ijiTECH International Journal of Innovative Technologies
WWW.IJITECH.ORG
ISSN 2321-8665
Vol.05,Issue.01,
January-2017,
Pages:0100-0104

WSN based DME's for Patients Health Monitoring System During Natural Disasters

SD. RESHMA[1], CH. SURESH[2]
[1]PG Scholar, Dept of ECE, Andhra Engineering College, JNTUA Anantapur University, AP, India.
[2]Associate Professor, Dept of ECE, Andhra Engineering College, JNTUA Anantapur University, AP, India.

Abstract: Electricity-operated durable medical equipment (DME), such as ventilators, dialysis machines, and patient monitoring devices, are life-supporting machines used extensively by patients at home. While convenient and economical, at-home use of DME is susceptible to power outages, especially the ones caused by natural disasters that often occur in large area and for a long duration. There is little existing technology allowing hospitals to monitor DME-dependent patients without using the current infrastructure, such as the landlines, the cell towers, Ethernet cable or the Internet. Reported herein is a novel wireless system that utilizes a radio ad hoc network to automatically report the patient's information and location, and the DME information and status to a nearby hospital when a power outage is detected. This system consists of two parts: a hospital-based receiving device, called the Base Station node, and multiple transmitting devices, called User Nodes, each connected to the DME at patients' homes. The Base Station and User Nodes is each built with a Teensy® microcontroller, a GPS receiver module, and an Xbee® radio implementing the Zigbee® protocol. Additionally, each User Node contains a status LED and an internal lithium-ion battery connected by a charge controller. User Nodes are programmed to obtain the GPS location of the patient, monitor the DME status, communicate with nearby nodes, transmit the data and relay information to the Base Station through the radio ad hoc network the nodes form in the case of a power outage. The Base Station device is programmed to receive and convey the information transmitted from the User Nodes to a nearby hospital's patient monitoring computer through a USB connection. This system works without relying on the infrastructure, and allows hospital staff to know the information and locations of DME and their users and provide help needed during power outages.

Keywords: Durable Medical Equipment (DME), Assistant Secretary for Preparedness and Response (ASPR).

I. INTRODUCTION

Durable medical equipment (DME) is any medical device used at home by patients for monitoring and/or treating diseases [1]. There are two types of DME: passive equipment and active equipment, the latter reliant on electricity to operate. Life-supporting active DME include dialysis machines, ventilators, oxygen concentrators, etc. [2]. At-home use of DME is not only convenient and economical, but also leads to a better quality of life for the patient. Despite aforementioned benefits, at-home DME are susceptible to power outages, especially those ca used by natural disasters. During difficult times like this, the DME dependent patients had to face the life-threatening situation because their machines had stopped functioning. While most at-home DME are equipped with integrated batteries to keep them functioning during power outages, their rechargeable batteries typically last only 1 hour with lead-acid batteries and2-3 hours with newer lithium-ion batteries [4]. Thus, there is a critical need for a means of communication between the medical staff at a hospital and patients at home during natural disasters without needing current infrastructure such as landlines or cell towers that are often unavailable during natural disasters. Aware of the severity of this problem, the Assistant Secretary for Preparedness and Response (ASPR) of the U.S. Department of Health & Human Services through its partner, www.innocentive.com, launched a challenge in 2013to seek ideas that might solve this communication issue [5].

II. MATERIALS AND METHODS

A. Materials

The Xbee® shield was purchased from Sparkfun [8]. The Teensy® 3.1 development board was ordered from PJRC [9].The GPS shield is a generic version and bought from Ebay. All other parts and tools were obtained from Radio Shack ®and Home Depot®.

B. Engineering goal

To solve the communication issue when the infrastructure is disabled and help DME-dependent patients, it was proposed to engineer a novel DME tracking and reporting system based on wireless nodes with radios following the Zigbee® (IEEE802.15 standard) pecifications [10], operating at the frequency band of 2.4 GHz and consuming little power, though transmitting at short distances and at low data rates[11]. This system would comprise of two parts: multiple transmitting devices located in patients' homes and connected to patients' DME, called User Nodes, for gathering relevant live data to be transferred to the hospital and one central receiving device located in a local hospital, called the Base Station, for collecting the patient and DME information sent by the User Nodes. The system would have a modular design and scalable implementation, providing flexibility for further optimizations (for example, substituting the current radio module with other radios that have the same interface).The information includes patient information (i.e., name, age, disease, type and brand of DME being used, etc.), GPS location of the patient and DME, and the power outage status (i.e. how long the DME has been running using battery power and how much battery life is remaining). All patient data and information would be encrypted with symmetric-key encryption so that only the administering hospital could receive and decrypt the information in compliance of the HIPAA laws [12].

C. Hardware Design and Assembly

The proposed hardware was based on the Teensy® version3.1 Development board [9], having an MK20DX256 32-bitmicroprocessor based on ARM Cortex-M4 and 256KB of flash storage and 64KB of RAM. This board has 3 standard asynchronous serial ports (protocol: 8 data bits, 1 stop bit and no parity), in addition to a USB programming ports capable of transmitting data in 4800 Baud increments. These serial ports were used to communicate with the radio, receive GPS information and retrieve information from the DME. This board was chosen due to its easiness of prototyping and relatively low cost. Additionally, it is programmable with the C language, which was used to program all User Nodes and the Base Station. The radio used to communicate among modules was the Xbee® Pro Series 1 Point-to-Point device for communicating with IEEE protocol 802.15.4. It has a rated power output of 1mW and can transmit 90 m within line of sight, either broadcasting its information or uni-casting to a radio whose serial number is the same as the set destination address [13].Moreover, the module inputs and outputs data through astandard serial interface at 115.2 kilobits per second. The GPS module used in this prototype is a generic GPS board that uses the u-blox6 GPS module and interfaces with the microcontroller through serial interface at 9600 bits/second[14]. Its horizontal position accuracy is within 2.5 m and outputs its data with NMEA GPRMC statements, which includes geographic location and UTC time. Fig. 1(a) shows the hardware design of the User Node and Base Station, both using the same assembly. Fig. 1(b) is a picture of the physical setup of the hardware.

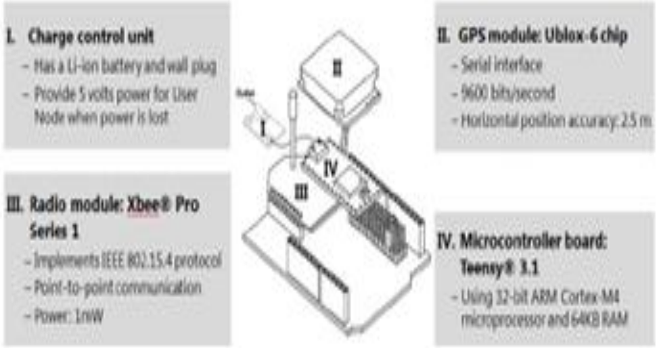


Fig.1(a). Hardware Design of the User Node and Base Station.

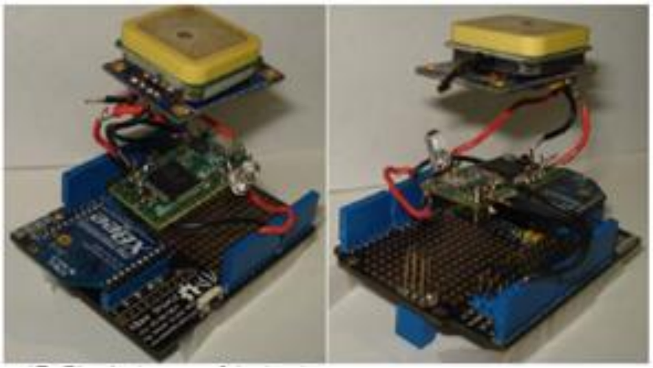


Fig.1(b). Physical Setup of the Hardware.

D. Low Level Software Design

The program for the User Nodes was designed first. In order to have a platform on which to program the algorithm for implementing the routing protocol, low level drivers needed to control the Xbee® radio module and the GPS were written as the manufacturers either did not provide the driver code or/and the code was not well documented. After each driver was created, the driver algorithms were tested for accuracy and bugs, with each of them being debugged after being created as shown in Fig.2. Firstly, a generic serial reader code was created to access the serial interfaces which was able to read the serial ports for data and output the data in variable string Out, containing a maximum amount of characters defined at the beginning of the code by the constant, MAX_READ. Since the serial port is asynchronous and the node is without knowledge of the size of the upcoming information, the Serial was read at the same rate at which it was being filled, delaying the reading to accommodate a few characters to fill the 64-Byte Serial buffer. This is to ensure that the Serial buffer neither overflows nor becomes prematurely empty, at which point the reading stops and truncates the other information left in the Serial buffer. All codes were written modularly to support customization according to the platform and the usage. The serial reader code was a dependency for both the Xbee® radio driver and the GPS radio driver as it read their outputs.

```
typedef struct
RequestPacket{
uint32_t packetType;
uint32_t myHaddress;
uint32_t myLaddress;
uint32_t
magicNumber;
}
RequestPacket;
```

```
typedef struct
ReplyPacket{
uint32_t packetType;
uint32_t myHaddress;
uint32_t myLaddress;
float Latitude;
float Longitude;
uint32_t
magicNumber;
}
ReplyPacket;
```

```
typedef struct DataPacket{
uint32_t packetType;
uint32_t myHaddress;
uint32_t myLaddress;
float Latitude;
float Longitude;
float destinationLatitude,
float destinationLongitude;
char data[256];
uint32_t magicNumber;
}
DataPacket;
```

Fig.2. fundamental data structures used in the routing protocol implementation.

E.High Level Software Design & Protocol Implementation

While there is documented, proprietary software to create amesh network using the Xbee® radios known as DigiMesh®[18], such a protocol for routing was not used for the interest of providing a modular platform on which other, more efficient routing protocols could be easily implemented [19].The algorithm implemented is a greedy geographic routing, whereby each node polls nearby nodes and the data is sent to the node closest to the final destination [20]. The algorithm is stateless and reactive, with the nodes not requiring information about previous nodes the packet was routed through and also dynamically routed. However, due to the modularity of the program, the routing algorithm can be easily changed. Following the routing protocol, three types of information packets were used to perform the routing: the Request packet, the Reply packet, and the Data packet; which were in the form of a struct instance that was cast into a string before sending. The data structures of the three packets are shown in Figure. The algorithm implemented runs continuously through a loop and polls the inputs and outputs. It was planned and implemented through the high level functions. Fig. 3 is an overview of the algorithm, including the flags and variables.

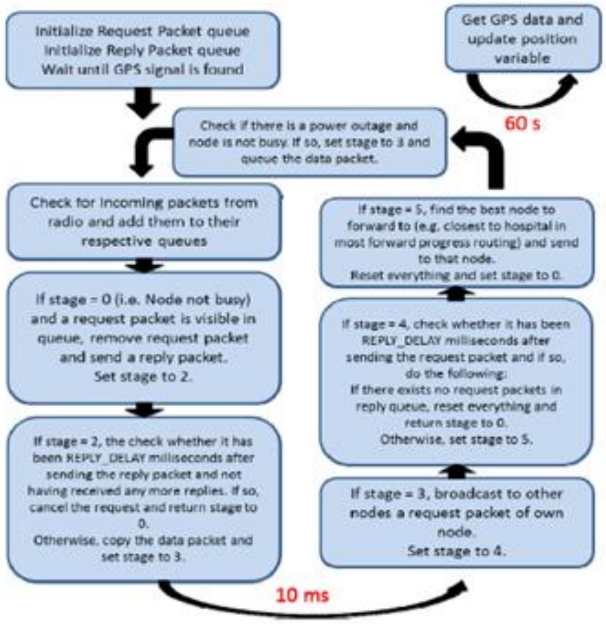


Fig.3. An overview of the algorithm.

Variables and Constants:

- stage = 0 when the node is not busy with any routing or sending of data
- stage = 2 when the node is waiting for a packet
- stage = 3 when there is a data packet to send and the node has sent out a
- request packet
- stage = 4 when node is waiting for replies to its request packet
- stage = 5 when the node has received request packets and has found a forwarder.
- REPLY_DELAY is the number of milliseconds node waits for replies.

F. Power Consumption Measurement

To measure the power consumption of the DME tracking device, one node was connected in series with nine 1Ω ± 5%resistors, connected such that they are in 3 parallel groups of 3series resistors with a final equivalent resistance of 1Ω ± 5%.While not decreasing the error, this arrangement was used to narrow the distribution of possible values towards the mean(i.e. create a higher probability that the resistor is closer to its marked value). Using an ATMEGA328-based microcontroller, the voltage across the resistive load was measured to the nearest millivolt and outputted through the USB serial port at a5-ms interval. Simultaneously, the node performed either a standard data transmission or a GPS updating, in which it updates the latitude and longitude values, in addition to the UTC time based on NMEA sentences from the GPS module. The node in each of the tests was powered by a regulated 5Vsource. The wiring diagram of power consumption test was shown in Fig. 4.

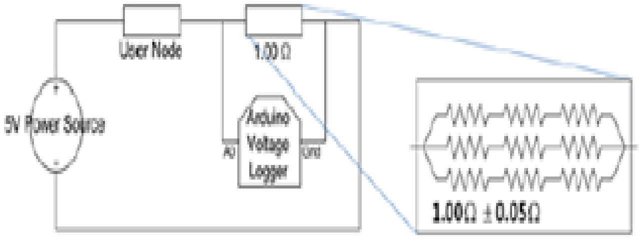


Fig.4. wiring diagram of power consumption test.

G. Field Test

In order to ascertain the functionality of the routing protocol, as well as to determine the time used to relay the data packet, a field test was conducted, in which 5 User Nodes and 1 Base Station were placed linearly at a distance 90 meter sapart from each other such that each node could only receive and transmit to adjacent nodes (Fig.5). The distance of 90meters was used based on the radio range of the Xbee® module. The nodes were each tested such that they could only transmit to their immediate neighbors. Additionally, the software of each node was modified by adding in a timestamp parameter into each of the data structures. This timestamp was used to collect information about the length of time taken for each transmission. This time stamp information was collected from the GPS module. The getGPS function was slightly modified to also record and update the UTC time variable in the program. The UTC time was found to the nearest second on the start of the second as the GPS module only outputs data on the beginning of every second. Since the GPS is read every 60 seconds, there needed to be a function written and tested to interpolate the exact UTC time, in addition to a milliseconds extension, to accurately record the time taken for information to be transferred as shown in Fig.7. The time interpolation function takes into account the time interval recorded by the Teensy® since the last updating of the UTC time, and adds the time to the UTC time and the milliseconds extension. To test the accuracy of the interpolated time, two nodes were loaded with the same

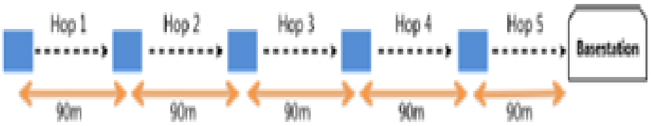


Fig.5. Experimental setup in field test.

program, and were programmed to output their exact interpolated time via USB serial onto separate computers. Then, the second computer was connected to via remote desktop with a known latency such that the two serial windows were adjacent on a screen. Multiple screenshots of the two adjacent windows verified that, compensating for remote desktop latency, the two nodes displayed the same time, shown in Fig. 6.


```
21:15:40.583
21:15:40.593
21:15:40.603
```
Remote computer: 67ms latency
```
21:15:40.623
21:15:40.633
21:15:40.643
21:15:40.653
21:15:40.663
21:15:40.673
```

Fig.6. UTC time measurements from 2 nodes to verify the relative accuracy of the time interpolation function.

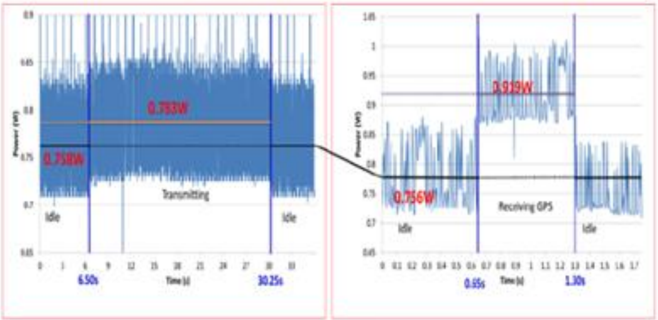


Fig.7. Power drawn from the user node for idling, relaying packets and receiving GPS.

III. RESULTS AND DISCUSSION

A. Field Test Results

The results from the field simulation test were summarized as shown in Figs.8 and 9.

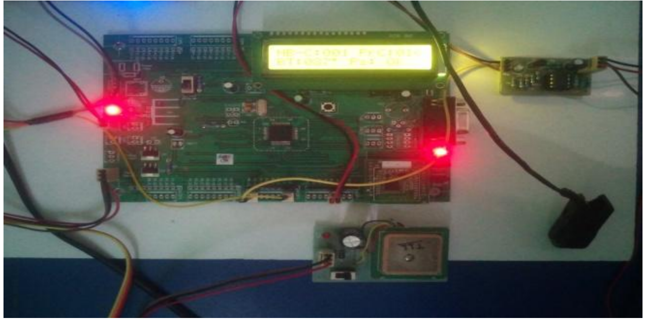
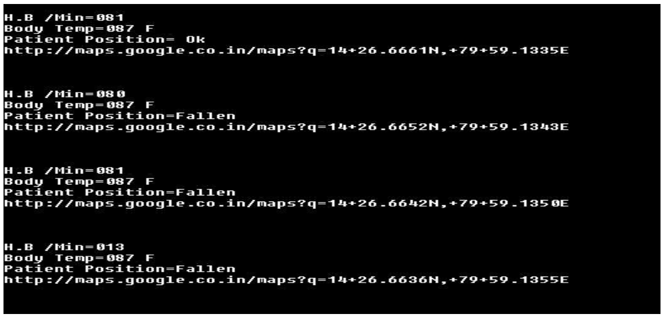


Fig.8.


```
H.B /Min=081
Body Temp=087 F
Patient Position= Ok
http://maps.google.co.in/maps?q=14+26.6661N,+79+59.1335E


H.B /Min=080
Body Temp=087 F
Patient Position=Fallen
http://maps.google.co.in/maps?q=14+26.6652N,+79+59.1343E


H.B /Min=081
Body Temp=087 F
Patient Position=Fallen
http://maps.google.co.in/maps?q=14+26.6642N,+79+59.1350E


H.B /Min=013
Body Temp=087 F
Patient Position=Fallen
http://maps.google.co.in/maps?q=14+26.6636N,+79+59.1355E
```

Fig.9.

The field test not only shows that the routing algorithm could be viably implemented, but also shows the modularity of the software, as the data structures were modified without needing to modify any other parameters.

IV. CONCLUSION

Given testing data, it was found that the prototype design of the DME tracking system was feasible to implement and would meet the requirement for securely transmitting patient data, location information, and the status of DME to a near by hospital during power outages. Although the maximum radio range for the current pilot prototype was found to be 90 m, the advantage of modular design allows this proof-of-concept system to be easily scalable by simply employing more powerful radio modules or having specially placed forwarding nodes to facilitate the forwarding of information from more distant homes. In a medium patient density situation, for instance, a radio with an indirect (i.e. non line-of-sight) range of >4.70 km could be employed. Additionally, the implementation of this DME tracking system is relatively inexpensive, utilizing commercially available low cost general-purpose microcontrollers and general-purpose radios. When produced in one circuit in mass production, the cost will be even lowered.

V. REFERENCES

[1]Durablemedicalequipment.http://en.wikipedia.org/wiki/Durable_medical_equipment

[2] National Coverage Determination (NCD) for Durable MedicalEquipmentReferenceList(280.1).http://www.cms.gov/medicarecoveragedatabase/details/ncddetails.spx?NCDId=190&ncdver=2&NCAId=27&NcaName=Electrostimulation+for+Wounds&IsPopup=y&bc=AAAAAAAAAgAAAA%3D%3D&

[3] Baseline country survey on medical devices, 2013 update: MedicalEquipment−Total density per million population computed tomography.http://gamapserver.who.int/gho/interactive_charts/health_technologies/ medical_equipment/atlas.html

[4] B. Norman. What Will You Do if the Power Goes Out?http://alsn.mda.org/article/what-will-you-do-if-power-goes-out

[5] HHS Press Office. HHS selects winners in idea challenge foremergencyresponse.http://www.hhs.gov/news/press/2014pres/02/20140220a.html

[6]GPS Trackers for Today’s World. http://www.pocket-finder.com/

[7] Zoombak Personal GPS Locators. The next generation ZoombakiseZoom.http://www.securusgps.com/zoombak.aspx

[8]SparkFun Xbeeshield.https://www.sparkfun.com/ products/12847

[9]TeensyUSBDevelopmentBoard.https://www.pjrc.com/teensy/

[10] ZigBee® Alliance. http://www.zigbee.org/

Author's Profile:

Sd.Reshma is currently received her B.Tech Degree in Electronics and Communication Engineering in Atmakur Engineering College from JNTUA Anantapur, Anantapur in 2014and pursuing M.TECH Degree in Embedded Systems in Andhra Engineering College, Atmakur, affiliated to JNTUA, Anantapur in 2016.

Ch.Suresh has received his B.Tech EIC in N.B.K.R.I.ST from Sri Venkateswara University, Tirupathi in 2007. M.Tech Degree in Embedded Systemsfrom National Institute of Electronics and Information Technology (NIELIT), Calicut University. In 2009 Respectively. He is dedicated to teaching field from the last 5 years and 2 years Industry Experience. At present he is working as Assistant Professor in Andhra Engineering College, Atmakur, S.P.S.R Nellore Dt.