing user-id and stop bit is always attached.

Conditional data: To set the condition such as trigger when less than or greater than some threshold value[2], this data is critical. The conditional data contains user-id, pin number for which condition is to be set, logical operator and threshold value. This can be set by a doctor or the patient based on the doctor's prescription.

IV. Standards

The ISO/IEEE 11073 Medical / Health Device Communication Standards are a family of ISO, IEEE and CEN joint standards addressing the interoperability of medical devices. The ISO/IEEE 11073 standard family defines parts of a system, with which it is possible, to exchange and evaluate vital signs data between different medical devices, as well as remote control these devices.

IEEE 1073 protocol how it suit our purpose:

IEEE 1073 provides domain information model for almost all the point of care medical devices like blood pressure meter, glucometer etc. It also sets the communication models like baseband and polling. Continua Health alliance guideline specifies standards for patient bed-side devices. We have used all those device specific models and DIM in our project. Device specific standards are already present but when client-server type of application is implemented for remote calculations rather than calculations performed at the patient end and data stored at the server, as prevalent today; we need a control protocol which can select the DIM to be used for every client request.

The modification that has been done:

As we aim to remove all those devices from the patient end and replace them with a single data acquisition device with ports for plugging the sensors and communicating the raw data to the server where based on the port from which data is captured, a parameter calculating algorithm is called and parameter such as blood-pressure, glucose level I are calculated.
For this purpose, besides the DIM we need a source identification model (SIM)
For that a static array is defined which allot a id for each application
AttributeList := SEQUENCE OF AVA-Type

AVA-Type := SEQUENCE{attribute-id,
attribute-value ANY DEFINED BY attribute-id}.

Attribute Type Syntax: OID-Type ::= INT-U16

Attribute	ID
Blood pressure	0
Blood glucose	1
Temperature	2
Humidity	3

When the communication between the client and the server is established, along with the set-up information, this attribute id is also transmitted which allows the server to understand the source port id of data.
In the communication data format first three bits are used to specify the source and the rest of the sixteen bit data are used for user-id information.

Communication model:
IEEE 1073 specifies basic profiles that are used:
Polling and Baseline

In our system we have a mobile application which sends the initiating signal by sending the pin numbers from which data needs to be polled from the agent. Once the data exchange is initiated then we switch over to the baseline profile. The agent now sends the data only when there is a change and after a timeslot it again stops transmitting data and checks for polling information.
Actually for raw data train which are low frequency signals, a blend of both polling and baseline profile is needed. For this purpose we have used the 1073 communication protocols set. Only timing and triggering events are defined by us.
APSDE supports fragmentation and reassembly of packets and provides reliable data transport. Now we have used this fragmentation approach to distinguish between User id, actual sensor data and the differential environmental data (Temperature-Humidity) to recalibrate the actual data based on the Temperature-humidity change. Application object represents different application types (or profiles) that can be defined on a single Zigbee device[4]. As our system is a generic and aimed to support various medical parameter calculation and related actuation, so we have defined nearly twenty application objects such as blood-pressure, body temperature, blood sugar etc. In the context of a profile, a group of related attributes is termed a "**cluster**" and identified with a **clustered**. Typically a cluster represents a sort of interface (or part ofit) of the APO to the other APOs. In ourcase, these application objects are part of one or more clustered.
The recalibration is done using the information from the Source Information model which also provides information about the type of sensor. For different sensor type, variance of

the data due to variation in temperature-humidity is different. For this we have a case structure, provided vendors maintain the variance due to temperature-humidity as per the protocol for a particular type.

There are different types of sensors available and when we tried to implement server based parameter measurement system, problem arose which was associated with the sensor types because for some sensors variation may be linear while for others it may be non-linear. But when a data packet arrives at the server, it has information about the parameter to be measured and by its virtue we can call the appropriate function. But due to variation in sensor technologies, we also need information about sensor type which must be defined and transmitted by the client with the data packet. For this a separate sequence is made.

Sensor type ::= SEQUENCE OF AVA-Type
AVA-Type ::= SEQUENCE { sensor type_id,
sensor type_value ANY DEFINED BY sensor type_id }
sensor Type Syntax: OID-Type ::= INT-U16

Sensor type	ID (decimal representation of bits,some are reserved for future types , if any)
Resistive	0
Capacitive	1
Positive thermistors	2
Negative thermistors	3

V. IMPLEMENTATION AND RESULT

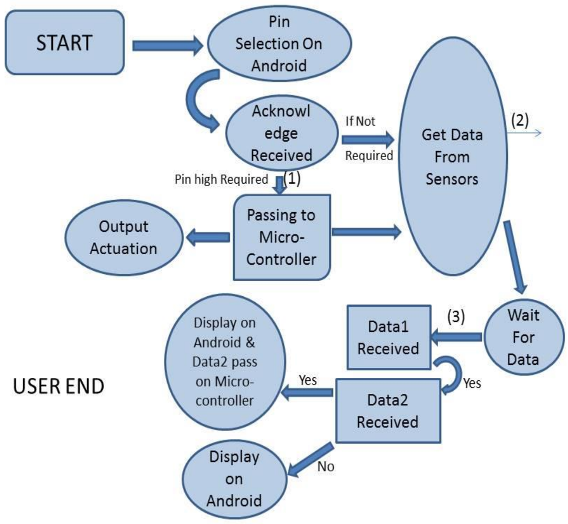

Figure 4. User End Algorithm

We build a prototype which consists of a plugging device consisting of input ports for sensor plugging, output ports where any device such as automatic injection, infusion pumps etc can be interfaced and microcontroller for driving these devices. The microcontroller is very basic which does not have high processing capability, large memory and do not support RTOS, thus quite cheap as we only intent to use its switching ability.

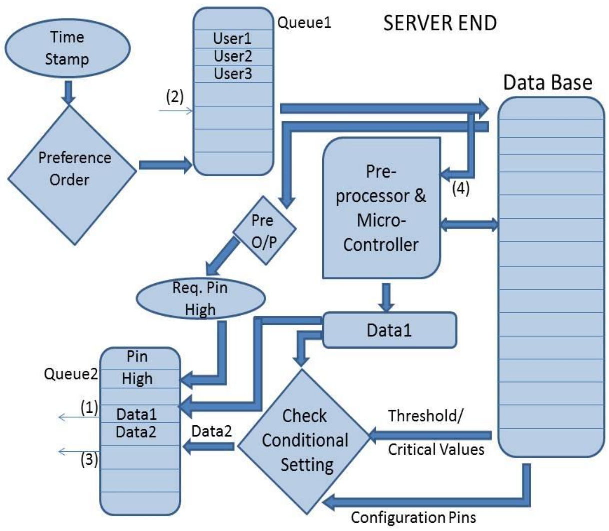

Figure 5.Server End Algorithm

It is coded such that it gets the input and passes the data to Zigbee module and accepts the data returned from the trans-receiver to acuate the output devices. At the user end we have also designed an Android mobile application, through which user/doctor can set the threshold value or condition which should trigger the output device so that patient care can automate, user can also set the configuration setting of output and the sensor from which data is desired to be checked.

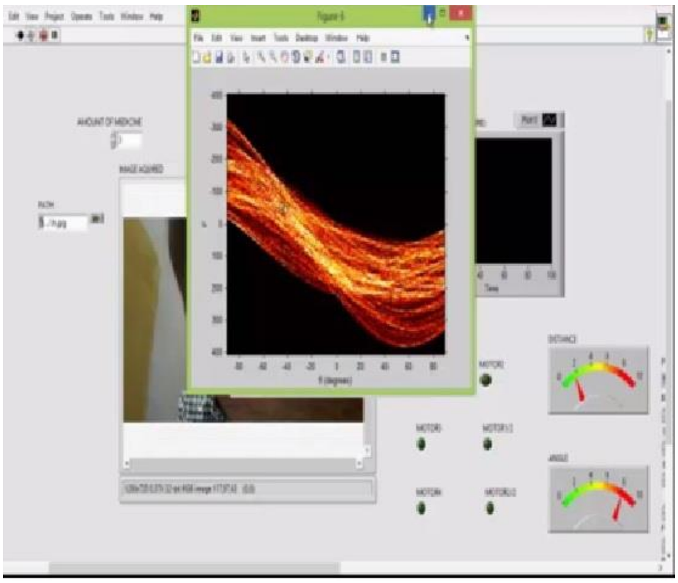

Figure 6. Running Matlab Through Labview

Display is also integrated in that application and it directly communicate with the transreceiver at the user end to transmit data and storage at the database at the server end.
Two zigbee modules are used, one at the user end and the other at the server end. Both acts as transreceiver.
At the server, a queue, a labview application which has various matlab code inside and depending upon the sensor plugging pin number (which acts as sensor id), appropriate code is executed and various filters, amplifiers are executed to derive the parameters. We have implemented blood pressure calculating algorithm, cancer detection algorithm using Gabor filter and an Infrared based algorithm to detect vein from the patient's image captured using an infrared camera[5].

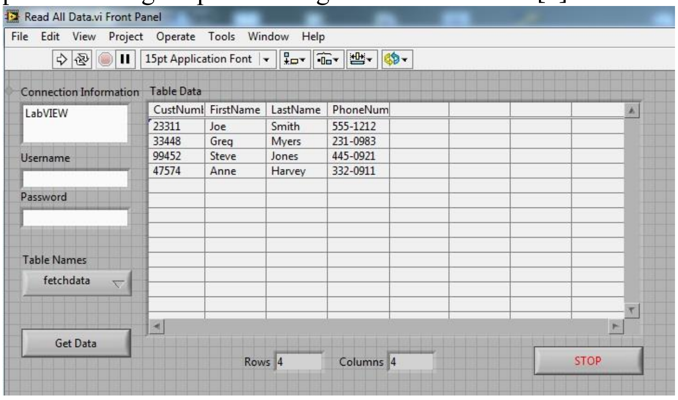

Figure 7.Database Interaction Using Labview

Application example:

Administration of intra-venous drug to the patient's body can be carried out by using an infra-red camera and then processing it using edge detection and Hough's transform.

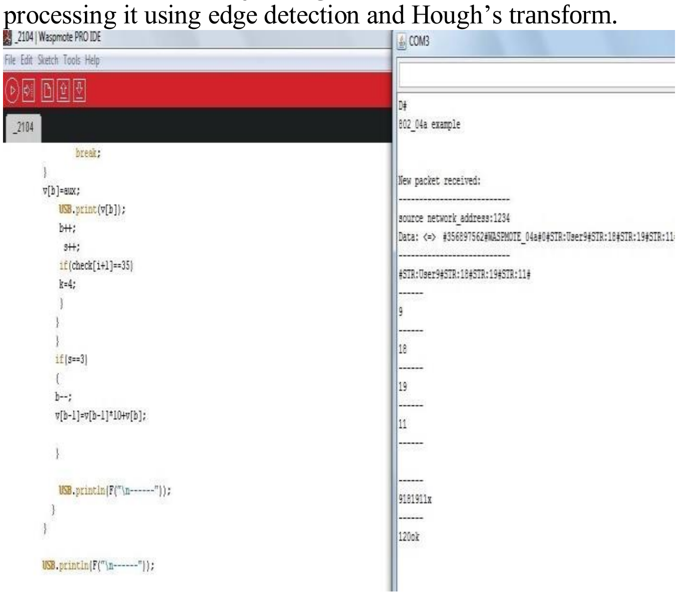

Figure 8.Zigbee Object Profiling

Now when user selects the camera pin from the android application, a pop-up dialog appears asking for the intended application. When user selects automatic drug delivery, and inserts hand, the motor connected with camera aligns itself. The image is captured and transmitted to the server where pin number is checked from the previous signal and appropriate code is called from labview and tat Matlab code finds out the appropriate position where injection is to be administered by considering the surface of the patient's body to be a Cartesian coordinate system. It then sends the data(r, theta) to the microcontroller via zigbee to actuate the motors attached to the automatic injection system.
The manner of actuation can vary from one device to the other, so there are two provisions available for the first time user. User can either upload the device driver to the server or user can configure the sequence of operation and respective delays and duty-cycle from the android application.

VI. CONCLUSION AND ADVANTAGES

The existing health monitoring systems uses a data logging device connected to various patient monitoring devices which transmits data to a central data repository which is analysed either automatically or manually. The presented system uses those basic blocks but instead of the devices which have sensors and processors to calculate the parameters like blood-pressure, blood sugar etc, we have only sensors with the user, a common plugging device and the parameter is calculated at the server end and then analysed. Further there is a scope for triggering a device like automatic injection system, alarm etc based on the value of the parameter calculated or the condition combining multiple parameters and the threshold level.
In terms of standards which emphasise on interoperability, object based model we have devised a system which can accept any sensor and also recalibrate the reading using the temperature-humidity feedback system and can be used to interface any device for patient care. User has access to configure the way of operation, condition which makes it more customisable. It is a common platform and all the device specific, object oriented standards and terminologies can be implemented on a single server which lowers the cost of health equipments and home health monitoring and interoperability is fostered.
With this system it is very easy to implement a new algorithm whereas till today when a new algorithm is accepted for its higher efficiency, it is very difficult to be introduced for public use because the codes are embedded into device memory and up-gradation can only change some of the features, but hardware cannot be changed. Also it takes a lot of time.

REFERENCES

[1] The Institute of Electrical and Electronics Engineers, Inc. *Health informatics - Point-of-care medical device communication - Application Profiles*. ISO/IEEE 11073-20000.The Institute of Electrical and Electronics Engineers.

[2] J Gerard J. Holzmann. The model checker Spin. *IEEE Transactions on Software Engineering*, 23(5):279–295, May 1997.

[3] European Committee for Standardization (CEN). *Health informatics - Interoperability of patient connected medical devices*, 2000. ENV 13735: 2000 E.

[4] Ed Callaway, Paul Gorday, Lance Hester, Jose A. Gutierrez, Marco Naeve, Bob Heile, Venkat Bahl, "Home Networking with IEEE 802.15.4: A Developing Standard for LowRate Wireless Personal Area Networks", IEEE Communications Magazine, August 2002.

[5] Paolo Baronti, Prashant Pillai, Vince Chook, Stefano Chessa, Alberto Gotta, Y. Fun Hu, "Wireless Sensor Networks: a Survey on the State of the Art and the 802.15.4 and ZigBee Standards", Computer Communication, Volume 30 , Issue 7, pages 16551695, 2007.