Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 86 (2016) 160 – 163

Procedia Computer Science

2016 International Electrical Engineering Congress, iEECON2016, 2-4 March 2016, Chiang Mai, Thailand

Wireless Heart Rate Monitoring System using MQTT

Komkrit Chooruang[a] *, Pongpat Mangkalakeeree[a]

[a]*Faculty of Industrial Technology, Nakhon Phanom University, Nakhon Phanom, Thailand*

Abstract

This paper presents the heart rate monitoring system using ESP8266 Wi-Fi module on the Arduino microcontroller and Message Queuing Telemetry Transport (MQTT) for messaging protocol that designed for lightweight communications. The developed system is intended to remotely monitor the real-time heart rate of a patient. This design use simple infrared light and photo detector to detect and pick up the rate of heart beat signal and send measured data wirelessly to the MQTT broker where running on a Raspberry pi, a low-cost, credit-card sized computer. The preliminary result demonstrated that the system provides useful information and helpful for nursing related health care tasks.

Peer-review under responsibility of the Organizing Committee of iEECON2016

Keywords: Heart rate monitor; Message Queuing Telemetry Transport (MQTT); Wireless sensor network

1. Introduction

The heart rate monitor can be used in several ways and applications such as in the hospitals, elderly health care, personal emergency response [1, 2] or sport training [3]. In the developing countries or rural hospitals, health care budgets are very limited and increasing expenditure is affected to the quality of service in a small hospital. Monitoring of heart rate is essential for real-time information that would allow emergency detection and for evaluating the risk of heart failure. However, commercial heart rate monitors are expensive and not every hospital can afford it. There are some researchers have demonstrating a low-cost heart rate monitor and adopting wireless technology into this kind of system using ZigBee, XBEE, Bluetooth and RF modules [1, 3-5]. However, some of those designs still required high power, time consuming and big data overhead for communication protocol. In this

* Corresponding author. Tel.: +6-604-253-2477; fax: +6-604-253-2479.
E-mail address: komkritc@npu.ac.th

Peer-review under responsibility of the Organizing Committee of iEECON2016
doi:10.1016/j.procs.2016.05.045

work, we aim to build inexpensive prototype of simple wireless sensor network for heart rate monitoring system based on Wi-Fi connectivity and utilize a light weight MQTT for a publish-subscribe based messaging protocol.

2. Methodology

The system mainly includes 4 parts: a heart rate sensor module, a microcontroller equipped with ESP8266 Wi-Fi, MQTT on the Raspberry pi and monitoring software. Figure 1 illustrates the block diagram of the system. Figure 2 shows the heart rate sensor and its schematic that is well built to fit over a patient's fingertips and reads the amount of infrared light reflected by the blood circulating inside of patient's body. When the heart pump, the blood pressure rises sharply and so the amount of infrared light from the emitter's LED that get increased and reflected back to the photo detector. The photo detector passes more current when it get more reflected light and then become a voltage drop. The two consecutive operational amplifiers are used to establish a signal baseline and emphasize the peaks and filter out the noise. The filtered signal will be read by analog-to-digital pin on the Arduino microcontroller board.

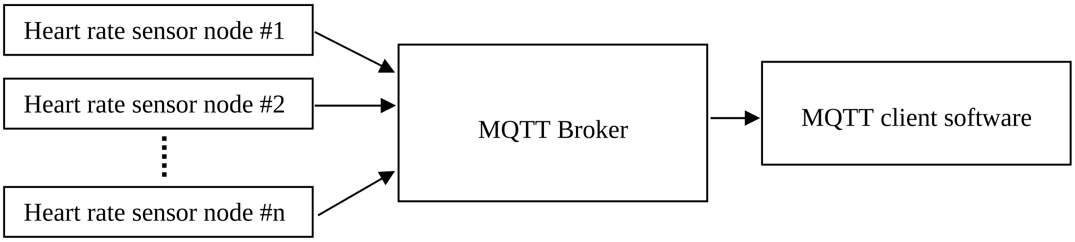
Heart rate sensor node #1
Heart rate sensor node #2
Heart rate sensor node #n
MQTT Broker
MQTT client software

Fig. 1 Block diagram of the system

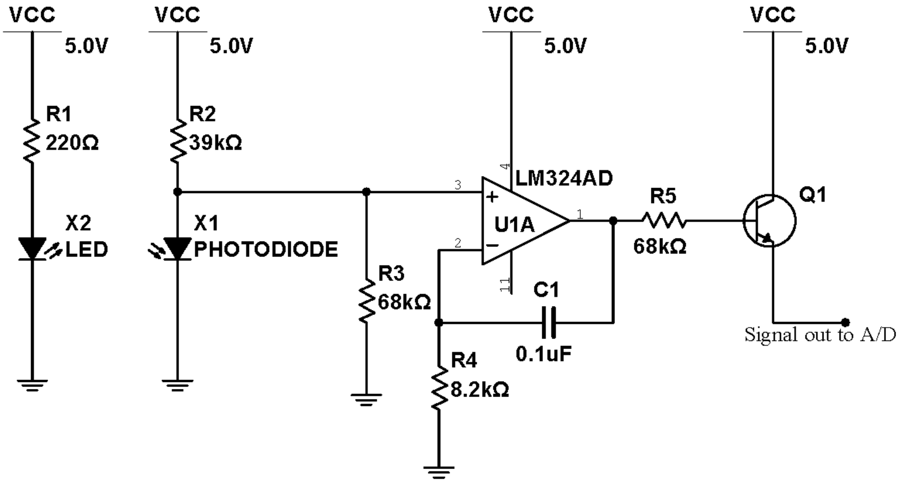

Fig. 2 Schematic of heart rate sensor

The ESP8266 is an integrated chip designed for the needs of an increasingly internet of things devices. It provides a complete self-contained Wi-Fi networking capabilities, allowing it to act as host applications or offload all Wi-Fi networking functions. This module is installed on the Arduino microcontroller board for adding the Wi-Fi functionality. In every minute, the Arduino is reading the heart rate pulse from the sensors that connected to its internal A/D and calculate the heart rate and then update to the server which is connected on the same network.

The conventional HTTP protocol is great for doing a request and getting response such as when the clients want to ask some information on the server and then get the answer back but it is not the best solution when the source information should push a change into many clients and there is no built-in support for quality of service. As a result, the conventional HTTP protocol required more bandwidth and need to keep sever alive to answer incoming request, does take a lot of power consumption. Therefore, the HTTP protocol does not meet our requirements to the battery operated devices. Thus, Message Queuing Telemetry Transport (MQTT) [6] is chosen as the protocol for this work because it does required small code footprint and easy to scale the sensor nodes so it is very suited for the condition where the network bandwidth is limited. MQTT is publish-subscribe based messaging protocol and provide loose coupling to support dynamic system environments where high volumes of messages and sensors or events need to be made available. It also provides multiple deterministic message delivery qualities of service to reflect tradeoffs between bandwidth, availability and delivery guarantees. The MQTT clients ideally can be handled up to 10,000 clients and simple for application developers in implementing the communication protocol.

The MQTT software or broker in this system is the Mosquitto software [7] which is an open source implementation of an MQTT broker. The Mosquitto software is installed on a credit-card sized computer, the Raspberry Pi. The multi-user or nurses can retrieve and monitor the patient's heart rate data easily using MQTT client software developed in Node.js for our initial evaluation. The server can be configured for public access that allows remotely monitoring at anywhere through the Internet.

3. Results and Discussions

The heart rate from the sensor modules were measured in beats per minute (BPM) for five subjects then compared with the heart rate use the fingers to do the measuring manually and the percentage error of those reading was determined. It was found that the percentage error was in the range of about 2%-6% as shown in Table 1.

Table 1

No.	Heart Rate (BPM) (sensor module)	Heart Rate (BPM) (manual measurement)	Error (%)
1	75	73	2.74%
2	66	68	2.94%
3	68	72	5.56%
4	62	65	4.62%
5	67	70	4.29%

The maximum distance of the sensor nodes that can communicate wirelessly to the server (Raspberry Pi) was investigated. The router in our test was the D-LINK model DIR-605L. The Raspberry Pi and sensor nodes, the ESP8266 with built-in antenna were connected on the router, thus there were on the same network that allow them to communicate to each other. The developed application in Node.js was used in the software part for displaying the monitored heart rate and the modifications can be easily revised for different needs or environments of the users. In Figure 3, it shows the decay of the received signal strength indicator or RSSI in dBm with increasing distance of sensor nodes. The maximum distance was about 28 meters at -88 dBm. However, the distance can be extended using Wi-Fi repeater or using the high power router.

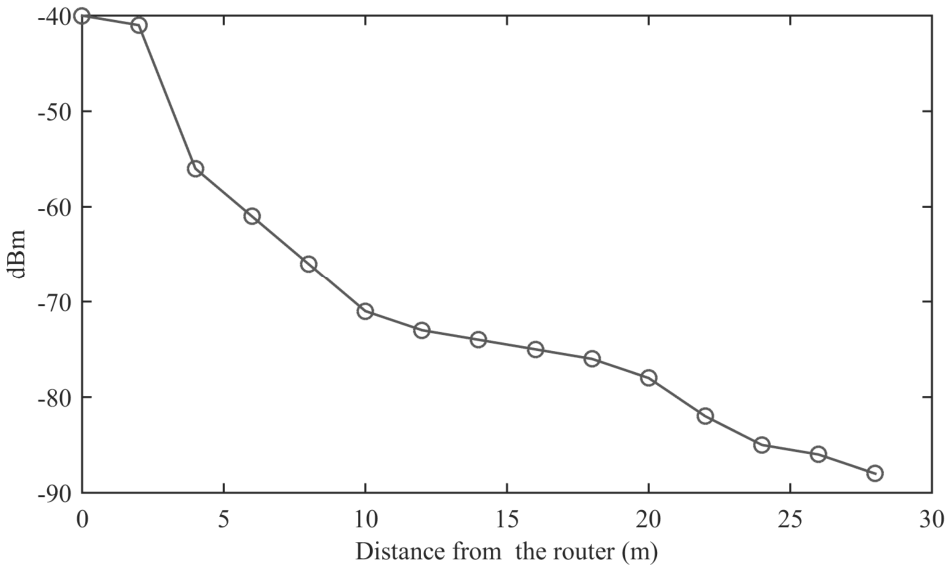

Fig. 3 Received signal strength indicator

4. Conclusion

We proposed a wireless heart rate monitor system with wireless communication using infrared heart rate sensor, ESP8266 Wi-Fi module on the Arduino and Mosquitto-MQTT broker on the Raspberry Pi. The result showed that it is able to measure, transmit and receive the heart rate data of the patients in real-time. The low power consumption and low bandwidth of MQTT protocol makes this system practical for patients and low-cost that quite suitable for small or rural hospitals.

References

1. D. L. Larkai and W. Ruiheng, "Wireless Heart Rate Monitor in Personal Emergency Response System," in Design and Diagnostics of Electronic Circuits & Systems (DDECS), 2015 IEEE 18th International Symposium on, 2015, pp. 299-300.
2. L. Tae-Soo, H. Joo-Hyun, and C. Myeong-Chan, "Biomedical Digital Assistant for Ubiquitous Healthcare," in Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE, 2007, pp. 1790-1793.
3. N. H. Mahmood, N. Uyop, N. Zulkarnain, F. K. C. Harun, M. F. Kamarudin, and A. Linoby, "LED indicator for heart rate monitoring system in sport application," in Signal Processing and its Applications, 2011 IEEE 7th International Colloquium on, 2011, pp. 64-66.
4. N. S. A. Zulkifli, F. K. C. Harun, and N. S. Azahar, "XBee wireless sensor networks for Heart Rate Monitoring in sport training," in Biomedical Engineering (ICoBE), 2012 International Conference on, 2012, pp. 441-444.
5. G.-c. Liu and H.-y. Yu, "Design and implementation of a Bluetooth 4.0-based heart rate monitor system on iOS platform," in Communications, Circuits and Systems (ICCCAS), 2013 International Conference on, 2013, pp. 112-115.
6. Dave Locke , MQ telemetry transport (mqtt) v3. 1 protocol specification. IBM developerWorks Technical Library] , 2010 [online] Available: http://www.ibm.com/developerworks/webservices/library/ws-mqtt/index.html
7. Mosquitto - An Open Source MQTT Broker. Available: http://mosquitto.org/