Journal of Engineering and Technology for Industrial Applications, 2017. Edition. 09.Vol: 03
https://www.itegam-jetia.org
ISSN ONLINE: 2447-0228
DOI: https://dx.doi.org/10.5935/2447-0228.20170021

Monitoring System of Vital Signs using Zigbee Communication

Eiji Daniel Rodrigues Maeda[1], Moisés Pereira Bastos[2]

[1]Universidade do Estado do Amazonas. Avenida Darcy Vargas, 1200. Parque 10. Amazonas - Brasil.

Email: maedaedr@gmail.com, mpbastos@uea.edu.br;

Received: February 13th, 2017

Accepted: February 25th, 2017

Published: March 30th, 2017

ABSTRACT

Vital signs are those elements that characterize the being is alive. Variations, sometimes even if small, may indicate problems that, if are not perceived, can aggravate and endanger the person. The problem is that these signs are not easily discernible, much less to the naked eye, making it difficult to monitor these changes in pattern, making a late rescue. Thus a system was developed with the function to do this monitoring, alerting when detect a variation and indicating the possible problem. Was used in the project an Arduino Uno microcontroller and a zigbee communication network in order to use a means of communication wireless low power consumption and reliable transmission. The project therefore aims to develop a monitoring based on standard vital signs, sending to a central control an alert if there is a problem and indicate the possible failure, accelerating decision making and giving a better chance at survival of the monitored patient.

Keywords: vital signs, monitoring, zigbee.

I. INTRODUCTION

Even a patient who doens't has a serious symptom in a moment can suddenly have a change in the clinical condition in minutes. Mild discomfort may be the beginning of a more serious problem. Without constant monitoring is a risk the patient runs. But it is totally impractical the idea of a technician, nurse or doctor checking the vital signs of each patient steadily, paying attention to all.

There is equipment in hospitals that do constant monitoring, but are typically large and restricted to wards where the clinical status of the patient is already aggravated and it is really necessary monitoring of vital signs.

The monitoring system proposed in this paper, for testing purposes, aims to make reading body temperature, monitoring the same in real time.

Among the signals indicative of life, the temperature was chosen for better perception and easy comparison with other measuring devices (mercury thermometers and digital thermometers). Thus the data obtained are more robust and reliable, enabling better development of the system for the future introduction of other sensors.

An Arduino Uno was used, because is small and low power consumption, and is compatible with a device for sensing vital signs of Cooking hacks. This device has the ability to read 9 different signals , 8 of them simultaneously. The reading of body temperature is handled by a Model 405 Air Temperature Probe, a sensor device provided by the developer.

If any variation of healthy pattern of temperature occurs, an alert is sent via wireless zigbee communication using two XBee PRO 802.15.4 modules for sending and receiving data. This alert is received by a central computer, which will check the condition through a supervision system and notify the competent responsible.

II. MATERIALS AND METHODS

The developed system has the function of monitoring in real time the condition of the patient. With this monitoring is possible to detect variations in patterns and send an alert by transmission module. This system consists of the Arduino Uno, a

sensor to read temperature, a shield eHealth and Xbee module who will make the data transmission, as the figure 1.

II.1 ARDUINO

The Arduino Uno was initially designed with the intent of being a handy microcontroller and at the same time be a platform that anyone could use them. It soon became popular and started to be used to create new projects and to introduce easily the programming microcontrollers [1]. Arduino is a programmable microcontroller that can interact with the ambient using hardware device combined with its software. This programming is done through the IDE.

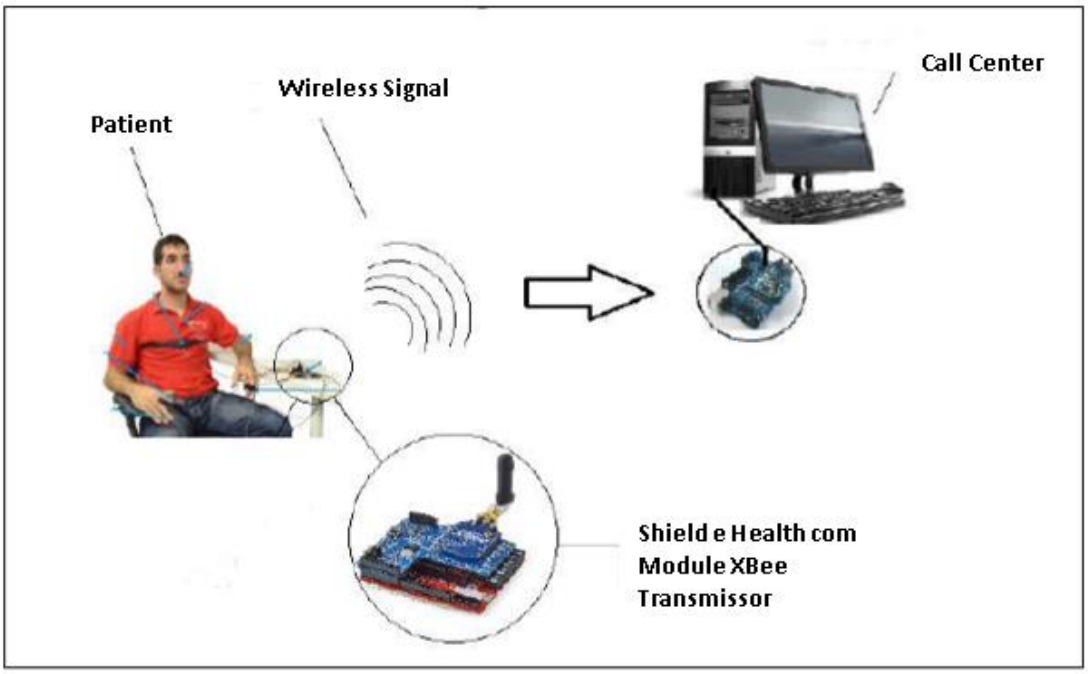

Figure 1: Diagram of the System Architecture.
Fonte: Autores, (2017).

(Integrated Development Environment), a software that uses based programming in C. With this IDE the developer can pass a set of instructions that the Arduino will understand and execute [2]..

The Arduino is also open hardware and software, which means that can be used by anyone, and any device. This feature allows the use of various types of "shields", which are circuit boards that are coupled to add functionality in Arduino [2].
It also has an integrated voltage of 3.3 V, which helps the stability of protections that caused problems in older models [1]. This model is also compatible with electronic card required for reading sensors, which was vitally important in choosing..

II.2 SHIELD EHEALTH SENSOR PLATAFORM V2.0

The eHealth Sensor Platform v2.0 of Cooking Hacks, presented in figure 2, was released in August 2013 with the intention of providing the academic community with the tools necessary for the development of new applications and products in the medical field [3]. For powering during the tests we use the USB. However some USB ports may not be able to provide the necessary current. If this happens you can also use an external source (12V - 2A) [3].

The eHealth has a library based on C ++ programming language which allows the interface between the program and the sensors. It is necessary to download the library, available on the Cooking Hacks page.

After the download completes, the file should be unpacked and placed on the Arduino library folder located in the case of Windows operating system in "MyDocuments\\Arduino\\libraries". The libraries (.cpp and .h file) will not work if they are placed directly in the folder or if they are placed in an extra folder. To be sure of the success of the installation, it was found that the new libraries appeared in "Sketch -> Import Library" in the Arduino IDE.

However, it should be appreciated that these libraries have been developed for Arduino version 1.0.1. Therefore, some features may not work for newer IDE's (1.6.x), and should use versions 1.0.x.

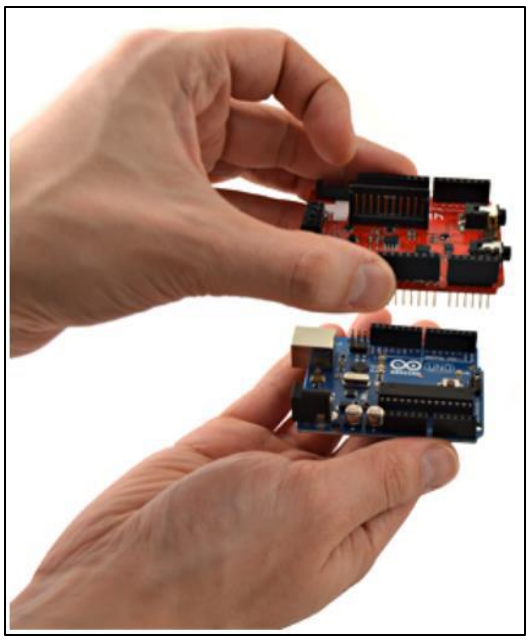

Figura 2 Using eHealth with Arduino UNO.
Fonte: [3].

II.3 BODY TEMPERATURE READING

The body temperature reading needs attention to factors who influence the data, like the place where the sensor will read the temperature, the period of the day and the activity level of the person. In a health adult the temperature oscillates about 0,5 Celsius degrees during the day, with the lowest temperature in the morning and the highest in the late afternoon [3]. In general, the human temperature can be divided in 4 groups as is showed in the table 1.

Table 1: Temperature Scales.

Hypothermia	< 35 °C (95 °F)
Normal	36.5 – 38.3 °C (97.7 – 99.5 °F)
Fever or hyperthermia	> 37.5-38.3 °C (99.5 – 100.9 °F)
Hyperpyrexia	> 40.0-41.5 °C (104 – 106.7 °F)

Fonte: Autores, (2017).

The programming is made with calculus using values of the Model 405 Air Temperature Probe table. If the sensor is a different one of the mentioned, parameters changes are needed inside the eHealth.cpp library. The temperature measurement is made by tension measure in a Wheatstone Bridge. Although the pattern values in the programming provide acceptable values, a more reliable reading is possible through a resistance calibration [3]. This calibration is made measuring the bridge values and the reference tension (Figure 3), changing the values in the eHealth.cpp library if needed (Figure 4).

A digital thermometer was used to compare temperature with the reading in the eHealth sensor. With this additional tool was possible compare and evaluate the real body temperature and the sensor measuring, which need to be properly calibrated to a satisfactory reading.

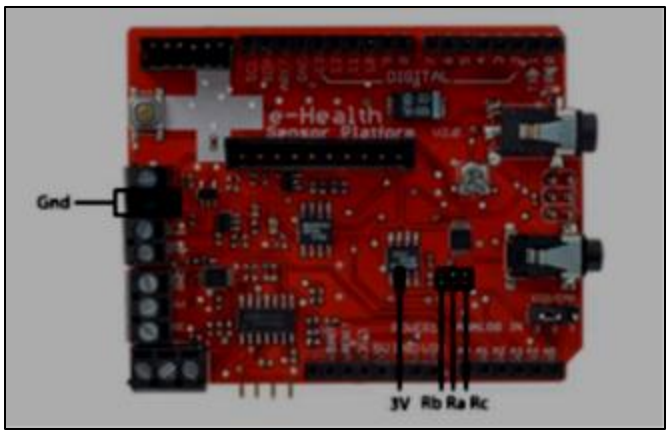

Figure 3: Whetstone bridge resistances in a eHealth shield.
Fonte: Autores, (2017).

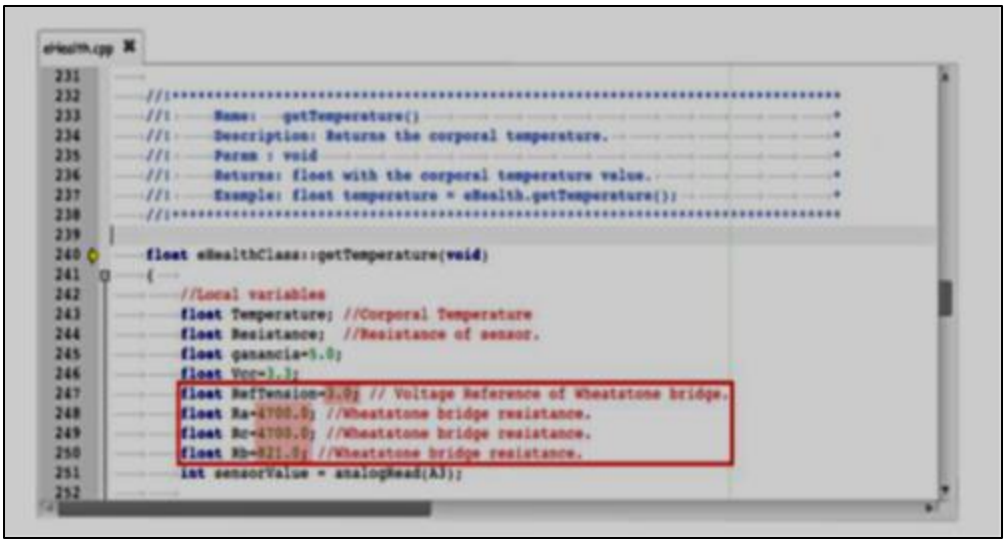

Figure 4: Calibration Values.
Fonte: Autores, (2017).

II.4 ZIGBEE WIRELESS MODULE

A XBee PRO module was chosen to this project. The XBee PRO is based in the 802.15.4 protocol and was chosen because has a low power consumption, a satisfactory reach (more than 100 meters depending of the antenna) and the project doesn't need a large bandwidth. A previous configuration is necessary to the communication. The modules need to be in the same network and channel, the module destiny address needs to be set as well (DH and DL parameters), to determined who will send and who will receive the datas.

III. RESULTS

The system was tested before the calibration (with the standard resistances values) and after the calibration. In the first case, 5 patients had the temperature monitored by 5 minutes each. The test was made 3 times with each patient in different periods. In the first monitoring, the values were the developed standard. The reference tension was setted in 3.0Vand the resistances Ra, Rb and Rc were 4700, 821 and 4700 respectively. In this case the sensor measuring was about 1 ºC different of the digital thermometer measuring (next 36.5 ºC). The test result is showed in the table 2.

Table 2: Test with standard resistances values.

	Test 1	Test 2	Test 3
P1	35,23	35,34	35,14
P2	35,35	35,72	35,62
P3	35,81	36,14	35,87
P4	35,36	35,88	35,35
P5	36,25	35,52	35,68

Fonte: Autores, (2017).

A multimeter was used aiming the reading of the real resistance value, but the precision was not enough to an adequate read, generating variations between 15 and 45 ºC.

A new read was made with other multimeter. The temperature stabilizes, but the read indicates next 34,3 ºC (hypothermia), what was not real.

With a not satisfied result, a new multimeter was used. With the digital multimeter Fluke 158 the reference tension reading was 2.99 and the resistances Ra, Rb and Rc had measured 4502, 818 and 4702 respectively. With these resistance values, the eHealth read a satisfactory measure, with next 0.5 ºC variations (Table 3).

Table 3: Test with resistances calibrated.

	Test 1	Test 2	Test 3
P1	35,58	36,34	36,74
P2	35,38	36,40	36,22
P3	35,66	36,42	36,37
P4	35,34	35,32	36,85
P5	36,46	36,23	36,73

Fonte: Autores, (2017).

The conditions of normal temperature, hypothermia, fever and hyperpyrexia was adjusted on the program based in the table 1 values.

As result was generated a monitoring screen which shows the current sensor temperature value in contact with a patient. A start button is used to initialize the monitoring. After press button, the temperature start to be showed on the screen (Figure 5). If the temperature is different of the standard setted, an alarm is showed on the screen indicating if the patient is hypothermic, fever or hyperpyxia.

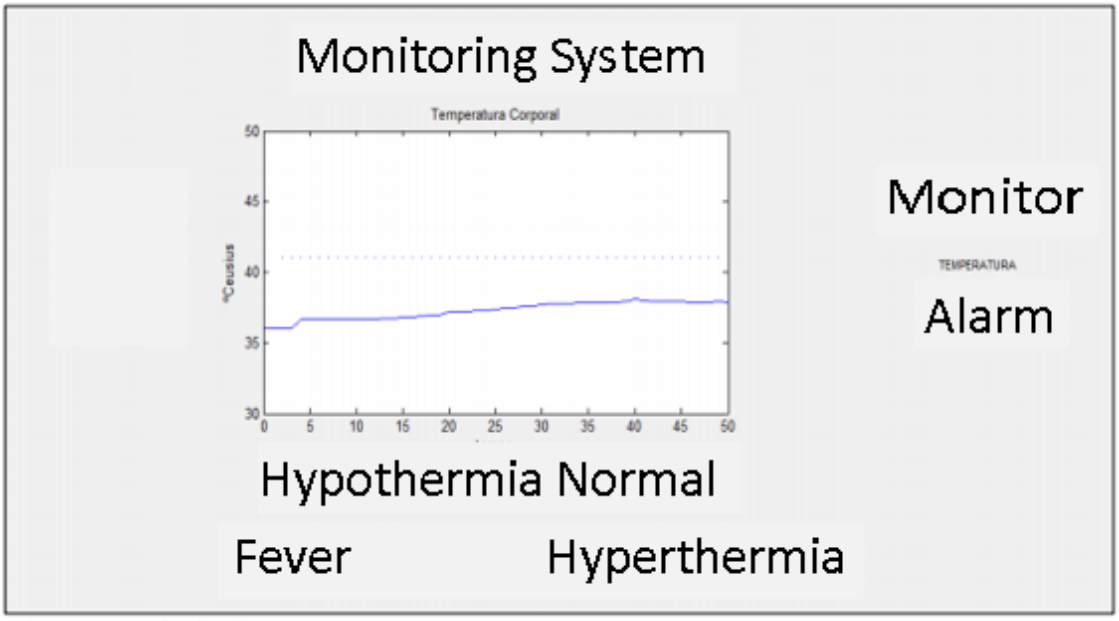

Figure 5: Monitoring Screen.
Fonte: Autores, (2017).

IV. CONCLUSION

The test results show that is necessary make a great calibration, because even the lower variation in the tension and resistance values can cause noise in measure reading. A auxiliary digital thermometer is needed to verify and compare with the real temperature, getting a better reliability in the data read. Thereat is conclusive that is possible a satisfactory monitoring of a patient temperature by the eHealth data read, proving the functionary of the temperature sensor and being extended to the other eHealth sensors with the due calibrations and data comparision

V. ACKNOWLEDGEMENTS

The study was partially supported by UEA (University of State of Amazonas).

VI. REFERENCES

[1] COOKING-HACKS. **eHealth Sensor Platform v2.**0. 2013. Disponível em: < https://www.cooking-hacks.com/documentation/tutorials/ehealth-biometric-sensor-platform-arduino-raspberry-pi-medical>

[2] HOCHENBAUM JOSHUA NOBLE, M. E. J. **Arduino em Ação**. [S.l.]: Novatec Editora, 2013.

[3] MCROBERTS, M. **Arduino Básico**. [S.l.]: Novatec, 2011. SALEIRO, E. E. M. Zigbee, Uma abordagem prática. 2014.